

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 14, 2012

G. Bumgardner  
Cisco  
June 12, 2012

Automatic Multicast Tunneling  
draft-ietf-mboned-auto-multicast-14

Abstract

This document describes Automatic Multicast Tunneling (AMT), a protocol for delivering multicast traffic from sources in a multicast-enabled network to receivers that lack multicast connectivity to the source network. The protocol uses UDP encapsulation and unicast replication to provide this functionality.

The AMT protocol is specifically designed to support rapid deployment by requiring minimal changes to existing network infrastructure.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on December 14, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

## Table of Contents

1. Introduction . . . . .	3
2. Applicability . . . . .	4
3. Terminology . . . . .	5
3.1. Requirements Notation . . . . .	5
3.2. Definitions . . . . .	5
3.3. Abbreviations . . . . .	6
4. Protocol Overview . . . . .	8
4.1. General Architecture . . . . .	8
4.2. General Operation . . . . .	17
5. Protocol Description . . . . .	32
5.1. Protocol Messages . . . . .	32
5.2. Gateway Operation . . . . .	47
5.3. Relay Operation . . . . .	62
6. Security Considerations . . . . .	73
7. IANA Considerations . . . . .	76
7.2. IPv4 Address Prefix Allocation for IGMP Source Addresses . . . . .	76
8. Contributors . . . . .	77
9. Acknowledgments . . . . .	78
10. References . . . . .	79
10.1. Normative References . . . . .	79
10.2. Informative References . . . . .	79
Appendix A. Implementation Notes . . . . .	82
Author's Address . . . . .	85

## 1. Introduction

The advantages and benefits provided by multicast technologies are well known. There are a number of application areas that are ideal candidates for the use of multicast, including media broadcasting, video conferencing, collaboration, real-time data feeds, data replication, and software updates. Unfortunately, many of these applications lack multicast connectivity to networks that carry traffic generated by multicast sources. The reasons for the lack of connectivity vary, but are primarily the result of service provider policies and network limitations.

Automatic Multicast Tunneling (AMT) is a protocol that uses UDP-based encapsulation to overcome the aforementioned lack of multicast connectivity. AMT enables sites, hosts or applications that do not have native multicast access to a network with multicast connectivity to a source, to request and receive SSM [RFC4607] and ASM [RFC1112] traffic from a network that does provide multicast connectivity to that source.

## 2. Applicability

This document describes a protocol that may be used to deliver multicast traffic from a multicast enabled network to sites that lack multicast connectivity to the source network. This document does not describe any methods for sourcing multicast traffic from isolated sites as this topic is out of scope.

AMT is not intended to be used as a substitute for native multicast, especially in conditions or environments requiring high traffic flow. AMT uses unicast replication to reach multiple receivers and the bandwidth cost for this replication will be higher than that required if the receivers were reachable via native multicast.

### 3. Terminology

#### 3.1. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

#### 3.2. Definitions

This document adopts the following definitions for use in describing the protocol:

**Downstream:**

A downstream interface or connection that faces away from the multicast distribution root or towards multicast receivers.

**Upstream:**

An upstream interface or connection that faces a multicast distribution root or source.

**Non-Broadcast Multi-Access (NBMA):**

A non-broadcast multiple-access (NBMA) network or interface is one to which multiple network nodes (hosts or routers) are attached, but where packets are transmitted directly from one node to another node over a virtual circuit or physical link. NBMA networks do not support multicast or broadcast traffic - a node that sources multicast traffic must replicate the multicast packets for separate transmission to each node that has requested the multicast traffic.

**Multicast Receiver:**

An entity that requests and receives multicast traffic. A receiver may be a router, host, application, or application component. The method by which a receiver transmits group membership requests and receives multicast traffic varies according to receiver type.

**Group Membership Database:**

A group membership database describes the current multicast subscription/reception state for an interface or system.

**Reception State:**

The multicast subscription state of a pseudo, virtual or physical network interface. See group membership database.

**Subscription:**

A group or state entry in a group membership database or reception state table.

**Group Membership Protocol:**

The term "group membership protocol" is used as a generic reference to the Internet Group Management (IGMP) ([RFC1112], [RFC2236], [RFC3376]) or Multicast Listener Discovery ([RFC2710], [RFC3810]) protocols.

**Multicast Protocol:**

The term "multicast protocol" is used as a generic reference to multicast routing protocols used to join or leave multicast distribution trees such as PIM-SM [RFC4601].

**Network Address Translation (NAT):**

Network Address Translation is the process of modifying the source IP address and port numbers carried by an IP packet while transiting a network node (See [RFC2663]). Intervening NAT devices may change the source address and port carried by messages sent from an AMT gateway to an AMT relay, possibly producing changes in protocol state and behavior.

**Anycast:**

A network addressing and routing method in which packets from a single sender are routed to the topologically nearest node in a group of potential receivers all identified by the same destination address. See [RFC4786].

**3.3. Abbreviations**

AMT - Automatic Multicast Tunneling Protocol.

ASM - Any-Source Multicast.

DoS - Denial-of-Service (attack) and DDoS for distributed-DoS.

IGMP - Internet Group Management Protocol (v1, v2 and v3).

IP - Internet Protocol (v4 and v6).

MAC - Message Authentication Code (or Cookie).

MLD - Multicast Listener Discovery protocol (v1 and v2).

NAT - Network Address Translation (or translation node).

NBMA - Non-Broadcast Multi-Access (network, interface or mode)

SSM - Source-Specific Multicast.

PIM - Protocol Independent Multicast.

#### 4. Protocol Overview

This section provides an informative description of the protocol. A normative description of the protocol and implementation requirements may be found in section Section 5.

##### 4.1. General Architecture

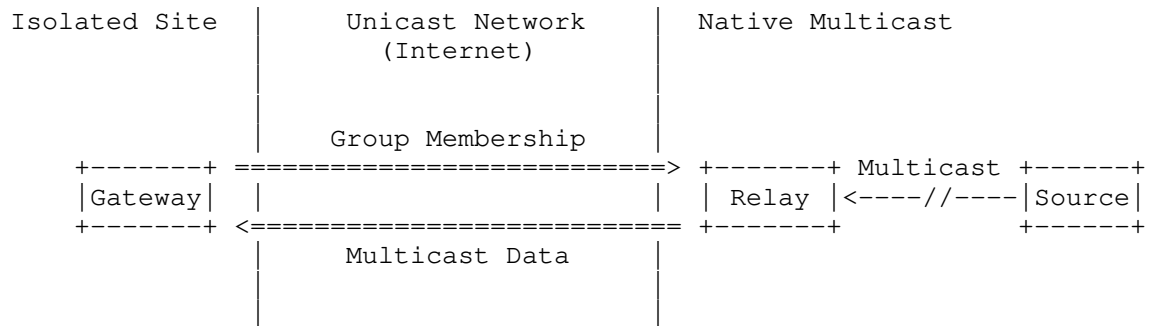


Figure 1: Basic AMT Architecture

The AMT protocol employs a client-server model in which a "gateway" sends requests to receive specific multicast traffic to a "relay" which responds by delivering the requested multicast traffic back to the gateway.

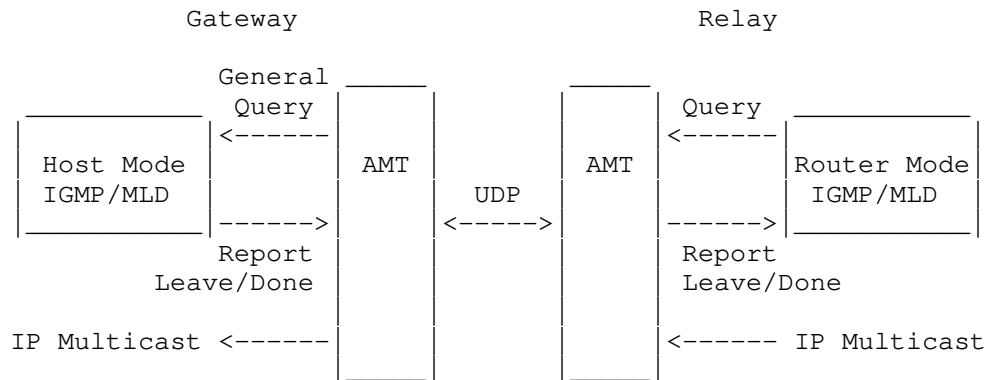
Gateways are generally deployed within networks that lack multicast support or lack connectivity to a multicast-enabled network containing multicast sources of interest.

Relays are deployed within multicast-enabled networks that contain, or have connectivity to, multicast sources.

##### 4.1.1. Relationship to IGMP and MLD Protocols

AMT relies on the Internet Group Management (IGMP) [RFC3376] and Multicast Listener Discovery (MLD) [RFC3810] protocols to provide the functionality required to manage, communicate, and act on changes in multicast group membership. A gateway or relay implementation does not necessarily require a fully-functional, conforming implementation of IGMP or MLD to adhere to this specification, but the protocol description that appears in this document assumes that this is the case. The minimum functional and behavioral requirements for the IGMP and MLD protocols are described in Section 5.2.1 and Section 5.3.1.





Multicast Reception State Managed By IGMP/MLD

A gateway runs the host portion of the IGMP and MLD protocols to generate group membership updates that are sent via AMT messages to a relay. A relay runs the router portion of the IGMP and MLD protocols to process the group membership updates to produce the required changes in multicast forwarding state. A relay uses AMT messages to send incoming multicast IP datagrams to gateways according to their current group membership state.

The primary function of AMT is to provide the handshaking, encapsulation and decapsulation required to transport the IGMP and MLD messages and multicast IP datagrams between the gateways and relays. The IGMP and MLD messages that are exchanged between gateways and relays are encapsulated as complete IP datagrams within AMT control messages. Multicast IP datagrams are replicated and encapsulated in AMT data messages. All AMT messages are sent via unicast UDP/IP.

#### 4.1.2. Gateways

The downstream side of a gateway services one or more receivers - the gateway accepts group membership requests from receivers and forwards requested multicast traffic back to those receivers.

The upstream side of a gateway connects to relays. A gateway sends encapsulated IGMP and MLD messages to a relay to indicate an interest in receiving specific multicast traffic.

## 4.1.2.1. Architecture

Each gateway possesses a logical pseudo-interface:

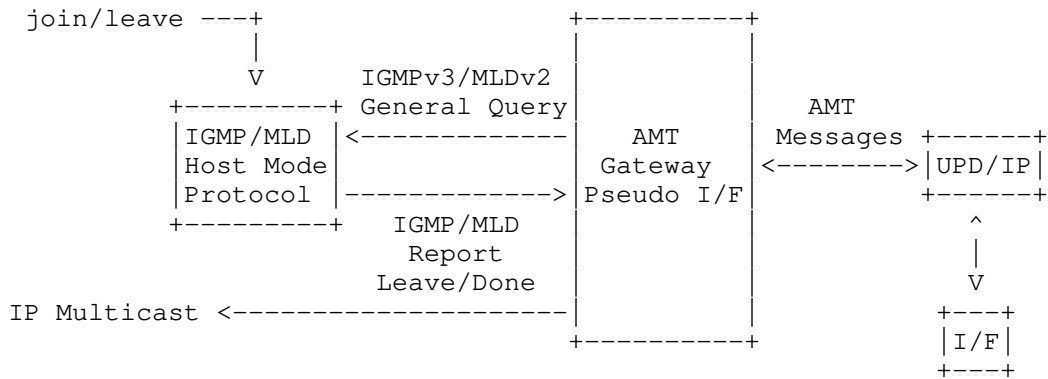


Figure 2: AMT Gateway Pseudo-Interface

The pseudo-interface is conceptually a network interface on which the gateway executes the host portion of the IPv4/IGMP (v2 or v3) and IPv6/MLD (v1 or v2) protocols. The multicast reception state of the pseudo-interface is manipulated using the IGMP or MLD service interface. The IGMP and MLD host protocols produce IP datagrams containing group membership messages that the gateway will send to the relay. The IGMP and MLD protocols also supply the retransmission and timing behavior required for protocol robustness.

All AMT encapsulation, decapsulation and relay interaction is assumed to occur within the pseudo-interface.

A gateway host or application may create separate interfaces for IPv4/IGMP and IPv6/MLD. A gateway host or application may also require additional pseudo-interfaces for each source or domain-specific relay address.

Within this document, the term "gateway" may be used as a generic reference to an entity executing the gateway protocol, a gateway pseudo-interface, or a gateway device that has one or more interfaces connected to a unicast inter-network and one or more AMT gateway pseudo-interfaces.



#### 4.1.2.2. Use-Cases

Use-cases for gateway functionality include:

##### IGMP/MLD Proxy

An IGMP/MLD proxy that runs AMT on an upstream interface and router-mode IGMP/MLD on downstream interfaces to provide host access to multicast traffic via the IGMP and MLD protocols.

##### Virtual Network Interface

A virtual network interface or pseudo network device driver that runs AMT on a physical network interface to provide socket layer access to multicast traffic via the IGMP/MLD service interface provided by the host IP stack.

##### Application

An application or application component that implements and executes IGMP/MLD and AMT internally to gain access to multicast traffic.

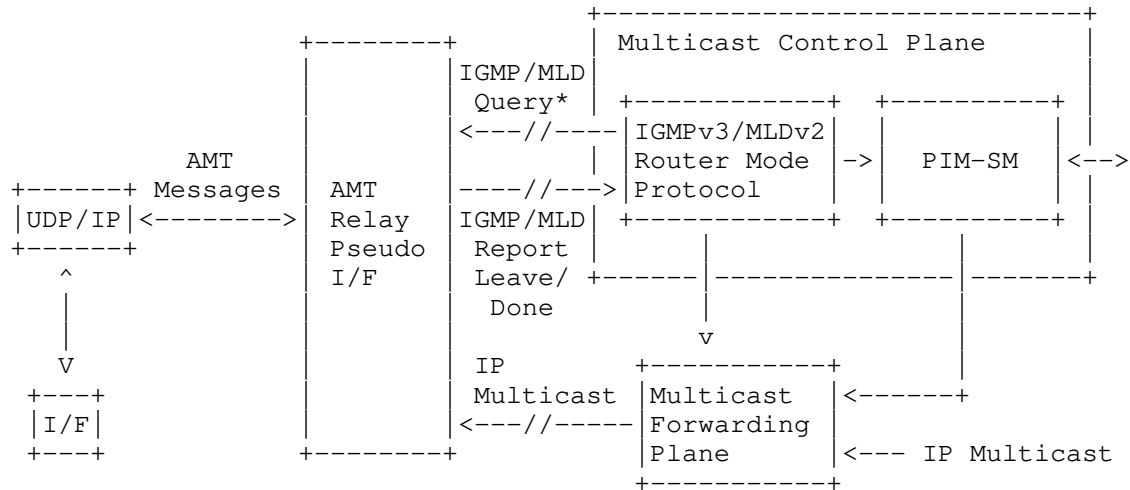
#### 4.1.3. Relays

The downstream side of a relay services gateways - the relay accepts encapsulated IGMP and MLD group membership messages from gateways and encapsulates and forwards the requested multicast traffic back to those gateways.

The upstream side of a relay communicates with a native multicast infrastructure - the relay sends join and prune/leave requests towards multicast sources and accepts requested multicast traffic from those sources.

## 4.1.3.1. Architecture

Each relay possesses a logical pseudo-interface:



\* Queries, if generated, are consumed by the pseudo-interface.

## AMT Relay Pseudo-Interface (Router-Based)

The pseudo-interface is conceptually a network interface on which the relay runs the router portion of the IPv4/IGMPv3 and IPv6/MLDv2 protocols. Relays do not send unsolicited IGMPv3/MLDv2 query messages to gateways so relays must consume or discard any local queries normally generated by IGMPv3 or MLDv2.

A relay maintains group membership state for each gateway connected through the pseudo-interface as well as for the entire pseudo-interface (if multiple gateways are managed via a single interface). Multicast packets received on upstream interfaces on the relay are routed to the pseudo-interface where they are replicated, encapsulated and sent to interested gateways. Changes in the pseudo-interface group membership state may trigger the transmission of multicast protocol requests upstream towards a given source or rendezvous point and cause changes in internal routing/forwarding state.

The relay pseudo-interface is a architectural abstraction used to describe AMT protocol operation. For the purposes of this document, the pseudo-interface is most easily viewed as an interface to a single gateway - encapsulation, decapsulation, and other AMT-specific processing occurs "within" the pseudo-interface while forwarding and

replication occur outside of it.

An alternative view is to treat the pseudo-interface as a non-broadcast multi-access (NBMA) network interface whose link layer is the unicast-only network over which AMT messages are exchanged with gateways. Individual gateways are conceptually treated as logical NBMA links on the interface. In this architectural model, group membership tracking, replication and forwarding functions occur in the pseudo-interface.

This document does not specify any particular architectural solution - a relay developer may choose to implement and distribute protocol functionality as required to take advantage of existing relay platform services and architecture.

Within this document, the term "relay" may be used as a generic reference to an entity executing the relay protocol, a relay pseudo-interface, or a relay device that has one or more network interfaces with multicast connectivity to a native multicast infrastructure, zero or more interfaces connected to a unicast inter-network, and one or more relay pseudo-interfaces.

#### 4.1.3.2. Use-Cases

Use-cases for relay functionality include:

##### Multicast Router

A multicast router that runs AMT on a downstream interface to provide gateway access to multicast traffic. A "relay router" uses a multicast routing protocol (e.g. PIM-SM RFC4601 [RFC4601]) to construct a forwarding path for multicast traffic by sending join and prune messages to neighboring routers to join or leave multicast distribution trees for a given SSM source or ASM rendezvous point.

##### IGMP/MLD Proxy Router

An IGMP/MLD proxy that runs AMT on a downstream interface and host-mode IGMPv3/MLDv2 on a upstream interface. This "relay proxy" sends group membership reports to a local, multicast-enabled router to join and leave specific SSM or ASM groups.

#### 4.1.4. Deployment

The AMT protocol calls for a relay deployment model that uses anycast addressing [RFC1546][RFC4291] to pair gateways with relays.

Under this approach, one or more relays advertise a route for the same IP address prefix. To find a relay with which to communicate, a

gateway sends a message to an anycast IP address within that prefix. This message is routed to the topologically-nearest relay that has advertised the prefix. The relay that receives the message responds by sending its unicast address back to the gateway. The gateway uses this address as the destination address for any messages it subsequently sends to the relay.

The use of anycast addressing provides the following benefits:

- o Relays may be deployed at multiple locations within a single multicast-enabled network. Relays might be installed "near" gateways to reduce bandwidth requirements, latency and limit the number of gateways that might be serviced by a single relay.
- o Relays may be added or removed at any time thereby allowing staged deployment, scaling and hot-swapping - the relay discovery process will always return the nearest operational relay.
- o Relays may take themselves offline when they exhaust resources required to service additional gateways. Existing gateway connections may be preserved, but new gateway requests would be routed to the next-nearest relay.

#### 4.1.4.1. Public Versus Private

Ideally, the AMT protocol would provide a universal solution for connecting receivers to multicast sources - that any gateway could be used to access any globally advertised multicast source via publicly-accessible, widely-deployed relays. Unfortunately, today's Internet does not yet allow this, because many relays will lack native multicast access to sources even though they may be globally accessible via unicast.

In these cases, a provider may deploy relays within their own source network to allow for multicast distribution within that network. Gateways that use these relays must use a provider-specific relay discovery mechanism or a private anycast address to obtain access to these relays.

#### 4.1.5. Discovery

To execute the gateway portion of the protocol, a gateway requires a unicast IP address of an operational relay. This address may be obtained using a number of methods - it may be statically assigned or dynamically chosen via some form of relay discovery process.

As described in the previous section, the AMT protocol provides a relay discovery method that relies on anycast addressing. Gateways

are not required to use AMT relay discovery, but all relay implementations must support it.

The AMT protocol uses the following terminology when describing the discovery process:

**Relay Discovery Address Prefix:**

The anycast address prefix used to route discovery messages to a relay.

**Relay Discovery Address:**

The anycast destination address used when sending discovery messages.

**Relay Address:**

The unicast IP address obtained as a result of the discovery process.

#### 4.1.5.1. Relay Discovery Address Selection

The selection of an anycast Relay Discovery Address may be source-dependent, as a relay located via relay discovery must have multicast connectivity to a desired source.

Similarly, the selection of a unicast Relay address may be source-dependent, as a relay contacted by a gateway to supply multicast traffic must have native multicast connectivity to the traffic source

Methods that might be used to perform source-specific or group-specific relay selection are highly implementation-dependent and are not further addressed by this document. Possible approaches include the use of static lookup tables, DNS-based queries, or a provision of a service interface that accepts join requests on (S,G,relay-discovery-address) or (S,G,relay-address) tuples.

#### 4.1.5.2. IANA-Assigned Relay Discovery Address Prefix

This document calls for IANA to allocate an anycast address prefix for use in advertising and discovering publicly accessible relays.

A relay discovery address is constructed from the anycast address prefix by setting the low-order octet of the prefix address to 1 (for both IPv4 and IPv6).

Public relays must advertise a route to the anycast address prefix and configure an interface to respond to the relay discovery address.

The IANA address assignments are discussed in Section 7.



## 4.2. General Operation

### 4.2.1. Message Sequences

The AMT protocol defines the following messages for control and encapsulation. These messages are exchanged as UDP/IP datagrams, one message per datagram.

**Relay Discovery:**

Sent by gateways to solicit a Relay Advertisement from any relay.  
Used to find a relay with which to communicate.

**Relay Advertisement:**

Sent by relays as a response to a Relay Discovery message. Used to deliver a relay address to a gateway.

**Request:**

Sent by gateways to solicit a Membership Query message from a relay.

**Membership Query:**

Sent by relays as a response to a Request message. Used to deliver an encapsulated IGMPv3 or MLDv2 query message to the gateway.

**Membership Update:**

Sent by gateways to deliver an encapsulated IGMP or MLD report/leave/done message to a relay.

**Multicast Data:**

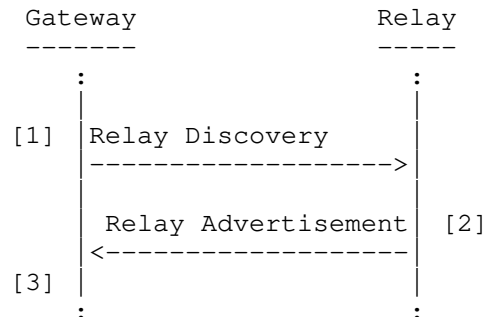
Sent by relays to deliver an encapsulated IP multicast datagram or datagram fragment to a gateway.

**Teardown:**

Sent by gateways to stop the delivery of Multicast Data messages requested in an earlier Membership Update message.

The following sections describe how these messages are exchanged to execute the protocol.

## 4.2.1.1. Relay Discovery Sequence



AMT Relay Discovery Sequence

The following sequence describes how the Relay Discovery and Relay Advertisement messages are used to find a relay with which to communicate:

1. The gateway sends a Relay Discovery message containing a random nonce to the Relay Discovery Address. If the Relay Discovery Address is an anycast address, the message is routed to topologically-nearest network node that advertises that address.
2. The node receiving the Relay Discovery message sends a Relay Advertisement message back to the source of the Relay Discovery message. The message carries a copy of the nonce contained in the Relay Discovery message and the unicast IP address of a relay.
3. When the gateway receives the Relay Advertisement message it verifies that the nonce matches the one sent in the Relay Discovery message, and if it does, uses the relay address carried by the Relay Advertisement as the destination address for subsequent AMT messages.

Note that the responder need not be a relay - the responder may obtain a relay address by some other means and return the result in the Relay Advertisement (i.e., the responder is a load-balancer or broker).

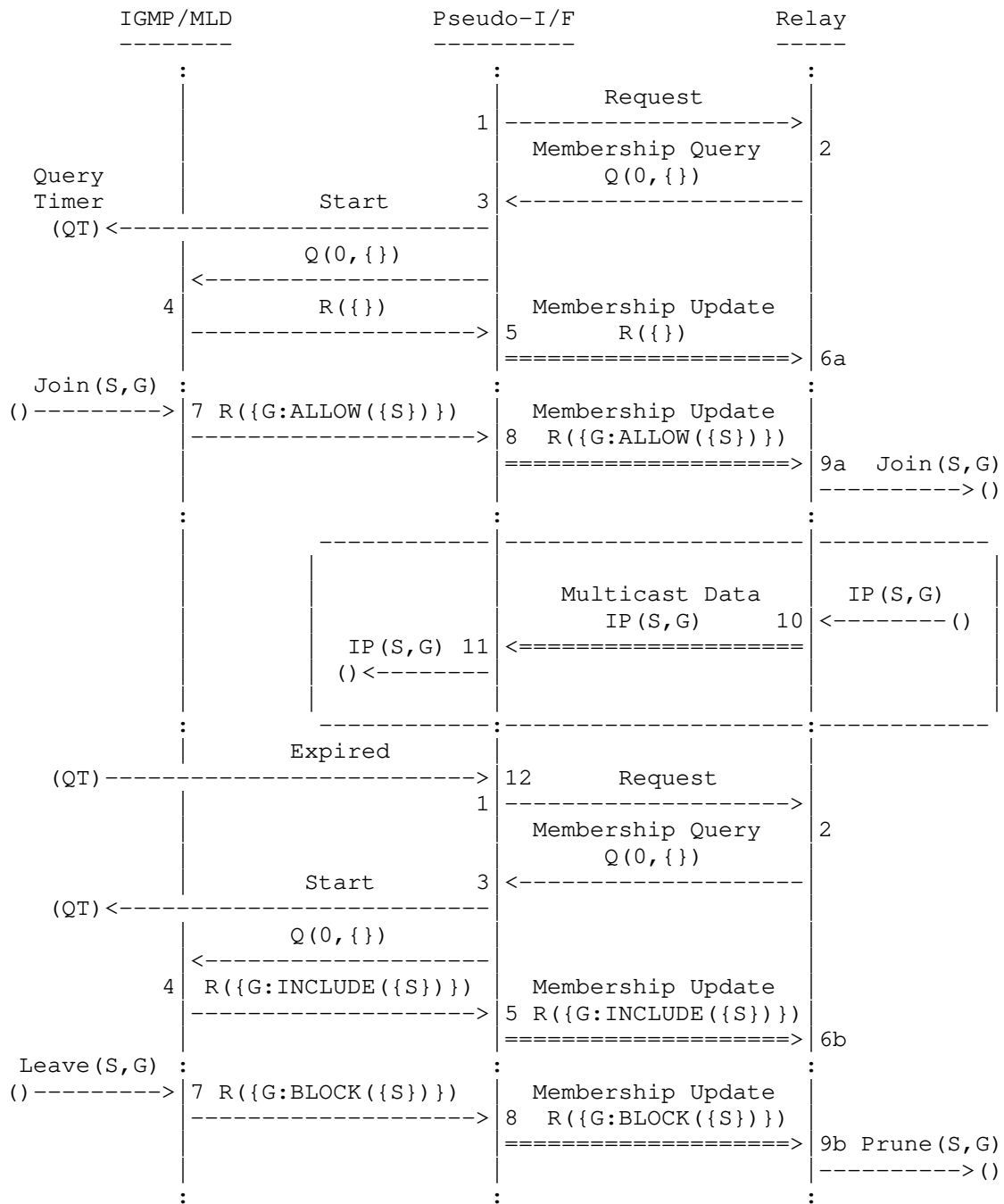
## 4.2.1.2. Membership Update Sequence

There exists a significant difference between normal IGMP and MLD behavior and that required by AMT. An IGMP/MLD router acting as a querier normally transmits query messages on a network interface to construct and refresh group membership state for the connected

network. These query messages are multicast to all IGMP/MLD enabled hosts on the network. Each host responds by multicasting report messages that describe their current multicast reception state.

However, AMT does not allow relays to send unsolicited query messages to gateways, as the set of active gateways may be unknown to the relay and potentially quite large. Instead, AMT requires each gateway to periodically send a message to a relay to solicit a general-query response. A gateway accomplishes this by sending a Request message to a relay. The relay responds by sending Membership Query message back to the gateway. The Membership Query message carries an encapsulated general query that is processed by the IGMP or MLD protocol implementation on the gateway to produce a membership/listener report. Each time the gateway receives a Membership Query message it starts a timer whose expiration will trigger the start of a new Request->Membership Query message exchange. This timer-driven sequence is used to mimic the transmission of a periodic general query by an IGMP/MLD router. This query cycle may continue indefinitely once started by sending the initial Request message.

A membership update occurs when an IGMP or MLD report, leave or done message is passed to the gateway pseudo-interface. These messages may be produced as a result of the aforementioned general-query processing or as a result of receiver interaction with the IGMP/MLD service interface. Each report is encapsulated and sent to the relay after the gateway has successfully established communication with the relay via a Request and Membership Query message exchange. If a report is passed to the pseudo-interface before the gateway has received a Membership Query message from the relay, the gateway may discard the report or queue the report for delivery after a Membership Query is received. Subsequent IGMP/MLD report/leave/done messages that are passed to the pseudo-interface are immediately encapsulated and transmitted to the relay.



Membership Update Sequence (IGMPv3/MLDv2 Example)

The following sequence describes how the Request, Membership Query, and Membership Update messages are used to report current group membership state or changes in group membership state:

1. A gateway sends a Request message to the relay that contains a random nonce and a flag indicating whether the relay should return an IGMPv3 or MLDv2 general query.
2. When the relay receives a Request message, it generates a message authentication code (MAC) by computing a hash value from a private secret and the nonce, source IP address, and source UDP port carried by the Request message. The relay then sends a Membership Query message to the gateway that contains the request nonce, the MAC, and an IGMPv3 or MLDv2 general query.
3. When the gateway receives a Membership Query message, it verifies that the request nonce matches the one sent in the last Request, and if it does, the gateway saves the request nonce and MAC for use in sending subsequent Membership Update messages. The gateway starts a timer whose expiration will trigger the transmission of a new Request message and extracts the encapsulated general query message for processing by the IGMP or MLD protocol. The query timer duration is specified by the relay in the QQIC field in the IGMPv3 or MLDv2 general query.
4. The gateway's IGMP or MLD protocol implementation processes the general query to produce a current-state report.
5. When an IGMP or MLD report is passed to the pseudo-interface, the gateway encapsulates the report in a Membership Update message and sends it to the relay. The request nonce and MAC fields in the Membership Update are assigned the values from the last Membership Query message received for the corresponding group membership protocol (IGMPv3 or MLDv2).
6. When the relay receives a Membership Update message, it computes a MAC from a private secret and the request nonce, source IP address, and source UDP port carried by the message. The relay accepts the Membership Update message if the received MAC matches the computed MAC, otherwise the message is ignored. If the message is accepted, the relay may proceed to allocate, refresh, or modify tunnel state. This includes making any group membership, routing and forwarding state changes and issuing any upstream protocol requests required to satisfy the state change. The diagram illustrates two scenarios:
  - A. The gateway has not previously reported any group subscriptions and the report does not contain any group

subscriptions, so the relay takes no action.

- B. The gateway has previously reported a group subscription so the current-state report lists all current subscriptions. The relay responds by refreshing tunnel or group state and resetting any related timers.
7. A receiver indicates to the gateway that it wishes to join (allow) or leave (block) specific multicast traffic. This request is typically made using some form IGMP/MLD service interface (as described in Section 2 of [RFC3376] or Section 3 of [RFC3810]). The IGMP/MLD protocol responds by generating an IGMP or MLD state-change message.
  8. When an IGMP or MLD report/leave/done message is passed to the pseudo-interface, the gateway encapsulates the message in a Membership Update message and sends it to the relay. The request nonce and MAC fields in the Membership Update are assigned the values from the last Membership Query message received for the corresponding group membership protocol (IGMP or MLD).

The IGMP and MLD protocols may generate multiple messages to provide robustness against packet loss - each of these must be encapsulated in a new Membership Update message and sent to the relay. The Querier Robustness Variable (QRV) field in the last IGMP/MLD query delivered to the IGMP/MLD protocol is typically used to specify the number of repetitions (i.e., the host adopts the QRV value as its own Robustness Variable value).

9. When the relay receives a Membership Update message, it again computes a MAC from a private secret and the request nonce, source IP address, and source UDP port carried by the message. The relay accepts the Membership Update message if the received MAC matches the computed MAC, otherwise the message is ignored. If the message is accepted, the relay processes the encapsulated IGMP/MLD and allocates, modifies or deletes tunnel state accordingly. This includes making any group membership, routing and forwarding state changes and issuing any upstream protocol requests required to satisfy the state change. The diagram illustrates two scenarios:
  - A. The gateway wishes to add a group subscription.
  - B. The gateway wishes to delete a previously reported group subscription.

10. Multicast datagrams transmitted from a source travel through the native multicast infrastructure to the relay. When the relay receives a multicast IP datagram that carries a source and destination address for which a gateway has expressed an interest in receiving (via the Membership Update message), it encapsulates the datagram into a Multicast Data message and sends it to the gateway using the source IP address and UDP port carried by the Membership Update message as the destination address.
11. When the gateway receives a Multicast Data message, it extracts the multicast packet from the message and passes it on to the appropriate receivers.
12. When the query timer expires the gateway sends a new Request message to the relay to start a new membership update cycle.

The MAC-based source-authentication mechanism described above provides a simple defense against malicious attempts to exhaust relay resources via source-address spoofing. Flooding a relay with spoofed Request or Membership Update messages may consume computational resources and network bandwidth, but will not result in the allocation of state because the Request message is stateless and spoofed Membership Update messages will fail source-authentication and be rejected by the relay.

A relay will only allocate new tunnel state if the IGMP/MLD report carried by the Membership Update message creates one or more group subscriptions.

A relay deallocates tunnel state after one of the following events; the gateway sends a Membership Update message containing a report that results in the deletion of all remaining group subscriptions, the IGMP/MLD state expires (due to lack of refresh by the gateway), or the relay receives a valid Teardown message from the gateway.

A gateway that accepts or reports group subscriptions for both IPv4 and IPv6 addresses will send separate Request and Membership Update messages for each protocol (IPv4/IGMP and IPv6/MLD).

#### 4.2.1.3. Teardown Sequence

A gateway sends a Teardown message to a relay to request that it stop delivering Multicast Data messages to a tunnel endpoint created by an earlier Membership Update message. This message is intended to be used following a gateway address change (See Section 4.2.2.1) to stop the transmission of undeliverable or duplicate multicast data messages. Support for the Teardown message is optional - gateways

are not required to send them and relays are not required to act upon them.



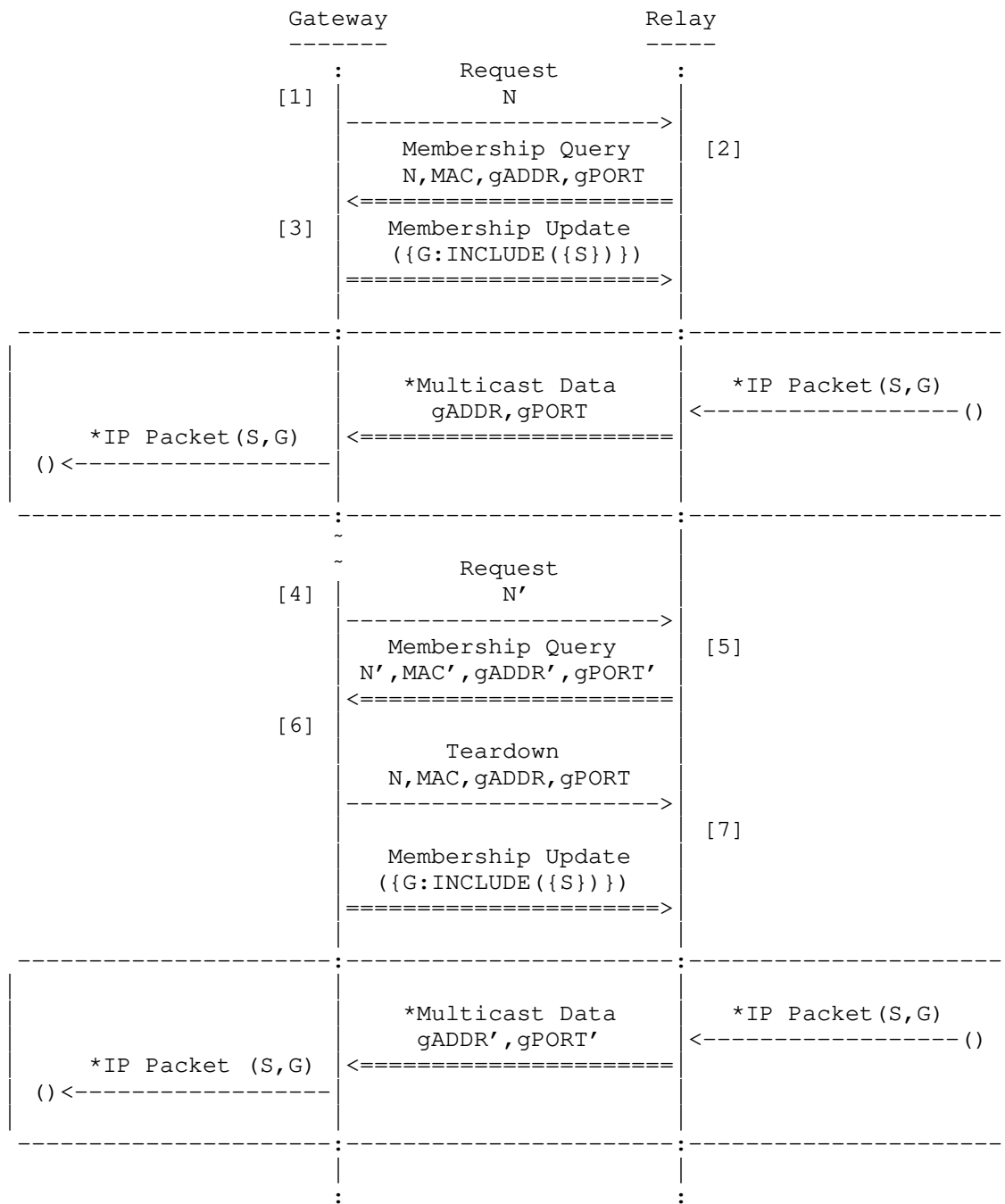


Figure 3: Teardown Message Sequence (IGMPv3/MLDv2 Example)

The following sequence describes how the Membership Query and Teardown message are used to detect an address change and stop the delivery of Multicast Data messages to an address:

1. A gateway sends a Request message containing a random nonce to the relay.
2. The relay sends a Membership Query message to the gateway that contains the source IP address (gADDR) and source UDP port (gPORT) values from the Request message. These values will be used to identify the tunnel should one be created by a subsequent Membership Update message.
3. When the gateway receives a Membership Query message that carries the gateway address fields, it compares the gateway IP address and port number values with those received in the previous Membership Query (if any). If these values do not match, this indicates that the Request message arrived at the relay carrying a different source address than the one sent previously. At this point in the sequence, no change in source address or port has occurred.
4. The gateway sends a new Request message to the relay. However, this Request message arrives at the relay carrying a different source address than that of the previous Request due to some change in network interface, address assignment, network topology or NAT mapping.
5. The relay again responds by sending a Membership Query message to the gateway that contains the new source IP address (gADDR') and source UDP port (gPORT') values from the Request message.
6. When the gateway receives the Membership Query message, it compares the gateway address and port number values against those returned in the previous Membership Query message.
7. If the reported address or port has changed, the gateway sends a Teardown message to the relay that contains the request nonce, MAC, gateway IP address and gateway port number returned in the earlier Membership Query message. The gateway may send the Teardown message multiple times where the number of repetitions is governed by the Querier Robustness Variable (QRV) value contained in the IGMPv3/MLDv2 general query carried by the original Membership Query. The gateway continues to process the new Membership Query message as usual.
8. When the relay receives a Teardown message, it computes a MAC from a private secret and the request nonce, gateway IP address,

and gateway port number carried by the Teardown message. The relay accepts the Teardown message if the received MAC matches the computed MAC, otherwise the message is ignored. If the message is accepted, the relay makes any group membership, routing and forwarding state changes required to stop the transmission of Multicast Data messages to that address.

#### 4.2.1.4. Timeout and Retransmission

The AMT protocol does not establish any requirements regarding what actions a gateway should take if it fails to receive a response from a relay. A gateway implementation may wait for an indefinite period of time to receive a response, may set a time limit on how long to wait for a response, may retransmit messages should the time limit be reached, may limit the number of retransmissions, or may simply report an error.

For example, a gateway may retransmit a Request message if it fails to receive a Membership Query or expected Multicast Data messages within some time period. If the gateway fails to receive any response to a Request after several retransmissions or within some maximum period of time, it may reenter the relay discovery phase in an attempt to find a new relay. This topic is addressed in more detail in Section 5.2.

#### 4.2.2. Tunneling

From the standpoint of a relay, an AMT "tunnel" is identified by the IP address and UDP port pair used as the destination address for sending encapsulated multicast IP datagrams to a gateway. This address is referred here as the tunnel endpoint address.

A gateway sends a Membership Update message to a relay to add or remove group subscriptions to a tunnel endpoint. The tunnel endpoint is identified by the source IP address and source UDP port carried by the Membership Update message when it arrives at a relay (this address may differ from that carried by the message when it exited the gateway as a result of network address translation).

The Membership Update messages sent by a single gateway host may originate from several source addresses or ports - each unique combination represents a unique tunnel endpoint. A single gateway host may legitimately create and accept traffic on multiple tunnel endpoints, e.g., the gateway may use separate ports for the IPv4/IGMP and IPv6/MLD protocols.

A tunnel is "created" when a gateway sends a Membership Update message containing an IGMP or MLD membership report that creates one

or more group subscriptions when none currently existed for that tunnel endpoint address.

A tunnel ceases to exist when all group subscriptions for a tunnel endpoint are deleted. This may occur as a result of the following events:

- o The gateway sends an IGMP or MLD report, leave or done message to the relay that deletes the last group subscription linked to the tunnel endpoint.
- o The gateway sends a Teardown message to the relay that causes it to delete any and all subscriptions bound to the tunnel endpoint.
- o The relay stops receiving updates from the gateway until such time that per-group or per-tunnel timers expire, causing the relay to delete the subscriptions.

The tunneling approach described above conceptually transforms a unicast-only inter-network into an NBMA link layer, over which multicast traffic may be delivered. Each relay, plus the set of all gateways using the relay, together may be thought of as being on a separate logical NBMA link, where the "link layer" address is a UDP/IP address-port pair provided by the Membership Update message.

#### 4.2.2.1. Address Roaming

As described above, each time a relay receives a Membership Update message from a new source address-port pair, the group subscriptions described by that message apply to the tunnel endpoint identified by that address.

This can cause problems for a gateway if the address carried by the messages it sends to a relay changes unexpectedly. These changes may cause the relay to transmit duplicate, undeliverable or unrequested traffic back towards the gateway or an intermediate device. This may create congestion and have negative consequences for the gateway, its network, or multicast receivers, and in some cases, may also produce a significant amount of ICMP traffic directed back towards the relay by a NAT, router or gateway host.

There are several scenarios in which the address carried by messages sent by a gateway may change without that gateway's knowledge, as for example, when:

- o The message originates from a different interface on a gateway that possesses multiple interfaces.

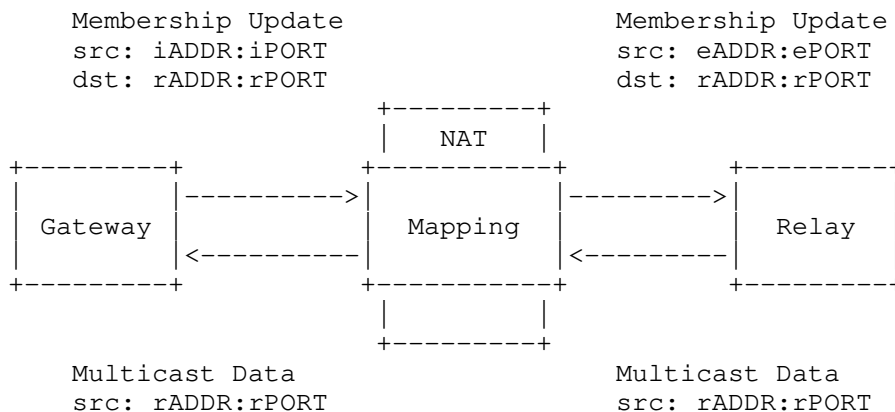
- o The DHCP assignment for a gateway interface changes.
- o The gateway roams to a different wireless network.
- o The address mapping applied by an intervening network-translation-device (NAT) changes as a result of mapping expiration or routing changes in a multi-homed network.

In the case where the address change occurs between the transmission of a Request message and subsequent Membership Update messages, the relay will simply ignore any Membership Update messages from the new address because MAC authentication will fail (see Section 4.2.1.2). The relay may continue to transmit previously requested traffic, but no duplication will occur, i.e., the possibility for the delivery of duplicate traffic does not arise until a Request message is received from the new address.

The protocol provides a method for a gateway to detect an address change and explicitly request that the relay stop sending traffic to a previous address. This process involves the Membership Query and Teardown messages and is described in Section 4.2.1.3.

#### 4.2.2.2. Network Address Translation

The messages sent by a gateway to a relay may be subject to network address translation (NAT) - the source IP address and UDP port carried by an IP packet sent by the gateway may be modified multiple times before arriving at the relay. In the most restrictive form of NAT, the NAT device will create a new mapping for each combination of source and destination IP address and UDP port. In this case, bi-directional communication can only be conducted by sending outgoing packets to the source address and port carried by the last incoming packet.



dst: iADDR:iPORT

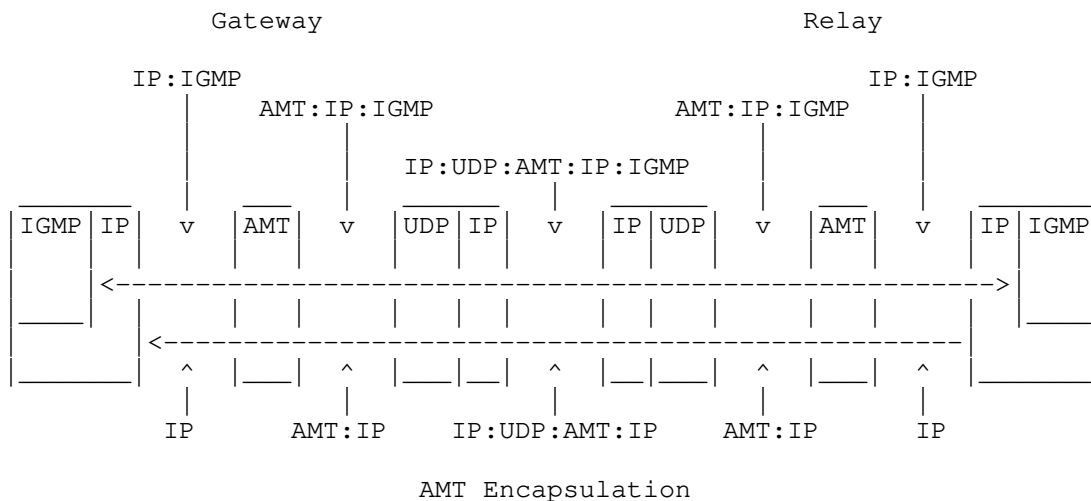
dst: eADDR:ePORT

## Network Address Translation in AMT

AMT provides automatic NAT traversal by using the source IP address and UDP port carried by the Membership Update message as received at the relay as the destination address for any Multicast Data messages the relay sends back as a result.

The NAT mapping created by a Membership Update message will eventually expire unless it is refreshed by a passing message. This refresh will occur each time the gateway performs the periodic update required to refresh group state within the relay (See Section 4.2.1.2).

## 4.2.2.3. UDP Encapsulation



## AMT Encapsulation

The IGMP and MLD messages used in AMT are exchanged as complete IP datagrams. These IP datagrams are encapsulated in AMT messages that are transmitted using UDP. The same holds true for multicast traffic - each multicast IP datagram or datagram fragment that arrives at the relay is encapsulated in an AMT message and transmitted to one or more gateways via UDP.

The IP protocol of the encapsulated packets need not match the IP protocol used to send the AMT messages. AMT messages sent via IPv4 may carry IPv6/MLD packets and AMT messages sent via IPv6 may carry IPv4/IGMP packets.

The checksum field contained in the UDP header of the messages

requires special consideration. Of primary concern is the cost of computing a checksum on each replicated multicast packet after it is encapsulated for delivery to a gateway. Many routing/forwarding platforms do not possess the capability to compute checksums on UDP encapsulated packets as they may not have access to the entire datagram.

To avoid placing an undue burden on the relay platform, the protocol specifically allows zero-valued UDP checksums on the multicast data messages. This is not an issue in UDP over IPv4 as the UDP checksum field may be set to zero. However, this is a problem for UDP over IPv6 as that protocol requires a valid, non-zero checksum in UDP datagrams [RFC2460]. Messages sent over IPv6 with a UDP checksum of zero may fail to reach the gateway. This is a well known issue for UDP-based tunneling protocols that is described [I-D.ietf-6man-udpzero]. A recommended solution is described in [I-D.ietf-6man-udpchecksums].

## 5. Protocol Description

This section provides a normative description of the AMT protocol.

### 5.1. Protocol Messages

The AMT protocol defines seven message types for control and encapsulation. These messages are assigned the following names and numeric identifiers:

Message Type	Message Name
1	Relay Discovery
2	Relay Advertisement
3	Request
4	Membership Query
5	Membership Update
6	Multicast Data
7	Teardown

These messages are exchanged as IPv4 or IPv6 UDP datagrams.

#### 5.1.1. Relay Discovery

A Relay Discovery message is used to solicit a response from a relay in the form of a Relay Advertisement message.

The UDP/IP datagram containing this message MUST carry a valid, non-zero UDP checksum and carry the following IP address and UDP port values:

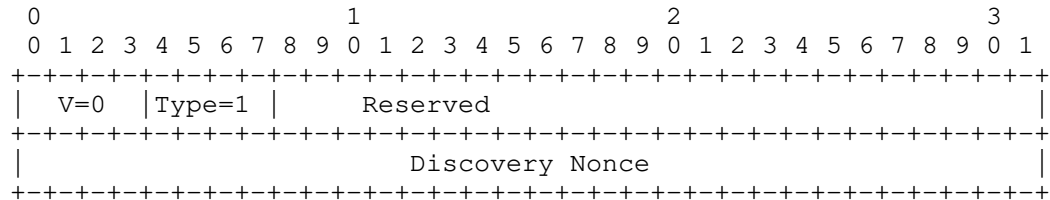
Source IP Address - The IP address of the gateway interface on which the gateway will listen for a relay response. Note: The value of this field may be changed as a result of network address translation before arriving at the relay.

Source UDP Port - The UDP port number on which the gateway will listen for a relay response. Note: The value of this field may be changed as a result of network address translation before arriving at the relay.



Destination IP Address - An anycast or unicast IP address, i.e., the Relay Discovery Address advertised by a relay.

Destination UDP Port - The IANA-assigned AMT port number.



Relay Discovery Message Format

#### 5.1.1.1. Version (V)

The protocol version number for this message is 0.

#### 5.1.1.2. Type

The type number for this message is 1.

#### 5.1.1.3. Reserved

Reserved bits that MUST be set to zero by the gateway and ignored by the relay.

#### 5.1.1.4. Discovery Nonce

A 32-bit random value generated by the gateway and echoed by the relay in a Relay Advertisement message. This value is used by the gateway to correlate Relay Advertisement messages with Relay Discovery messages. Discovery nonce generation is described in Section 5.2.3.4.5.

#### 5.1.2. Relay Advertisement

The Relay Advertisement message is used to supply a gateway with a unicast IP address of a relay. A relay sends this message to a gateway when it receives a Relay Discovery message from that gateway.

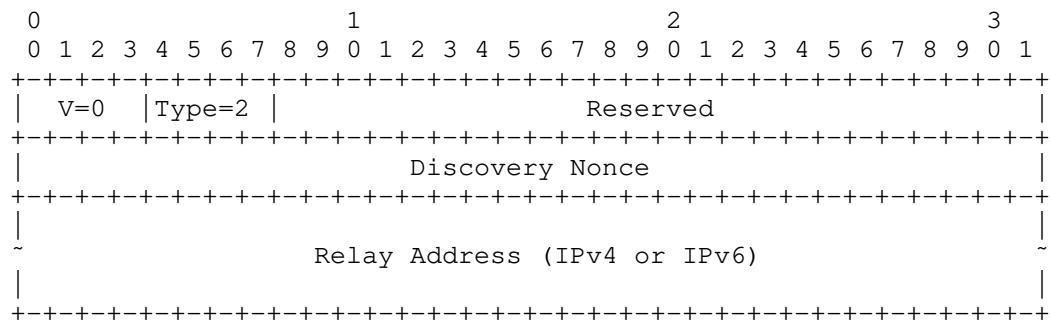
The UDP/IP datagram containing this message MUST carry a valid, non-zero UDP checksum and carry the following IP address and UDP port values:

Source IP Address - The destination IP address carried by the Relay Discovery message (i.e., the Relay Discovery Address advertised by the relay).

Source UDP Port - The destination UDP port carried by the Relay Discovery message (i.e., the IANA-assigned AMT port number).

Destination IP Address - The source IP address carried by the Relay Discovery message. Note: The value of this field may be changed as a result of network address translation before arriving at the gateway.

Destination UDP Port - The source UDP port carried by the Relay Discovery message. Note: The value of this field may be changed as a result of network address translation before arriving at the gateway.



Relay Advertisement Message Format

#### 5.1.2.1. Version (V)

The protocol version number for this message is 0.

#### 5.1.2.2. Type

The type number for this message is 2.

#### 5.1.2.3. Reserved

Reserved bits that MUST be set to zero by the relay and ignored by the gateway.

#### 5.1.2.4. Discovery Nonce

A 32-bit value copied from the Discovery Nonce field (Section 5.1.1.4) contained in the Relay Discovery message. The gateway uses this value to match a Relay Advertisement to a Relay Discovery message.

#### 5.1.2.5. Relay Address

The unicast IPv4 or IPv6 address of the relay. A gateway uses the length of the UDP datagram containing the Relay Advertisement message to determine the address family; i.e., length - 8 = 4 (IPv4) or 16 (IPv6). The relay returns an IP address for the protocol used to send the Relay Discovery message, i.e., an IPv4 relay address for an IPv4 discovery address or an IPv6 relay address for an IPv6 discovery address.

#### 5.1.3. Request

A gateway sends a Request message to a relay to solicit a Membership Query response.

The successful delivery of this message marks the start of the first stage in the three-way handshake used to create or update state within a relay.

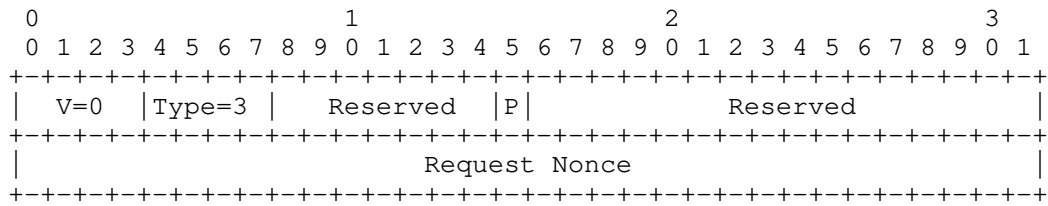
The UDP/IP datagram containing this message MUST carry a valid, non-zero UDP checksum and carry the following IP address and UDP port values:

Source IP Address - The IP address of the gateway interface on which the gateway will listen for a response from the relay. Note: The value of this field may be changed as a result of network address translation before arriving at the relay.

Source UDP Port - The UDP port number on which the gateway will listen for a response from the relay. Note: The value of this field may be changed as a result of network address translation before arriving at the relay.

Destination IP Address - The unicast IP address of the relay.

Destination UDP Port - The IANA-assigned AMT port number.



Request Message Format

## 5.1.3.1. Version (V)

The protocol version number for this message is 0.

## 5.1.3.2. Type

The type number for this message is 3.

## 5.1.3.3. Reserved

Reserved bits that MUST be set to zero by the gateway and ignored by the relay.

## 5.1.3.4. P Flag

The "P" flag is set to indicate which group membership protocol the gateway wishes the relay to use in the Membership Query response:

## Value Meaning

- 0 The relay MUST respond with a Membership Query message that contains an IPv4 packet carrying an IGMPv3 general query message.
- 1 The relay MUST respond with a Membership Query message that contains an IPv6 packet carrying an MLDv2 general query message.

## 5.1.3.5. Request Nonce

A 32-bit random value generated by the gateway and echoed by the relay in a Membership Query message. This value is used by the relay to compute the Response MAC value and is used by the gateway to correlate Membership Query messages with Request messages. Request nonce generation is described in Section 5.2.3.5.6.

#### 5.1.4. Membership Query

A relay sends a Membership Query message to a gateway to solicit a Membership Update response, but only after receiving a Request message from the gateway.

The successful delivery of this message to a gateway marks the start of the second-stage in the three-way handshake used to create or update tunnel state within a relay.

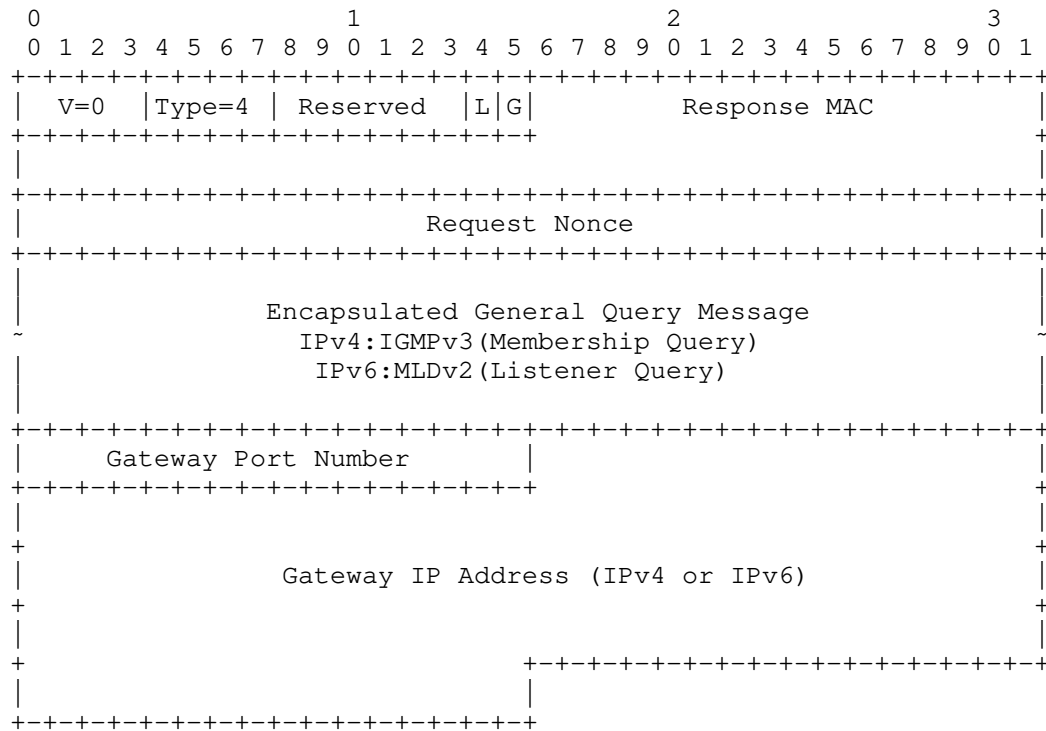
The UDP/IP datagram containing this message MUST carry a valid, non-zero UDP checksum and carry the following IP address and UDP port values:

Source IP Address - The destination IP address carried by the Request message (i.e., the unicast IP address of the relay).

Source UDP Port - The destination UDP port carried by the Request message (i.e., the IANA-assigned AMT port number).

Destination IP Address - The source IP address carried by the Request message. Note: The value of this field may be changed as a result of network address translation before arriving at the gateway.

Destination UDP Port - The source UDP port carried by the Request message. Note: The value of this field may be changed as a result of network address translation before arriving at the gateway.



Membership Query Message Format

## 5.1.4.1. Version (V)

The protocol version number for this message is 0.

## 5.1.4.2. Type

The type number for this message is 4.

## 5.1.4.3. Reserved

Reserved bits that MUST be set to zero by the relay and ignored by the gateway.

## 5.1.4.4. Limit (L) Flag

A 1-bit flag set to 1 to indicate that the relay is NOT accepting Membership Update messages from new gateway tunnel endpoints and that it will ignore any that are. A value of 0 has no special significance - the relay may or may not be accepting Membership Update messages from new gateway tunnel endpoints. A gateway checks

this flag before attempting to create new group subscription state on the relay to determine whether it should restart relay discovery. A gateway that has already created group subscriptions on the relay may ignore this flag. Support for this flag is RECOMMENDED.

#### 5.1.4.5. Gateway Address (G) Flag

A 1-bit flag set to 0 to indicate that the message does NOT carry the Gateway Port and Gateway IP Address fields, and 1 to indicate that it does. A relay implementation that supports the optional teardown procedure (See Section 5.3.3.5) SHOULD set this flag and the Gateway Address field values. If a relay sets this flag, it MUST also include the Gateway Address fields in the message. A gateway implementation that does not support the optional teardown procedure (See Section 5.2.3.7) MAY ignore this flag and the Gateway Address fields if they are present.

#### 5.1.4.6. Response MAC

A 48-bit source authentication hash generated by the relay as described in Section 5.3.5. The gateway echoes this value in subsequent Membership Update messages to allow the relay to verify that the sender of a Membership Update message was the intended receiver of a Membership Query sent by the relay.

#### 5.1.4.7. Request Nonce

A 32-bit value copied from the Request Nonce field (Section 5.1.3.5) carried by a Request message. The relay will have included this value in the Response MAC hash computation. The gateway echoes this value in subsequent Membership Update messages. The gateway also uses this value to match a Membership Query to a Request message.

#### 5.1.4.8. Encapsulated General Query Message

An IP-encapsulated IGMP or MLD message generated by the relay. This field will contain one of the following IP datagrams:

IPv4:IGMPv3 Membership Query

IPv6:MLDv2 Listener Query

The source address carried by the query message should be set as described in Section 5.3.3.3.

The Querier's Query Interval Code (QQIC) field in the general query is used by a relay to specify the time offset a gateway should use to schedule a new three-way handshake to refresh the group membership

state within the relay (current time + Query Interval).

The Querier's Robustness Variable (QRV) field in the general query is used by a relay to specify the number of times a gateway should retransmit unsolicited membership reports, encapsulated within Membership Update messages, and optionally, the number of times to send a Teardown message.

#### 5.1.4.9. Gateway Address Fields

The Gateway Port Number and Gateway Address fields are present in the Membership Query message if, and only if, the "G" flag is set.

A gateway need not parse the encapsulated IP datagram to determine the position of these fields within the UDP datagram containing the Membership Query message - if the G-flag is set, the gateway may simply subtract the total length of the fields (18 bytes) from the total length of the UDP datagram to obtain the offset.

##### 5.1.4.9.1. Gateway Port Number

A 16-bit UDP port containing a UDP port value.

The Relay sets this field to the value of the UDP source port of the Request message that triggered the Query message.

##### 5.1.4.9.2. Gateway IP Address

A 16-byte IP address that, when combined with the value contained in the Gateway Port Number field, forms the gateway endpoint address that the relay will use to identify the tunnel instance, if any, created by a subsequent Membership Update message. This field may contain an IPv6 address or an IPv4 address stored as an IPv4-compatible IPv6 address, where the IPv4 address is prefixed with 96 bits set to zero (See [RFC4291]). This address must match that used by the relay to compute the value stored in the Response MAC field.

#### 5.1.5. Membership Update

A gateway sends a Membership Update message to a relay to report a change in group membership state, or to report the current group membership state in response to receiving a Membership Query message. The gateway encapsulates the IGMP or MLD message as an IP datagram within a Membership Update message and sends it to the relay, where it may (see below) be decapsulated and processed by the relay to update group membership and forwarding state.

A gateway cannot send a Membership Update message until it receives a



Membership Query from a relay because the gateway must copy the Request Nonce and Response MAC values carried by a Membership Query into any subsequent Membership Update messages it sends back to that relay. These values are used by the relay to verify that the sender of the Membership Update message was the recipient of the Membership Query message from which these values were copied.

The successful delivery of this message to the relay marks the start of the final stage in the three-way handshake. This stage concludes when the relay successfully verifies that sender of the Membership Update message was the recipient of a Membership Query message sent earlier. At this point, the relay may proceed to process the encapsulated IGMP or MLD message to create or update group membership and forwarding state on behalf of the gateway.

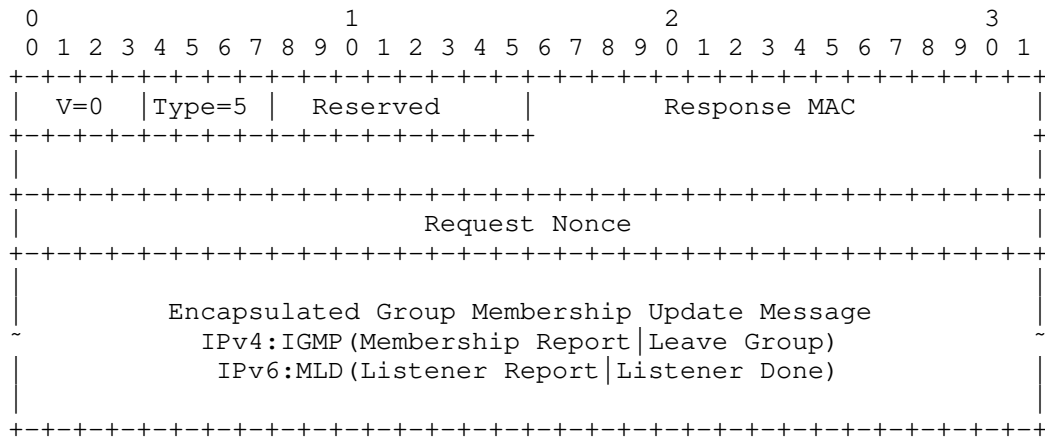
The UDP/IP datagram containing this message MUST carry a valid, non-zero UDP checksum and carry the following IP address and UDP port values:

Source IP Address - The IP address of the gateway interface on which the gateway will listen for Multicast Data messages from the relay. The address must be the same address used to send the initial Request message or the message will be ignored. Note: The value of this field may be changed as a result of network address translation before arriving at the relay.

Source UDP Port - The UDP port number on which the gateway will listen for Multicast Data messages from the relay. This port must be the same port used to send the initial Request message or the message will be ignored. Note: The value of this field may be changed as a result of network address translation before arriving at the relay.

Destination IP Address - The unicast IP address of the relay.

Destination UDP Port - The IANA-assigned AMT UDP port number.



Membership Update Message Format

## 5.1.5.1. Version (V)

The protocol version number for this message is 0.

## 5.1.5.2. Type

The type number for this message is 5.

## 5.1.5.3. Reserved

Reserved bits that MUST be set to zero by the gateway and ignored by the relay.

## 5.1.5.4. Response MAC

A 48-bit value copied from the Response MAC field (Section 5.1.4.6) in a Membership Query message. Used by the relay to perform source authentication.

## 5.1.5.5. Request Nonce

A 32-bit value copied from the Request Nonce field in a Request or Membership Query message. Used by the relay to perform source authentication.

## 5.1.5.6. Encapsulated Group Membership Update Message

An IP-encapsulated IGMP or MLD message produced by the host-mode IGMP or MLD protocol running on a gateway pseudo-interface. This field will contain one of the following IP datagrams:

IPv4:IGMPv2 Membership Report

IPv4:IGMPv2 Leave Group

IPv4:IGMPv3 Membership Report

IPv6:MLDv1 Multicast Listener Report

IPv6:MLDv1 Multicast Listener Done

IPv6:MLDv2 Multicast Listener Report

The source address carried by the message should be set as described in Section 5.2.1.

#### 5.1.6. Multicast Data

A relay sends a Multicast Data message to deliver an multicast IP datagram or datagram fragment to a gateway.

The checksum field in the UDP header of this message MAY contain a value of zero when sent over IPv4 but SHOULD, if possible, contain a valid, non-zero value when sent over IPv6 (See Section 4.2.2.3).

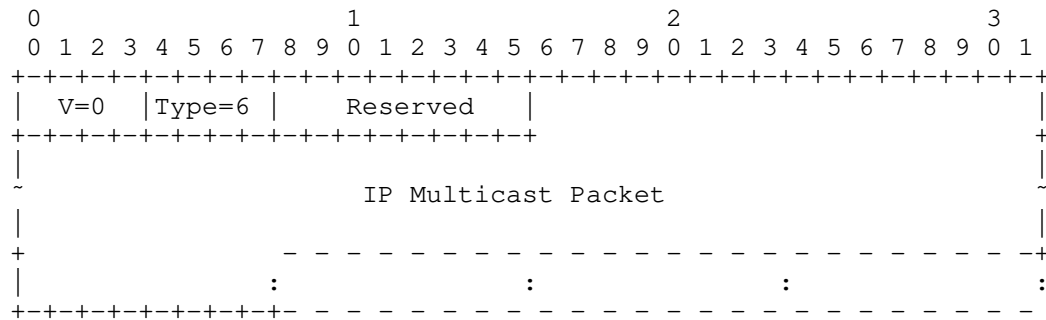
The UDP/IP datagram containing this message MUST carry the following IP address and UDP port values:

Source IP Address - The unicast IP address of the relay.

Source UDP Port - The IANA-assigned AMT port number.

Destination IP Address - A tunnel endpoint IP address, i.e., the source IP address carried by the Membership Update message sent by a gateway to indicate an interest in receiving the multicast packet. Note: The value of this field may be changed as a result of network address translation before arriving at the gateway.

Destination UDP Port - A tunnel endpoint UDP port, i.e., the source UDP port carried by the Membership Update message sent by a gateway to indicate an interest in receiving the multicast packet. Note: The value of this field may be changed as a result of network address translation before arriving at the gateway.



Multicast Data Message Format

## 5.1.6.1. Version (V)

The protocol version number for this message is 0.

## 5.1.6.2. Type

The type number for this message is 6.

## 5.1.6.3. Reserved

Bits that MUST be set to zero by the relay and ignored by the gateway.

## 5.1.6.4. IP Multicast Data

A complete IPv4 or IPv6 multicast datagram or datagram fragment.

## 5.1.7. Teardown

A gateway sends a Teardown message to a relay to request that it stop sending Multicast Data messages to a tunnel endpoint created by an earlier Membership Update message. A gateway sends this message when it detects that a Request message sent to the relay carries an address that differs from that carried by a previous Request message. The gateway uses the Gateway IP Address and Gateway Port Number Fields in the Membership Query message to detect these address changes.

To provide backwards compatibility with early implementations of the AMT protocol, support for this message and associated procedures is considered OPTIONAL – gateways are not required to send this message and relays are not required to act upon it.

The UDP/IP datagram containing this message MUST carry a valid, non-

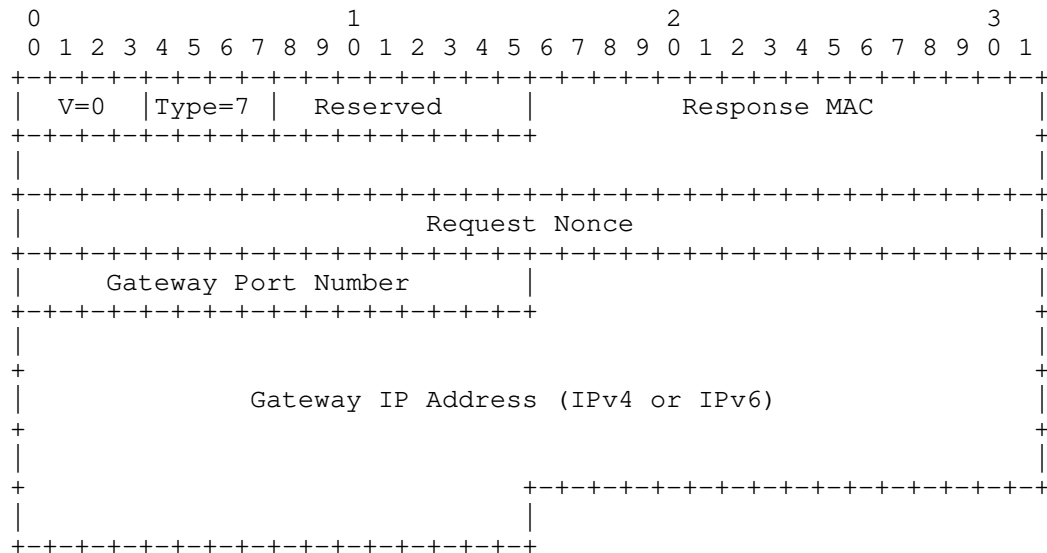
zero UDP checksum and carry the following IP address and UDP port values:

Source IP Address - The IP address of the gateway interface used to send the message. This address may differ from that used to send earlier messages. Note: The value of this field may be changed as a result of network address translation before arriving at the relay.

Source UDP Port - The UDP port number. This port number may differ from that used to send earlier messages. Note: The value of this field may be changed as a result of network address translation before arriving at the relay.

Destination IP Address - The unicast IP address of the relay.

Destination UDP Port - The IANA-assigned AMT port number.



Membership Teardown Message Format

#### 5.1.7.1. Version (V)

The protocol version number for this message is 0.

#### 5.1.7.2. Type

The type number for this message is 7.

#### 5.1.7.3. Reserved

Reserved bits that MUST be set to zero by the gateway and ignored by the relay.

#### 5.1.7.4. Response MAC

A 48-bit value copied from the Response MAC field (Section 5.1.4.6) in the last Membership Query message the relay sent to the gateway endpoint address of the tunnel to be torn down. The gateway endpoint address is provided by the Gateway IP Address and Gateway Port Number fields carried by the Membership Query message. The relay validates the Teardown message by comparing this value with one computed from the Request Nonce, Gateway Port Number and Gateway IP Address fields (just as it does in the Membership Update message).

#### 5.1.7.5. Request Nonce

A 32-bit value copied from the Request Nonce field (Section 5.1.4.7) in the last Membership Query message the relay sent to the gateway endpoint address of the tunnel to be torn down. The gateway endpoint address is provided by the Gateway IP Address and Gateway Port Number fields carried by the Membership Query message. This value must match that used by the relay to compute the value stored in the Response MAC field.

#### 5.1.7.6. Gateway Port Number

A 16-bit UDP port number that, when combined with the value contained in the Gateway IP Address field, forms the tunnel endpoint address that the relay will use to identify the tunnel instance to tear down. The relay provides this value to the gateway using the Gateway Port Number field (Section 5.1.4.9.1) in a Membership Query message. This port number must match that used by the relay to compute the value stored in the Response MAC field.

#### 5.1.7.7. Gateway IP Address

A 16-byte IP address that, when combined with the value contained in the Gateway Port Number field, forms the tunnel endpoint address that the relay will use to identify the tunnel instance to tear down. The relay provides this value to the gateway using the Gateway IP Address field (Section 5.1.4.9.2) in a Membership Query message. This field may contain an IPv6 address or an IPv4 address stored as an IPv4-compatible IPv6 address, where the IPv4 address is prefixed with 96 bits set to zero (See [RFC4291]). This address must match that used by the relay to compute the value stored in the Response MAC field.

## 5.2. Gateway Operation

The following sections describe gateway implementation requirements. A non-normative discussion of gateway operation may be found in Section 4.2.

### 5.2.1. IP/IGMP/MLD Protocol Requirements

Gateway operation requires a subset of host mode IPv4/IGMP and IPv6/MLD functionality to provide group membership tracking, general query processing, and report generation. A gateway MAY use IGMPv2 (ASM), IGMPv3 (ASM and SSM), MLDv1 (ASM) or MLDv2 (ASM and SSM).

An application with embedded gateway functionality must provide its own implementation of this subset of the IPv4/IGMP and IPv6/MLD protocols. The service interface used to manipulate group membership state need not match that described in the IGMP and MLD specifications, but the actions taken as a result SHOULD be similar to those described in Section 5.1 of [RFC3376] and Section 6.1 of [RFC3810]. The gateway application will likely need to implement many of the same functions as a host IP stack, including checksum verification, dispatching, datagram filtering and forwarding, and IP encapsulation/decapsulation.

The IP-encapsulated IGMP messages generated by the gateway IPv4/IGMP implementation MUST conform to the description found in Section 4 of [RFC3376]. These datagrams MUST possess the IP headers, header options and header values called for in [RFC3376], with the following exception; the source IP address for an IGMP report datagram MAY be set to the "unspecified" address (all octets are zero ) but SHOULD be set to an address in the address range specifically assigned by IANA for use in the IGMP messages sent from a gateway to a relay (i.e. 154.7.1.2 through 154.7.1.254 as described in Section 7). This exception is made because the gateway pseudo-interface might not possess an assigned address, and even if such an address exists, that address would not be a valid link-local source address on any relay interface. The rationale for using the aforementioned source addresses is primarily one of convenience - a relay will accept an IGMP report carried by a Membership Update message regardless of the source address it carries. See Section 5.3.1.

The IP-encapsulated MLD messages generated by the gateway IPv6/MLD implementation MUST conform to the description found in Section 5 of [RFC3810]. These datagrams MUST possess the IP headers, header options and header values called for in [RFC3810], with the following exception; the source IP address for an MLD report datagram MAY be set to the "unspecified" address (all octets are zero ) but SHOULD be set to an IPv6 link-local address in the range FE80::/64 excluding

FE80::1 and FE80::2. This exception is made because the gateway pseudo-interface might not possess a valid IPv6 address. As with IGMP, a relay will accept an MLD report carried by a Membership Update message regardless of the source address it carries. See Section 5.3.1.

The gateway IGMP/MLD implementation SHOULD retransmit unsolicited membership state-change reports and merge new state change reports with pending reports as described in Section 5.1 of [RFC3376] and Section 6.1 of [RFC3810]. The number of retransmissions is specified by the relay in the Querier's Robustness Variable (QRV) field in the last general query forwarded by the pseudo-interface.

The gateway IGMP/MLD implementation SHOULD handle general query messages as described in Section 5.2 of [RFC3376] and Section 6.2 of [RFC3810], but MAY ignore the Max Resp Code field value and generate a current state report without any delay.

An IPv4 gateway implementation MUST accept IPv4 datagrams that carry the general query variant of the IGMPv3 Membership Query message, as described in Section 4 of [RFC3376]. The gateway MUST accept the IGMP datagram regardless of the IP source address carried by that datagram.

An IPv6 gateway implementation MUST accept IPv6 datagrams that carry the general query variant of the MLDv2 Multicast Listener Query message, as described in Section 5 of [RFC3810]. The gateway MUST accept the MLD datagram regardless of the IP source address carried by that datagram.

#### 5.2.2. Pseudo-Interface Configuration

A gateway host may possess or create multiple gateway pseudo-interfaces, each with a unique configuration that describes a binding to a specific IP protocol, relay address, relay discovery address or upstream network interface.

##### 5.2.2.1. Relay Discovery Address

If a gateway implementation uses AMT relay discovery to obtain a relay address, it must first be supplied with a relay discovery address. The relay discovery address may be an anycast or unicast address. A gateway implementation may rely on a static address assignment or some form of dynamic address discovery. This specification does not require that a gateway implementation use any particular method to obtain a relay discovery address - an implementation may employ any method that returns a suitable relay discovery address.



#### 5.2.2.2. Relay Address

Before a gateway implementation can execute the AMT protocol to request and receive multicast traffic, it must be supplied with a unicast relay address. A gateway implementation may rely on static address assignment or support some form of dynamic address discovery. This specification does not require the use of any particular method to obtain a relay address - an implementation may employ any method that returns a suitable relay address.

#### 5.2.2.3. Upstream Interface Selection

A gateway host that possesses multiple network interfaces or addresses may allow for an explicit selection of the interface to use when communicating with a relay. The selection might be made to satisfy connectivity, tunneling or IP protocol requirements.

#### 5.2.2.4. Optional Retransmission Parameters

A gateway implementation that supports retransmission MAY require the following information:

Discovery Timeout

Initial time to wait for a response to a Relay Discovery message.

Maximum Relay Discovery Retransmission Count

Maximum number of Relay Discovery retransmissions to allow before terminating relay discovery and reporting an error.

Request Timeout

Initial time to wait for a response to a Request message.

Maximum Request Retransmission Count

Maximum number of Request retransmissions to allow before abandoning a relay and restarting relay discovery or reporting an error.

Maximum Retries Count For "Destination Unreachable"

The maximum number of times a gateway should attempt to send the same Request or Membership Update message after receiving an ICMP "Destination Unreachable".

#### 5.2.3. Gateway Service

In the following descriptions, a gateway pseudo interface is treated as a passive entity managed by a gateway service. The gateway pseudo-interface provides the state and the gateway service provides the processing. The term "gateway" is used when describing service

behavior with respect to a single pseudo-interface.

#### 5.2.3.1. Startup

When a gateway pseudo-interface is started, the gateway service begins listening for AMT messages sent to the UDP endpoint(s) associated with the pseudo-interface and for any locally-generated IGMP/MLD messages passed to the pseudo-interface. The handling of these messages is described below.

When the pseudo-interface is enabled, the gateway service MAY:

- o Optionally execute the relay discovery procedure described in Section 5.2.3.4.
- o Optionally execute the membership query procedure described in Section 5.2.3.5 to start the periodic membership update cycle.

#### 5.2.3.2. Handling AMT Messages

A gateway MUST ignore any datagram it receives that cannot be interpreted as a Relay Advertisement, Membership Query, or Multicast Data message. The handling of Relay Advertisement, Membership Query, and Multicast Data messages is addressed in the sections that follow.

While listening for AMT messages, a gateway may be notified that an ICMP Destination Unreachable message was received as a result of an AMT message transmission. Handling of ICMP Destination Unreachable messages is described in Section 5.2.3.9.

#### 5.2.3.3. Handling Multicast Data Messages

A gateway may receive Multicast Data messages after it sends a Membership Update message to a relay that adds a group subscription. The gateway may continue to receive Multicast Data messages long after the gateway sends a Membership Update message that deletes existing group subscriptions. The gateway MUST be prepared to receive these messages at any time, but MAY ignore them or discard their contents if the gateway no longer has any interest in receiving the multicast datagrams contained within them.

A gateway MUST ignore a Multicast Data message if it fails to satisfy any of the following requirements:

- o The source IP address and UDP port carried by the Multicast Data message MUST be equal to the destination IP address and UDP port carried by the matching Membership Update message (i.e., the current relay address).

- o The destination address carried by the encapsulated IP datagram MUST fall within the multicast address allocation assigned to the relevant IP protocol, i.e., 224.0.0.0/4 for IPv4 and FF00::/8 for IPv6.

The gateway extracts the encapsulated IP datagram and forwards it to the local IP protocol implementation for checksum verification, fragmented datagram reassembly, source and group filtering, and transport-layer protocol processing.

Because AMT uses UDP encapsulation to deliver multicast datagrams to gateways, it qualifies as a tunneling protocol subject to the limitations described in [I-D.ietf-6man-udpzero]. If supported, a gateway SHOULD employ the solution described in [I-D.ietf-6man-udpchecksums] to ensure that the local IP stack does not discard IPv6 datagrams with zero checksums. If Multicast Data message datagrams are processed directly within the gateway (instead of the host IP stack), the gateway MUST NOT discard any of these datagrams because they carry a UDP checksum of zero.

#### 5.2.3.4. Relay Discovery Procedure

This section describes gateway requirements related to the relay discovery message sequence described in Section 4.2.1.1.

##### 5.2.3.4.1. Starting Relay Discovery

A gateway may start or restart the relay discovery procedure in response to the following events:

- o When a gateway pseudo-interface is started (enabled).
- o When the gateway wishes to report a group subscription when none currently exist.
- o Before sending the next Request message in a membership update cycle, i.e., each time the query timer expires (see below).
- o After the gateway fails to receive a response to a Request message.
- o After the gateway receives a Membership Query message with the L-flag set to 1.

#### 5.2.3.4.2. Sending a Relay Discovery Message

A gateway sends a Relay Discovery message to a relay to start the relay discovery process.

The gateway MUST send the Relay Discovery message using the current Relay Discovery Address and IANA-assigned UDP port number as the destination. The Discovery Nonce value in the Relay Discovery message MUST be computed as described in Section 5.2.3.4.5.

The gateway MUST save a copy of Relay Discovery message or save the Discovery Nonce value for possible retransmission and verification of a Relay Advertisement response.

When a gateway sends a Relay Discovery message, it may be notified that an ICMP Destination Unreachable message was received as a result of an earlier AMT message transmission. Handling of ICMP Destination Unreachable messages is described in Section 5.2.3.9.

#### 5.2.3.4.3. Waiting for a Relay Advertisement Message

A gateway MAY retransmit a Relay Discovery message if it does not receive a matching Relay Advertisement message within some timeout period. If the gateway retransmits the message multiple times, the timeout period SHOULD be adjusted to provide an random exponential back-off. The RECOMMENDED timeout is a random value in the range  $[\text{initial\_timeout}, \text{MIN}(\text{initial\_timeout} * 2^{\text{retry\_count}}, \text{maximum\_timeout})]$ , with a RECOMMENDED initial\_timeout of 1 second and a RECOMMENDED maximum\_timeout of 120 seconds (which is the recommended minimum NAT mapping timeout described in [RFC4787]).

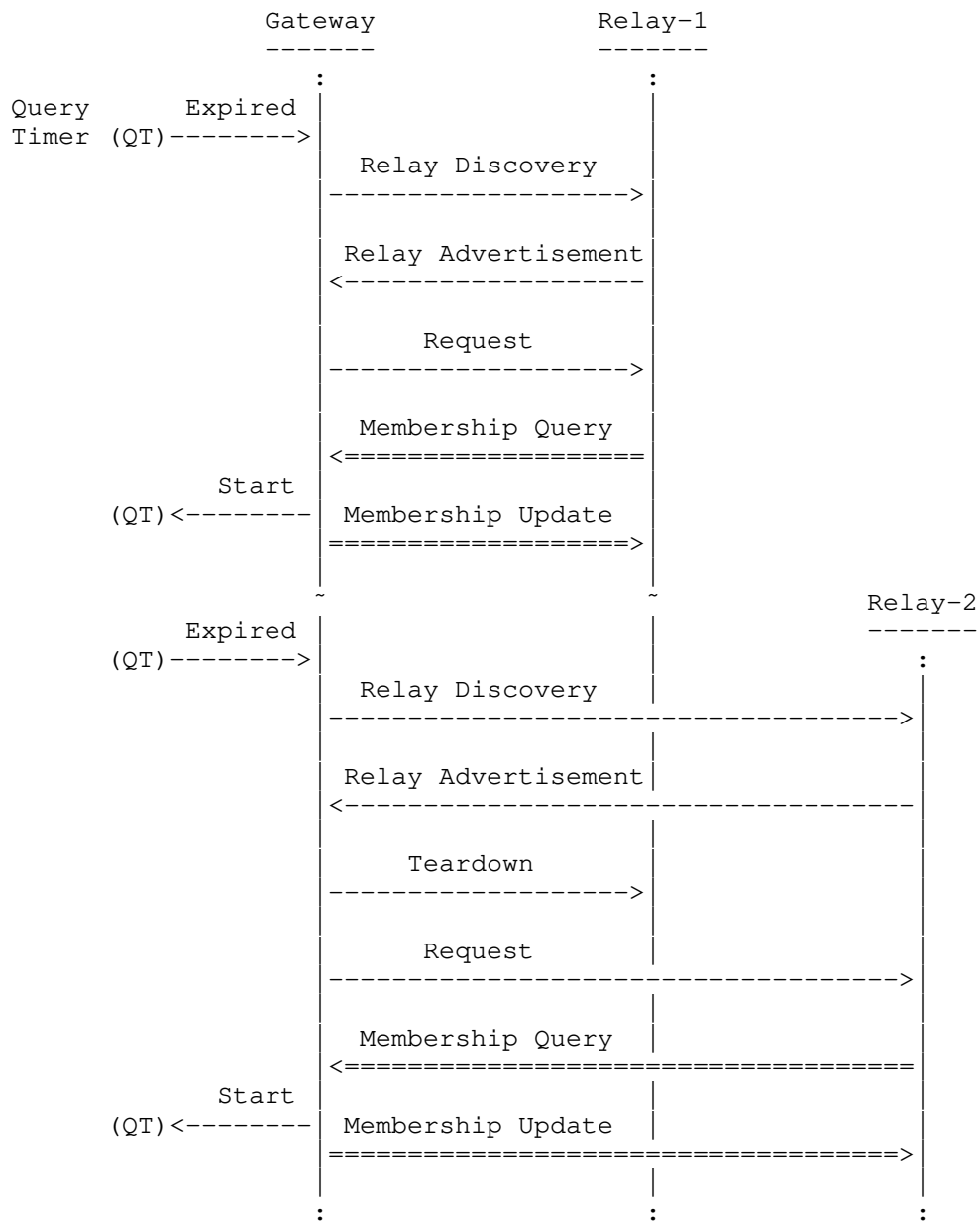
#### 5.2.3.4.4. Handling a Relay Advertisement Message

When a gateway receives a Relay Advertisement message it must first determine whether it should accept or ignore the message. A gateway MUST ignore a Relay Advertisement message if it fails to satisfy any of the following requirements:

- o The gateway MUST be waiting for a Relay Advertisement message.
- o The Discovery Nonce value contained in the Relay Advertisement message MUST equal to the Discovery Nonce value contained in the Relay Discovery message.
- o The source IP address and UDP port of the Relay Advertisement message MUST equal to the destination IP address and UDP port of the matching Relay Discovery message.

Once a gateway receives a Relay Advertisement response to a Relay Discovery message, it SHOULD ignore any other Relay Advertisements that arrive on the AMT interface until it sends a new Relay Discovery message.

If a gateway executes the relay discovery procedure at the start of each membership update cycle and the relay address returned in the latest Relay Advertisement message differs from the address returned in a previous Relay Advertisement message, then the gateway SHOULD send a Teardown message (if supported) to the old relay address, using information from the last Membership Query message received from that relay, as described in Section 5.2.3.7. This behavior is illustrated in the following diagram.



Teardown After Relay Address Change

#### 5.2.3.4.5. Discovery Nonce Generation

The discovery nonce MUST be a random, non-zero, 32-bit value, and if possible, SHOULD be computed using a cryptographically secure pseudo random number generator. A new nonce SHOULD be generated each time the gateway restarts the relay discovery process. The same nonce SHOULD be used when retransmitting a Relay Discovery message.

#### 5.2.3.5. Membership Query Procedure

This section describes gateway requirements related to the membership update message sequence described in Section 4.2.1.2.

##### 5.2.3.5.1. Starting the Membership Update Cycle

A gateway may send a Request message to start a membership update cycle (following the optional relay discovery procedure) in response to the following events:

- o When the gateway pseudo-interface is activated.
- o When the gateway wishes to report a group subscription when none currently exist.

Starting the membership update cycle when a gateway pseudo-interface is started provides several benefits:

- o Better performance by allowing state-change reports to be sent as they are generated, thus minimizing the time to join.
- o More robustness by relying on unsolicited state-change reports to update group membership state rather than the current-state reports generated by the membership update cycle. Unsolicited state-change reports are typically retransmitted multiple times while current-state reports are not.
- o Simplified implementation by eliminating any need to queue IGMP/MLD messages for delivery after a Membership Query is received, since the IGMP/MLD state-change messages may be sent as they are generated.

However, this approach places an additional load on relays as a gateway will send periodic requests even when it has no multicast subscriptions. To reduce load on a relay, a gateway SHOULD only send a Membership Update message while it has active group subscriptions. A relay will still need to compute a Response MAC for each Request, but will not be required to recompute it a second time to authenticate a Membership Update message that contains no

subscriptions.

#### 5.2.3.5.2. Sending a Request Message

A gateway sends a Request message to a relay to solicit a Membership Query response and start the membership update cycle.

A gateway constructs a Request message containing a Request Nonce value computed as described in Section 5.2.3.5.6. The gateway **MUST** set the "P" flag in the Request message to identify the protocol the gateway wishes the relay to use for the general query response.

A gateway **MUST** send a Request message using the current Relay Address and IANA-assigned AMT port number as the destination.

A gateway **MUST** save a copy of the Request message or save the Request Nonce and P-flag values for possible retransmission and verification of a Membership Query response.

When a gateway sends a Request message, it may be notified that an ICMP Destination Unreachable message was received as a result of an earlier AMT message transmission. Handling of ICMP Destination Unreachable messages is described in Section 5.2.3.9.

#### 5.2.3.5.3. Waiting for a Membership Query Message

A gateway **MAY** retransmit a Request message if it does not receive a matching Membership Query message within some timeout period. If the gateway retransmits the message multiple times, the timeout period **SHOULD** be adjusted to provide an random exponential back-off. The **RECOMMENDED** timeout is a random value in the range [initial\_timeout, MIN(initial\_timeout \* 2^retry\_count, maximum\_timeout)], with a **RECOMMENDED** initial\_timeout of 1 second and a **RECOMMENDED** maximum\_timeout of 120 seconds (which is the recommended minimum NAT mapping timeout described in [RFC4787]).

If a gateway that uses relay discovery does not receive a Membership Query within a specified time period or after a specified number of retries, the gateway **SHOULD** stop waiting for a Membership Query message and restart relay discovery to locate another relay.

#### 5.2.3.5.4. Handling a Membership Query Message

When a gateway receives a Membership Query message it must first determine whether it should accept or ignore the message. A gateway **MUST** ignore a Membership Query message, or the encapsulated IP datagram within it, if the message fails to satisfy any of the following requirements:



- o The gateway MUST be waiting for a Membership Query message.
- o The Request Nonce value contained in the Membership Query MUST equal the Request Nonce value contained in the Request message.
- o The source IP address and UDP port of the Membership Query MUST equal the destination IP address and UDP port of the matching Request message (i.e., the current relay address).
- o The encapsulated IP datagram MUST carry an IGMPv3 or MLDv2 message. The protocol MUST match the protocol identified by the "P" flag in the Request message.
- o The IGMPv3 or MLDv2 message MUST be a general query message.
- o The total length of the encapsulated IP datagram as computed from the lengths contained in the datagram header(s) MUST NOT exceed the available field length within the Membership Query message.

Once a gateway receives a Membership Query response to a Request message, it SHOULD ignore any other Membership Query messages that arrive on the AMT interface until it sends a new Request message.

The gateway MUST save the Membership Query message, or the Request Nonce, Response MAC, Gateway IP Address and Gateway Port Number fields for use in sending subsequent Membership Update and Teardown messages.

The gateway extracts the encapsulated IP datagram and forwards it to the local IP protocol implementation for checksum verification and dispatching to the IGMP or MLD implementation running on the pseudo-interface. The gateway MUST NOT forward any octets that might exist between the encapsulated IP datagram and the end of the message or Gateway Address fields.

The MLD protocol specification indicates that senders should use a link-local source IP address in message datagrams. This requirement must be relaxed for AMT because gateways and relays do not normally share a common subnet. For this reason, a gateway implementation MUST accept MLD (and IGMP) query message datagrams regardless of the source IP address they carry. This may require additional processing on the part of the gateway that might be avoided if the relay and gateway use the IPv4 and IPv6 addresses allocated for use in AMT encapsulated control packets as described in Section 5.2.1.

The gateway MUST start a timer that will trigger the next iteration of the membership update cycle by executing the membership query procedure. The gateway SHOULD compute the timer duration from the

Querier's Query Interval Code carried by the general-query. A gateway MAY use a smaller timer duration if required to refresh a NAT mapping that would otherwise timeout. A gateway MAY use a larger timer duration if it has no group subscriptions to report.

If the gateway supports the Teardown message and the G-flag is set in the Membership Query message, the gateway MUST compare the Gateway IP Address and Gateway Port Number on the new Membership Query message with the values carried by the previous Membership Query message. If either value has changed the gateway MUST send a Teardown message to the relay as described in Section 5.2.3.7.

If the L-flag is set in the Membership Query message, the relay is reporting that it is NOT accepting Membership Update messages that create new tunnel endpoints and will simply ignore any that do. If the L-flag is set and the gateway is not currently reporting any group subscriptions to the relay, the gateway SHOULD stop sending periodic Request messages and restart the relay discovery procedure (if discovery is enabled) to find a new relay with which to communicate. The gateway MAY continue to send updates even if the L-flag is set, if it has previously reported group subscriptions to the relay, one or more subscriptions still exist and the gateway endpoint address has not changed since the last Membership Query was received (see previous paragraph).

#### 5.2.3.5.5. Handling Query Timer Expiration

When the query timer (started in the previous step) expires, the gateway should execute the membership query procedure again to continue the membership update cycle.

#### 5.2.3.5.6. Request Nonce Generation

The request nonce MUST be a random value, and if possible, SHOULD be computed using a cryptographically secure pseudo random number generator. A new nonce MUST be generated each time the gateway starts the membership query process. The same nonce SHOULD be used when retransmitting a Request message.

#### 5.2.3.6. Membership Update Procedure

This section describes gateway requirements related to the membership update message sequence described in Section 4.2.1.2.

The membership update process is primarily driven by the host-mode IGMP or MLD protocol implementation running on the gateway pseudo-interface. The IGMP and MLD protocols produce current-state reports in response to general queries generated by the pseudo-interface via

AMT and produce state-change reports in response to receiver requests made using the IGMP or MLD service interface.

#### 5.2.3.6.1. Handling an IGMP/MLD IP Datagram

The gateway pseudo-interface **MUST** accept the following IP datagrams from the IPv4/IGMP and IPv6/MLD protocols running on the pseudo-interface:

- o IPv4 datagrams that carry an IGMPv2, or IGMPv3 Membership Report or an IGMPv2 Leave Group message as described in Section 4 of [RFC3376].
- o IPv6 datagrams that carry an MLDv1 or MLDv2 Multicast Listener Report or an MLDv1 Multicast Listener Done message as described in Section 5 of [RFC3810].

The gateway must be prepared to receive these messages any time the pseudo-interface is running. The gateway **MUST** ignore any datagrams not listed above.

A gateway that waits to start a membership update cycle until after it receives a datagram containing an IGMP/MLD state-change message **MAY**:

- o Discard IGMP or MLD datagrams until it receives a Membership Query message, at which time it processes the Membership Query message as normal to eventually produce a current-state report on the pseudo-interface which describes the end state (RECOMMENDED).
- o Insert IGMP or MLD datagrams into a queue for transmission after it receives a Membership Query message.

If and when a gateway receives a Membership Query message (for IGMP or MLD) it sends any queued or incoming IGMP or MLD datagrams to the relay as described in the next section.

#### 5.2.3.6.2. Sending a Membership Update Message

A gateway cannot send a Membership Update message to a relay until it has received a Membership Query message from a relay. If the gateway has not yet located a relay with which to communicate, it **MUST** first execute the relay discovery procedure described in Section 5.2.3.4 to obtain a relay address. If the gateway has a relay address, but has not yet received a Membership Query message, it **MUST** first execute the membership query procedure described in Section 5.2.3.5 to obtain a Request Nonce and Response MAC that can be used to send a Membership Update message.

Once a gateway possesses a valid Relay Address, Request Nonce and Response MAC, it may encapsulate the IP datagram containing the IGMP/MLD message into a Membership Update message. The gateway MUST copy the Request Nonce and Response MAC values from the last Membership Query received from the relay into the corresponding fields in the Membership Update. The gateway MUST send the Membership Update message using the Relay Address and IANA-assigned AMT port number as the destination.

When a gateway sends a Membership Update message, it may be notified that an ICMP Destination Unreachable message was received as a result of an earlier AMT message transmission. Handling of ICMP Destination Unreachable messages is described in Section 5.2.3.9.

#### 5.2.3.7. Teardown Procedure

This section describes gateway requirements related to the teardown message sequence described in Section 4.2.1.3.

Gateway support for the Teardown message is OPTIONAL but RECOMMENDED.

A gateway that supports Teardown SHOULD make use of Teardown functionality if it receives a Membership Query message from a relay that has the "G" flag set to indicate that it contains valid gateway address fields.

##### 5.2.3.7.1. Handling a Membership Query Message

As described in Section 5.2.3.5.4, if a gateway supports the Teardown message, has reported active group subscriptions, and receives a Membership Query message with the "G" flag set, the gateway MUST compare the Gateway IP Address and Gateway Port Number on the new Membership Query message with the values carried by the previous Membership Query message. If either value has changed the gateway MUST send a Teardown message as described in the next section.

##### 5.2.3.7.2. Sending a Teardown Message

A gateway sends a Teardown message to a relay to request that it stop delivering Multicast Data messages to the gateway and delete any group memberships created by the gateway.

When a gateway constructs a Teardown message, it MUST copy the Request Nonce, Response MAC, Gateway IP Address and Gateway Port Number fields from the Membership Query message that provided the Response MAC for the last Membership Update message sent, into the corresponding fields of the Teardown message.

A gateway MUST send the Teardown message using the Relay Address and IANA-assigned AMT port number as the destination. A gateway MAY send the Teardown message multiple times for robustness. The gateway SHOULD use the Querier's Robustness Variable (QRV) field contained in the query encapsulated within the last Membership Query to set the limit on the number of retransmissions. If the gateway sends the Teardown message multiple times, it SHOULD insert a delay between each transmission using the timing algorithm employed in IGMP/MLD for transmitting unsolicited state-change reports. The RECOMMENDED default delay value is 1 second.

When a gateway sends a Teardown message, it may be notified that an ICMP Destination Unreachable message was received as a result of an earlier AMT message transmission. Handling of ICMP Destination Unreachable messages is described in Section 5.2.3.9.

#### 5.2.3.8. Shutdown

When a gateway pseudo-interface is stopped and the gateway has existing group subscriptions, the gateway SHOULD either:

- o Send a Teardown message to the relay as described in Section 5.2.3.7, but only if the gateway supports the Teardown message, and the current relay is returning gateway address fields in Membership Query messages, or
- o Send a Membership Update message to the relay that will delete existing group subscriptions.

#### 5.2.3.9. Handling ICMP Destination Unreachable Responses

A gateway may receive an ICMP "Destination Unreachable" message [RFC0792] after sending an AMT message. Whether the gateway is notified that an ICMP message was received is highly dependent the gateway IP stack behavior and gateway implementation.

If the reception of an ICMP Destination Unreachable message is reported to the gateway while waiting to receive an AMT message, the gateway may respond as follows, depending on platform capabilities and which outgoing message triggered the ICMP response:

1. The gateway MAY simply abandon the current relay and restart relay discovery (if used). This is the least desirable approach as it does not allow for transient network changes.
2. If the last message sent was a Relay Discovery or Request message, the gateway MAY simply ignore the ICMP response and continue waiting for incoming AMT messages. If the gateway is

configured to retransmit Relay Discovery or Request messages, the normal retransmission behavior for those messages is preserved to prevent the gateway from prematurely abandoning a relay.

3. If the last message sent was a Membership Update message, the gateway MAY start a new membership update and associated Request retransmission cycle.

If the reception of an ICMP Destination Unreachable message is reported to the gateway when attempting to transmit a new AMT message, the gateway may respond as follows, depending on platform capabilities and which outgoing message triggered the ICMP response:

1. The gateway MAY simply abandon the current relay and restart relay discovery (if used). This is the least desirable approach as it does not allow for transient network changes.
2. If the last message sent was a Relay Discovery, Request or Teardown message, the gateway MAY attempt to transmit the new message. If the gateway is configured to retransmit Relay Discovery, Request or Teardown messages, the normal retransmission behavior for those messages is preserved to prevent the gateway from prematurely abandoning a relay.
3. If the last message sent was a Membership Update message, the gateway SHOULD start a new membership update and associated Request retransmission cycle.

### 5.3. Relay Operation

The following sections describe relay implementation requirements. A non-normative discussion of relay operation may be found in Section 4.2.

#### 5.3.1. IP/IGMP/MLD Protocol Requirements

A relay requires a subset of router-mode IGMP and MLD functionality to provide group membership tracking and report processing.

A relay accessible via IPv4 MUST support IPv4/IGMPv3 and MAY support IPv6/MLDv2. A relay accessible via IPv6 MUST support IPv6/MLDv2 and MAY support IPv4/IGMPv3.

A relay MUST apply the forwarding rules described in Section 6.3 of [RFC3376] and Section 7.3 of [RFC3810].

A relay MUST handle incoming reports as described in Section 6.4 of [RFC3376] and Section 7.4 of [RFC3810] with the exception that

actions that lead to queries MAY be modified to eliminate query generation. A relay MUST accept IGMP and MLD report datagrams regardless of the IP source address carried by those datagrams.

All other aspects of IGMP/MLD router behavior, such as the handling of queries, querier election, etc., are not used or required for relay operation.

#### 5.3.2. Startup

If a relay is deployed for anycast discovery, the relay MUST advertise an anycast Relay Discovery Address Prefix into the unicast routing system of the anycast domain. An address within that prefix, i.e., a Relay Discovery Address, MUST be assigned to a relay interface.

A unicast IPv4 and/or IPv6 address MUST be assigned to the relay interface that will be used to send and receive AMT control and data messages. This address or addresses are returned in Relay Advertisement messages.

The remaining details of relay "startup" are highly implementation-dependent and are not addressed in this document.

#### 5.3.3. Running

When a relay is started, it begins listening for AMT messages on the interface to which the unicast Relay Address(es) has been assigned, i.e., the address returned in Relay Advertisement messages.

##### 5.3.3.1. Handling AMT Messages

A relay MUST ignore any message other than a Relay Discovery, Request, Membership Update or Teardown message. The handling of Relay Discovery, Request, Membership Update, and Teardown messages is addressed in the sections that follow.

Support for the Teardown message is OPTIONAL. If a relay does not support the Teardown message, it MUST also ignore this message.

A relay that conforms to this specification MUST ignore any message with a Version field value other than zero.

##### 5.3.3.2. Handling a Relay Discovery Message

This section describes relay requirements related to the relay discovery message sequence described in Section 4.2.1.1.

A relay MUST accept and respond to Relay Discovery messages sent to an anycast relay discovery address or the unicast relay address. If a relay receives a Relay Discovery message sent to its unicast address, it MUST respond just as it would if the message had been sent to its anycast discovery address.

When a relay receives a Relay Discovery message it responds by sending a Relay Advertisement message back to the source of the Relay Discovery message. The relay MUST use the source IP address and UDP port of the Relay Discovery message as the destination IP address and UDP port. The relay MUST use the destination IP address and UDP port of the Relay Discovery as the source IP address and UDP port to ensure successful NAT traversal.

The relay MUST copy the value contained in the Discovery Nonce field of the Relay Discovery message into the Discovery Nonce field in the Relay Advertisement message.

If the Relay Discovery message was received as an IPv4 datagram, the relay MUST return an IPv4 address in the Relay Address field of the Relay Advertisement message. If the Relay Discovery message was received as an IPv6 datagram, the relay MUST return an IPv6 address in the Relay Address field.

#### 5.3.3.3. Handling a Request Message

This section describes relay requirements related to the membership query portion of the message sequence described in Section 4.2.1.2.

When a relay receives a Request message it responds by sending a Membership Query message back to the source of the Request message.

The relay MUST use the source IP address and UDP port of the Request message as the destination IP address and UDP port for the Membership Query message. The source IP address and UDP port carried by the Membership Query MUST match the destination IP address and UDP port of the Request to ensure successful NAT traversal.

The relay MUST return the value contained in the Request Nonce field of the Request message in the Request Nonce field of the Membership Query message. The relay MUST compute a MAC value, as described in Section 5.3.5, and return that value in the Response MAC field of the Membership Query message.

If a relay supports the Teardown message, it MUST set the G-flag in the Membership Query message and return the source IP address and UDP port carried by the Request message in the corresponding Gateway IP Address and Gateway Port Number fields. If the relay does not



support the Teardown message it SHOULD NOT set these fields as this may cause the gateway to generate unnecessary Teardown messages.

If the P-flag in the Request message is 0, the relay MUST return an IPv4-encapsulated IGMPv3 general query in the Membership Query message. If the P-flag is 1, the relay MUST return an IPv6-encapsulated MLDv2 general query in the Membership Query message.

If the relay is not accepting Membership Update messages that create new tunnel endpoints due to resource limitations, it SHOULD set the L-flag in the Membership Query message to notify the gateway of this state. Support for the L-flag is OPTIONAL. See Section 5.3.3.8.

The IGMPv3 general query datagram that a relay encapsulates within a Membership Query message MUST conform to the descriptions found in Section 4.1 of [RFC3376]. These datagrams MUST possess the IP headers, header options and header values called for in [RFC3376], with the following exception; the source IP address for an IGMP general query datagram MAY be set to the "unspecified" address (all octets are zero) but SHOULD be set to an address in the address range specifically assigned by IANA for use in the IGMP messages sent from a relay to a gateway (i.e. 154.7.1.1 as described in Section 7). This exception is made because the source address that a relay might normally send may not be a valid source address on any gateway interface. The rationale for using the aforementioned source addresses is primary one of convenience - a gateway will accept an IGMP query regardless of the source address it carries. See Section 5.2.1.

The MLDv2 general query datagram that a relay encapsulates within a Membership Query message MUST conform to the descriptions found in Section 5.1 of [RFC3810]. These datagrams MUST possess the IP headers, header options and header values called for in [RFC3810], with the following exception; the source IP address for an MLD general query datagram MAY be set to the "unspecified" address (all octets are zero) but SHOULD be set to an IPv6 link-local address in the range FE80::/64. A relay may use a dynamically-generated link-local address or the fixed address FE80::2. As with IGMP, a gateway will accept an MLD query regardless of the source address it carries. See Section 5.2.1.

A relay MUST set the Querier's Query Interval Code (QQIC) field in the general query to supply the gateway with a suggested time duration to use for the membership query timer. The QQIC field is defined in Section 4.1.1 in [RFC3376] and Section 5.1.3 in [RFC3810]. A relay MAY adjust this value to affect the rate at which the Request messages are sent from a gateway. However, a gateway is allowed to use a shorter duration than specified in the QQIC field, so a relay

may be limited in its ability to spread out Requests coming from a gateway.

A relay MUST set the Querier's Robustness Variable (QRV) field in the general query to a non-zero value. This value SHOULD be greater than one. If a gateway retransmits membership state change messages, it will retransmit them (robustness variable - 1) times.

A relay SHOULD set the Max Resp Code field in the general query to a value of 1 to trigger an immediate response from the gateway (some host IGMP/MLD implementations may not accept a value of zero). A relay SHOULD NOT use the IGMPv2/MLDv2 Query Response Interval variable, if available, to generate the Max Resp Code field value as the Query Response Interval variable is used in setting the duration of group state timers and must not be set to such a small value. See Section 5.3.3.7.

#### 5.3.3.4. Handling a Membership Update Message

This section describes relay requirements related to the membership update portion of the message sequence described in Section 4.2.1.2.

When a relay receives a Membership Update message it must first determine whether it should accept or ignore the message. A relay MUST NOT make any changes to group membership and forwarding state if the message fails to satisfy any of the following requirements:

- o The IP datagram encapsulated within the message MUST be one of the following:
  - \* IPv4 datagram carrying an IGMPv2 or IGMPv3 Membership Report message.
  - \* IPv4 datagram carrying an IGMPv2 Leave Group message.
  - \* IPv6 datagram carrying an MLDv1 or MLDv2 Multicast Listener Report message.
  - \* IPv6 datagram carrying MLDv1 Multicast Listener Done message.
- o The encapsulated IP datagram MUST satisfy the IP header requirements for the IGMP or MLD message type as described in Section 4 of [RFC3376], Section 2 of [RFC2236], Section 5 of [RFC3810], and Section 3 of [RFC2710], with the following exception - a relay MUST accept an IGMP or MLD message regardless of the IP source address carried by the datagram.

- o The total length of the encapsulated IP datagram as computed from the lengths contained in the datagram header(s) MUST NOT exceed the available field length within the Membership Update message.
- o The computed checksums for the encapsulated IP datagram and its payload MUST match the values contained therein. Checksum computation and verification varies by protocol; See [RFC0791] for IPv4, [RFC3376] for IGMPv3, and [RFC4443] for MLD (ICMPv6).
- o If processing of the encapsulated IGMP or MLD message would result in an allocation of new state or a modification of existing state, the relay MUST authenticate the source of the Membership message by verifying that the value contained in the Response MAC field equals the MAC value computed from the fields in the Membership Update message datagram. Because the private secret used to compute Response MAC values may change over time, the relay MUST retain the previous version of the private secret to use in authenticating Membership Updates sent during the subsequent query interval. If the first attempt at Response MAC authentication fails, the relay MUST attempt to authenticate the Response MAC using the previous private secret value unless  $2 \times \text{query\_interval}$  time has elapsed since the private secret change. See Section 5.3.5. An alternative approach to Response MAC generation that avoids repeated Response MAC computations may be found in Appendix A.1.

A relay MAY skip source authentication to reduce the computational cost of handling Membership Update messages if the relay can make a trivial determination that the IGMP/MLD message carried by the Membership Update message will produce no changes in group membership or forwarding state. The relay does not need to compute and compare MAC values if it finds there are no group subscriptions for the source of the Membership Update message and either of the following is true:

- o The encapsulated IP datagram is an IGMPv3 Membership Report or MLDv2 Multicast Listener Report message that contains no group records. This may often be the case for gateways that continuously repeat the membership update cycle even though they have no group subscriptions to report.
- o The encapsulated IP datagram is an IGMPv2 Leave Group or MLDv1 Multicast Listener Done message.

The IGMP and MLD protocol specifications indicate that senders SHOULD use a link-local source IP address in message datagrams. This requirement must be relaxed for AMT because gateways and relays do not share a common subnet. For this reason, a relay implementation

MUST accept IGMP and MLD datagrams regardless of the source IP address they carry.

Once a relay has determined that the Membership Update message is valid, it processes the encapsulated IGMP or MLD membership message to update group membership state and communicates with the multicast protocol to update forwarding state and possibly send multicast protocol messages towards upstream routers. The relay MUST ignore any octets that might exist between the encapsulated IP datagram and the end of the Membership Update message.

As described in Section 4.2.2, a relay uses the source IP address and source UDP port carried by a Membership Update messages to identify a tunnel endpoint. A relay uses the tunnel endpoint as the destination address for any Multicast Data messages it sends as a result of the group membership and forwarding state created by processing the IGMP/MLD messages contained in Membership Update messages received from the endpoint.

If a Membership Update message originates from a new endpoint, the relay MUST determine whether it can accept updates from a new endpoint. If a relay has been configured with a limit on the total number of endpoints, or a limit on the total number of endpoints for a given source address, then the relay MAY ignore the Membership Update message and possibly withdraw any Relay Discovery Address Prefix announcement that it might have made. See Section 5.3.3.8.

A relay MUST maintain some form of group membership database for each endpoint. The per-endpoint databases are used update a forwarding table containing entries that map an (\*,G) or (S,G) subscription to a list of tunnel endpoints.

A relay MUST maintain some form of group membership database representing a merger of the group membership databases of all endpoints. The merged group membership database is used to update upstream multicast forwarding state.

A relay MUST maintain a forwarding table that maps each unique (\*,G) and (S,G) subscription to a list of tunnel endpoints. A relay uses this forwarding table to provide the destination address when performing UDP/IP encapsulation of the incoming multicast IP datagrams to form Multicast Data messages.

If a group filter mode for a group entry on a tunnel endpoint is EXCLUDE, the relay SHOULD NOT forward datagrams that originate from sources in the filter source list unless the relay architecture does not readily support source filtering. A relay MAY ignore the source list if necessary because gateways are expected to do their own

source filtering.

#### 5.3.3.5. Handling a Teardown Message

This section describes relay requirements related to the teardown message sequence described in Section 4.2.1.3.

When a relay (that supports the Teardown message) receives a Teardown message, it **MUST** first authenticate the source of the Teardown message by verifying that the Response MAC carried by the Teardown message is equal to a MAC value computed from the fields carried by the Teardown message. The method used to compute the MAC differs from that used to generate and validate the Membership Query and Membership Update messages in that the source IP address and source UDP port number used to compute the MAC are taken from the Gateway IP Address and Gateway Port Number field in the Teardown message rather than from the IP and UDP headers in the datagram that carries the Teardown message. The MAC computation is described Section 5.3.5. A relay **MUST** ignore a Teardown message If the computed MAC does not equal the value of the Response MAC field.

If a relay determines that a Teardown message is authentic, it **MUST** immediately stop transmitting Multicast Data messages to the endpoint identified by the Gateway IP Address and Gateway Port Number fields in the message. The relay **MUST** eventually delete any group membership and forwarding state associated with the endpoint, but **MAY** delay doing so to allow a gateway to recreate group membership state on a new endpoint and thereby avoid making unnecessary (temporary) changes in upstream routing/forwarding state.

The state changes made by a relay when processing a Teardown message **MUST** be identical to those that would be made as if the relay had received an IGMP/MLD report that would cause the IGMP or MLD protocol to delete all existing group records in the group membership database associated with the endpoint. The processing of the Teardown message should trigger or mimic the normal interaction between IGMP or MLD and a multicast protocol to produce required changes in forwarding state and possibly send prune/leave messages towards upstream routers.

#### 5.3.3.6. Handling Multicast IP Datagrams

When a multicast IP datagram is forwarded to the relay pseudo-interface, the relay **MUST**, for each gateway that has expressed an interest in receiving the datagram, encapsulate the IP datagram into a Multicast Data message and send that message to the gateway. This process is highly implementation dependent, but conceptually requires the following steps:

- o Use the IP datagram source and destination address to look up the appropriate (\*,G) or (S,G) entry in the endpoint forwarding table created for the pseudo-interface as a result of IGMP/MLD processing.
- o Possibly replicate the datagram for each gateway endpoint listed for that (\*,G) or (S,G) entry.
- o Encapsulate the IP datagram in a UDP/IP Membership Data message, using the endpoint UDP/IP address as the destination address and the unicast relay address and IANA-assigned port as the source UDP/IP address. To ensure successful NAT traversal, the source address and port MUST match the destination address and port carried by the Membership Update message sent by the gateway to create the forwarding table entry.
- o If possible, the relay SHOULD compute a valid, non-zero checksum for the UDP datagram carrying the Membership Data message. See Section 4.2.2.3.
- o Send the message to the gateway.

The relay pseudo-interface MUST ignore any other IP datagrams forwarded to the pseudo-interface.

#### 5.3.3.7. State Timers

A relay MUST maintain a timer or timers whose expiration will trigger the removal of any group subscriptions and forwarding state previously created for a gateway endpoint should the gateway fail to refresh the group membership state within a specified time interval.

A relay MAY use a variant of the IGMPv3/MLDv2 state management protocol described in Section 6 of [RFC3376] or Section 7 of [RFC3810], or may maintain a per-endpoint timer to trigger the deletion of group membership state.

If a per-endpoint timer is used, the relay MUST restart this timer each time it receives a new Membership Update message from the gateway endpoint.

The endpoint timer duration MAY be computed from tunable IGMP/MLD variables as follows:

$$((\text{Robustness\_Variable}) * (\text{Query\_Interval})) + \text{Query\_Response\_Interval}$$

If IGMP/MLD default values are used for these variables, the gateway will timeout after  $125s * 2 + 10s = 260s$ . The timer duration MUST be

greater than the query interval suggested in the last Membership Query message sent to the gateway endpoint.

Regardless of the timers used (IGMPv3/MLDv2 or endpoint), the Query\_Response\_Interval value SHOULD be greater than or equal to 10s to allow for packet loss and round-trip time in the Request/Membership Query message exchange.

#### 5.3.3.8. Relay Resource Management

A relay may be configured with various service limits to ensure a minimum level of performance for gateways that connect to it.

If a relay has determined that it has reached or exceeded maximum allowable capacity or has otherwise exhausted resources required to support additional gateways, it SHOULD withdraw any Relay Discovery Address Prefix it has advertised into the unicast internetwork and SHOULD set the L-flag in any Membership Query messages it returns to gateways while in this state.

If the relay receives an update from a gateway that adds group membership or forwarding state for an endpoint that has already reached maximum allowable state entries, the relay SHOULD continue to accept updates from the gateway but ignore any group membership/forwarding state additions requested by that gateway.

If the relay receives an update from a gateway that would create a new tunnel endpoint for a source IP address that has already reached the maximum allowable number of endpoints (maximum UDP ports), it should simply ignore the Membership Update.

#### 5.3.4. Shutdown

The following steps should be treated as an abstract description of the shutdown procedure for a relay:

- o Withdraw the Relay Discovery Address Prefix advertisement (if used).
- o Stop listening for Relay Discovery messages.
- o Stop listening for control messages from gateways.
- o Stop sending data messages to gateways.
- o Delete all AMT group membership and forwarding state created on the relay, coordinating with the multicast routing protocol to update the group membership state on upstream interfaces as

required.

#### 5.3.5. Response MAC Generation

A Response MAC is produced by a hash digest computation. A Response MAC value is computed from a Request message for inclusion in a Membership Query message, is computed from a Membership Update message to authenticate the Response MAC carried within that message, and is computed from fields in a Teardown message to authenticate the Response MAC carried within that message.

Gateways treat the Response MAC field as an opaque value, so a relay implementation may generate the MAC using any method available to it. The hash function RECOMMENDED for use in computing the Response MAC is the MD5 hash digest [RFC1321], though hash functions or keyed-hash functions of greater cryptographic strength may be used.

The digest MUST be computed over the following values:

- o The Source IP address of the message (or Teardown Gateway IP Address field)
- o The Source UDP port of the message (or Teardown Gateway Port Number field)
- o The Request Nonce contained in the message.
- o A private secret known only to the relay

An Response MAC generation solution that satisfies these requirements is described in Appendix A.1.

#### 5.3.6. Private Secret Generation

The private secret, or hash-key, is a random value that the relay includes in the Response MAC hash digest computation. A relay SHOULD periodically compute a new private secret. The RECOMMENDED maximum interval is 2 hours. A relay MUST retain the prior secret for use in verifying MAC values that were sent to gateways just prior to the use of the new secret.

The private secret SHOULD be computed using a cryptographically-secure pseudo-random number generator. The private secret width SHOULD equal that of the hash function used to compute the Response MAC, e.g., 128-bits for an MD5 hash.



## 6. Security Considerations

AMT is not intended to be a strongly secured protocol. In general, the protocol provides the same level of security and robustness as is provided by the UDP, IGMP and MLD protocols on which it relies. The lack of strong security features can largely be attributed to the desire to make the protocol light-weight by minimizing the state and computation required to service a single gateway, thereby allowing a relay to service a larger number of gateways.

Many of the threats and vectors described in [RFC3552] may be employed against the protocol to launch various types of denial-of-service attacks that can affect the functioning of gateways or their ability to locate and communicate with a relay. These scenarios are described below.

As is the case for UDP, IGMP and MLD, the AMT protocol provides no mechanisms for ensuring message delivery or integrity. The protocol does not provide confidentiality - multicast groups, sources and streams requested by a gateway are sent in the clear.

The protocol does use a three-way handshake to provide trivial source authentication for state allocation and updates (see below). The protocol also requires gateways and relays to ignore malformed messages and those messages that do not carry expected address values or protocol payload types or content.

### 6.1. Relays

The three-way handshake provided by the membership update message sequence (See (Section 4.2.1.2)) provides a defense against source-spoofing-based resource-exhaustion attacks on a relay by requiring source authentication before state allocation. However, attackers may still attempt to flood a relay with Request and Membership Update messages to force the relay to make the hash computations in an effort to consume computational resources. Implementations may choose to limit the frequency with which a relay responds to Request messages sent from a single IP address or IP address and UDP port pair, but support for this functionality is not required. The three-way handshake provides no defense against an eavesdropping or man-in-the-middle attacker.

Attackers that execute the gateway protocol may consume relay resources by instantiating a large number of tunnels or joining a large number of multicast streams. A relay implementation should provide a mechanism for limiting the number of tunnels (Multicast Data message destinations) that can be created for a single gateway source address. Relays should also provide a means for limiting the

number of joins per tunnel instance as a defense against these attacks.

Relays may withdraw their AMT anycast prefix advertisement when they reach configured maximum capacity or exhaust required resources. This behavior allows gateways to use the relay discovery process to find the next topologically-nearest relay that has advertised the prefix. This behavior also allows a successful resource exhaustion attack to propagate from one relay to the next until all relays reachable using the anycast address have effectively been taken offline. This behavior may also be used to acquire the unicast addresses for individual relays which can then be used to launch a DDoS attack on all of the relays without using the relay discovery process. To prevent wider disruption of AMT-based distribution network, relay anycast address advertisements can be limited to specific administrative routing domains. This will isolate such attacks to a single domain.

## 6.2. Gateways

A passive eavesdropper may launch a denial-of-service attack on a gateway by capturing a Membership Query or Membership Update message and using the request nonce and message authentication code carried by the captured message to send a spoofed a Membership Update or Teardown message to the relay. The spoofed messages may be used to modify or destroy group membership state associated with the gateway, thereby changing or interrupting the multicast traffic flows.

A passive eavesdropper may also spoof Multicast Data messages in an attempt to overload the gateway or disrupt or supplant existing traffic flows. A properly implemented gateway will filter Multicast Data messages that do not originate from the expected relay address and should filter non-multicast packets and multicast IP packets whose group or source addresses are not included in the current reception state for the gateway pseudo-interface.

An active eavesdropper may launch a man-in-the-middle attack in which messages normally exchanged between a gateway and relay are intercepted, modified, spoofed or discarded by the attacker. The attacker may deny access to, modify or replace requested multicast traffic. The AMT protocol provides no means for detecting or defending against a man-in-the-middle attack - any such functionality must be provided by multicast receiver applications through independent detection and validation of incoming multicast datagrams.

The anycast discovery technique for finding relays (see Section 4.1.4) introduces a risk that a rogue router or a rogue AS could introduce a bogus route to a specific Relay Discovery Address

prefix, and thus divert or absorb Relay Discovery messages sent by gateways. Network managers must guarantee the integrity of their routing to a particular Relay Discovery Address prefix in much the same way that they guarantee the integrity of all other routes.

### 6.3. Encapsulated IP Packets

An attacker forging or modifying a Membership Query or Membership Update message may attempt to embed something other than an IGMP or MLD message within the encapsulated IP packet carried by these messages in an effort to introduce these into the recipient's IP stack. A properly implemented gateway or relay will ignore any such messages – and may further choose to ignore Membership Query messages that do not contain a IGMP/MLD general queries or Membership Update messages that do not contain IGMP/MLD membership reports.

Properly implemented gateways and relays will also filter encapsulated IP packets that appear corrupted or truncated by verifying packet length and checksums.

## 7. IANA Considerations

### 7.1. IPv4 and IPv6 Anycast Prefix Allocation

IANA should allocate an IPv4 prefix and an IPv6 prefix dedicated to the public AMT Relays to advertise to the native multicast backbone (as described in Section 4.1.4). The prefix length should be determined by the IANA; the prefix should be large enough to guarantee advertisement in the default-free BGP networks.

#### 7.1.1. IPv4

A prefix length of 24 will meet this requirement.

Internet Systems Consortium (ISC) has offered 154.7.0/24 for this purpose.

#### 7.1.2. IPv6

A prefix length of 32 will meet this requirement. IANA has previously set aside the range 2001::/16 for allocating prefixes for this purpose.

### 7.2. IPv4 Address Prefix Allocation for IGMP Source Addresses

IANA should allocate an IPv4 prefix dedicated for use in IGMP messages exchanged between gateways and relays. This address range is intended for use within tunnels constructed between a gateway and relay, and as such, is not intended to be globally routable.

A prefix length of 24 will meet this requirement.

Internet Systems Consortium (ISC) has offered 154.7.1/24 for this purpose.

### 7.3. UDP Port number

IANA has reserved UDP port number 2268 for AMT.

## 8. Contributors

The following people provided significant contributions to the design of the protocol and earlier versions of this specification:

Thomas Morin  
France Telecom - Orange  
2, avenue Pierre Marzin  
Lannion 22300  
France  
Email: thomas.morin@orange.com

Dirk Ooms  
OneSparrow  
Belegstraat 13; 2018 Antwerp;  
Belgium  
EMail: dirk@onesparrow.com

Tom Pusateri  
!j  
2109 Mountain High Rd.  
Wake Forest, NC 27587  
USA  
Email: pusateri@bangj.com

Dave Thaler  
Microsoft Corporation  
One Microsoft Way  
Redmond, WA 98052-6399  
USA  
Email: dthaler@microsoft.com

## 9. Acknowledgments

The authors would like to thank the following individuals for their suggestions, comments, and corrections:

Amit Aggarwal  
Mark Altom  
Toerless Eckert  
Marshall Eubanks  
Gorry Fairhurst  
Dino Farinacci  
Lenny Giuliano  
Andy Huang  
Tom Imburgia  
Patricia McCrink  
Han Nguyen  
Doug Nortz  
Pekka Savola  
Robert Sayko  
Greg Shepherd  
Steve Simlo  
Mohit Talwar  
Lorenzo Vicisano  
Kurt Windisch  
John Zwiebel

The anycast discovery mechanism described in this document is based on similar work done by the NGTrans WG for obtaining automatic IPv6 connectivity without explicit tunnels ("6to4"). Tony Ballardie provided helpful discussion that inspired this document.

Juniper Networks was instrumental in funding several versions of this draft as well as an open source implementation.

## 10. References

### 10.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.
- [RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, August 2006.
- [RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

### 10.2. Informative References

- [I-D.ietf-6man-udpchecksums]  
Eubanks, M. and P. Chimento, "UDP Checksums for Tunneled Packets", draft-ietf-6man-udpchecksums-02 (work in progress), March 2012.
- [I-D.ietf-6man-udpzero]  
Fairhurst, G. and M. Westerlund, "IPv6 UDP Checksum Considerations", draft-ietf-6man-udpzero-05 (work in progress), December 2011.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791,

September 1981.

- [RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, August 1989.
- [RFC1546] Partridge, C., Mendez, T., and W. Milliken, "Host Anycasting Service", RFC 1546, November 1993.
- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.
- [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.
- [RFC3053] Durand, A., Fasano, P., Guardini, I., and D. Lento, "IPv6 Tunnel Broker", RFC 3053, January 2001.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.
- [RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, January 2005.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol



Version 6 (IPv6) Specification", RFC 4443, March 2006.

- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.
- [RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, December 2006.

## Appendix A. Implementation Notes

### A.1. Response MAC Generation and Keying

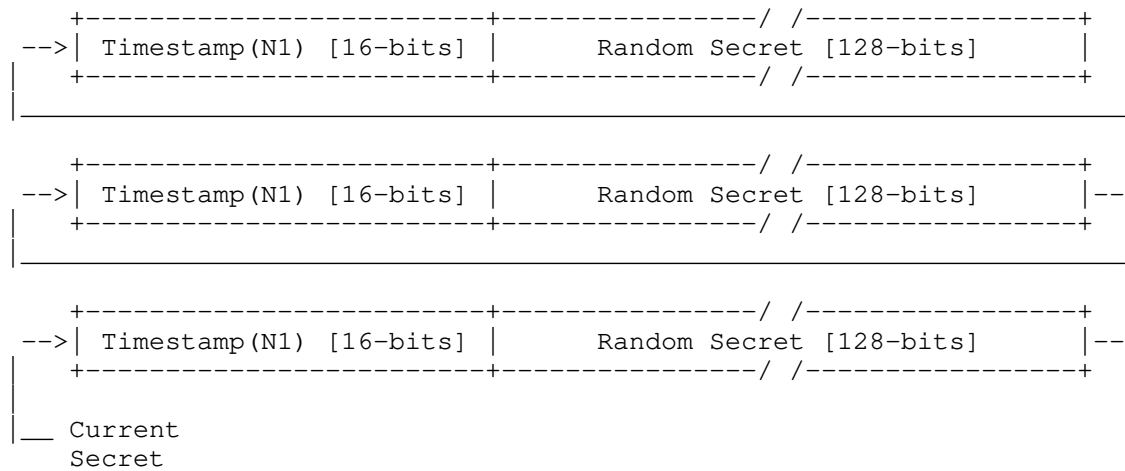
This specification does not require relays to use any particular method to compute the Response MAC field value - only that it contain a hash of the source IP address, source UDP port, request nonce, and a private secret known only to the relay. This allows the relay implementor a significant amount of leeway in the computation and structure of the value stored in the Response MAC field.

Section 5.3.6 states that a relay should periodically compute a new private secret (or hash-key) for MAC generation. To prevent the relay from rejecting Membership Update messages that contain Response MAC values computed from an old secret, the relay is required to retain the previous secret so that it can re-attempt authentication using the old secret, should authentication fail after recomputing the MAC using the new secret. However, this approach requires a relay to do at least two hash computations for every Membership Update message that carries an old or a invalid MAC. A better approach would be to include information within the message that the relay could use to choose a single secret for authentication rather relying on sequential authentication failures to test all possible secrets.

The solution proposed here is to compute and exchange an "authentication cookie" rather than a simple hash value in the Response MAC field. The authentication cookie would combine a timestamp with a hash value. The timestamp is used to calculate the age of the cookie, allowing the relay to reject a message if the cookie's age is greater than some maximum allowable value. If the cookie has not expired, the relay uses the timestamp to lookup the secret that was in use at that time and then compute and compare the hash portion of the cookie to authenticate the message source.

A second purpose served by including the timestamp in the MAC field is that it allows the relay to contribute an unpredictable value to the authentication hash. This contribution provides a defense against attempts to use a hash reversal algorithm to determine the relay's private secret as the hash result will change over time even if the nonce carried by the Request message does not.





#### Private Secret Queue

The timestamp is not only used to compute the age of the MAC, but is also used to lookup the private secret used to generate the MAC. Each time a new private secret is computed, the value and the time at which the value was computed is pushed into a fixed-length queue of recent values (typically only 2-deep). The relay uses the timestamp contained in the MAC field to lookup the appropriate secret. The relay iterates over the list of secrets, starting with the newest entry, until it finds the first secret with a timestamp that is older than that contained in the MAC field. The relay then uses that secret to compute the MAC that will be compared with that carried by the message.

Author's Address

Gregory Bumgardner  
Cisco  
3700 Cisco Way  
San Jose, CA 95134  
USA

Phone: +1 408 853 4993  
Email: gbumgard@cisco.com



MBONED Working Group  
Internet Draft  
Intended status: Standard  
Expires: December 2012

N. Kumar  
S. Venaas  
Cisco Systems, Inc  
June 28, 2012

PIM/MLD flags for IPv4-IPv6 Multicast Translation Procedure  
draft-kumar-mboned-64mcast-embedded-address-00.txt

#### Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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/lid-abstracts.txt>

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

This Internet-Draft will expire on December 28, 2012.

#### Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal

Provisions and are provided without warranty as described in  
the Simplified BSD License.

## Abstract

This document discusses the procedure that helps to identify  
IPv4 embedded IPv6 Multicast address without any embedded  
flags in the address. This document specifies the usage of  
additional data or attribute in MLD and PIM that helps  
identify this address. This document is not conclusive and is  
open for discussion.

## Table of Contents

1. Introduction.....	2
2. Conventions used in this document.....	3
3. Terminology.....	4
4. Procedure.....	4
4.1. 64I Join Attribute.....	5
4.2. 64I Auxiliary Data.....	5
5. Use Cases.....	6
5.1. IPv4 Receiver and Source connected over IPv6-Only network.....	6
5.2. IPv6 Receiver Connected to IPv4 Source through IPv4 multicast access network and IPv6 Multicast network.....	7
5.3. IPv6 Receiver and IPv4 Source.....	9
6. Security Considerations.....	10
7. IANA Considerations.....	10
8. References.....	10
8.1. Normative References.....	10
8.2. Informative References.....	10
9. Acknowledgments.....	11

## 1. Introduction

As part of IPv4 to IPv6 migration, there are multiple  
standards developed for smooth transition for Unicast.  
Section 3 of [I-D.ietf-mboned-v4v6-mcast-ps] specifies



different possible scenarios for IPv4 to IPv6 multicast transition as below,

1. IPv4 Receiver and Source connected over IPv6-Only network
2. IPv6 Receiver Connected to IPv4 Source through IPv4 multicast access network and IPv6 Multicast network.
3. IPv6 Receiver and Source connected to IPv4-Only network.
4. IPv6 Receiver and IPv4 Source.
5. IPv4 Receiver and IPv6 Source.

Section 3.6 of [I-D.ietf-mboned-v4v6-mcast-ps] identifies the use cases involving IPv4 source as highest priority.

There are also various solutions proposed (ex., [I-D.ietf-software-mesh-multicast], [I-D.ietf-software-dslite-multicast]) addressing the above use cases requirement which requires to embed IPv4 multicast address into IPv6 address. This IPv4-embedded IPv6 multicast address will be used as group address within IPv6 cloud.

Currently [I-D.ietf-mboned-64-multicast-address-format] defines a new bit in IPv6 Multicast address that signals any router that IPv4 Multicast address is embedded as last 32 bits. This may create backward compatibility issue.

This document defines a set of procedures, a new PIM join attribute [RFC 5384] and a new MLD Auxiliary Data that helps achieve the above without a need for any bit embedded within IPv6 Multicast address.

## 2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

### 3. Terminology

(S4, G4)/(\*, G4): (S, G) or (\*, G) in IPv4 address format

(S6, G6)/(\*, G6): (S, G) or (\*, G) in IPv6 address format

### 4. Procedure

Any AFBR on receiving (S4, G4) or (\*, G4) PIM join or IGMP Report message and if the S6 after translation is not IPv4 translatable address and if the upstream is IPv6 PIM neighbor MUST include transitive 64I JOIN ATTRIBUTE (Section 4.1) in IPv6 PIM Join and embed IPv4 group address in last 32 bits of IPv6 Multicast SSM range address.

Any AFBR on receiving (S4, G4) or (\*, G4) PIM join or IGMP Report message and if the S6 after translation is IPv4 translatable address and if the upstream is IPv6 PIM neighbor SHOULD include transitive 64I JOIN ATTRIBUTE in IPv6 PIM Join and embed IPv4 group address in last 32 bits of IPv6 Multicast SSM range address.

Any AFBR on receiving (S4, G4) or (\*, G4) PIM Join or IGMP Report message and if S6 after translation is not IPv4 translatable address and if upstream is IPv6 cloud without PIM neighbor MUST include 64I Auxiliary Data (Section 4.2) in MLDv2 Report Message.

Any AFBR on receiving (S4, G4) or (\*, G4) PIM Join or IGMP Report message and if S6 after translation is IPv4 translatable address and if upstream is IPv6 cloud without PIM neighbor SHOULD include 64I Auxillary Data in MLDv2 Report Message.

Any AFBR on receiving IPv4 PIM Join with 64I JOIN ATTRIBUTE MUST carry forward the attribute in IPv6 PIM Join sent upstream.

Any router on receiving IPv6 PIM Join with 64I JOIN ATTRIBUTE and if upstream is IPv6 cloud without PIM neighbor MUST include 64I Auxillary Data in MLDv2 Report message.

Any AFBR on receiving (S6, G6) PIM Join for SSM range address without 64I JOIN ATTRIBUTE and if the IPv6 Source in Join is well known prefix (64:FF9B::/96) or IPv4 translatable IPv6

address [RFC 6052] and if the upstream is IPv4 PIM neighbor,  
 MUST pull the last 32 bits to generate IPv4 group address.

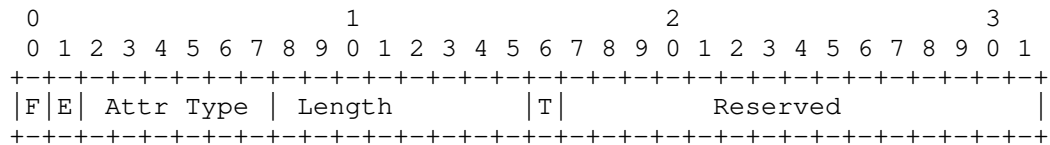
Any router on receiving (S6, G6) PIM Join from SSM range  
 without 64I JOIN ATTRIBUTE and if Source address is well  
 known prefix (64:FF9B::/96) or IPv4 translatable IPv6 address  
 [RFC 6052] and if the upstream is IPv6 PIM neighbor, MUST  
 include 64I JOIN ATTRIBUTE.

Any router on receiving MLD Report with 64I Auxiliary Data  
 MUST include 64I JOIN ATTRIBUTE in IPv6 PIM join sent  
 Upstream for the group.

While the above procedure is defined with SSM range address  
 as an example, it is applicable for any (S6, G6) from ASM  
 range.

#### 4.1. 64I Join Attribute

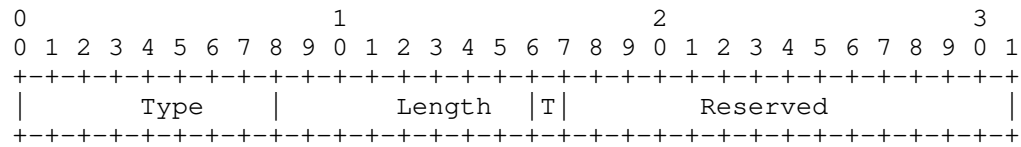
Below is the format of new PIM JOIN ATTRIBUTE specified in  
 this document,



F bit: 1, Transitive Attribute  
 E bit: As mentioned in [RFC 5384]  
 Attr Type: TBD  
 Length: 2  
 T bit: 1  
 Reserved: Reserved field for future use.

#### 4.2. 64I Auxiliary Data

Below is the format of new Auxiliary Data specified in this  
 document,



Type: TBD  
Length:  
T Flag: 1  
Reserved: Reserved bit for future use.

## 5. Use Cases

In this document, we also specify the behavior of high priority scenarios with above procedure.

### 5.1. IPv4 Receiver and Source connected over IPv6-Only network

This scenario simply known as 4-6-4 is shown below in Figure 1.

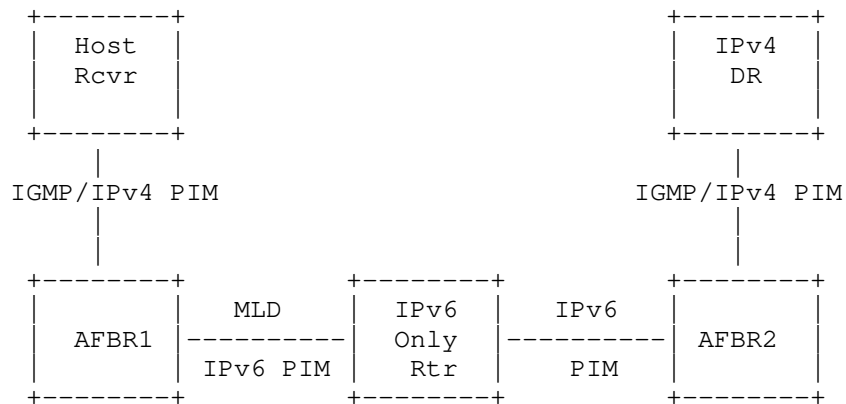


Figure 1: 4-6-4 Scenario

AFBR1 on receiving (S4, G4) or (\*, G4) PIM Join or IGMP Report will perform the below,

1. If Upstream is IPv6 PIM neighbor, should embed the IPv4 multicast group into last 32 bits of IPv6 Multicast SSM range address and send (S6, G6) PIM join with 64I JOIN ATTRIBUTE.
2. If Upstream is IPv6 MLD router, should embed the IPv4 multicast group into last 32 bits of IPv6 Multicast SSM range address and send MLDv2 Report with 64I Auxillary Data.

AFBR2 on receiving (S6, G6) PIM Join without 64I JOIN ATTRIBUTE and if upstream is IPv4 cloud can derive the IPv4 multicast group address from last 32 bits.

Since F bit will be set in 64I JOIN ATTRIBUTE, it will be delivered to AFBR2 even if any router along the path doesn't understand the attribute.

IPv6-only Rtr on receiving (S6, G6) PIM Join with 64I JOIN ATTRIBUTE will send across to AFBR2 with attribute. Since 64I JOIN ATTRIBUTE is transitive in nature, this behavior doesn't change even if IPv6-Only Rtr doesn't understand the attribute.

IPv6-only Rtr on receiving (S6, G6) MLD Report with 64I Auxiliary Data will include 64I JOIN ATTRIBUTE in upstream PIM join for (S6, G6).

AFBR2 on receiving (S6, G6) PIM Join with 64I JOIN ATTRIBUTE must derive the IPv4 multicast group address from the last 32 bits.

## 5.2. IPv6 Receiver Connected to IPv4 Source through IPv4 multicast access network and IPv6 Multicast network

This scenario simply known as 6-4-6-4 is shown in Figure 2.

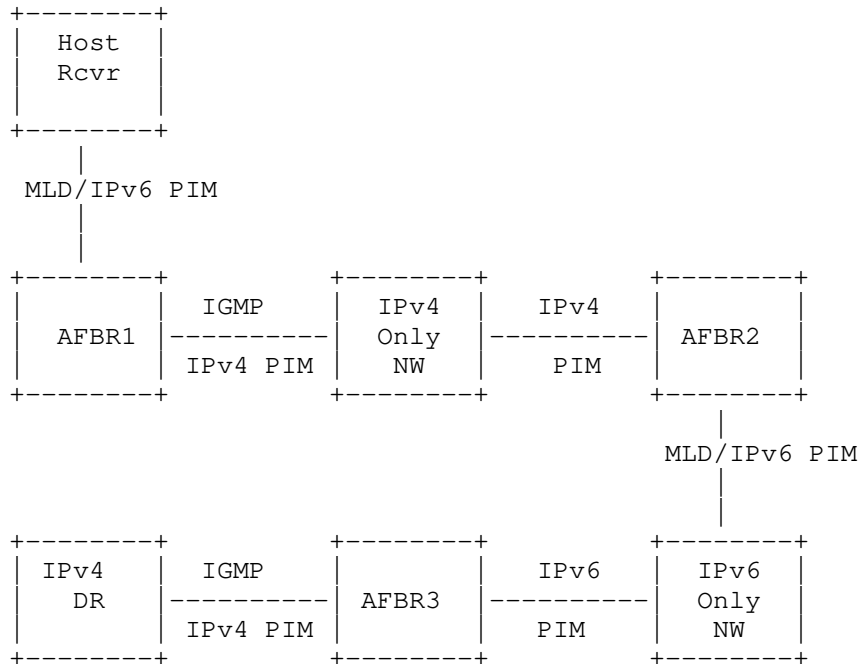


Figure 2: 6-4-6-4 Scenario

In Figure 2, AFBR3 will act as IP/ICMP translator and will advertise IPv4 prefixes into IPv6 cloud as either well known prefix (64:FF9B::/96) or IPv4 translatable IPv6 prefix.

In this scenario, AFBR1 or the DR router MUST include 64I JOIN ATTRIBUTE or 64I Auxiliary Data if the source is well known prefix (64:FF9B::/96). AFBR1 or the DR router SHOULD include 64I JOIN ATTRIBUTE or 64I Auxiliary Data if the source is with IPv4 translatable IPv6 prefix. How AFBR1/DR will understand if S6 belongs to IPv4 translatable IPv6 prefix is outside the scope of this document.

Various solutions are available by which AFBR1 will send the join towards AFBR2. This basically depends if multicast is enabled or disabled on IPv4 cloud. Depending on the solution,

AFBR1 will either send IPv6 PIM Join encapsulated within IPv4 PIM join or IPv6 PIM Join over some tunnel.

AFBR2 on receiving (S6, G6) PIM Join over tunnel or (S6, G6) PIM Join encapsulated within (S4, G4) will send 64I JOIN ATTRIBUTE or 64I Auxiliary Data upstreams towards AFBR3.

AFBR3 on receiving (S6, G6) Join with 64I JOIN ATTRIBUTE MUST derive the IPv4 group address from last 32 bits.

AFBR3 on receiving (S6, G6) PIM join without 64I JOIN ATTRIBUTE MUST check if S6 falls within well known prefix (64:FF9B::/96) or IPv4 translatable IPv6 Prefix. If S6 is within the above range, it MUST derive IPv4 group from the last 32 bits of G6.

### 5.3. IPv6 Receiver and IPv4 Source

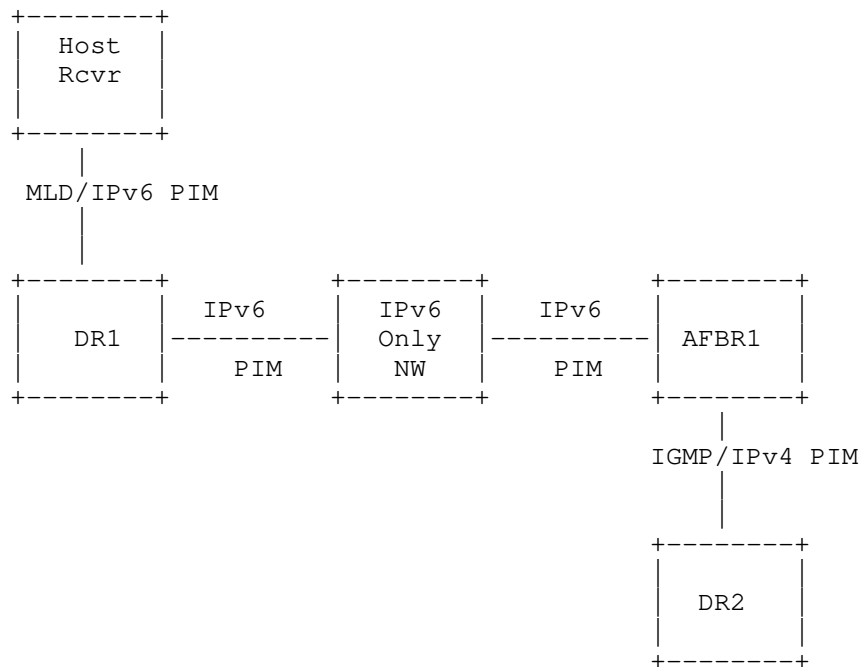


Figure 3: 6-4 Scenario

This scenario works similar to Section 5.2 except that IPv6 cloud is not partitioned by IPv4 cloud.

## 6. Security Considerations

Security consideration specified in [RFC 5384] and [RFC 6052] are applicable here as well.

## 7. IANA Considerations

TBD.

## 8. References

### 8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC 5234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 5234, January 2008.

### 8.2. Informative References

[I-D.ietf-mboned-v4v6-mcast-ps]  
Jacquenet, C., Boucadair, M., Lee, Y., Qin, J., Tsou, T., and Q. Sun, "IPv4-IPv6 Multicast: Problem Statement and Use Cases", draft-ietf-mboned-v4v6-mcast-ps-00 (work in progress), May 2012.

[I-D.ietf-mboned-64-multicast-address-format]  
Boucadair, M., Qin, J., Lee, Y., Venaas, S., Li, X. and Xu, M, "IPv4-Embedded IPv6 Multicast Address Format", draft-ietf-mboned-64-multicast-address-format-02 (work in progress), February



2012.

[I-D.ietf-softwire-dslite-multicast]

Qin, J., Boucadair, M., Jacquenet, C., Lee, Y.,  
and Q. Wang, "Multicast Extension to DS-Lite  
Technique in Broadband Deployments",  
Draft-ietf-softwire-dslite-multicast-02 (work in  
progress), May 2012.

[I-D.ietf-softwire-mesh-multicast]

Xu, M., Cui, Y., Yang, S., Wu, J., Metz, C., and  
G. Shepherd, "Softwire Mesh Multicast",  
Draft-ietf-softwire-mesh-multicast-02 (work in  
progress), April 2012.

[RFC 5384] Boers, A., Wijnands, I. and Rosen, E., "The  
Protocol Independent Multicast (PIM) Join  
Attribute Format", RFC 5384, Nov 2008.

[RFC 4291] Hinden, R. and S. Deering, "IP Version 6  
Addressing Architecture", RFC 4291, February 2006.

[RFC 4607] Holbrook, H. and B. Cain "Source-Specific  
Multicast for IP", RFC 4607, August 2006.

[RFC 6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M.,  
and X. Li, "IPv6 Addressing of IPv4/IPv6  
Translators", RFC 6052, October 2010.

## 9. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

## Authors' Addresses

Stig Venaas  
Cisco Systems, Inc.  
Tasman Drive  
San Jose, CA 95134  
USA  
Email: stig@cisco.com

Internet-DraftPIM/MLD flags for IPv4-IPv6 Multicast Translation  
Procedure June 2012

Nagendra Kumar  
Cisco Systems  
Cessna Business Park, Sarjapura Marathalli Outer Ring Road  
Bangalore, KARNATAKA 560 087  
India  
Email: naikumar@cisco.com



Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: January 17, 2013

M. McBride  
H. Lui  
Huawei Technologies  
July 16, 2012

Multicast in the Data Center Overview  
draft-mcbride-armd-mcast-overview-02

Abstract

There has been much interest in issues surrounding massive amounts of hosts in the data center. These issues include the prevalent use of IP Multicast within the Data Center. Its important to understand how IP Multicast is being deployed in the Data Center to be able to understand the surrounding issues with doing so. This document provides a quick survey of uses of multicast in the data center and should serve as an aid to further discussion of issues related to large amounts of multicast in the data center.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 17, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Multicast Applications in the Data Center . . . . .	3
2.1. Client-Server Applications . . . . .	3
2.2. Non Client-Server Multicast Applications . . . . .	4
3. L2 Multicast Protocols in the Data Center . . . . .	6
4. L3 Multicast Protocols in the Data Center . . . . .	7
5. Challenges of using multicast in the Data Center . . . . .	7
6. Layer 3 / Layer 2 Topological Variations . . . . .	9
7. Address Resolution . . . . .	9
7.1. Solicited-node Multicast Addresses for IPv6 address resolution . . . . .	9
7.2. Direct Mapping for Multicast address resolution . . . . .	9
8. Acknowledgements . . . . .	10
9. IANA Considerations . . . . .	10
10. Security Considerations . . . . .	10
11. Informative References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

Data center servers often use IP Multicast to send data to clients or other application servers. IP Multicast is expected to help conserve bandwidth in the data center and reduce the load on servers. IP Multicast is also a key component in several data center overlay solutions. Increased reliance on multicast, in next generation data centers, requires higher performance and capacity especially from the switches. If multicast is to continue to be used in the data center, it must scale well within and between datacenters. There has been much interest in issues surrounding massive amounts of hosts in the data center. There was a discussion, in ARMD, involving the issues with address resolution for non ARP/ND multicast traffic in data centers. This document provides a quick survey of multicast in the data center and should serve as an aid to further discussion of issues related to multicast in the data center.

ARP/ND issues are not addressed in this document except to explain how address resolution occurs with multicast. ARP/ND issues are addressed in [I-D.armd-problem-statement]

## 2. Multicast Applications in the Data Center

There are many data center operators who do not deploy Multicast in their networks for scalability and stability reasons. There are also many operators for whom multicast is critical and is enabled on their data center switches and routers. For this latter group, there are several uses of multicast in their data centers. An understanding of the uses of that multicast is important in order to properly support these applications in the ever evolving data centers. If, for instance, the majority of the applications are discovering/signaling each other, using multicast, there may be better ways to support them then using multicast. If, however, the multicasting of data is occurring in large volumes, there is a need for good data center overlay multicast support. The applications either fall into the category of those that leverage L2 multicast for discovery or of those that require L3 support and likely span multiple subnets.

### 2.1. Client-Server Applications

IPTV servers use multicast to deliver content from the data center to end users. IPTV is typically a one to many application where the hosts are configured for IGMPv3, the switches are configured with IGMP snooping, and the routers are running PIM-SSM mode. Often redundant servers are sending multicast streams into the network and the network is forwarding the data across diverse paths.

Windows Media servers send multicast streaming to clients. Windows Media Services streams to an IP multicast address and all clients subscribe to the IP address to receive the same stream. This allows a single stream to be played simultaneously by multiple clients and thus reducing bandwidth utilization.

Market data relies extensively on IP multicast to deliver stock quotes from the data center to a financial services provider and then to the stock analysts. The most critical requirement of a multicast trading floor is that it be highly available. The network must be designed with no single point of failure and in a way the network can respond in a deterministic manner to any failure. Typically redundant servers (in a primary/backup or live live mode) are sending multicast streams into the network and the network is forwarding the data across diverse paths (when duplicate data is sent by multiple servers).

With publish and subscribe servers, a separate message is sent to each subscriber of a publication. With multicast publish/subscribe, only one message is sent, regardless of the number of subscribers. In a publish/subscribe system, client applications, some of which are publishers and some of which are subscribers, are connected to a network of message brokers that receive publications on a number of topics, and send the publications on to the subscribers for those topics. The more subscribers there are in the publish/subscribe system, the greater the improvement to network utilization there might be with multicast.

## 2.2. Non Client-Server Multicast Applications

Routers, running Virtual Routing Redundancy Protocol (VRRP), communicate with one another using a multicast address. VRRP packets are sent, encapsulated in IP packets, to 224.0.0.18. A failure to receive a multicast packet from the master router for a period longer than three times the advertisement timer causes the backup routers to assume that the master router is dead. The virtual router then transitions into an unsteady state and an election process is initiated to select the next master router from the backup routers. This is fulfilled through the use of multicast packets. Backup router(s) are only to send multicast packets during an election process.

Overlays may use IP multicast to virtualize L2 multicasts. IP multicast is used to reduce the scope of the L2-over-UDP flooding to only those hosts that have expressed explicit interest in the frames. VXLAN, for instance, is an encapsulation scheme to carry L2 frames over L3 networks. The VXLAN Tunnel End Point (VTEP) encapsulates frames inside an L3 tunnel. VXLANs are identified by a

24 bit VXLAN Network Identifier (VNI). The VTEP maintains a table of known destination MAC addresses, and stores the IP address of the tunnel to the remote VTEP to use for each. Unicast frames, between VMs, are sent directly to the unicast L3 address of the remote VTEP. Multicast frames are sent to a multicast IP group associated with the VNI. Underlying IP Multicast protocols (PIM-SM/SSM/BIDIR) are used to forward multicast data across the overlay.

The Ganglia application relies upon multicast for distributed discovery and monitoring of computing systems such as clusters and grids. It has been used to link clusters across university campuses and can scale to handle clusters with 2000 nodes

Windows Server, cluster node exchange, relies upon the use of multicast heartbeats between servers. Only the other interfaces in the same multicast group use the data. Unlike broadcast, multicast traffic does not need to be flooded throughout the network, reducing the chance that unnecessary CPU cycles are expended filtering traffic on nodes outside the cluster. As the number of nodes increases, the ability to replace several unicast messages with a single multicast message improves node performance and decreases network bandwidth consumption. Multicast messages replace unicast messages in two components of clustering:

- o Heartbeats: The clustering failure detection engine is based on a scheme whereby nodes send heartbeat messages to other nodes. Specifically, for each network interface, a node sends a heartbeat message to all other nodes with interfaces on that network. Heartbeat messages are sent every 1.2 seconds. In the common case where each node has an interface on each cluster network, there are  $N * (N - 1)$  unicast heartbeats sent per network every 1.2 seconds in an N-node cluster. With multicast heartbeats, the message count drops to N multicast heartbeats per network every 1.2 seconds, because each node sends 1 message instead of  $N - 1$ . This represents a reduction in processing cycles on the sending node and a reduction in network bandwidth consumed.
- o Regroup: The clustering membership engine executes a regroup protocol during a membership view change. The regroup protocol algorithm assumes the ability to broadcast messages to all cluster nodes. To avoid unnecessary network flooding and to properly authenticate messages, the broadcast primitive is implemented by a sequence of unicast messages. Converting the unicast messages to a single multicast message conserves processing power on the sending node and reduces network bandwidth consumption.

Multicast addresses in the 224.0.0.x range are considered link local multicast addresses. They are used for protocol discovery and are



flooded to every port. For example, OSPF uses 224.0.0.5 and 224.0.0.6 for neighbor and DR discovery. These addresses are reserved and will not be constrained by IGMP snooping. These addresses are not to be used by any application.

### 3. L2 Multicast Protocols in the Data Center

The switches, in between the servers and the routers, rely upon igmp snooping to bound the multicast to the ports leading to interested hosts and to L3 routers. A switch will, by default, flood multicast traffic to all the ports in a broadcast domain (VLAN). IGMP snooping is designed to prevent hosts on a local network from receiving traffic for a multicast group they have not explicitly joined. It provides switches with a mechanism to prune multicast traffic from links that do not contain a multicast listener (an IGMP client). IGMP snooping is a L2 optimization for L3 IGMP.

IGMP snooping, with proxy reporting or report suppression, actively filters IGMP packets in order to reduce load on the multicast router. Joins and leaves heading upstream to the router are filtered so that only the minimal quantity of information is sent. The switch is trying to ensure the router only has a single entry for the group, regardless of how many active listeners there are. If there are two active listeners in a group and the first one leaves, then the switch determines that the router does not need this information since it does not affect the status of the group from the router's point of view. However the next time there is a routine query from the router the switch will forward the reply from the remaining host, to prevent the router from believing there are no active listeners. It follows that in active IGMP snooping, the router will generally only know about the most recently joined member of the group.

In order for IGMP, and thus IGMP snooping, to function, a multicast router must exist on the network and generate IGMP queries. The tables (holding the member ports for each multicast group) created for snooping are associated with the querier. Without a querier the tables are not created and snooping will not work. Furthermore IGMP general queries must be unconditionally forwarded by all switches involved in IGMP snooping. Some IGMP snooping implementations include full querier capability. Others are able to proxy and retransmit queries from the multicast router.

In source-only networks, however, which presumably describes most data center networks, there are no IGMP hosts on switch ports to generate IGMP packets. Switch ports are connected to multicast source ports and multicast router ports. The switch typically learns about multicast groups from the multicast data stream by using a type

of source only learning (when only receiving multicast data on the port, no IGMP packets). The switch forwards traffic only to the multicast router ports. When the switch receives traffic for new IP multicast groups, it will typically flood the packets to all ports in the same VLAN. This unnecessary flooding can impact switch performance.

#### 4. L3 Multicast Protocols in the Data Center

There are three flavors of PIM used for Multicast Routing in the Data Center: PIM-SM [RFC4601], PIM-SSM [RFC4607], and PIM-BIDIR [RFC5015]. SSM provides the most efficient forwarding between sources and receivers and is most suitable for one to many types of multicast applications. State is built for each S,G channel therefore the more sources and groups there are, the more state there is in the network. BIDIR is the most efficient shared tree solution as one tree is built for all S,G's, therefore saving state. But it is not the most efficient in forwarding path between sources and receivers. SSM and BIDIR are optimizations of PIM-SM. PIM-SM is still the most widely deployed multicast routing protocol. PIM-SM can also be the most complex. PIM-SM relies upon a RP (Rendezvous Point) to set up the multicast tree and then will either switch to the SPT (shortest path tree), similar to SSM, or stay on the shared tree (similar to BIDIR). For massive amounts of hosts sending (and receiving) multicast, the shared tree (particularly with PIM-BIDIR) provides the best potential scaling since no matter how many multicast sources exist within a VLAN, the tree number stays the same. IGMP snooping, IGMP proxy, and PIM-BIDIR have the potential to scale to the huge scaling numbers required in a data center.

#### 5. Challenges of using multicast in the Data Center

When IGMP/MLD Snooping is not implemented, ethernet switches will flood multicast frames out of all switch-ports, which turns the traffic into something more like a broadcast.

VRRP uses multicast heartbeat to communicate between routers. The communication between the host and the default gateway is unicast. The multicast heartbeat can be very chatty when there are thousands of VRRP pairs with sub-second heartbeat calls back and forth.

Link-local multicast should scale well within one IP subnet particularly with a large layer3 domain extending down to the access or aggregation switches. But if multicast traverses beyond one IP subnet, which is necessary for an overlay like VXLAN, you could potentially have scaling concerns. If using a VXLAN overlay, it is

necessary to map the L2 multicast in the overlay to L3 multicast in the underlay or do head end replication in the overlay and receive duplicate frames on the first link from the router to the core switch. The solution could be to run potentially thousands of PIM messages to generate/maintain the required multicast state in the IP underlay. The behavior of the upper layer, with respect to broadcast/multicast, affects the choice of head end (\*,G) or (S,G) replication in the underlay, which affects the opex and capex of the entire solution. A VXLAN, with thousands of logical groups, maps to head end replication in the hypervisor or to IGMP from the hypervisor and then PIM between the TOR and CORE 'switches' and the gateway router.

Requiring IP multicast (especially PIM BIDIR) from the network can prove challenging for data center operators especially at the kind of scale that the VXLAN/NVGRE proposals require. This is also true when the L2 topological domain is large and extended all the way to the L3 core. In data centers with highly virtualized servers, even small L2 domains may spread across many server racks (i.e. multiple switches and router ports).

It's not uncommon for there to be 10-20 VMs per server in a virtualized environment. One vendor reported a customer requesting a scale to 400VM's per server. For multicast to be a viable solution in this environment, the network needs to be able to scale to these numbers when these VMs are sending/receiving multicast.

A lot of switching/routing hardware has problems with IP Multicast, particularly with regards to hardware support of PIM-BIDIR.

Sending L2 multicast over a campus or data center backbone, in any sort of significant way, is a new challenge enabled for the first time by overlays. There are interesting challenges when pushing large amounts of multicast traffic through a network, and have thus far been dealt with using purpose-built networks. While the overlay proposals have been careful not to impose new protocol requirements, they have not addressed the issues of performance and scalability, nor the large-scale availability of these protocols.

There is an unnecessary multicast stream flooding problem in the link layer switches between the multicast source and the PIM First Hop Router (FHR). The IGMP-Snooping Switch will forward multicast streams to router ports, and the PIM FHR must receive all multicast streams even if there is no request from receiver. This often leads to waste of switch cache and link bandwidth when the multicast streams are not actually required. [I-D.pim-umf-problem-statement] details the problem and defines design goals for a generic mechanism to restrain the unnecessary multicast stream flooding.

## 6. Layer 3 / Layer 2 Topological Variations

As discussed in [I-D.armd-problem-statement], there are a variety of topological data center variations including L3 to Access Switches, L3 to Aggregation Switches, and L3 in the Core only. Further analysis is needed in order to understand how these variations affect IP Multicast scalability

## 7. Address Resolution

### 7.1. Solicited-node Multicast Addresses for IPv6 address resolution

Solicited-node Multicast Addresses are used with IPv6 Neighbor Discovery to provide the same function as the Address Resolution Protocol (ARP) in IPv4. ARP uses broadcasts, to send an ARP Requests, which are received by all end hosts on the local link. Only the host being queried responds. However, the other hosts still have to process and discard the request. With IPv6, a host is required to join a Solicited-Node multicast group for each of its configured unicast or anycast addresses. Because a Solicited-node Multicast Address is a function of the last 24-bits of an IPv6 unicast or anycast address, the number of hosts that are subscribed to each Solicited-node Multicast Address would typically be one (there could be more because the mapping function is not a 1:1 mapping). Compared to ARP in IPv4, a host should not need to be interrupted as often to service Neighbor Solicitation requests.

### 7.2. Direct Mapping for Multicast address resolution

With IPv4 unicast address resolution, the translation of an IP address to a MAC address is done dynamically by ARP. With multicast address resolution, the mapping from a multicast IP address to a multicast MAC address is derived from direct mapping. In IPv4, the mapping is done by assigning the low-order 23 bits of the multicast IP address to fill the low-order 23 bits of the multicast MAC address. When a host joins an IP multicast group, it instructs the data link layer to receive frames that match the MAC address that corresponds to the IP address of the multicast group. The data link layer filters the frames and passes frames with matching destination addresses to the IP module. Since the mapping from multicast IP address to a MAC address ignores 5 bits of the IP address, groups of 32 multicast IP addresses are mapped to the same MAC address. As a result a multicast MAC address cannot be uniquely mapped to a multicast IPv4 address. Planning is required within an organization to select IPv4 groups that are far enough away from each other as to not end up with the same L2 address used. Any multicast address in the [224-239].0.0.x and [224-239].128.0.x ranges should not be

considered. When sending IPv6 multicast packets on an Ethernet link, the corresponding destination MAC address is a direct mapping of the last 32 bits of the 128 bit IPv6 multicast address into the 48 bit MAC address. It is possible for more than one IPv6 Multicast address to map to the same 48 bit MAC address.

## 8. Acknowledgements

The authors would like to thank the many individuals who contributed opinions on the ARMD wg mailing list about this topic: Linda Dunbar, Anoop Ghanwani, Peter Ashwoodsmith, David Allan, Aldrin Isaac, Igor Gashinsky, Michael Smith, Patrick Frejborg, Joel Jaeggli and Thomas Narten.

## 9. IANA Considerations

This memo includes no request to IANA.

## 10. Security Considerations

No security considerations at this time.

## 11. Informative References

- [I-D.armd-problem-statement]  
Narten, T., Karir, M., and I. Foo,  
"draft-ietf-armd-problem-statement", February 2012.
- [I-D.pim-umf-problem-statement]  
Zhou, D., Deng, H., Shi, Y., Liu, H., and I. Bhattacharya,  
"draft-dizhou-pim-umf-problem-statement", October 2010.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas,  
"Protocol Independent Multicast - Sparse Mode (PIM-SM):  
Protocol Specification (Revised)", RFC 4601, August 2006.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for  
IP", RFC 4607, August 2006.
- [RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano,  
"Bidirectional Protocol Independent Multicast (BIDIR-  
PIM)", RFC 5015, October 2007.

Authors' Addresses

Mike McBride  
Huawei Technologies  
2330 Central Expressway  
Santa Clara, CA 95050  
USA

Email: michael.mcbride@huawei.com

Helen Lui  
Huawei Technologies  
Building Q14, No. 156, Beiqing Rd.  
Beijing, 100095  
China

Email: helen.liu@huawei.com

MBONED Working Group  
Internet Draft  
Intended status: BCP  
Expires: January 9, 2013

Percy S. Tarapore  
Robert Sayko  
AT&T  
Ram Krishnan  
Brocade  
July 9, 2012

Multicast Considerations in Support of CDN-I  
draft-tarapore-mboned-multicast-cdni-00.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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/lid-abstracts.txt>

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

This Internet-Draft will expire on January 9, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Abstract

This document examines the current capabilities of multicast to support content distribution in an environment involving multiple Service Providers joining together to form a Content Distribution Network Interconnection (CDN-I) Federation.

## Table of Contents

1. Introduction.....	2
2. CDN Nomenclature.....	3
3. Multicast Use Cases for a CDN-I Federation.....	4
3.1. Native Multicast Use Case.....	5
3.2. Automatic Multicast Tunneling Use Cases.....	5
3.2.1. AMT Interconnection Between P-CDN and S-CDN.....	6
3.2.2. AMT Tunnel Connecting S-CDN and EU.....	7
3.2.3. AMT Tunnel Connecting EU to P-CDN Through Non-Multicast S-CDN.....	7
4. Content Types Suitable for Multicast-based CDN.....	8
4.1. Live Content.....	8
4.2. "Delayed-Play" Download.....	8
4.3. "Instant-Play" Download.....	8
5. Evaluation of Native Multicast for CDN.....	9
6. Evaluation of AMT for CDN.....	9
7. Security Considerations.....	10
8. IANA Considerations.....	10
TBD.....	10
9. Conclusions.....	10
10. References.....	11
10.1. Normative References.....	11
10.2. Informative References.....	11
11. Acknowledgments.....	11

## 1. Introduction

Content Providers (CP) are experiencing significant growth in demand for all types of internet-based content. A single "over-the-top" CP would require significant resources to deliver content that could be requested from anywhere in the world. Service Providers (SP) are taking advantage of this situation by forming Content Distribution Network (CDN) Federations for the purpose of distributing content on behalf of Content Providers (CP). There are several advantages to such CDN Federations:



- o CPs can simply contract with one or more SPs in a CDN Federation for delivery of their content. This enables CPs to concentrate on their main objective - creation of content.
- o SPs can expand their geographic reach via distribution agreements with Federation members without developing costly resources outside their local territories.

Multicast-based delivery mechanisms are a natural fit for content distribution in the proposed CDN Federations. The scope of this document is strictly focused on the interactions between CDN Federation members to support multicast-based content distribution. The purpose of this document is the detailed examination of applicable multicast techniques and the identification of detailed data/metadata/parameters that will have to be exchanged by CDN Federation members to enable multicast-based content distribution.

## 2. CDN Nomenclature

Terminology utilized to describe end-to-end user requests is described as follows.

There are many entities involved in distribution of the content from the CP all the way to the End User (EU). Figure 1 is a diagram depicting the basic logical relationships among the various roles involved in content delivery. Besides the CP and EU, the two remaining major entities are the two members of a CDN Federation - the Primary CDN Provider (P-CDN) and the Supporting CDN Provider (S-CDN) [A-0200003]. The relationships between these entities are as follows - see Figure 1.

1. The Content Provider owns the content and specifies conditions of delivery and use. The End User interacts with the CP (link 1 in the figure) for authentication and authorization, and to reach an agreement to obtain specific content (content selection, content purchase, acknowledgement of conditions of use). The CP has the legal right to distribute content and specify conditions for distribution.
2. The CP has an agreement and interacts with the P-CDN for deploying content (link 2).
3. The P-CDN in turn has an agreement with an S-CDN for deploying and distributing content (link 3).

4. The End User is attached to S-CDN for access and obtains the content from the S-CDN (link 4). The End User also interacts with the CP (link 1) as indicated above.

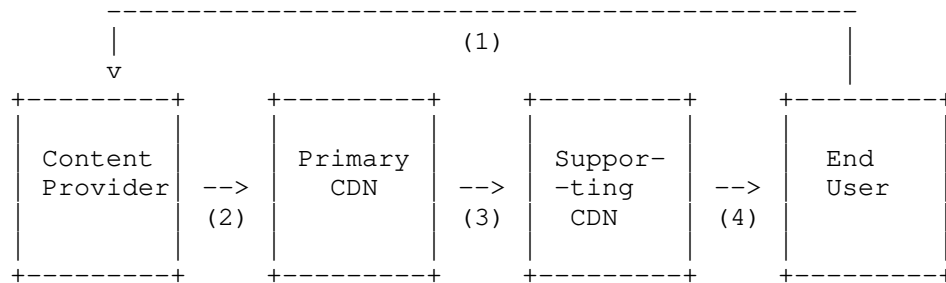


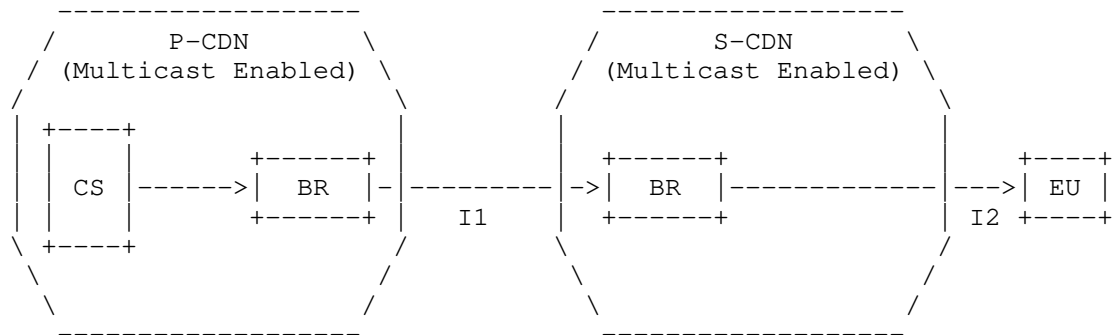
Figure 1 - Relationships in a CDN Federation

Note that all SPs in the CDN Federation can play the role of P-CDN (active relationship with a CP) as well as an S-CDN (attach EUs and distribute content from P-CDN to EUs).

### 3. Multicast Use Cases for a CDN-I Federation

Use cases involving multicast methods for distributing content in a CDN Federation have been described in [A-0200004].

### 3.1. Native Multicast Use Case



CS = Content Server

BR = Border Router

I1 = P-CDN and S-CDN Multicast Interconnection (MBGP or BGMP)

I2 = S-CDN and EU Multicast Connection

Figure 1 - Content Distribution via End to End Native Multicast

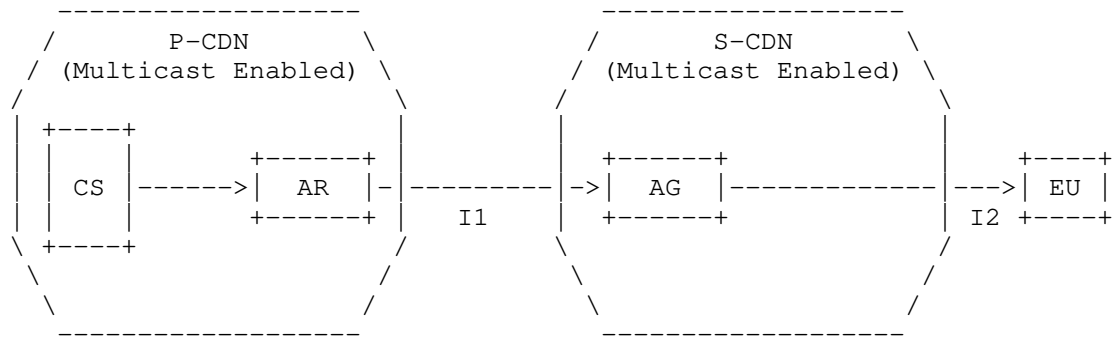
This case assumes that both CDN Providers as well as the interconnection between them and the connection between the EU and S-CDN are all multicast enabled.

A variation of this "pure" Native Multicast case is when the interconnection I1 between the CDNs is multicast enabled via a Generic Routing Encapsulation Tunnel (GRE) [RFC2784] instead of utilizing MBGP or BGMP protocols.

### 3.2. Automatic Multicast Tunneling Use Cases

In reality, the initial introduction of multicast may not be fully multicast enabled resulting in "Multicast Islands" requiring Automatic Multicast Tunnels (AMT) for enabling multicast connections between them [IETF-ID-AMT].

### 3.2.1. AMT Interconnection Between P-CDN and S-CDN

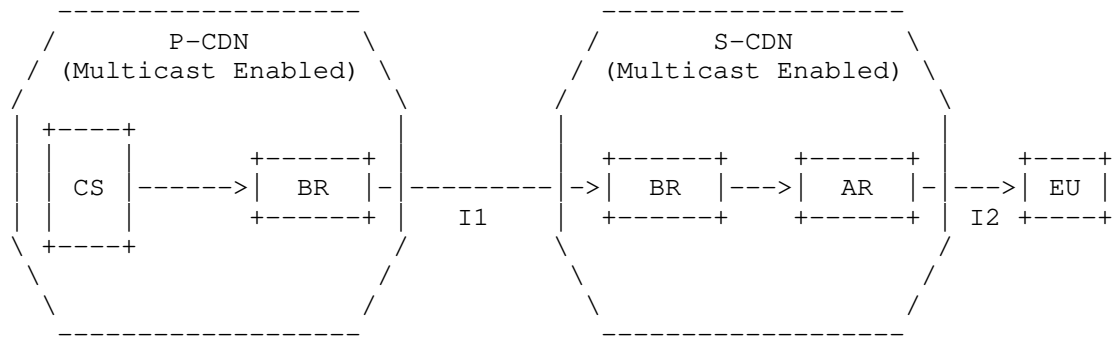


AR = AMT Relay  
AG = AMT Gateway  
I1 = AMT Interconnection between P-CDN and S-CDN  
I2 = S-CDN and EU Multicast Connection

Figure 3 - AMT Interconnection between P-CDN and S-CDN

This configuration assumes both CDN Providers are multicast enabled. Only the interconnection between them is not multicast enabled and hence, an AMT tunnel is established between them as shown in Figure 3.

### 3.2.2. AMT Tunnel Connecting S-CDN and EU



CS = Content Server

BR = Border Router

AR = AMT Relay

I1 = P-CDN and S-CDN Multicast Interconnection (MBGP or BGMP)

I2 = AMT Connection between S-CDN and EU

Figure 4 - AMT Tunnel Connecting S-CDN and EU

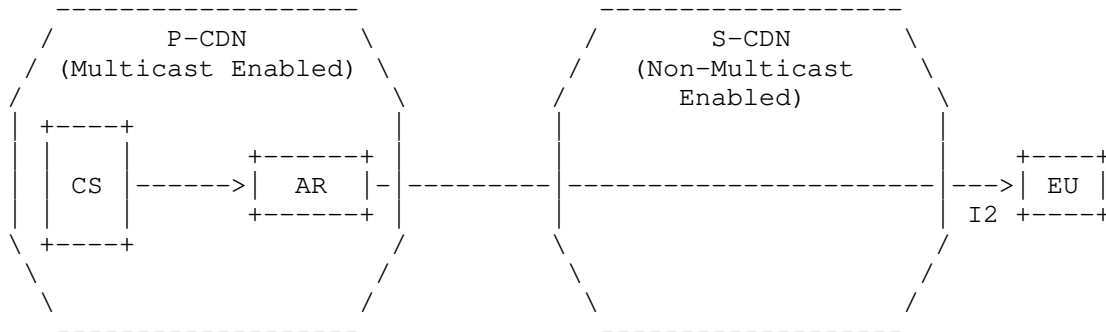
This case involves EU devices that are not multicast enabled. Hence an AMT Tunnel is established between the S-CDN AMT Relay and the EU device. This implies one tunnel per EU - potentially several AMT tunnels may need to be setup.

Note that there could be configurations involving both situations described in 3.2.1 and 3.2.2.

### 3.2.3. AMT Tunnel Connecting EU to P-CDN Through Non-Multicast S-CDN

This Use Case assumes that EU attached to the non-multicast enabled S-CDN has a device populated with a client that establishes an AMT tunnel to the AMT Relay in the P-CDN.

This configuration is needed when the S-CDN is not multicast-enabled. This is the most "extreme" AMT case as the length of the tunnels as well as the number of tunnels can be large.



CS = Content Source

AR = AMT Relay

I2 = AMT Tunnel Connecting EU to P-CDN Relay through Non-Multicast Enabled S-CDN.

Figure 5 - AMT Tunnel Connecting P-CDN AMT Relay and EU

#### 4. Content Types Suitable for Multicast-based CDN

This section highlights applications and content types that are suitable for multicast-based delivery in a CDN Federation. Any unique aspects of specific applications/content types that require special attention are duly noted.

##### 4.1. Live Content

Live events and presentations such as live radio and sporting events are examples. Delivery is via simple multicast means.

Additional detail TBD

##### 4.2. "Delayed-Play" Download

This includes download of movies and software updates. Delivery is via repeated multicasting of content.

Additional detail TBD

##### 4.3. "Instant-Play" Download

This includes Video-on-Demand (VoD) and on-demand streaming. Delivery is via simultaneous repeated multicast of content segments.

Additional detail TBD

## 5. Evaluation of Native Multicast for CDN

Use Case 3.1 describes Native Multicast configurations. This is the "simplest" multicast case in that a single standard set of protocols supports end-to-end content delivery from the CP to EU via two or more fully multicast-enabled CDN Providers. It also provides for efficient use of bandwidth and resources.

Use Case 2a does deploy an AMT Tunnel for interconnecting two CDN Providers; the rest of the configuration is Native Multicast - this assumes that the EU devices are also multicast-enabled.

Thus existing Native Multicast capabilities need to be examined to determine their ability to fully support content distribution in a CDN Federation. A list of issues requiring examination is as follows:

- o Delivery - Identification and communication of {Source, Group} information and DNS information for provisioning across CDNs. Details to be provided.
- o Routing/Peering - Identification and acknowledgement of external IP addresses particularly when utilizing a GRE Tunnel for interconnecting CDNs. Details to be provided.
- o Back-Office Functions - Identification of appropriate data/metadata collected by Native Multicast to support usage of content for billing, settlements, logging, etc. Details to be provided.
- o Security - Determine ability of Native Multicast to deal with security risks such as bot attacks, denial of service, etc. Details to be provided.
- o Others - To Be Determined

## 6. Evaluation of AMT for CDN

Use Cases 3.2.1, 3.2.2, and 3.2.3 describe the possible configurations involving AMT Tunnels. The likeliest scenario is a combination of Use Cases 3.2.1 and 3.2.2.

Use Case 3.2.3 becomes problematic if the length of the AMT Tunnels connecting the EUs to the P-CDN AMT Gateway become prohibitively long.

In all cases, there may be a concern if the total number of AMT Tunnels required is large. The list of issues that need to be examined for the AMT scenarios to support content distribution in a CDN Federation includes all identified issues in the Native Multicast case:

- o Delivery - Identification and communication of {Source, Group} information and DNS information for provisioning across CDNs. Details to be provided.
- o Routing/Peering - Identification and acknowledgement of external IP addresses when utilizing AMT Tunnels for interconnecting CDNs. Details to be provided.
- o Back-Office Functions - Identification of appropriate data/metadata collected via AMT to support usage of content for billing, settlements, logging, etc. Details to be provided.
- o Security - Determine ability of AMT to deal with security risks such as bot attacks, denial of service, etc. Details to be provided.
- o Others - To Be Determined

These have to be separately investigated for the AMT cases. In addition, there may be a need to examine the scope of additional resources in terms of bandwidth capacity and additional network elements particularly for Use Cases 3.2.2 and 3.2.3.

## 7. Security Considerations

TBD

## 8. IANA Considerations

TBD

## 9. Conclusions

TBD



## 10. References

### 10.1. Normative References

[RFC2784] D. Farinacci, T. Li, S. Hanks, D. Meyer, P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000

[IETF-ID-AMT] G. Bumgardner, "Automatic Multicast Tunneling", draft-ietf-mboned-auto-multicast-13, April 2012, Work in progress

### 10.2. Informative References

[A-0200003] P. Tarapore, "CDN Interconnection Use Case Specifications and High Level Requirements", ATIS Standard A-0200003, June 2011 (contact Nicole Butler at nbutler@atis.org using code IETF12 to receive a free copy before September 30, 2012)

[A-0200004] P. Tarapore and R. Sayko, "CDN Interconnection Use Cases and Requirements for Multicast-Based Content Distribution", ATIS Standard A-0200004, January 2012 (contact Nicole Butler at nbutler@atis.org using code IETF12 to receive a free copy before September 30, 2012)

## 11. Acknowledgments

Authors' Addresses

Percy S. Tarapore  
AT&T  
Phone: 1-732-420-4172  
Email: tarapore@att.com

Robert Sayko  
AT&T  
Phone: 1-732-420-3292  
Email: rs1983@att.com

Ram Krishnan  
Brocade  
Phone:  
Email: ramk@brocade.com



Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: January 17, 2013

Q. Sun  
China Telecom  
C. Zhou  
Huawei Technologies  
July 16, 2012

Multicast transition path optimization in IPv4 and IPv6 networks  
draft-zhou-mboned-multtrans-path-optimization-02

## Abstract

This document describes a mechanism to optimize the path between the multicast router and multicast source in both IPv4 and IPv6 networks. The basic idea is that when a multicast translation router has an IPv4 path and an IPv6 path to the same multicast data source, and both IPv4 and IPv6 joins are received, only one path is used. One path is pruned, instead of the same traffic flowing over both v4 and v6 paths. By adding a metric to the IPv4 path, the multicast translation router can determine which path to receive multicast data: IPv4 path, IPv6 path or both. Therefore, an optimization path will typically be chosen when an identical v4/v6 traffic flow exists.

## Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in .

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 17, 2013.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. Scenarios . . . . .	4
4. Solution Overview . . . . .	4
4.1. A general topology for IPv4 and IPv6 multicast networks . .	5
4.2. Parsing MTR to two virtual Routers . . . . .	6
4.3. Selecting interfaces to Source or RP . . . . .	7
4.4. Selecting a multicast data flow from upstream interface . .	7
4.5. Modifications to mulitcast Router . . . . .	7
5. Security Considerations . . . . .	8
6. Acknowledgments . . . . .	8
7. IANA Considerations . . . . .	8
8. Informative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

It is common to use multi-access LANs such as Ethernet for transmitting multicast data in networks. Section 3.6 of[RFC4601] describes Multi-Access Transit LANs.

The PIM Assert message could be used when there are two identical multicast data flows (IPv4 and IPv6). When duplicate data packets appear on the LAN from different routers, the routers notice this and then select a single forwarder. This selection is performed using PIM Assert messages, which solve the problem in favor of the upstream router that has (S,G) state; Or, if neither or both router has (S,G) state, then the problem is solved in favor of the router with the best metric to the RP for RP trees, or the best metric to the source via source-specific trees.

During IPv6 transition, it is common that there are many IPv4 networks and IPv6 networks that connected to each other, which means that multiple multicast translation routers(MTR) exist at the edge of a network. For robustness, reliability and load balance purpose, MTR function could be implemented in several nodes in the network. MTR can be the mAFTR (Multicast AFTR ) mentioned in [draft-ietf-softwire-dslite-multicast]. mAFTR can encapsulate IPv4 multicast data in IPv6 tunnel. MTR can also be the mXlate (Multicast Translator) as mentioned in [draft-lee-behave-v4v6-mcast-fwk]. mXlate can translate IPv4 multicast data to IPv6 multicast data.

As a result, MTR (mXlate or mAFTR) will have more than one path to reach the RP or source S in IPv4 networks and IPv6 networks. Or in other words, they will have two upstream routers: one is IPv6 router, and the other is IPv4 router. MTR can reach the RP or source S by both paths. Since MTR can receive both IPv4 and IPv6 (\*,G) (or (S,G)) Join request, it needs to select a best path to RP or S in both IPv4 and IPv6 networks. When it receives the two identical multicast data flows via IPv4 and IPv6 interfaces, MTR needs to send Prune Message to the worse path interface. Figure 1 shows the scenario that MTR can reach source S through both IPv4 path and IPv6 path.

## 2. Terminology

This document makes use of the following terms:

mXlate: A multicast translator mentioned in [draft-lee-behave-v4v6-mcast-fwk].

mAFTR: A multicast Address Family Transition Router mentioned in [draft-ietf-softwire-dslite-multicast].

MTR: A multicast translation router, it can be mAFTR or mXlate.

PIM-SM: Protocol Independent Multicast-Sparse Mode

RP: Rendezvous Point

### 3. Scenarios

During the multicast transition from IPv4 to IPv6, there may be a router which receives IPv4 join (PIM or IGMP) on one interface, and an IPv6 join (PIM or MLD) on another interface (or it could even be the same interface). The router should support IPv4 PIM and IPv6 PIM that is translation capable. Assume these joins are for both IPv4 (S4,G4) and IPv6 (S6,G6), and that there are active sources for both, sending basically the same content. Either because there is a real source for both, or some upstream router is translating. The router could then simply send upstream joins for both of these, and forward the traffic as needed without translation.

However, if the router is aware that these two sources are really the same content, it could select to join just one of the streams, and translate as needed for the one downstream that wants a different protocol. In this case, there will be a tradeoff between bandwidth on the upstream links, and the cost of translation (both on this device, and perhaps the quality of the stream). When PIM Assert message is used to achieve this, the metrics for IPv4 and IPv6 should be comparable and all the PIM devices on the link should support PIM assert.

### 4. Solution Overview

This section gives a solution for the issues mentioned above.

## 4.1. A general topology for IPv4 and IPv6 multicast networks

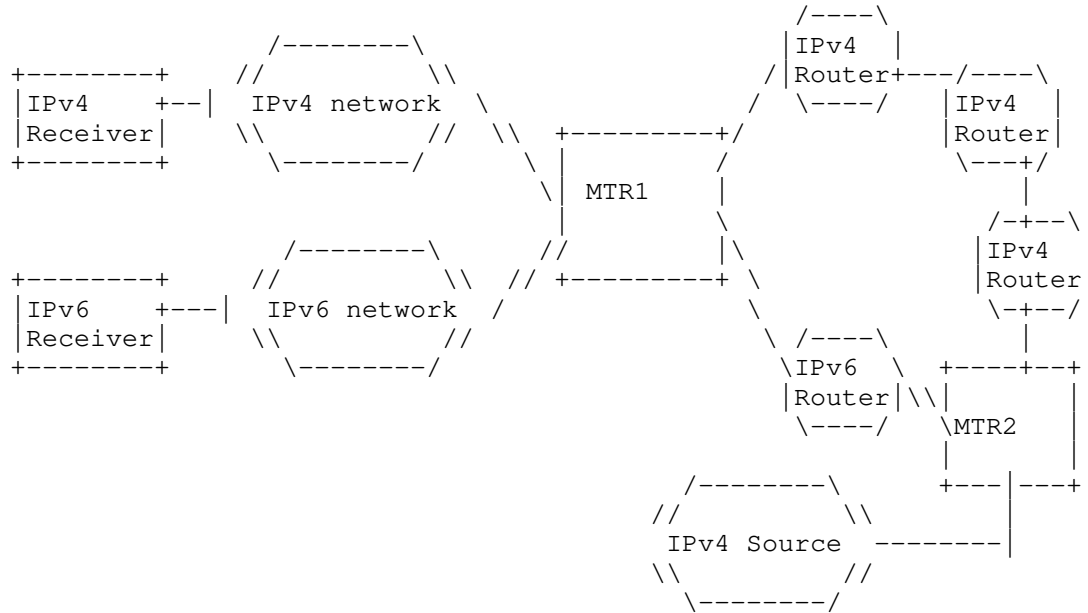
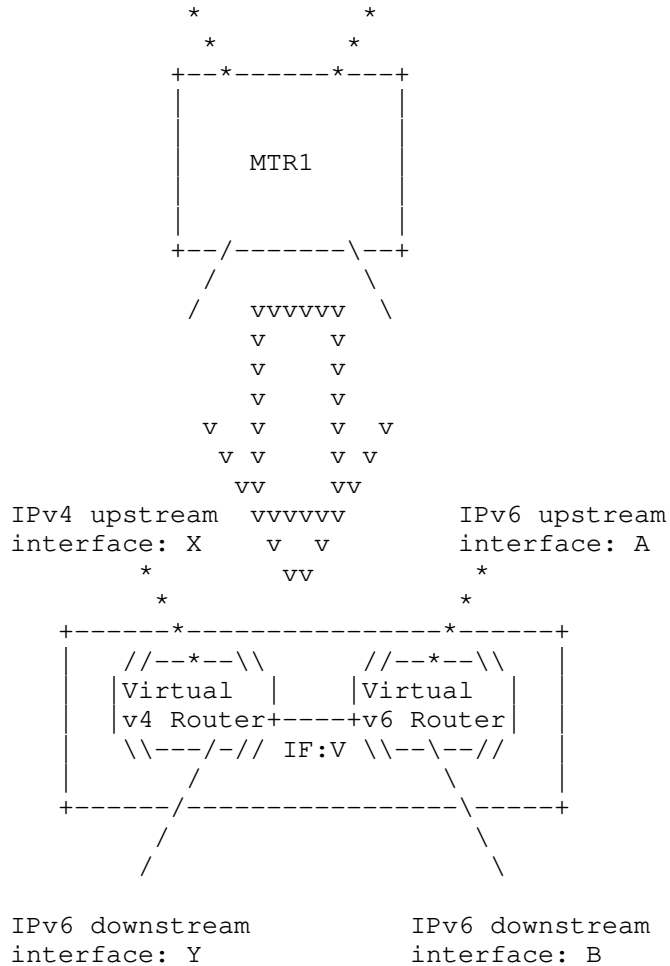


Figure 1: MTR can reach IPv4 Source through IPv4 path and IPv6 path

Figure 1 shows that MTR1 can access IPv4 Source through IPv4 path or IPv6 path. MTR1 has two upstream routers, one is IPv4 Router and the other is IPv6 Router. MTR1 receives IPv4 (\*,G) or (S,G) Join request from IPv4 network and IPv6 (\*,G) or (S,G) Join request from IPv6 network. MTR1 can send Join request to RP or source S from interface connected to IPv4 Router or from interface connected to IPv6 Router. MTR1 may also send Join request from both upstream interfaces. In this case, MTR1 need to select a best path to RP or S in both IPv4 and IPv6 networks. MTR1 sends Prune Message to the worse path, when it receives two identical multicast data flows in IPv4 and IPv6 upstream interface. MTR1 may receive two identical multicast data flows at the same time and stop interworking multicast data flow between IPv4 network and IPv6 network.



## 4.2. Parsing MTR to two virtual Routers



For simplification, we use two virtual Routers to replace MTR1 Router. Figure 2 shows that MTR1 can be taken as two virtual Routers. The one on the left is a Virtual IPv4 Router, the one on the right is a Virtual IPv6 Router. Virtual IPv4 Router has an IPv4 upstream interface X and an IPv4 downstream interface Y. Virtual IPv6 Router has an IPv6 upstream interface A and an IPv6 downstream interface B. The interface between two virtual Routers is V.

When MTR receives two multicast data flows (one from IPv4 interface and the other from IPv6 interface), it compares two flows according to [draft-ietf-mboned-64-multicast-address-format] to confirm whether they are identical data flows. If they are the same, select one or

two. When MTR Receives a IPv6 (S, G) or (\*, G)Join, virtual IPv6 Router selects an interface to send Join message. The interface can be IPv6 upstream interface A or IPv4 upstream interface X (via interface V).

#### 4.3. Selecting interfaces to Source or RP

The steps to select an interface to S or RP.

1. Set the Metric value  $m1$  for translation or encapsulation from IPv4 multicast to IPv6 multicast data.
2. From interface A connecting IPv6 Router, MTR can get the metric  $m2$  to reach S or RP by PIM assert message sent from IPv6 Router.
3. From interface X connecting IPv4 Router, MTR can get the metric  $m3$  to reach S or RP by PIM assert message from IPv4 Router.
4. When MTR receives a IPv6 PIM Join message, virtual IPv6 Router compares  $m2$  and  $m3+m1$ . If  $m2 > m3+m1$ , sending PIM Join message from IPv4 interface; If  $m2 < m3+m1$ , sending PIM Join message from IPv6 interface; If  $m2 = m3+m1$ , MTR can choose interface X or A to send PIM Join message.

#### 4.4. Selecting a multicast data flow from upstream interface

The steps for selecting a multicast data flow from upstream interface

1. MTR receives two identical multicast flows from IPv6 and IPv4 Router. The address formats of the two flows follows [draft-ietf-mboned-64-multicast-address-format].
2. If virtual IPv6 Router receives multicast data from interface V, it will compare  $m2$  and  $m3+m1$  (the value is from last section).
3. If  $m2 > m3+m1$ , MTR will send PIM Prune Messages to IPv6 interface A; If  $m2 < m3+m1$  MTR will send PIM Prune Messages to interface X via virtual interface V. MTR will not translate multicast data from IPv4 to IPv6 or encapsulate IPv4 multicast data in IPv6 packets. If  $m2 = m3+m1$ , MTR selects any interface to receive multicast data and sends PIM Prune Messages to the other interface.

#### 4.5. Modifications to multicast Router

The main modifications to the edge PIM-SM Router include:

Edge PIM-SM Router needs to check multicast data flow from IPv4 and IPv6 interfaces based on

[draft-ietf-mboned-64-multicast-address-format] to determine whether they are the same multicast data flow.

Edge PIM-SM Router sends PIM Assert messages via IPv4 and IPv6 interfaces with different Metric value.

Edge PIM-SM Router may stop translating/encapsulating IPv4 multicast flow to IPv6 multicast flow or send Prune Messages to stop receiving IPv6/IPv4 multicast flow.

## 5. Security Considerations

## 6. Acknowledgments

Thanks Ronald Bonica, Stig Venaas and Yiu Lee for their valuable comments.

## 7. IANA Considerations

## 8. Informative References

[RFC4601] IETF, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification (Revised)", Aug 2006, <<http://datatracker.ietf.org/doc/rfc4601/>>.

[draft-ietf-mboned-64-multicast-address-format]  
IETF, "IPv4-Embedded IPv6 Multicast Address Format", Feb 2012, <<http://datatracker.ietf.org/doc/draft-ietf-mboned-64-multicast-address-format/>>.

[draft-ietf-software-dslite-multicast]  
IETF, "Multicast Extensions to DS-Lite Technique in Broadband Deployments", Oct 2011, <<http://datatracker.ietf.org/doc/draft-ietf-software-dslite-multicast/>>.

[draft-lee-behave-v4v6-mcast-fwk]  
IETF, "IPv4/IPv6 Multicast Translation Framework", Feb 2011, <<http://tools.ietf.org/id/draft-lee-behave-v4v6-mcast-fwk-00.txt>>.

Authors' Addresses

Qiong Sun  
China Telecom  
Xizhimenneidajie Xicheng District  
Beijing, 100035  
China

Phone:  
Fax:  
Email: [sunqiong@ctbri.com.cn](mailto:sunqiong@ctbri.com.cn)  
URI:

Cathy Zhou  
Huawei Technologies  
Section F, R&D Building, Huawei Longgang Production Base  
Shenzhen, 518129  
China

Phone:  
Fax:  
Email: [cathy.zhou@huawei.com](mailto:cathy.zhou@huawei.com)  
URI:

