Network Working Group   G. Bumgardner
Internet-Draft
Intended status: Standards Track   April 24, 2014
Expires: October 26, 2014

                    Automatic Multicast Tunneling
                  draft-ietf-mboned-auto-multicast-17

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 October 26, 2014.

Table of Contents

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.

AMT is designed to be deployed at the border of networks possessing
native multicast capabilities where access and provisioning can be
managed by the AMT service provider.

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 (NMBA):
   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 state for an interface or system. See Section 3 in
    [RFC3376] for a detailed definition.

Reception State:
    The multicast subscription state of a pseudo, virtual or physical
    network interface. Often synonymous with group membership
    database.

Subscription:
    A group or state entry in a group membership database or reception
    state table. The presence of a subscription entry indicates
    membership in an IP multicast group.

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

```
Isolated Site |    Unicast Network   |  Native Multicast
              |       (Internet)     |
              |                      |
              |                      |
              |   Group Membership   |
   +-------+ =========================> +-------+ Multicast +------+
   |Gateway|  |                      |  | Relay |<----//----|Source|
   +-------+ <========================= +-------+           +------+
              |    Multicast Data    |
              |                      |
              |                      |
```
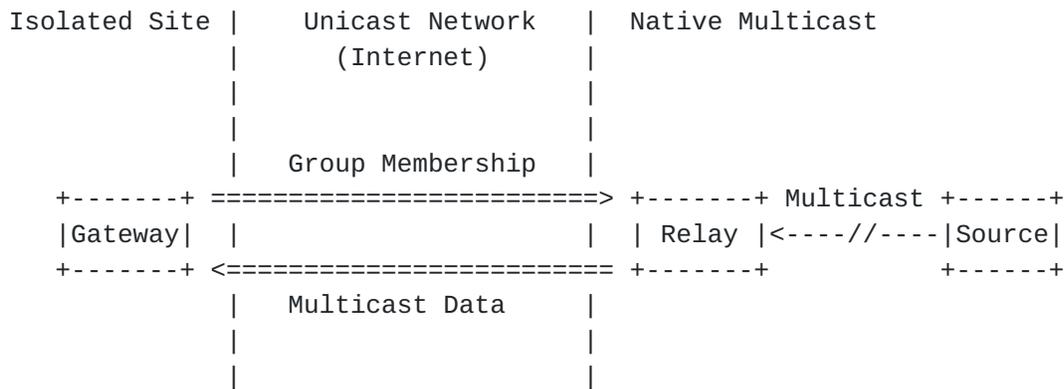
Figure 1: Basic AMT Architecture

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.

```
              Gateway                          Relay

           General _____            _____
 _____  Query |     |         |     | Query _____
|         |  |<------|     |         |     |  |<------|          |
| Host Mode |       | AMT |         | AMT |         |Router Mode|
| IGMP/MLD  |       |     |  UDP    |     |         | IGMP/MLD  |
|_____|------>|     |  |<----->|     |  |------>|_____|
           Report |     |         |     |  | Report
         Leave/Done |     |         |     |  | Leave/Done
                   |     |         |     |  |
  IP Multicast <------|     |         |     |  |<------ IP Multicast
                   |_____|         |_____|
```
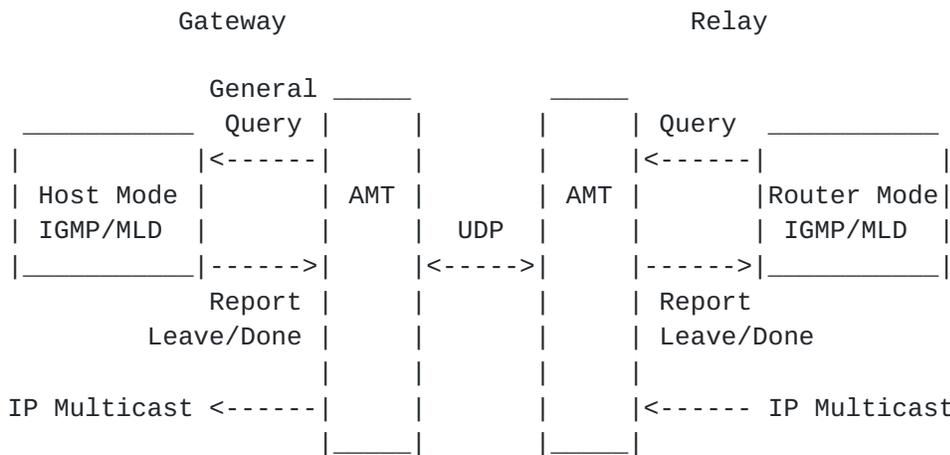
             Figure 2: Multicast Reception State Managed By IGMP/MLD

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 gateway
   functionality may be directly implemented in the host requesting the
   multicast service or within an application running on a host.

   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:

```
   join/leave ---+                    +----------+
              |                    |          |
              V       IGMPv3/MLDv2 |          |
         +---------+ General Query|          |   AMT
         |IGMP/MLD |<-------------|   AMT    | Messages +------+
         |Host Mode|              | Gateway  |<-------->|UDP/IP|
         |Protocol |------------->|Pseudo I/F|          +------+
         +---------+   IGMP/MLD   |          |             ^
                       Report    |          |             |
                       Leave/Done |          |             V
    IP Multicast <--------------------|          |          +---+
                                  +----------+          |I/F|
                                                        +---+
```
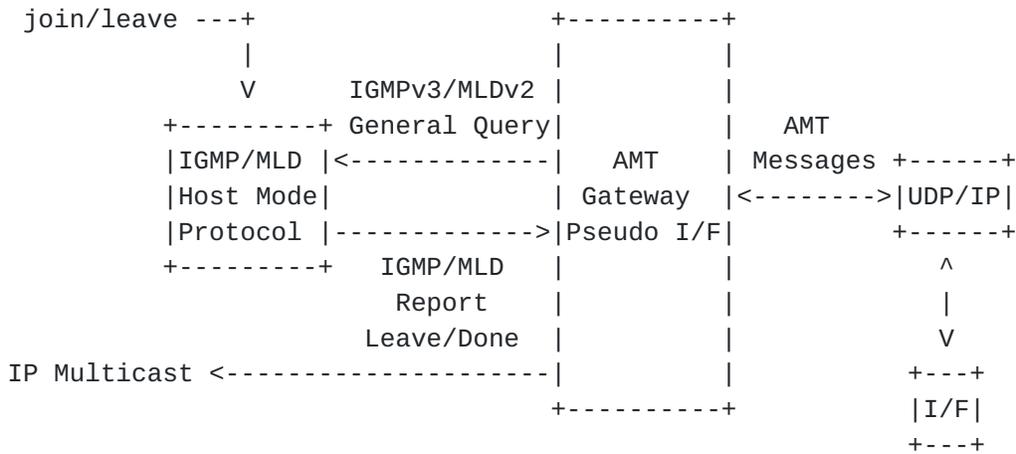
                Figure 3: AMT Gateway Pseudo-Interface

   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.

The following diagram illustrates how an existing host IP stack
implementation might be used to provide AMT gateway functionality to
a multicast application:

```
    +-----------------------------------------------------+
    |Host                                                 |
    |    _____       |
    |    |                                      |        |
    |    |    _____          |        |
    |    |    |                      |          |        |
    |    |    |                      v          |        |
    |    |    |      +-----------+  +--------------+      |
    |    |    |      |Application|  |  AMT Daemon  |      |
    |    |    |      +-----------+  +--------------+      |
    |    |    | join/leave |   ^ data        ^ AMT       |
    |    |    |            |   |              |          |
    |    |    |       +----|---|-------------|-+         |
    |    |    |       |  __|   |_____      | |        |
    |    |    |       | | |            |      | |        |
    |    |    |       | | |   Sockets  |      | |        |
    |    |    |       +-|------+-------+-|---|-+         |
    |    |    |       | | IGMP |  TCP  | |UDP| |         |
    |    |    |       +-|------+-------+-|---|-+         |
    |    |    |       | | ^        IP     | | |          |
    |    |    |       | | | |  _____|  | |         |
    |    |    |       | | | | |              | |         |
    |    |    |       +-|-|-|---------------|-+         |
    |    |    |       | | | |               |          |
    |    |    | IP(IGMP)| | |IP(UDP(data))   |IP(UDP(AMT))  |
    |    |    |       v | |               v          |
    |    |    |    +-----------+       +---+          |
    |    |    |    |Virtual I/F|       |I/F|          |
    |    |    |    +-----------+       +---+          |
    |    |    |        |   ^             ^            |
    |    |    | IP(IGMP)|   |IP(UDP(data))  |          |
    |    |    |_____|   |IP(IGMP)       |          |
    |    |            |               |          |
    |    |_____|               |          |
    |                                 |          |
    +---------------------------------|----------+
                                      v
                                 AMT Relay
```

Figure 4: Virtual Interface Implementation Example

In this example, the host IP stack uses a virtual network interface
to interact with a gateway pseudo-interface implementation.

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:

```
                                      +------------------------------+
                  +--------+          | Multicast Control Plane      |
                  |        |IGMP/MLD|                                |
                  |        | Query* | +------------+  +----------+ |
                  |        |<---//----|IGMPv3/MLDv2|  |Multicast | |
           AMT    |        |          | |Router Mode |->|Routing   |<->
    +------+ Messages | AMT    |----//--->|Protocol    | |Protocol  | |
    |UDP/IP|<-------->| Relay  |IGMP/MLD| +------------+  +----------+ |
    +------+          | Pseudo | Report |       |             |       |
       ^              | I/F    | Leave/ +------|--------------|-------+
       |              |        |  Done          |             |
       |              |        |                v             |
       V              |        | IP          +-----------+    |
     +---+            |        | Multicast |Multicast  |<------+
     |I/F|            |        |<---//-----|Forwarding |
     +---+            +--------+           |Plane      |<--- IP Multicast
                                           +-----------+
```
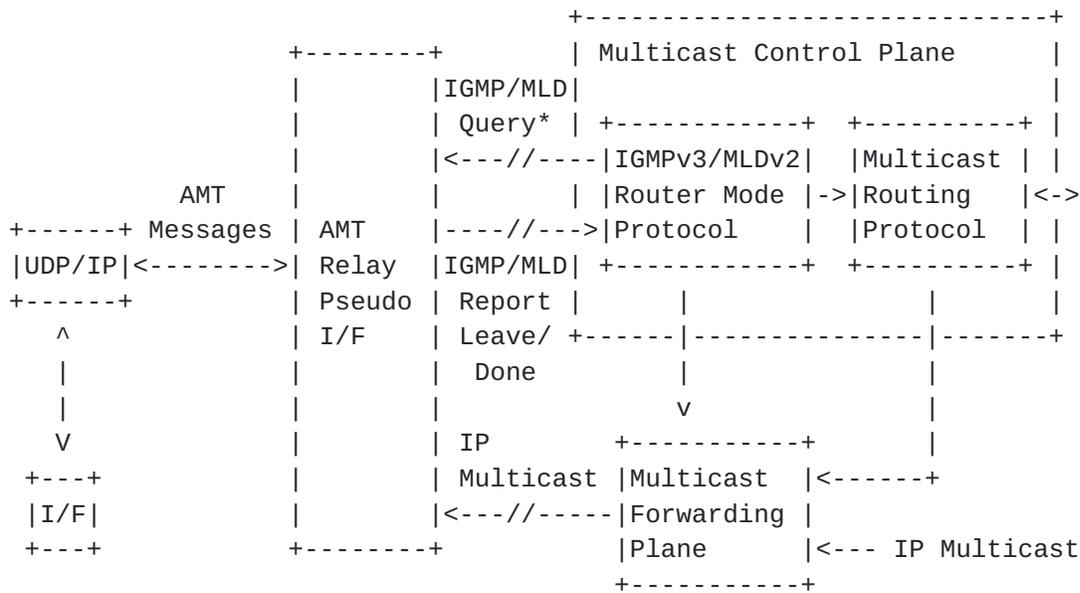
 * Queries, if generated, are consumed by the pseudo-interface.

       Figure 5: 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. Note that the
protocol mandates the use of IGMPv3 and MLDv2 for query messages.
The AMT protocol is primarily intended for use in SSM applications
and relies on several values provided by IGMPv3/MLDv2 to control
gateway behavior.

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.

4.1.4.2. Congestion Considerations

AMT relies on UDP to provide best-effort delivery of multicast data
to gateways. Neither AMT or the UDP protocol provide the congestion
control mechanisms required to regulate the flow of data messages
passing through a network. While congestion remediation might be
provided by multicast receiver applications via multicast group

selection or upstream reporting mechanisms, there are no means by
which to ensure such mechanisms are employed. To limit the possible
congestion across a network or wider Internet, AMT service providers
are expected to deploy AMT relays near the provider's network border
and its interface with edge routers. The provider must limit relay
address advertisements to those edges to prevent distant gateways
from being able to access a relay and potentially generate flows that
consume or exceed the capacity of intervening links.

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

IANA has assigned an address prefix for use in advertising and
discovering publicly accessible relays.

A relay discovery address is constructed from the 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 address prefix (e.g. via
BGP [RFC4271]) 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

```
                    Gateway                Relay
                    -------                -----
                       :                      :
                       |                      |
                   [1] |Relay Discovery       |
                       |------------------->|
                       |                      |
                       | Relay Advertisement| [2]
                       |<------------------|
                   [3] |                      |
                       :                      :
```
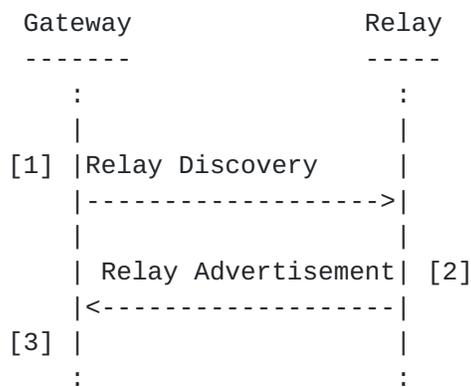
                 Figure 6: AMT Relay Discovery Sequence

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.

```
            IGMP/MLD              Pseudo-I/F              Relay
            --------              ----------              -----
               :                      :                    :
               |                      |       Request      |
               |                     1|------------------->|
               |                      |  Membership Query  |2
   Query       |                      |       Q(0,{})      |
   Timer       |        Start        3|<-------------------|
    (QT)<-----------------------------|                    |
               |        Q(0,{})       |                    |
               |<-------------------- |                    |
             4 |         R({})        |  Membership Update |
               |-------------------->|5        R({})       |
               |                      |===================>|6a
   Join(S,G) :                        :                    :
   ()-------->|7 R({G:ALLOW({S})})   |  Membership Update  |
               |-------------------->|8  R({G:ALLOW({S})}) |
               |                      |===================>|9a  Join(S,G)
               |                      |                    |---------->()
               :                      :                    :
               |     -----------|------------------|-----------
               |    |           |                  |           |
               |    |           |   Multicast Data |  IP(S,G)  |
               |    |           |        IP(S,G)  10|<--------() |
               |    |  IP(S,G) 11|<==================|           |
               |    | ()<--------|                  |           |
               |    |           |                  |           |
               :     -----------:------------------:-----------
               |     Expired    |                  |
    (QT)-------------------------->|12     Request    |
               |                     1|------------------->|
               |                      |  Membership Query  |2
               |                      |       Q(0,{})      |
               |        Start        3|<-------------------|
    (QT)<-----------------------------|                    |
               |        Q(0,{})       |                    |
               |<-------------------- |                    |
             4 | R({G:INCLUDE({S})}) |  Membership Update  |
               |-------------------->|5 R({G:INCLUDE({S})})|
               |                      |===================>|6b
  Leave(S,G) :                        :                    :
   ()-------->|7 R({G:BLOCK({S})})   |  Membership Update  |
               |-------------------->|8  R({G:BLOCK({S})}) |
               |                      |===================>|9b Prune(S,G)
               |                      |                    |---------->()
               :                      :                    :
```

            Figure 7: 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
    message source IP address, source UDP port, request nonce and a
    private secret. 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 Querier's Query Interval Code (QQIC) field in the
    IGMPv3 or MLDv2 general query. The QQIC field is defined in
    Section 4.1.7 of [RFC3376] and Section 5.1.9 of [RFC3810]).

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 the message source IP address, source UDP port,
    request nonce and a private secret. 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). The QRV
    field is defined in Section 4.1.6 in [RFC3376] and Section 5.1.8
    in [RFC3810].

9. When the relay receives a Membership Update message, it again
    computes a MAC from the message source IP address, source UDP
    port, request nonce and a private secret. 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 (See
Section 4.2.1.3).

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. Gateway support for the Teardown message is optional -
   gateways are not required to send them and may instead rely on group
   membership to expire on the relay.
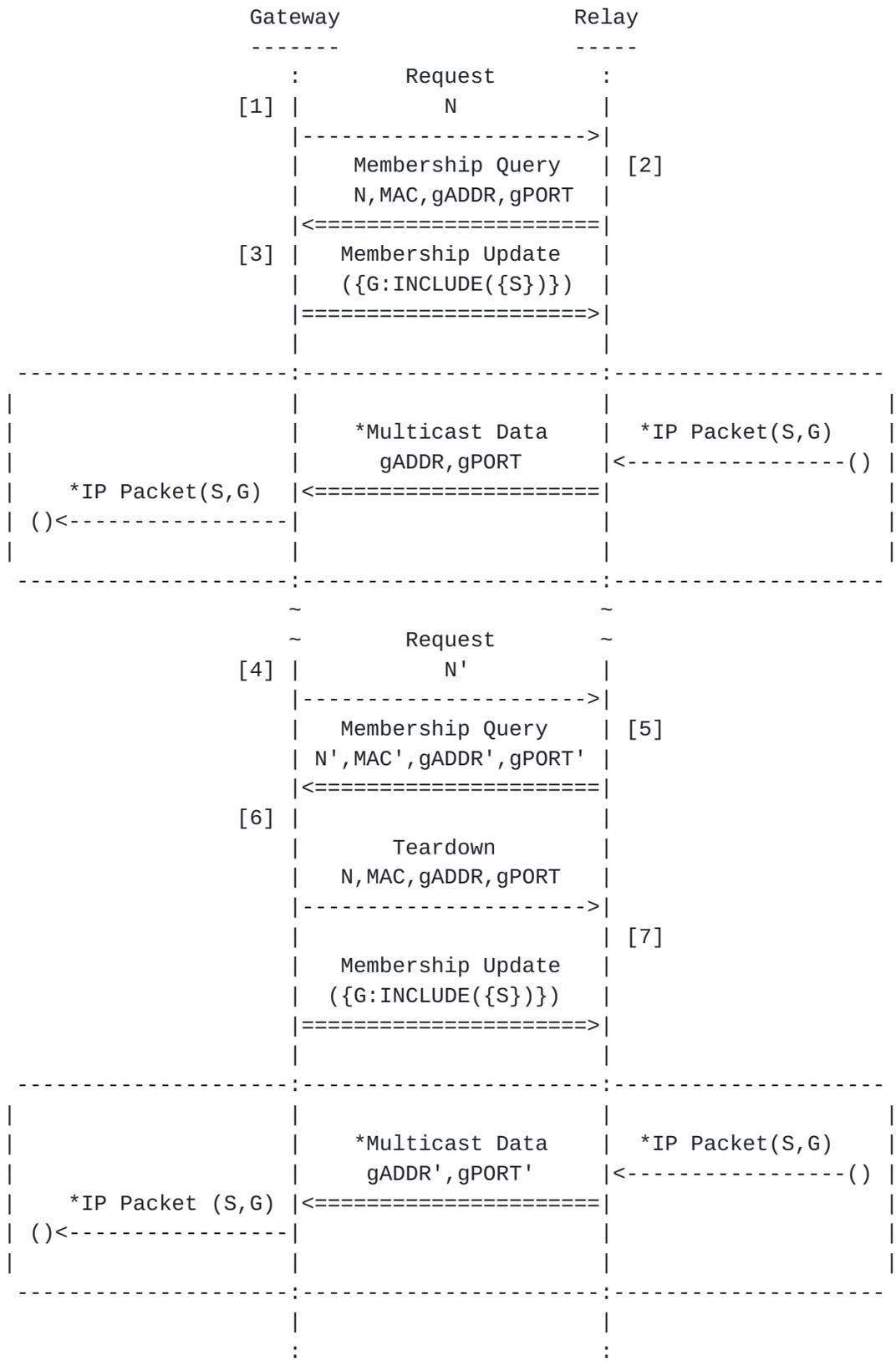
```
                      Gateway                Relay
                      -------                -----
                         :         Request      :
                   [1] |              N          |
                       |------------------------>|
                       |     Membership Query    | [2]
                       |     N,MAC,gADDR,gPORT    |
                       |<========================|
                   [3] |     Membership Update   |
                       |    ({G:INCLUDE({S})})    |
                       |========================>|
                       |                         |
        -------------------:------------------------:-------------------
        |                  |                        |                  |
        |                  |     *Multicast Data    | *IP Packet(S,G)   |
        |                  |       gADDR,gPORT      |<---------------() |
        |    *IP Packet(S,G) |<======================|                  |
        | ()<---------------|                        |                  |
        |                  |                        |                  |
        -------------------:------------------------:-------------------
                         ~                          ~
                         ~         Request          ~
                   [4] |             N'          |
                       |------------------------>|
                       |     Membership Query    | [5]
                       |  N',MAC',gADDR',gPORT'   |
                       |<========================|
                   [6] |                         |
                       |         Teardown        |
                       |     N,MAC,gADDR,gPORT    |
                       |------------------------>|
                       |                         | [7]
                       |     Membership Update   |
                       |    ({G:INCLUDE({S})})    |
                       |========================>|
                       |                         |
        -------------------:------------------------:-------------------
        |                  |                        |                  |
        |                  |     *Multicast Data    | *IP Packet(S,G)   |
        |                  |      gADDR',gPORT'     |<---------------() |
        |    *IP Packet (S,G) |<======================|                  |
        | ()<---------------|                        |                  |
        |                  |                        |                  |
        -------------------:------------------------:-------------------
                       |                         |
                       :                         :


            Figure 8: 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 (See Section 4.1.6 in [RFC3376] and
    Section 5.1.8 in [RFC3810]). 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 the message source IP address, source UDP port, request
    nonce and a private secret. 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.

```
     Membership Update                 Membership Update
        src: iADDR:iPORT                  src: eADDR:ePORT
        dst: rADDR:rPORT                  dst: rADDR:rPORT
                        +---------+
                        |  NAT    |
    +---------+         +-----------+       +---------+
    |         |---------->|         |--------->|         |
    | Gateway |         | Mapping   |        | Relay   |
    |         |<----------|         |<---------|         |
    +---------+         +-----------+       +---------+
                        |         |
                        +---------+
        Multicast Data                    Multicast Data
        src: rADDR:rPORT                  src: rADDR:rPORT
        dst: iADDR:iPORT                  dst: eADDR:ePORT
```
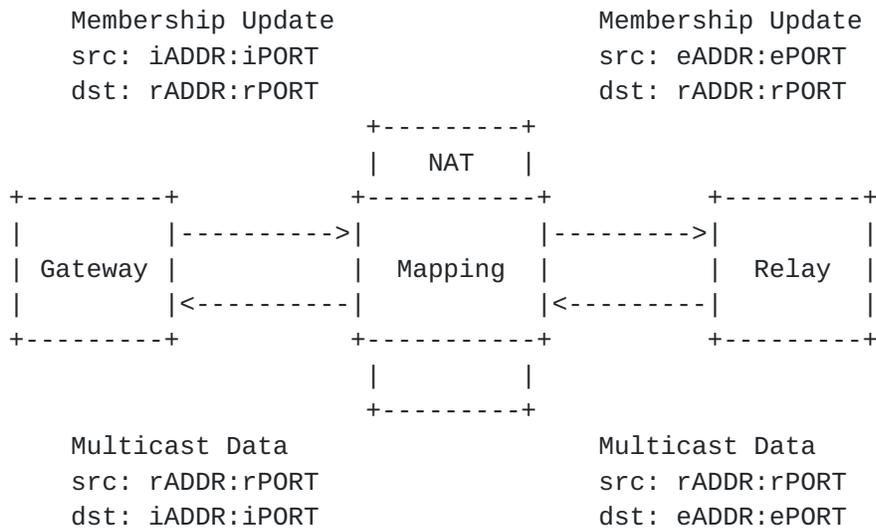
                Figure 9: Network Address Translation in AMT

   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**

```
            Gateway                                Relay

        IP:IGMP                                   IP:IGMP
            |      AMT:IP:IGMP          AMT:IP:IGMP   |
            |           |                   |         |
            |           |  IP:UDP:AMT:IP:IGMP   |     |
   _____  |   ___  |  _____  |  _____  |   ___  |  _____
  |IGMP|IP|  v  |AMT|  v  |UDP|IP|  v  |IP|UDP|  v  |AMT|  v  |IP|IGMP|
  |    | | |    | | |     | | | |     | | | |     | | |    | | |     |
  |    |<-------------------------------------------------------->|     |
  |___| |  |   |  | |     | | | |     | | | |     | | |    | | |___|
  |     |   |<--------------------------------------------------|     |
  |_____| ^ |___| ^ |___|__| ^ |__|___| ^ |___| ^ |_____|
            |         |             |             |         |
           IP       AMT:IP     IP:UDP:AMT:IP    AMT:IP       IP
```
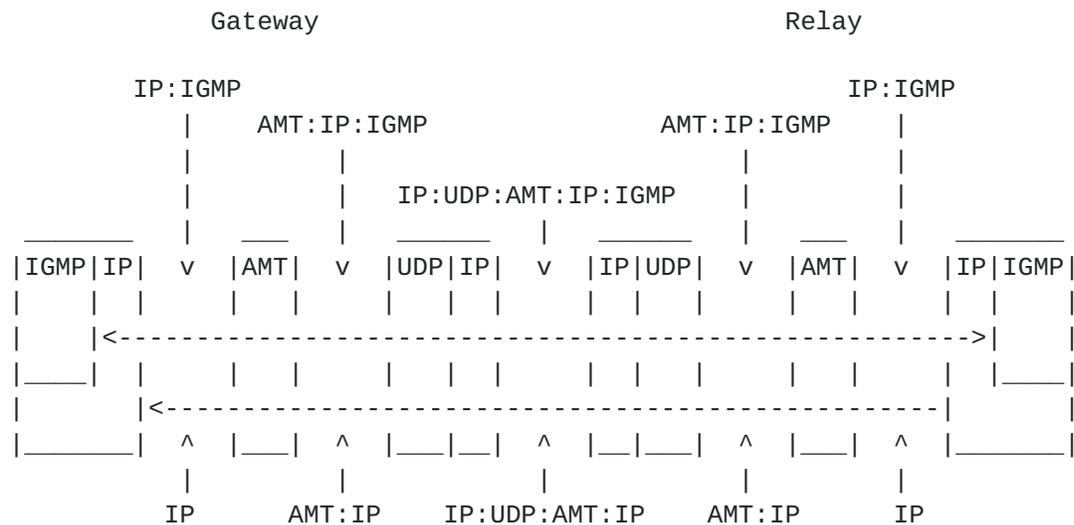
                    Figure 10: AMT Encapsulation

   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 [RFC6936]. A
   recommended solution is described in [RFC6935].

4.2.2.4. UDP Fragmentation

   Naive encapsulation of a multicast IP datagrams within an AMT data
   messages may produce UDP datagrams that might require fragmentation
   if their size exceeds the MTU of network path between the relay and a
   gateway. Many multicast applications, especially those related to
   media streaming, are designed to deliver independent data samples in
   separate packets, without fragmentation, to ensure some number of
   complete samples can be delivered even in the presence of packet
   loss. To prevent or reduce undesirable fragmentation, the AMT
   protocol describes specific procedures for handling multicast
   datagrams whose encapsulation might exceed the path MTU. These
   procedures are described in Section 5.3.3.6.

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 (See
      Section 7.2).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  V=0  |Type=1 |     Reserved                                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Discovery Nonce                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                  Figure 11: 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.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  V=0  |Type=2 |                  Reserved                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Discovery Nonce                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                 Relay Address (IPv4 or IPv6)                  ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
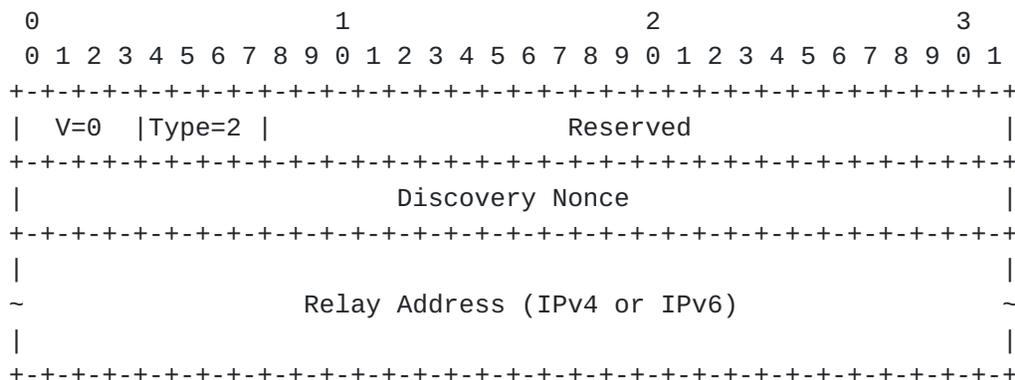
                Figure 12: Relay Advertisement Message Format

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.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| V=0  |Type=3 |   Reserved  |P|            Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Request Nonce                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                     Figure 13: 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.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  V=0  |Type=4 | Reserved  |L|G|        Response MAC           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                                 +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Request Nonce                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|              Encapsulated General Query Message               |
~                 IPv4:IGMPv3(Membership Query)                 ~
|                  IPv6:MLDv2(Listener Query)                   |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Gateway Port Number      |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
|                                                               |
+                                                               +
|              Gateway IP Address (IPv4 or IPv6)                |
+                                                               +
|                                                               |
+                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
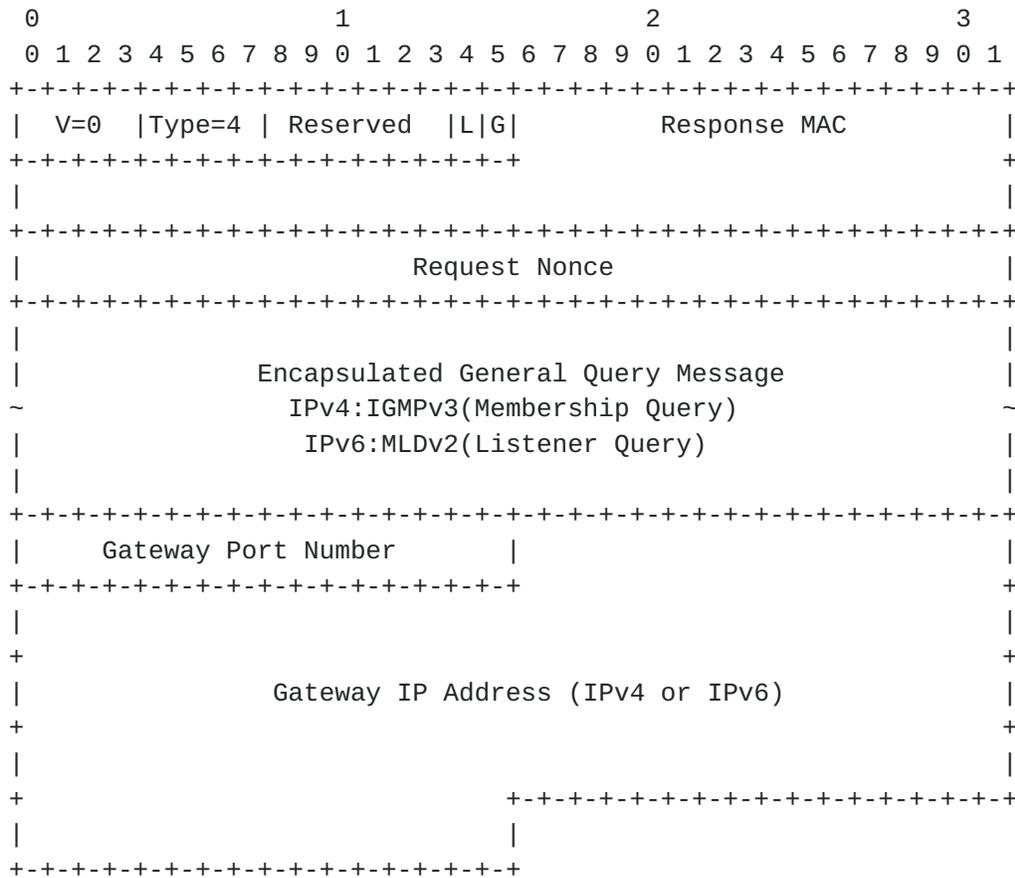
                  Figure 14: Membership Query Message Format

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 QQIC
field is defined in Section 4.1.7 in [RFC3376] and Section 5.1.9 in
   [RFC3810].

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. The QRV field is defined in Section 4.1.6
   in [RFC3376] and Section 5.1.8 in [RFC3810].

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 a 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 port number.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  V=0  |Type=5 |   Reserved    |          Response MAC         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Request Nonce                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|          Encapsulated Group Membership Update Message         |
~             IPv4:IGMP(Membership Report|Leave Group)          ~
|              IPv6:MLD(Listener Report|Listener Done)          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
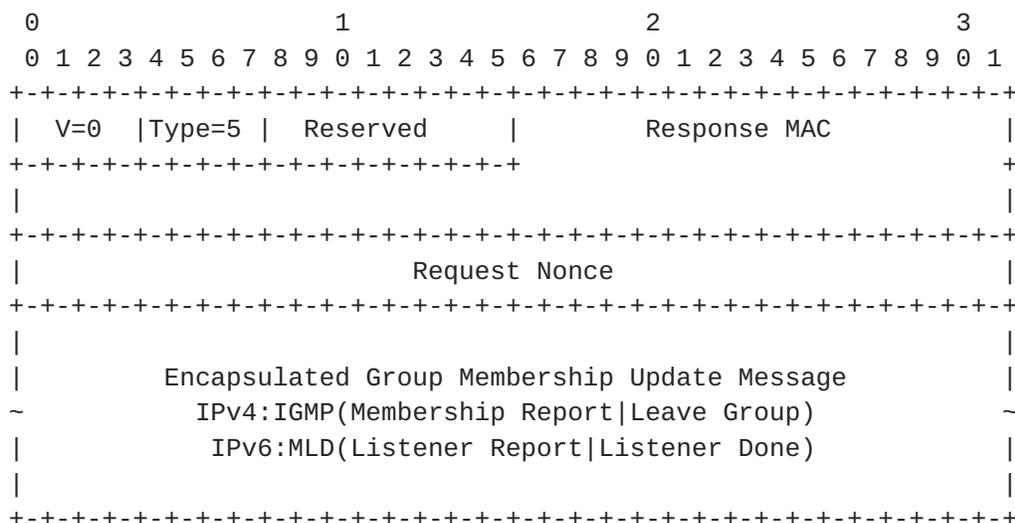
Figure 15: Membership Update Message Format

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 of 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.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  V=0  |Type=6 |    Reserved   |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
   |                                                               |
   ~                     IP Multicast Packet                       ~
   |                                                               |
   +             - - - - - - - - - - - - - - - - - - - - - - - -+
   |            :               :               :               :
   +-+-+-+-+-+-+-+-+- - - - - - - - - - - - - - - - - - - - - - - - -
```
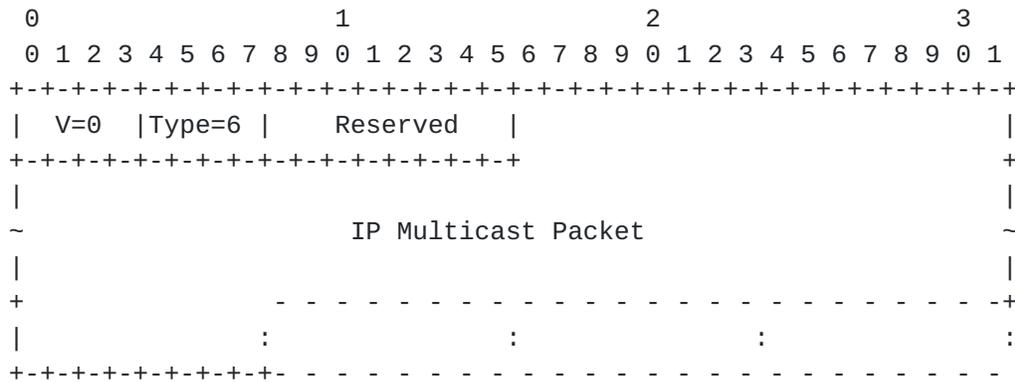
                  Figure 16: Multicast Data Message Format

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.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| V=0  |Type=7 |  Reserved     |           Response MAC         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                                +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Request Nonce                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Gateway Port Number        |                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                              +
|                                                               |
+                                                               +
|              Gateway IP Address (IPv4 or IPv6)                |
+                                                               +
|                                                               |
+                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
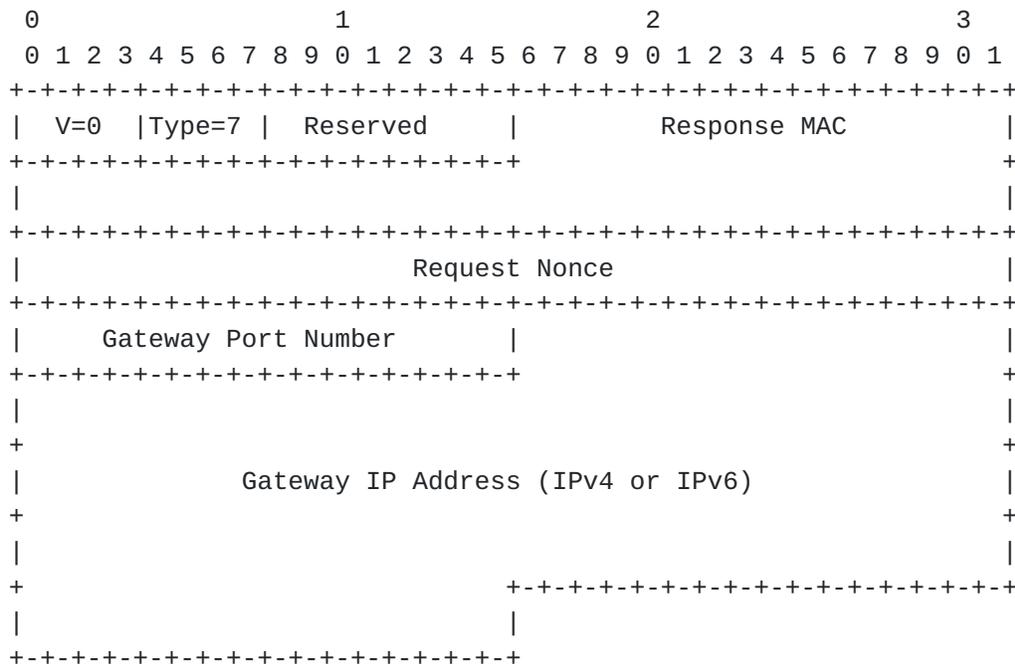
Figure 17: Membership Teardown Message Format

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 Gateway IP Address, Gateway Port Number, Request Nonce fields and a private secret (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 used 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 encapsulated IGMP datagrams generated by a gateway MUST conform
to the descriptions 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; a gateway MAY use any
source address value in an IGMP report datagram including the
"unspecified" address (all octets are zero). This exception is made
because a gateway pseudo-interface might not possess a valid IPv4
address, and even if an address has been assigned to the interface,
that address might not be a valid link-local source address on any
relay interface. It is for this reason that a relay must accept
encapsulated IGMP reports regardless of the source address they
carry. See Section 5.3.1.

The encapsulated MLD messages generated by a gateway 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; a gateway MAY use any
source address value in an MLD report datagram including the
"unspecified" address (all octets are zero). This exception is made

because a gateway pseudo-interface might not possess a valid IPv6
address, and even if an address has been assigned to the interface,
that address might not be a valid link-local source address on any
relay interface. As with IGMP, it is for this reason that a relay
must accept encapsulated MLD reports regardless of the source address
they carry. 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. See
Section 4.1.6 in [RFC3376] and Section 5.1.8 in [RFC3810].

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.

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

   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 [RFC6936]. If supported, a gateway SHOULD
employ the solution described in [RFC6936] 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 AMT 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
   [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]).

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.

```
                    Gateway             Relay-1
                    -------             -------
                       :                   :
    Query     Expired |                   |
    Timer (QT)------->|                   |
                      |  Relay Discovery  |
                      |------------------>|
                      |                   |
                      | Relay Advertisement|
                      |<------------------|
                      |                   |
                      |     Request       |
                      |------------------>|
                      |                   |
                      | Membership Query  |
                      |<==================|
              Start   |                   |
             (QT)<--------| Membership Update  |
                      |==================>|
                      |                   |
                      ~                   ~        Relay-2
             Expired |                   |        -------
            (QT)-------->|                   |           :
                      |  Relay Discovery  |           |
                      |----------------------------------->|
                      |                   |           |
                      | Relay Advertisement|          |
                      |<-----------------------------------|
                      |                   |           |
                      |     Teardown      |           |
                      |------------------>|           |
                      |                   |           |
                      |     Request       |           |
                      |----------------------------------->|
                      |                   |           |
                      | Membership Query  |           |
                      |<===================================|
              Start   |                   |           |
             (QT)<--------| Membership Update  |          |
                      |===================================>|
                      |                   |           |
                      :                   :           :
```
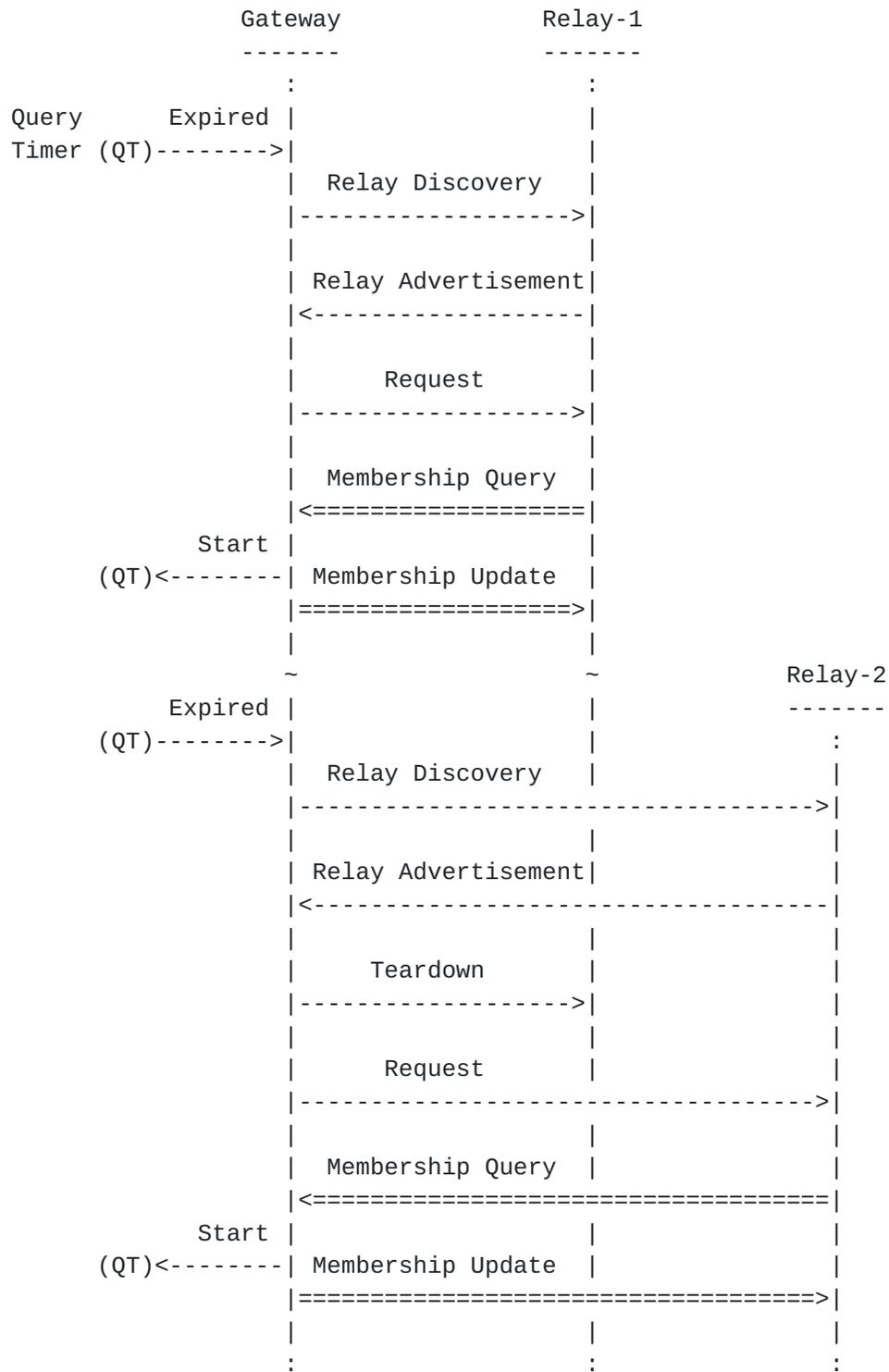
                   Figure 18: Teardown After Relay Address Change

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 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 (See Section 4.1.6 in
[RFC3376] and Section 5.1.7 in [RFC3810]). 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 on
firewall and 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 encapsulated IGMPv3 general query datagrams generated by a relay
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;
a relay MAY use any source IP address for an IGMP general query
datagram including the "unspecified" address (all octets are zero).
This exception is made because any source address that a relay might
normally send may not be a valid link-local address on any gateway
interface. It is for this reason that a gateway must accept
encapsulated IGMP queries regardless of the source address they
carry. See Section 5.2.1.

The encapsulated MLDv2 general query datagrams generated by a relay
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;
a relay MAY use any source IP address for an MLD general query
datagram including the "unspecified" address (all octets are zero).
This exception is made because any source address that a relay might
normally send may not be a valid link-local address on any gateway
interface. As with IGMP, it is for this reason that a gateway must
accept encapsulated MLD queries regardless of the source address they
carry. 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 4.1.7 in [RFC3376] and Section 5.1.9 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. The QRV field
   is defined in Section 4.1.6 in [RFC3376] and Section 5.1.8 in
   [RFC3810].

   A relay SHOULD set the Maximum Response 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 IGMPv3/MLDv2 Query Response
   Interval variable, if available, to generate the Maximum Response
   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. The Maximum Response Code field is defined in
   Section 4.1.1 in [RFC3376] and Section 5.1.3 in [RFC3810]. 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*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 or messages and send that message or
   messages 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 If the multicast IP datagram size exceeds the Tunnel MTU as
      determined according to the procedure described in
      Section 5.3.3.6.1, the relay must execute the procedure described
      in Section 5.3.3.6.2.

   o Encapsulate and transmit the IP datagram according to the
      procedure described in Section 5.3.3.6.3.

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

5.3.3.6.1. Path and Tunnel MTU

   A relay MUST compute a Tunnel MTU (TMTU) value for each AMT tunnel
   that originates on the relay. A relay will use the TMTU value to
   determine whether an incoming multicast IP datagram can be delivered
   downstream in a Membership Data message without fragmentation. A
   relay MUST compute the TMTU by subtracting the size of the Membership
   Data message headers (IP, UDP, and AMT) from the current Path MTU
   (PMTU) associated with each AMT tunnel. The relay MUST maintain a
   PMTU value on a per-tunnel or per-relay basis. A relay MUST support
   one or both of the following methods for determining the PMTU value:

   o The relay MAY provide a configuration option that establishes a
      fixed PMTU that will be applied to all AMT tunnels originating at
      the relay.

   o The relay MAY dynamically adjust PMTU value(s) in response to
      receipt of ICMP/ICMPv6 "Datagram Too Big" messages as described in
      [RFC1191] and [RFC1981].

If a relay supports dynamic adjustment of per-tunnel or per-relay
PMTU values in response to ICMP messages, the relay MUST provide a
configuration option that disables this feature and also provide a
configuration option that establishes a minimum PMTU for all tunnels.
These configuration options may be used to mitigate certain types of
denial of service attacks (See ([Section 6])). When dynamic PMTU
adjustments are disabled, the PMTU for all tunnels MUST default to
the Link MTU (first-hop) on the downstream interface.

[5.3.3.6.2]. MTU Filtering Procedure

This section defines procedures that a relay must execute when it
receives a multicast datagram whose size is greater than the Tunnel
MTU of the tunnel or tunnels through which it must be delivered.

[5.3.3.6.2.1]. IPv4 Multicast IP Datagrams

If the DF bit in the multicast datagram header is set to 1 (Don't
Fragment), the relay MUST discard the packet and, if the datagram
originated from an SSM source, send an ICMPv4 [[RFC0792]] Destination
Unreachable message to the source, with type equal to 4
(fragmentation needed and DF set). The ICMP Destination Unreachable
message MUST contain an next-hop MTU (as specified by [[RFC1191]]) and
the relay MUST set the next-hop MTU to the TMTU associated with the
tunnel or tunnels. If the DF bit in the multicast datagram header is
set to 0 (May Fragment), the relay MUST fragment the datagram and
encapsulate each fragment within Multicast Data messages for
transmission through the tunnel or tunnels. This ensures that
gateways will receive complete, non-fragmented Multicast Data
messages, containing fragmented multicast datagram payloads. The
relay SHOULD avoid generating a separate ICMP message for each
tunnel, but instead send a single ICMP message with a Next-hop MTU
equal to the smallest TMTU of all tunnels to which the datagram was
to be forwarded.

[5.3.3.6.2.2]. IPv6 Multicast IP Datagrams

The relay MUST discard the packet and, if the datagram originated
from an SSM source, send an ICMPv6 [[RFC4443]] Packet Too Big message
to the payload source. The MTU specified in the Packet Too Big
message MUST be equal to the TMTU associated with the tunnel or
tunnels. The relay SHOULD avoid generating a separate ICMPv6 message
for each tunnel, but instead send a single ICMPv6 message with a
Next-hop MTU equal to the smallest TMTU of all tunnels to which the
datagram was to be forwarded.

5.3.3.6.3. Encapsulation Procedure

   A relay encapsulates a multicast IP datagram in a UDP/IP Membership
   Data message, using the tunnel endpoint UDP/IP address as the
   destination address and the unicast relay address and IANA-assigned
   AMT port number 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.

   If possible, the relay SHOULD compute a valid, non-zero checksum for
   the UDP datagram carrying the Multicast Data message. See
   Section 4.2.2.3.

   The following sections describe additional requirements related to
   the IP protocol of the tunnel and that of the multicast IP datagram.

5.3.3.6.3.1. Tunneling over IPv4

   When a relay delivers an IPv4 payload over an IPv4 tunnel, and the DF
   Bit in the payload header is set to 1 (Don't Fragment), the relay
   MUST set the DF bit in the Multicast Data IP header to 1. When a
   relay delivers an IPv4 payload over an IPv4 tunnel, and the DF Bit in
   the payload header is set to 0 (May Fragment), by default, the relay
   MUST set the DF bit in the Multicast Data IP header to 1. However, a
   relay MAY provide a configuration option that allows the DF bit to be
   copied from the payload header to the Multicast Data IP header to
   allow downstream fragmentation of the Multicast Data message. When a
   relay delivers an IPv6 payload over an IPv4 tunnel, the relay MUST
   set the DF bit in the Multicast Data IP header to 1. The relay MUST
   NOT transmit a Multicast Data message with an IP header in which the
   MF (More Fragments) bit is set to 1.

5.3.3.6.3.2. Tunneling over IPv6

   When a tunneling over IPv6, a relay MUST NOT emit a Multicast Data
   message datagram containing an IPv6 fragment header.

5.3.3.6.4. Handling Destination Unreachable Messages

   If a relay receives a sequence of ICMP or ICMPv6 messages of type
   "Destination Unreachable" in response to transmission of a sequence
   of AMT Multicast Data messages to a gateway, the relay SHOULD
   discontinue sending messages to that gateway and shutdown the tunnel
   for that gateway (Handling of ICMP "Destination Unreachable" messages
   with code 4, "fragmentation required" is covered in
   Section 5.3.3.6.1). If a relay provides this capability, it MUST
   provide a configuration option that indicates what number of

sequential "Destination Unreachable" messages can be received and
ignored before the relay will automatically shutdown a tunnel.

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:

((Robustness_Variable) * (Query_Interval)) + 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 computation is required in the following situations:

   o To generate a Response MAC value from a Request message for
      inclusion in a Membership Query message.

   o To generate a Response MAC value from a Membership Update message
      for use in authenticating the Response MAC carried within that
      message.

   o To generate a Response MAC value from 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.

The Path and Tunnel MTU adjustment (discovery) procedure described in
Section 5.3.3.6.1 is vulnerable to two denial of service attacks (see
Section 8 of [RFC1191] for details). Both attacks are based upon on
a malicious party sending forged ICMPv4 Destination Unreachable or
ICMPv6 Packet Too Big messages to a host. In the first attack, the
forged message indicates an inordinately small Path MTU. In the
second attack, the forged message indicates an inordinately large
Path MTU. In both cases, throughput is adversely affected. In order
to mitigate such attacks, relay implementations MUST include a
configuration option to disable Path MTU adjustments on AMT tunnels.

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

   The following unicast prefixes have been assigned to provide anycast
   routing of relay discovery messages to public AMT Relays as described
   in Section 4.1.4.

7.1.1. IPv4

   We suggest that IANA assign an x.x.x.x/24 from the IPv4 Recovered
   Address Space Registry, but any /24 which has been unassigned and
   unadvertised for at least twelve months is acceptable. The block
   should be registered as follows:

```
           +----------------------+----------------+
           | Attribute            | Value          |
           +----------------------+----------------+
           | Address Block        | x.x.x.x./24    |
           | Name                 | AMT            |
           | RFC                  | [TBD]          |
           | Allocation Date      | [TBD]          |
           | Termination Date     | N/A            |
           | Source               | True           |
           | Destination          | True           |
           | Forwardable          | True           |
           | Global               | True           |
           | Reserved-by-Protocol | False          |
           +----------------------+----------------+
```

7.1.2. IPv6

IANA should register the following special-purpose address block for
IPv6 anycast AMT relay discovery.

```
+----------------------+----------------+
| Attribute            | Value          |
+----------------------+----------------+
| Address Block        | 2001:0003::/32 |
| Name                 | AMT            |
| RFC                  | [TBD]          |
| Allocation Date      | [TBD]          |
| Termination Date     | N/A            |
| Source               | True           |
| Destination          | True           |
| Forwardable          | True           |
| Global               | True           |
| Reserved-by-Protocol | False          |
+----------------------+----------------+
```

7.2. UDP Port Number

The UDP port number 2268 has been reserved with IANA for use in the
implementation and deployment of AMT. The protocol described by this
document continues to use this port number according to the intent of
the original request. IANA should assign this port number to AMT
upon acceptance of this I-D.

8. Contributors

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

        Amit Aggarwal
        Microsoft Corporation
        One Microsoft Way
        Redmond, WA 98052-6399
        USA
        Email: amitag@microsoft.com

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

        Dirk Ooms
        OneSparrow
        Robert Molsstraat 11; 2018 Antwerp
        Belgium
        EMail: dirk@onesparrow.com

        Tom Pusateri
        !j
        Wake Forest, NC
        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:

10. References

10.1. Normative References

   [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.

   [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

   [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September
              1981.

   [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5,
              RFC 792, September 1981.

   [RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5,
              RFC 1112, August 1989.

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

   [RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321,
              April 1992.

   [RFC1546] Partridge, C., Mendez, T., and W. Milliken, "Host
              Anycasting Service", RFC 1546, November 1993.

   [RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery
              for IP version 6", RFC 1981, August 1996.

   [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.

   [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC
              Text on Security Considerations", BCP 72, RFC 3552, July
              2003.

   [RFC4271]   Rekhter, Y., Li, T., and S. Hares, "A Border Gateway
               Protocol 4 (BGP-4)", RFC 4271, January 2006.

   [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.

   [RFC4786]   Abley, J. and K. Lindqvist, "Operation of Anycast
               Services", BCP 126, RFC 4786, December 2006.

   [RFC6935]   Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and
               UDP Checksums for Tunneled Packets", RFC 6935, April 2013.

   [RFC6936]   Fairhurst, G. and M. Westerlund, "Applicability Statement
               for the Use of IPv6 UDP Datagrams with Zero Checksums",
               RFC 6936, April 2013.

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 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.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| V=0  |4 or 5|   Reserved   | |           Response MAC          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                                  +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Request Nonce                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
```
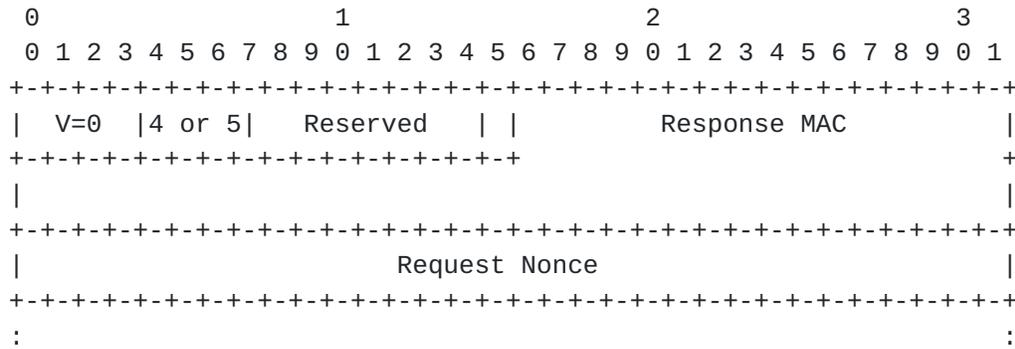
                  Figure 19: The Opaque Response MAC Field

A relay may use the opaque Response MAC field to store a cookie as
follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| V=0  |4 or 5|   Reserved   | |            Timestamp            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      MD5(Secret,Timestamp,IP_ADDR,IP_PORT,Request-Nonce)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Request Nonce                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
```
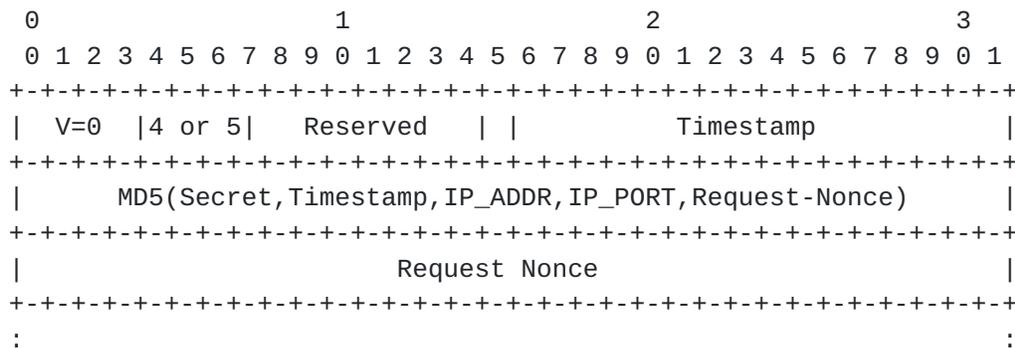
    Figure 20: Using The Response MAC Field To Carry An Authentication
                               Cookie

The timestamp is an unsigned integer measured relative to the start
time of relay. The age of the MAC is computed by subtracting the MAC
timestamp from the current system timestamp. The operands must be

unsigned 16-bit integers and the subtraction must use unsigned
arithmetic to allow for timestamp wrap-around. The timestamp
resolution must provide range sufficient to handle the maximum
allowable age for a MAC, e.g., a resolution of 1 second allows a
maximum age of 18 hours. The timestamp should start at a random
value by adding a random offset, computed at startup, to the current
system time.

```
      +------------------------+-------------/ /----------------+
  -->| Timestamp(N1) [16-bits] |    Random Secret [128-bits]    |
  |   +------------------------+-------------/ /----------------+
  |_____
                                                                    |
      +------------------------+-------------/ /----------------+  |
  -->| Timestamp(N1) [16-bits] |    Random Secret [128-bits]    |--
  |   +------------------------+-------------/ /----------------+
  |_____
                                                                    |
      +------------------------+-------------/ /----------------+  |
  -->| Timestamp(N1) [16-bits] |    Random Secret [128-bits]    |--
  |   +------------------------+-------------/ /----------------+
  |
  |__ Current
      Secret
```
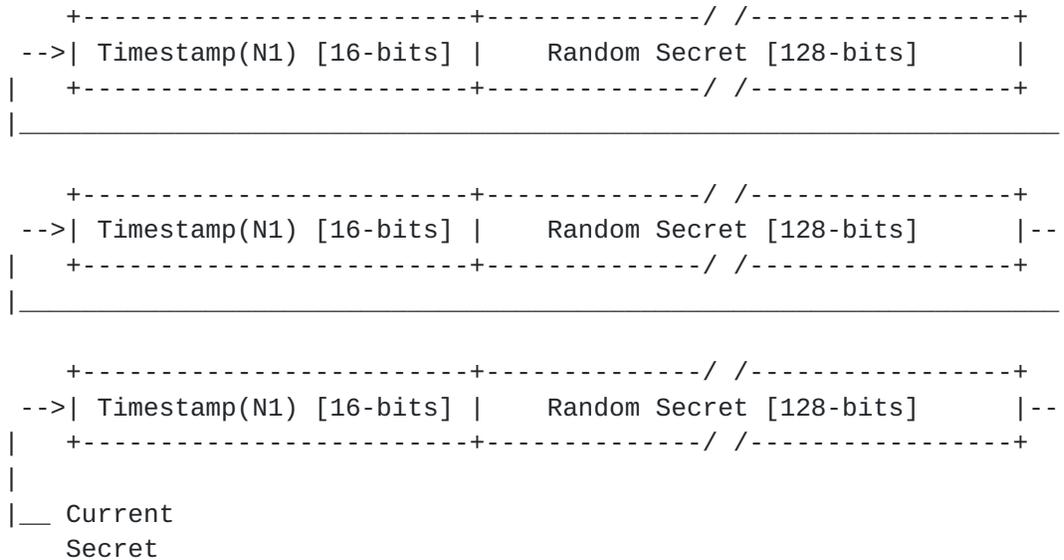
                     Figure 21: Private Secret Queue

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

   Phone: +1 541 343 6790
   Email: gbumgard@gmail.com