DNS Reverse IP AMT Discovery
draft-ietf-mboned-driad-amt-discovery-01

Abstract

   This document updates RFC 7450 (Automatic Multicast Tunneling, or
   AMT) by extending the relay discovery process to use a new DNS
   resource record named AMTRELAY when discovering AMT relays for
   source-specific multicast channels. The reverse IP DNS zone for a
   multicast sender's IP address is configured to use AMTRELAY resource
   records to advertise a set of AMT relays that can receive and forward
   multicast traffic from that sender over an AMT tunnel.

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 https://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 August 18, 2019.

Copyright Notice

Table of Contents

1. Introduction

   This document defines DNS Reverse IP AMT Discovery (DRIAD), a
   mechanism for AMT gateways to discover AMT relays that are capable of
   forwarding multicast traffic from a known source IP address.

   AMT (Automatic Multicast Tunneling) is defined in [RFC7450], and
   provides a method to transport multicast traffic over a unicast
   tunnel, in order to traverse non-multicast-capable network segments.

   Section 4.1.5 of [RFC7450] explains that the relay selection process
   for AMT is intended to be more flexible than the particular discovery
   method described in that document, and further explains that the
   selection process might need to depend on the source of the multicast
   traffic in some deployments, since a relay must be able to receive
   multicast traffic from the desired source in order to forward it.

   That section goes on to suggest DNS-based queries as a possible
   solution. DRIAD is a DNS-based solution, as suggested there. This
   solution also addresses the relay discovery issues in the
   "Disadvantages" lists in Section 3.3 of [RFC8313] and Section 3.4 of
   [RFC8313].

   The goal for DRIAD is to enable multicast connectivity between
   separate multicast-enabled networks when neither the sending nor the
   receiving network is connected to a multicast-enabled backbone,
   without pre-configuring any peering arrangement between the networks.

   This document updates Section 5.2.3.4 of [RFC7450] by adding a new
   extension to the relay discovery procedure.

1.1. Background

   The reader is assumed to be familiar with the basic DNS concepts
   described in [RFC1034], [RFC1035], and the subsequent documents that
   update them, particularly [RFC2181].

The reader is also assumed to be familiar with the concepts and
terminology regarding source-specific multicast as described in
[RFC4607] and the use of IGMPv3 [RFC3376] and MLDv2 [RFC3810] for
group management of source-specific multicast channels, as described
in [RFC4604].

The reader should also be familiar with AMT, particularly the
terminology listed in Section 3.2 of [RFC7450] and Section 3.3 of
[RFC7450].

1.2. Terminology

1.2.1. Relays and Gateways

When reading this document, it's especially helpful to recall that
once an AMT tunnel is established, the relay receives native
multicast traffic and sends unicast tunnel-encapsulated traffic to
the gateway, and the gateway receives the tunnel-encapsulated
packets, decapsulates them, and forwards them as native multicast
packets, as illustrated in Figure 1.

```
 Multicast  +-----------+  Unicast  +-------------+  Multicast
>---------> | AMT relay | >=======> | AMT gateway | >--------->
            +-----------+           +-------------+
```
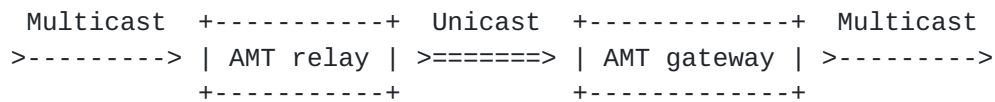
Figure 1: AMT Tunnel Illustration

1.2.2. Definitions

```
+------------+-----------------------------------------------------+
|       Term | Definition                                          |
+------------+-----------------------------------------------------+
|      (S,G) | A source-specific multicast channel, as described in|
|            | [RFC4607]. A pair of IP addresses with a source host|
|            | IP and destination group IP.                        |
|            |                                                     |
|  discovery | A broker or load balancer for AMT relay discovery,  |
|     broker | as mentioned in section 4.2.1.1 of [RFC7450].       |
|            |                                                     |
| downstream | Further from the source of traffic, as described in |
|            | [RFC7450].                                          |
|            |                                                     |
|       FQDN | Fully Qualified Domain Name, as described in        |
|            | [RFC8499]                                           |
|            |                                                     |
|    gateway | An AMT gateway, as described in [RFC7450]           |
|            |                                                     |
|     L flag | The "Limit" flag described in Section 5.1.1.4 of    |
|            | [RFC7450]                                           |
|            |                                                     |
|      relay | An AMT relay, as described in [RFC7450]             |
|            |                                                     |
|        RPF | Reverse Path Forwarding, as described in [RFC5110]  |
|            |                                                     |
|         RR | A DNS Resource Record, as described in [RFC1034]    |
|            |                                                     |
|     RRType | A DNS Resource Record Type, as described in         |
|            | [RFC1034]                                           |
|            |                                                     |
|        SSM | Source-specific multicast, as described in [RFC4607]|
|            |                                                     |
|   upstream | Closer to the source of traffic, as described in    |
|            | [RFC7450].                                          |
+------------+-----------------------------------------------------+
```

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
"OPTIONAL" in this document are to be interpreted as described in
[RFC2119] and [RFC8174] when, and only when, they appear in all
capitals, as shown here.

2. Relay Discovery Operation

2.1. Overview

The AMTRELAY resource record (RR) defined in this document is used to
publish the IP address or domain name of a set of AMT relays or
discovery brokers that can receive, encapsulate, and forward
multicast traffic from a particular sender.

The sender is the owner of the RR, and configures the zone so that it
contains a set of RRs that provide the addresses or domain names of
AMT relays (or discovery brokers that advertise relays) that can
receive multicast IP traffic from that sender.

This enables AMT gateways in remote networks to discover an AMT relay
that is capable of forwarding traffic from the sender. This in turn
enables those AMT gateways to receive the multicast traffic tunneled
over a unicast AMT tunnel from those relays, and then to pass the
multicast packets into networks or applications that are using the
gateway to subscribe to traffic from that sender.

This mechanism only works for source-specific multicast (SSM)
channels. The source address of the (S,G) is reversed and used as an
index into one of the reverse mapping trees (in-addr.arpa for IPv4,
as described in Section 3.5 of [RFC1035], or ip6.arpa for IPv6, as
described in Section 2.5 of [RFC3596]).

This mechanism should be treated as an extension of the AMT relay
discovery procedure described in Section 5.2.3.4 of [RFC7450]. A
gateway that supports this method of AMT relay discovery SHOULD use
this method whenever it's performing the relay discovery procedure,
and the source IP addresses for desired (S,G)s are known to the
gateway, and conditions match the requirements outlined in
Section 2.4.

Some detailed example use cases are provided in Section 3, and other
applicable example topologies appear in Section 3.3 of [RFC8313],
Section 3.4 of [RFC8313], and Section 3.5 of [RFC8313].

2.2. Signaling and Discovery

This section describes a typical example of the end-to-end process
for signaling a receiver's join of a SSM channel that relies on an
AMTRELAY RR.

The example in Figure 2 contains 2 multicast-enabled networks that
are both connected to the internet with non-multicast-capable links,
and which have no direct association with each other.

A content provider operates a sender, which is a source of multicast
traffic inside a multicast-capable network.

An end user who is a customer of the content provider has a
multicast-capable internet service provider, which operates a
receiving network that uses an AMT gateway. The AMT gateway is
DRIAD-capable.

The content provider provides the user with a receiving application
that tries to subscribe to at least one (S,G). This receiving
application could for example be a file transfer system using FLUTE
[RFC6726] or a live video stream using RTP [RFC3550], or any other
application that might subscribe to a SSM channel.
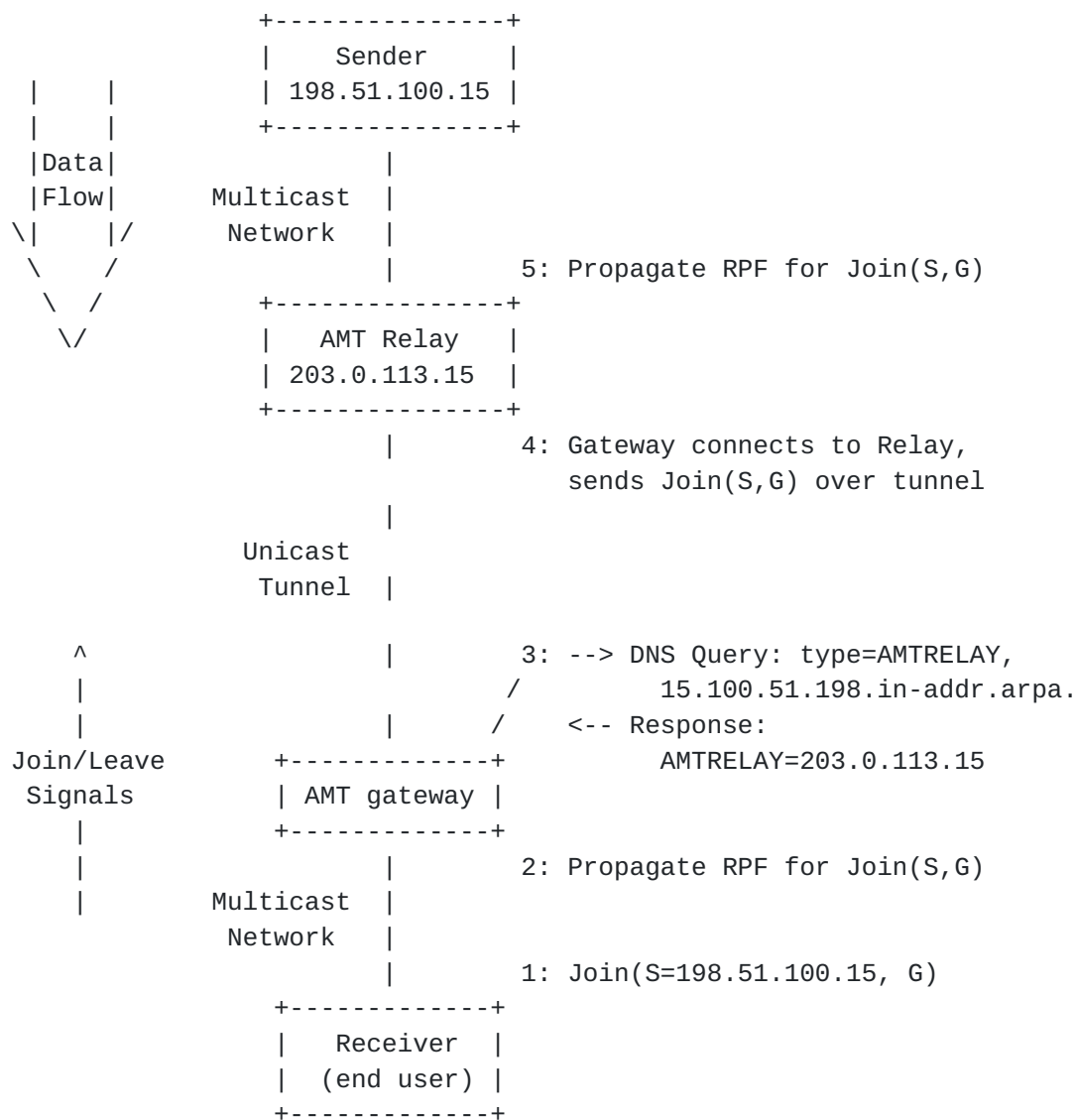
```
                       +---------------+
                       |    Sender     |
      |    |           | 198.51.100.15 |
      |    |           +---------------+
      |    |
     |Data|                   |
     |Flow|      Multicast    |
     \|    |/       Network    |
      \    /                   |            5: Propagate RPF for Join(S,G)
       \  /               +---------------+
        \/                |   AMT Relay   |
                          | 203.0.113.15  |
                          +---------------+
                                  |            4: Gateway connects to Relay,
                                                  sends Join(S,G) over tunnel

                                  |
                       Unicast
                        Tunnel  |

         ^                      |        3: --> DNS Query: type=AMTRELAY,
         |                              /        15.100.51.198.in-addr.arpa.
         |                      |      /     <-- Response:
     Join/Leave       +-------------+            AMTRELAY=203.0.113.15
      Signals         | AMT gateway |
         |            +-------------+
         |                      |        2: Propagate RPF for Join(S,G)
         |            Multicast  |
                       Network   |
                                 |        1: Join(S=198.51.100.15, G)
                    +-------------+
                    |   Receiver  |
                    |  (end user) |
                    +-------------+


                     Figure 2: DRIAD Messaging
```

In this simple example, the sender IP is 198.51.100.15, and the relay
IP is 203.0.113.15.

The content provider has previously configured the DNS zone that
contains the domain name "15.100.51.198.in-addr.arpa.", which is the
reverse lookup domain name for his sender. The zone file contains an
AMTRELAY RR with the Relay's IP address. (See Section 4.3 for
details about the AMTRELAY RR format and semantics.)

The sequence of events depicted in Figure 2 is as follows:

1. The end user starts the app, which issues a join to the (S,G):
    (198.51.100.15, 232.252.0.2).

2. The join propagates with RPF through the multicast-enabled
    network with PIM [RFC7761] or another multicast routing
    mechanism, until the AMT gateway receives a signal to join the
    (S,G).

3. The AMT gateway performs a reverse DNS lookup for the AMTRELAY
    RRType, by sending an AMTRELAY RRType query for the FQDN
    "15.100.51.198.in-addr.arpa.", using the reverse IP domain name
    for the sender's source IP address (the S from the (S,G)), as
    described in Section 3.5 of [RFC1035].

    The DNS resolver for the AMT gateway uses ordinary DNS recursive
    resolution until it has the authoritative result that the content
    provider configured, which informs the AMT gateway that the relay
    address is 203.0.113.15.

4. The AMT gateway performs AMT handshakes with the AMT relay as
    described in Section 4 of [RFC7450], then forwards a Membership
    report to the relay indicating subscription to the (S,G).

5. The relay propagates the join through its network toward the
    sender, then forwards the appropriate AMT-encapsulated traffic to
    the gateway, which decapsulates and forwards it as native
    multicast through its downstream network to the end user.

2.3. Happy Eyeballs

2.3.1. Overview

Often, multiple choices of relay will exist for a gateway using DRIAD
for relay discovery. It is RECOMMENDED that DRIAD-capable gateways
implement a Happy Eyeballs [RFC8305] algorithm to support connecting
to multiple relays in parallel.

The parallel discovery logic of a Happy Eyeballs algorithm serves to
reduce join latency for the initial join of a SSM channel. This
section and Section 2.4.2 taken together provide guidance on use of a
Happy Eyeballs algorithm for the case of establishing AMT
connections.

2.3.2. Connection Definition

Section 5 of [RFC8305] non-normatively describes success at a
connection attempt as "generally when the TCP handshake completes".

There is no normative definition of a connection in the AMT
specification [RFC7450], and there is no TCP connection involved in
an AMT tunnel.

However, the concept of an AMT connection in the context of a Happy
Eyeballs algorithm is a useful one, and so this section provides the
following normative definition:

o An AMT connection is completed successfully when the gateway
   receives from a newly discovered relay a valid Membership Query
   message (Section 5.1.4 of [RFC7450]) that does not have the L flag
   set.

See Section 2.5.5 for further information about the relevance of the
L flag to the establishment of a Happy Eyeballs connection.

2.4. Optimal Relay Selection

2.4.1. Overview

The reverse source IP DNS query of an AMTRELAY RR is a good way for a
gateway to discover a relay that is known to the sender.

However, it is NOT necessarily a good way to discover the best relay
for that gateway to use, because the RR will only provide information
about relays known to the source.

If there is an upstream relay in a network that is topologically
closer to the gateway and able to receive and forward multicast
traffic from the sender, that relay is better for the gateway to use,
since more of the network path uses native multicast, allowing more
chances for packet replication. But since that relay is not known to
the sender, it won't be advertised in the sender's reverse IP DNS
record. An example network that illustrates this scenario is
outlined in Section 3.1.2.

It's only appropriate for an AMT gateway to discover an AMT relay by
querying an AMTRELAY RR owned by a sender when all of these
conditions are met:

1. The gateway needs to propagate a join of an (S,G) over AMT,
    because in the gateway's network, no RPF next hop toward the
    source can propagate a native multicast join of the (S,G); and

2. The gateway is not already connected to a relay that forwards
    multicast traffic from the source of the (S,G); and

3. The gateway is not configured to use a particular IP address for
    AMT discovery, or a relay discovered with that IP is not able to
    forward traffic from the source of the (S,G); and

4. The gateway is not able to find an upstream AMT relay with DNS-SD
    [RFC6763], using "_amt._udp" as the Service section of the
    queries, or a relay discovered this way is not able to forward
    traffic from the source of the (S,G) (as described in
    Section 2.5.4.1 or Section 2.5.5).

When the above conditions are met, the gateway has no path within its
local network that can receive multicast traffic from the source IP
of the (S,G).

In this situation, the best way to find a relay that can forward the
required traffic is to use information that comes from the operator
of the sender. When the sender has configured an AMTRELAY RR,
gateways can use the DRIAD mechanism defined in this document to
discover the relay information provided by the sender.

2.4.2. Preference Ordering

This section defines a preference ordering for relay addresses during
the relay discovery process. Gateways are encouraged to implement a
Happy Eyeballs algorithm, but even gateways that do not implement a
Happy Eyeballs algorithm SHOULD use this ordering, except as noted.

When establishing an AMT tunnel to forward multicast data, it's very
important for the discovery process to prioritize the network
topology considerations ahead of address selection considerations, in
order to gain the packet replication benefits from using multicast
instead of unicast tunneling in the multicast-capable portions of the
network path.

The intent of the advice and requirements in this section is to
describe how a gateway should make use of the concurrency provided by
a Happy Eyeballs algorithm to reduce the join latency, while still

prioritizing network efficiency considerations over Address Selection
considerations.

Section 4 of [RFC8305] requires a Happy Eyeballs algorithm to sort
the addresses with the Destination Address Selection defined in
Section 6 of [RFC6724], but for the above reasons, that requirement
is superseded in the AMT discovery use case by the following
considerations:

1. Prefer Local Relays

    Figure 5 and Section 3.1.2 provide a motivating example to prefer
    DNS-SD [RFC6763] for discovery strictly ahead of using the
    AMTRELAY RR controlled by the sender for AMT discovery.

    For this reason, it's RECOMMENDED that AMT gateways by default
    perform service discovery using DNS Service Discovery (DNS-SD)
    [RFC6763] for _amt._udp.<domain> (with <domain> chosen as
    described in Section 11 of [RFC6763]) and use the AMT relays
    discovered that way in preference to AMT relays discoverable via
    the mechanism defined in this document (DRIAD).

2. Let Sender Manage Relay Provisioning

    A related motivating example in the sending-side network is
    provided by considering a sender which needs to instruct the
    gateways on how to select between connecting to Figure 6 or
    Figure 7 (from Section 3.2), in order to manage load and failover
    scenarios in a manner that operates well with the sender's
    provisioning strategy for horizontal scaling of AMT relays.

    In this example about the sending-side network, the precedence
    field described in Section 4.2.1 is a critical method of control
    so that senders can provide the appropriate guidance to gateways
    during the discovery process.

    Therefore, after DNS-SD, the precedence from the RR MUST be used
    for sorting preference ahead of the Destination Address Selection
    ordering from Section 6 of [RFC6724], so that only relay IPs with
    the same precedence are directly compared according to the
    Destination Address Selection ordering.

3. Let Sender Manage Non-DRIAD discovery

    It's also RECOMMENDED that if the well-known anycast IP addresses
    defined in Section 7 of [RFC7450] are suitable for discovering an
    AMT relay that can forward traffic from the source, that a DNS
    record with the AMTRELAY RRType be published by the sender for

those IP addresses along with any other appropriate AMTRELAY RRs
to indicate the best relative precedences for receiving the
source traffic.

Accordingly, AMT gateways SHOULD by default prefer relays first by
DNS-SD if available, then by DRIAD as described in this document (in
precedence order, as described in Section 4.2.1), then with the
anycast addresses defined in Section 7 of [RFC7450] (namely:
192.52.193.1 and 2001:3::1) if those IPs weren't listed in the
AMTRELAY RRs.

This default behavior MAY be overridden by administrative
configuration where other behavior is more appropriate for the
gateway within its network.

Among relay addresses that have an equivalent preference as described
above, a Happy Eyeballs algorithm for AMT MUST use the Destination
Address Selection defined in Section 6 of [RFC6724], as required by
[RFC8305].

Among relay addresses that still have an equivalent preference after
the above orderings, a gateway MUST make a non-deterministic choice
for relay preference ordering, in order to support load balancing by
DNS configurations that provide many relay options. (Note that
gateways not implementing a Happy Eyeballs algorithm are not required
to use the Destination Address Selection ordering, but are still
required to use non-deterministic ordering among equally preferred
relays.)

Note also that certain relay addresses may be excluded from
consideration by the hold-down timers described in Section 2.5.4.1 or
Section 2.5.5. These relays constitute "unusable destinations" under
Rule 1 of the Destination Address Selection, and are also not part of
the superseding considerations described above.

The discovery and connection process for the relay addresses in the
above described ordering MAY operate in parallel, subject to delays
prescribed by the Happy Eyeballs requirements described in Section 5
of [RFC8305] for successively launched concurrent connection
attempts.

2.4.3. Connecting to Multiple Relays

In some deployments, it may be useful for a gateway to connect to
multiple upstream relays and subscribe to the same traffic, in order
to support an active/active failover model. A gateway SHOULD NOT be
configured to do so without guaranteeing that adequate bandwidth is
available.

A gateway configured to do this SHOULD still use the same preference
ordering logic from Section 2.4.2 for both connections. (Note that
this ordering allows for overriding by explicit administrative
configuration where required.)

2.5. Guidelines for Restarting Discovery

2.5.1. Overview

It's expected that gateways deployed in different environments will
use a variety of heuristics to decide when it's appropriate to
restart the relay discovery process, in order to meet different
performance goals (for example, to fulfill different kinds of service
level agreements).

In general, restarting the discovery process is always safe for the
gateway and relay during any of the events listed in this section,
but may cause a disruption in the forwarded traffic if the discovery
process results in choosing a different relay, because this changes
the RPF forwarding tree for the multicast traffic upstream of the
gateway. This is likely to result in some dropped or duplicated
packets from channels actively being tunneled from the old relay to
the gateway.

The degree of impact on the traffic from choosing a different relay
may depend on network conditions between the gateway and the new
relay, as well as the network conditions and topology between the
sender and the new relay, as this may cause the relay to propagate a
new RPF join toward the sender.

Balancing the expected impact on the tunneled traffic against likely
or observed problems with an existing connection to the relay is the
goal of the heuristics that gateways use to determine when to restart
the discovery process.

The non-normative advice in this section should be treated as
guidelines to operators and implementors working with AMT systems
that can use DRIAD as part of the relay discovery process.

2.5.2. Updates to Restarting Events

Section 5.2.3.4.1 of [RFC7450] lists several events that may cause a
gateway to start or restart the discovery procedure.

This document provides some updates and recommendations regarding the
handling of these and similar events. The first 5 events are copied
here and numbered for easier reference, and the following events are
newly added for consideration in this document:

1. When a gateway pseudo-interface is started (enabled).

2. When the gateway wishes to report a group subscription when none
    currently exist.

3. Before sending the next Request message in a membership update
    cycle.

4. After the gateway fails to receive a response to a Request
    message.

5. After the gateway receives a Membership Query message with the L
    flag set to 1.

6. When the gateway wishes to report a (S,G) subscription with a
    source address that does not currently have other group
    subscriptions.

7. When there is a network change detected, for example when a
    gateway is operating inside an end user device or application,
    and the device joins a different network, or when the domain
    portion of a DNS-SD domain name changes in response to a DHCP
    message or administrative configuration.

8. When congestion or substantial loss is detected in the stream of
    AMT packets from a relay.

9. When the gateway has reported one or more (S,G) subscriptions,
    but no traffic is received from the source for some timeout.
    (See Section 2.5.4.1).

This list is not exhaustive, nor are any of the listed events
strictly required to always force a restart of the discovery process.

Note that during event #1, a gateway may use DNS-SD, but does not
have sufficient information to use DRIAD, since no source is known.

2.5.3. Tunnel Stability

In general, subscribers to active traffic flows that are being
forwarded by an AMT gateway are less likely to experience a
degradation in service (for example, from missing or duplicated
packets) when the gateway continues using the same relay, as long the
relay is not overloaded and the network conditions remain stable.

Therefore, gateways SHOULD avoid performing a full restart of the
discovery process during routine cases of event #3 (sending a new
Request message), since it occurs frequently in normal operation.

However, see Section 2.5.4, Section 2.5.6, and Section 2.5.4.3 for
more information about exceptional cases when it may be appropriate
to use event #3.

2.5.4. Traffic Health

2.5.4.1. Absence of Traffic

If a gateway indicates one or more (S,G) subscriptions in a
Membership Update message, but no traffic for any of the (S,G)s is
received in a reasonable time, it's appropriate for the gateway to
restart the discovery process.

If the gateway restarts the discovery process multiple times
consecutively for this reason, the timeout period SHOULD be adjusted
to provide a 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 4 seconds
and a RECOMMENDED maximum_timeout of 120 seconds.

Note that the recommended initial_timeout is larger than the initial
timout recommended in the similar algorithm from Section 5.2.3.4.3 of
[RFC7450]. This is to provide time for RPF Join propagation in the
sending network. Although the timeout values may be administratively
adjusted to support performance requirements, operators are advised
to consider the possibility of join propagation delays between the
sender and the relay when choosing an appropriate timeout value.

Gateways restarting the discovery process because of an absence of
traffic MUST use a hold-down timer that removes this relay from
consideration during subsequent rounds of discovery while active.
The hold-down SHOULD last for no less than 3 minutes and no more than
10 minutes.

2.5.4.2. Loss and Congestion

In some gateway deployments, it is also feasible to monitor the
health of traffic flows through the gateway, for example by detecting
the rate of packet loss by communicating out of band with receivers,
or monitoring the packets of known protocols with sequence numbers.
Where feasible, it's encouraged for gateways to use such traffic
health information to trigger a restart of the discovery process
during event #3 (before sending a new Request message).

However, to avoid synchronized rediscovery by many gateways
simultaneously after a transient network event upstream of a relay

results in many receivers detecting poor flow health at the same
time, it's recommended to add a random delay before restarting the
discovery process in this case.

The span of the random portion of the delay should be no less than 10
seconds by default, but may be administratively configured to support
different performance requirements.

2.5.4.3. Ancient Discovery Information

In most cases, a gateway actively receiving healthy traffic from a
relay that has not indicated load with the L flag should prefer to
remain connected to the same relay, as described in Section 2.5.3.

However, a relay that appears healthy but has been forwarding traffic
for days or weeks may have an increased chance of becoming unstable.
Gateways may benefit from restarting the discovery process during
event #3 (before sending a Request message) after the expiration of a
long-term timeout, on the order of multiple hours, or even days in
some deployments.

It may be beneficial for such timers to consider the amount of
traffic currently being forwarded, and to give a higher probability
of restarting discovery during periods with an unusually low data
rate, to reduce the impact on active traffic while still avoiding
relying on the results of a very old discovery.

Other issues may also be worth considering as part of this heuristic;
for example, if the DNS expiry time of the record that was used to
discover the current relay has not passed, the long term timer might
be restarted without restarting the discovery process.

2.5.5. Relay Loaded or Shutting Down

The L flag (see Section 5.1.4.4 of [RFC7450]) is the preferred
mechanism for a relay to signal overloading or a graceful shutdown to
gateways.

A gateway that supports handling of the L flag should generally
restart the discovery process when it processes a Membership Query
packet with the L flag set. If an L flag is received while a
concurrent Happy Eyeballs discovery process is under way for multiple
candidate relays (Section 2.3), the relay sending the L flag SHOULD
NOT be considered for the relay selection.

It is also RECOMMENDED that gateways avoid choosing a relay that has
recently sent an L flag, with approximately a 10-minute hold-down.
Gateways SHOULD treat this hold-down timer in the same way as the

hold-down in Section 2.5.4.1, so that the relay is removed from
consideration for short-term subsequent rounds of discovery.

2.5.6. Relay Discovery Messages vs. Restarting Discovery

A gateway should only send DNS queries with the AMTRELAY RRType or
the DNS-SD DNS queries for an AMT service as part of starting or
restarting the discovery process.

However, all AMT relays are required to support handling of Relay
Discovery messages (e.g. in Section 5.3.3.2 of [RFC7450]).

So a gateway with an existing connection to a relay can send a Relay
Discovery message to the unicast address of that AMT relay. Under
stable conditions with an unloaded relay, it's expected that the
relay will return its own unicast address in the Relay Advertisement,
in response to such a Relay Discovery message. Since this will not
result in the gateway changing to another relay unless the relay
directs the gateway away, this is a reasonable exception to the
advice against handling event #3 described in Section 2.5.3.

This behavior is discouraged for gateways that do support the L flag,
to avoid sending unnecessary packets over the network.

However, gateways that do not support the L flag may be able to avoid
a disruption in the forwarded traffic by sending such Relay Discovery
messages regularly. When a relay is under load or has started a
graceful shutdown, it may respond with a different relay address,
which the gateway can use to connect to a different relay. This kind
of coordinated handoff will likely result in a smaller disruption to
the traffic than if the relay simply stops responding to Request
messages, and stops forwarding traffic.

This style of Relay Discovery message (one sent to the unicast
address of a relay that's already forwarding traffic to this gateway)
should not be considered a full restart of the relay discovery
process. It is recommended for gateways to support the L flag, but
for gateways that do not support the L flag, sending this message
during event #3 may help mitigate service degradation when relays
become unstable.

2.5.7. Independent Discovery Per Traffic Source

Relays discovered via the AMTRELAY RR are source-specific relay
addresses, and may use different pseudo-interfaces from each other
and from relays discovered via DNS-SD or a non-source-specific
address, as described in Section 4.1.2.1 of [RFC7450].

Restarting the discovery process for one pseudo-interface does not
require restarting the discovery process for other pseudo-interfaces.
Gateway heuristics about restarting the discovery process should
operate independently for different tunnels to relays, when
responding to events that are specific to the different tunnels.

2.6. DNS Configuration

Often an AMT gateway will only have access to the source and group IP
addresses of the desired traffic, and will not know any other name
for the source of the traffic. Because of this, typically the best
way of looking up AMTRELAY RRs will be by using the source IP address
as an index into one of the reverse mapping trees (in-addr.arpa for
IPv4, as described in Section 3.5 of [RFC1035], or ip6.arpa for IPv6,
as described in Section 2.5 of [RFC3596]).

Therefore, it is RECOMMENDED that AMTRELAY RRs be added to reverse IP
zones as appropriate. AMTRELAY records MAY also appear in other
zones, but the primary intended use case requires a reverse IP
mapping for the source from an (S,G) in order to be useful to most
AMT gateways.

When performing the AMTRELAY RR lookup, any CNAMEs or DNAMEs found
MUST be followed. This is necessary to support zone delegation.
Some examples outlining this need are described in [RFC2317].

See Section 4 and Section 4.3 for a detailed explanation of the
contents for a DNS Zone file.

2.7. Waiting for DNS resolution

The DNS query functionality is expected to follow ordinary standards
and best practices for DNS clients. A gateway MAY use an existing
DNS client implementation that does so, and MAY rely on that client's
retry logic to determine the timeouts between retries.

Otherwise, a gateway MAY re-send a DNS query if it does not receive
an appropriate DNS response within some timeout period. If the
gateway retries multiple times, the timeout period SHOULD be adjusted
to provide a random exponential back-off.

As with the waiting process for the Relay Advertisement message from
Section 5.2.3.4.3 of [RFC7450], 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.

3. Example Deployments

3.1. Example Receiving Networks

3.1.1. Tier 3 ISP

   One example of a receiving network is an ISP that offers multicast
   ingest services to its subscribers, illustrated in Figure 3.

   In the example network below, subscribers can join (S,G)s with MLDv2
   or IGMPv3 as described in [RFC4604], and the AMT gateway in this ISP
   can receive and forward multicast traffic from one of the example
   sending networks in Section 3.2 by discovering the appropriate AMT
   relays with a DNS lookup for the AMTRELAY RR with the reverse IP of
   the source in the (S,G).
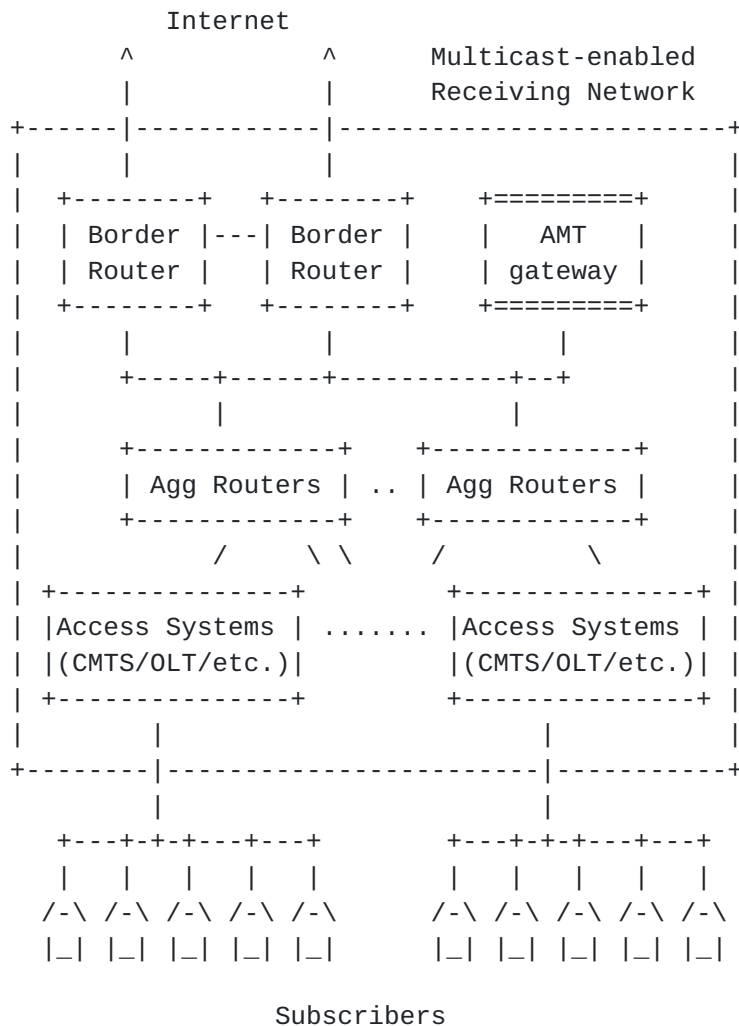
```
                     Internet
              ^               ^      Multicast-enabled
              |               |      Receiving Network
       +------|------------|-----------------------+
       |      |            |                        |
       |   +--------+   +--------+    +=========+    |
       |   | Border |---| Border |    |   AMT   |    |
       |   | Router |   | Router |    | gateway |    |
       |   +--------+   +--------+    +=========+    |
       |      |            |              |          |
       |      +-----+------+-----------+--+          |
       |            |                  |             |
       |      +------------+    +------------+       |
       |      | Agg Routers | .. | Agg Routers |     |
       |      +------------+    +------------+        |
       |          /    \ \     /        \            |
       | +---------------+        +---------------+  |
       | |Access Systems | ....... |Access Systems | |
       | |(CMTS/OLT/etc.)|         |(CMTS/OLT/etc.)| |
       | +---------------+        +---------------+  |
       |        |                        |           |
       +--------|------------------------|-----------+
                |                        |
          +---+-+-+---+---+        +---+-+-+---+---+
          |   |   |   |   |        |   |   |   |   |
          /-\ /-\ /-\ /-\ /-\      /-\ /-\ /-\ /-\ /-\
          |_| |_| |_| |_| |_|      |_| |_| |_| |_| |_|


                       Subscribers

                  Figure 3: Receiving ISP Example
```

. **Small Office**

   Another example receiving network is a small branch office that
   regularly accesses some multicast content, illustrated in Figure 4.

   This office has desktop devices that need to receive some multicast
   traffic, so an AMT gateway runs on a LAN with these devices, to pull
   traffic in through a non-multicast next-hop.

   The office also hosts some mobile devices that have AMT gateway
   instances embedded inside apps, in order to receive multicast traffic
   over their non-multicast wireless LAN. (Note that the "Legacy
   Router" is a simplification that's meant to describe a variety of
   possible conditions- for example it could be a device providing a
   split-tunnel VPN as described in [RFC7359], deliberately excluding
   multicast traffic for a VPN tunnel, rather than a device which is
   incapable of multicast forwarding.)

```
                     Internet
                 (non-multicast)
                        ^
                        |                    Office Network
          +----------|-------------------------------+
          |          |                                |
          |     +---------------+ (Wifi)   Mobile apps |
          |     | Modem+ | Wifi | - - - -  w/ embedded  |
          |     | Router |  AP  |          AMT gateways |
          |     +---------------+                       |
          |          |                                  |
          |          |                                  |
          |     +---------------+                       |
          |     | Legacy Router |                       |
          |     |   (unicast)   |                       |
          |     +---------------+                       |
          |      /        |        \                    |
          |     /         |         \                   |
          | +--------+ +--------+ +--------+=========+   |
          | | Phones | | ConfRm | | Desks  |   AMT   |   |
          | | subnet | | subnet | | subnet | gateway |   |
          | +--------+ +--------+ +--------+=========+   |
          |                                              |
          +----------------------------------------------+
```

              Figure 4: Small Office (no multicast up)

   By adding an AMT relay to this office network as in Figure 5, it's
   possible to make use of multicast services from the example
   multicast-capable ISP in Section 3.1.1.

```
              Multicast-capable ISP
                     ^
                     |                  Office Network
          +----------|----------------------------------+
          |          |                                   |
          |     +---------------+ (Wifi)   Mobile apps   |
          |     | Modem+ | Wifi | - - - -  w/ embedded   |
          |     | Router |  AP  |           AMT gateways |
          |     +---------------+                        |
          |          |                +=======+          |
          |          +---Wired LAN---|  AMT  |          |
          |          |                | relay |          |
          |     +---------------+    +=======+          |
          |     | Legacy Router |                        |
          |     |   (unicast)   |                        |
          |     +---------------+                        |
          |       /       |       \                      |
          |      /        |        \                     |
          | +--------+ +--------+ +--------+=========+   |
          | | Phones | | ConfRm | | Desks  |   AMT   |   |
          | | subnet | | subnet | | subnet | gateway |   |
          | +--------+ +--------+ +--------+=========+   |
          |                                              |
          +----------------------------------------------+
```
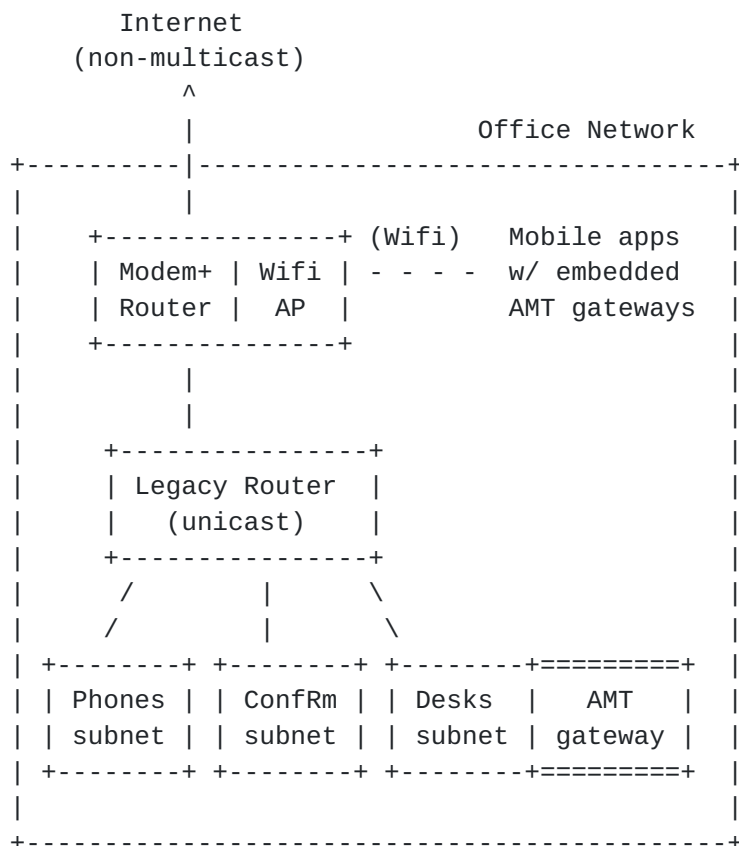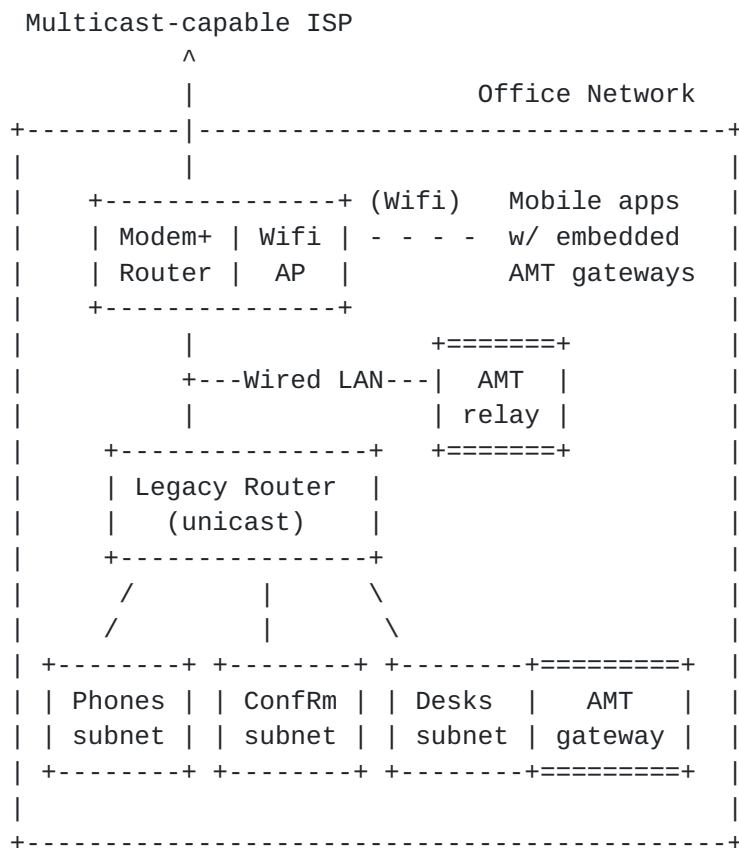
                   Figure 5: Small Office Example

   When multicast-capable networks are chained like this, with a network
   like the one in Figure 5 receiving internet services from a
   multicast-capable network like the one in Figure 3, it's important
   for AMT gateways to reach the more local AMT relay, in order to avoid
   accidentally tunneling multicast traffic from a more distant AMT
   relay with unicast, and failing to utilize the multicast transport
   capabilities of the network in Figure 3.

3.2. Example Sending Networks

3.2.1. Sender-controlled Relays

   When a sender network is also operating AMT relays to distribute
   multicast traffic, as in Figure 6, each address could appear as an
   AMTRELAY RR for the reverse IP of the sender, or one or more domain
   names could appear in AMTRELAY RRs, and the AMT relay addresses can
   be discovered by finding a A or AAAA records from those domain names.

```
                              Sender Network
          +------------------------------------+
          |                                    |
          | +--------+   +======+  +======+ |
          | | Sender |   |  AMT |  |  AMT |  |
          | +--------+   | relay |  | relay | |
          |     |        +======+  +======+ |
          |     |            |         |      |
          |     +-----+------+----------+      |
          |           |                        |
          +-----------|------------------------+
                      v
                 Internet
               (non-multicast)
```

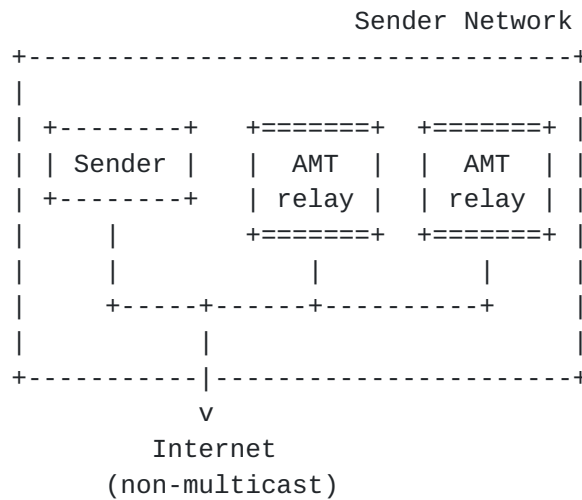   Figure 6: Small Office Example

3.2.2. Provider-controlled Relays

   When an ISP offers a service to transmit outbound multicast traffic
   through a forwarding network, it might also offer AMT relays in order
   to reach receivers without multicast connectivity to the forwarding
   network, as in Figure 7. In this case it's RECOMMENDED that the ISP
   also provide at least one domain name for the AMT relays for use with
   the AMTRELAY RR.

   When the sender wishes to use the relays provided by the ISP for
   forwarding multicast traffic, an AMTRELAY RR should be configured to
   use the domain name provided by the ISP, to allow for address
   reassignment of the relays without forcing the sender to reconfigure
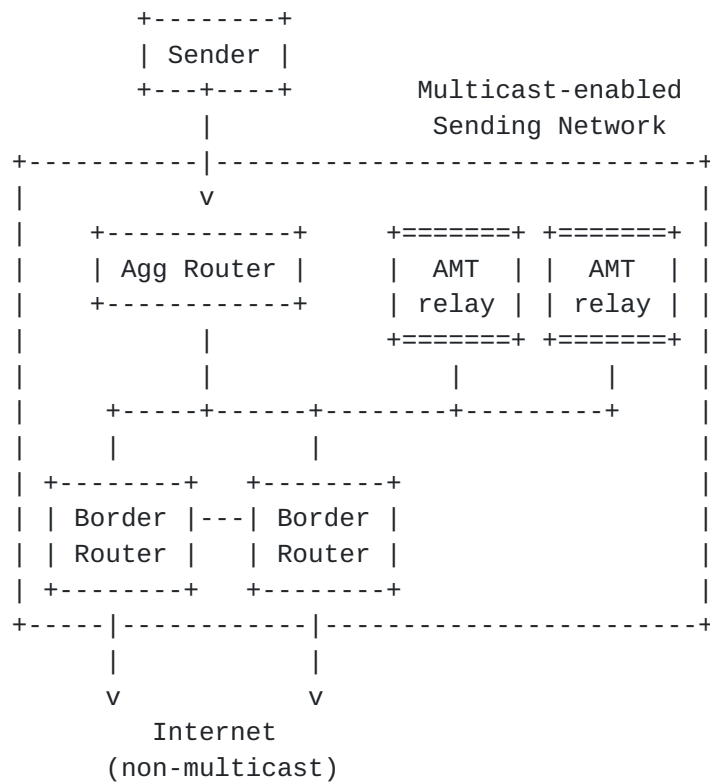   the corresponding AMTRELAY RRs.

```
                  +--------+
                  | Sender |
                  +---+----+          Multicast-enabled
                      |               Sending Network
            +-----------|----------------------------+
            |           v                             |
            |    +------------+    +======+ +======+  |
            |    | Agg Router |    | AMT  | | AMT  |  |
            |    +------------+    | relay| | relay|  |
            |           |         +======+ +======+  |
            |           |              |        |     |
            |     +-----+------+-------+--------+     |
            |     |            |                      |
            | +--------+   +--------+                 |
            | | Border |---| Border |                 |
            | | Router |   | Router |                 |
            | +--------+   +--------+                 |
            +-----|------------|---------------------+
                  |            |
                  v            v
                  Internet
               (non-multicast)
```

                   Figure 7: Sending ISP Example

[4](#). AMTRELAY Resource Record Definition
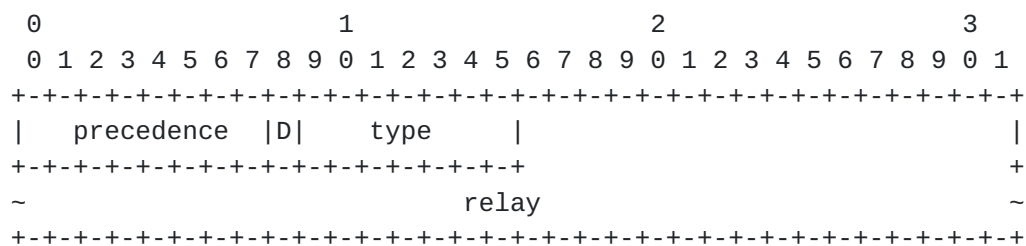
[4.1](#). AMTRELAY RRType

   The AMTRELAY RRType has the mnemonic AMTRELAY and type code 260
   (decimal).

   The AMTRELAY RR is class independent.

[4.2](#). AMTRELAY RData Format

   The AMTRELAY RData consists of a 8-bit precedence field, a 1-bit
   "Discovery Optional" field, a 7-bit type field, and a variable length
   relay field.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   precedence  |D|    type     |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
   ~                             relay                             ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.2.1. RData Format - Precedence

   This is an 8-bit precedence for this record. It is interpreted in
   the same way as the PREFERENCE field described in Section 3.3.9 of
   [RFC1035].

   Relays listed in AMTRELAY records with a lower value for precedence
   are to be attempted first.

4.2.2. RData Format - Discovery Optional (D-bit)

   The D bit is a "Discovery Optional" flag.

   If the D bit is set to 0, a gateway using this RR MUST perform AMT
   relay discovery as described in Section 4.2.1.1 of [RFC7450], rather
   than directly sending an AMT Request message to the relay.

   That is, the gateway MUST receive an AMT Relay Advertisement message
   (Section 5.1.2 of [RFC7450]) for an address before sending an AMT
   Request message (Section 5.1.3 of [RFC7450]) to that address. Before
   receiving the Relay Advertisement message, this record has only
   indicated that the address can be used for AMT relay discovery, not
   for a Request message. This is necessary for devices that are not
   fully functional AMT relays, but rather load balancers or brokers, as
   mentioned in Section 4.2.1.1 of [RFC7450].

   If the D bit is set to 1, the gateway MAY send an AMT Request message
   directly to the discovered relay address without first sending an AMT
   Discovery message.

   This bit should be set according to advice from the AMT relay
   operator. The D bit MUST be set to zero when no information is
   available from the AMT relay operator about its suitability.

4.2.3. RData Format - Type

   The type field indicates the format of the information that is stored
   in the relay field.

   The following values are defined:

   o type = 0: The relay field is empty (0 bytes).

   o type = 1: The relay field contains a 4-octet IPv4 address.

   o type = 2: The relay field contains a 16-octet IPv6 address.

   o type = 3: The relay field contains a wire-encoded domain name.
      The wire-encoded format is self-describing, so the length is
      implicit. The domain name MUST NOT be compressed. (See
      [Section 3.3 of [RFC1035]](...) and [Section 4 of [RFC3597]](...).)

4.2.4. RData Format - Relay

   The relay field is the address or domain name of the AMT relay. It
   is formatted according to the type field.

   When the type field is 0, the length of the relay field is 0, and it
   indicates that no AMT relay should be used for multicast traffic from
   this source.

   When the type field is 1, the length of the relay field is 4 octets,
   and a 32-bit IPv4 address is present. This is an IPv4 address as
   described in [Section 3.4.1 of [RFC1035]](...). This is a 32-bit number in
   network byte order.

   When the type field is 2, the length of the relay field is 16 octets,
   and a 128-bit IPv6 address is present. This is an IPv6 address as
   described in [Section 2.2 of [RFC3596]](...). This is a 128-bit number in
   network byte order.

   When the type field is 3, the relay field is a normal wire-encoded
   domain name, as described in [Section 3.3 of [RFC1035]](...). Compression
   MUST NOT be used, for the reasons given in [Section 4 of [RFC3597]](...).

   For a type 3 record, the D-bit and preference fields carry over to
   all A or AAAA records for the domain name. There is no difference in
   the result of the discovery process when it's obtained by type 1 or
   type 2 AMTRELAY records with identical D-bit and preference fields,
   vs. when the result is obtained by a type 3 AMTRELAY record that
   resolves to the same set of IPv4 and IPv6 addresses via A and AAAA
   lookups.

4.3. AMTRELAY Record Presentation Format

4.3.1. Representation of AMTRELAY RRs

   AMTRELAY RRs may appear in a zone data master file. The precedence,
   D-bit, relay type, and relay fields are REQUIRED.

   If the relay type field is 0, the relay field MUST be ".".

   The presentation for the record is as follows:

      IN AMTRELAY precedence D-bit type relay

4.3.2. Examples

   In a DNS authoritative nameserver that understands the AMTRELAY type,
   the zone might contain a set of entries like this:

```
   $ORIGIN 100.51.198.in-addr.arpa.
   10     IN AMTRELAY  10 0 1 203.0.113.15
   10     IN AMTRELAY  10 0 2 2001:DB8::15
   10     IN AMTRELAY 128 1 3 amtrelays.example.com.
```

   This configuration advertises an IPv4 discovery address, an IPv6
   discovery address, and a domain name for AMT relays which can receive
   traffic from the source 198.51.100.10. The IPv4 and IPv6 addresses
   are configured with a D-bit of 0 (meaning discovery is mandatory, as
   described in Section 4.2.2), and a precedence 10 (meaning they're
   preferred ahead of the last entry, which has precedence 128).

   For zone files in name servers that don't support the AMTRELAY RRType
   natively, it's possible to use the format for unknown RR types, as
   described in [RFC3597]. This approach would replace the AMTRELAY
   entries in the example above with the entries below:

```
   10   IN TYPE260  \# (
         6  ; length
         0a ; precedence=10
         01 ; D=0, relay type=1, an IPv4 address
         cb00710f ) ; 203.0.113.15
   10   IN TYPE260  \# (
         18 ; length
         0a ; precedence=10
         02 ; D=0, relay type=2, an IPv6 address
         20010db8000000000000000000000000f ) ; 2001:db8::15
   10   IN TYPE260  \# (
         24 ; length
         80 ; precedence=128
         83 ; D=1, relay type=3, a wire-encoded domain name
         09616d7472656c617973076578616d706c6503636f6d ) ; domain name
```

   See Appendix A for more details.

5. IANA Considerations

   This document updates the IANA Registry for DNS Resource Record Types
   by assigning type 260 to the AMTRELAY record.

   This document creates a new registry named "AMTRELAY Resource Record
   Parameters", with a sub-registry for the "Relay Type Field". The
   initial values in the sub-registry are:

```
           +-------+---------------------------------------+
           | Value | Description                           |
           +-------+---------------------------------------+
           |   0   | No relay is present.                  |
           |   1   | A 4-byte IPv4 address is present      |
           |   2   | A 16-byte IPv6 address is present     |
           |   3   | A wire-encoded domain name is present |
           | 4-255 | Unassigned                            |
           +-------+---------------------------------------+
```

Values 0, 1, 2, and 3 are further explained in Section 4.2.3 and
Section 4.2.4. Relay type numbers 4 through 255 can be assigned with
a policy of Specification Required (as described in [RFC8126]).

6. Security Considerations

[TBD: these 3 are just the first few most obvious issues, with just
sketches of the problem. Explain better, and look for trickier
issues.]

6.1. Record-spoofing

If AMT is used to ingest multicast traffic, providing a false
AMTRELAY record to a gateway using it for discovery can result in
Denial of Service, or artificial multicast traffic from a source
under an attacker's control.

Therefore, it is important to ensure that the AMTRELAY record is
authentic, with DNSSEC [RFC4033] or other operational safeguards that
can provide assurance of the authenticity of the record contents.

6.2. Local Override

The local relays, while important for overall network performance,
can't be secured by DNSSEC.

6.3. Congestion

Multicast traffic, particularly interdomain multicast traffic,
carries some congestion risks, as described in Section 4 of
[RFC8085]. Network operators are advised to take precautions
including monitoring of application traffic behavior, traffic
authentication, and rate-limiting of multicast traffic, in order to
ensure network health.

7. Acknowledgements

This specification was inspired by the previous work of Doug Nortz,
Robert Sayko, David Segelstein, and Percy Tarapore, presented in the
MBONED working group at IETF 93.

Thanks to Jeff Goldsmith, Toerless Eckert, Mikael Abrahamsson, Lenny
Giuliano, and Mark Andrews for their very helpful comments.

8. References

8.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities",
           STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987,
           <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and
           specification", STD 13, RFC 1035, DOI 10.17487/RFC1035,
           November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", BCP 14, RFC 2119,
           DOI 10.17487/RFC2119, March 1997,
           <https://www.rfc-editor.org/info/rfc2119>.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS
           Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997,
           <https://www.rfc-editor.org/info/rfc2181>.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A.
           Thyagarajan, "Internet Group Management Protocol, Version
           3", RFC 3376, DOI 10.17487/RFC3376, October 2002,
           <https://www.rfc-editor.org/info/rfc3376>.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi,
           "DNS Extensions to Support IP Version 6", STD 88,
           RFC 3596, DOI 10.17487/RFC3596, October 2003,
           <https://www.rfc-editor.org/info/rfc3596>.

[RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record
           (RR) Types", RFC 3597, DOI 10.17487/RFC3597, September
           2003, <https://www.rfc-editor.org/info/rfc3597>.

[RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener
           Discovery Version 2 (MLDv2) for IPv6", RFC 3810,
           DOI 10.17487/RFC3810, June 2004,
           <https://www.rfc-editor.org/info/rfc3810>.

   [RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet
              Group Management Protocol Version 3 (IGMPv3) and Multicast
              Listener Discovery Protocol Version 2 (MLDv2) for Source-
              Specific Multicast", RFC 4604, DOI 10.17487/RFC4604,
              August 2006, <https://www.rfc-editor.org/info/rfc4604>.

   [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for
              IP", RFC 4607, DOI 10.17487/RFC4607, August 2006,
              <https://www.rfc-editor.org/info/rfc4607>.

   [RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown,
              "Default Address Selection for Internet Protocol Version 6
              (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012,
              <https://www.rfc-editor.org/info/rfc6724>.

   [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service
              Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013,
              <https://www.rfc-editor.org/info/rfc6763>.

   [RFC7450] Bumgardner, G., "Automatic Multicast Tunneling", RFC 7450,
              DOI 10.17487/RFC7450, February 2015,
              <https://www.rfc-editor.org/info/rfc7450>.

   [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage
              Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085,
              March 2017, <https://www.rfc-editor.org/info/rfc8085>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

   [RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2:
              Better Connectivity Using Concurrency", RFC 8305,
              DOI 10.17487/RFC8305, December 2017,
              <https://www.rfc-editor.org/info/rfc8305>.

8.2. Informative References

   [RFC2317] Eidnes, H., de Groot, G., and P. Vixie, "Classless IN-
              ADDR.ARPA delegation", BCP 20, RFC 2317,
              DOI 10.17487/RFC2317, March 1998,
              <https://www.rfc-editor.org/info/rfc2317>.

   [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V.
              Jacobson, "RTP: A Transport Protocol for Real-Time
              Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550,
              July 2003, <https://www.rfc-editor.org/info/rfc3550>.

   [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S.
              Rose, "DNS Security Introduction and Requirements",
              RFC 4033, DOI 10.17487/RFC4033, March 2005,
              <https://www.rfc-editor.org/info/rfc4033>.

   [RFC5110] Savola, P., "Overview of the Internet Multicast Routing
              Architecture", RFC 5110, DOI 10.17487/RFC5110, January
              2008, <https://www.rfc-editor.org/info/rfc5110>.

   [RFC6726] Paila, T., Walsh, R., Luby, M., Roca, V., and R. Lehtonen,
              "FLUTE - File Delivery over Unidirectional Transport",
              RFC 6726, DOI 10.17487/RFC6726, November 2012,
              <https://www.rfc-editor.org/info/rfc6726>.

   [RFC7359] Gont, F., "Layer 3 Virtual Private Network (VPN) Tunnel
              Traffic Leakages in Dual-Stack Hosts/Networks", RFC 7359,
              DOI 10.17487/RFC7359, August 2014,
              <https://www.rfc-editor.org/info/rfc7359>.

   [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I.,
              Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent
              Multicast - Sparse Mode (PIM-SM): Protocol Specification
              (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March
              2016, <https://www.rfc-editor.org/info/rfc7761>.

   [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for
              Writing an IANA Considerations Section in RFCs", BCP 26,
              RFC 8126, DOI 10.17487/RFC8126, June 2017,
              <https://www.rfc-editor.org/info/rfc8126>.

   [RFC8313] Tarapore, P., Ed., Sayko, R., Shepherd, G., Eckert, T.,
              Ed., and R. Krishnan, "Use of Multicast across Inter-
              domain Peering Points", BCP 213, RFC 8313,
              DOI 10.17487/RFC8313, January 2018,
              <https://www.rfc-editor.org/info/rfc8313>.

   [RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS
              Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499,
              January 2019, <https://www.rfc-editor.org/info/rfc8499>.

Appendix A. Unknown RRType construction

   In a DNS resolver that understands the AMTRELAY type, the zone file
   might contain this line:

      IN AMTRELAY 128 0 3 amtrelays.example.com.

   In order to translate this example to appear as an unknown RRType as
   defined in [RFC3597], one could run the following program:

   <CODE BEGINS>
```
     $ cat translate.py
     #!/usr/bin/env python3
     import sys
     name=sys.argv[1]
     wire=''
     for dn in name.split('.'):
       if len(dn) > 0:
         wire += ('%02x' % len(dn))
         wire += (''.join('%02x'%ord(x) for x in dn))
     print(len(wire)//2) + 2
     print(wire)

     $ ./translate.py amtrelays.example.com
     24
     09616d7472656c617973076578616d706c6503636f6d
```
   <CODE ENDS>

   The length and the hex string for the domain name
   "amtrelays.example.com" are the outputs of this program, yielding a
   length of 22 and the above hex string.

   22 is the length of the wire-encoded domain name, so to this we add 2
   (1 for the precedence field and 1 for the combined D-bit and relay
   type fields) to get the full length of the RData, and encode the
   precedence, D-bit, and relay type fields as octets, as described in
   Section 4.

   This results in a zone file entry like this:

```
    IN TYPE260  \# ( 24 ; length
           80 ; precedence = 128
           03 ; D-bit=0, relay type=3 (wire-encoded domain name)
           09616d7472656c617973076578616d706c6503636f6d ) ; domain name
```

Author's Address

   Jake Holland
   Akamai Technologies, Inc.
   150 Broadway
   Cambridge, MA 02144
   United States of America

   Email: jakeholland.net@gmail.com