Mboned
Internet-Draft
Updates: 7450 (if approved)
Intended status: Standards Track
Expires: July 29, 2019

J. Holland
Akamai Technologies, Inc.

January 25, 2019

DNS Reverse IP AMT Discovery
draft-ietf-mboned-driad-amt-discovery-00

Abstract

This document updates RFC 7450 (AMT) by extending the relay discovery process to use a new DNS resource record for source-specific AMT relay discovery when joining source-specific multicast channels. A multicast sender configures a reverse IP DNS zone with the new AMTRELAY RR (defined in this document) to advertise a set of relays that can receive and forward multicast traffic from that sender inside a unicast AMT tunnel, in order to transit non-multicast-capable network segments.

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 July 29, 2019.

Table of Contents

1. Introduction

This document defines DNS Reverse IP AMT Discovery (DRIAD), a mechanism for AMT gateways to discover AMT relays which 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 relay selection might need to depend on the source of the multicast traffic, since a relay must be able to receive multicast traffic from the desired source in order to forward it.

That section suggests 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.

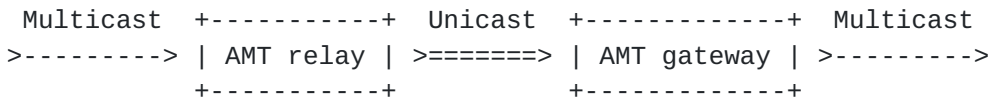


Figure 1: AMT Tunnel Illustration

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.
downstream	Further from the source of traffic.
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.

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 an AMT relay that can receive, encapsulate, and forward multicast traffic from a particular sender.

The sender is the owner of the RR, and configures the RR so that it contains the address or domain name of an AMT relay 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 2.3.

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.

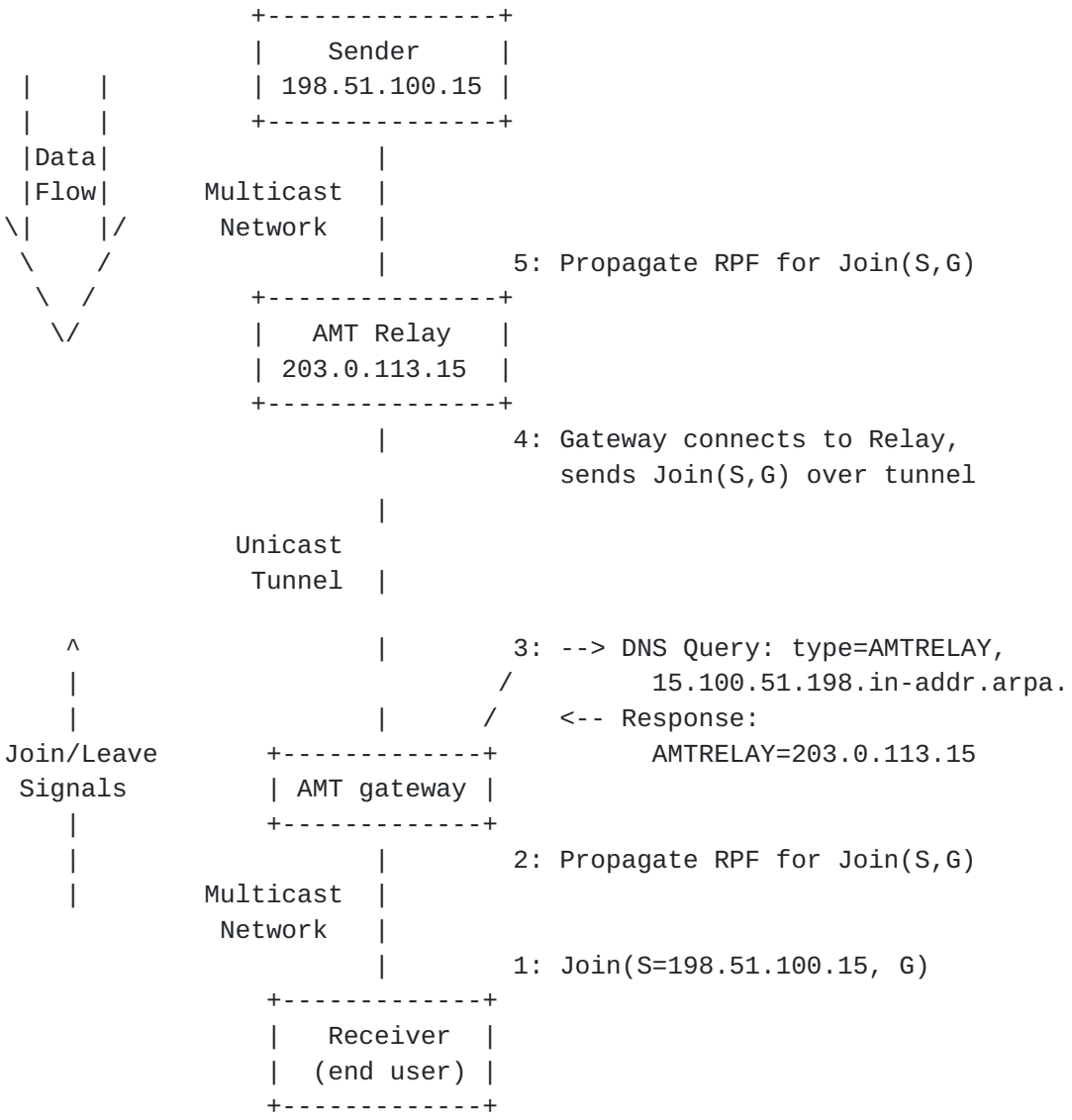


Figure 2: DRIAD Messaging

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. Optimal Relay Selection

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 IP 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)

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 the AMTRELAY RR defined in this document, gateways can use the DRIAD mechanism defined in this document to discover the relay information provided by the sender.

2.4. Guidelines for Restarting Discovery

2.4.1. 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).

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

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 events are copied here and numbered for easier reference:

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.

There are several new events that gateway heuristics may appropriately use to restart the discovery process, including:

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

2. When the DNS TTL expires for an AMTRELAY RR or for a domain name contained within the AMTRELAY RR.

3. 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.

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

This list is not exhaustive, nor are any of the listed events always strictly required to 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.

2.4.2. 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), but see Section 2.4.3 and Section 2.4.5 for more information about exceptions when it may be appropriate to use this event.

Likewise, some operators might use a short DNS TTL expiration (event #7) to allow for more responsive load balancing. If a gateway frequently sees short DNS TTLs (for example, under approximately 15 minutes) for some sources, a helpful heuristic may be to avoid restarting the discovery process for those sources, for example with an exponential backoff, or a hold-down timer that depends on the health or bit-rate of the active and subscribed traffic currently being forwarded through the tunnel.

2.4.3. Flow Health

In some gateway deployments, it is 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 clients, or monitoring 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.

2.4.4. Relay Loading and Shutdown

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. 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 MAY use heuristics such as this hold-down to override selection of a relay preferred by the precedence field in the AMTRELAY RR (see Section 4.2.1).

2.4.5. 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 2.4.2.

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.

2.4.6. Connecting to Multiple Relays

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.

2.5. 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.

2.6. 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).

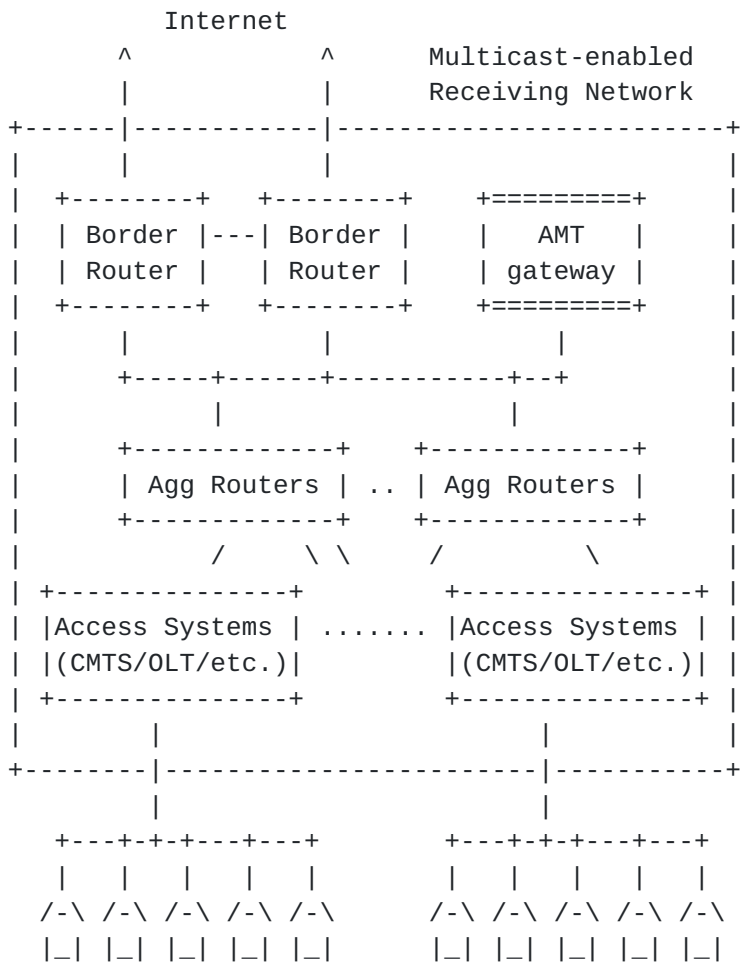


Figure 3: Receiving ISP Example

3.1.2. 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.)

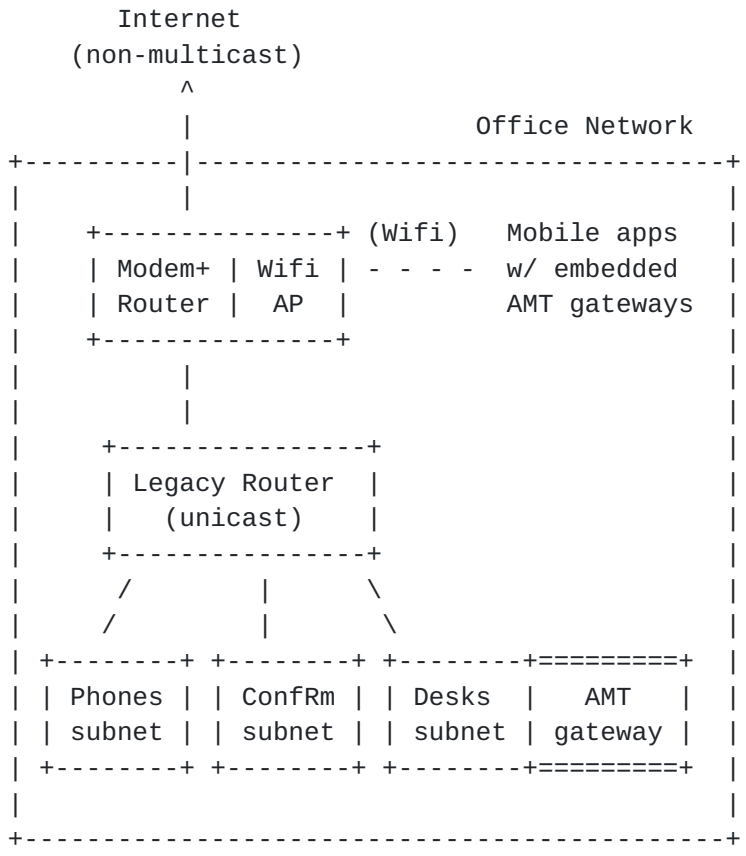


Figure 4: Small Office (no multicast up)

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.

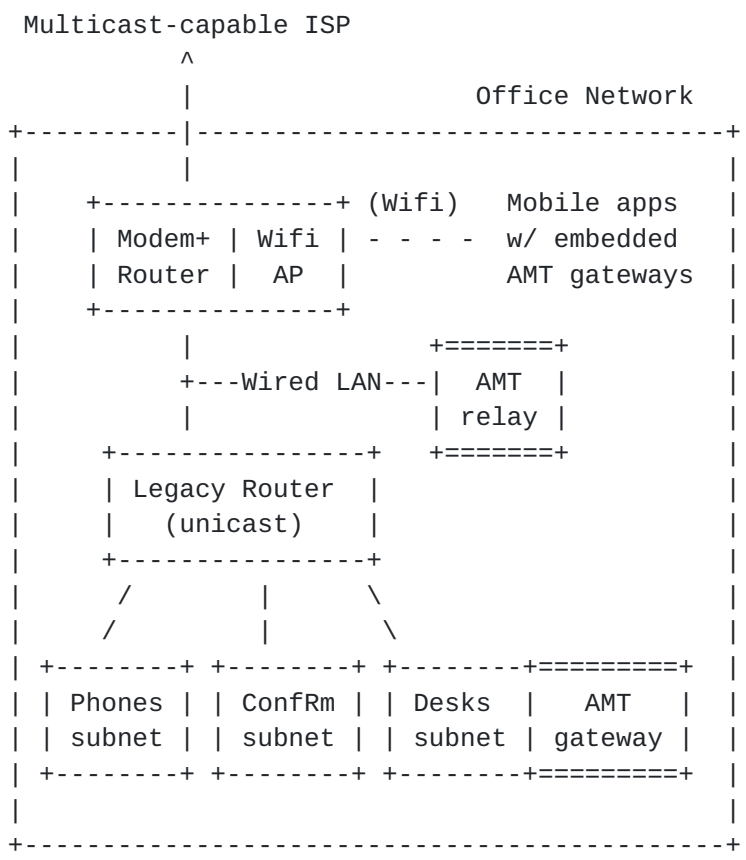


Figure 5: Small Office Example

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.

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).

It's also RECOMMENDED that when 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 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 discover the most-preferred relay first by DNS-SD, 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.

The discovery and connection process for multiple relays MAY operate in parallel, but when forwarding multicast group membership reports with new joins from an AMT gateway, membership reports SHOULD be forwarded to the most-preferred relays first, falling back to less preferred relays only after failing to receive traffic for an appropriate timeout, and only after reporting a leave to any more-preferred connected relays that have failed to subscribe to the traffic.

It is RECOMMENDED that the default timeout for receiving traffic be no less than 3 seconds, but the value MAY be overridden by administrative configuration, where known groups or channels need a different timeout for successful application performance.

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 an A or AAAA record from those domain names.

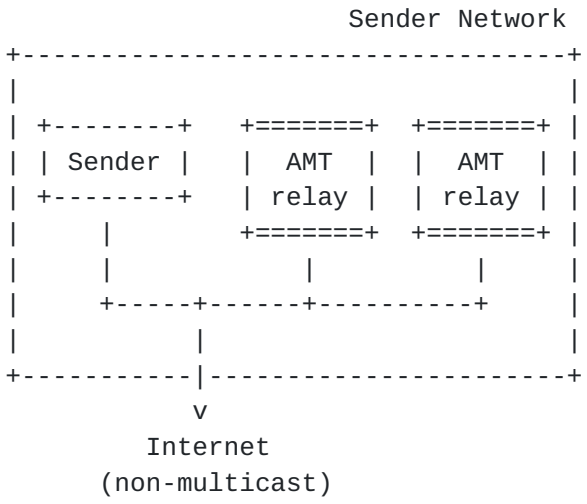


Figure 6: Small Office Example

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 a domain name for the AMT relays for use with the discovery process defined in this document.

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.

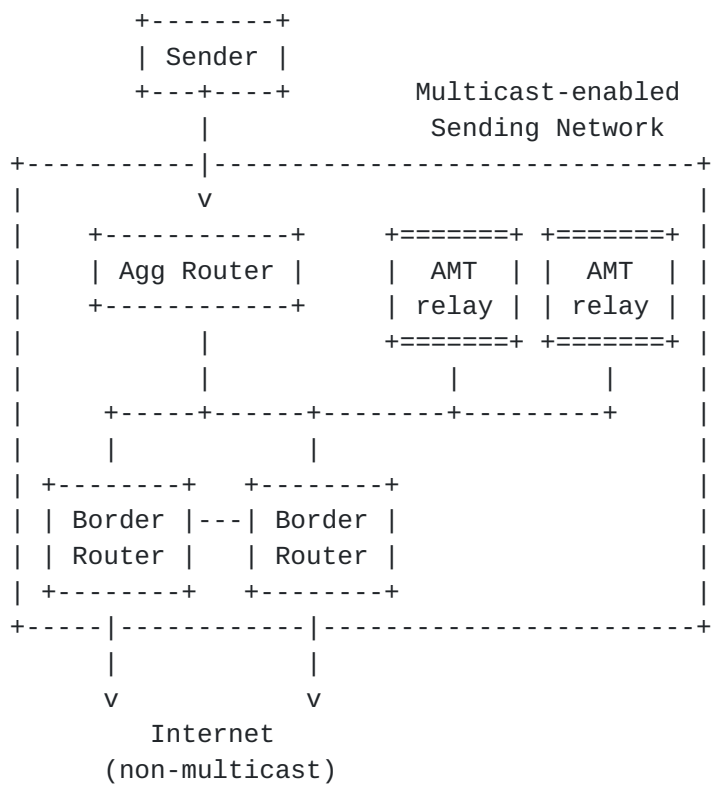


Figure 7: Sending ISP Example

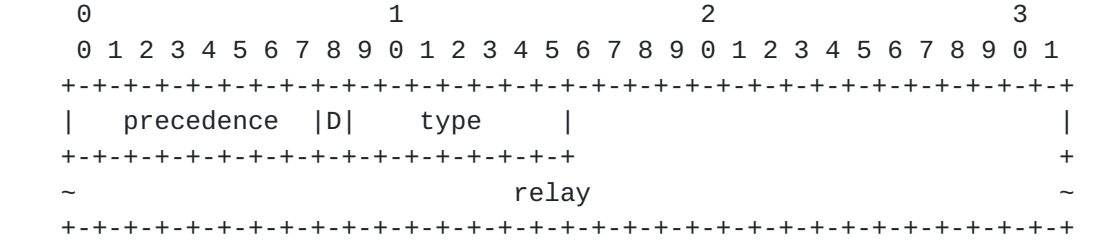
4. AMTRELAY Resource Record Definition

4.1. AMTRELAY RRType

The AMTRELAY RRType has the mnemonic AMTRELAY and type code TBD1 (decimal).

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.



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.

Where there is a tie in precedence, the default choice of relay MUST be non-deterministic, to support load balancing. The AMT gateway operator MAY override this default choice with explicit configuration when it's necessary for administrative purposes.

For example, one network might prefer to tunnel IPv6 multicast traffic over IPv6 AMT and IPv4 multicast traffic over IPv4 AMT to avoid routeability problems in IPv6 from affecting IPv4 traffic and vice versa, while another network might prefer to tunnel both kinds of traffic over IPv6 to reduce the IPv4 space used by its AMT gateways. In this example scenario or other cases where there is an administrative preference that requires explicit configuration, a receiving network MAY make systematically different precedence choices among records with the same precedence value.

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:

- type = 0: The relay field is empty (0 bytes).
- type = 1: The relay field contains a 4-octet IPv4 address.
- type = 2: The relay field contains a 16-octet IPv6 address.
- 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].

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:

[To be removed (TBD): replace 65280 with the IANA-assigned value TBD1, here and in Appendix B.]

```
10   IN TYPE65280  \# (
       6  ; length
       0a ; precedence=10
       01 ; D=0, relay type=1, an IPv4 address
       cb00710f ) ; 203.0.113.15
10   IN TYPE65280  \# (
       18 ; length
       0a ; precedence=10
       02 ; D=0, relay type=2, an IPv6 address
       20010db8000000000000000000000000f ) ; 2001:db8::15
10   IN TYPE65280  \# (
       24 ; length
       80 ; precedence=128
       83 ; D=1, relay type=3, a wire-encoded domain name
       09616d7472656c617973076578616d706c6503636f6d ) ; domain name
```

See Appendix B for more details.

5. IANA Considerations

This document updates the IANA Registry for DNS Resource Record Types by assigning type TBD1 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>.

[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>.

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>.

[RFC4025] Richardson, M., "A Method for Storing IPsec Keying Material in DNS", RFC 4025, DOI 10.17487/RFC4025, March 2005, <https://www.rfc-editor.org/info/rfc4025>.

[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>.

[RFC5507] IAB, Faltstrom, P., Ed., Austein, R., Ed., and P. Koch, Ed., "Design Choices When Expanding the DNS", RFC 5507, DOI 10.17487/RFC5507, April 2009, <https://www.rfc-editor.org/info/rfc5507>.

[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>.

[RFC6895] Eastlake 3rd, D., "Domain Name System (DNS) IANA Considerations", BCP 42, RFC 6895, DOI 10.17487/RFC6895, April 2013, <https://www.rfc-editor.org/info/rfc6895>.

[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. New RRType Request Form

This is the template for requesting a new RRType recommended in Appendix A of [RFC6895].

A. Submission Date:

B.1 Submission Type:
[x] New RRTYPE [ ] Modification to RRTYPE
B.2 Kind of RR:
[x] Data RR [ ] Meta-RR

C. Contact Information for submitter (will be publicly posted):
Name: Jake Holland
Email Address: jakeholland.net@gmail.com
International telephone number: +1-626-486-3706
Other contact handles: jholland@akamai.com

D. Motivation for the new RRTYPE application.
It provides a bootstrap so AMT (RFC 7450) gateways can discover an AMT relay that can receive multicast traffic from a specific source, in order to signal multicast group membership and receive multicast traffic over a unicast tunnel using AMT.

E. Description of the proposed RR type.
This description can be provided in-line in the template, as an attachment, or with a publicly available URL.
Please see draft-ietf-mboned-driad-amt-discovery.

F. What existing RRTYPE or RRTYPEs come closest to filling that need and why are they unsatisfactory?
Some similar concepts appear in IPSECKEY, as described in Section 1.2 of [RFC4025]. The IPSECKEY RRType is unsatisfactory because it refers to IPSec Keys instead of to AMT relays, but the motivating considerations for using reverse IP and for providing a precedence are similar--an AMT gateway often has access to a source address for a multicast (S,G), but does not have access to a relay address that can receive multicast traffic from the source, without administrative configuration.

Defining a format for a TXT record could serve the need for AMT relay discovery semantics, but Section 5 of [RFC5507] provides a compelling argument for requesting a new RRType instead.

G. What mnemonic is requested for the new RRTYPE (optional)?
AMTRELAY

H. Does the requested RRTYPE make use of any existing IANA registry or require the creation of a new IANA subregistry in DNS Parameters?
Yes, IANA is requested to create a subregistry named "AMT Relay Type Field" in a "AMTRELAY Resource Record Parameters" registry. The field values are defined in Section 4.2.3 and Section 4.2.4, and a summary table is given in Section 5.

I. Does the proposal require/expect any changes in DNS servers/resolvers that prevent the new type from being processed as an unknown RRTYPE (see RFC3597)?
No.

J. Comments:
It may be worth noting that the gateway type field from Section 2.3 of [RFC4025] and Section 2.5 of [RFC4025] is very similar to the Relay Type field in this request. I tentatively assume that trying to re-use that sub-registry is a worse idea than duplicating it, but I'll invite others to consider the question and voice an opinion, in case there is a different consensus.
https://www.ietf.org/assignments/ipseckey-rr-parameters/ipseckey-rr-parameters.xml

Appendix B. 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)
  print(wire)

  $ ./translate.py amtrelays.example.com
  22
  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.

This results in a zone file entry like this:

```
 IN TYPE65280  \# ( 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