Network Working Group S. Cheshire
Internet-Draft Apple Inc.
Intended status: Standards Track T. Lemon
Expires: January 4, 2018 Nominum, Inc.
July 3, 2017

Multicast DNS Discovery Relay
draft-sctl-dnssd-mdns-relay-00

Abstract

This document extends the Discovery Proxy for Multicast DNS-Based Service Discovery specification. It describes a lightweight relay mechanism, a Discovery Relay, which allows Discovery Proxies to provide service on links to which the hosts on which they are running are not directly attached.

Status of This Memo

This Internet-Draft will expire on January 4, 2018.

Copyright Notice

Table of Contents

1. Introduction

The Discovery Proxy for Multicast DNS-Based Service Discovery [I-D.ietf-dnssd-hybrid] specification defines a mechanism for discovering services on a subnetted network using Multicast DNS (mDNS) [RFC6762], through the use of Discovery Proxies, which issue mDNS requests on various links in the network on behalf of a host attempting service discovery.

In the original Discovery Proxy specification, it is assumed that for every link on which services will be discovered, a host will be present running a full Discovery Proxy. This document introduces a lightweight Discovery Relay which can be used to provide discovery services on a link without requiring a full Discovery Proxy on every link.

The Discovery Relay operates by listening for TCP connections from Discovery Proxies. When a Discovery Proxy conneects, the connection is authenticated and secured using TLS. The Discovery Proxy can then send messages that will be relayed to specified links. The Discovery Proxy may also specify one or more links from which it wishes to receive mDNS traffic. DNS Session Signaling [I-D.ietf-dnsop-session-signal] is used as a framework for conveying interface and IP header information associated with each message.

The Discovery Relay functions essentially as a set of one or more virtual interfaces for the Discovery proxy, one on each link to which the Discovery Relay is connected. In a complex network, it is possible that more than one Discovery Relay will be connected to the same link; in this case, the Discovery Proxy ideally should only be

using one such Relay Proxy per link, since using more than one will generate duplicate traffic.

How such duplication is detected and avoided is out of scope for this document: in principle it could be detected using HNCP [RFC7788] or configured using some sort of orchestration software in conjunction with NETCONF [RFC6241] or CPE WAN Management Protocol [TR-069].

2. Terminology

The following definitions may be of use:

mDNS Agent A host which sends and/or responds to mDNS queries.

Discovery Proxy A network service which receives well-formed questions using the DNS protocol, performs multicast DNS queries to answer those questions, and responds with those answers using the DNS protocol.

Discovery Relay A network service which sends mDNS messages on behalf of a Discovery Proxy and relays mDNS messages to a Discovery Relay.

link A maximal set of network connection points such that any host connected to any connection point may send a packet to a host connected to any other connection point without the help of a layer 3 router.

whitelist A list of one or more IP addresses from which a Discovery Relay may accept connections.

silently discard When a message that is not supported or not permitted is received, and the required response to that message is to "silently discard" it, that means that no response is sent by the service that is discarding the message to the service that sent it. The service receiving the message may log the event, and may also count such events: "silently" does not preclude such behavior.

Director A central or coordinated controlling function in an orchestrated network of Discovery Proxies and Discovery Relays (Section 4.1).

Performer The interface through which the Director directs the behavior of Discovery Proxies and Discovery Relays (Section 4.1).

3. Protocol Overview

This document describes a way for Discovery Proxies to communicate with mDNS agents on networks to which they are not directly connected using a Discovery Relay. As such, there are two parts to the protocol: connections between Discovery Proxies and Discovery Relays, and communications between Discovery Relays and mDNS agents.

3.1. Connections between Discovery and Discovery Relays

Discovery Relays listen for connections. Connections between Discovery Proxies and Discovery Relays are established by Discovery Proxies. Connections are authenticated and encrypted using TLS, with both client and server certificates. Connections are long-lived: a Discovery Proxy is expected to send many queries over the same connection, and Discovery Relays will forward all mDNS traffic from subscribed interfaces over the connection.

The stream encapsulated in TLS will carry DNS frames as in the DNS TCP protocol [RFC1035] Section 4.2.2. However, all messages will be DNS Session Signaling messages [I-D.ietf-dnsop-session-signal]. There will be three types of such messages:

- Subscribe messages from Discovery Proxy to Discovery Relay

- mDNS messages from Discovery Proxy to Discovery Relay

- mDNS messages from Discovery Relay to Discovery Proxy

Subscribe messages from the Discovery Proxy to the Discovery Relay indicate to the Discovery Relay that mDNS messages from one or more specified links are to be relayed to the Discovery Proxy.

mDNS messages from a Discovery Proxy to a Discovery Relay cause the Discovery Relay to re-transmit the mDNS message on one or more links to which the Discovery Relay host is directly attached.

mDNS messages from a Discovery Relay to a Discovery Proxy are sent whenever an mDNS message is received on a link to which the Discovery Relay has subscribed.

Discovery Relays are responsible for keeping connections alive when no traffic has been sent during a keepalive period [I-D.ietf-dnsop-session-signal] Section 4.

3.2. mDNS Messages On Links

Discovery Relays listen for mDNS traffic on all configured links. When a mDNS message is received on a link, it is forwarded on every open Discovery Proxy connection that is subscribed to mDNS traffic on that link. In the event of congestion, where a particular Discovery Proxy connection has no buffer space for an mDNS message that would otherwise be forwarded to it, the mDNS message is not forwarded to it. Normal mDNS retry behavior is used to recover from this sort of packet loss. Discovery Relays are not expected to buffer more than a few mDNS packets.

Discovery Relays accept mDNS traffic from Discovery Proxies. Such traffic is forwarded to zero or more more links to which the Discovery Relay host is directly connected.

4. Orchestration

In order for one or more Discovery Proxies to make use of one or more Discovery Relays to provide service discovery on one or more links, the set of links on which service will be provided must be known, the set of Discovery Relays for those links must be known, and the set of Discovery Proxies allowed to connect to those Discovery Relays must be known. We assume that this information is maintained in some sort of orchestration system.

Although it is of course possible to configure such an environment with a set of static configuration files, it is most useful to consider such a network to be dynamic, with links potentially being added and removed, Discovery Proxies being added and removed, and Discovery Relays being added and removed. This document takes no position on which specific orchestration system will be used, but does specify the inputs and outputs of such a system that will be required for successful operation. In the case of static configuration, these inputs and outputs are also the same; the only difference is that they do not change without human intervention.

It is not strictly necessary that all participants in the orchestration process have complete information. It may be desirable for example to have more than one Discovery Proxy managed by an orchestration system, but to have different Discovery Proxies support different links. The set of primitives described here can be used to implement configurations where multiple Discovery Proxies are present and supporting disjoint, overlapping or identical sets of links.

There is a special case of orchestration that may be desirable in some settings: when a node may need to be capable of providing either Discovery Proxy service or Discovery Relay service, and is configured

to provide Discovery Proxy service, it would be useful to have a way to automatically configure the Discovery Relay to use the Discovery Proxy just on that one node, without requiring a network-wide orchestration system. In the case of a node that supports orchestration through HNCP, however, this is unnecessary: HNCP will work to provide orchestration even on a single node.

4.1. Orchestration System Functional Overview

Conceptually, the orchestration system has two parts: the part that manages the network, and the part an instance of which is present on each node in the network that is orchestrated by the system. In a cooperative system such as HNCP [RFC7788], orchestration is done cooperatively, and the two functions are present on every participating node. In a managed system using NETCONF [RFC6241], a central service pushes configuration information to managed nodes, and pulls status information from managed nodes. For this discussion, which of these models is used (or whether some other model is used) is immaterial. The functional division is the same in either case: conceptually there is one function that does the orchestration called the Director, and there are one more more functions to which the orchestration applies, called Performers.

The Director is receptive to primitives from Performers. Performers apply primitives announced to them by the Director, and announce primitives to the Director. The Director announces primitives to Performers, based on its operating model and its configuration, based either in changes to the network or to announcements from Performers.

It is permissible for nodes to provide both Discovery Proxy and Discovery Relay service at the same time. In this case, there is a further conceptual functional division: on such a node, there are two Performers: the Discovery Proxy Performer and the Discovery Relay Performer. These may be the same program, or they may be functionally separate; which is the case is beyond the scope of this document. The reason for making this distinction is to point out that on a node providing both services, both Performers may receive every announcement sent by the Director. And of course the Director receives announcements sent by either Performer.

4.2. Orchestration Primitives

4.2.1. Link Present

The 'Link Present' primitive is used by the Director to communicate the presence of a link to Performers. 'Link Present' primitives include the following data:

link identifier One or more opaque 32-bit identifiers, each of which identifies a link that is present on the orchestrated network. Each identifier is unique among all link identifiers managed by the Director. These link identifiers are used in the Discovery Relay protocol to identify links on which mDNS requests will be sent and received, and are consistent across all participants in the orchestration system.

4.2.2. Link Remove

The 'Link Remove' primitive is used by the Director to communicate to Performers that a link that was formerly present is no longer present. The 'Link Remove' primitive includes the following data:

link identifier One or more opaque 32-bit identifiers, as described in Section 4.2.1.

4.2.3. Discovery Proxy Available

The 'Discovery Proxy Available' primitive is used by Discovery Proxy Performers to announce their availability to the Director, and by the Director to announce to Discovery Relay Performers that Discovery Proxies are present and enabled. This primitive is only used for nodes that provide Discovery Proxy service and can use Discovery Relays: a Discovery Proxy that does not support Discovery Proxy service is never announced in this way. The 'Discovery Proxy Available' primitive includes the following data:

node identifier The node identifier of the Discovery Proxy, unique among all nodes managed by a Director.

IP addresses One or more IP addresses configured on the network interfaces of the node making the announcement. This list must include all IP addresses from which the Discovery Proxy might connect to Discovery Relays, but need not include any other IP addresses.

TLS Certificate A TLS PKI certificate or bare public key which will be used by the Discovery Proxy to authenticate itself when connecting to Discovery Relays.

4.2.4. Discovery Proxy Resigning

The 'Discovery Proxy Resigning' primitive is used by Discovery Proxies to announce to the Director that they are no longer available, and by the Director to announce to Discovery Relay performers that a Discovery Proxy is no longer present or enabled.

The 'Discovery Proxy Resigning' primitive includes the following data:

The node identifier of the Discovery Proxy, unique among all nodes managed by a Director.

4.2.5. Discovery Relay Available

The 'Discovery Relay Available' primitive is used by Discovery Relay Performers to inform the Director that they are available to provide service. It is used by the Director to announce to Discovery Proxy Performers that a Discovery Relay is available and enabled. The 'Discovery Relay Available' primitive includes the following data:

node identifier The node identifier of the Discovery Relay, unique among all nodes managed by a Director.

IP addresses A list of IP addresses on which the Discovery Relay may be contacted.

Port TCP Port on which the Discovery Relay will be listening for connections.

Server Certificate A TLS PKI certificate or bare public key which will be presented to Discovery Proxies when they initiate TLS connections with the Discovery Relay. This is used both to authenticate the Discovery Relay, and also to establish an encrypted connection between the two services.

Links A list of links on which the Discovery Relay provides service. Each link identifier corresponds to a link identified by a previous 'Link Present' primitive sent by the Director, as described in Section 4.2.1.

4.2.6. Discovery Relay Resigning

When a node providing Discovery Relay support can no longer continue to do so, it announces to the Director that it is no longer available using this primitive. The 'Discovery Relay Resigning' primitive includes the following data:

node identifier The node identifier of the Discovery Relay, unique among all nodes managed by a Director.

4.3. Orchestration System Behavior

4.3.1. Link Present

The Director detects new links, or is configured with new links by the network operator. It is responsible for noticing that a link to which more than one participating node is connected is the same link. For example, see Section 6.1 of [RFC7788]. When a new link is detected, the Director reports the presence of that link to all enabled Discovery Proxy Performers, and to all Discovery Relay Performers. If the Director becomes aware of more than one link at the same time, or within an implementation-specific interval, it may announce the presence of more than one link at a time using the 'Link Present' primitive.

4.3.2. Link Remove

The Director detects the removal of links, either as a result of routers that are connected to those links becoming unavailable, or as a result of manual changes to the configuration by the network operator. When a link that had previously been present is removed, the Director announces the removal of this link to all enabled Discovery Proxy performers and to all Discovery Relay performers. If the removal of more than one link is detected at the same time or within an implementation-specific interval, the removal of each such link may be announced in a single 'Link Remove' primitive.

4.3.3. Discovery Proxy Available

When the Director receives a 'Discovery Proxy Available' primitive, it records the information in its list of available Discovery Proxies (henceforth "Discovery Proxy List"). If that node had previously reported that Discovery Proxy service was available, the entry in Discovery Proxy List for that node is replaced with an entry generated from the new update; any information in the previous entry that is not present in the update is discarded.

Whether or not the Director enables Discovery Proxy service on the Discovery Proxy announced in a newly-received 'Discovery Proxy Available' primitive is dependent on the operational model and configuration of that particular orchestration system, which is out of scope for this document. The same is true as to whether service discovery is enabled on all known links, or not. We assume here that Discovery Proxy service may be available but not enabled on some nodes, whereas Discovery Relay service is generally available, since it will only be used by enabled Discovery Proxies on interfaces on which service discovery is enabled.

If the Director enables Discovery Proxy service on that node, the Discovery Proxy is announced to all nodes currently providing Discovery Relay service, using 'Discovery Proxy Available' primitives. In addition, the set of all known Discovery Relays, and the information provided by them to the orchestration system, is announced to the node providing the Discovery Proxy service, using one or more 'Discovery Relay Available' primitives.

When a 'Discovery Proxy Available' primitive is received from a Discovery Proxy Performer for which service is already enabled, but the update includes different information than was present in the previous announcement, the Discovery Proxy service is re-announced to every Discovery Relay Performer.

4.3.4. Discovery Proxy Resigning

When the Director receives a 'Discovery Proxy Resigning' primitive from a Discovery Proxy Performer that had previously sent a 'Discovery Proxy available' primitive, the Director first determines if Discovery Proxy service had been enabled on that node. If so, 'Discovery Proxy Resigning' notifications are sent to Discovery Relay Performers.

The Director may, as a result of a node's resignation from providing Discovery Proxy service, enable Discovery Proxy on some other node. If so, it does so as described in Section 4.3.3.

In addition to any announcements sent as a result of a node's resignation from providing Discovery Proxy service, the Director also looks for an entry in the Discovery Proxy List for that node. If one is present, it is removed.

4.3.5. Discovery Relay Available

When the Director receives a 'Discovery Relay Available' primitive, it records the information in its list of available Discovery Relay Performers (henceforth "Discovery Relay List"). If that list already contains an entry for the Performer making the new report, the entry from the list is discarded and a new one generated from the new announcement.

Whether or not the Director enables service discovery through a particular Discovery Relay is dependent on the operation of that particular orchestration system, which is out of scope for this document. It is assumed that a Director may or may not enable a particular Discovery Relay.

If the Director enables service discovery through the relay that made the announcement, the relay is announced to all enabled Discovery Proxy Performers. In addition, if the relay had not previously been enabled for service discovery, the Director sends a 'Discovery Proxy Available' primitive to that Performer for each Discovery Proxy Performer on the Discovery Proxy List.

4.3.6. Discovery Relay Resigning

When the Director receives a 'Discovery Relay Resigning' primitive, it checks to see if the node making the announcement had previously been listed as providing Discovery Relay service; if so, the entry for that node is removed from the list. If Discovery Relay service was enabled for that node, all nodes providing Discovery Proxy service are notified that this node is no longer providing Discovery Relay service, by sending a 'Discovery Relay Resigning' primitive to each such node.

4.3.7. Node Available

The orchestration system may or may not track the coming and going of nodes that provide service discovery. If it does, depending on the operation of the system, it may be necessary to send some notification to the node to trigger its announcement of service discovery services. How this is done is out of scope for this document.

4.3.8. Node Resigning

The orchestration system may or may not track the coming and going of nodes that provide service discovery. If it does, then when the departure of a node that has previously announced Discovery Relay and/or Discovery Proxy service should result in the synthesis of resignation events for those services on that node. The exact operation of this mechanism is out of scope for this document.

4.4. Discovery Proxy Performer Behavior

Nodes may provide both Discovery Proxy and Discovery Relay service: the two services share no ports and are mutually compatible. When a node is providing both services, the behaviors described in this section are specific to the operation of the Discovery Proxy service on that node, not to the Discovery Relay service.

4.4.1. Link Present

When a node that is providing Discovery Proxy service receives a link present notification, it checks to see if it currently has Discovery Relay service configured for each such link. For any such link for which it does not have Discovery Relay service configured, it identifies the set of Relay Proxies that provide service on that link. It then chooses a Discovery Relay node from this set using a random number generator. If it already has a connection to the Relay Proxy, it attempts to subscribe to mDNS messages from that link. If it does not have a connection, it attempts to establish one. If that succeeds, it attempts to subscribe to mDNS messages from that link. If the outcome of each of these attempts to get Discovery Relay service on the new link fails, it eliminates this Discovery Relay from the set and repeats the process until the set is empty.

If no attempt to subscribe to mDNS messages on the link is successful, then service discovery on that link is not possible. The Discovery Proxy node maintains a list of links on which Discovery Relay service is desired but not available; when an attempt to get Discovery Relay service on a link fails, either because no node is providing Discovery Relay service on that link, or because attempting to get service on that link from all nodes claiming to provide it has failed, the link is added to this list.

4.4.2. Link Removed

When a link is removed, the Discovery Relay checks its list of connections to Discovery Relays for subscription for mDNS messages on that link. If one is present, the Discovery Relay unsubscribes from mDNS messages on that link. If there are no subscriptions present on that connection, the Discovery Relay terminates the connection. If the link is on the list of links for which Discovery Relay service is desired but not available, the link is removed from that list.

4.4.3. Discovery Proxy Available

Discovery Proxy Performers send 'Discovery Proxy Available' primitives to the Director whenever their configuration changes in a way that affects the content of the primitive, and also whenever their node becomes newly available to the Director. In addition to notifying the Director when they first become connected to the Director's orchestration system, they must also notify the Director when they disconnect and reconnect.

When a node with Discovery Proxy service becomes available to the orchestration system, it informs the orchestration system that it can provide Discovery Proxy service. It also provides the orchestration

system with a list of IP addresses from which it may originate connections to Discovery Relays, and provides a TLS PKI cert or suitable bare public key which will be used for TLS Client Authentication.

Whenever the set of IP addresses from which the Discovery Proxy may initiate a connection to a Discovery Relay changes, the Discovery Proxy sends a new 'Discovery Proxy Available' primitive with its complete information, as above. It may be desirable for the Discovery Proxy node to choose a specific IP address from which all such connections will originate, so as to minimize the number of such updates that may be required, but this behavior is optional.

It is not ordinarily the case that the key or certificate used for authentication will change, but if it does, the Discovery Proxy node sends a complete new 'Discovery Proxy Available' primitive, which will contain the new key or certificate.

4.4.4. Discovery Proxy Resigning

When a node that had previously provided Discovery Proxy service is no longer able to do so for any reason, it announces this to the orchestration system using a 'Discovery Proxy Resigning' primitive.

4.4.5. Discovery Relay Available

When a node providing Discovery Proxy service receives a 'Discovery Relay Available' notification, it adds that Discovery Relay to its list of available Discovery Relays. If the Discovery Relay is already on the list, the information the list entry is compared to the new information provided in the 'Discovery Relay Available' primitive. If a connection to that Discovery Relay is present, and the destination IP address of that connection is no longer on the list of IP addresses supported by the Discovery Relay, or the public key of the Discovery Relay has changed, the connection is dropped and the process described in Section 4.4.6 is followed.

Otherwise, if there is a connection to the Discovery Relay, the list of links subscribed to on that connection is compared to the list of served links listed in the 'Discovery Relay Available' primitive; any links for which subscriptions exist that are not listed in the 'Discovery Relay Available' announcement are unsubscribed, and those links added to the list of links on which Discovery Relay service is not available.

At this point the process described in Section 4.4.1 is followed for each link on the list of links for which Discovery Relay service is not available.

4.4.6. Discovery Relay Resigning

Discovery Relay drops its connection to that Discovery Relay and puts all links for which subscriptions existed on that connection onto the list of links on which Discovery Relay service is not available. Because it is possible that another Discovery Relay is available for that link, the Discovery Proxy node again follows the process described in Section 4.4.1.

4.5. Discovery Relay Performer Behavior

Nodes that support service discovery may support both Discovery Proxy and Discovery Relay. Behaviors described here are specific to nodes that are providing Discovery Relay service. A node that provides both types of service will follow both the behavior described here and the behavior described for Discovery Proxy nodes.

4.5.1. Link Present

When a Discovery Relay performer receives a link present notification, it determines for each link announced whether it has an interface that is directly connected to that link. If so, it determines whether it has previously announced the availability of service on that link. If not, it adds the link to the list of links on which it provides Discovery Relay service (henceforth "Discovery Relay link list").

If as a result of a 'Link Present' announcement the Discovery Relay link list has changed, the Discovery Relay performer sends a new 'Discovery Relay Available' primitive to the Director.

4.5.2. Link Removed

When the Discovery Relay Performer receives a 'Link Removed' primitive, for each link mentioned in the primitive it checks to see if it is currently providing service on that link. For each link mentioned in the primitive for which it is providing service, it deletes that link from its list of links on which it is providing service. If any links were deleted from the list, the Discovery Relay Performer sends a new 'Discovery Relay Available' message to the Director.

4.5.3. Discovery Proxy Available

Directors send 'Discovery Proxy Available' primitives to Discovery Relay Performers when new Discovery Proxy Performers announce their availability, and also when Discovery Proxy Performers announce changes to their configuration. When a Discovery Relay Performer

receives one of these primitives, it updates its Discovery Proxy IP address whitelist with the set of IP addresses from the primitive, and updates the Discovery Proxy authentication certificate as well. If the Discovery Proxy is connected to the Discovery Relay and either the certificate changed, or the source IP address of the connection is no longer on the whitelist, the Discovery Relay drops the connection.

4.5.4. Discovery Proxy Resigning

Directors send 'Discovery Proxy Resigning' messages to Discovery Relay Performers when Discovery Proxy Performers indicate that they are no longer available, or when they are disabled by the orchestration system. When a Discovery Relay Performer receives this primitive, it checks to see if any connections from that Discovery Proxy are present. Any such connections are terminated.

4.5.5. Discovery Relay Available

Discovery Relay Performers send 'Discovery Relay Available' primitives to the Director whenever their configuration changes in a way that affects the content of the primitive, and also whenever their node becomes newly available to the Director. In addition to notifying the Director when they first become connected to the Director's orchestration system, they must also notify the Director when they disconnect and reconnect.

Discovery Relays listen for connections from Discovery Proxies. Because no port is reserved for Discovery Relays, it is not useful to announce the availability of the service until the service is listening for connections, at which point it will know which port it is listening on. Therefore, before sending a 'Discovery Relay Available' primitive, a Discovery Relay Performer must have received its listening port from the Discovery Relay service.

4.5.6. Discovery Relay Resigning

When a node providing Discovery Relay service must stop providing that service, it sends a 'Discovery Relay Resigning' primitive to the Director.

5. Connections between Discovery Proxies and Discovery Relays

When a Discovery Relay starts, it opens a passive TCP listener to receive connections from Discovery Proxies. This listener may be bound to one or more source IP addresses, or to the wildcard address, depending on the TCP implementation. When a connection is received, the relay must first validate that it is a connection to an IP

address to which connections are allowed. For example, it may be that only connections to ULAs are allowed, or to the IP addresses configured on certain interfaces. If the listener is bound to a specific IP address, this check is unnecessary.

The relay must then validate that the source IP address of the connection is on its whitelist. If the connection is not permitted either because of the source address or the destination address, the Discovery Relay responds to the TLS Client Hello message from the Discovery Proxy with a TLS user_canceled alert ([I-D.ietf-tls-tls13] Section 6.1).

Otherwise, the Discovery Relay will attempt to complete a TLS handshake with the Discovery Proxy. Discovery Proxies are required to send the post_handshake_auth extension ([I-D.ietf-tls-tls13] Section 4.2.5). If a relay proxy receives a ClientHello message with no post_handshake_auth extension, the Discovery Relay rejects the connection with a certificate_required alert ([I-D.ietf-tls-tls13] Section 6.2).

Once the TLS handshake is complete, the Discovery Relay MUST request post-handshake authentication as described in ([I-D.ietf-tls-tls13] Section 4.6.2). If the Discovery Proxy refuses to send a certificate, or the key presented does not match the key associated with the IP address from which the connection originated, or the CertificateVerify does not validate, the connection is dropped with the TLS access_denied alert ([I-D.ietf-tls-tls13] Section 6.2).

Once the connection is established and authenticated, it is treated as a DNS TCP connection [RFC1035].

Aliveness of connections between Discovery Proxies and Relays is maintained as described in Section 4 of [I-D.ietf-dnsop-session-signal]. Discovery Proxies must also honor the 'Retry Delay' TLV (section 5 of [I-D.ietf-dnsop-session-signal]) if sent by the Discovery Relay.

Discovery Proxies may establish more than one connection to a specific Discovery Relay. This would happen in the case that a TCP connection stalls, and the Discovery Proxy is able to reconnect before the previous connection has timed out. It could also happen as a result of a server restart. It is not likely that two active connections from the same Discovery Proxy would be present at the same time, but it must be possible for additional connections to be established. The Discovery Relay may drop the old connection when the new one has been fully established, including a successful TLS handshake. What it means for two connections to be from the same Discovery Proxy is that the connections both have source addresses

that belong to the same proxy, and that they were authenticated using the same client certificate.

6. Traffic from Relays to Proxies

The mere act of connecting to a Discovery Relay does not result in any mDNS traffic being forwarded. In order to request that mDNS traffic from a particular link be forwarded on a particular connection, the Discovery Proxy must send a session signaling message containing one or more MDNS Link Request TLVs (Section 9.1) indicating the link from which traffic is requested.

When such a message is received, the Discovery Relay validates that each specified link is available for forwarding, and that forwarding is enabled for that link. For each such message the Discovery Relay validates each link specified and includes in a single response a list of zero or more MDNS Link Invalid TLVs)Section 9.2) for links that are not valid, and zero or more MDNS Link Subscribed TLVs (Section 9.3) for links that are valid. For each valid link, it begins forwarding all mDNS traffic from that link to the Discovery Proxy. Delivery is not guaranteed: if there is no buffer space, packets will be dropped. It is expected that regular mDNS retry processing will take care of retransmission of lost packets. The amount of buffer space is implementation dependent, but generally should not be more than the bandwidth delay product of the TCP connection [RFC1323].

mDNS messages from Relays to Proxies are framed within DNS Session Signaling messages. This allows multiple TLVs to be included. Each forwarded mDNS message is contained in an MDNS Message TLV Section 9.4. The layer 2 source address of the message, if known, MAY be encoded in a Layer 2 Source TLV (Section 9.5). The source IP address of the message MUST be encoded in a IP Source Address TLV (Section 9.6). The source port of the message MUST be encoded in an IP Source port TLV (Section 9.7). The link on which the message was received MUST be encoded in a Link Identifier TLV (Section 9.8). The Discovery Proxy MUST silently ignore unrecognized TLVs in mDNS messages, and MUST NOT discard mDNS messages that include unrecognized TLVs.

A Discovery Proxy may discontinue listening for mDNS messages on a particular link by sending a session signaling message containing an MDNS Link Discontinue TLV (Section 9.9). Subsequent messages from that link that had previously been queued indicating may arrive. The Discovery Proxy should silently ignore such messages. The Discovery Relay MUST discontinue generating such messages as soon as the request is received. The Discovery Relay does not respond to this

message other than to discontinue forwarding mDNS messages from the specified links.

7. Traffic from Proxies to Relays

Like mDNS traffic from relays, each mDNS message sent by a Discovery Proxie to a Discovery Relay is encapsulated in an MDNS Message TLV (Section 9.4) within a session signaling message. Each message MUST contain one or more Link Identifier TLVs (Section 9.8). The Discovery Relay will transmit the message to the mDNS port and multicast address on each link. The message MUST include one or more IP family TLVs (Section 9.10). For each such TLVs that is included, the message will be sent on each link using the specified IP family. If no family codes are recognized, no packets will be transmitted.

8. Discovery Proxy Behavior

Discovery Proxies treat links for which Discovery Relay service is being used as if they were virtual interfaces; in other words, a Discovery Proxy serving multiple links using multiple Discovery Relays behaves the same as a Discovery Proxy serving multiple links using multiple physical network interfaces.

Discovery Proxies responding to mDNS messages for non-link-local IP addresses where the unicast bit is set respond directly, rather than through a proxy. Link-local responses are not supported for links to which Discovery Proxies are not directly connected.

9. Session Signaling TLVs

This document defines a modest number of new DNS Session Signaling TLVs.

9.1. MDNS Link Request

The MDNS Link Request TLV conveys a 32-bit link identifier from which a Discovery Proxy is requesting that a Discovery Relay forward mDNS traffic. The link identifier comes from the orchestration system (see Section 4.2.1). The SSOP-TYPE for this TLV is TBD1. The SSOP-LENGTH is always 4. The SSOP-DATA is the 32-bit identifier in network byte order.

9.2. MDNS Link Invalid

The MDNS Link Invalid TLV is returned in response to a session signaling message containing an MDNS Link Request, and returns the 32-bit identifier that was contained in that request. The link identifier comes from an MDNS Link Request TLV in the message being

responded to. The TLV indicates that the specified link identifier does not refer to a valid link, either because the link is not supported by the Discovery Relay, or because the identifier is not known. The SSOP-TYPE for this TLV is TBD2. The SSOP-LENGTH is always 4. The SSOP-DATA is the 32-bit identifier in network byte order.

9.3. MDNS Link Subscribed

The MDNS Link Subscribed TLV is returned in response to a session signaling message containing an MDNS Link Request, and returns the 32-bit identifier from a MDNS Link Request TLV that was contained in that request. It indicates that MDNS messages for the specified link have been successfully subscribed. The SSOP-TYPE for this TLV is TBD3. The SSOP-LENGTH is always 4. The SSOP-DATA is the 32-bit identifier in network byte order.

9.4. MDNS Message

The MDNS Message TLV is used to encapsulate an mDNS message that is being forwarded from a link to a Discovery Proxy, or is being forwarded from a Discovery Proxy to a link. The SSOP-TYPE for this TLV is TBD4. SSOP-LENGTH is the length of the application layer payload of the MDNS message. SSOP-DATA is the application layer payload of the message.

9.5. Layer 2 Source Address

The Layer 2 Source Address TLV is used to report the layer 2 address from which an mDNS message was received. This TLV is optionally present in session signaling messages from Discovery Relays to Discovery Proxies that contain mDNS messages when the source link-layer address is known. The SSOP-TYPE is TBD5. SSOP-LENGTH is variable, depending on the length of link-layer addresses on the link from which the message was received. SSOP-data is the link-layer address as it was received on the link.

9.6. IP Source Address

The IP Source Address TLV is used to report the IP source address from which an mDNS message was received. This TLV is present in session signaling messages from Discovery Relays to Discovery Proxies that contain mDNS messages. SSOP-TYPE is TBD6. SSOP-LENGTH is either 4, for an IPv4 address, or 16, for an IPv6 address. SSOP-DATA is the IP Address.

9.7. IP Source Port

The IP Source Port TLV is used to report the IP source port from which an mDNS message was received. This TLV is present in session signaling messages from Discovery Relays to Discovery Proxies. SSOP-TYPE is TBD7. SSOP-LENGTH is 2. SSOP-DATA is the source port in network byte order.

9.8. Link Identifier

This option is used both in session signaling messages from Discovery Proxies to Discovery Relays that contain mDNS messages, and in message from Discovery Relays to Discovery Proxies that contain mDNS messages. In the former case, it indicates a link to which the message should be forwarded; in the latter case, it indicates the link on which the message was received. SSOP-TYPE is TBD8. SSOP-LENGTH is 4. SSOP-DATA is a 32-bit link identifier as described in Section 4.2.1.

9.9. MDNS Discontinue

This option is used by Discovery Proxies to unsubscribe to mDNS messages on the specified link. More than one may be present in a single session signaling message. SSOP-TYPE is TBD9. SSOP-LENGTH is 4. SSOP-DATA is a 32-bit link identifier as described in Section 4.2.1.

9.10. IP Address Family

This option is used in mDNS messages sent by Discovery Proxies to links to indicate to the Discovery Relay which IP address family or families should be used when transmitting the message on the link. More than one may be present in a single session signaling message. SSOP-TYPE is TBD10. SSOP-LENGTH is 1. SSOP-DATA is a 8-bit IP family identifier. A value of 1 indicates IPv4. A value of 2 indicates IPv6. Other values are reserved, and MUST be ignored if not recognized.

10. Security Considerations

11. IANA Considerations

The IANA is kindly requested to update the DNS Session Signaling Type Codes Registry [I-D.ietf-dnsop-session-signal] by allocating codes for each of the TBD type codes listed in the following table, and by updating this document, here and in Section 9. Each type code should list this document as its reference document.

Opcode	Status	Name
TBD1	Standard	MDNS Link Request
TBD2	Standard	MDNS Link Invalid
TBD3	Standard	MDNS Link Subscribed
TBD4	Standard	MDNS Messsage
TBD5	Standard	Layer Two Source Address
TBD6	Standard	IP Source Address
TBD7	Standard	IP Destination Address
TBD8	Standard	Link Identifier
TBD9	Standard	MDNS Discontinue
TBD10	Standard	IP Address Family

DNS Session Signaling Type Codes to be allocated

12. IANA Considerations

13. Acknowledgments

14. References

14.1. Normative References

[I-D.ietf-dnsop-session-signal]
Bellis, R., Cheshire, S., Dickinson, J., Dickinson, S., Mankin, A., and T. Pusateri, "DNS Session Signaling", draft-ietf-dnsop-session-signal-02 (work in progress), March 2017.

[I-D.ietf-dnssd-hybrid]
Cheshire, S., "Discovery Proxy for Multicast DNS-Based Service Discovery", draft-ietf-dnssd-hybrid-06 (work in progress), March 2017.

[I-D.ietf-tls-tls13]
Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-20 (work in progress), April 2017.

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

[RFC1323] Jacobson, V., Braden, R., and D. Borman, "TCP Extensions for High Performance", RFC 1323, DOI 10.17487/RFC1323, May 1992, <http://www.rfc-editor.org/info/rfc1323>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <http://www.rfc-editor.org/info/rfc6762>.

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

[RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016, <http://www.rfc-editor.org/info/rfc7788>.

14.2. Informative References

[TR-069] Broadband Forum, "CPE WAN Management Protocol", November 2013, <https://www.broadband-forum.org/technical/download/TR-069_Amendment-5.pdf>.

Authors' Addresses

Stuart Cheshire
Apple Inc.
1 Infinite Loop
Cupertino, California 95014
USA

Phone: +1 408 974 3207
Email: cheshire@apple.com

Ted Lemon
Nominum, Inc.
800 Bridge Parkway
Redwood City, California 94065
United States of America

Phone: +1 650 381 6000
Email: ted.lemon@nominum.com