

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 16, 2018

S. Cheshire  
Apple Inc.  
T. Lemon  
Barefoot Consulting  
November 12, 2017

**Multicast DNS Discovery Relay**  
**draft-sctl-dnssd-mdns-relay-02**

Abstract

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

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 May 16, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Terminology</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">Protocol Overview</a>	<a href="#">5</a>
<a href="#">3.1.</a>	<a href="#">Connections between Proxies and Relays (overview)</a>	<a href="#">5</a>
<a href="#">3.2.</a>	<a href="#">mDNS Messages On Multicast Links</a>	<a href="#">6</a>
<a href="#">4.</a>	<a href="#">Connections between Proxies and Relays (details)</a>	<a href="#">6</a>
<a href="#">5.</a>	<a href="#">Traffic from Relays to Proxies</a>	<a href="#">7</a>
<a href="#">6.</a>	<a href="#">Traffic from Proxies to Relays</a>	<a href="#">8</a>
<a href="#">7.</a>	<a href="#">Discovery Proxy Behavior</a>	<a href="#">9</a>
<a href="#">8.</a>	<a href="#">DSO TLVs</a>	<a href="#">10</a>
<a href="#">8.1.</a>	<a href="#">mDNS Link Request</a>	<a href="#">10</a>
<a href="#">8.2.</a>	<a href="#">mDNS Discontinue</a>	<a href="#">10</a>
<a href="#">8.3.</a>	<a href="#">mDNS Message</a>	<a href="#">10</a>
<a href="#">8.4.</a>	<a href="#">Link Identifier</a>	<a href="#">10</a>
<a href="#">8.5.</a>	<a href="#">Layer Two Source Address</a>	<a href="#">11</a>
<a href="#">8.6.</a>	<a href="#">IP Source</a>	<a href="#">11</a>
<a href="#">9.</a>	<a href="#">Provisioning</a>	<a href="#">11</a>
<a href="#">9.1.</a>	<a href="#">Provisioned Objects</a>	<a href="#">12</a>
<a href="#">9.1.1.</a>	<a href="#">Discovery Proxy</a>	<a href="#">12</a>
<a href="#">9.1.2.</a>	<a href="#">Multicast Link</a>	<a href="#">13</a>
<a href="#">9.1.3.</a>	<a href="#">Discovery Relay</a>	<a href="#">13</a>
<a href="#">9.2.</a>	<a href="#">Configuration Files</a>	<a href="#">14</a>
<a href="#">9.3.</a>	<a href="#">Discovery Proxy Configuration</a>	<a href="#">15</a>
<a href="#">9.4.</a>	<a href="#">Discovery Relay Configuration</a>	<a href="#">15</a>
<a href="#">10.</a>	<a href="#">Security Considerations</a>	<a href="#">16</a>
<a href="#">11.</a>	<a href="#">IANA Considerations</a>	<a href="#">16</a>
<a href="#">12.</a>	<a href="#">Acknowledgments</a>	<a href="#">16</a>
<a href="#">13.</a>	<a href="#">References</a>	<a href="#">17</a>
<a href="#">13.1.</a>	<a href="#">Normative References</a>	<a href="#">17</a>
<a href="#">13.2.</a>	<a href="#">Informative References</a>	<a href="#">18</a>
	<a href="#">Authors' Addresses</a>	<a href="#">18</a>



## 1. Introduction

The Discovery Proxy for Multicast DNS-Based Service Discovery [[I-D.ietf-dnssd-hybrid](#)] is a mechanism for discovering services on a subnetted network through the use of Discovery Proxies, which issue Multicast DNS (mDNS) requests [[RFC6762](#)] on various multicast links in the network on behalf of a remote host performing DNS-Based Service Discovery [[RFC6763](#)].

In the original Discovery Proxy specification, it is assumed that for every multicast 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 multicast link without requiring a full Discovery Proxy on every multicast link.

The Discovery Relay operates by listening for TCP connections from Discovery Proxies. When a Discovery Proxy connects, the connection is authenticated and secured using TLS. The Discovery Proxy can then specify one or more multicast links from which it wishes to receive mDNS traffic. The Discovery Proxy can also send messages to be transmitted on its behalf on one or more of those multicast links. DNS Stateful Operations (DSO) [[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 remote virtual interfaces for the Discovery proxy, one on each multicast 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 multicast link; in this case, the Discovery Proxy ideally should only be using one such Relay Proxy per multicast 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 relays received mDNS messages to a Discovery Proxy, and can transmit mDNS messages on behalf of that Discovery Proxy.

**multicast link** A maximal set of network connection points, such that any host connected to any connection point in the set may send a packet with a link-local multicast destination address (specifically the mDNS link-local multicast destination address [[RFC6762](#)]) that will be received by all hosts connected to all other connection points in the set. Note that it is becoming increasingly common for a multicast link to be smaller than its corresponding unicast link. For example it is becoming common to have multiple Wi-Fi Access Points on a shared Ethernet backbone, where the multiple Wi-Fi Access Points and their shared Ethernet backbone form a single unicast link (a single IPv4 subnet, or single IPv6 prefix) but not a single multicast link. Unicast packets between two hosts on that IPv4 subnet or IPv6 prefix are correctly delivered, but multicast packets are not forwarded between the various Wi-Fi Access Points. Given the slowness of Wi-Fi multicast, the decision to not forward multicast packets between Wi-Fi Access Points is reasonable, and that further supports the need for technologies like Discovery Proxy and Discovery Relay to facilitate discovery on these networks.

**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.



### **3. Protocol Overview**

This document describes a way for Discovery Proxies to communicate with mDNS agents on remote multicast links 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 Proxies and Relays (overview)**

Discovery Relays listen for incoming connection requests. 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 a single 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 DSO messages [\[I-D.ietf-dnsop-session-signal\]](#). There will be three types of such messages:

- o Control messages from Discovery Proxy to Discovery Relay
- o mDNS messages from Discovery Relay to Discovery Proxy
- o mDNS messages from Discovery Proxy to Discovery Relay

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

mDNS messages from a Discovery Proxy to a Discovery Relay cause the Discovery Relay to transmit the mDNS message on one or more multicast 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 multicast 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 (See [\[I-D.ietf-dnsop-session-signal\] Section 4](#)).





### **3.2. mDNS Messages On Multicast Links**

Discovery Relays listen for mDNS traffic on all configured multicast links that have at least one active subscription from a Discovery Proxy. When an mDNS message is received on a multicast link, it is forwarded on every open Discovery Proxy connection that is subscribed to mDNS traffic on that multicast 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. Excess mDNS packets are silently discarded. In reality this is expected to be a nonissue. Particularly on networks like Wi-Fi, multicast packets are transmitted at rates ten or even a hundred times slower than unicast packets. This means that even at peak multicast packets rates, it is likely that a unicast TCP connection will be able to carry those packets with ease.

Discovery Proxies send mDNS messages they wish to have sent on their behalf on remote multicast link(s) on which the Discovery Proxy has an active subscription. A Discovery Relay will not transmit mDNS packets on any multicast link on which the remote Discovery Proxy does not have an active subscription, since it makes no sense for a Discovery Proxy to ask to have a query sent on its behalf if it's not able to receive the responses to that query.

## **4. Connections between Proxies and Relays (details)**

When a Discovery Relay starts, it opens a passive TCP listener to receive incoming connection requests from Discovery Proxies. This listener may be bound to one or more source IP addresses, or to the wildcard address, depending on the 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.

## 5. 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 multicast link be forwarded on a particular connection, the Discovery Proxy must send one or more DSO messages containing mDNS a Link Request TLV ([Section 8.1](#)) indicating the multicast link from which traffic is requested.

When such a message is received, the Discovery Relay validates that the specified multicast link is available for forwarding, and that



forwarding is enabled for that multicast link. For each such message the Discovery Relay validates the multicast link specified and includes in a single response RCODE 0 if the multicast link specified is valid, or RCODE 11 (DSOFAIL) otherwise. For each valid multicast 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](#)]. The Discovery Relay should use the TCP\_NOTSENT\_LOWAT mechanism [[NOTSENT](#)][[PRIO](#)] or equivalent, to avoid building up a backlog of data in excess of the amount necessary to have in flight to fill the bandwidth delay product of the TCP connection.

mDNS messages from Relays to Proxies are framed within DSO messages. Each DSO message can contain multiple TLVs, but only a single mDNS message is conveyed per DSO message. Each forwarded mDNS message is contained in an mDNS Message TLV ([Section 8.3](#)). The layer two source address of the message, if known, MAY be encoded in a Layer Two Source TLV ([Section 8.5](#)). The source IP address and port of the message MUST be encoded in a IP Source ([Section 8.6](#)). The multicast link on which the message was received MUST be encoded in a Link Identifier TLV ([Section 8.4](#)). 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 multicast link by sending a DSO message containing an mDNS Link Discontinue TLV ([Section 8.2](#)). Subsequent messages from that link that had previously been queued may arrive after listening has been discontinued. 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.

## **[6.](#) Traffic from Proxies to Relays**

Like mDNS traffic from relays, each mDNS message sent by a Discovery Proxy to a Discovery Relay is encapsulated in an mDNS Message TLV ([Section 8.3](#)) within a DSO message. Each message MUST contain one or more Link Identifier TLVs ([Section 8.4](#)). The Discovery Relay will transmit the message to the mDNS port and multicast address on each link specified in the message using the specified IP address family.



## **7. Discovery Proxy Behavior**

Discovery Proxies treat multicast links for which Discovery Relay service is being used as if they were virtual interfaces; in other words, a Discovery Proxy serving multiple multicast links using multiple Discovery Relays behaves the same as a Discovery Proxy serving multiple multicast links using multiple physical network interfaces. In this section we refer to multicast links served directly by the Discovery Proxy as locally-connected links, and multicast links served through the Discovery Relay as relay-connected links.

What this means is that when a Discovery Proxy receives a DNSSD query from a client, it will generate mDNS messages on the relevant multicast link for which it is acting as a proxy. For locally-connected links, those messages will be sent directly. For relay-connected links, the messages will be sent through the Discovery Relay that is being used to serve that multicast link.

Responses from devices on locally-connected links are processed normally. Responses from devices on relay-connected links are received by the Discovery Relay, encapsulated, and forwarded to the Discovery Proxy; the discovery proxy then processes these messages using the link-identifying information included in encapsulation.

Discovery Proxies do not respond to mDNS queries on relay-connected links. If an mDNS query is received from a Discovery Relay, the Discovery Proxy silently discards it.

In principle it could be the case that some device is capable of performing service discovery using mDNS, but not using the DNS protocol. Responding to mDNS queries received from the Discovery Relay could address this use case. However, it is believed that no such devices exist, and therefore the preferred behavior is that all queries be resolved with unicast rather than multicast.





## **8. DSO TLVs**

This document defines a modest number of new DSO TLVs.

### **8.1. mDNS Link Request**

The mDNS Link Request TLV conveys a link identifier from which a Discovery Proxy is requesting that a Discovery Relay forward mDNS traffic. The link identifier comes from the provisioning configuration (see [Section 9](#)). The SSOP-TYPE for this TLV is TBD-R. The SSOP-LENGTH is always 5. The SSOP-DATA is the 8-bit address family followed by the 32-bit link identifier, in network byte order. An address family value of 1 indicates IPv4 and 2 indicates IPv6, as recorded in the IANA Registry of Address Family Numbers [[AdFam](#)].

### **8.2. mDNS Discontinue**

This option is used by Discovery Proxies to unsubscribe to mDNS messages on the specified multicast link. More than one may be present in a single DSO message. SSOP-TYPE is TBD-D. SSOP-LENGTH is 5. SSOP-DATA is the 8-bit address family followed by the 32-bit link identifier, in network byte order, as described in [Section 9](#).

### **8.3. mDNS Message**

The mDNS Message TLV is used to encapsulate an mDNS message that is being forwarded from a multicast link to a Discovery Proxy, or is being sent from a Discovery Proxy for transmission on a multicast link. Only the application layer payload of the mDNS message is carried in the DSO mDNS Message TLV, i.e., just the DNS message itself, beginning with the DNS Message ID, not the IP or UDP headers. The SSOP-TYPE for this TLV is TBD-M. SSOP-LENGTH is the length of the encapsulated mDNS message. SSOP-DATA is the content of the encapsulated mDNS message.

### **8.4. Link Identifier**

This option is used both in DSO messages from Discovery Relays to Discovery Proxies that contain received mDNS messages, and from Discovery Proxies to Discovery Relays that contain mDNS messages to be transmitted on the multicast link. In the former case, it indicates the multicast link on which the message was received; in the latter case, it indicates the multicast link on which the message should be transmitted. SSOP-TYPE is TBD-L. SSOP-LENGTH is 5. SSOP-DATA is the 8-bit address family followed by the 32-bit link identifier, in network byte order, as described in [Section 9](#).



### **8.5. Layer Two Source Address**

The Layer Two Source Address TLV is used to report the link-layer address from which an mDNS message was received. This TLV is optionally present in DSO messages from Discovery Relays to Discovery Proxies that contain mDNS messages when the source link-layer address is known. The SSOP-TYPE is TBD-2. 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.

### **8.6. IP Source**

The IP Source TLV is used to report the IP source address and port from which an mDNS message was received. This TLV is present in DSO messages from Discovery Relays to Discovery Proxies that contain mDNS messages. SSOP-TYPE is TBD-A. SSOP-LENGTH is either 6, for an IPv4 address, or 18, for an IPv6 address. SSOP-DATA is the source port, followed by the IP Address, in network byte order.

## **9. Provisioning**

In order for a Discovery Proxy to use Discovery Relays, it must be configured with sufficient information to identify multicast links on which service discovery is to be supported and connect to discovery relays supporting those multicast links, if it is not running on a host that is directly connected to those multicast links.

A Discovery Relay must be configured both with a set of multicast links to which the host on which it is running is connected, on which mDNS relay service is to be provided, and also with a list of one or more Discovery Proxies authorized to use it.

On a network supporting DNS Service Discovery using Discovery Relays, more than one different Discovery Relay implementation is likely be present. While it may be that only a single Discovery Proxy is present, that implementation will need to be able to be configured to interoperate with all of the Discovery Relays that are present. Consequently, it is necessary that a standard set of configuration parameters be defined for both Discovery Proxies and Discovery Relays.

DNS Service Discovery generally operates within a constrained set of links, not across the entire internet. This section assumes that what will be configured will be a limited set of links operated by a single entity or small set of cooperating entities, among which services present on each link should be available to users on that link and every other link. This could be, for example, a home



network, a small office network, or even a network covering an entire building or small set of buildings. The set of Discovery Proxies and Discovery Relays within such a network will be referred to in this section as a 'Discovery Domain'.

Depending on the context, several different candidates for configuration of Discovery Proxies and Discovery relays may be applicable. The simplest such mechanism is a configuration file, but regardless of provisioning mechanism, certain configuration information needs to be communicated to the devices, as outlined below.

### **9.1. Provisioned Objects**

Three types of objects must be described in order for Discovery Proxies and Discovery Relays to be provisioned: Discovery Proxies, Multicast Links, and Discovery Relays. "Human-readable" below means actual words or proper names that will make sense to an untrained human being. "Machine-readable" means a name that will be used by machines to identify the entity to which the name refers. Each entity must have a machine-readable name and may have a human-readable name. Every name must be unique: no two entities can have the same human-readable name nor machine-readable name.

#### **9.1.1. Discovery Proxy**

The description of a Discovery Proxy consists of:

`name` a machine-readable name used to reference this discovery proxy in provisioning.

`hr-name` an optional human-readable name which can appear in provisioning, monitoring and debugging systems. Must be unique within a Discovery Domain.

`public-key` a public key that identifies the Discovery Proxy. This key can be shared across services on the Discovery Proxy Host. The public key is used both to uniquely identify the Discovery Proxy and to authenticate connections from it.

`private-key` the private key corresponding to the public key.

`source-ip-addresses` a list of IP addresses that may be used by the Discovery Proxy when connecting to Discovery Relays. These addresses should be addresses that are configured on the Discovery Proxy Host. They should not be temporary addresses. All such addresses must be reachable within the Discovery Domain.



**public-ip-addresses** a list of IP addresses that may be used to submit DNS queries to the Discovery Proxy. This is not used for interoperation with Discovery Relays, but is mentioned here for completeness: this list of addresses may differ from the 'source-ip-addresses' list. If any of these addresses are reachable from outside of the Discovery Domain, services in that domain will be discoverable outside of the domain.

**multicast links** a list of multicast links on which this Discovery Proxy is expected to provide service

The private key should never be distributed to other hosts; all of the other information describing a Discovery Proxy can be safely shared with Discovery Relays.

#### **9.1.2. Multicast Link**

The description of a multicast link consists of:

**link-identifier** An 32-bit identifier that uniquely identifies that link within the Discovery Domain. Each link MUST have exactly one such identifier. Link Identifiers do not have any special semantics, and are not intended to be human-readable.

**ldh-name** A fully-qualified domain name for the multicast link that is used to form an LDH domain name as described in [[I-D.ietf-dnssd-hybrid](#)], section 5.3. This name is used to identify the link during provisioning, and must be present.

**hr-name** A human-readable user-friendly fully-qualified domain name for the multicast link. This name MUST be unique within the Discovery Domain. Each multicast link MUST have exactly one such name.

The 'ldh-name' and 'hr-name' names can be used to form the LDH and human-readable domain names as described in [[I-D.ietf-dnssd-hybrid](#)], section 5.3.

#### **9.1.3. Discovery Relay**

The description of a Discovery Relay consists of:

**name** a required machine-readable identifier used to reference the relay

**hr-name** an optional human-readable name which can appear in provisioning, monitoring and debugging systems. Must be unique within a Discovery Domain.





`public-key` a public key that identifies the Discovery Relay. This key can be shared across services on the Discovery Relay Host. Indeed, if a Discovery Proxy and Discovery Relay are running on the same host, the same key may be used for both. The public key uniquely identifies the Discovery Relay and is used by the Discovery Proxy to verify that it is talking to the intended Discovery Relay after a TLS connection has been established.

`private-key` the private key corresponding to the public key.

`connect-tuples` a list of IP address/port tuples that may be used to connect to the Discovery Relay. The relay may be configured to listen on all addresses on a single port, but this is not required, so the port as well as the address must be specified.

`multicast links` a list of multicast links to which this relay is physically connected.

The private key should never be distributed to other hosts; all of the other information describing a Discovery Relay can be safely shared with Discovery Proxies.

## **9.2. Configuration Files**

For this discussion, we assume the simplest possible means of configuring Discovery Proxies and Discovery Relays: the configuration file. Any environment where changes will happen on a regular basis will either require some automatic means of generating these configuration files as the network topology changes, or will need to use a more automatic method for configuration, such as HNCP.

There are many different ways to organize configuration files. This discussion assumes that multicast links, relays and proxies will be specified as objects, as described above, perhaps in a master file, and then the specific configuration of each proxy or relay will reference the set of objects in the master file, referencing objects by name. This approach is not required, but is simply shown as an example. In addition, the private keys for each relay and proxy must appear only in that relay or proxy's configuration file.

The master file contains a list of Discovery Relays, Discovery Proxies and Multicast Links. Each object has a name and all the other data associated with it. We do not formally specify the format of the file, but it might look something like this:



```
Relay upstairs
  public-key xxx
  connect-tuple 192.0.2.1 1917
  connect-tuple fd00::1 1917
  link upstairs-wifi
  link upstairs-wired
Relay downstairs
  public-key xxx
  connect-tuple 192.51.100.1 2088
  connect-tuple fd00::2 2088
  link downstairs-wifi
  link downstairs-wired
Proxy main
  public-key xxx
  address 203.1.113.1
Link upstairs-wifi
  id 1
  name "Upstairs Wifi"
Link upstairs-wired
  id 2
  hr-name "Upstairs Wired"
Link downstairs-wifi
  id 3
  name "Downstairs Wifi"
Link downstairs-wired
  id 4
  hr-name "Downstairs Wired"
```

### **9.3. Discovery Proxy Configuration**

The Discovery Proxy configuration contains enough information to identify which Discovery Proxy is being configured, enumerate the list of multicast links it is intended to serve, and provide keying information it can use to authenticate to Discovery Relays. It may also contain custom information about the port and/or IP address(es) on which it will respond to DNS queries.

An example configuration, following the convention used in this section, might look something like this:

```
Proxy
[placeholder, to be completed later]
```

### **9.4. Discovery Relay Configuration**



## [10.](#) Security Considerations

## [11.](#) IANA Considerations

The IANA is kindly requested to update the DSO 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 8](#). Each type code should list this document as its reference document.

Opcode	Status	Name
TBD-R	Standard	mDNS Link Request
TBD-D	Standard	mDNS Discontinue
TBD-M	Standard	mDNS Messsage
TBD-2	Standard	Layer Two Source Address
TBD-A	Standard	IP Source
TBD-L	Standard	Link Identifier

DSO Type Codes to be allocated

## [12.](#) Acknowledgments



## **13. References**

### **13.1. Normative References**

- [I-D.ietf-dnsop-session-signal]  
Bellis, R., Cheshire, S., Dickinson, J., Dickinson, S., Mankin, A., and T. Pusateri, "DNS Stateful Operations", [draft-ietf-dnsop-session-signal-04](#) (work in progress), September 2017.
- [I-D.ietf-dnssd-hybrid]  
Cheshire, S., "Discovery Proxy for Multicast DNS-Based Service Discovery", [draft-ietf-dnssd-hybrid-07](#) (work in progress), September 2017.
- [I-D.ietf-tls-tls13]  
Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", [draft-ietf-tls-tls13-21](#) (work in progress), July 2017.
- [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>>.
- [RFC1323] Jacobson, V., Braden, R., and D. Borman, "TCP Extensions for High Performance", [RFC 1323](#), DOI 10.17487/RFC1323, May 1992, <<https://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, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", [RFC 6762](#), DOI 10.17487/RFC6762, February 2013, <<https://www.rfc-editor.org/info/rfc6762>>.
- [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>>.
- [RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", [RFC 7788](#), DOI 10.17487/RFC7788, April 2016, <<https://www.rfc-editor.org/info/rfc7788>>.





### **13.2. Informative References**

- [AdFam] "IANA Address Family Numbers Registry",  
<[https://www.iana.org/assignments/  
address-family-numbers/](https://www.iana.org/assignments/address-family-numbers/)>.
- [NOTSENT] "TCP\_NOTSENT\_LOWAT socket option", July 2013,  
<<https://lwn.net/Articles/560082/>>.
- [PRIO] "Prioritization Only Works When There's Pending Data to  
Prioritize", January 2014, <[https://insouciant.org/tech/  
prioritization-only-works-when-theres-pending-data-to-  
prioritize/](https://insouciant.org/tech/prioritization-only-works-when-theres-pending-data-to-prioritize/)>.
- [TR-069] Broadband Forum, "CPE WAN Management Protocol", November  
2013, <[https://www.broadband-forum.org/technical/download/  
TR-069\\_Amendment-5.pdf](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](mailto:cheshire@apple.com)

Ted Lemon  
Barefoot Consulting  
Brattleboro, Vermont 05301  
United States of America

Email: [mellon@fugue.com](mailto:mellon@fugue.com)

