Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 27 January 2022

T. Lemon
邓 乔宇 (J. Deng)
Apple Inc.

26 July 2021

Permissionless Advertising and Discovery of DNS-SD Authoritative Zones
draft-tljd-dnssd-zone-discover-00

Abstract

This document describes how to make DNS-SD browsing domains available for browsing and discovery without requiring special cooperation from the network infrastructure. Zones made available in this way are browsed using DNS or DNS Push. The mechanism for advertising them is Multicast DNS (mDNS). This allows DNS-SD browsers to benefit from the permissionless aspects of mDNS without relying on mDNS for all queries, which improves scalability and reliability in applications where many services may be advertised.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

DNS-SD currently provides permissionless advertising and discovery using multicast DNS [RFC6762]. Unfortunately, multicast DNS has some limitations. In addition to the obvious limitation that it only works between services and users that are connected to a single multicast domain (generally a single link), in many situations excessive use of multicast is unreliable, which can cause discovery to fail when a service is actually present.

By contrast, DNS service discovery using unicast traffic is not limited by the scope of reachability of link-local multicast. On networks where multicast is less reliable, or more costly, unicast DNS-SD is clearly advantageous. However, because of the hierarchical nature of DNS, existing solutions for providing unicast DNS rely on coordination with the network infrastructure. In many settings, particularly on home networks, this coordination is not available, and so we must fall back on multicast DNS, despite its limitations.

This document defines a new mechanism for discovering the availability of unicast DNS service discovery using multicast DNS. Although somewhat limited in the sense that this mechanism still relies on multicast DNS as a means of discovering the unicast service, this mechanism can substantially reduce reliance on multicast DNS, so that its limitations are minimized. Additionally, it makes it possible for devices that provide discovery to adjacent networks, such as stub networks, to overcome the limitations of link-local multicast for this application.

This document describes how to advertise and discover authoritative service for a DNS domain, and how to advertise the availability of service discovery for a DNS domain, using multicast DNS.

2. Glossary

Advertiser a service that is advertising its availability through some authority

Authority a device, connected to an infrastructure link, that is advertising service discovery for an authority dataset using mDNS.

Authority Dataset a collection of authoritative data to be advertised, and which can be treated as a single coherent set. More than one authority may advertise the same dataset; what makes it "the same dataset" is the expectation that whichever authority is asked for an answer to a particular question will generally give the same answer as any other authority.

DNSSD Browser a device, connected to an infrastructure link, that discovers authorities and uses them to discover services advertised by advertisers.

3. Data Model

The goal of the mechanism described in this document is to enable permissionless advertising and discovery of authoritative DNS servers that can be used for service discovery. From the perspective of a consumer of such a service on a particular IP link, there may be more than one such service. Each such service can be thought of as providing an authority dataset.

3.1. Authority Datasets

3.1.1. Multicast DNS

One example of an authority dataset is the set of services that can be discovered by a DNSSD Discovery Proxy [RFCxxx]. A discovery proxy acts as an authoritative DNS server mapping information advertised using multicast DNS on a particular link to a DNS zone.

There may be more than one discovery proxy for a particular multicast DNS link. If so, both discovery proxies can be expected to return the same answers to any questions asked by a DNS-SD client that is browsing for services within that zone. This is what is meant by an "authority dataset": the data returned by one authoritative server that answers for that dataset should be the same as the data returned by the other server.

Because of the dynamic nature of mDNS, there is no enforcement mechanism to ensure that two discovery proxies would answer the same DNS question in exactly the same way, but each server is functionally equivalent: there is no reason to prefer one server over the other, nor to query both servers to avoid missing data known to one but not the other.

3.1.2. Authoritative DNS

Another example of an authority dataset is an authoritative DNS server that maintains a DNS zone for DNS Service Discovery using the DNS-SD Service Registration Protocol. In this case, there is one primary authoritative server and, potentially, more than one secondary authoritative server. The secondary server databases are all dependent on the primary server's database, and are maintained using DNS zone transfers or some other hierarchical replication mechanism.

In this case, the primary and the secondary servers are all serving an authoritative zone, which is another example of an authority dataset. The authoritative zone may have different versions, which can be known using the zone serial number, but in principle each

authoritative server is equivalently valid: there is no reason to prefer one over the other. This is another example of an authority dataset.

3.1.3. SRP Replication

A third example of an authority dataset would be a set of one or more SRP servers that cooperate to maintain a common database using the SRP replication protocol [SRP Replication]. These servers are each authoritative DNS servers, in the sense that they answer authoritatively for questions within the DNS zone that they manage, but unlike a typical DNS authoritative service configuration, there is no hierarchy-no server is primary-and there are no discrete versions of the zone database, so there is no way to generate a meaningful serial number that could be used to manage zone transfers.

As with Discovery Proxies, although it's quite possible at any given moment in time that the same query to two different SRP replication peers will yield different answers, the dataset being managed by these servers is the same dataset, and therefore is also an authority dataset.

3.2. DNSSD Browsing Domains

Service discovery on a local link always implicitly includes one authority dataset: the set of all mDNS services advertised on that link. DNSSD browsers always search for services within this dataset. Additional datasets are made available by advertising additional legacy browsing domains locally. So each authority dataset other than the link-local authority dataset must be explicitly advertised using mDNS; when the DNSSD browser is asked to browse for local services without explicitly specifying a domain in which to browse, it attempts to discover that service in each of the authority datasets advertised locally for discovery, and also in the link-local dataset provided by mDNS.

Note that DNSSD [RFC6763] also provides for discovering browsing domains using DNS, either using the domain name search list or using the DNS reverse domain query [RFC6763 section ???]. DNS browsing domains are provided by the network infrastructure, and complement browsing domains that may be provided permissionlessly using mDNS.

4. Advertising an Authority Dataset to be Used for Service Discovery

There are four steps an authority must follow to advertise the availability of an authority dataset for service discovery:

* Choose a domain to represent that authority dataset

* Advertise itself as an authority that provides name service for that domain (and hence that dataset)

* Advertise its address information

* Advertise the availability of service discovery for that domain

4.1. Choosing a Domain to Advertise

When advertising a domain for discovery, it must be the case that all authorities servers that advertise that domain are advertising the same information. Thus, the domain being advertised can be treated as an identifier for a particular authority dataset. How this is accomplished is out of scope for this document; one solution is described in [SRP Replication].

Because the domain is being advertised using multicast DNS, we assume that there is no delegation in the global DNS; if there were, there would be no reason to advertise the domain using mDNS. Furthermore, in order to prevent domain spoofing using the technique described here, the DNS resolver that is discovering this domain is required to prove that no delegation for the domain being advertised using mDNS exists in the global DNS hierarchy.

Given that most stub resolvers at present do not support DNSSEC, the domain being advertised will have to be a subdomain of some domain that is known to be a locally-served domain [RFC6761?]. In this case the client can be sure that this domain never appears in the DNS by definition, rather than by validating the non-existence of the delegation.

Two domains that are ideal for this purpose are 'home.arpa' [Dot Home] and 'service.arpa' [SRP]. The 'home.arpa' domain is generally intended for use in home networks, so this is appropriate for use in cases where the device advertising the domain is expected to be installed in a home network; for devices that are not expected to be installed in a home network, 'service.arpa' is preferable.

Given that there is no way for a particular device to know for certain that the network setting in which it is installed is in fact 'a home' or 'not a home,' this advice is merely a suggestion: a consumer product should probably use 'home.arpa' and a commercial product should probably use 'service.arpa', and perhaps either device should be configurable, but in practice it is not crucial to get this perfectly correct.

Bearing in mind that there may be multiple authorities and multiple authority datasets being provided on the same infrastructure link, it is important to choose a domain that is not ambiguous. Therefore, devices advertising domains for discovery MUST NOT use 'home.arpa' or 'service.arpa' directly: when using these domains, a unique subdomain must be chosen below that domain, rather than using the root domain.

It may be useful to identify the subdomain in terms of some visible network identifier. For instance, if the authority dataset contains the set of services for a particular WiFi link, it might make sense to use the name 'SSID.wifi.home.arpa'. This shows the SSID of the WiFi link, and differentiates between WiFi and other technologies (e.g. Thread) that use something like an SSID. For Thread networks, the domain 'thread.home.arpa' is used for this purpose, for example, and the thread network name is used as the leftmost label (e.g., 'my-home.thread.home.arpa.').

We have stated that the domain must be provably nonexistent, which is a slight simplification. For completeness, we will point out that if the delegation is secure, and the server being advertised is able to sign records such that they validate, this is also permissible. But in this case, there's likely no utility to using mDNS to advertise the authoritative server and, furthermore, this solution requires the stub resolver to do DNSSEC validation, which is not commonly supported at present.

Although '.local' is a locally-served domain, it is by definition served using multicast DNS. For this reason, authorities MUST NOT use '.local' to advertise their authority dataset.

4.2. Advertising DNS Authoritative Service for a Domain using multicast DNS

To advertise DNS authoritative service for a domain, the authoritative server publishes an NS record for that domain using multicast DNS. For example, to publish DNS service for example.thread.home.arpa., the NS record published with mDNS would be:

```
example.thread.home.arpa.  IN NS thread-server-1.local
```

Note that the DNS server for example.com has a .local suffix, meaning that its address can be discovered using mDNS. This is not required. If the DNS server is in a domain that can be looked up using ordinary DNS service, multicast DNS service is not required. This solution is preferred, but requires coordination with infrastructure, so doesn't address the core use case of this document.

Before publishing its chosen domain, the authority MUST validate that no other authority is advertising that domain for a different authority dataset. The mechanism for this validation is out of scope for this document, and is specific to the replication mechanism being used. If no replication mechanism is being used, the authority MUST publish its NS record as a unique record.

4.3. Advertising Address information for an Authority

Address information for an authority may be advertised using DNS or mDNS. If the authority happens to have a name published in the DNS, it SHOULD use that name, since it reduces reliance on multicast. In most cases, this will not be possible, in which case the authority MUST advertise addresses that can be used to reach it using mDNS.

4.4. Advertising the Availability of Service Discovery for a Domain

In order for services within an authority dataset to be discovered by DNSSD browsers, the domain that identifies that authority dataset must be advertised as a domain in which discovery should be done. This is accomplished by advertising a PTR record in .local for the legacy browsing domain [RFC6763]. For example, if the domain being used is 'example.wifi.home.arpa.', then the PTR record would be as follows:

```
lb._dns-sd._udp.local IN PTR example.wifi.home.arpa.
```

When more than one authority is advertising discoverability for a particular authority dataset, there will be more than one of these records advertised, but this isn't a problem since they are not required to be unique. This record should not be advertised until the authority has successfully generated or discovered a domain that is unique to the authority dataset being advertised.

5. Discovering Authority Datasets

A DNSSD browser discovers the set of datasets available locally by issuing an mDNS query for lb._dns-sd._udp.local. This query will return zero or more PTR records. As each PTR record is returned, it is compared against the existing set of legacy browsing domains that the DNSSD browser maintains. If the target of the PTR record is not in this set, then it is checked for validity and, if valid, added to the set, and any ongoing browse operations begin to try to browse the new domain.

A browsing domain discovered using mDNS can only be valid if one of the following is true:

* It is a subdomain of a domain that is defined to be a locally served domain [???]

* The DNSSD browser has found a secure denial of existence for the domain and validated it using DNSSEC

* The DNSSD browser has found a secure delegation for the domain (in which case it MUST validate answers in that domain using DNSSEC).

In addition, a host on which a DNSSD browser is running may have discovered domains that would be considered valid because it is a locally-served domain, or because it can be proven not to exist in the DNS hierarchy, but for which authoritative service is already provided by the network infrastructure. In this case, the DNSSD browser MUST consider the information provided in multicast DNS to be invalid, and MUST only use the service information provided by the network infrastructure for that domain.

Examples of this would be a domain like 'home.arpa' that is served by the local infrastructure. In this case, if for example the local infrastructure answers with NXDOMAIN for 'example.wifi.home.arpa.' then even if the browser is not able to validate this answer, is MUST treat the mDNS advertisement for this domain as valid, since otherwise the presence of the locally served domain would prevent discovery in mDNS-advertised subdomains even though there is no conflict.

There are three types of browsing domains that might exist in the set of browsing domains maintained by the DNSSD browser.

* Link-Local: one for each network link to which the DNSSD browser is directly connected

* Infrastructure: browsing domains provided by the infrastructure

* Permissionless: browsing domains discovered using mDNS on local links

[RFC6763] and [DNS Push] already describe how to manage link-local and infrastructure browsing domains. Permissionless browsing domains are managed similarly. Each such domain will have one or more name servers. Each name server will, or will not, provide DNS Push service. If one or more servers provide DNS Push service, then the DNSSD browser will, when browsing, attempt to connect to one of the DNS Push servers for that browsing domain, until a successful connection is established or failure is detected. If failure is detected, or if there are no DNS Push servers, the DNSSD browser will use DNS datagrams [RFC1034] to browse that domain.

6. Security Considerations

Multicast DNS provides no mechanism for trust establishment other than the common connection to a shared link. DNSSD browsers are required to treat information about local authority datasets that are advertised using mDNS skeptically. The requirement in section [???] to validate

7. Informative References

Authors' Addresses

Ted Lemon
Apple Inc.
One Apple Park Way
Cupertino, California 95014
United States of America

Email: mellon@fugue.com

Joey Deng
Apple Inc.
One Apple Park Way
Cupertino, California 95014
United States of America

Email: deng.qiaoyu@gmail.com

Additional contact information:

邓乔宇
Apple Inc.
One Apple Park Way
Cupertino, California 95014
United States of America