Workgroup: DNSSD
Published: 4 March 2024
Intended Status: Standards Track
Expires: 5 September 2024
Authors: S. Cheshire T. Lemon
Apple Inc. Apple Inc.

Advertising Proxy for DNS-SD Service Registration Protocol

Abstract

An Advertising Proxy advertises the contents of a DNS zone, for example maintained using the DNS-SD Service Registration Protocol (SRP), using multicast DNS. This allows legacy clients to discover services registered with SRP using multicast DNS.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://dnssd-wg.github.io/draft-ietf-dnssd-advertising-proxy/draft-ietf-dnssd-advertising-proxy.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-dnssd-advertising-proxy/.

Discussion of this document takes place on the DNSSD Working Group mailing list (mailto:dnssd@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/dnssd/. Subscribe at https://www.ietf.org/mailman/listinfo/dnssd/.

Source for this draft and an issue tracker can be found at https://github.com/dnssd-wg/draft-ietf-dnssd-advertising-proxy.

Status of This Memo

Table of Contents

1. Introduction

DNS-Based Service Discovery [RFC6763] [ROADMAP] was designed to facilitate Zero Configuration IP Networking [RFC6760] [ZC]. When used with Multicast DNS [RFC6762] with ".local" domain names [RFC6761] this works well on a single link (a single broadcast domain).

However, in some applications, multicast may be a poor choice for advertising. Most obviously, multicast DNS is constrained to a single network link, and for example in the case of stub networks [STUBNET], service discovery for devices on the stub network by devices on the infrastructure network, or vice versa, requires some kind of proxy.

Also, even in single-link use cases, multicast isn't always the best choice. On some network media, multicast is inefficient and/or unreliable. Also, mDNS-based DNS-SD requires that each host providing services receive and process all mDNS service discovery messages, whether or not they are relevant to that host (for example, irrelevant service advertisements and queries for services not provided by the host). For power-constrained hosts, keeping a radio listening all the time for these messages is prohibitively expensive.

Ideally, in situations where multicast DNS is not the right choice, an obvious alternative is to use regular unicast DNS [RFC1035]. Unfortunately, this isn't always possible: the DNS protocol relies on a delegation hierarchy, and on per-network DNS resolvers.

The operational model for DNS service is that the infrastructure serving each network link provides a DNS resolver, and all DNS queries go to that resolver. Using unicast DNS for discovery of services through a stub network proxy, for example, would require that the stub network proxy be able to somehow register with the infrastructure DNS service. No standard mechanism for doing this currently exists.

This document describes a new type of proxy, an Advertising Proxy, which can be used to address some of these issues. An Advertising Proxy advertises the contents of some DNS zone (or zones) [RFC1034] to one or more network links using multicast DNS. This allows the DNS protocol, for example using the Service Registration Protocol registrar function [SRP], to be used by servers to advertise their services, while using the permissionless model of multicast DNS to make those services discoverable to devices on links supported by the Advertising Proxy.

One way of providing this service discovery functionality is through an Advertising Proxy. An advertising proxy functions to replicate some or all of the contents of a DNS zone using multicast DNS. This is limited by the fact that the DNS zone is a dataset, and the set of names discoverable in mDNS is constructed cooperatively, so the advertising proxy is not authoritative in the same sense that a DNS server is authoritative for any particular zone. This means that in some cases a name that is present in the DNS zone may conflict with a name already published by some other mDNS responder.

1.1. Conventions and Terminology Used in This Document

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in

BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Advertising Proxy

The advertising proxy works by publishing records from a DNS zone using multicast DNS [RFC6762]. The set of records published may include every record in the DNS zone, or a subset determined either automatically or administratively. When a record published in the DNS zone has the same name as a record published by some device that is not the advertising proxy, this will produce a conflict, and the advertising proxy must address this conflict.

Although we will in some cases refer to aspects of the mDNS protocol in this document, it is worth emphasizing that the typical implementor of an advertising proxy should not expect to have to do their own mDNS implementation: most platforms already provide a library that allows records and/or services to be published using mDNS. So our goal in discussing mDNS protocol details is to motivate choices that the implementor will make in their use of these libraries.

The simplest implementation of an advertising proxy will simply act on some signal indicating that the DNS zone has been updated. When this signal is received, the advertising proxy will terminate whatever mDNS registrations it is currently doing, and then iterate across the set of names in the zone, publishing all of the RRsets on each name in the DNS.

Of course, this simple approach to an advertising proxy has some issues. Multicast traffic on some link types (e.g., IEEE 802.11) consumes substantially more airtime than unicast traffic, so efforts should be made to minimize such traffic where possible. Also, the act of unpublishing an advertised record can be seen by consumers of the information in that record as an indication that whatever host or service that record represents is no longer present.

For these reasons, it is better if the advertising proxy can notice changes to the DNS zone, and only remove records from mDNS when they have been removed from the zone. New records in the zone will be newly advertised, of course, just as with the previous approach.

2.1. Mapping non-'.local.' domain names to '.local.'

Multicast DNS supports advertising of arbitrary names. However, the mechanism by which names in the DNS hierarchy are found is DNS, not mDNS, with one exception: names ending in '.local.'. Consequently, we can't simply publish names using mDNS with the same name that they have in the DNS hierarchy, because queries for those names will use DNS rather than mDNS.

But what does it mean to rewrite a name?

2.1.1. The DNS Dataset

DNS has the concept of a "zone," which is a subset of the domain name hierarchy over which some DNS server or servers assert authority. That authority is confirmed because a name server that is authoritative for a zone higher in the domain name hierarchy publishes a delegation naming the DNS server or servers that are authoritative for the subdomain.

As an example, the root ('.') zone has a set of authoritative servers that advertise it, and contains delegations for "top-level domains" like 'com.'. 'com.' is also a zone. However, it is not necessarily the case that a zone exists at only one level of the DNS hierarchy. A zone can include more than one level. For example, consider the reverse zone 'ip6.arpa.'. Here the subdomain will generally cover at least an IPv6 64-bit prefix.

So if we consider an IPv6 prefix '2001:db8:1234:5678::/64', the reverse zone for this will likely be '8.7.6.5.4.3.2.1.8.b.d.0.1.0.0.2.ip6.arpa.'. There will most likely be 16 more labels below the zone cut that are still part of the zone.

For the DNS Service Discovery use case, this is quite common. A service instance name in mDNS will typically look something like 'hostname._example._udp.<domain>.' A hostname on the other hand will look like 'hostname.<domain>.' So we need to distinguish between the name of the zone in which a name is advertised, the set of subdomains that actually represent additional domains in which names can be advertised, and subdomains that are part of the structure of the name.

For this reason, rather than speaking of DNS zones here, it would be more accurate to use another term that represents the <domain> part of the names above. We will use the term 'dataset' here, since this term is also used in the SRP Replication document [REPLICATION].

This means that an advertising proxy acts as a proxy for one or more datasets, rather than one or more zones. Which datasets the advertising proxy advertises with mDNS will either be determined automatically, or explicitly configured. It's entirely possible for the advertising proxy to act as a proxy for 'example.com.' and 'foo.example.com.'. Both 'example.com' and 'foo.example.com' are dataset names in this example.

2.1.2. How to rewrite names

Before advertising a record, the Advertising Proxy MUST rewrite the record. This means that both the owner name of the record and any

names that appear as part of the record must be rewritten if they are subdomains of the domain name of the DNS dataset. Of course the owner name will always be a subdomain of the DNS dataset name, but this won't always be true for the content of the resource record.

It may also make sense specifically for the PTR record to do a different rewrite for the owner name than for the target of the PTR record, depending on the way that we choose to rewrite the names. There are three ways that we can rewrite names, depending on the context:

*Rewrite the dataset name directly to '.local.'. This has the benefit of simplicity. When the data for which the advertising proxy is acting as proxy is DNSSD data, for any service that is being advertised, there will be a set of PTR records of the form '_example._transport.<domain>.'. If we rewrite <domain> to 'local', then a browse for the service '_example._transport' in the local domain will return the services being advertised.

The downside to this approach is that it doesn\'t safely handle name conflicts: if there are two datasets for which one or more advertising proxies are acting as proxy, and each zone contains a device with the name 'george', then this will produce a conflict that can't easily be resolved: the name is necessarily unique in each dataset, but when the two datasets are both mapped into the '.local.' namespace, they are in conflict. There is no way to resolve this.

Consider an RR that might appear in a dataset with a domain name of 'default.service.arpa.': '_example._transport.default.service.arpa. IN PTR hostname._example._transport.default.service.arpa.'. This would be rewritten to '_example._transport.local IN PTR hostname._example._transport.local.'.

*Use a dataset-specific subdomain of '.local.' for each dataset. For example, when we are maintaining the dataset using SRP, the SRP dataset will have a unique identifier, which we can in principle use here. So the domain into which we would rewrite the records from this dataset would then take the binary dataset ID and probably encode it as hexadecimal, producing '<dataset-id>.local.' as the target domain.

There is a problem with this approach however: devices browsing for services in '.local.' will not know to also browse in this subdomain of '.local.'. In order to fix this, the owner name of the PTR records being advertised will have to be rewritten into the '.local.' domain rather than '<dataset-id>.local.'. This is

perfectly okay, however, since these PTR records are not required to be unique.

If the domain being proxied has PTR records that are not being used to advertise services, and all PTR records are rewritten into the '.local.' domain, this could be a problem. However, since the primary use case for the advertising proxy is DNS Service Discovery, and additionally since PTR records generally are only used for service discovery and reverse lookups, it should be safe to rewrite PTR records to '.local.' for zones that are not reverse lookup zones. Reverse lookups are already documented in [RFC6762] and should not generally require an advertising proxy.

As an example of the PTR rewrite, if the dataset identifier is 5c4d2e9ab4 and the dataset domain name is default.service.arpa, the record '_example._transport.default.service.arpa. IN PTR hostname._example._transport.default.service.arpa' would be rewritten to '_example._transport.local IN PTR hostname._example._transport.5c4d2e9ab4.local.'.

*Append '.local.' to the dataset's domain name. For cases where there is no SRP replication dataset name, the dataset's domain name can be used in the same way. The reason not to do this is simply that it results in a longer name, and could theoretically run afoul of domain name length limits. The rewrite here would be from, e.g., 'hostname.default.service.arpa.' to 'hostname.default.service.arpa.local.'.

As with the dataset-specific subdomain of '.local.', the owner name for PTR records should be rewritten directly into '.local.'. So for example, the service advertisement in the earlier example, assuming a dataset domain name of 'default.service.arpa.', would be rewritten from '_example._transport.default.service.arpa. IN PTR hostname._example._transport.default.service.arpa' to '_example._transport.local IN PTR hostname._example._transport.default.service.arpa.local.'.

2.1.3. Handling of address records that are not global in scope

In some cases, the SRP requestor may register one or more address records for addresses that aren't valid or reachable on some link on which the advertising proxy could advertise them. The Advertising Proxy function MAY filter out such records entirely, or MAY explicitly advertise such records only on the link(s) on which they are reachable. This is optional because it requires the Advertising Proxy function to have enough information to make such a determination, which may not always be the case.

Where such determinations are possible, the advertising proxy SHOULD NOT advertise an IPv4 or IPv6 link-local address, or any other media-specific link-scoped address, on any link other than the link on which the SRP registration was received.

However, when a determination as to the link on which a particular address record is valid isn't being made, either because this capability isn't implemented by the advertising proxy, or because that information isn't available, the advertising proxy MUST advertise locally-scoped address.

2.1.4. Conflicts and Stale Data

There are two issues that can come up when advertising a DNS zone using mDNS that appear quite similar on the surface, but are actually fairly different. These both show up as a "conflict," but in fact only one is actually a conflict.

A conflict occurs when two different devices assert authority for the same name. However, stale data can also appear as a conflict, even though there is only one device asserting authority for the data. The problem of stale data occurs when we have more than one advertising proxy replicating the same DNS dataset.

For actual conflicts, there is no need to do anything special. Either the advertising proxy will prevail, or the other mDNS service will. If the advertising proxy does not prevail, it can attempt again to advertise the record after some reasonable interval has passed. This could either be the next time the record in question is updated (e.g. because of an SRP lease renewal) or a fixed interval.

mDNS specifies that conflicts should be resolved by renaming. Instead of continuing to try to claim the name that is in conflict, the advertising proxy MAY rename following the mDNS renaming method. In this case, the mapping between the new name and the name as it appears in the DNS dataset must be maintained until such time as the conflict no longer exists. So this approach requires maintaining some state.

If the advertising proxy renames owner names into a subdomain of '.local' rather than into '.local.', then actual conflicts should never occur, since what is being proxied is a single dataset. So when we encounter an apparent conflict using these models, it can't be an actual conflict, but rather stale data, and then the question is simply what to do next.

It should be the case in such situations that the problem will correct itself over time. Some advertising proxy will win the apparent conflict, and so some version of the data will be findable. However, following the usual mDNS conflict detection strategy, it is

most likely that a conflict will be resolved in favor of the stale data, not the new data. This can result in persistent stale data being advertised by the advertising proxy.

To prevent stale data winning over updated data, advertising proxies MUST support the Time Since Registered EDNS0 option [TSR].

2.1.5. No Text-Encoding Translation

As with a Discovery Proxy [RFC8766], an Advertising Proxy does no translation between text encodings [RFC6055]. Specifically, an Advertising Proxy does no translation between Punycode encoding [RFC3492] and UTF-8 encoding [RFC3629], either in the owner name of DNS records or anywhere in the RDATA of DNS records (such as the RDATA of PTR records, SRV records, NS records, or other record types like TXT, where it is ambiguous whether the RDATA may contain DNS names). All bytes are treated as-is with no attempt at text-encoding translation. A server implementing DNS-based Service Discovery [RFC6763] will use UTF-8 encoding for its unicast DNS-based record registrations, which the Advertising Proxy passes through without any text-encoding translation to the Multicast DNS subsystem. Queries from peers on the configured multicast-capable interface are answered directly from the advertised data without any text-encoding translation.

2.1.6. No Support for Reconfirm

For network efficiency, Multicast DNS [RFC6762] uses fairly long record lifetimes (typically 75 minutes). When a client is unable to reach a service that it discovered, Multicast DNS provides a "reconfirm" mechanism that enables the client to signal to the Multicast DNS subsystem that its cached data may be suspect, which causes the Multicast DNS subsystem to reissue queries, and remove the stale records if the queries are not answered.

Similarly, when using unicast service discovery with a Discovery Proxy [RFC8766], the DNS Push Notifications [RFC8765] protocol provides the RECONFIRM mechanism to signal that the Discovery Proxy should perform a local Multicast DNS reconfirm operation to re-verify the validity of the records.

When an Advertising Proxy is used, to support legacy clients that only implement Multicast DNS, reconfirm operations have no effect. If a device uses unicast Service Registration Protocol [SRP] to register its services with a service registry with Advertising Proxy capability, and the device then gets disconnected from the network, the Advertising Proxy will continue to advertise those records until the registrations expire. If a client discovers the service instance using Multicast DNS and is unable to reach it, and uses a Multicast

DNS reconfirm operation to re-verify the validity of the records, then the Advertising Proxy will continue to answer on behalf of the departed device until the record registrations expire. The Advertising Proxy has no reliable way to determine whether the additional Multicast DNS queries are due to a reconfirm operation, or due to other routine causes, like a client being rebooted, or disconnecting and then reconnecting to the network. The service registry has no reliable automatic way to determine whether a device that registered records has failed or disconnected from the network. Particularly with sleepy battery powered devices, the service registry does not know what active duty cycle any given service is expected to provide.

Consequently, reconfirm operations are not supported with an Advertising Proxy using multicast DNS. In cases where use of the reconfirm mechanism is important, clients should be upgraded to use the unicast DNS Push Notifications [RFC8765] protocol's RECONFIRM message. This RECONFIRM message provides an unambiguous signal to the service registry that it may be retaining stale records. (A future update to the Service Registration Protocol document [SRP] will consider ways that this unambiguous signal can be used to trigger expedited removal of stale data.)

3. Security Considerations

An Advertising Proxy may made data visible to eavesdroppers on the configured multicast-capable link(s).

4. IANA Considerations

This document has no IANA actions.

5. References

5.1. Normative References

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

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/

RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6760] Cheshire, S. and M. Krochmal, "Requirements for a Protocol to Replace the AppleTalk Name Binding Protocol (NBP)", RFC 6760, DOI 10.17487/RFC6760, February 2013, <https://www.rfc-editor.org/info/rfc6760>.

[RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", RFC 6761, DOI 10.17487/RFC6761, February 2013, <https://www.rfc-editor.org/info/rfc6761>.

[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>.

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

[RFC8765] Pusateri, T. and S. Cheshire, "DNS Push Notifications", RFC 8765, DOI 10.17487/RFC8765, June 2020, <https://www.rfc-editor.org/info/rfc8765>.

[SRP] Lemon, T. and S. Cheshire, "Service Registration Protocol for DNS-Based Service Discovery", Work in Progress, Internet-Draft, draft-ietf-dnssd-srp-25, 4 March 2024, <https://datatracker.ietf.org/api/v1/doc/document/draft-ietf-dnssd-srp/>.

[TSR] Lemon, T. and L. Qin, "Multicast DNS conflict resolution using the Time Since Received (TSR) RR", Work in Progress, Internet-Draft, draft-tllq-tsr-03, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-tllq-tsr-03>.

5.2. Informative References

[RFC3492] Costello, A., "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492, DOI 10.17487/RFC3492, March 2003, <https://www.rfc-editor.org/info/rfc3492>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/info/rfc3629>.

[RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, DOI 10.17487/RFC5625, August 2009, <https://www.rfc-editor.org/info/rfc5625>.

[RFC6055] Thaler, D., Klensin, J., and S. Cheshire, "IAB Thoughts on Encodings for Internationalized Domain Names", RFC 6055, DOI 10.17487/RFC6055, February 2011, <https://www.rfc-editor.org/info/rfc6055>.

[RFC7558] Lynn, K., Cheshire, S., Blanchet, M., and D. Migault, "Requirements for Scalable DNS-Based Service Discovery (DNS-SD) / Multicast DNS (mDNS) Extensions", RFC 7558, DOI 10.17487/RFC7558, July 2015, <https://www.rfc-editor.org/info/rfc7558>.

[RFC8766] Cheshire, S., "Discovery Proxy for Multicast DNS-Based Service Discovery", RFC 8766, DOI 10.17487/RFC8766, June 2020, <https://www.rfc-editor.org/info/rfc8766>.

[ROADMAP] Cheshire, S., "Service Discovery Road Map", Work in Progress, Internet-Draft, draft-cheshire-dnssd-roadmap-03, 23 October 2018, <https://datatracker.ietf.org/doc/html/draft-cheshire-dnssd-roadmap-03>.

[REPLICATION] Lemon, T., Keshavarzian, A., and J. Hui, "Automatic Replication of DNS-SD Service Registration Protocol Zones", Work in Progress, Internet-Draft, draft-ietf-dnssd-srp-replication-01, 28 February 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-dnssd-srp-replication-01>.

[STUBNET] Lemon, T. and J. Hui, "Automatically Connecting Stub Networks to Unmanaged Infrastructure", Work in Progress, Internet-Draft, draft-ietf-snac-simple-03, 30 January 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-snac-simple-03>.

[ZC] Cheshire, S. and D. H. Steinberg, "Zero Configuration Networking: The Definitive Guide", O'Reilly Media, Inc., ISBN 0-596-10100-7, December 2005.

Authors' Addresses

Stuart Cheshire
Apple Inc.
One Apple Park Way
Cupertino, California 95014
United States of America

Phone: +1 (408) 996-1010
Email: cheshire@apple.com

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

Phone: +1 (408) 996-1010
Email: elemon@apple.com