

dnssd
Internet-Draft
Intended status: Informational
Expires: April 23, 2014

S. Bhandari
B. Fajalia
R. Schmieder
S. Orr
A. Dutta
Cisco
October 20, 2013

Extending multicast DNS across local links in Campus and Enterprise
networks
draft-bhandari-dnssd-mdns-gateway-00

Abstract

This document describes the requirements for extending multicast DNS in enterprise networks. It provides an overview of a solution to extend multicast DNS services across links that have been implemented in routers, switches and wireless LAN controllers.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 <http://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 April 23, 2014.

Copyright Notice

Copyright (c) 2013 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 (<http://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

1. Introduction	3
1.1. Requirements	3
2. Conventions and Terminology Used in this Document	4
3. Solution overview	5
3.1. Service Cache	5
3.2. Service Filters	6
3.3. Service Announcement	6
3.4. Service Query	7
3.5. Service Probing	7
3.6. Service update, Service withdrawal	7
3.7. Service Refresh	7
3.8. mDNS Gateway for Wireless Network	8
3.8.1. Advertising services on wireless networks	8
3.8.2. Device Tracking	8
3.8.3. Mobility Considerations	9
3.8.4. mDNS traffic optimization	9
4. Challenges	10
5. Future work	10
6. IANA Considerations	11
7. Security Considerations	11
8. Acknowledgements	11
9. Normative References	11
Authors' Addresses	12

1. Introduction

Service discovery using multicast DNS (mDNS) as defined in [RFC6762] is limited in scope to L3 boundaries due to the use of link-local scoped multicast addresses. Networks are partitioned into multiple segments by means of virtual local area networks (VLANs) or subnet creation for various reasons. The need for network wide, seamless service discovery demands the extension of the discovery protocol beyond the L3 boundary. There are also challenges in wireless networks (802.11, 802.15.4 etc) where a large number multicast messages can impact wireless performance.

Enabling Service Discovery across L3 boundaries can be accomplished in one of the following ways using existing, unmodified protocols:

1. Unicast DNS-SD only: Use of DNS servers and allowing clients to use dynamic DNS updates and Long Lived Queries (LLQs) to announce and learn services dynamically [I-D.sekar-dns-llq]
2. mDNS only: Defining a mDNS gateway entity at the L3 boundaries extending service advertisements/discovery across the links it is attached to
3. Combination of unicast DNS and mDNS - Hybrid proxy approach as described in [I-D.cheshire-mdnsexthybrid]
4. mDNS utilizing extended scope multicast - Modifying mDNS to use a wider scope multicast address for message exchange

As a first step, this draft lists out the approach to use a mDNS gateway on a network element (2) to extend the service advertisement/discovery across network segments attached to the element. While this approach does not preclude (1) or (3), it allows the extension of service discovery in a limited number of segments with minimal provisioning. Approach (4) is not explored further as it would add to the flood of service discovery messages in the scope defined by the multicast address and it would also require changes on mDNS clients, which is undesirable.

1.1. Requirements

This section describes requirements for extending multicast DNS in an enterprise environment:

1. Extend service discovery across L3 boundaries
2. Defining and enforcing a policy to selectively filter services that are to be extended based on service type, service instance,

location of the provider/user, role of the device or user accessing/offering the service.

3. Defining and enforcing a policy to selectively filter queries and announcements from specific clients or over specific network links
4. Filtering of link-local-only information - Services that resolve to IPv4 and IPv6 link-local addresses only must not be extended beyond the local link. Suppression of resource records that contain link-local-only addresses from propagation beyond the local link
5. Optimization of mDNS queries/advertisements in wireless networks
6. Effectively handle roaming of mobile devices (changes in the Point of Attachment). Especially if those devices advertise services
7. Limit the services in response to queries with a subset of the services by geographic proximity
8. Handle conflict resolution of service instances and host names across the links where the service is extended
9. Protection of resources (memory and CPU) of the network element that participates in extending multicast DNS
10. Audit, logging of services that are denied based on policy

2. Conventions and Terminology Used in this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

This document uses the multicast DNS and DNS terminology conventions from [RFC6762] and [RFC6763]. It uses the same convention for services on the same link as defined in [I-D.cheshire-mdnsexthybrid], repeated here for quick reference:

Multicast DNS works between a hosts on the same link. A set of hosts is considered to be "on the same link", if:

when any host A from that set sends a packet to any other host B in that set, using unicast, multicast, or broadcast, the entire link-

layer packet payload arrives unmodified, and

a broadcast sent over that link by any host from that set of hosts can be received by every other host in that set

The link-layer **header** may be modified, such as in Token Ring Source Routing [802.5], but not the link-layer **payload**. In particular, if any device forwarding a packet modifies any part of the IP header or IP payload then the packet is no longer considered to be on the same link. This means that the packet may pass through devices such as repeaters, bridges, hubs or switches and still be considered to be on the same link for the purpose of this document, but not through a device such as an IP router that decrements the TTL or otherwise modifies the IP header.

- o mDNS gateway - An application that listens to services and extends the services across links

3. Solution overview

The solution introduces the mDNS gateway function which is co-located on the network element that connects to multiple links, typically an IP router. The mDNS gateway function will be responsible for:

- o Caching - Learn and cache services. Maintain services in the cache according to service announcements, service removals and the TTL of the records.
- o Respond to queries - Advertise in response to queries with services in the cache that are not in the same link-local domain where the query is received.
- o Service filtering - Filter services to be added to the cache and to be included in the advertisements as per configured policies.
- o Service redistribution - forwarding of unsolicited service announcements across links based on configuration
- o Active query - Service queries sent by the mDNS gateway itself to learn about services or keep services 'fresh' in the cache. Can be sent on one or more of the links the gateway is attached to.

3.1. Service Cache

The mDNS gateway maintains a database of DNS Resource Records (RR) required to advertise and resolve services. At a minimum, the cache will contain PTR, SRV, TXT and A/AAAA RRs for each service with NSEC

RR support for optimization. In addition, the link on which the service and host originate is also maintained in the cache. Records in the cache are refreshed based on TTL expiry.

3.2. Service Filters

A service filtering policy is configured with an action to permit or deny services into the cache or to filter services included in the response/advertisement messages based on matching criteria. The matching criteria can be defined based on:

- o Service type
- o Service instance names
- o Link on which the message is received
- o Type of message - query or advertisement
- o Location of the host querying or advertising a service

A Service filtering policy is applied for incoming and outgoing messages. A unique filtering policy can be applied Globally or per link.

When a mDNS message is received by the mDNS gateway matching an action set for the link, the policy match is then executed. The incoming advertisement is processed against the mDNS gateway inbound filtering policy applied on the link where the advertisement is received. If the action is 'permit' the service is added to the cache. If a response or advertisement is to be sent out, the outbound filtering policy applied on the interface is processed and if the resulting action is 'deny' then the service and its corresponding RRs are not included in the message sent out.

3.3. Service Announcement

The mDNS gateway listens for all service announcements. When a service announcement is received, the announcement and all the additional RRs learnt are added to the cache or ignored based on the result of the configured inbound filter policy.

The RRs containing link-local information e.g. A or AAAA RRs that contain link-local scoped IPv4 or IPv6 addresses are not stored in the cache.

When the mDNS gateway learns a service it can also forward the advertisement on other attached links.

3.4. Service Query

The mDNS gateway processes all queries against the configured filtering policy. If the response to the query is permitted then it constructs the answers and additional records required to resolve the service from its cache for the services that are permitted. Services that reside on the same link where the query is received are not included as the owner of the service will also see the query and would send the response directly. Only services learnt from different links are considered in the response.

Any query received for additional RRs to resolve the service e.g. query for SRV, A, AAAA etc are responded to in the same way. If the records do not exist in the cache due to expiry or purging of cache for any other reason, mDNS gateway sends out an explicit query to fetch the records on the link where the service resides.

3.5. Service Probing

According to [RFC6762] before registering a service, RR probing is performed to ensure unique names. As the mDNS gateway maintains a cache of all the RRs that are extended across the links it responds to probe records like any other query. This will help in detecting and resolving name space conflicts across links where service discovery has been extended.

3.6. Service update, Service withdrawal

When the mDNS gateway receives a service update or withdrawal it updates or removes the service and all corresponding records from its cache. It forwards the withdraw messages to other attached links.

3.7. Service Refresh

The RRs describing the service and resolving it have a TTL that defines the validity of the RR. The mDNS gateway can continuously refresh each of the RRs in the cache as per the TTL rules. For the purpose of optimization, the mDNS gateway can rely on the host interested in the RRs to trigger a refresh by setting the TTLs in the response to the time remaining since the record was learnt by the mDNS gateway. If a client is interested in the RR then it would trigger a refresh when a fraction of the TTL is reached. While responding to queries from hosts, the mDNS gateway inturn sends out queries to refresh the records that are about to expire on the source link where the records were learnt.

3.8. mDNS Gateway for Wireless Network

Deploying the mDNS gateway in wireless networks has a few additional requirements w.r.t to multicast radio optimization and mobility aspects. This section describes some additional capabilities added to the mDNS gateway to satisfy these requirements.

3.8.1. Advertising services on wireless networks

In order to conserve wireless bandwidth, the mDNS gateway sends service advertisements as L2 unicast messages to wireless devices .

In a wireless network, the mDNS gateway co-located on the network element that is providing the wireless service can act as a passive device and respond only if wireless clients send a mDNS query. When bridging is turned off the mDNS gateway and the Layer 2 optimization is enabled, the mDNS gateway will need to send the query response as layer 2 unicast messages even when the provider is on the same link as the requestor. Bridging of mDNS messages can be turned off based on configuration. This is useful in the following scenario:

1. Save computation resources on the device which are used to replicate the multicast packet as a L2 unicast for all wireless clients.
2. If the wireless client is in power saving mode, sending a mDNS advertisement as a L2 unicast would forcefully awake the client and it would result into more power consumption by the wireless client.

mDNS functionality is not impacted by acting as a passive gateway because the client would always send the mDNS query when inquiring for a service.

3.8.2. Device Tracking

Wireless clients are mobile in nature. The mDNS gateway should learn the service instance only from the authenticated wireless client. The mDNS gateway should tag each service instance from a wireless client with the client's MAC address. This MAC address should be used for device tracking. If the wireless client leaves the network, the mDNS gateway should not wait until the TTL expires but it should actively clean up the service instance provided by that wireless client. This is done to protect the mDNS gateway cache resources as well as to keep other clients from hearing about services that are no longer connected to the network..

3.8.3. Mobility Considerations

Wireless deployments supports seamless mobility. In such a scenario, the mDNS gateway needs to be aware of the client location. If the location changes, the mDNS gateway needs to update its mDNS cache. The mDNS gateway should tag each service instance with the device location. The device location can be derived based on the Access Point (AP) to which the wireless client is attached. If the client, which is providing any service, changes its location, this change needs to be reflected in the mDNS gateway. If the client roams from one mDNS gateway to another mDNS gateway, then the old gateway should provide the service instance information pertaining to the roamed client to the new gateway and then it must clear the mDNS cache for that particular client. If the mDNS gateway is not acting as a passive gateway, it may choose to update the network about the new service instance it has learnt.

3.8.4. mDNS traffic optimization

All mDNS packets are sent to the multicast link-local IP address. When the mDNS gateway starts forwarding the mDNS advertisements across L3 boundaries then the number of such advertisement on any network would increase. Wireless networks should be optimized for the increase in mulitcast traffic that will be generated by extending the service advertisement domain. If there are many mDNS packets going on air then it would impact other data traffic. Hence mDNS traffic optimization is required. One method of optimization the mDNS gateway could implement is sending the query reponse back as a L2 unicast to the requesting client.

When services are advertised, each record has an associated TTL value. When the TTL expires, the gateway needs to send a query (at 85%, 90% and 95% of the TTL) for that record to confirm its validity. If the TTL value of each record is different, then mDNS gateway needs to send a query for individual records. To minimize the mDNS traffic, queries for multiple RRs for that service record set can be initiated towards the source of the service. Such a query can be sent with the QU bit set as described in [RFC6762] to solicit a unicast response.

The mDNS gateway for wireless networks should act as a passive gateway as explained in Section 3.8.1. When it is acting as a passive gateway and bridging of mDNS packets is turned off it has to respond to queries on the link even when the provider of the service resides on the same link.

4. Challenges

This section lists out limitations and challenges faced as part of the the solution described in this draft.

1. Name conflict resolution across links: Name conflict resolution depends on probing followed by service registration. This is done by the host which is providing the service. Name conflict resolution across links depends on the mDNS gateway cache to have a conclusive list of names already present to be able to authoritatively respond to probe requests. However, this may not always be possible due to timing issues when the cache gets updated, records having expired from the cache etc.
2. Multi-homed hosts: There is also the case of a multihomed host connected via multiple links to the same mDNS gateway that may end up wrongly assuming conflict and getting into a continuous renaming loop.
3. Multiple mDNS gateways on the link: If there are multiple mDNS gateways enabled on the same link queries may get duplicate responses.
4. Loops in the network: If there is a loop in the network with multiple mDNS gateways enabled in such a topology it may end up continuously cycling the service around the loop and keeping the RRs alive forever.
5. Refreshing resource records: Balancing an excessive number of queries to maintain the records in the cache vs. having the cache up-to-date with all the known record names requires optimizations that may lead to corner cases where wrong results or conflicts arise.

5. Future work

The solution documented here is limited to extending services across links attached to a single network element or mDNS gateway. For a broader application, the service discovery solution described in [I-D.cheshire-mdnsxt-hybrid] should be realized with any provisioning as needed.

Similar to auto provisioning and realization of the hybrid proxy approach for homenet as described in [I-D.stenberg-homenet-dnssdext-hybrid-proxy-ospf] a solution needs to be built for enterprise and campus networks extending what has been described in this draft.

There are other considerations such as including the location information so that services can be ordered based on proximity of the service.

6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

7. Security Considerations

N/A

8. Acknowledgements

9. Normative References

- [I-D.cheshire-mdnsext-hybrid]
Cheshire, S., "Hybrid Unicast/Multicast DNS-Based Service Discovery", draft-cheshire-mdnsext-hybrid-02 (work in progress), July 2013.
- [I-D.sekar-dns-llq]
Sekar, K., "DNS Long-Lived Queries",
draft-sekar-dns-llq-01 (work in progress), August 2006.
- [I-D.stenberg-homenet-dnssdext-hybrid-proxy-ospf]
Stenberg, M., "Hybrid Unicast/Multicast DNS-Based Service Discovery Auto-Configuration Using OSPFv3",
draft-stenberg-homenet-dnssdext-hybrid-proxy-ospf-00 (work in progress), June 2013.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service

Discovery", RFC 6763, February 2013.

Authors' Addresses

Shwetha Bhandari
Cisco Systems, Inc.
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: shwethab@cisco.com

Bhavik Fajalia
Cisco Systems, Inc.
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: bfajalia@cisco.com

Ralph Schmieder
Cisco Systems, Inc.
City Plaza - 4th Floor
Stuttgart, BADEN-WURTEMBERG 70178
Germany

Email: rschmied@cisco.com

Stephen Orr
Cisco Systems, Inc.
1 Paragon Drive
Montvale, NJ 07645
USA

Email: sorr@cisco.com

Amit Dutta
Cisco Systems, Inc.
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: amdutta@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: July 26, 2014

S. Cheshire
Apple Inc.
January 22, 2014

Hybrid Unicast/Multicast DNS-Based Service Discovery
draft-cheshire-dnssd-hybrid-01

Abstract

Performing DNS-Based Service Discovery using purely link-local Multicast DNS enables discovery of services that are on the local link, but not (without some kind of proxy or similar special support) of services that are outside the local link. Using a very large local link with thousands of hosts improves service discovery, but at the cost of large amounts of multicast traffic.

Performing DNS-Based Service Discovery using purely Unicast DNS is more efficient, but requires configuration of DNS Update keys on the devices offering the services, which can be onerous for simple devices like printers and network cameras.

Hence a compromise is needed, that provides easy service discovery without requiring either large amounts of multicast traffic or onerous configuration.

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 <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 26, 2014.

Copyright Notice

Copyright (c) 2014 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 (<http://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

1. Introduction	3
2. Conventions and Terminology Used in this Document	3
3. Hybrid Proxy Operation	4
4. Implementation Status	9
5. IPv6 Considerations	11
6. Security Considerations	11
7. Intellectual Property Rights	11
8. IANA Considerations	11
9. Acknowledgments	11
10. References	12
10.1. Normative References	12
10.2. Informative References	12
Author's Address	13

1. Introduction

Multicast DNS [RFC6762] and its companion technology DNS-based Service Discovery [RFC6763] were created to provide IP networking with the ease-of-use and autoconfiguration for which AppleTalk was well known [RFC6760] [ZC].

Section 10 ("Populating the DNS with Information") of the DNS-SD specification [RFC6763] discusses possible ways that a service's PTR, SRV, TXT and address records can make their way into the DNS namespace, including manual zone file configuration [RFC1034] [RFC1035], DNS Update [RFC2136] [RFC3007] and proxies.

This document specifies a type of proxy called a Hybrid Proxy that uses Multicast DNS [RFC6762] to discover Multicast DNS records on its local link, and makes corresponding DNS records visible in the Unicast DNS namespace.

2. 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 "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

Multicast DNS works between a hosts on the same link. A set of hosts is considered to be "on the same link", if:

- o when any host A from that set sends a packet to any other host B in that set, using unicast, multicast, or broadcast, the entire link-layer packet payload arrives unmodified, and
- o a broadcast sent over that link by any host from that set of hosts can be received by every other host in that set

The link-layer **header** may be modified, such as in Token Ring Source Routing [802.5], but not the link-layer **payload**. In particular, if any device forwarding a packet modifies any part of the IP header or IP payload then the packet is no longer considered to be on the same link. This means that the packet may pass through devices such as repeaters, bridges, hubs or switches and still be considered to be on the same link for the purpose of this document, but not through a device such as an IP router that decrements the TTL or otherwise modifies the IP header.

3. Hybrid Proxy Operation

In its simplest form, each local link in an organization is assigned a unique Unicast DNS domain name, such as "Building 1.example.com." or "4th Floor.Building 1.example.com." (Grouping multiple local links under the same Unicast DNS domain name is to be specified in a future companion document, but for the purposes of this document, assume that each link has its own unique Unicast DNS domain name.)

Each link in an organization has a Hybrid Proxy which serves it. This function could be performed by a router on that link, or, with appropriate VLAN configuration, a single Hybrid Proxy could have a logical presence on, and serve as the Hybrid Proxy for, multiple links. In the organization's DNS server, NS records are used to delegate ownership of each defined link name (e.g., "Building 1.example.com.") to the Hybrid Proxy which serves that link.

Domain Enumeration PTR records [RFC6763] are also created to inform clients of available Device Discovery domains, e.g.,:

b._dns-sd._udp.example.com.	PTR	Building 1.example.com.
	PTR	Building 2.example.com.
	PTR	Building 3.example.com.
	PTR	Building 4.example.com.
lb._dns-sd._udp.example.com.	PTR	Building 1.example.com.

When a DNS-SD client issues a Unicast DNS query to discover services in a particular Unicast DNS (e.g., "_printer._tcp.Building 1.example.com. PTR ?") the normal DNS delegation mechanism results in that query being served from the delegated authoritative name server for that subdomain, namely the Hybrid Proxy on the link in question. Like a conventional Unicast DNS server, a Hybrid Proxy implements the usual Unicast DNS protocol [RFC1034] [RFC1035] over UDP and TCP. However, unlike a conventional Unicast DNS server that generates answers from the data in its manually-configured zone file, a Hybrid Proxy generates answers by performing a Multicast DNS query (e.g., "_printer._tcp.local. PTR ?") on its local link, and then, from the data in the Multicast DNS replies it receives, generating the corresponding Unicast DNS reply.

3.1. Data Translation

Generating the corresponding Unicast DNS reply involves, at the very least, rewriting the "local" suffix to the appropriate Unicast DNS domain (e.g., "Building 1.example.com").

In addition it would be desirable to suppress Unicast DNS replies for records that are not useful outside the local link. For example, DNS A and AAAA records for IPv4 link-local addresses [RFC3927] and IPv6 link-local addresses [RFC4862] should be suppressed. Similarly, for sites that have multiple private address realms [RFC1918], private addresses from one private address realm should not be communicated to clients in a different private address realm.

By the same logic, DNS SRV records that reference target host names that have no addresses usable by the requester should be suppressed, and likewise, DNS PTR records that point to DNS names with DNS SRV records that reference target host names that have no addresses usable by the requester should be also be suppressed.

The same reachability requirement for advertised services also applies to the Hybrid Proxy itself. The mechanism specified in this document only works if the Hybrid Proxy is reachable from the client making the request.

3.1.1.1. Application-Specific Data Translation

There may be cases where Application-Specific Data Translation is appropriate.

For example, AirPrint printers tend to advertise fairly verbose information about their capabilities in their DNS-SD TXT record. This information is a legacy from LPR printing, because LPR does not have in-band capability negotiation, so all of this information is put in the DNS-SD TXT record instead. IPP printing does have in-band capability negotiation, but for convenience printers tend to include the same capability information in their IPP DNS-SD TXT records as well. For local mDNS use this extra TXT record information is inefficient, but not fatal. However, when a Hybrid Proxy aggregates data from multiple printers on a link, and sends it via unicast (via UDP or TCP) this amount of unnecessary TXT record information can result in large replies. Therefore, a Hybrid Proxy that is aware of the specifics of an application-layer protocol such as Apple's AirPrint (which uses IPP) can elide unnecessary key/value pairs from the DNS-SD TXT record for better network efficiency.

Note that this kind of Application-Specific Data Translation is expected to be very rare. It is the exception, rather than the rule. This is an example of a common theme in computing. It is frequently the case that it is wise to start with a clean, layered design, with clear boundaries. Then, in certain special cases, those layer boundaries may be violated, where the performance and efficiency benefits outweigh the inelegance of the layer violation.

As in other similar situations, these layer violations optional. They are done only for efficiency reasons, and are not required for correct operation. A Hybrid Proxy can operate solely at the mDNS layer, without any knowledge of DNS-SD semantics, or of any DNS-SD client semantics.

3.2. Answer Aggregation

In a simple analysis, simply gathering multicast answers and forwarding them in a unicast reply seems adequate, but it raises the question of how long the Hybrid Proxy should wait to be sure that it has received all the Multicast DNS replies it needs to form a complete Unicast DNS reply. If it waits too little time, then it risks its Unicast DNS reply being incomplete. If it waits too long, then it creates a poor user experience at the client end.

This dilemma is solved by use of DNS Long-Lived Queries (DNS LLQ) [I-D.sekar-dns-llq]. The Hybrid Proxy replies immediately to the Unicast DNS query using the Multicast DNS records it already has in its cache (if any). This provides a good client user experience by providing a near-instantaneous response. Simultaneously, the Hybrid Proxy issues a Multicast DNS query on the local link to discover if there are any additional Multicast DNS records it did not already know about. Should additional Multicast DNS replies be received, these are then delivered to the client using DNS LLQ update messages. The timeliness of such LLQ updates is limited only by the timeliness of the device responding to the Multicast DNS query. If the Multicast DNS device responds quickly, then the LLQ update is delivered quickly. If the Multicast DNS device responds slowly, then the LLQ update is delivered slowly. The benefit of using LLQ is that the Hybrid Proxy can respond promptly because it doesn't have to delay its unicast reply to allow for the expected worst-case delay for receiving all the Multicast DNS replies. Even if a proxy were to try to provide reliability by assuming an excessively pessimistic worst-case time (thereby giving a very poor user experience) there would still be the risk of a slow Multicast DNS device taking even longer than that (e.g, a device that is not even powered on until ten seconds after the initial query is received) resulting in incomplete replies. Using LLQs solves this dilemma: even very late replies are not lost; they are delivered in subsequent LLQ update messages.

There are two factors that determine specifically how replies are generated. The first factor is whether the Hybrid Proxy already has at least one record in its cache that positively answers the question. The second factor is whether the query from the client includes the LLQ option (typical with long-lived service browsing PTR queries) or not (typical with one-shot operations like SRV or address record queries).

- o No answer in cache; no LLQ option: Do local mDNS query three times, and then return NXDOMAIN if no answer after three tries.
- o No answer in cache; with LLQ option: As above, do local mDNS query three times, and then return NXDOMAIN if no answer after three tries. However, the query remains active for as long as the client maintains the LLQ state, and if mDNS answers are received later, LLQ update messages are sent. (Reasoning: We don't need to rush to send an empty answer.)
- o At least one answer in cache; no LLQ option: Send reply right away to minimise delay. No local mDNS queries are performed. (Reasoning: Given RRSets TTL harmonisation, if the proxy has one answer in its cache, it should have all of them.)
- o At least one answer in cache; with LLQ option: As above, send reply right away to minimise delay. However, the query remains active for as long as the client maintains the LLQ state, and if additional mDNS answers are received later, LLQ update messages are sent. (Reasoning: We want UI that is displayed very rapidly, yet continues to remain accurate even as the network environment changes.)

4. Implementation Status

Some aspects of the mechanism specified in this document already exist in deployed software. Some aspects are new. This section outlines which aspects already exist and which are new.

4.1. Already Implemented and Deployed

Domain enumeration discovery by the client (the "b._dns-sd._udp" queries) is already implemented and deployed.

Unicast queries to the indicated discovery domain is already implemented and deployed.

These are implemented and deployed in Mac OS X 10.4 and later (including all versions of Apple iOS, on all iPhone and iPads), in Bonjour for Windows, and in Android 4.1 "Jelly Bean" (API Level 16) and later.

Domain enumeration discovery and unicast querying have been used for several years at IETF meetings to make Terminal Room printers discoverable from outside the Terminal room. When you Press Cmd-P on your Mac, or select AirPrint on your iPad or iPhone, and the Terminal room printers appear, that is because your client is doing unicast DNS queries to the IETF DNS servers.

4.2. Partially Implemented

The current APIs make multiple domains visible to client software, but most client UI today lumps all discovered services into a single flat list. This is largely a chicken-and-egg problem. Application writers were naturally reluctant to spend time writing domain-aware UI code when few customers today would benefit from it. If Hybrid Proxy deployment becomes common, then application writers will have a reason to provide better UI. Existing applications will work with the Hybrid Proxy, but will show all services in a single flat list. Applications with improved UI will group services by domain.

The Long-Lived Query mechanism [I-D.sekar-dns-llq] referred to in this specification exists and is deployed, but has not been standardized by the IETF. It is possible that the IETF may choose to standardize a different or better Long-Lived Query mechanism. In that case, the pragmatic deployment approach would be for vendors to produce Hybrid Proxies that implement both the deployed Long-Lived Query mechanism [I-D.sekar-dns-llq] (for today's clients) and a new IETF Standard Long-Lived Query mechanism (as the future long-term direction).

4.3. Not Yet Implemented

The translating/filtering Hybrid Proxy specified in this document. Once implemented, such a Hybrid Proxy will immediately make wide-area discovery available with today's existing clients and devices.

A mechanism to 'stitch' together multiple ".local." zones so that they appear as one. Such a mechanism will be specified in a future companion document.

5. IPv6 Considerations

An IPv4-only host and an IPv6-only host behave as "ships that pass in the night". Even if they are on the same Ethernet, neither is aware of the other's traffic. For this reason, each physical link may have **two** unrelated ".local." zones, one for IPv4 and one for IPv6. Since for practical purposes, a group of IPv4-only hosts and a group of IPv6-only hosts on the same Ethernet act as if they were on two entirely separate Ethernet segments, it is unsurprising that their use of the ".local." zone should occur exactly as it would if they really were on two entirely separate Ethernet segments.

It will be desirable to have a mechanism to 'stitch' together these two unrelated ".local." zones so that they appear as one. Such mechanism will need to be able to differentiate between a dual-stack (v4/v6) host participating in both ".local." zones, and two different hosts, one IPv4-only and the other IPv6-only, which are both trying to use the same name(s). Such a mechanism will be specified in a future companion document.

6. Security Considerations

A service proves its presence on a local link by its ability to answer link-local multicast queries on that link. If greater security is desired, then the Hybrid Proxy mechanism should not be used, and something with stronger security should be used instead, such as authenticated secure DNS Update [RFC2136] [RFC3007].

7. Intellectual Property Rights

Apple has submitted an IPR disclosure concerning the technique proposed in this document. Details are available on the IETF IPR disclosure page [IPR2119].

8. IANA Considerations

This document has no IANA Considerations.

9. Acknowledgments

Thanks to Markus Stenberg for helping develop the policy regarding the four styles of unicast reply.

10. References

10.1. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, May 2005.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, December 2012.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, December 2012.
- [I-D.sekar-dns-llq] Sekar, K., "DNS Long-Lived Queries", draft-sekar-dns-llq-01 (work in progress), August 2006.

10.2. Informative References

- [IPR2119] "Apple Inc.'s Statement about IPR related to Hybrid Unicast/Multicast DNS-Based Service Discovery", <<https://datatracker.ietf.org/ipr/2119/>>.
- [RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.
- [RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.
- [RFC6760] Cheshire, S. and M. Krochmal, "Requirements for a Protocol

to Replace the AppleTalk Name Binding Protocol (NBP)",
RFC 6760, December 2012.

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

Author's Address

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

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

DNS-SD/mDNS Extensions
Internet-Draft
Intended status: Informational
Expires: August 17, 2014

K. Lynn, Ed.
Consultant
S. Cheshire
Apple, Inc.
M. Blanchet
Viagenie
D. Migault
Orange
February 13, 2014

Requirements for Scalable DNS-SD/mDNS Extensions
draft-ietf-dnssd-requirements-01

Abstract

DNS-SD/mDNS is widely used today for discovery and resolution of services and names on a local link, but there are use cases to extend DNS-SD/mDNS to enable service discovery beyond the local link. This document provides a problem statement and a list of requirements.

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 <http://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 August 17, 2014.

Copyright Notice

Copyright (c) 2014 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 (<http://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

1. Introduction	2
2. Problem Statement	3
3. Basic Use Cases	5
4. Requirements	6
5. Namespace Considerations	7
6. Security Considerations	8
7. IANA Considerations	9
8. Acknowledgments	9
9. References	9
Authors' Addresses	11

1. Introduction

DNS-Based Service Discovery [DNS-SD] in combination with its companion technology Multicast DNS [mDNS] is widely used today for discovery and resolution of services and names on a local link. However, as users move to multi-link home or campus networks they find that mDNS does not work across routers. DNS-SD can also be used in conjunction with conventional unicast DNS to enable wide-area service discovery, but this capability is not yet widely deployed. This disconnect between customer needs and current practice has led to calls for improvement, such as the Educause petition [EP].

In response to this and similar evidence of market demand, several products now enable service discovery beyond the local link using different ad-hoc techniques. However, it is unclear which approach represents the best long-term direction for DNS-based service discovery protocol development.

DNS-SD/mDNS in its present form is also not optimized for network technologies where multicast transmissions are relatively expensive. Wireless networks such as [IEEE.802.11] may be adversely affected by excessive mDNS traffic due to the higher network overhead of multicast transmissions. Wireless mesh networks such as 6LoWPAN [RFC4944] are effectively multi-link subnets where multicasts must be forwarded by intermediate nodes.

It is in the best interests of end users, network administrators, and vendors for all interested parties to cooperate within the context of the IETF to develop an efficient, scalable, and interoperable standards-based solution.

This document defines the problem statement and gathers requirements for Scalable DNS-SD/mDNS Extensions.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

1.2. Terminology and Acronyms

Service: An endpoint (host and port) for a given application protocol. Services are identified by Service Instance Names.

DNS-SD: DNS-Based Service Discovery, as specified in [DNS-SD], is a conventional application of DNS Resource Records and messages to facilitate the discovery and location of services.

mDNS: Multicast DNS, as specified in [mDNS], is a transport protocol that facilitates DNS-SD on a local link in the absence of DNS infrastructure.

SSD: Scalable DNS-SD is a future extension of DNS-SD/mDNS that meets the requirements set forth in this document.

Scope of Discovery: A node in a local or global namespace, e.g., a DNS zone, that is the target of a given DNS-SD query.

Zero Configuration: A set of technologies including DNS-SD/mDNS that enable local address and name assignment in the absence of DHCP or DNS infrastructure. May also refer more generally to a deployment of SSD that requires no administration.

Incremental Deployment: An orderly transition, as a network installation evolves, from DNS-SD/mDNS to SSD.

2. Problem Statement

Service discovery beyond the local link is perhaps the most important feature currently missing from the DNS-SD/mDNS framework. Other issues and requirements are summarized below.

2.1. Multi-link Naming and Discovery

A list of desired DNS-SD/mDNS improvements from network administrators in the research and education community was issued in

the form of the Educause petition [EP]. The following is a summary of the technical issues:

- o Products that advertise services such as printing and multimedia streaming via DNS-SD/mDNS are not currently discoverable by devices on other links. It is common practice for enterprises and institutions to use wireless links for client access and wired networks for server infrastructure, typically on different subnets. DNS-SD used with conventional unicast DNS does work when devices are on different links, but the resource records that describe the service must somehow be entered into the unicast DNS namespace.
- o Entering DNS-SD records manually into a unicast DNS zone file works, but requires a DNS administrator to do that and is fragile when IP addresses of devices change dynamically, as is common when DHCP is used.
- o Automatically adding DNS-SD records using DNS Update works, but requires that the DNS server be configured to allow DNS Updates, and requires that devices be configured with the DNS Update credentials to permit such updates, which has proven to be onerous.
- o Therefore, a mechanism is desired that populates the DNS namespace with the appropriate DNS-SD records with less manual administration than typically needed for a unicast DNS server.

The following is a summary of the technical requirements:

- o It must scale to a range of hundreds to thousands of DNS-SD/mDNS enabled devices in a given environment.
- o It must simultaneously operate over a variety of network link technologies, such as wired and wireless networks.
- o It must not significantly increase network traffic (wired or wireless).
- o It must be cost-effective to manage at up to enterprise scale.

2.2. IEEE 802.11 Wireless LANs

Multicast DNS was originally designed to run on Ethernet - the dominant link-layer at the time. In shared Ethernet networks, multicast frames place little additional demand on the shared network medium compared to unicast frames. In IEEE 802.11 networks however, multicast frames are transmitted at a low data rate supported by all

receivers. In practice, this data rate leads to a larger fraction of airtime being devoted to multicast transmission. Some network administrators block multicast traffic or convert it to a series of link-layer unicast frames.

Wireless links may be orders of magnitude less reliable than their wired counterparts. To improve transmission reliability, the IEEE 802.11 MAC requires positive acknowledgement of unicast frames. It does not, however, support positive acknowledgement of multicast frames. As a result, it is common to observe much higher loss of multicast frames on wireless as compared to wired network technologies.

Enabling service discovery on IEEE 802.11 networks requires that the number of multicast frames be restricted to a suitably low value, or replaced with unicast frames to use the MAC's reliability features.

2.3. Low Power and Lossy Networks (LLNs)

Emerging wireless mesh networking technologies such as RPL [RFC6550] and 6LoWPAN present several challenges for the current DNS-SD/mDNS design. First, Link-Local multicast scope [RFC4291] is defined as a single-hop neighborhood. A single subnet prefix in a wireless mesh network may often span multiple links, therefore a larger multicast scope is required to span it [I-D.ietf-6man-multicast-scopes]. mDNS is not currently specified for greater than Link-Local scope.

Additionally, low-power nodes may be offline for significant periods either because they are "sleeping" or due to connectivity problems. In such cases LLN nodes might fail to respond to queries or defend their names using the current design.

3. Basic Use Cases

The following use cases are defined with different characteristics to help motivate, distinguish, and classify the target requirements. They cover a spectrum of increasing deployment and administrative complexity.

(A) Personal Area networks: the simplest example of a DNS-SD/mDNS network may consist of a single client and server, e.g., one laptop and one printer, on a common link. Such networks may not contain a router, but instead use Zero Configuration to mitigate the lack of infrastructure.

(B) Classic home networks, consisting of:

- * Single exit router: the network may have multiple upstream providers or networks, but all outgoing and incoming traffic goes through a single router.
- * One-level depth: multiple links on the network are bridged to form a single subnet, which is connected to the default router.
- * Single administrative domain: all nodes under the same admin entity. (However, this does not necessarily imply a network administrator.)

(C) Advanced home and small business networks
[I-D.ietf-homenet-arch]:

Like B but consist of multiple wired and/or wireless links, connected by routers, behind the single exit router. However, the forwarding nodes are largely self-configuring and do not require routing protocol administration. Such networks should also not require DNS administration.

(D) Enterprise networks:

Like C but consist of arbitrary network diameter under a single administrative domain. A large majority of the forwarding and security devices are configured.

(E) Higher Education networks:

Like D but core network may be under a central administrative domain while leaf networks are under local administrative domains.

(F) Mesh networks such as RPL/6LoWPAN:

Multi-link subnets with prefixes defined by one or more border routers. May comprise any part of networks C, D, or E.

4. Requirements

Any successful SSD solution(s) will have to strike the proper balance between competing goals such as scalability, deployability, and usability. With that in mind, none of the requirements listed below should be considered in isolation.

REQ1: The scope of the discovery should be either automatically determined by the discovering devices or configured (selected) in the case of multiple choices.

- REQ2: For use cases A, B, and C, there should be a zero configuration mode of operation.
- REQ3: For use cases D and E, there should be a way to configure the scope of the discovery and also support both smaller (e.g., department) and larger (e.g., campus-wide) discovery scopes.
- REQ4: For use cases D and E, there should be an incremental way to deploy the solution.
- REQ5: SSD should integrate or at least should not break any current link scope DNS-SD/mDNS protocols and deployments.
- REQ6: SSD must be capable of spanning multiple links (hops) and network technologies.
- REQ7: SSD must be scalable to thousands of nodes with minimal configuration and without degrading network performance. A possible figure of merit is that, as the number of services increases, the amount of traffic due to SSD on a given link remains relatively constant.
- REQ8: SSD should enable a way to provide a consistent user experience whether local or global services are being discovered.
- REQ9: The information presented by SSD should reflect reality. That is, new information should be available in a timely fashion and stale information should not persist.

5. Namespace Considerations

The unicast DNS namespace contains globally unique names. The mDNS namespace contains locally unique names. Clients discovering services may need to differentiate between local and global names or to determine that names in different namespaces identify the same service.

SSD should support rich internationalized labels within Service Instance Names, as DNS-SD/mDNS does today. SSD must not negatively impact the global DNS namespace or infrastructure.

The problem of publishing local services in the global DNS namespace may be generally viewed as exporting local resource records and their associated labels into some DNS zone. The issues related to defining labels that are interoperable between local and global namespaces are discussed in [I-D.sullivan-dnssd-mdns-dns-interop].

6. Security Considerations

Insofar as SSD may automatically gather DNS-SD resource records and publish them over a wide area, the security issues are likely to be the union of those discussed in [mDNS] and [DNS-SD]. The following sections highlight potential threats that are posed by deploying DNS-SD over multiple links or by automating DNS-SD administration.

6.1. Scope of Discovery

As mDNS is currently restricted to a single link, the scope of the advertisement is limited, by design, to the shared link between client and server. In a multi-link scenario, the owner of the advertised service may not have a clear indication of the scope of its advertisement.

If the advertisement propagates to a larger set of links than expected, this may result in unauthorized clients (from the perspective of the owner) connecting to the advertised service. It also discloses information (about the host and service) to a larger set of potential attackers.

If the scope of the discovery is not properly setup or constrained, then information leaks will happen outside the appropriate network.

6.2. Multiple Namespaces

There is a possibility of conflicts between the local and global DNS namespaces. Without adequate feedback, a client may not know if the target service is the correct one, therefore enabling potential attacks. [Example? KEL]

6.3. Authorization

DNSSEC can assert the validity but not the veracity of records in a zone file. The trust model of the global DNS relies on the fact that human administrators either a) manually enter resource records into a zone file, or b) configure the DNS server to authenticate a trusted device (e.g., a DHCP server) that can automatically maintain such records.

An imposter may register on the local link and appear as a legitimate service. Such "rogue" services may then be automatically registered in wide area DNS-SD.

6.4. Authentication

Up to now, the "plug-and-play" nature of mDNS devices has relied only on physical connectivity. If a device is visible via mDNS then it is assumed to be trusted. This is no longer likely to be the case in larger networks.

If there is a risk that clients may be fooled by the deployment of rogue services, then application layer authentication should probably be considered.

6.5. Privacy Considerations

Mobile devices such as smart phones that can expose the location of their owners by registering services in arbitrary zones pose a risk to privacy. Such devices must not register their services in arbitrary zones without the approval of their operators. However, it should be possible to configure one or more "safe" zones, e.g., based on subnet prefix, in which mobile devices may automatically register their services.

7. IANA Considerations

This document currently makes no request of IANA.

Note to RFC Editor: this section may be removed upon publication as an RFC.

8. Acknowledgments

We gratefully acknowledge contributions and review comments made by RJ Atkinson, Tim Chown, Guangqing Deng, Ralph Droms, Educause, David Farmer, Matthew Gast, Peter Van Der Stok, and Thomas Narten.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.

- [RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.
- [mDNS] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.
- [DNS-SD] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.

9.2. Informative References

- [I-D.ietf-6man-multicast-scopes]
Droms, R., "IPv6 Multicast Address Scopes", draft-ietf-6man-multicast-scopes-02 (work in progress), November 2013.
- [I-D.ietf-homenet-arch]
Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", draft-ietf-homenet-arch-11 (work in progress), October 2013.
- [I-D.sullivan-dnssd-mdns-dns-interop]
Sullivan, A., "Requirements for Labels to Interoperate Between mDNS and DNS", draft-sullivan-dnssd-mdns-dns-interop-00 (work in progress), January 2014.
- [EP] "Educause Petition", <https://www.change.org/petitions/from-educause-higher-ed-wireless-networking-admin-group>, July 2012.
- [IEEE.802.11]
"Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE Std 802.11-2012, 2012, <<http://standards.ieee.org/getieee802/download/802.11-2012.pdf>>.
- [static] "Manually Adding DNS-SD Service Discovery Records to an Existing Name Server", July 2013, <<http://www.dns-sd.org/ServerStaticSetup.html>>.

Authors' Addresses

Kerry Lynn (editor)
Consultant

Phone: +1 978 460 4253
Email: kerlyn@ieee.org

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

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

Marc Blanchet
Viagenie
246 Aberdeen
Quebec , Quebec G1R 2E1
Canada

Email: Marc.Blanchet@viagenie.ca
URI: <http://www.viagenie.ca>

Daniel Migault
Orange
38-40 rue du General Leclerc
Issy-les-Moulineaux 92130
France

Phone: +33 1 45 29 60 52
Email: mglb.biz@gmail.com

dnssd
Internet-Draft
Intended status: Informational
Expires: August 18, 2014

D. Otis
Trend Micro
February 14, 2014

mDNS X-link review
draft-otis-dnssd-mdns-xlink-02

Abstract

Multicast DNS will not normally extend beyond the MAC Bridge. Such limitations are problematic when desired services are beyond the reach of multicast mDNS. This document explores options for overcoming this limitation.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

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 <http://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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 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 (<http://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

1. Introduction	3
2. Possible Solutions	3
2.1. Selective Forwarding based on IGMP or MLD snooping	3
2.2. RBridge	4
2.3. L2TP VPN	4
2.4. VLAN	4
2.5. Convert mDNS to DNS	5
3. IANA Considerations	6
4. Security Considerations	6
5. Acknowledgements	7
6. References - Informative	7
Author's Address	9

1. Introduction

mDNS [RFC6762] normally allows MAC entities to make their services known on MAC Bridged LANs without use of centralized discovery services. Multicast limits the range of this publication to LANs able to forward mDNS frames. A Bridge is a mechanism transparent to end stations on LANs interconnected by Bridges designated to forward frames normally through participation in a Spanning Tree Algorithm.

A Bridge forwards frames based on prior source MAC associations with incoming frames on different LAN ports. Source MAC and LAN port associations are recommended to expire in 300 seconds. Frames containing source multicast MAC are silently discarded as invalid. Frames containing a destination MAC on the same LAN port already associated with the MAC are silently discarded. A valid incoming frame with a destination not previously associated with a different LAN port is forwarded (flooded) to all other LAN ports, otherwise when a MAC destination address is associated with a different LAN port from which the frame was received, the frame is selectively forwarded to this port. All broadcast and multicast MAC are flooded to all other LAN ports because the MAC does not represent a valid source. Flooding operation may create a storm of replicated frames having an unknown MAC destination whenever forwarding is enabled on LAN ports connected in a loop.

In IEEE 802.11 wireless networks, multicast frames are transmitted at a low data rate supported by all receivers. Multicast on wireless networks may thereby lower overall network throughput. Some network administrators block multicast traffic or convert it to a series of link-layer unicast frames.

Wireless links may be orders of magnitude less reliable than their wired counterparts. To improve transmission reliability, the IEEE 802.11 MAC requires positive acknowledgement of unicast frames. It does not, however, support positive acknowledgement of multicast frames. As a result, it is common to observe much higher loss of multicast frames on wireless compared against wired network technologies.

2. Possible Solutions

2.1. Selective Forwarding based on IGMP or MLD snooping

Internet Group Management Protocol (IGMP) [RFC3376] supports multicast on IPv4 networks. Multicast Listener Discovery (MLD) [RFC3810] supports multicast management on IPv6 networks using ICMPv6 messaging in contrast to IGMP's bare IP encapsulation. This

management allows routers to announce their multicast membership to neighboring routers. To optimize which LANs receive forwarded multicast frames, IGMP or MLD snooping can be used to determine the presence of listeners as a means to permit selective forwarding of multicast frames.

2.2. RBridge

RBridges [RFC6325] are compatible with previous IEEE 802.1 customer bridges as well as IPv4 and IPv6 routers and end nodes. RBridges may support either IEEE 802.3 or other link technologies. RBridges are invisible to current IP routers as bridges are and, like routers, terminate the Bridge spanning tree protocol. The RBridge design supports VLANs and optimization of the distribution of multi-destination frames based on VLAN ID or on IP-derived multicast groups. It also allows unicast forwarding tables at transit RBridges to be sized according to the number of RBridges (rather than the number of end nodes), which allows their forwarding tables to be substantially smaller than in conventional customer bridges.

[RFC3927] provides an overview of IPv4 address complexities related with dealing with multiple segments and interfaces. IPv6 introduces new paradigms in respect to interface address assignments which offer scoping as explained in [RFC4291]. The use of RBridge has the capacity of greatly simplifying this environment while also eliminating bottlenecks imposed by a Spanning Tree Algorithm.

If it can be determined an additional layer can be added within RBridge to implement selective multicast forwarding, input for this extension should be defined to assist with mDNS management.

2.3. L2TP VPN

L2TP VPN [RFC3931] with experimental [RFC4045] attempt to handle multicast by mitigating redundant traffic which remains fairly problematic.

2.4. VLAN

There are several products being introduced into the market that attempt to solve the problem stated in the charter. They normally use VLAN [RFC5517] to selectively extend multicast forwarding beyond Bridge limitations. This does not represent a general solution but can support specific services being offered by dynamic devices within a local IP address space.

2.5. Convert mDNS to DNS

Rather than using MAC as an exchange basis, IP addresses made visible by DNS [RFC1035] that conform with [RFC6763] can be used instead. Direct access to an IP address is better assured with a single DHCP [RFC2131] or [RFC3315] server for IPv4 and IPv6 respectively that responds to interconnected networks. In such a configuration, it is possible to have DHCP indicate which DNS server is to be used as a means to offer combined local and Internet namespace.

Automation needed to populate the information published in DNS normally depends on Kerberos [RFC4120] and LDAP [RFC2251] servers supporting either a campus or corporate network.

Automated conversion of mDNS into unicast DNS can be problematic from a security standpoint as can the propagation of multicast frames. mDNS only requires compliance with [RFC5198] rather than IDNA2008 [RFC5895]. This means mDNS does not ensure instances are visually unique and may contain spaces and punctuation not permitted by IDNA2008. mDNS also permits name compression of SRV target names that DNS currently does not ensure support.

Public Suffix lists might help simplify the creation of A-Labels from UTF-8 user input by offering matching items for user selection. A Public Suffix list represents DNS domain names reserved for registrations by appropriate authorities. This still leaves the domain registered above the public suffix, but its validation should involve fewer transactions.

Replacing ASCII punctuation and spaces in the label with the '_' character, except when located as the leftmost character, may reduce some handling issues related to end of string parsing, since labels in DNS normally do not contain spaces or punctuation. Nevertheless, DNS is able to handle such labels within sub-domains of registered domains.

Services outside the ".local." domain may have applications obtaining domain search lists provided by DHCP ([RFC2131] and [RFC3315] for IPv4 and IPv6 respectively or RA DNSSL [RFC6106] also for IPv6. Internet domains need to be published in DNS as A-Labels [RFC3492] because IDNA2008 compliance depends on A-label enforcement by registrars. Therefore A-Labels and not U-Labels must be published in DNS for Internet domains at this time. There is also a DNS extension to support the live browse feature found in mDNS.

The SRV scheme used by mDNS has also been widely adopted in the Windows OS since it offered a functional replacement for Windows Internet Name Service (WINS) as their initial attempt which lacked

sufficient name hierarchy.

It is unknown whether sufficient filtering of mDNS to expose just those services likely needed will sufficiently protect wireless networks. The extent RBridge use and something analogous to IGMP or MLD for selective forwarding might help to mitigate otherwise spurious traffic is unknown.

Open source of corporate server implementations based on a Debian distro are currently available with plug-ins able to support Windows and OS X.

2.5.1. Reliable Wireless Multicast

[RFC6951] transport protocol was designed to efficiently exchange frames rather than byte streams. It can operate with partial reliability [RFC3758] while still allowing receivers to detect and request specific lost frames. This might be possible while also using multicast MACs and IP Addresses. This protocol currently has not been structured to support multicast. This transport also extends the DNS 16 bit transactional nonce not even present in mDNS with an additional 32 bit random session ID.

3. IANA Considerations

This document requires no IANA consideration.

4. Security Considerations

Layer 2 Bridging that might be used to extend mDNS is not inherently secure. See [RFC6325] for a list of possible concerns and mitigation methods.

Conveying both the MAC and IP address beyond the LAN may enable attacks that would have otherwise been prevented.

Moving mDNS services into DNS MUST only publish services able to withstand this greater exposure.

Any query for a name ending with ".local." MUST be resolved using mDNS.

It is not uncommon for CPE equipment's DNS settings being maliciously modified. Often this equipment does not create or retain settings logs, where a reset or power cycling removes evidence of tampering.

Establishing ".local." as the first domain offered in a domain search list could ensure local services receive higher priority, but such a priority could also permit local spoofing of services otherwise resolved using DNS. A priority on local resolution may also result in a 3 second additional delay for global resolutions.

5. Acknowledgements

The authors wish to acknowledge valuable contributions from the following: Dave Rand, Michael Tuexen

6. References - Informative

- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC2251] Wahl, M., Howes, T., and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [RFC3492] Costello, A., "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492, March 2003.
- [RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, May 2004.

- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, May 2005.
- [RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.
- [RFC4045] Bourdon, G., "Extensions to Support Efficient Carrying of Multicast Traffic in Layer-2 Tunneling Protocol (L2TP)", RFC 4045, April 2005.
- [RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.
- [RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.
- [RFC5517] HomChaudhuri, S. and M. Foschiano, "Cisco Systems' Private VLANs: Scalable Security in a Multi-Client Environment", RFC 5517, February 2010.
- [RFC5895] Resnick, P. and P. Hoffman, "Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008", RFC 5895, September 2010.
- [RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.
- [RFC6165] Banerjee, A. and D. Ward, "Extensions to IS-IS for Layer-2 Systems", RFC 6165, April 2011.
- [RFC6325] Perlman, R., Eastlake, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.

- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.
- [RFC6951] Tuexen, M. and R. Stewart, "UDP Encapsulation of Stream Control Transmission Protocol (SCTP) Packets for End-Host to End-Host Communication", RFC 6951, May 2013.

Author's Address

Douglas Otis
Trend Micro
10101 N. De Anza Blvd
Cupertino, CA 95014
USA

Phone: +1.408.257-1500
Email: doug_otis@trendmicro.com

IETF
Internet-Draft
Intended status: Informational
Expires: July 26, 2014

A. Sullivan
Dyn
January 22, 2014

Requirements for Labels to Interoperate Between mDNS and DNS
draft-sullivan-dnssd-mdns-dns-interop-00

Abstract

Despite its name, DNS-Based Service Discovery can use naming systems other than the Domain Name System when looking for services. Different name systems use different conventions for the characters allowed in any name. In order for DNS-SD to be used effectively in environments where multiple different name systems are in use, it is important to follow a common set of conventions for naming. This memo presents an outline of the requirements for selection of labels for mDNS and DNS when they are expected to interoperate in this manner.

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 <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 26, 2014.

Copyright Notice

Copyright (c) 2014 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 (<http://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

1. Introduction	2
1.1. Conventions and terms used in this document	3
2. Requirements for a profile for label interoperation	3
3. DNS-SD portions	4
3.1. The <Instance> Portion of the Service Instance Name .	4
3.2. The <Service> Portion of the Service Instance Name .	5
3.3. The <Domain> Portion of the Service Instance Name . .	5
4. Acknowledgements	5
5. IANA Considerations	5
6. Security Considerations	5
7. Informative References	5
Author's Address	6

1. Introduction

DNS-Based Service Discovery (DNS-SD, [RFC6763]) specifies a mechanism for discovering services using queries both to the Domain Name System (DNS, [RFC1034], [RFC1035]) and to Multicast DNS (mDNS, [RFC6762]). Conventional use of the DNS generally follows the host name rules [RFC0952] for labels -- the so-called LDH rule. That convention is the reason behind the development of Internationalized Domain Names for Applications (IDNA2008, [RFC5890], [RFC5891], [RFC5892], [RFC5893], [RFC5894], [RFC5895]). It is worth noting that the LDH rule is a convention, and not a strict rule of the DNS. It is assumed to be true widely enough, however, that in many circumstances names cannot be used unless they cleave to the LDH rule.

At the same time, mDNS requires that labels be encoded in UTF-8, and permits a range of characters in labels that are not permitted by IDNA2008 or the LDH rule. For example, mDNS encourages the use of spaces and punctuation in mDNS names (see [RFC6763], section 4.1.3). It does not restrict which Unicode code points may be used in those labels, so long as the code points are UTF-8 in Net-Unicode [RFC5198] format.

Users of applications are, of course, frequently unconcerned with (not to say oblivious to) the name-resolution system(s) in service at any given moment, and are inclined simply to use the same names in different contexts. As a result, the same string might be tried as a name using different name resolution technologies. If DNS-SD is to be used in an environment where both mDNS and DNS are to be queried

for services, then the names to be queried will need to be compatible with the rules and conventions for both DNS and mDNS.

One approach to interoperability under these circumstances is to use a single operational convention for names under the different naming systems. This memo posits such a use profile, and outlines what is necessary to make it work.

1.1. Conventions and terms used in this document

Wherever appropriate, this memo uses the terminology defined in Section 2 of [RFC5890]. In particular, the reader is assumed to be familiar with the terms "U-label", "LDH label", and "A-label" from that document. Similarly, the reader is assumed to be familiar with the U+NNNN notation for Unicode code points used in [RFC5890] and other documents dealing with Unicode code points. In the interests of brevity and consistency, the definitions are not repeated here.

This memo refers to names in the DNS as though the LDH rule and IDNA2008 are strict requirements. They are not. DNS labels are, in principle, just collections of octets, and therefore in principle the LDH rule is not a constraint. In practice, applications often intercept labels that do not conform to the LDH rule and apply IDNA and other transformations.

The term "owner name" (common to the DNS vernacular) is used here to apply not just to the names to be looked up in the DNS, but to any name that might be looked up either in the DNS or using mDNS.

2. Requirements for a profile for label interoperation

Any interoperability between mDNS and DNS will require interoperability across some of the portions of a DNS-SD Service Instance Name (see Section 3) that are implicated in regular mDNS and DNS lookups. The open question is which of the portions are implicated. In any case, if a given portion is implicated, the profile will need to apply to all labels in that portion.

Because the profile will need to apply to names that might need to interoperate with names in the DNS, and because mDNS permits labels that IDNA does not, the profile will reduce the labels that may be used with mDNS. Consequently, some recommendations from [RFC6763] will not really be possible to implement using names subject to the profile. In particular, [RFC6763], section 4.1.3 recommends that rich text, human-readable labels be used, and includes punctuation and space characters in the examples. It is not clear whether such uses will be possible, because spaces and most punctuation are permitted neither in U-labels nor in LDH labels. In addition, the

same section recommends that labels always be stored and communicated as UTF-8, even in the DNS. Because IDNA2008 libraries will treat any Unicode-encoded labels as candidate U-labels and attempt to perform resolution in A-label form, the advice to store and transmit labels as UTF-8 in the DNS is likely to encounter problems. By contrast, mDNS normally uses UTF-8.

U-labels cannot contain upper case letters. That restriction extends to ASCII-range upper case letters that work fine in LDH-labels. It may be confusing that the character "A" works in the DNS when none of the characters in the label has a diacritic, but does not work when there is such a diacritic in the label. Labels in mDNS names may contain upper case characters, so the profile will need either to restrict the use of upper case or come up with a reliable and predictable convention for case folding.

3. DNS-SD portions

DNS-SD specifies three portions of the owner name for a DNS-SD resource record. These are the <Instance> portion, the <Service> portion, and the <Domain>. The owner name made of these three parts is called the Service Instance Name. It is worth observing that a portion may be more than one label long. See [RFC6763], section 4.1.

3.1. The <Instance> Portion of the Service Instance Name

[RFC6763] is clear that the <Instance> portion of the Service Instance Name is intended for presentation to users, and therefore virtually any character is permitted in it. There are two ways that a profile might address this portion; a specification of the profile will need to select one of these strategies.

The first option is to treat this portion as likely to be intercepted by system-wide IDNA-aware resolvers. In this case, the portion needs to be made subject to the profile, thereby curtailing what characters may appear in this portion. This approach permits DNS-SD to use any standard system resolver but presents inconsistencies with the DNS-SD specification and with DNS-SD that is exclusively mDNS-based.

The second option is to specify that the portion never be handled by "normal" DNS resolution, and that it instead be handled by a special DNS-SD resolution path. In this case, DNS-SD works as it always does, but at the cost of a possibly more complicated system-wide resolver or special resolution code built into the DNS-SD system.

3.2. The <Service> Portion of the Service Instance Name

DNS-SD includes a <Service> component in the Service Instance Name. This component is not really user-facing data, but is instead control data embedded in the Service Instance Name. This component includes so-called "underscore labels", which are labels prepended with U+005F (_). The underscore label convention was established by DNS SRV ([RFC2782]) for identifying metadata inside DNS names. A system-wide resolver (or DNS middlebox) that cannot handle underscore labels will not work with DNS-SD at all, so it is safe to suppose that such resolvers will not attempt to do special processing on these labels. Therefore, the <Service> portion of the Service Instance Name will not be subject to the profile.

3.3. The <Domain> Portion of the Service Instance Name

The <Domain> portion of the service instance name forms an integral part of the QNAME submitted for DNS resolution, and a system-wide resolver that is IDNA2008-aware is likely to interpret labels with UTF-8 in the QNAME as candidates for IDNA2008 processing. Therefore, these labels will need to be subject to the profile.

4. Acknowledgements

The author gratefully acknowledges the insights of Kerry Lynn.

5. IANA Considerations

This memo makes no requests of IANA.

6. Security Considerations

This memo presents some requirements for future development, but does not specify anything. Therefore, it has no implications for security.

7. Informative References

- [RFC0952] Harrenstien, K., Stahl, M., and E. Feinler, "DoD Internet host table specification", RFC 952, October 1985.
- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.
- [RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.
- [RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.
- [RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.
- [RFC5892] Faltstrom, P., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, August 2010.
- [RFC5893] Alvestrand, H. and C. Karp, "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)", RFC 5893, August 2010.
- [RFC5894] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale", RFC 5894, August 2010.
- [RFC5895] Resnick, P. and P. Hoffman, "Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008", RFC 5895, September 2010.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.

Author's Address

Andrew Sullivan
Dyn
150 Dow St.
Manchester, NH 03101
U.S.A.

Email: asullivan@dyn.com