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

Table of Contents

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-mdnsext-hybrid]

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-mdnsext-hybrid], 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.

- 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:

- Caching - Learn and cache services. Maintain services in the cache according to service announcements, service removals and the TTL of the records.

- 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.

- Service filtering - Filter services to be added to the cache and to be included in the advertisements as per configured policies.

- Service redistribution - forwarding of unsolicited service announcements across links based on configuration

- 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:

- Service type
- Service instance names
- Link on which the message is received
- Type of message - query or advertisement
- 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 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 optimazation is enabled, the mDNS gateway will need to send the query responseas 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 posible 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-mdnsext-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-WURTTEMBERG 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