Mutlticast DNS (mDNS)

Summary of Basic Functionalities

by *Antonios Atlasis*
aatlasis@secfu.net

This white paper is a selective extract of the mDNS IETF RFC 6762 aiming at providing summary of the basic functionalities of this protocol. It was used as a basis for my ongoing mDNS research, and it is uploaded for conveniences purposes for the community.

Introduction

As described in [RFC 6762], multicast DNS (mDNS) provides the ability to perform DNS-like operations on the local link in the absence of any conventional unicast DNS server. The option to use mDNS for names not ending in ".local" SHOULD[1] be supported, but it SHOULD be disabled by default.

The mDNS requests and responses are sent via UDP from source port 5353 to destination port 5353. If this is not the case regarding the source port, mDNS packets MUST silently be ignored. Moreover, all mDNS responses SHOULD be sent with IP TTL set to 255.

Placing multiple questions in a single query is allowed. Similarly, multiple responses can be sent in one multicast packet.

mDNS messages carried by UDP may be up to the IP MTU of the physical interface, less the space required for the IP header and the UDP header. If required, mDNS packets may even be fragmented in case of a single mDNS resource record that is too large to fit in a single MTU-sized multicast response packet. That is, an mDNS packet larger than the interface MTU which is sent using fragments MUST not contain more than one resource record. However, even when fragmentation is used, an mDNS packet MUST NOT exceed 9000 bytes.

mDNS Domains

Names ending in ".local" have only local significance; the same is true for the link-local reverse mapping domains "254.169.in-addr.arpa" and "8-b.e.f.ip6.arpa". Each IP link has its own private ".local". Any DNS query for a name ending with ".local" MUST be sent to the multicast addresses 224.0.0.251 and FF02::FB for IPv4 and IPv6 respectively.

However, according to [RFC 6763], when there are not conventional DNS servers available, DNS queries that do not end with ".local" MAY be sent to these multicast addresses. Resolving global names via local multicast is also allowed, although in such a case the use of DNSSEC is recommended.

1 When in capital, SHOULD, MUST, MAY, etc. are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" (RFC 2119).

Conceptually, an mDNS domain is a single DNS zone. Multicast DNS domains are not delegated from their parent domain via use of Name Server (NS) records, and there is no concept of delegation of subdomains with a multicast DNS domain.

mDNS has no SOA (Start of Authority) Records, whilst unsolicited Resource Record (RR) responses can be used to announce new records.

mDNS domains are reserved by IANA.

mDNS Responses

When responding to queries using qtype “ANY” (255) and/or qclass “ANY” (255), an mDNS responder MUST respond with ALL of its records that match the query. This is subtly different from how qtype “ANY” and qclass “ANY” work in unicast DNS, where at least one response is required to answer the corresponding query. .

As far as address queries are concerned, when an mDNS responder sends a multicast DNS response message containing its own address records, it MUST include all addresses that are valid on the interface, both IPv4 and IPv6 ones, link-locals and global unicast.

mDNS responses MUST NOT contain any questions in the Question Section.

As explained in section 6.1 of [RFC 6762], there can also be cases where a responder has to respond with asserting the nonexistence of a record (that is, sending back a negative response) using a DNS NSEC record [RFC 4034]. In this case the NSEC record is not used for its usual DNSSEC security properties (explained in [RFC 4034]), but, as a way of expressing which records do or do not exist with a given name. A responder MUST only generates negative responses to queries for which it has legitimate ownership of the name, rrtype and rrclass in the section.

Cache Maintenance

An mDNS querier typically takes the first response it receives. mDNS answers contain a Time-to-Live (TTL) value that indicate for how many seconds this answer is valid. An DNS TTL=0 indicates that the corresponding record has been deleted.

To perform cache maintenance, an mDNS querier should plan to retransmit its query after at least 50% of the record lifetime is elapsed. Specifically, it is recommended that the querier should plan to issue a query at 80% of the record lifetime, and then if no answer is received, at 85%, 90%, and 95%. If still a response is not received, the corresponding record is deleted when the TTL expires.

Direct Unicast Queries and Responses

The most usual questions request multicast responses and for this reason, they are referred as “QM” questions; instead, if they request unicast responses they are referred as “QU” questions. A special flag in each query, the QU bit, denotes if this is a unicast query (the QU bit is set), or not (the QU bit is not set). When receiving a question with the unicast-response bit set, a responder SHOULD usually

respond with a unicast packet directed back to the querier. It should be noted though that an mDNS querier MUST only accept unicast responses if they answer a recently sent query (e.g. within the last two seconds).

However, since it is possible for a unicast query to be received from a machine outside the local link, responders SHOULD check that the source address in the query packet matches the local subnet for that link and silently ignore the packet if not.

mDNS Probing and Announcing

Whenever an mDNS responder starts up, or its connectivity has changed for any reason, it MUST perform two steps: Probing and Announcing.

In Probing, the hosts sends an mDNS query asking to see if the resource records (e.g. a host's address record) going to announce are already in use. All probe queries SHOULD be done using query type “ANY” (255) to elicit answers. Moreover, a host can use a single message to probe for all of its resource records instead of needing a separate message for each. For example, a host can can simultaneously probe for uniqueness of its “A” record and all its SRV records in the same query message. When responding to queries using qtype “ANY” and/or qclass ANY, a multicast DNS responder MUST respond with ALL of its records that match the query. Any answer containing a record in question MUST be considered a conflicting one and the conflict MUST be resolved using a pre-defined in the RFC procedure (see paragraph).

A probe query can be distinguished from a normal query by the fact that a probe query contains a proposed record in the Authority section that answers the question in the Question section.

In the second step, Announcing, an mDNS responder sends unsolicited mDNS responses containing, in the Answer section, all of its newly registered resource records. Specifically, an mDNS responder MUST send at least two unsolicited responses, one second apart. To provide increased robustness, a responder MAY send up to eight unsolicited responses.

Conflicts and their Resolution

RFC 6762 also provides a conflict resolution mechanism.

During the Probing and Announcing process, when a probing mDNS is sent, other devices have just 750 msec to respond to defend their record. The probes SHOULD be sent as “QU” questions with the unicast response bit set, to allow a defending host to respond immediately via unicast. If a conflicting mDNS response is received, the probing host SHOULD choose new names for its conflicting records, as appropriate. If fifteen conflicts occur within any 10-second period, the host MUST wait ate least five seconds before each successive additional attempt. After one minute of probing, if the mDNS responder has been unable to find any unused name, it should log an error message to inform the user or the operator.

In the event of a name conflict, the new host should configure a new host name; the existing one will not take any action.

Generally speaking, if an mDNS responder (e.g. Host A) observes some other mDNS responder (e.g. Host B) sending an mDNS response message containing a resource record which is intended to be a member of a unique resource record set owned solely by that responder and contains the same name, rrtype, and rrclass as one of Host A's resource records, but different rdata, then this is a conflict and the recipient of the conflicting mDNS response MUST start the Probing process again; this process will determine the winner or the loser. The loser MUST cease using this record immediately and reconfigure.

On the other hand, in the aforementioned case if the received mDNS response contains a resource record with the same name, rrtype, and rrclass and identical rdata, then if the TTL of Host B's resource record given in the message is less than half the true TTL from Hosts A's, then host A MUST mark its record to be announced via multicast, since queriers receiving the record from Host B may delete the record soon. Doing so, host A ensures that the record will be retained for the desired time.

Updating mDNS Records and the Cache-Flush Bit

A host may update the contents of any of its records at any time.

If the rdata of any of a hosts mDNS records changes, the host MUST repeat the Announcing step to update neighbor cache. If necessary (e.g. after a reboot), Probing process has to be used before that. Then, the host MUST send a series of unsolicited announcements to update cache entries in its neighbor hosts. If the record is the one that it has been verified unique, the host sets the most significant bit of the rrclass field of the Resource Record (i.e. the cache-flush bit). Recipients MUST wait 1 sec before flushing from their cache the expired mDNS records. The cache-flush bit MUST NOT be set in any Resource Records in the Known-Answer list of any query message.

In the case of shared records, a host MUST send a “*goodbye*” announcement with RR TTL zero for the old rdata (i.e. an unsolicited mDNS response packet giving the same resource record name, rrtype, rrclass, and rdata, but an RR TTL of zero). The cache-flush bit MUST NOT ever be set in any shared resource record, since this would cause all the other shared versions of this resource record with different rdata from different responders to be immediately deleted from all the caches on the network.

In the case of unique records, a host SHOULD omit the “goodbye” announcement, since the cache-flush bit on the newly announced records will cause old rdata to be flushed from peer caches anyway.

The cache-flush bit does not apply to questions listed in the Question Section of a Multicast DNS message. Finally, the cache-flush bit does not apply to records received within the last second.

mDNS Packets Suppression

RFC 6762 foresees the suppression of mDNS packets under various cases. Specifically, as described mDNS packets MUST be suppressed in the following cases:

- An mDNS responder receives an mDNS query which in its “Known Answer” section includes the answer with an RR TTL at least half of the correct value the answer that this responder would send.
- An mDNS responder receives a response from another host that contains the same answer record and the TTL of that record is not less than the one this responder would send.
- An mDNS querier receives a query from another host that contains the “QM” question it wants to send and the “Known-Answer” section of that record does not contain any records that this host would not also put on it own “Known-Answer” section (i.e. when the already query queries for the same resource records).
- When an mDNS query is sent with the TC (Truncated) bit set (this happens when “Known-Answers” do not fit in one mDNS packet), potential responders should wait for about half a second so as to give the margin for further related queries.

Source Address Check

As explained, all mDNS responses (including responses sent via unicast) SHOULD be sent with IP TTL set to 255; however, non-conforming packets do not have to be discarded. To ensure that off-link mDNS messages are not received and accepted RFC 6762 defines that a host sending mDNS queries to a link-local destination address (including the 224.0.0.251 and FF02::FB link-local multicast addresses) MUST only accept responses to that query that originate from the local link, and silently discard any other response packets.

The test whether a response has originated on the local link is performed as following:

- All responses received with a destination address in the IP header that is the mDNS IPv4 link-local multicast address 224.0.0.251 or the mDNS IPv6 link-local multicast address FF02::FB are necessarily deemed to have originated on the local link, regardless of source IP address.
- For responses received with a unicast destination address in the IP header, the source IP address in the packet is checked to see if it is an address on a local subnet.

References

[RFC 4034], R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, “Resource Records for the DNS Security Extensions”, IETF RFC 4034, March 2005.

[RFC 6762], S. Cheshire, M. Krochmal, “Multicast DNS”, IETF RFC 6762, February 2013.

[RFC 6763], S. Cheshire, M. Krochmal, “DNS-Based Service Discovery”, IETF RFC 6763, February 2013.