Stream: Internet Engineering Task Force (IETF)
RFC: 9664
Category: Standards Track
Published: October 2024
ISSN: 2070-1721
Authors: S. Cheshire *Apple Inc.*
T. Lemon *Apple Inc.*

RFC 9664
An EDNS(0) Option to Negotiate Leases on DNS Updates

Abstract

This document describes an Extension Mechanisms for DNS (EDNS(0)) option that can be used by DNS Update requestors and DNS servers to include a lease lifetime in a DNS Update or response, allowing a server to garbage collect stale resource records that have been added by DNS Updates.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9664.

Copyright Notice

Table of Contents

1. Introduction

A Dynamic DNS Update [RFC2136] allows for a mapping from a persistent hostname to a dynamic IP address. This capability is particularly beneficial to mobile hosts, whose IP address may frequently change with location. However, the mobile nature of such hosts often means that

dynamically updated resource records are not properly deleted. For instance, consider a mobile user who publishes address records via dynamic update. If this user moves their laptop out of range of the Wi-Fi access point, the address record containing stale information may remain on the server indefinitely. Thus, an extension to Dynamic Update is required to tell the server to automatically delete resource records if they are not refreshed after a period of time.

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

2.1. Abbreviations

DNS-SD: DNS-based Service Discovery [RFC6763]

EDNS(0): Extension Mechanisms for DNS [RFC6891]

3. Mechanisms

The EDNS(0) Update Lease option is included in a standard DNS Update message [RFC2136] within an EDNS(0) OPT pseudo-RR [RFC6891].

4. Update Message Format

Dynamic DNS Update Leases Requests and Responses are formatted as standard DNS Dynamic Update messages [RFC2136]. This update **MUST** include the EDNS(0) OPT RR, as described in [RFC6891]. This OPT RR **MUST** include an EDNS(0) Option as shown below.

The Update Lease EDNS(0) option is formatted as follows:

Field Name	Field Type	Description
OPTION-CODE	u_int16_t	UPDATE-LEASE (2)
OPTION-LENGTH	u_int16_t	4 or 8
LEASE	u_int32_t	desired lease (request) or granted lease (response), in seconds

Field Name	Field Type	Description
KEY-LEASE	u_int32_t	optional desired (or granted) lease for KEY records, in seconds

Table 1

Update Requests contain, in the LEASE field of the OPT RDATA, an unsigned 32-bit integer indicating the lease lifetime, in seconds, desired by the requestor, represented in network (big-endian) byte order. In Update Responses, this field contains the actual lease granted by the server. The lease granted by the server may be less than, greater than, or equal to the value requested by the requestor.

There are two variants of the EDNS(0) UPDATE-LEASE option: the basic (4-byte) variant and the extended (8-byte) variant.

In the basic (4-byte) variant, the LEASE indicated in the Update Lease option applies to all resource records in the Update section.

In the extended (8-byte) variant, the Update Lease communicates two lease lifetimes. The LEASE indicated in the Update Lease option applies to all resource records in the Update section *except* for KEY records. The KEY-LEASE indicated in the Update Lease option applies to KEY records in the Update section.

The KEY record can be given a special lease time because this record is used in the DNS-SD Service Registration Protocol [RFC9665] to reserve a name (or names) when the service is not present.

In the case of a KEY record and some other record, obviously the KEY LEASE applies to the key, and the LEASE applies to the other record. If more than one record that is not a KEY record is added by the update, the LEASE (not the KEY LEASE) is applied to all such records. Records that are removed are permanently removed.

4.1. Types of DNS Update Request Messages

This document describes two types of updates: Registrations and Refreshes. A Registration is a DNS Update Request that is intended to add information not already present on the DNS server. A Refresh is intended simply to renew the lease on a previous Registration without changing anything. Both messages are DNS Update messages, so the term "DNS Update message" is to specify behavior that is the same for both types of DNS Update messages.

In some cases, it may be necessary to add new information without removing old information. For the purpose of this document, such messages are Registrations, although in effect, they may also refresh whatever information is unchanged from a previous registration.

4.2. Requestor Behavior

DNS Update requestors **MUST** send an Update Lease option with any DNS Update that is not intended to be present indefinitely. The Update Lease option **SHOULD** specify a time interval that is no shorter than 1800 seconds (30 minutes). Requestors **MAY** specify a shorter lease if they anticipate that the records being updated will change in less than 30 minutes. Requestors that expect the updated records to be relatively static **SHOULD** request appropriately longer leases.

If the DNS response received by the requestor does not include an Update Lease option, this is an indication that the DNS server does not support the Update Lease option. In this case, the requestor **SHOULD** continue sending Refresh messages (see below) as if the server had returned an identical update lease option in its response.

If the DNS response does include an Update Lease option, the requestor **MUST** use the interval or intervals returned in this option when determining when to send Refresh messages. This is true both if the interval or intervals returned by the server are shorter and if they are longer.

When sending a Registration, the requestor **MUST** delay the initial transmission by a random amount of time across the range of 0-3000 milliseconds, with a granularity of no more than 10 milliseconds. This prevents synchronization of multiple devices of the same type at a site upon recovery from a power failure. This requirement applies only to the initial Registration on startup; since Refreshes include a random factor as well, any synchronization that occurs after such an event should quickly randomize.

Note: the 10 ms granularity is a scheduling requirement intended to result in an even spread of requests so that every request doesn't come an exact number of seconds after startup. This requirement should not be construed as requiring anything of the link layer on which the packet is transmitted: the link layer may well impose its own constraints on the timing at which a message is sent, and this document does not claim to override such constraints.

Note: the use of a 3000 ms (3-second) random interval as opposed to some other random interval is to allow for enough time to meaningfully spread the load when many devices renew at once, without delaying so long that the delay in discovery of devices becomes obvious to an end user. A 3-second random delay means that if there are, for example, 100 devices, and the random number generator spread is even, we would have one renewal every 30 ms. In practice, on relatively constrained devices acting as Service Registration Protocol (SRP) servers, we are seeing the processing time for an SRP registration taking on the order of 7 ms, so this seems reasonable.

4.3. Server Behavior

DNS servers implementing the Update Lease option **MUST** include an Update Lease option in response to any successful DNS Update (RCODE=0) that includes an Update Lease option. Servers **MAY** return a different lease interval or intervals than specified by the requestor, granting relatively longer or shorter leases to reduce network traffic due to Refreshes or to reduce stale data, respectively.

Note that both the 4-byte and 8-byte variant are valid on both clients and servers, but clients and servers may exist that do not support the newer 8-byte variant. Therefore, clients and servers that do support this variant must account for the possibility that the server with which they are communicating does not.

A client that receives a 4-byte variant from a server when it sent an 8-byte variant **MUST** treat the 4-byte variant as specifying both the lease time and the key lease time. A server that supports the 8-byte variant **MUST** treat the 4-byte variant as specifying both the lease time and the key lease time. When a server receives a 4-byte variant, it **MUST** respond with a 4-byte variant. In this case, the key and the other records expire at the same time.

5. Refresh Messages

A Refresh message is a DNS Update message that is sent to the server after an initial DNS Update has been sent in order to prevent the update's records from being garbage collected.

5.1. Refresh Message Format

Refresh messages are formatted like Dynamic Update Leases Requests and Responses (see Section 4). The Refresh message is constructed with the assumption that the result of the previous Registration or Refresh is still in effect. In the case that the records added in a previous update were for some reason garbage collected, the Refresh message will result in those records being added again.

The Refresh message **SHOULD NOT** include any update prerequisites that would fail if the requestor's previous Registration or Refresh is still in effect. It also **SHOULD NOT** include prerequisites that would fail if the records affected by the previous Registration or Refresh are no longer present; that is, the Refresh should also work as a Registration. There may be cases where this is not possible; in which case, the response from the server can be used to determine how to proceed when the Refresh fails.

An update message that changes the server state resulting from a previous Refresh or Registration is a Registration, not a Refresh.

The Update Lease option in a Refresh message contains the desired new lease for Requests, and the actual granted lease for Responses. The LEASE interval indicated in the Update Lease option applies to all resource records in the Update section of the Refresh request, except that if a KEY-LEASE interval is included as well, that interval applies to any KEY records included in the Update section.

5.2. Requestor Behavior

A requestor that intends for its records from a previous Registration or Refresh to remain active **MUST** send a Refresh message before the lease elapses; otherwise, the records will be removed by the server.

In order to prevent records expiring, requestors **MUST** refresh resource records before they expire. At the time of registration, the client computes an interval that is 80% of the lease time plus a random offset between 0% and 5% of the lease time. The random offset is to prevent refreshes from being synchronized. When this interval has expired, the client **MUST** refresh the message if the data in the initial Registration should continue to be advertised.

For Refresh messages, the server is expected to return an Update Lease option, if supported, just as with the initial Registration. As with the Registration, the requestor **MUST** use the intervals specified by the server when determining when to send the next Refresh message.

When sending Refresh messages, the requestor **MUST** include an Update Lease option, as it did for the initial Registration. The Update Lease option **MAY** either specify the same intervals as in the initial Registration or use the values returned by the server in the previous Update Response, whether it was a response to a Registration or a Refresh. As with responses to Registrations, the requestor **MUST** use the interval or intervals returned by the server in the response when determining when to send the next Refresh message.

5.2.1. Coalescing Refresh Messages

If the requestor has performed multiple successful Registrations with a single server, the requestor **MAY** include Refreshes for all such Registrations to that server in a single message. This effectively places all records for a requestor on the same expiration schedule, reducing network traffic due to Refreshes.

In doing so, the requestor includes in the Refresh message all existing updates to the server, including those not yet close to expiration, so long as at least one resource record in the message has elapsed at least 75% of its original lease. If the requestor uses UDP, the requestor **MUST NOT** coalesce Refresh messages if doing so would cause truncation of the message; in this case, the requestor should either send multiple messages or use TCP to send the entire update at once.

Requestors **SHOULD NOT** send Refresh messages when all of the records in the Refresh have more than 50% of their lease interval remaining before expiry. However, there may be cases where the requestor needs to send an early Refresh, and it **MAY** do so. For example, a power-constrained (sleepy) device may need to send an update when the radio is powered so as to avoid having to power it up later.

Another case where this may be needed is if the lease interval registered with the server is no longer appropriate and the Requestor wishes to negotiate a different lease interval. However, in this case, if the server does not honor the requested interval in its response, the requestor **MUST NOT** retry this negotiation.

5.3. Server Behavior

Upon receiving a valid Refresh Request, the server **MUST** send an acknowledgment. This acknowledgment is identical to the Update Response format described in Section 4 and contains the new lease of the resource records being Refreshed. The server **MUST NOT** increment the serial number of a zone as the result of a Refresh.

However, the server's state may not match what the client expects. In this case, a Refresh may actually appear to be a Registration, from the server's perspective. If the Refresh changes the contents of the zone, the server **MUST** update the zone serial number.

6. Retransmission Strategy

The DNS protocol, including DNS updates, can operate over UDP or TCP. When using UDP, reliable transmission must be guaranteed by retransmitting if a DNS UDP message is not acknowledged in a reasonable interval. Section 4.2.1 of [RFC1035] provides some guidance on this topic, as does Section 1 of [RFC1536]. Section 3.1.3 of [RFC8085] also provides useful guidance that is particularly relevant to DNS.

7. Garbage Collection

If the Update Lease of a resource record elapses without being refreshed, the server **MUST NOT** return the expired record in answers to queries. The server **MAY** delete the record from its database. The lease interval or intervals returned by the server to the requestor are used in determining when the lease on a resource record has expired.

For all resource records other than a KEY record included in a DNS Update request, the Update Lease is the LEASE value in the Update Lease option. For KEY records, if the optional KEY-LEASE value was included, this interval is used rather than the interval specified in the LEASE. If the KEY-LEASE was not specified, the interval specified in the LEASE is used.

8. Security Considerations

Section 8 of [RFC2136] describes problems that can occur around DNS updates. Servers implementing this specification should follow these recommendations.

Several additional issues can arise when relying on the Update Lease option. First, a too-long lease time is not much different than no lease time: the records associated with this lease time will effectively never be cleaned up. Servers implementing the Update Lease should have a default upper bound on the maximum acceptable value both for the LEASE and KEY-LEASE values sent by the client. Servers **MAY** provide a way for the operator to change this upper limit. Default values for these limits of 24 hours and 7 days, respectively, are **RECOMMENDED**.

The second issue is that a too-short lease can result in increased server load as requestors rapidly renew the lease. A delay in renewing could result in the data being removed prematurely. Servers implementing Update Lease **MUST** have a default minimum lease interval that avoids this issue. We RECOMMEND a minimum of 30 seconds for both the LEASE and KEY-LEASE intervals. However, in most cases, much longer lease times (for example, an hour) are **RECOMMENDED**.

There may be some cost associated with renewing leases. A malicious (or buggy) client could renew at a high rate in order to overload the server more than it would be overloaded by query traffic. This risk is present for a regular DNS update as well. The server **MUST** enforce a minimum

interval between updates. After a Refresh or Registration has been successfully processed and acknowledged, another Update of either type from the client during that interval **MUST** be silently ignored by the server.

Some authentication strategy should be used when accepting DNS updates. A shared secret [RFC8945] or public key signing (e.g., SIG(0) [RFC2931]) should be required. Keys should have limited authority: compromise of a key should not result in compromise of the entire contents of one or more zones managed by the server. Key management strategy is out of scope for this document. An example of a key management strategy can be found in [RFC9665], which uses "First Come, First Served Naming" rather than an explicit trust establishment process to confer update permission to a set of records.

9. IANA Considerations

IANA has updated the "DNS EDNS0 Option Codes (OPT)" registry [EDNS0Codes] as regards value 2 as follows:

Value: 2
Name: Update Lease
Status: Standard
Reference: RFC 9664

10. References

10.1. Normative References

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <https://www.rfc-editor.org/info/rfc2136>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

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

10.2. Informative References

[RFC1536] Kumar, A., Postel, J., Neuman, C., Danzig, P., and S. Miller, "Common DNS Implementation Errors and Suggested Fixes", RFC 1536, DOI 10.17487/RFC1536, October 1993, <https://www.rfc-editor.org/info/rfc1536>.

[RFC2931] Eastlake 3rd, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, DOI 10.17487/RFC2931, September 2000, <https://www.rfc-editor.org/info/rfc2931>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/info/rfc6763>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC8945] Dupont, F., Morris, S., Vixie, P., Eastlake 3rd, D., Gudmundsson, O., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", STD 93, RFC 8945, DOI 10.17487/RFC8945, November 2020, <https://www.rfc-editor.org/info/rfc8945>.

[RFC9665] Lemon, T. and S. Cheshire, "Service Registration Protocol for DNS-Based Service Discovery", RFC 9665, DOI 10.17487/RFC9665, October 2024, <https://www.rfc-editor.org/info/rfc9665>.

[EDNS0Codes] IANA, "DNS EDNS0 Option Codes (OPT)", <https://www.iana.org/assignments/dns-parameters>.

Acknowledgments

Thanks to Marc Krochmal and Kiren Sekar for their work in 2006 on the precursor to this document. Thanks also to Roger Pantos and Chris Sharp for their contributions. Thanks to Chris Box, Esko Dijk, Jonathan Hui, Peter van Dijk, Abtin Keshvarzian, Nathan Dyck, Steve Hanna, Gabriel Montenegro, Kangping Dong, and Tim Wicinski for their working group reviews of this document. Thanks to David Dong, Olafur Gudmundsson, Brian Trammel, and Shivan Sahib for their directorate reviews and IANA reviews.

Authors' Addresses

Stuart Cheshire
Apple Inc.
One Apple Park Way
Cupertino, CA 95014
United States of America
Phone: +1 408 974 3207
Email: cheshire@apple.com

Ted Lemon
Apple Inc.
P.O. Box 958
Brattleboro, VT 05302
United States of America
Email: mellon@fugue.com