A   M. S. Lenders   C. Amsüss   C. Gündoğan
         uFU Berlin   HAW Hamburg
         t
         h
         o
         r
         s
         :
          T. C. Schmidt   M. Wählisch
          HAW Hamburg   FU Berlin

DNS over CoAP (DoC)

Abstract

This document defines a protocol for sending DNS messages over the
Constrained Application Protocol (CoAP). These CoAP messages are
protected by DTLS-Secured CoAP (CoAPS) or Object Security for
Constrained RESTful Environments (OSCORE) to provide encrypted DNS
message exchange for constrained devices in the Internet of Things
(IoT).

Discussion Venues

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

Discussion of this document takes place on the Constrained RESTful
Environments Working Group mailing list (core@ietf.org), which is
archived at https://mailarchive.ietf.org/arch/browse/core/.

Source for this draft and an issue tracker can be found at https://
github.com/core-wg/draft-dns-over-coap.

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 https://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 9 March 2023.

Copyright Notice

Table of Contents

1. Introduction

This document defines DNS over CoAP (DoC), a protocol to send DNS [RFC1035] queries and get DNS responses over the Constrained Application Protocol (CoAP) [RFC7252]. Each DNS query-response pair is mapped into a CoAP message exchange. Each CoAP message is secured by DTLS [RFC6347] or Object Security for Constrained RESTful Environments (OSCORE) [RFC8613] to ensure message integrity and confidentiality.

The application use case of DoC is inspired by DNS over HTTPS [RFC8484] (DoH). DoC, however, aims for the deployment in the

constrained Internet of Things (IoT), which usually conflicts with
the requirements introduced by HTTPS.

To prevent TCP and HTTPS resource requirements, constrained IoT
devices could use DNS over DTLS [RFC8094]. In contrast to DNS over
DTLS, DoC utilizes CoAP features to mitigate drawbacks of datagram-
based communication. These features include: block-wise transfer,
which solves the Path MTU problem of DNS over DTLS (see [RFC8094],
section 5); CoAP proxies, which provide an additional level of
caching; re-use of data structures for application traffic and DNS
information, which saves memory on constrained devices.

To prevent resource requirements of DTLS or TLS on top of UDP (e.g.,
introduced by DNS over QUIC [RFC9250]), DoC allows for lightweight
end-to-end payload encryption based on OSCORE.

```
                - FETCH coaps://[2001:db8::1]/
              /
            /
          CoAP request
+--------+  [DNS query]   +--------+   DNS query    +--------+
|  DoC   |--------------->|  DoC   |...............>|  DNS   |
| Client |<---------------| Server |<...............| Server |
+--------+  CoAP response +--------+  DNS response  +--------+
          [DNS response]
```
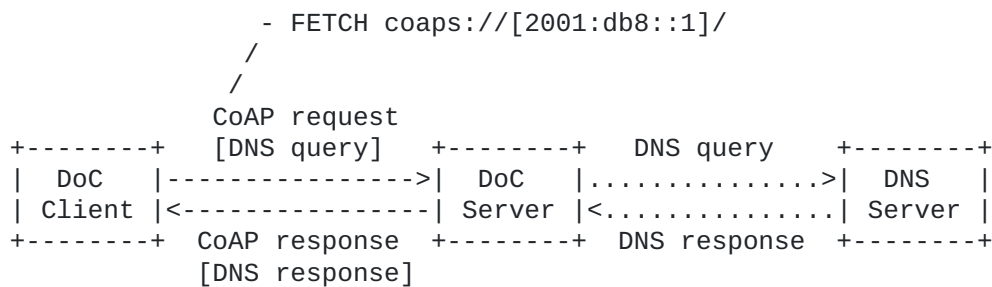
Figure 1: Basic DoC architecture

The most important components of DoC can be seen in Figure 1: A DoC
client tries to resolve DNS information by sending DNS queries
carried within CoAP requests to a DoC server. That DoC server may or
may not resolve that DNS information itself by using other DNS
transports with an upstream DNS server. The DoC server then replies
to the DNS queries with DNS responses carried within CoAP responses.

2. Terminology

A server that provides the service specified in this document is
called a "DoC server" to differentiate it from a classic "DNS
server". Correspondingly, a client using this protocol to retrieve
the DNS information is called a "DoC client".

The term "constrained nodes" is used as defined in [RFC7228].

The terms "CoAP payload" and "CoAP body" are used as defined in
[RFC7959].

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.

3. Selection of a DoC Server

In this document, it is assumed that the DoC client knows the DoC server and the DNS resource at the DoC server. Possible options could be manual configuration of a URI [RFC3986] or CRI [I-D.ietf-core-href], or automatic configuration, e.g., using a CoRE resource directory [RFC9176], DHCP or Router Advertisement options [I-D.ietf-add-dnr]. Automatic configuration **SHOULD** only be done from a trusted source.

When discovering the DNS resource through a link mechanism that allows describing a resource type (e.g., the Resource Type Attribute in [RFC6690]), the resource type "core.dns" can be used to identify a generic DNS resolver that is available to the client.

4. Basic Message Exchange

4.1. The "application/dns-message" Content-Format

This document defines the Internet media type "application/dns-message" for the CoAP Content-Format. This media type is defined as in [RFC8484] Section 6, i.e., a single DNS message encoded in the DNS on-the-wire format [RFC1035]. Both DoC client and DoC server **MUST** be able to parse contents in the "application/dns-message" format.

4.2. DNS Queries in CoAP Requests

A DoC client encodes a single DNS query in one or more CoAP request messages the CoAP FETCH [RFC8132] method. Requests **SHOULD** include an Accept option to indicate the type of content that can be parsed in the response.

The CoAP request **SHOULD** be carried in a Confirmable (CON) message, if the transport used does not provide reliable message exchange.

4.2.1. Request Format

When sending a CoAP request, a DoC client **MUST** include the DNS query in the body of the CoAP request. As specified in [RFC8132] Section 2.3.1, the type of content of the body **MUST** be indicated using the Content-Format option. This document specifies the usage of Content-Format "application/dns-message" (details see Section 4.1). A DoC server **MUST** be able to parse requests of Content-Format "application/dns-message".

4.2.2. Support of CoAP Caching

The DoC client **SHOULD** set the ID field of the DNS header always to 0 to enable a CoAP cache (e.g., a CoAP proxy en-route) to respond to the same DNS queries with a cache entry. This ensures that the CoAP Cache-Key (see [RFC8132] Section 2) does not change when multiple DNS queries for the same DNS data, carried in CoAP requests, are issued.

4.2.3. Examples

The following example illustrates the usage of a CoAP message to resolve "example.org. IN AAAA" based on the URI "coaps://[2001:db8::1]/". The CoAP body is encoded in "application/dns-message" Content Format.

```
FETCH coaps://[2001:db8::1]/
Content-Format: application/dns-message
Accept: application/dns-message
Payload: 00 00 01 20 00 02 00 00 00 00 00 00 07 65 78 61 [binary]
         6d 70 6c 65 03 6f 72 67 00 00 1c 00 01 c0 0c 00 [binary]
         01 00 01                                        [binary]
```

4.3. DNS Responses in CoAP Responses

Each DNS query-response pair is mapped to a CoAP REST request-response operation. DNS responses are provided in the body of the CoAP response. A DoC server **MUST** be able to produce responses in the "application/dns-message" Content-Format (details see [Section 4.1](#)) when requested. A DoC client **MUST** understand responses in "application/dns-message" format when it does not send an Accept option. Any other response format than "application/dns-message" **MUST** be indicated with the Content-Format option by the DoC server.

4.3.1. Response Codes and Handling DNS and CoAP errors

A DNS response indicates either success or failure in the Response code of the DNS header (see [RFC1035] Section 4.1.1). It is **RECOMMENDED** that CoAP responses that carry any valid DNS response use a "2.05 Content" response code.

CoAP responses use non-successful response codes **MUST NOT** contain a DNS response and **MUST** only be used on errors in the CoAP layer or when a request does not fulfill the requirements of the DoC protocol.

Communication errors with a DNS server (e.g., timeouts) **SHOULD** be indicated by including a SERVFAIL DNS response in a successful CoAP response.

A DoC client might try to repeat a non-successful exchange unless otherwise prohibited. The DoC client might also decide to repeat a non-successful exchange with a different URI, for instance, when the response indicates an unsupported Content-Format.

4.3.2. Support of CoAP Caching

The DoC server **MUST** ensure that any sum of the Max-Age value of a CoAP response and any TTL in the DNS response is less or equal to the corresponding TTL received from an upstream DNS server. This also includes the default Max-Age value of 60 seconds (see [RFC7252], section 5.10.5) when no Max-Age option is provided. The DoC client **MUST** then add the Max-Age value of the carrying CoAP response to all TTLs in a DNS response on reception and use these calculated TTLs for the associated records.

The **RECOMMENDED** algorithm to assure the requirement for the DoC is to set the Max-Age option of a response to the minimum TTL of a DNS response and to subtract this value from all TTLs of that DNS response. This prevents expired records unintentionally being served from an intermediate CoAP cache. Additionally, it allows for the ETag value for cache validation, if it is based on the content of the response, not to change even if the TTL values are updated by an upstream DNS cache. If only one record set per DNS response is assumed, a simplification of this algorithm is to just set all TTLs in the response to 0 and set the TTLs at the DoC client to the value of the Max-Age option.

4.3.3. Examples

The following examples illustrate the replies to the query "example.org. IN AAAA record", recursion turned on. Successful responses carry one answer record including address 2001:db8:1::1:2:3:4 and TTL 58719.

A successful response:

```
2.05 Content
Content-Format: application/dns-message
Max-Age: 58719
Payload: 00 00 81 a0 00 01 00 01 00 00 00 00 07 65 78 61 [binary]
         6d 70 6c 65 03 6f 72 67 00 00 1c 00 01 c0 0c 00 [binary]
         1c 00 01 00 01 37 49 00 10 20 01 0d b8 00 01 00 [binary]
         00 00 01 00 02 00 03 00 04                      [binary]
```

When a DNS error (SERVFAIL in this case) is noted in the DNS response, the CoAP response still indicates success:

```
2.05 Content
Content-Format: application/dns-message
Payload: 00 00 81 a2 00 01 00 00 00 00 00 00 07 65 78 61 [binary]
         6d 70 6c 65 03 6f 72 67 00 00 1c 00 01         [binary]
```

When an error occurs on the CoAP layer, the DoC server **SHOULD** respond with an appropriate CoAP error, for instance "4.15 Unsupported Content-Format" if the Content-Format option in the request was not set to "application/dns-message" and the Content-Format is not otherwise supported by the server.

5. CoAP/CoRE Integration

5.1. DoC Server Considerations

In the case of CNAME records in a DNS response, a DoC server **SHOULD** follow common DNS resolver behavior [RFC1034] by resolving a CNAME until the originally requested resource record type is reached. This reduces the number of message exchanges within an LLN.

The DoC server **SHOULD** send compact answers, i.e., additional or authority sections in the DNS response should only be sent if needed or if it is anticipated that they help the DoC client to reduce additional queries.

5.2. Observing the DNS Resource

There are use cases where updating a DNS record might be necessary on the fly. Examples of this include e.g. [RFC8490], Section 4.1.2, but just saving messages by omitting the query for a subscribed name might also be valid. As such, the DNS resource **MAY** be observable as specified in [RFC7641].

5.3. OSCORE

It is **RECOMMENDED** to carry DNS messages end-to-end encrypted using OSCORE [RFC8611]. The exchange of the security context is out of scope of this document.

6. Considerations for Unencrypted Use

While not recommended, DoC can be used without any encryption (e.g., in very constrained environments where encryption is not possible or necessary). It can also be used when lower layers provide secure communication between client and server. In both cases, potential benefits of unencrypted DoC usage over classic DNS are e.g. block-wise transfer or alternative CoAP Content-Formats to overcome link-layer constraints.

7. Security Considerations

TODO Security

8. IANA Considerations

8.1. New "application/dns-message" Content-Format

IANA is requested to assign CoAP Content-Format ID for the DNS message media type in the "CoAP Content-Formats" sub-registry, within the "CoRE Parameters" registry [RFC7252], corresponding the "application/dns-message" media type from the "Media Types" registry:

Media-Type: application/dns-message

Encoding: -

Id: TBD

Reference: [TBD-this-spec]

8.2. New "core.dns" Resource Type

IANA is requested to assign a new Resource Type (rt=) Link Target Attribute, "core.dns" in the "Resource Type (rt=) Link Target Attribute Values" sub-registry, within the "CoRE Parameters" register [RFC6690].

Attribute Value: core.dns

Description: DNS over CoAP resource.

Reference: [TBD-this-spec] Section 3

9. References

9.1. 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/rfc/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/rfc/
            rfc2119>.

[RFC6347]   Rescorla, E. and N. Modadugu, "Datagram Transport Layer
            Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347,
            January 2012, <https://www.rfc-editor.org/rfc/rfc6347>.

[RFC7252]   Shelby, Z., Hartke, K., and C. Bormann, "The Constrained
            Application Protocol (CoAP)", RFC 7252, DOI 10.17487/
            RFC7252, June 2014, <https://www.rfc-editor.org/rfc/
            rfc7252>.

[RFC7641]   Hartke, K., "Observing Resources in the Constrained
            Application Protocol (CoAP)", RFC 7641, DOI 10.17487/
            RFC7641, September 2015, <https://www.rfc-editor.org/rfc/
            rfc7641>.

[RFC7959]   Bormann, C. and Z. Shelby, Ed., "Block-Wise Transfers in
            the Constrained Application Protocol (CoAP)", RFC 7959,
            DOI 10.17487/RFC7959, August 2016, <https://www.rfc-
            editor.org/rfc/rfc7959>.

[RFC8132]   van der Stok, P., Bormann, C., and A. Sehgal, "PATCH and
            FETCH Methods for the Constrained Application Protocol
            (CoAP)", RFC 8132, DOI 10.17487/RFC8132, April 2017,
            <https://www.rfc-editor.org/rfc/rfc8132>.

[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/rfc/rfc8174>.

[RFC8613]   Selander, G., Mattsson, J., Palombini, F., and L. Seitz,
            "Object Security for Constrained RESTful Environments
            (OSCORE)", RFC 8613, DOI 10.17487/RFC8613, July 2019,
            <https://www.rfc-editor.org/rfc/rfc8613>.

9.2. Informative References

[I-D.ietf-add-dnr] Boucadair, M., Reddy, T., Wing, D., Cook, N.,
            and T. Jensen, "DHCP and Router Advertisement Options for
            the Discovery of Network-designated Resolvers (DNR)",
            Work in Progress, Internet-Draft, draft-ietf-add-dnr-13,
            13 August 2022, <https://datatracker.ietf.org/doc/html/
            draft-ietf-add-dnr-13>.

[I-D.ietf-core-href] Bormann, C. and H. Birkholz, "Constrained
            Resource Identifiers", Work in Progress, Internet-Draft,

draft-ietf-core-href-10, 7 March 2022, <https://
                  datatracker.ietf.org/doc/html/draft-ietf-core-href-10>.

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

   [RFC3986]   Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
                  Resource Identifier (URI): Generic Syntax", STD 66, RFC
                  3986, DOI 10.17487/RFC3986, January 2005, <https://
                  www.rfc-editor.org/rfc/rfc3986>.

   [RFC6690]   Shelby, Z., "Constrained RESTful Environments (CoRE) Link
                  Format", RFC 6690, DOI 10.17487/RFC6690, August 2012,
                  <https://www.rfc-editor.org/rfc/rfc6690>.

   [RFC7228]   Bormann, C., Ersue, M., and A. Keranen, "Terminology for
                  Constrained-Node Networks", RFC 7228, DOI 10.17487/
                  RFC7228, May 2014, <https://www.rfc-editor.org/rfc/
                  rfc7228>.

   [RFC8094]   Reddy, T., Wing, D., and P. Patil, "DNS over Datagram
                  Transport Layer Security (DTLS)", RFC 8094, DOI 10.17487/
                  RFC8094, February 2017, <https://www.rfc-editor.org/rfc/
                  rfc8094>.

   [RFC8484]   Hoffman, P. and P. McManus, "DNS Queries over HTTPS
                  (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018,
                  <https://www.rfc-editor.org/rfc/rfc8484>.

   [RFC8490]   Bellis, R., Cheshire, S., Dickinson, J., Dickinson, S.,
                  Lemon, T., and T. Pusateri, "DNS Stateful Operations",
                  RFC 8490, DOI 10.17487/RFC8490, March 2019, <https://
                  www.rfc-editor.org/rfc/rfc8490>.

   [RFC8611]   Akiya, N., Swallow, G., Litkowski, S., Decraene, B.,
                  Drake, J., and M. Chen, "Label Switched Path (LSP) Ping
                  and Traceroute Multipath Support for Link Aggregation
                  Group (LAG) Interfaces", RFC 8611, DOI 10.17487/RFC8611,
                  June 2019, <https://www.rfc-editor.org/rfc/rfc8611>.

   [RFC9176]   Amsüss, C., Ed., Shelby, Z., Koster, M., Bormann, C.,
                  and P. van der Stok, "Constrained RESTful Environments
                  (CoRE) Resource Directory", RFC 9176, DOI 10.17487/
                  RFC9176, April 2022, <https://www.rfc-editor.org/rfc/
                  rfc9176>.

   [RFC9250]   Huitema, C., Dickinson, S., and A. Mankin, "DNS over
                  Dedicated QUIC Connections", RFC 9250, DOI 10.17487/
                  RFC9250, May 2022, <https://www.rfc-editor.org/rfc/
                  rfc9250>.

Appendix A. Change Log

Authors' Addresses

Martine Sophie Lenders
Freie Universität Berlin

Email: m.lenders@fu-berlin.de

Christian Amsüss

Email: christian@amsuess.com

Cenk Gündoğan
HAW Hamburg

Email: cenk.guendogan@haw-hamburg.de

Thomas C. Schmidt
HAW Hamburg

Email: t.schmidt@haw-hamburg.de

Matthias Wählisch
Freie Universität Berlin

Email: m.waehlisch@fu-berlin.de