Workgroup: CoRE
Internet-Draft: draft-lenders-dns-over-coap-02
Published: 25 October 2021
Intended Status: Standards Track
Expires: 28 April 2022
Authors: M.S. Lenders   C. Amsüss   C. Gündoğan
FU Berlin   HAW Hamburg
T.C. Schmidt   M. Wählisch
HAW Hamburg   FU Berlin

DNS Queries 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 TODO

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

Status of This Memo

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 [I-D.ietf-dprive-dnsoquic]), 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.

TBD: additional feature sets of CoAP/CoRE

* resource directory for DoC service discovery,

* ...

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

TBD:

* URI or should we rather go straight to CRI?

* Support for more than one URI by DoC server.

* DoC server identity, key exchange, ...

3.1. URI Alternatives

TBD:

* CRI [I-D.ietf-core-href] or CoRAL [I-D.ietf-core-coral]

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].

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.

To enable reliable message exchange, the CoAP request **SHOULD** be carried in a Confirmable (CON) message.

4.2.1. Request Format

When sending a CoAP request, a DoC client **MUST** include the DNS query in the body (i.e. the payload, or the concatenated payloads) 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).

If block-wise transfer [RFC7959] is supported by the client, more than one CoAP request message **MAY** be used. If more than one CoAP request message is used to encode the DNS query, it must be chained together using the Block1 option in those CoAP requests.

The FETCH request is sent to the URI specified in Section 3.

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, which may consist of several CoAP request-response pairs if block-wise transfer is involved. DNS responses are provided in the body (i.e. the payload, or the concatenated payloads) of the CoAP response. A DoC server **MUST** indicate the type of content of the body using the Content-Format option, and **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.

If supported, a DoC server **MAY** transfer the DNS response in more than one CoAP responses using the Block2 option [RFC7959].

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 any payload and may 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

It is **RECOMMENDED** to set the Max-Age option of a response to the minimum TTL in the Answer section of a DNS response. This prevents expired records unintentionally being served from a CoAP cache.

It is **RECOMMENDED** that DoC servers set an ETag option on large responses (TBD: more concrete guidance) that have a short Max-Age relative to the expected clients' caching time. Thus, clients that need to revalidate a response can do so using the established ETag mechanism. With responses large enough to be fragmented, it's best practice for servers to set an ETag anyway. As specified in [RFC7252] and [RFC8132], if the response associated with the ETag is still valid, the response uses the "2.03 Valid" code and consequently carries no payload.

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. Proxies and caching

TBD:

* TTL vs. Max-Age

* Responses that are not globally valid

*General CoAP proxy problem, but what to do when DoC server is a DNS proxy, response came not yet in but retransmission by DoC client was received (see Figure 2)

  -send empty ACK (maybe move to best practices appendix)

```
DoC client            DoC proxy           DNS server
     |  CoAP req [rt 1]  |                     |
     |------------------>|  DNS query [rt 1]   |
     |                   |-------------------->|
     |  CoAP req [rt 2]  |                     |
     |------------------>|      DNS resp       |
     |     CoAP resp     |<--------------------|
     |<------------------|                     |
     |                   |                     |
```

Figure 2: CoAP retransmission (rt) is received before DNS query could have been fulfilled.

5.2. OBSERVE (modifications)?

*TBD

*DoH has considerations on Server Push to deliver additional, potentially outstanding requests + response to the DoC client for caching

*OBSERVE does not include the request it would have been generated from ==> cannot be cached without corresponding request having been send over the wire.

*If use case exists: extend OBSERVE with option that contains "promised" request (see [RFC7540], section 8.2)?

*Other caveat: clients can't cache, only proxys so value needs to be evaluated

*Potential use case: [RFC8490] Section 4.1.2

5.3. OSCORE

*TBD

*With OSCORE DTLS might not be required

6. URI configuration

*TBD

*Maybe out-of-scope?

* DHCP and RA options to deliver? [I-D.peterson-doh-dhcp]

* CoRE-RD [I-D.ietf-core-resource-directory] (...; can not express URI templates)

* When no actual templating is involved: regular resource discovery ("rt=core.dns"?) through .well-known/core

7. Considerations for Unencrypted Use

* TBD

* DTLS-transport should be used

* Non-DTLS can have benefits: Blockwise-transfer for IEEE 802.15.4, additional layer of caching, ...

8. Security Considerations

TODO Security

9. IANA Considerations

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]

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/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>.

[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>.

10.2. Informative References

[I-D.ietf-core-coral] Amsüss, C. and T. Fossati, "The Constrained RESTful Application Language (CoRAL)", Work in Progress, Internet-Draft, draft-ietf-core-coral-04, 25 October 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-core-coral-04>.

[I-D.ietf-core-href] Bormann, C. and H. Birkholz, "Constrained Resource Identifiers", Work in Progress, Internet-Draft, draft-ietf-core-href-06, 25 July 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-core-href-06>.

[I-D.ietf-core-resource-directory] Amsüss, C., Shelby, Z., Koster, M., Bormann, C., and P. V. D. Stok, "CoRE Resource Directory", Work in Progress, Internet-Draft, draft-ietf-core-resource-directory-28, 7 March 2021, <https://

datatracker.ietf.org/doc/html/draft-ietf-core-resource-directory-28>.

[I-D.ietf-dprive-dnsoquic] Huitema, C., Dickinson, S., and A. Mankin, "DNS over Dedicated QUIC Connections", Work in Progress, Internet-Draft, draft-ietf-dprive-dnsoquic-06, 20 October 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-dprive-dnsoquic-06>.

[I-D.peterson-doh-dhcp] Peterson, T., "DNS over HTTP resolver announcement Using DHCP or Router Advertisements", Work in Progress, Internet-Draft, draft-peterson-doh-dhcp-01, 21 October 2019, <https://datatracker.ietf.org/doc/html/draft-peterson-doh-dhcp-01>.

[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>.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <https://www.rfc-editor.org/rfc/rfc7540>.

[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>.

Appendix A. Change Log

TBD:

* Request text duplication

A.1. Since draft-lenders-dns-over-coap-01

* Remove GET and POST methods

* Add note on ETag and response codes

* Provide requirement conflict for DNS over QUIC

* Clarify Content-Format / Accept handling

A.2. Since draft-lenders-dns-over-coap-00

* Soften Content-Format requirements in Section 4.2.1 and Section 4.3

* Clarify "CoAP payload"/"CoAP body" terminology

* Fix nits and typos

A.3. Since draft-lenders-dns-over-coaps-00

* Clarification in abstract that both DTLS and OSCORE can be used as secure transport

* Restructuring of Section 4:

  - Add dedicated Section 4.1 on Content-Format

  - Add overview table about usable and required features for request method types to Section 4.2

  - Add dedicated Section 4.2.2 and Section 4.3.2 on caching requirements for CoAP requests and responses

* Fix nits and typos

Acknowledgments

TODO acknowledge.

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