Workgroup: CoRE
Internet-Draft:
draft-ietf-core-dns-over-coap-06
Published: 4 March 2024
Intended Status: Standards Track
Expires: 5 September 2024
Authors: M. S. Lenders C. Amsüss C. Gündoğan
TU Dresden Huawei Technologies
T. C. Schmidt M. Wählisch
HAW Hamburg TU Dresden & Barkhausen Institut

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 5 September 2024.

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 [RFC9147] 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. Constrained IoT devices may be restricted in memory, power consumption, link layer frame sizes, throughput, and latency. They may only have a handful kilobytes of both RAM and ROM. They may sleep for long durations of time, after which they need to refresh the named resources they know about. Name resolution in such scenarios must take into account link layer frame sizes of only a few hundred bytes, bit rates in the magnitute of kilobits per second, and latencies of several seconds [RFC7228].

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 payload encryption based on OSCORE.

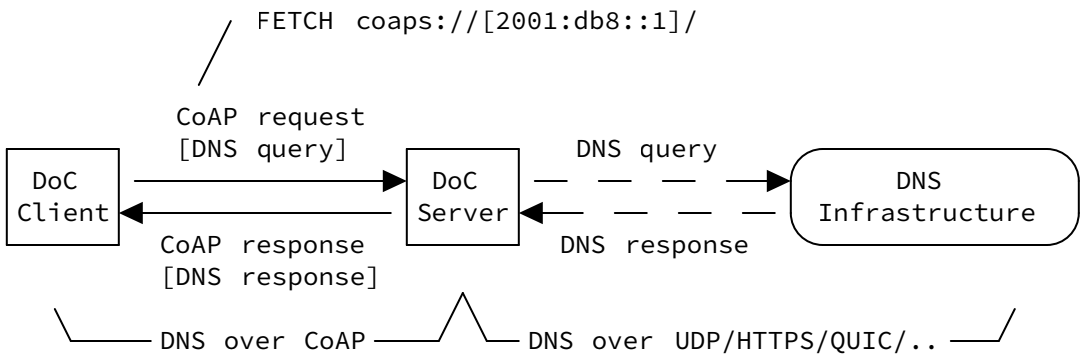

Figure 1: Basic DoC architecture

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 is a DNS client (i.e., a stub or recursive resolver) that resolves DNS information by using other DNS transports such as DNS over UDP [RFC1035], DNS over HTTPS [RFC8484], or DNS over QUIC [RFC9250] when communicating with the upstream DNS infrastructure. Using that information, the DoC server then replies to the queries of the DoC client with DNS responses carried within CoAP responses.

Note that this specification is disjunct from DoH since the CoRE-specific FETCH method is used. This was done to take benefit from having the DNS query in the payload as with POST, but still having the caching advantages we would gain with GET. Having the DNS query in the payload means we do not need extra base64 encoding, which would increase code complexity and message sizes. We are also able to transfer a query block-wise.

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". A DoC server acts either as a DNS stub resolver [RFC8499] or a DNS recursive resolver [RFC8499].

A client using the service specified in this document 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], Section 2.

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 [RFC9463] or discovery of designated resolvers [RFC9462]. Automatic configuration **SHOULD** only be done from a trusted source.

Support for SVCB Resource Records [RFC9460], [RFC9461] or DNR Service Parameters [RFC9463] are not specified in this document. [I-D.lenders-core-dnr] explores solutions for CoAP for these mechanisms.

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.

While there is no path specified it is **RECOMMENDED** to use the root path "/" for the DNS resource to keep the CoAP requests small.

4. Basic Message Exchange

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

This document defines a CoAP Content-Format number for the Internet media type "application/dns-message" to be the mnemonic 553--based on the port assignment of DNS. 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 that use the CoAP FETCH [RFC8132] method. Requests **SHOULD** include an Accept option to indicate the type of content that can be parsed in the response.

Since CoAP provides reliability of the message layer (e.g. CON) the retransmission mechanism of the DNS protocol as defined in [RFC1035] is not needed.

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

For reliability and energy saving measures content decoupling and thus en-route caching on proxies takes a far greater role than it does, e.g., in HTTP. Likewise, CoAP utilizes cache validation to refresh stale cache entries without large messages which often uses hashing over the message content for ETag generation. As such, the approach to guarantee the same cache key for DNS responses as proposed in DoH ([RFC8484], section 5.1) is not sufficient and needs to be updated so that the TTLs in the response are more often the same regardless of query time.

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. DNS Push

DNS Push requires additional overhead, which conflicts with constrained resources, This is the reason why it is **RECOMMENDED** to use CoAP Observe [RFC7641] instead of DNS Push in the DoC domain.

If the CoAP request indicates that the DoC client wants to observe a resource record, a DoC server **MAY** use a DNS Subscribe message [RFC8765] instead of a classic DNS query to fetch the information on behalf of a DoC client. If this is not supported by the DoC server, it **MUST** act as if the resource were not observable.

Whenever the DoC server receives a DNS Push message [RFC8765] from the DNS infrastructure for an observed resource record, the DoC server sends an appropriate Observe response to the DoC client.

If no more DoC clients observe a resource record for which the DoC server has an open subscription, the DoC server **MUST** use a DNS Unsubscribe message [RFC8765] to close its subscription to the resource record as well.

5.2. OSCORE

It is **RECOMMENDED** to carry DNS messages encrypted using OSCORE [RFC8613] between the DoC client and the DoC server. The establishment and maintenance of the OSCORE Security Context is out of the scope of this document.

If cache retrieval of OSCORE responses is desired, it can be achieved, for instance, by using the method defined in [I-D.amsuess-core-cachable-oscore]. This has, however, implications on message sizes and security properties, which are compiled in that document.

5.3. Mapping DoC to DoH

This document provides no specification how to map between DoC and DoH, e.g., at a CoAP-HTTP-proxy, and it is **NOT RECOMMENDED**. Rewriting the FETCH method (Section 4.2) and the TTL rewriting (Section 4.3.2) as specified in this draft would be non-trivial. It is **RECOMMENDED** to use a DNS forwarder to map between DoC and DoH, as would be the case for mapping between any other DNS transport.

6. Considerations for Unencrypted Use

The use of DoC without a security mode of CoAP is **NOT RECOMMENDED**. Without a security mode, a large number of possible attacks need to be evaluate in the context of the application's threat model. This includes threats that are mitigated even by DNS over UDP: For example, the random ID of the DNS header afford some protection against off-path cache poisoning attacks---a threat that might be mitigated by using random large token values in the CoAP request.

7. Implementation Status

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available

implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

7.1. DoC Client

The authors of this document provide a DoC client implementation available in the IoT operating system RIOT.

Level of maturity: production

Version compability: draft-ietf-core-dns-over-coap-04

License: LGPL-2.1

Contact information: Martine Lenders <m.lenders@fu-berlin.de>

Last update of this information: October 2023

7.2. DoC Server

The authors of this document provide a DoC server implementation in Python.

Level of maturity: production

Version compability: draft-ietf-core-dns-over-coap-04

License: MIT

Contact information: Martine Lenders <m.lenders@fu-berlin.de>

Last update of this information: October 2023

8. Security Considerations

When using unencrypted CoAP (see Section 6), setting the ID of a DNS message to 0 as specified in Section 4.2.2 opens open the DNS cache of a DoC client to cache poisoning attacks via response spoofing. This documents requires an unpredictable CoAP token in each DoC query from the client when CoAP is not secured to mitigate such an attack over DoC (see Section 6).

For encrypted usage with DTLS or OSCORE the impact of a fixed ID on security is limited, as both harden against injecting spoofed responses. Consequently, it is of little concern to leverage the benefits of CoAP caching by setting the ID to 0.

9. IANA Considerations

9.1. 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 to the "application/dns-message" media type from the "Media Types" registry:

Media-Type: application/dns-message

Encoding: -

Id: 553 (suggested)

Reference: [TBD-this-spec]

9.2. 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

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>.

[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>.

[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>.

[RFC9147] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", RFC 9147, DOI 10.17487/RFC9147, April 2022, <https://www.rfc-editor.org/rfc/rfc9147>.

10.2. Informative References

[DoC-paper] Lenders, M., Amsüss, C., Gündogan, C., Nawrocki, M., Schmidt, T., and M. Wählisch, "Securing Name Resolution in the IoT: DNS over CoAP", Association for Computing Machinery (ACM), Proceedings of the ACM on Networking vol. 1, no. CoNEXT2, pp. 1-25, DOI 10.1145/3609423, September 2023, <https://doi.org/10.1145/3609423>.

[I-D.amsuess-core-cachable-oscore] Amsüss, C. and M. Tiloca, "Cacheable OSCORE", Work in Progress, Internet-Draft, draft-amsuess-core-cachable-oscore-08, 10 January 2024, <https://datatracker.ietf.org/doc/html/draft-amsuess-core-cachable-oscore-08>.

[I-D.ietf-core-href] Bormann, C. and H. Birkholz, "Constrained Resource Identifiers", Work in Progress, Internet-Draft, draft-ietf-core-href-14, 9 January 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-core-href-14>.

[I-D.lenders-core-dnr] Lenders, M. S., Amsüss, C., Schmidt, T. C., and M. Wählisch, "Discovery of Network-designated CoRE Resolvers", Work in Progress, Internet-Draft, draft-lenders-core-dnr-00, 4 March 2024, <https://datatracker.ietf.org/doc/html/draft-lenders-core-dnr-00>.

[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>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/rfc/rfc7942>.

[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>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/rfc/rfc8499>.

[RFC8765] Pusateri, T. and S. Cheshire, "DNS Push Notifications", RFC 8765, DOI 10.17487/RFC8765, June 2020, <https://www.rfc-editor.org/rfc/rfc8765>.

[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>.

[RFC9460] Schwartz, B., Bishop, M., and E. Nygren, "Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records)", RFC 9460, DOI 10.17487/RFC9460, November 2023, <https://www.rfc-editor.org/rfc/rfc9460>.

[RFC9461] Schwartz, B., "Service Binding Mapping for DNS Servers", RFC 9461, DOI 10.17487/RFC9461, November 2023, <https://www.rfc-editor.org/rfc/rfc9461>.

[RFC9462] Pauly, T., Kinnear, E., Wood, C. A., McManus, P., and T. Jensen, "Discovery of Designated Resolvers", RFC 9462, DOI 10.17487/RFC9462, November 2023, <https://www.rfc-editor.org/rfc/rfc9462>.

[RFC9463] Boucadair, M., Ed., Reddy.K, T., Ed., Wing, D., Cook, N., and T. Jensen, "DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers (DNR)", RFC 9463, DOI 10.17487/RFC9463, November 2023, <https://www.rfc-editor.org/rfc/rfc9463>.

Appendix A. Evaluation

The authors of this document presented the design, implementation, and analysis of DoC in their paper "Securing Name Resolution in the IoT: DNS over CoAP" [DoC-paper].

Appendix B. Change Log

B.1. Since draft-ietf-core-dns-over-coap-05

*Add references to relevant SVCB/DNR RFCs and drafts

B.2. Since draft-ietf-core-dns-over-coap-04

*Add note on cachable OSCORE

*Address early IANA review

B.3. Since draft-ietf-core-dns-over-coap-03

*Amended Introduction with short contextualization of constrained environments

*Add Appendix A on evaluation

B.4. Since draft-ietf-core-dns-over-coap-02

*Move implementation details to Implementation Status (in accordance with [RFC7942])

*Recommend root path to keep the CoAP options small

*Set Content-Format for application/dns-message to 553

*SVCB/DNR: Move to Server Selection Section but leave TBD based on DNSOP discussion for now

*Clarify that DoC and DoC are disjunct

*Clarify mapping between DoC and DoH

*Update considerations on unencrypted use

*Don't call OSCORE end-to-end encrypted

B.5. Since draft-ietf-core-dns-over-coap-01

*Specify DoC server role in terms of DNS terminology

*Clarify communication of DoC to DNS infrastructure is agnostic of the transport

*Add subsection on how to implement DNS Push in DoC

*Add appendix on reference implementation

B.6. Since draft-ietf-core-dns-over-coap-00

*SVGify ASCII art

*Move section on "DoC Server Considerations" (was Section 5.1) to its own draft (draft-lenders-dns-cns)

*Replace layer violating statement for CON with statement of fact

*Add security considerations on ID=0

B.7. Since draft-lenders-dns-over-coap-04

*Removed change log of draft-lenders-dns-over-coap

Acknowledgments

The authors of this document want to thank Carsten Bormann, Ben Schwartz, Marco Tiloca, and Tim Wicinski for their feedback and comments.

Authors' Addresses

Martine Sophie Lenders
TUD Dresden University of Technology
Helmholtzstr. 10
D-01069 Dresden
Germany

Email: martine.lenders@tu-dresden.de

Christian Amsüss

Email: christian@amsuess.com

Cenk Gündoğan
Huawei Technologies
Riesstrasse 25
D-80992 Munich
Germany

Email: cenk.gundogan@huawei.com

Thomas C. Schmidt
HAW Hamburg
Berliner Tor 7

D-20099 Hamburg
Germany

Email: t.schmidt@haw-hamburg.de

Matthias Wählisch
TUD Dresden University of Technology & Barkhausen Institut
Helmholtzstr. 10
D-01069 Dresden
Germany

Email: m.waehlisch@tu-dresden.de