

Workgroup: Network Working Group

Internet-Draft:

draft-bormann-core-responses-02

Published: 3 March 2024

Intended Status: Informational

Expires: 4 September 2024

Authors: C. Bormann

C. Amsüss

Universität Bremen TZI

CoAP: Non-traditional response forms

Abstract

In CoAP as defined by RFC 7252, responses are always unicast back to a client that posed a request. The present memo describes two forms of responses that go beyond that model. These descriptions are not intended as advocacy for adopting these approaches immediately, they are provided to point out potential avenues for development that would have to be carefully evaluated.

About This Document

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

Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-bormann-core-responses/>.

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

Source for this draft and an issue tracker can be found at <https://github.com/core-wg/core-responses>.

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 4 September 2024.

Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- 1. [Introduction](#)
 - 1.1. [Terminology](#)
- 2. [Sending non-traditional responses](#)
 - 2.1. [Preconditions to sending non-traditional responses](#)
 - 2.2. [Responses without request](#)
- 3. [OSCORE processing for non-traditional responses](#)
- 4. [Response with embedded request](#)
- 5. [Response for configured request](#)
 - 5.1. [Examples for configured requests](#)
 - 5.1.1. [Example: Periodic request](#)
 - 5.1.2. [Example: Event driven request](#)
 - 5.1.3. [Example: Configured observe](#)
 - 5.2. [Multicast responses](#)
 - 5.3. [Respond-To option](#)
 - 5.4. [Leisure-For-Responses Option](#)
- 6. [IANA Considerations](#)
- 7. [Security Considerations](#)
- 8. [References](#)
 - 8.1. [Normative References](#)
 - 8.2. [Informative References](#)
- [Appendix A. CoAP extensions explained by non-traditional responses](#)
 - A.1. [Observation](#)
 - A.2. [Responses to multicast requests](#)
 - A.3. [Triangular responses \(Response-To\)](#)
 - A.4. [Other current documents](#)
- [Acknowledgements](#)
- [Authors' Addresses](#)

1. Introduction

In CoAP as defined by RFC 7252, responses are always unicast back to a client that posed a request. A server may want to send a response to a request that it did not receive, may want to multicast a response, or both.

The descriptions in this specification are not intended as advocacy for adopting these approaches immediately, they are provided to point out potential avenues for development that would have to be carefully evaluated.

1.1. Terminology

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.

The term "byte" is used in its now customary sense as a synonym for "octet".

Terms used in this draft:

Non-traditional response: A response that is not the single response generated for a request received on the same transport.

Non-matching response: A response that has properties (typically options) that make it incompatible with the original request, and thus in particular unsuitable as a cached response to that request (but possibly suitable to populate the cache for a similar request). Options that make a response non-matching need to be proxy unsafe.

For example, a Block2 response with a different value of block number $\times$ block size than indicated in the request is non-matching.

Configured request: A request that reaches the server in another way than by transmitting a usual CoAP request on the same communication channel a response is expected on.

Embedded request: A request that is provided by the server to the recipient of its response by embedding it into the response.

2. Sending non-traditional responses

Non-traditional responses are sets of responses produced for a single request, or responses sent without a transmitted request.

Where tokens are involved, all non-traditional responses use the request's token; in any case, they are bound to the original request (e.g. by using the same request_kid/request_piv pair in OSCORE [RFC8613]). Where message IDs are involved, one of the non-traditional response (the first sent, not necessarily the first received as generally the network might reorder messages) can be sent as a piggybacked response in an ACK (thus sharing the request's message ID), the others are CON or NON responses.

Some established responses (observations defined in [RFC7641], and responses to multicast requests in [I-D.ietf-core-groupcomm-bis]) match this definition and already follow the guidance set out here for non-traditional responses; [Appendix A](#) gives details for them.

A second response differing from the first that can be sent by a non-deduplicating server responding to a retransmission of a request is not non-traditional because there is a second request -- that is probably the last corner case at the line separating traditional from non-traditional responses.

2.1. Preconditions to sending non-traditional responses

A server may send multiple responses to a request if there is any property in the request that indicates the client's intention to receive them. This is typically indicated by a request option, and rarely in external properties of the message (in the multicast case, the destination address).

A mechanism for eliciting multiple responses must specify the conditions under which a token gets freed, as the traditional arrival of the response is insufficient. It may also specify for which requests the token can be reused immediately in follow-up requests. On unordered transports, or when it's a client's follow-up request and not a response that terminates the token, the client needs to wait with reuse until no reordered non-traditional responses can be expected anymore.

If a non-traditional response answers the original request, no further action is required (this is the case of observation: ordering is added on top of that to ensure that only the latest response is used). If the response does not answer the original request, it must be non-matching, either by an option introduced with the eliciting option or by a generic option like Response-For.

2.2. Responses without request

Endpoints may agree out of band on a token (or other request-matching details). One way to do that is to exchange a "phantom request", which is a request that client and server will agree to

have sent and received, respectively, without it actually being sent between those endpoints.

As tokens are managed by the client, that request needs to be generated by the client, or in close collaboration with the client (for example by the client allowing a third party to use a subset of its token values in order to set up non-traditional responses).

3. OSCORE processing for non-traditional responses

OSCORE [[RFC8613](#)] is built with the general assumption that requests are processed into exactly one response. The specification contains explicit provisions for Observe requests, and a whole protocol extension for multicast requests.

OSCORE's binding between requests and responses remains unmodified: Each response is cryptographically bound to an OSCORE request. Therefore, any phantom request needs to be an OSCORE request as well, and the parties need to agree on the sender and sequence number of the phantom request. An easy way to do that securely is to deliver the phantom request in a way that the server can do the full OSCORE request processing on it. The server may process the OSCORE request into internal data structures at reception time, or may process it whenever a response is to be sent. In the latter case, it may need to relax the requirements of Section [8.2](#) ([Verifying the Request](#)) of [[RFC8613](#)] item 3.

To avoid reinventing the same rules as for Observe requests for any other non-traditional response, this document defines a set of processing instructions which can be referenced when specifying their options. These rules generalize Sections [8.3](#) ([Protecting the Response](#)) and [8.4](#) ([Verifying the Response](#)) of [[RFC8613](#)]:

*In 8.3 step 3, "use the AEAD nonce from the request" is only an option once, i.e., after the sequence number expressed in that request was removed from the replay window. This option is usually taken in the first response, necessitating the use of encoded Sender Sequence Numbers in later responses. (Non-traditional responses such as Observe that rely on message ordering may require that the request's nonce is used either in the first response or not at all.) CA: We could also just mandate the "either the first or never" behavior.

As a convenient effect, this generalized rule also implies that when a server performs Appendix [B.1.2](#) ([Replay Window](#)) of [[RFC8613](#)], it needs to use its own Partial IV for the nonce (which without this generalized rule necessitated a "MUST" statement in the appendix).

It is unclear why one would delay sending the one response that has the least overhead, but that may be lack of imagination. An approach where instances can not generally be duplicated and are used at most once (as in an affine type system) can make this doable in a safe way. In the end it's a tradeoff between implementer flexibility and specification simplicity.

*In 8.4 between steps 5 and 6, the Sender Sequence Number of the response establishes an order in the received messages, which users of non-traditional responses may rely on. If an option specified that only the first response may use the request's nonce, then the one response that uses it is ordered before all other responses to the same request.

*If the handling of multiple responses is not idempotent, then at 8.4 step 5:

- For responses that use a Sender Sequence Number from the server, the client consults the replay window before decryption, and removes its number from the replay window after successful decryption.

- For responses that use the request's Sender Sequence Number, duplication is tracked for each request.

As a simplification, applications that only process the latest response may track the latest sequence number for deduplication.

*In 8.4 step 8, the Option establishing the non-traditional responses may specify that error conditions processing a response are not fatal for the whole request. This should be done when an Option allows immediate follow-up requests. This is the case for the Observe option: When an observation is refreshed, a response encrypted with the earlier request's request_kid may still be in flight. That in-flight response will fail decryption, but responses generated after the server has received the refresh will be decryptable again.

4. Response with embedded request

A server can send a response to a request that it did not actually receive by embedding the request which the response answers in the response.

The option "Response-For" contains a request packaged as in [Section 5.3](#) of [\[RFC8613\]](#). The response is then intended to serve as a response to this request.

No.	C	U	N	R	Name	Format	Length	Default
TBD	C	-	-	-	Response-For	opaque	0-1023	(none)

Table 1: The Response-For Option

The CoAP Token becomes meaningless for this form of response; responses with embedded requests are therefore sent with a zero-length Token. (In essence, the "Response-For" option takes the place of the request the Token usually stands for.)

The congestion control considerations for confirmable and non-confirmable messages apply unchanged.

5. Response for configured request

A request may reach the server using a different means than that used for the response. For instance, the request may be configured in the server. Without limiting generality, we speak about *configured requests*.

The client **MUST** be cognizant of that configuration as the request uses a token from the token name space it controls.

5.1. Examples for configured requests

5.1.1. Example: Periodic request

A server may be configured to act on a configured request every day at 12:00.

5.1.2. Example: Event driven request

A server may be configured to act on a configured request each time it reboots.

5.1.3. Example: Configured observe

A server may be configured with a GET request from a client that includes an Observe option with value 0. This means that the server will send updates to the state of the resource addressed by the GET request to the configured address of the client.

The considerations of [Section 4.5](#) of [\[RFC7641\]](#) apply. How losing interest reflects back into to configuration and whether there is some form of error notification to the source of the configuration is out of scope of the present specification.

5.2. Multicast responses

A server **MAY** send a response to a multicast address. (This needs to be a response to a configured request as a normal request cannot be sent *from* a multicast address.)

Note that, as the originator of a multicast response is a unicast address, the relaxation of matching rules described in [Section 8.2](#) of [\[RFC7252\]](#) does not apply.

The token space in CoAP is owned by the client, which is identified by a transport endpoint (address/port). Here, the address is a multicast address, so the token name space is shared by all nodes joined to that multicast address. The assumption for multicast responses is that, for each multicast group, there is some form of management for the token space (and the port number) that everyone can participate that needs to join that multicast group; the specific form of management is out of the scope of this specification. Note that this means that multicast responses **MUST NOT** be sent to unmanaged multicast addresses such as All CoAP Nodes ([Section 12.8](#) of [\[RFC7252\]](#)).

Multicast responses are always non-confirmable. The congestion control considerations for non-confirmable multicast messages apply unchanged.

5.3. Respond-To option

What has been called "configured request" here may also be triggered by a usual CoAP request that carries the Respond-To option. (The term "configured request" is still appropriate as the server ought to be configured to accept this option; see [Section 7](#).)

If a single client wants to request a server to send the response to a specific multicast address, it can include the "Respond-To" option. This contains an opaque string with the port number as a 16-bit number (in network byte order), followed by the IP address (4-byte IPv4 or 16-byte IPv6).

No.	C	U	N	R	Name	Format	Length	Default
TBD	C	U	-	-	Respond-To	opaque	6-18	(none)

Table 2: The Respond-To Option

5.4. Leisure-For-Responses Option

This new option indicates a number expressed as a uint. It allows the server to send that number of non-traditional response messages in addition to the requested response. They are to be sent without undue delay after the original response.

No.	C	U	N	R	Name	Format	Length	Default
TBD		U	-		Leisure-For-Responses	uint	1-4	0

Table 3: The Leisure-For-Responses Option

The option is elective, but unsafe for proxies (as the option would otherwise cause multiple responses to a proxy that expects only one and that needs to be a matching response). A proxy that chooses not to implement it may forward the request with the Leisure-For-Responses option removed.

On its own, the option does not indicate which kind of additional responses the client would expect (though further elective proxy-safe no-cache-key options can be added on top of that to give better guidance), and the server may choose not to send any at all.

Intermediaries may add or remove the option, and use incoming responses to populate their cache. They may serve additional responses from their cache, but in most cases the sensible course of action is to forward the additional responses the origin server sends.

Use cases for Leisure-For-Responses include sending further blocks in a Block2 transfer (which are obviously non-matching and thus don't need a Response-For), or serving follow-up documents (a response containing a single link can be followed by a representation of the linked resource, which needs a Request-For header that indicates the URI).

6. IANA Considerations

This draft adds the following option numbers to the CoAP Option Numbers registry of [\[RFC7252\]](#):

Number	Name	Reference
TBD	Response-For	RFCThis
TBD	Respond-To	RFCThis
TBD	Leisure-For-Responses	RFCThis

Table 4: CoAP Option Numbers

7. Security Considerations

TBD

(Clearly, multicast responses pose a potential for amplification, in particular if unverified sources can cause them via Respond-To. Discuss how to mitigate.)

A Respond-To option can be used to incite a server to send data to a third party. This ought not be done blindly, i.e., only with considered application assent.

The CoAP request/response mechanism allows the client to ascertain a level of authentication (not resistant though to on-path attackers unless the communication is protected) and freshness of the response: The Token echoed in the response shows that the responder had knowledge of the (fresh) request ([Section 5.3.1](#) of [\[RFC7252\]](#)). Responses with embedded requests can not be authenticated or checked for freshness this way. Their content therefore is less trustworthy than normal responses unless authenticated in another way (e.g., via [\[RFC8613\]](#)).

8. References

8.1. Normative References

- [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>>.
- [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>>.
- [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>>.

8.2. Informative References

- [I-D.ietf-core-groupcomm-bis] Dijk, E., Wang, C., and M. Tiloca, "Group Communication for the Constrained Application Protocol (CoAP)", Work in Progress, Internet-Draft, draft-ietf-core-groupcomm-bis-10, 23 October 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-core-groupcomm-bis-10>>.
- [I-D.ietf-core-observe-multicast-notifications] Tiloca, M., Höglund, R., Amsüss, C., and F. Palombini, "Observe Notifications as CoAP Multicast Responses", Work in Progress, Internet-Draft, draft-ietf-core-observe-

multicast-notifications-07, 23 October 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-core-observe-multicast-notifications-07>>.

[I-D.tiloca-core-groupcomm-proxy] Tiloca, M. and E. Dijk, "Proxy Operations for CoAP Group Communication", Work in Progress, Internet-Draft, draft-tiloca-core-groupcomm-proxy-09, 31 August 2023, <<https://datatracker.ietf.org/doc/html/draft-tiloca-core-groupcomm-proxy-09>>.

[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>>.

Appendix A. CoAP extensions explained by non-traditional responses

A.1. Observation

This section describes the Observe option [[RFC7641](#)] in the terms of this document, [so nothing in here should contradict that document].

When Observe:0 is present in a request, this sets up non-traditional responses until either of the following conditions is met:

- *A follow-up request on the same token carries an Observe:1 option.

(This is primarily in here because; Observe:1 and No-Response:any could be combined; otherwise, the other conditions suffice).

- *Any response does not carry an Observe option.

- *Any response has a non-successful status.

Follow-up requests are limited to extending the request ETag set. Responses are obviously non-matching by their Observe option; each hop discards the Observe option for the purpose of caching and refreshes its cache with the most recent one as per the Observe value.

A.2. Responses to multicast requests

As with observe, this just phrases the existing mechanism in the context of this generalization.

When the destination address of a CoAP request is a multicast address, that token is valid for any member of that group (which, for the purpose of the client, is any server at all) on any port.

(Except for that the implications of having received a multicast request still need to be followed, it might be seen as a template for creating a phantom request to any endpoint, if that suits the reader's mental model.)

Responses can only be sent for up to the deployment's Leisure time (see [Section 8.2](#) of [\[RFC7252\]](#)) plus the application's timeout (in proxy situations, this needs to be communicated explicitly in the Multicast-Signaling option of [\[I-D.tiloca-core-groupcomm-proxy\]](#)).

A.3. Triangular responses (Response-To)

The Response-To option can be viewed as a shorthand notation for "Consider this a No-Response:any request, but take a copy of it, make it into a CoAP-over-UDP request with that particular address as a source and any address of yours as a response, and treat that as a phantom request".

[It may make sense to add an explicit return token, and include a No-Response option; that might allow it to be used even across proxies.]

A.4. Other current documents

[\[I-D.ietf-core-observe-multicast-notifications\]](#) is a straightforward application of the phantom requests (the concept was developed there); Leisure-For-Responses could help it around the topic of joining a multicast group securely through a proxy.

[\[I-D.tiloca-core-groupcomm-proxy\]](#) seems to fit well with the concepts here as well, and might be simplified by it both in terminology and by replacing Response-Forwarding with Response-For(Proxy-Scheme, Uri-Host).

Acknowledgements

TBD

Authors' Addresses

Carsten Bormann
Universität Bremen TZI
Postfach 330440
D-28359 Bremen
Germany

Phone: [+49-421-218-63921](tel:+49-421-218-63921)
Email: cabo@tzi.org

Christian Amsüss

Email: christian@amsuess.com