CoRE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 14, 2021

M. Tiloca
RISE AB
E. Dijk
IoTconsultancy.nl

July 13, 2020

Proxy Operations for CoAP Group Communication
draft-tiloca-core-groupcomm-proxy-01

Abstract

This document specifies the operations performed by a forward-proxy, when using the Constrained Application Protocol (CoAP) in group communication scenarios. Proxy operations involve the processing of individual responses from servers, as reply to a single request sent by the client over unicast to the proxy, and then distributed by the proxy over IP multicast to the servers. When receiving the different responses via the proxy, the client is able to distinguish them and their originator servers, by acquiring their addressing information.

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 January 14, 2021.

Table of Contents

1. Introduction

The Constrained Application Protocol (CoAP) [RFC7252] allows the presence of forward-proxies, as intermediary entities supporting clients to perform requests on their behalf.

CoAP supports also group communication over IP multicast [I-D.ietf-core-groupcomm-bis], where a group request can be addressed to multiple recipient servers, each of which may reply with an individual unicast response. As discussed in Section 2.3.3 of [I-D.ietf-core-groupcomm-bis], this group communication scenario poses a number of issues and limitations to proxy operations.

In particular, the client sends a single unicast request to the proxy, which the proxy forwards to a group of servers over IP multicast. Later on, the proxy delivers back to the client multiple responses to the original unicast request. As defined by [RFC7252], the multiple responses are delivered to the client inside separate CoAP messages, all matching (by Token) to the client's original unicast request. A possible alternative approach of performing aggregation of responses into a single CoAP response would require a specific aggregation content-format, which is not available yet. Both these approaches have open issues.

This specification considers the former approach of how the proxy forwards the individual responses to a CoAP group request back to the client. The described method addresses all the related issues raised in Section 2.3.3 of [I-D.ietf-core-groupcomm-bis].

To this end, a dedicated signaling protocol is defined, using two new CoAP options. In particular, the client can explicitly confirm its support for receiving multiple responses to a proxied unicast request, i.e. one per originator server, and for how long it is willing to wait for responses. Also, each server originating a response indicates to the client its own addressing information. This enables the client to distinguish the multiple, diffent responses by origin and to possibly contact one or more of the individual servers by a unicast request, optionally bypassing the forward-proxy.

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.

Readers are expected to be familiar with terms and concepts defined in CoAP [RFC7252], Group Communication for CoAP [I-D.ietf-core-groupcomm-bis], OSCORE [RFC8613] and Group OSCORE [I-D.ietf-core-oscore-groupcomm].

2. The Multicast-Signaling Option

The Multicast-Signaling Option defined in this section has the properties summarized in Figure 1, which extends Table 4 of [RFC7252]. The option is intended only for CoAP requests.

Since the option is not Safe-to-Forward, the column "N" indicates a dash for "not applicable". The value of the Multicast-Signaling

Option specifies a timeout value in seconds, encoded as an unsigned integer (see Section 3.2 of [RFC7252]).

No.	C	U	N	R	Name	Format	Length	Default
TBD1		x	-		Multicast-Signaling	uint	1-5 B	(none)

C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable
(*) See below.

Figure 1: The Multicast-Signaling Option.

This document specifically defines how this option is used by a client, to indicate to a forward-proxy its support for and interest in receiving multiple responses to a proxied CoAP group request, i.e. one per originator server, and for how long it is willing to wait for receiving responses via that proxy (see Section 5.1 and Section 5.2).

The client, when sending a CoAP group request to a proxy via IP unicast, to be forwarded by the proxy to a targeted group of servers, includes the Multicast-Signaling Option into the request. The option value indicates after what time period in seconds the client will stop accepting responses matching its original unicast request, with the exception of notifications if CoAP Observe is used [RFC7641]. This allows the intermediary proxy to stop forwarding responses back to the client, if received from the servers later than a timeout expiration.

The Multicast-Signaling Option is of class U for OSCORE [RFC8613][I-D.ietf-core-oscore-groupcomm].

3. The Response-Forwarding Option

The Response-Forwarding Option defined in this section has the properties summarized in Figure 2, which extends Table 4 of [RFC7252]. The option is intended only for CoAP responses, and builds on the Base-Uri option from Section 3 of [I-D.bormann-coap-misc].

Since the option is intended only for responses, the column "N" indicates a dash.

No.	C	U	N	R	Name	Format	Length	Default
TBD2			-		Response-Forwarding	string	8-20 B	(none)

C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable
(*) See below.

Figure 2: The Response-Forwarding Option.

This document specifically defines how this option is used by a proxy that forwards a request originated by a client over IP multicast.

Upon receiving a response to that request from a server, the proxy includes the Response-Forwarding Option into the response sent to the originator client (see Section 5). The proxy uses the option to indicate to the client the addressing information of the server generating the response.

The client can use the addressing information of the server specified in the option to identify the response originator and possibly send later unicast requests to directly, or via the proxy as CoAP unicast requests.

The option value is an absolute-URI with no query component ([RFC3986], Section 4.3). If the port number is omitted in the authority component, it is assumed that the port number where to send unicast requests to the server is the same port number specified in the group URI of the original unicast CoAP group request sent to the proxy (see Section 5.1).

The Response-Forwarding Option is of class E for OSCORE [RFC8613][I-D.ietf-core-oscore-groupcomm].

4. Requirements and Objectives

This specification assumes that the following requirements are fulfilled.

- REQ1. The CoAP proxy is explicitly configured (white-list) to allow proxied CoAP group requests from specific client(s).

- REQ2. The CoAP proxy MUST identify a client sending a CoAP group request, in order to verify whether the client is white-listed to do so. For example, this can rely on one of the following.

    * A DTLS channel [RFC6347][I-D.ietf-tls-dtls13] between the client and the proxy, where the client has also been authenticated during the secure channel establishment.

    * A pairwise OSCORE Security Context between the client and the proxy, as described in Appendix A.

- REQ3. If secure, end-to-end communication is required between the client and the servers in the CoAP group, exchanged messages MUST be protected by using Group OSCORE [I-D.ietf-core-oscore-groupcomm], as discussed in Section 5.2 of [I-D.ietf-core-groupcomm-bis]. This requires the client and the servers to have previously joined the correct OSCORE group, for instance by using the approach described in [I-D.ietf-ace-key-groupcomm-oscore]. The correct OSCORE group to join can be pre-configured or alternatively discovered, for instance by using the approach described in [I-D.tiloca-core-oscore-discovery].

This specification defines how to achieve the following objectives.

- OBJ1. The CoAP proxy gets an indication from the client that it is in fact interested to and capable to receive multiple responses to its unicast request containing a CoAP group URI.

- OBJ2. The CoAP proxy learns how long it should wait for responses to a proxied request, before starting to ignore following responses (except for notifications, if CoAP Observe is used [RFC7641]).

- OBJ3. The CoAP proxy returns individual unicast responses to the client, each of which matches the original unicast request to the proxy.

- OBJ4. The CoAP client is able to distinguish the different responses to the original unicast request, as well as their corresponding originator servers.

- OBJ5. The CoAP client is enabled to optionally contact one or more of the responding servers in the future, either directly or via the CoAP proxy.

5. Protocol Description

5.1. Request Sending

In order to send a request addressed to a group of servers via the CoAP proxy, the client proceeds as follows.

1. The client prepares a request addressed to the CoAP proxy. The request specifies the group URI as a string in the Proxi-URI option, or by using the Proxy-Scheme option with the group URI constructed from the URI-* options (see Section 2.3.3 of [I-D.ietf-core-groupcomm-bis]).

2. The client MUST retain the Token value used for this original unicast request beyond the reception of a first response matching it. To this end, the client follows the same rules for Token retention defined for multicast requests in Section 2.3.1 of [I-D.ietf-core-groupcomm-bis]. In particular, it picks an amount of time T to wait for before freeing up the Token value, such that:

   * T is smaller than the amount of time Tr it may pick to wait for before potentially reusing the Token value.

   * T includes the expected worst-case time taken by the request and response processing on the forward-proxy plus the servers in the addressed CoAP group.

   * T includes the expected worst-case round-trip delay between client and proxy, and between proxy and servers.

3. The client includes the Multicast-Signaling Option defined in Section 2 into the unicast request to send to the proxy. The option value specifies an amount of time T' < T. The difference (T - T') should include the expected worst-case round-trip time between the client and the forward-proxy.

4. The client processes the request as defined in [I-D.ietf-core-groupcomm-bis], and also as in [I-D.ietf-core-oscore-groupcomm] when secure group communication is used between the client and the servers.

5. If OSCORE is used to protect the leg between the client and the proxy (see REQ2 in Section 4), the client (further) protects the unicast request as resulting at the end of step 4. In particular, the client uses the pairwise OSCORE Security Context it has with the proxy, as described in Appendix A.1.

6. The client sends the request to the proxy as a unicast CoAP message.

5.2. Request Processing at the Proxy

Upon receiving the request from the client, the proxy proceeds as follows.

1. If OSCORE is used to protect the leg between the client and the proxy (see REQ2 in Section 4), the proxy decrypts the request using the pairwise OSCORE Security Context it has with the client, as described in Appendix A.4.

2. The proxy identifies the client, and verifies that the client is in fact white-listed to have its requests proxyied to CoAP group URIs.

3. The proxy verifies the presence of the Multicast-Signaling Option, as a confirmation that the client is fine to receive multiple responses matching the same original request.

4. The proxy retrieves the value T' from the Multicast-Signaling Option, and then removes the option from the client's request.

5. The proxy forwards the client's request to the group of servers. In particular, the proxy sends it as a CoAP group request over IP multicast, addressed to the group URI specified by the client.

6. The proxy sets a timeout with the value T' retrieved from the Multicast-Signaling Option of the original unicast request. The proxy will ignore responses to the forwarded group request coming from servers, if received after the timeout expiration, with the exception of Observe notifications (see Section 5.4).

5.3. Request and Response Processing at the Server

Upon receiving the group request from the proxy, a server proceeds as follows.

1. The server processes the group request as defined in [I-D.ietf-core-groupcomm-bis], and also as in [I-D.ietf-core-oscore-groupcomm] when secure group communication is used between the client and the server.

2. The server processes the response to be forwarded back to the client as defined in [I-D.ietf-core-groupcomm-bis], and also as in [I-D.ietf-core-oscore-groupcomm] when secure group communication is used between the client and the server.

5.4. Response Processing at the Proxy

Upon receiving a response matching the group request before the amount of time T' has elapsed, the proxy proceeds as follows.

1. The proxy includes the Response-Forwarding Option defined in Section 3 into the response. The proxy specifies as option value the addressing information of the server generating the response, encoded as an absolute-URI as defined in Section 3. In particular:

   * The authority component MUST specify the server IPv6 address if the multicast request was destined for an IPv6 multicast address, and MUST specify the server IPv4 address if the multicast request was destined for an IPv4 address.

   * The authority component MUST specify the port number of the server as the source port number of the response, if this differs from the port number specified in the group URI of the original unicast CoAP group request (see Section 5.1). Otherwise, the authority component MAY omit the port number.

   When using Observe [RFC7641], the proxy includes the Response-Forwarding Option also in every notification, including non-2.xx notifications resulting in removing the proxy from the list of observers.

2. If OSCORE is used to protect the leg between the client and the proxy (see REQ2 in Section 4), the proxy (further) protects the response using the pairwise OSCORE Security Context it has with the client, as described in Appendix A.3.

3. The proxy forwards the response back to the client.

Upon timeout expiration, i.e. T' seconds after having sent the group request over IP multicast, the proxy frees up its local Token value associated to that request. Thus, following late responses to the same group request will be discarded and not forwarded back to the client.

When using CoAP Observe [RFC7641], the Token value is freed up only if, after the timeout expiration, no 2.xx (Success) responses matching the group request and also including an Observe option have been received. Then, as long as observations are active with servers in the group for the target resource of the group request, notifications from those servers are forwarded back to the client.

5.5. Response Processing at the Client

Upon receiving from the proxy a response matching the original unicast request before the amount of time T has elapsed, the client proceeds as follows.

1. The client processes the response as defined in [I-D.ietf-core-groupcomm-bis].

2. If OSCORE is used to protect the leg between the client and the proxy (see REQ2 in Section 4), the client decrypts the response using the pairwise OSCORE Security Context it has with the proxy, as described in Appendix A.4.

3. If secure group communication is used between the client and the servers, the client processes the response, possibly as outcome of step 2, as defined in [I-D.ietf-core-oscore-groupcomm].

4. The client identifies the originator server, whose addressing information is specified as value of the Response-Forwarding Option. If the port number is omitted in the value of the Response-Forwarding Option, the client MUST assume that the port number where to send unicast requests to the server is the same port number specified in the group URI of the original unicast CoAP group request sent to the proxy (see Section 5.1).

In particular, the client is able to distinguish different responses as originated by different servers. Optionally, the client may contact one or more of those servers individually, i.e. directly (bypassing the proxy) or indirectly (via a proxied CoAP unicast request).

Upon the timeout expiration, i.e. T seconds after having sent the original unicast request to the proxy, the client frees up its local Token value associated to that request. Note that, upon this timeout expiration, the Token value is not eligible for possible reuse yet (see Section 5.1). Thus, until the actual amount of time before enabling Token reusage has elapsed, following late responses to the same request forwarded by the proxy will be discarded, as not matching (by Token) any active request from the client.

When using CoAP Observe [RFC7641], the Token value is freed up only if, after the timeout expiration, no 2.xx (Success) responses matching the original unicast request and also including an Observe option have been received. If at least one such response has been received, the client continues receiving those notifications as forwarded by the proxy, as long as the observation for the target resource of the original unicast request is active.

6. Security Considerations

The security considerations from [RFC7252][I-D.ietf-core-groupcomm-bis][RFC8613][I-D.ietf-core-oscore-groupcomm] hold for this document.

Furthermore, the following additional considerations hold.

6.1. Client Authentication

As per the requirement REQ2 (see Section 4), the client has to authenticate to the proxy when sending a group request to forward. This leverages an established security association between the client and the proxy, that the client uses to protect the group request, before sending it to the proxy.

Note that, if the group request is (also) protected with Group OSCORE, i.e. end-to-end between the client and the servers, the proxy can authenticate the client by successfully verifying the countersignature embedded in the group request. This requires that, for each client to authenticate, the proxy stores the public key used by that client in the OSCORE group.

Nevertheless, the client SHOULD still rely on a full-fledged, pairwise secure association with the proxy. In addition to ensuring the integrity of group requests sent to the proxy (see Section 6.2 and Section 6.3), this prevents the proxy from forwarding replayed group requests with a valid countersignature, as injected by an active, on-path adversary.

6.2. Multicast-Signaling Option

The Multicast-Signaling Option is of class U for OSCORE [RFC8613][I-D.ietf-core-oscore-groupcomm]. This allows the proxy to access the option value and retrieve the timeout value T', as well as to remove the option altogether before forwarding the group request to the servers.

The security association between the client and the proxy MUST provide message integrity, so that further possible intermediaries as well as on-path active adversaries are not able to remove the option or alter its content, before the group request reaches the proxy. Removing the option would otherwise result in the proxy not forwarding the group request to the servers. Instead, altering the option content would result in the proxy accepting and forwarding back responses for an amount of time different than the one actually indicated by the client.

The security association between the client and the proxy SHOULD also provide message confidentiality. Otherwise, further intermediares as well as on-path passive adversaries would be able to simply access the option content, and thus learn for how long the client is willing to receive responses from the servers in the group via the proxy. This may in turn be used to perform a more efficient, selective suppression of responses from the servers.

When the client (further) protects the unicast request sent to the proxy with OSCORE (see Appendix A) and/or with DTLS, both message integrity and message confidentiality are achieved in the leg between the client and the proxy.

6.3. Response-Forwarding Option

The Response-Forwarding Option is of class E for OSCORE [RFC8613][I-D.ietf-core-oscore-groupcomm], and thus can be protected end-to-end between the client and the proxy, which includes the option into a server response before forwarding it back to the client.

Since the security association between the client and the proxy provides message integrity, any further intermediaries or on-path active adversaries are not able to undetectably remove the Response-Forwarding Option from a forwarded server response. This ensures that the client can correctly distinguish the different responses and identify their corresponding originator server.

7. IANA Considerations

This document has the following actions for IANA.

7.1. CoAP Option Numbers Registry

IANA is asked to enter the following option numbers to the "CoAP Option Numbers" registry defined in [RFC7252] within the "CoRE Parameters" registry.

Number	Name	Reference
TBD1	Multicast-Signaling	[[this document]]
TBD2	Response-Forwarding	[[this document]]

8. References

8.1. Normative References

[I-D.ietf-core-groupcomm-bis]
Dijk, E., Wang, C., and M. Tiloca, "Group Communication for the Constrained Application Protocol (CoAP)", draft-ietf-core-groupcomm-bis-00 (work in progress), March 2020.

[I-D.ietf-core-oscore-groupcomm]
Tiloca, M., Selander, G., Palombini, F., and J. Park, "Group OSCORE - Secure Group Communication for CoAP", draft-ietf-core-oscore-groupcomm-09 (work in progress), June 2020.

[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>.

[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/info/rfc3986>.

[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/info/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/info/rfc7641>.

[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>.

[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/info/rfc8613>.

8.2. Informative References

[I-D.bormann-coap-misc]
Bormann, C. and K. Hartke, "Miscellaneous additions to CoAP", draft-bormann-coap-misc-27 (work in progress), November 2014.

[I-D.ietf-ace-key-groupcomm-oscore]
Tiloca, M., Park, J., and F. Palombini, "Key Management for OSCORE Groups in ACE", draft-ietf-ace-key-groupcomm-oscore-07 (work in progress), June 2020.

[I-D.ietf-tls-dtls13]
Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", draft-ietf-tls-dtls13-38 (work in progress), May 2020.

[I-D.tiloca-core-oscore-discovery]
Tiloca, M., Amsuess, C., and P. Stok, "Discovery of OSCORE Groups with the CoRE Resource Directory", draft-tiloca-core-oscore-discovery-05 (work in progress), March 2020.

[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/info/rfc6347>.

Appendix A. Using OSCORE Between Client and Proxy

This section describes how OSCORE is used to protect messages exchanged by a client and a proxy, using their pairwise OSCORE Security Context.

This is especially convenient for the communication scenario addressed in this document, when the client already supports and uses Group OSCORE [I-D.ietf-core-oscore-groupcomm], to protect messages end-to-end with the servers.

A.1. Protecting the Request

Before sending the CoAP request to the proxy, the client protects it using the pairwise OSCORE Security Context it has with the proxy.

The client processes the CoAP request as defined in [RFC8613], with the following differences.

- The Proxy-Uri option, if present, is not decomposed and recomposed as defined in Section 4.1.3.3 of [RFC8613].

- The following options, if present, are processed as Class E.
  - Proxy-Uri, Proxy-Scheme, Uri-Host and Uri-Port.
  - OSCORE, which is present if Group OSCORE is used between the client and the servers, to achieve end-to-end secure group communication.

Furthermore, the Multicast-Signaling Option is processed as Class E.

As in [RFC8613], the resulting message includes an outer OSCORE Option, which reflects the usage of the pairwise OSCORE Security Context between the client and the proxy.

A.2. Verifying the Request

The proxy verifies the CoAP request as defined in [RFC8613].

If secure group communication is also used between the client and the servers, the resulting request to be forwarded to the servers is protected with Group OSCORE [I-D.ietf-core-oscore-groupcomm], and it includes a different OSCORE Option, which reflects the usage of the group OSCORE Security Context between the client and the servers.

A.3. Protecting the Response

The proxy protects the CoAP response received from a server, using the pairwise OSCORE Security Context it has with the client.

The proxy processes the CoAP response as defined in [RFC8613], with the difference that the OSCORE Option, if present, is processed as Class E. This is the case if Group OSCORE is used between the client and the servers, to achieve end-to-end secure group communication.

As in [RFC8613], the resulting message to be forwarded back to the client includes a different OSCORE Option, which reflects the usage of the pairwise OSCORE Security Context between the client and the proxy.

A.4. Verifying the Response

The client verifies the CoAP response received from the proxy as defined in [RFC8613].

If secure group communication is also used between the client and the servers, the resulting response is protected with Group OSCORE [I-D.ietf-core-oscore-groupcomm]. In particular, it includes a

different OSCORE Option, which reflects the usage of the group OSCORE Security Context between the client and the servers.

Acknowledgments

The authors sincerely thank Christian Amsuess, Jim Schaad and Goeran Selander for their comments and feedback.

The work on this document has been partly supported by VINNOVA and the Celtic-Next project CRITISEC.

Authors' Addresses

Marco Tiloca
RISE AB
Isafjordsgatan 22
Kista SE-16440 Stockholm
Sweden

Email: marco.tiloca@ri.se

Esko Dijk
IoTconsultancy.nl
_____\
Utrecht
The Netherlands

Email: esko.dijk@iotconsultancy.nl