Proxy Operations for CoAP Group Communication

draft-tiloca-core-groupcomm-proxy-00

Marco Tiloca, RISE
Esko Dijk, IoTconsultancy.nl

IETF CoRE WG virtual interim, April 8th, 2020

Motivation

- CoAP supports group communication over IP multicast
  - *draft-ietf-core-groupcomm-bis*

- The use of proxies introduces a number of issues
  - Clients to be whitelisted and authenticated on the proxy
  - The client may receive multiple responses to a single *unicast* request
  - The client may not be able to distinguish responses and origin servers
  - The proxy does not know when to stop handling responses

- Possible approaches for proxy to handle the responses
  - Individually forwarded back to the client
  - Forwarded back to the client as a single aggregated response

Contribution

- Description of proxy operations for CoAP group communication
  - Addressed all issues in *draft-ietf-core-groupcomm-bis*

- Considered approach to handle responses:
  - Individually forwarded back to the client

- Assumptions
  - The proxy is explicitly configured to support group communication
  - Clients are whitelisted on the proxy, and identified by the proxy
  - Group OSCORE is used for secure group communication (end-to-end, client to server).

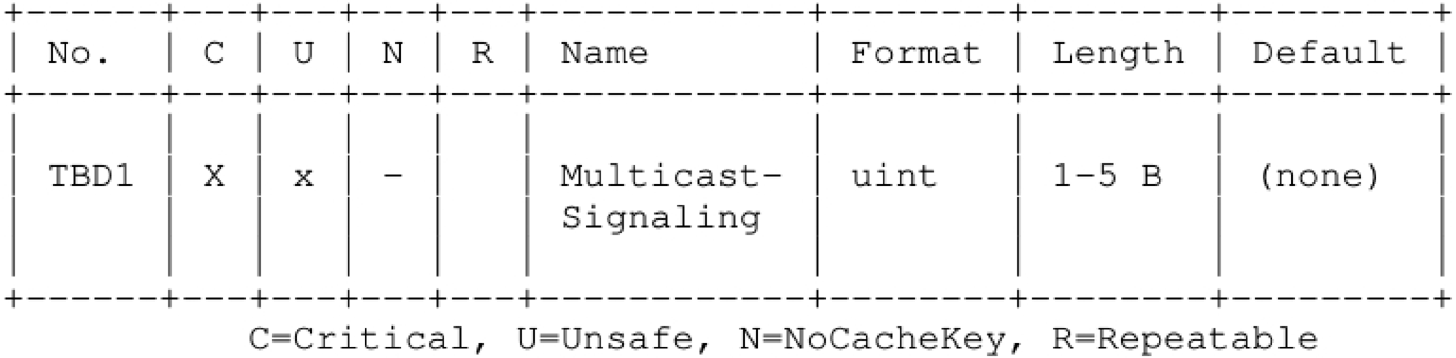
Rationale

- Signaling protocol with two new CoAP options
  - Along the lines of Thomas' comments for *draft-dijk-core-groupcomm-bis*

- In the request addressed to the proxy, the client indicates:
  - To be interested in and capable of handling multiple responses
  - For how long the proxy should collect and forward back responses

- In a response to a group request, the server indicates its IP address
  - The client can distinguish the responses and the different servers
  - The client becomes able to (directly, or via proxy) contact the server individually via unicast

Multicast-Signaling option



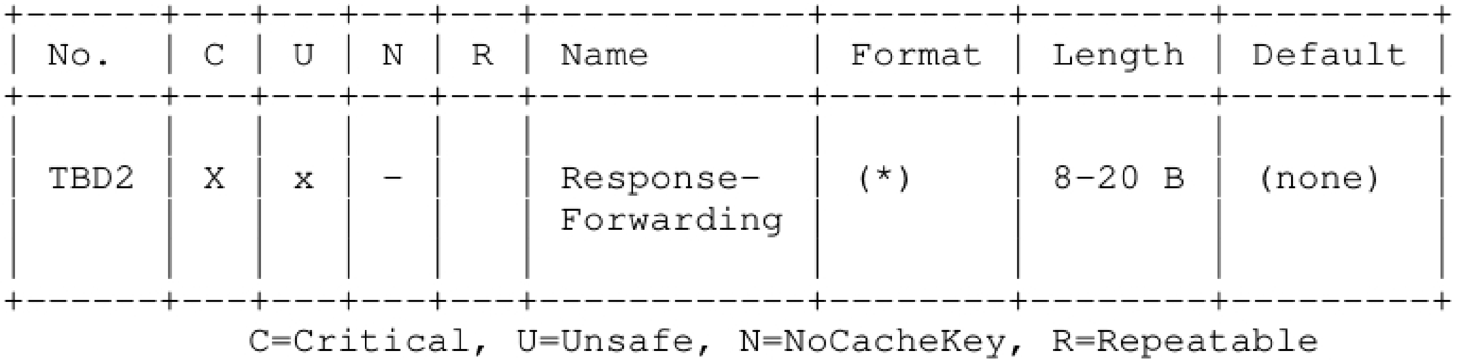
No.	C	U	N	R	Name	Format	Length	Default
TBD1	X	x	-		Multicast- Signaling	uint	1-5 B	(none)

C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable

- Used only in requests
  - Presence: explicit claim of support and interest from the client
  - Value: indication to the proxy on how long to handle unicast responses

- Class I for OSCORE
  - Allows the proxy to see it but not to remove it

Response-Forwarding option



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

C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable

- Used only in responses
  - Presence: allows the client to distinguish responses and originator servers
  - Value: IP address of the server, as a tagged CBOR byte string

- Class E for OSCORE

Workflow: C -> P

- C prepares a request addressed to P
  - The group URI is included in the Proxi-Uri option or the URI-* options

- C chooses T seconds, as token retention time
  - T < Tr , with Tr = token reuse time
  - T considers processing at the proxy and involved RTTs

- C includes the Multicast-Signaling option, with value T′ < T

- C sends the request to P via unicast
  - C retains the token beyond the reception of a first matching response

Workflow: P -> S

- P identifies C and verifies it is whitelisted

- P verifies the presence of the Multicast-Signaling option
  - P extracts the timeout value T′

- P forwards the request to the group of servers, over IP multicast

- P will handle responses for the following T′ seconds
  - Observe notifications are an exception – they are handled until the Observe client state is cleared.

Workflow: S -> P

- S knows there’s a client behind the proxy, by detecting the Multicast-Signaling Option.

- S includes the Response-Forwarding option in the response
  - The option value is the IP address of the server, as a tagged CBOR byte string

Workflow: P -> C

- P forwards responses back to C, individually as they come

- P frees-up its token towards the group of servers after T′ seconds
  - Late responses > T’ will not match and not be forwarded to C
  - Observe notifications are the exception

- C retrieves the Response-Forwarding option
  - C distinguishes different responses from different origin servers
  - C is able to later contact a server individually, either directly or indirectly

- C frees-up its token towards the proxy after T seconds
  - Again, Observe notifications are the exception

Open points

- Mostly from Christian's comments – Thanks!
- Alternative design proposed – to consider
  - Proxy removes the Multicast-Signaling Option from request;
  - Proxy adds the Response-Forwarding Options and its IP address info to responses
  - No end-to-end security for the information in both Options
- If the proxy authenticates the client with a <C,P> OSCORE context …
  - We have a use case for "nested OSCORE"
  - Should we define it? Would this same document be appropriate?
- This document is general enough, as about "proxy operations"
  - Should it define also response aggregation as alternative approach?

Summary

- Defined proxy operations for CoAP group communication
  - Embedded signaling protocol, using two new CoAP options
  - The proxy separately forwards back individual responses to the client for a defined time period T'
  - The client can distinguish the origin servers and corresponding responses

- Main next step: address Christian's comments and open points

- Need for comments and feedback

Thank you!

Comments/questions?

https://gitlab.com/crimson84/draft-tiloca-core-groupcomm-proxy