Proxy Operations for CoAP Group Communication

draft-ietf-core-groupcomm-proxy-03

Marco Tiloca, RISE
Esko Dijk, IoTconsultancy.nl

IETF 121 Meeting – Dublin – November 5th, 2024

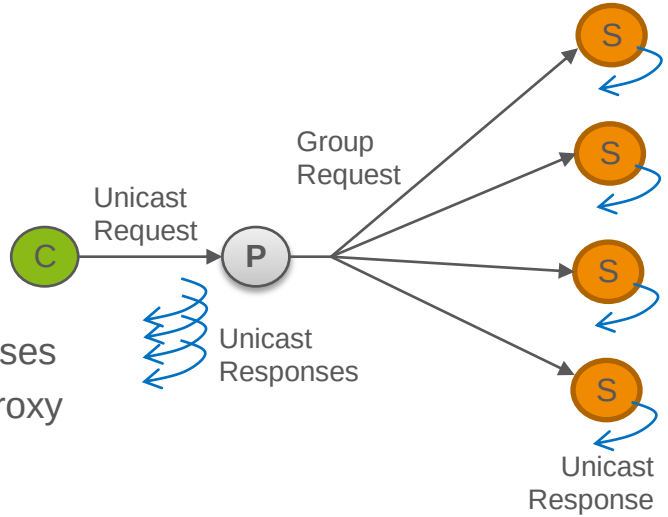
Recap

- **Scope: definition of proxy operations for CoAP group communication**
  - Signaling protocol between client and proxy, with two new CoAP options
  - Individual responses from the CoAP servers are relayed back to the client
  - Support for forward-proxies, reverse-proxies, chain of proxies, and HTTP-to-CoAP proxies
  - Updated CoAP freshness model and validation model for cached responses in groups

- **The proxy is explicitly configured to support group communication**
  - Clients are allowed-listed on the proxy, and identified by the proxy

- **Address issues with proxies discussed in *draft-ietf-core-groupcomm-bis***
  - Now compiled in the dedicated Appendices E and F of that document

Gist of the protocol



- **In the unicast request addressed to the proxy, the client indicates:**
  - To be interested and capable of handling multiple responses
  - For how long the proxy should collect and forward back responses
  - In the new CoAP option **Multicast-Timeout**, removed by the proxy
- **In each response to the group request, the proxy includes addressing information pertaining to the server**
  - In the new CoAP option **Reply-From** (was Response-Forwarding, then was Reply-To)
  - The client can distinguish responses and different servers
  - The client can later contact an individual server (directly if possible, or again via the proxy)
- **Group OSCORE can be used for end-to-end security between client and servers**
- **Security is used between Client and Proxy, especially to identify the Client**
  - (D)TLS or OSCORE (see *draft-ietf-core-oscore-capable-proxies*)

Updates since v -01

- **Several minor clarifications and editorial fixes**

- **Reply-From Option**
  - Changed name from “Reply-To”, as suggested at IETF 119 (was issue #2)
  - Security considerations: discussed what a lack of confidentiality would mean
    - Exposed addressing information of origin servers
    - Easier for an active adversary to selectively suppress follow-up, individual requests

- **Multicast-Timeout Option**
  - Consistent with its “uint” format, it can take value 0
  - Clarified that the empty option on the wire is the result of the recommended encoding

Updates since v -01

- **Made RFC 7967 (No-Response Option) a normative reference**
  - Expected use combined with Multicast-Timeout=0
  - Change consistent with what suggested by Christian for *draft-ietf-core-groupcomm-bis*

- **Handling of multiple responses at the client (Section 5.5.1)**
  - Aligned with the text suggested by Christian for *draft-ietf-core-groupcomm-bis*
  - Generalized as to “where” exactly the client handles multiple responses
    - Presumably at the application, as expected to have more context to better decide

- **Revised examples with reverse-proxy (Appendix A)**
  - Fixed and consistent destination addresses
  - Fixed and consistent values of the Uri-Host Option

Updates since v -01

- **Error handling at a proxy processing a request (Section 5.2.1)**
  - The proxy identifies the client, per the security association protecting the request to forward
  - Specified missing case: the proxy cannot verify that the client is allowed-listed
  - Outcome: the request is not forwarded, and the proxy replies with a 4.01 error response

- **Error handling in proxy chains (Section 8.1)**
  - An inadequate value in the Multicast-Timeout Option yields a 5.05 error response
  - Specified missing case: the origin client receiving the 5.05 response can send a new request to the first proxy, with a greater value in the Multicast-Timeout Option
  - Clarified existing case: a proxy receiving the 5.05 response can send a new request to the next proxy, or send a 5.05 response to the (previous hop towards the) origin client

Updates since v -01

- **HTTP Reply-From Header Field (Section 9.2)**
  - Consistent with the Reply-From Option, it can include one or two pieces of information
  - Suggestion at IETF 119: use HTTP Structured Field Values (RFC 9651) – Thanks Christian!
  - Now done (was issue #1). In short:
    - List Header Field of 1 or 2 “Byte Sequence Items”, consistent with the value of the CoAP Reply-From Option to convert from/to

- **HTTP Group-ETag Header Field (Section 9.3)**
  - Like the Group-ETag Option, it enables validation of a full set of cached responses at the proxy
  - In HTTP requests: List Header Field of N “Byte Sequence Items”
    - As many as the CoAP Group-ETag Options in a CoAP request
  - In HTTP responses: List Header Field of 1 “Byte Sequence Item”
  - An HTTP-to-CoAP proxy locally runs the response validation like the CoAP-to-CoAP proxy
    - The CoAP response 2.03 (Valid) corresponds to the HTTP response 304 (Not modified)

Next steps

- **Latest issue #3 from Christian**
  - The Reply-From Option might be renamed again and/or become part of a cluster of Options
  - Relatable, not-yet-defined options can be useful:
    - For a server, to indicate its canonical address or a wish for transport switching
    - For a client, to indicate a specific hostname to use if reversing the client and server roles

- **Main points to address in next versions**
  - Address comment from IANA: specify preferred value range for the new CoAP Option Numbers
  - Cancellation of ongoing response forwarding
  - Response forwarding to an HTTP client via streamed delivery, using Transfer-Coding:chunked
  - Response revalidation between proxy and servers, when using Group OSCORE
    - Placeholder note in Sections 7.2.1 and 7.2.2: introduce an outer ETag Option
    - Perhaps it can be defined in *draft-amsuess-core-cachable-oscore* ?
  - Revisit and extend the RFC 8075 security considerations on HTTP-to-CoAP proxies
  - Add examples with an HTTP-to-CoAP proxy

- **Comments and reviews are welcome!**

Thank you!

Comments/questions?

https://github.com/core-wg/groupcomm-proxy