Group Communication for the Constrained Application Protocol (CoAP)

Work in progress towards
draft-ietf-core-groupcomm-bis-04

Esko Dijk, IoTconsultancy.nl
Chonggang Wang, InterDigital
Marco Tiloca, RISE

CoRE WG Interim Meeting, June 9th, 2021

Recap and Goal

- Intended normative successor of experimental RFC 7390 (if approved)
  - As a Standards Track document
  - Obsoletes RFC 7390; Updates RFC 7252 and RFC 7641
- Be standard reference for implementations that are now based on RFC 7390, e.g.:
  - “Eclipse Californium 2.0.x” (Eclipse Foundation)
  - “Implementation of CoAP Server & Client in Go” (OCF)
- What’s in scope?
  - CoAP group communication over UDP/IP, including latest developments
  - (Observe/Blockwise/Security ...)
  - Caching and re-validation of responses
  - Unsecured CoAP or group-OSCORE-secured communication
  - Principles for secure group configuration
  - Use cases (appendix)

Content moved out

- General caching model at the proxy
  - Definition/Creation/Maintenance of the different types of cache entries
  - Case with end-to-end security based on Cacheable OSCORE
    - https://datatracker.ietf.org/doc/draft-amsuess-core-cachable-oscore/

- Response re-validation between Client and Proxy
  - Based on a new Group-ETag option

- Both moved to the Editor's copy of *draft-tiloca-core-groupcomm-proxy*
  - https://gitlab.com/crimson84/draft-tiloca-core-groupcomm-proxy/-/tree/v-04

Review and comments from John

- Under processing
  - https://mailarchive.ietf.org/arch/msg/core/xy3ImeWkbqziBhqs4NCGwNP6R7U/
  - https://mailarchive.ietf.org/arch/msg/core/Z9wxTL3-hAEDT_P8JOQztTfxlVU/

- New tasks for authors to work on next (feedback is welcome!)
  - Issue #20 – New Section on additions/replacements to updated/obsoleted documents
  - Issue #21 – Not excluding transports other than UDP/IP multicast
    (initial analysis: should be doable, no roadblock)
  - Issue #22 – Add considerations on amplification attacks

- Open points - requiring feedback to make progress
  - Issue #22 – Aspect of source IP spoofing scenarios; what are relevant attack scenarios.

Github issues

- #15 “Move intro text of Section 5 into an own Section 5.1?”
  - Editorial, already closed
  - Added introductory text, split the following long text into subsections

- #12 – Referring examples in an obsoleted RFC, i.e., 7390
  - Done, referring to RFC 7390 examples in the past tense. (“RFC7390] showed ... [RFC7390] defined ...”)

- #14 – Interaction between Observe and No-Response Options
  - OLD: “A server that adds a client to the list (as a new entry) of observers for a resource due to an Observe request MUST respond to this request and not suppress it.”
  - NEW: “If a server adds a client (as a new entry) to the list of observers for a resource due to an Observe request, the server SHOULD respond to this request and SHOULD NOT suppress the response. An exception to the above is the overriding of response suppression according to a CoAP No-Response Option {{RFC7967}} ...” Ok with this?
  - General considerations on Observe with No-Response can be better handled in the CorrClarr document
    - https://github.com/core-wg/corrclar

Github issues

- #16 – Size-based packet filtering in 6LoWPAN security considerations
  - Done, added size-based filtering as RECOMMENDATION in security considerations for 6LoWPAN/MPL.

- #17 – Valid cases of forward/reverse proxy with e2e security and with two-leg security
  - Currently, nothing is said about combination “secure group communication” with “proxy in the path”
  - Proposal is to include a new Section 5.3 to address what are the valid combinations.
  - We don’t expect much text, can be appropriate section references to draft-groupcomm-proxy.
  - For discussion: which ‘combinations’ make sense?

- #19 – Consider how to handle the Q-Block options?
  - From https://datatracker.ietf.org/doc/html/draft-ietf-core-new-block-14#section-4.8 :
    - "*Servers MUST ignore multicast requests that contain the Q-Block2 Option.*
      As a reminder, Block2 Option can be used as stated in Section 2.8 of [RFC7959]."
  - Done, above is now written with *draft-ietf-core-new-block* as informative reference

Github issues

- Part of #11 – Clarify rules of caching at origin clients
  - https://github.com/core-wg/groupcomm-bis/issues/11#issuecomment-794184440
  - https://github.com/core-wg/groupcomm-bis/issues/11#issuecomment-795069740
  - https://github.com/core-wg/groupcomm-bis/issues/11#issuecomment-795290596
  - New members can join the group at any time; a local cache entry for responses to a group request may not cover all the responses sent since the latest cache refresh. This needs rules:
    - In this document, for origin clients. Simple solution proposed, see the Editor's copy:
      Group request always needs to be sent out, unless client has fresh responses cached for all group servers. Can only do this when group is known.
      - https://core-wg.github.io/groupcomm-bis/v-04/draft-ietf-core-groupcomm-bis.html
    - In *groupcomm-proxy* , for the "aggregated" cache entry at the proxy (already in its Editor's copy).
- #11 – Placement of new caching features
  - Caching at origin clients → see previous point
  - Caching at proxies → moved to *groupcomm-proxy*
  - Revalidation between Client and Proxy → moved to *groupcomm-proxy*
  - Revalidation between origin client and origin servers → Current proposal is to use ETag Option. For discussion.

Github issues

#11 – Placement of new caching features

- Revalidation between origin Client and origin Servers
- Section 8.2.1 of RFC 7252 left this for further study

› Which document?

- This document seems preferable to a different (new) document
- *groupcomm-proxy* can reuse what defined here, for revalidation between Proxy and origin Servers

› Exact approach? Different open proposals but good to converge to one only

1. A new Multi-ETag option proposed in *groupcomm-bis-03*
2. Original ETag option, see separate open issue and draft proposed text in Editor's copy
   - https://github.com/core-wg/groupcomm-bis/issues/18
3. New CoAP option suggested by Christian, as a single "aggregated" tag understandable by all servers; a single instance would be included in the request; no need for addressing information
4. New CoAP option suggested by Christian, specifying a pure time-based indication

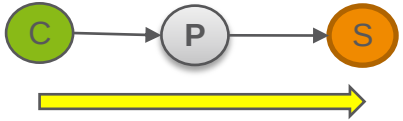
Recap of the Multi-ETag Option

Between Client and Servers

No.	C	U	N	R	Name	Format	Length	Default
TBD1				x	Multi-ETag	(*)	any	(none)

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

- New Multi-Etag option
  - Only for group requests
  - One instance per server in the group to revalidate against



- Option value: CBOR sequence of 1+M elements
  - First element: addressing information of the server, encoded as in *groupcomm-proxy*
  - The following M elements are entity-tag values, as CBOR byte strings

- A server processes only the Multi-Etag option pertaining to itself, unlike ETag
  - What follows uses ETag, as in RFC 7252

Alternatives for response validation

- Multi-ETag (previous slide)
- ETag option
  - Simple: ETag Option used as normal
  - Server SHOULD (but not required) embed a compact, server-specific ID in ETag value.
  - Client needs to handle potential cases of ‘value conflict’ in ETags from different servers.
  - «Legacy» servers not aware of this ETag feature will just ignore the option (=ok)
  - Proposed text in Editor’s copy. Ok to define this? Attempts to cover all use cases with minimal code impact.
    - https://core-wg.github.io/groupcomm-bis/v-04/draft-ietf-core-groupcomm-bis.html
- New CoAP option with single “aggregated” tag
  - Single tag is understandable by all servers
  - Do we want to explore this further?
- New CoAP option with a pure time-based indication
  - e.g. client includes a time period value that indicates last time this same group request was sent.
  - if that last response (based on time period) is still valid, the server responds 2.03 Valid.
  - Do we want to explore this further?

Next steps

- Address comments from review John – Github issues created
- Work on Github open points/issues
  - Decide on the validation method between client and origin servers
- Submit v-04 before the IETF 111 cut-off

Thank you!

Comments/questions?

https://github.com/core-wg/groupcomm-bis/tree/v-04