Group Communication for the Constrained Application Protocol (CoAP)

Towards draft-ietf-core-groupcomm-bis-06

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

CoRE WG interim meeting, January 19, 2022
Following IETF 112 ...

› Jaime requested to add examples, especially on:
  – Encoding of application group names, e.g., in CoAP requests (topic of issue #28)
  – Discovery of CoAP groups and application groups from CoAP servers (topic of issue #29)

› Opportunity to also improve the text addressing issues #28 and #29

› Opened new PR #32 addressing the points above
  – Plus examples of message exchange
  – Plus more clarifications and editorial improvements
  – https://github.com/core-wg/groupcomm-bis/pull/32
2.2.1 - Name encoding of app groups

- Revised methods, each with an example

- Through the CoAP request, in the group URI
  - in the path component (Recommended)
  - in the query component (2 possible ways)
  - in the authority component as a whole
  - in the host subcomponent
  - in the port subcomponent

- Through the CoAP request, but not in the group URI
  - In the Uri-Host Option, added before sending
  - New custom, application-specific, CoAP option

- Not on the wire ➔ implicit understanding from application/network context
2.2.1 - Name encoding of app groups

› Revised methods, each with an example

› Through the CoAP request, in the group URI
  – in the path component (Recommended)
  – in the query component (2 possible ways)
  – in the authority component as a whole
  – in the host subcomponent
  – in the port subcomponent

› Through the CoAP request, but not in the group URI
  – In the Uri-Host Option, added before sending
  – New custom, application-specific, CoAP option

› Not on the wire → implicit understanding from application/network context
2.2.1 - Name encoding of app groups

› Revised methods, each with an example

› Through the CoAP request, in the group URI
  – in the path component (Recommended)
  – in the query component (2 possible ways)
  – in the authority component as a whole
  – in the host subcomponent
  – in the port subcomponent

› Through the CoAP request, but not in the group URI
  – In the Uri-Host Option, added before sending
  – New custom, application-specific, CoAP option

› Not on the wire → implicit understanding from application/network context
2.2.3 – Group discovery

› Revised methods, each with an example

› Given a CoAP group, discover …
   – The associated application groups
   – The servers in it and each group’s resources

› Given an application group, discover …
   – The associated CoAP group
   – The servers in it and each group’s resources

› Discover …
   – Any application group (*)
     › The associated CoAP group
     › The servers in it and each group’s resources

(*) Possible to filter by group type
2.2.3 – Group discovery

› Revised methods, each with an example

› Given a **CoAP group**, discover …
   – The associated **application groups**
   – The **servers** in it and each group’s resources

› Given an application group, discover …
   – The associated **CoAP group**
   – The servers in it and each group’s resources

› Discover …
   – Any application group (*)
     › The associated CoAP group
     › The servers in it and each group’s resources

---

// Request to all members of the CoAP group
Req: GET coap://grp.example.org:5685/.well-known/core?rt=g.*

// Response from server S1, as member of:
//   - The CoAP group "grp.example.org:5685"
//   - The application group "gp1"
Res: 2.05 Content
Content-Format: 40
Payload:
</gp/gp1>;rt=g.light

// Response from server S2, as member of:
//   - The CoAP group "grp.example.org:5685"
//   - The application groups "gp1" and "gp2"
Res: 2.05 Content
Content-Format: 40
Payload:
</gp/gp1>;rt=g.light,
</gp/gp2>;rt=g.temp

Figure 11: Discovery of application groups associated to a CoAP group

(*) Possible to filter by group type
2.2.3 – Group discovery

› Revised methods, each with an example

› Given a CoAP group, discover …
  – The associated application groups
  – The servers in it and each group’s resources

› Given an application group, discover …
  – The associated CoAP group
  – The servers in it and each group’s resources

› Discover …
  – Any application group (*)
    › The associated CoAP group
    › The servers in it and each group’s resources

(*) Possible to filter by group type

2.2.3 – Group discovery

// Request to realm-local members of the application group "gp1"
Req: GET coap://[ff03::fd]/.well-known/core?href=/gp/gp1

// CoAP response from server 51, as member of:
// - The CoAP group "grp.example.org:5685"
// - The application group "gp1"
Res: 2.05 Content
Content-Format: 40
Payload:
<coap://grp.example.org:5685/gp/gp1>;rt=g.light

// CoAP response from server 52, as member of:
// - The CoAP group "grp.example.org:5685"
// - The application groups "gp1"
Res: 2.05 Content
Content-Format: 40
Payload:
<coap://grp.example.org:5685/gp/gp1>;rt=g.light

Figure 12: Discovery of members of an application group, together with the associated CoAP group
Appendix B – Message exchange

› New Appendix B
  - Three examples

› Example 1 (plain)
  - Request over multicast
  - Responses follow

› Example 2 (observe)
  - Observation request over multicast
  - Two rounds of notifications

› Example 3 (blockwise)
  - Request with Block2 over multicast
  - Following exchanges over unicast

Figure 16: Example of Non-confirmable group request, followed by Non-confirmableResponses
Next steps

› If no objection, merge PR #32 soon and submit v -06

› If all is well, we can close issues #28 and #29

› Working Group Last Call (for -06)
Thank you!
Comments/questions?

https://github.com/core-wg/groupcomm-bis/
Goal

› Normative successor of experimental RFC 7390
  – Obsoletes RFC 7390, Updates RFC 7252 / 7641

› New standard reference for implementations now based on RFC 7390

› Scope
  – CoAP group communication, including latest features:
    Observe/Blockwise/Security …
  – Unsecured & group-OSCORE-secured
  – Definition of group types & Secure group configuration
Motivation (backup slide)

› RFC 7390 was published in 2014
  – CoAP functionalities available by then were covered
  – No group security solution was available to indicate
  – It is an Experimental document (started as Informational)

› What has changed?
  – More CoAP functionalities have been developed (Block-Wise, Observe)
  – RESTful interface for membership configuration is not really used
  – Group OSCORE provides group end-to-end security for CoAP

› Practical considerations
  – Group OSCORE clearly builds on RFC 7390 normatively
  – However, it can refer RFC 7390 only informationally