A pubsub architecture for CoAP
draft-ietf-core-coap-pubsub

Jaime Jiménez

IETF122, 2025-03-18

Updates since IETF 121

Version -15 to -16
- Various updates throughout the document based on AD review.
- IANA clarifications

Version -16 to -17
- Addressing reviews by shepherd
- Fixing IANA formatting

Updates since IETF 121

Version -17 to -18

- Addressed issues
  - #64 Single-quotes / double-quotes / no-quotes around the topic property names
  - #65 Section 2.3.3 "Topic Discovery" uses well-known/core, but 2.3.2 says this is not mandatory to support
  - #66 Use of relative URIs in the "topic-data" URI field
  - #67 CBOR diag notation is sometimes using final comma at last item in map
- rt, ct, obs attribute elision (more about this in the discussion at the end)
- Editorial changes

Updates since IETF 121

Coming version -18 to 19

- IANA early review
- Addressed issues #68, #69
  - 68 Explain better optimization of attributes
  - 69 Fix terminology for topic-data resource by removing double quotes / Minor editorial fixes
- Addressed Marco's review

Hackaton Updates

- Small updates on the aiocoap-pubsub-broker implementation.
- Cleanup of the examples.
- Script to show a full demo for topic creation, subscription, etc.

Recap: Architecture Overview

```
  CoAP                      CoAP                 CoAP
  clients                  server                clients
.-----------.          .----------.  observe  .------------.
|           | publish  |          |<----------+            |
| publisher +--------->+          +---------->| subscriber |
|           |          |          +---------->|            |
'-----------'          |          |           '------------'
      ...              |  broker  |                ...
      ...              |          |                ...
.-----------.          |          |  observe  .------------.
|           | publish  |          |<----------+            |
| publisher +--------->|          +---------->| subscriber |
|           |          |          +---------->|            |
'-----------'          '----------'           '------------'
```

Recap: Topic Lifecycle

```
                HALF                          FULLY
               CREATED       Delete          CREATED
                ____        topic-data        ____     Publish
------------>  |    |  <-------------------  |    |  ------------.
   Create      |    |                        |    |               |
               |____|  ------------------->  |____|  <-----------'
                      \      Publish     /            Subscribe
               |   ^   \       ___       /   |   ^
         Read/ |   |    '-->  |   |  <--'    |   | Read/
        Update |   |  Delete  |___|  Delete  |   | Update
         Topic  '-'   Topic          Topic    '-'  Topic
                             DELETED
```

Discussion items

- Behaviour of `initialize` set to "true".
    - When used, the topic is in fully-created state and clients can subscribe to the topic before data has been published. When the first publication arrives, it returns a 2.04 cause the topic is already fully-created, even if it was with a zero-length (empty) payload.
- Topic-data deletion.
    - When topic-data resource is deleted, the broker does not delete the "topic-data" topic property, it is kept as HALF-CREATED state and only publishing is allowed, not to subscribe or to read. Same thing applies for a topic that was created using "initialize".
- Rate limiting.
    - Currently the rate limiting section defines a behaviour for publishers. Clients subscribed to topic-data from rate-limited publishers do not see any specific information about that.

Discussion items

Response with rt attribute	Response without rt attribute
Payload: `</ps/2e3570>;rt="core.ps.conf"`	Payload: `</ps/2e3570>`

- Eliding attributes from query results.
  - When doing a query (`rt=<something>`) on a resource, i.e. one defined in scope of the pubsub draft, the draft itself can define the rules for that. The resource can be mandated (MUST) to support this query so we know that any and all results will have the correct rt type. There's no need to include that rt type in each link result, thereby saving lots of bytes on the wire.
  - If the "rt" on a returned link is 100% known to the client parsing the links (whether rt is explicitly included or not), and there is no "ct" (content-format) attribute given, then the default "ct" as defined by this document is applied. Because it follows from "rt" there's no need to add "ct" attributes where it follows the default. This also saves bytes on the wire.

Conclusions

- Got several reviews and addressed the comments.
- More reviews are welcomed.
    - CA made a review yesterday and I will check.
    - Preferably in the form of PR rather than only email.