CoAP in space

draft-gomez-tiptop-coap-00

Intended Status: Informational

Carles Gomez

Universitat Politècnica de Catalunya

Sergio Aguilar

Sateliot

MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES
Cofinanciado por la Unión Europea
AGENCIA ESTATAL DE INVESTIGACIÓN

This work was funded in part by the Spanish MCIU/AEI/10.13039/501100011033/FEDER/UE through project PID2023-146378NB-I00

IETF 124 Montreal, TIPTOP, November 2025

1. Introduction (I/III)

- Characteristics of deep space communication and networking [draft-many-tiptop-usecase]:
  - Long delays (e.g., minutes or hours)
  - Intermittent communication opportunities
  - Lower bandwidth
  - Asymmetrical bandwidth
  - Limited computing resources
  - Limited energy resources

1. Introduction (II/III)

- Characteristics of Internet of Things (IoT) communication and networking:
  - Long delays (e.g., minutes or hours)
  - Intermittent communication opportunities
  - Lower bandwidth
  - Asymmetrical bandwidth
  - Limited computing resources
  - Limited energy resources

1. Introduction (III/III)

- Constrained Application Protocol (CoAP):
  - Designed for constrained-node network environments (IoT)
  - Lightweight operation (fixed header of only 4 bytes!), asynch. message exchanges, flexibility, security
  - Based on the REST architecture of the web
    - Requests (clients) and responses (servers)
    - Easy mapping with HTTP
- draft-many-tiptop-ip-architecture-02:
  - Application layer: HTTP and CoAP
- draft-gomez-tiptop-coap-00:
  - Guidance on using CoAP in deep space environments

3.1. Overview and underlying transport

- Originally designed on top of UDP (RFC 7252)
  - Message sublayer:
    - Optional reliability
    - Simple congestion control
  - Message types:
    - Confirmable (CON)
      - » Needs an Acknowledgment (ACK)
      - » Stop & wait
      - » Timer-based retransmission with exponential back-off
    - Non-confirmable (NON)

```
+----------------------+
|      Application     |
+----------------------+
+----------------------+  \
|  Requests/Responses  |  |
|----------------------|  | CoAP
|       Messages       |  |
+----------------------+  /
+----------------------+
|          UDP         |
+----------------------+
```

- Subsequently, CoAP over reliable transports (RFC 8323)
  - TCP, TLS, WebSockets
    - Significant performance penalty in deep space (e.g., initial handshakes)
    - Reliability is not optional
    - No support for multicast
    - Greater header size

Deep space: CoAP over UDP

3.2. Main CoAP parameters and times relevant to deep space (I/II)

- NSTART
  - Max number of outstanding interactions
  - Default value: 1
- ACK_TIMEOUT (AT), ACK_RANDOM_FACTOR (AF)
  - Initial RTO, by default randomly chosen from [2 s, 3 s]
  - Needs adaptation for deep space
- MAX_RETRANSMIT
  - Default value: 4
  - Due to exponential back-off, lower than default may be suitable

3.2. Main CoAP parameters and times relevant to deep space (II/II)

- MAX_LATENCY
  - Max time since a datagram is sent until it is received
  - Defined as 100 s
- NON_LIFETIME
  - Max time since a NON message is sent until it is received
  - Default value: MAX_LATENCY (i.e., 100 s)
- EXCHANGE_LIFETIME
  - Max time since first transmission attempt of a CON until its ACK
  - Default value: 247 s

Need to be increased for deep space

4. Caching

- Responses may be cached to reduce response time and resource consumption on future, equivalent requests
  - Suitable for deep space scenarios

- Cached response can be reused if “fresh”
  - Origin server sets the Max-Age option
  - By default, Max-Age = 60 seconds
  - Maximum possible Max-Age = $2^{32} - 1$ seconds (~136 years)
  - Max-Age needs to be set based on the expected latency from origin server to caching CoAP endpoint

5. Proxying (I/II)

- CoAP-to-CoAP proxy
  - Including message aggregation (Section 8)

```
+-------+              +-------------+                  +-------+
| CoAP  |              |    CoAP     |                  | CoAP  |
+-------+              +------+------+                  +-------+
|  UDP  |              |  UDP | UDP  |                  |  UDP  |
+-------+              +------+------+                  +-------+
|  IP   |              |  IP  |  IP  |                  |  IP   |
+-------+              +------+------+                  +-------+
|  L2   |              |  L2  |  L2  |                  |  L2   |
+-------+              +------+------+                  +-------+
|  L1   |              |  L1  |  L1  |                  |  L1   |
+-------+      (*)     +------+------+      (*)         +-------+
   |                      ^      |                         ^
   >>>>>>>>>>>>>>>>>>>>>>>>^      >>>>>>>>>>>>>>>>>>>>>>>>>^

  CoAP                 CoAP-to-CoAP                     CoAP
 client                    proxy                   origin server
```

5. Proxying (II/II)

- HTTP-to-CoAP proxy

```
+-------+               +-------+-------+               +-------+
| HTTP  |               | HTTP  | CoAP  |               | CoAP  |
+-------+               +-------+-------+               +-------+
| QUIC  |               | QUIC  |  UDP  |               |  UDP  |
+-------+               +-------+-------+               +-------+
|  IP   |               |  IP   |  IP   |               |  IP   |
+-------+               +-------+-------+               +-------+
|  L2   |               |  L2   |  L2   |               |  L2   |
+-------+               +-------+-------+               +-------+
|  L1   |               |  L1   |  L1   |               |  L1   |
+-------+     (*)       +-------+-------+     (*)       +-------+
   |                       ^      |                        ^
   >>>>>>>>>>>>>>>>>>>>>>>>^      >>>>>>>>>>>>>>>>>>>>>>>>>^

  HTTP                    HTTP-to-CoAP                    CoAP
 client                   cross-proxy                 origin server
```

CoAP extensions

- 6. Observe (RFC 7641)
  - Allows a server to send notifications
    - Beneficial in deep space
  - Representation of the current state of a resource to observers
  - Observers need to initially register their interest

- 7. Block-wise transfers (RFC 7959, RFC 9177)
  - Allows to reliably carry large payloads
  - Based on NON messages
  - Efficient recovery of multiple missing blocks
  - Parameters need to be fine-tuned for deep space

9. CoAP group communication

- A client may send multicast CoAP request messages over UDP/IP multicast as default transport
- Each server in the destination group sends a response message back to the client
    - A response can be suppressed
- [I-D.ietf-core-groupcomm-bis]:
    - Minimum time between reuse of Token values for group requests, MIN_TOKEN_REUSE_TIME, to be greater than 500 seconds (default)
    - Needs to be increased for many deep space scenarios

10. Security

- Two main alternatives:
  - DTLS
    - Handshakes…
  - Object Security (OSCORE, RFC 8613):
    - End-to-end application-layer payload protection
      - » Even in the presence of proxies
    - Shared security context, may be based on pre-shared materials
    - Avoids initial handshake and related performance penalty, critical in deep space

Thanks!
Questions? Comments?

Carles Gomez

Universitat Politècnica de Catalunya

Sergio Aguilar

Sateliot