Constrained Application Protocol (CoAP) Performance Measurement Option

draft-fz-core-coap-pm-04

Hybrid, Mar 2023, IETF 116

Giuseppe Fioccola (Huawei)
Tianran Zhou (Huawei)
Mauro Cociglio (Telecom Italia)
Fabio Bulgarella (Telecom Italia)
Massimo Nilo (Telecom Italia)
Fabrizio Milan (Telecom Italia)

Motivation

A mechanism to measure the performance in CoAP can be useful to verify and meet the operational requirements.

- It is resource consuming to read IDs / sequence numbers and store timestamps for constrained nodes.

✓ Performance Measurement in constrained environment needs straightforward methodologies!

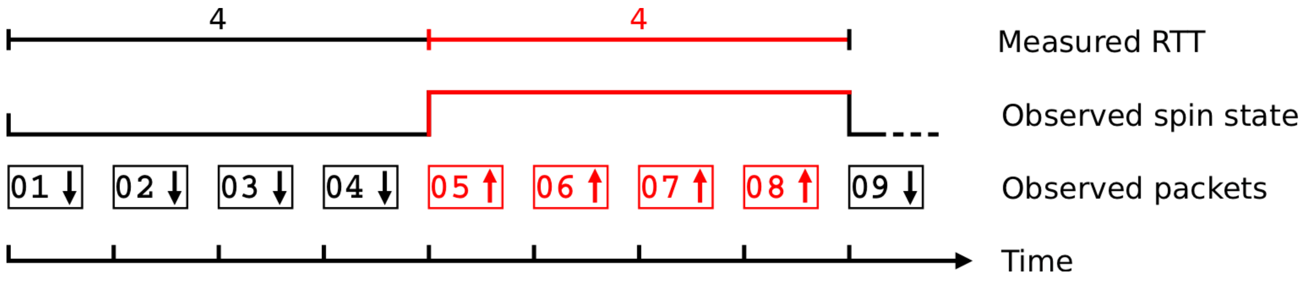
✓ It must be a simple mechanism for network diagnostic requiring just a minimal amount of collaboration from the endpoints.

Explicit Flow Measurement (EFM) techniques employ few marking bits, inside the header of each packet, for loss and delay measurement.

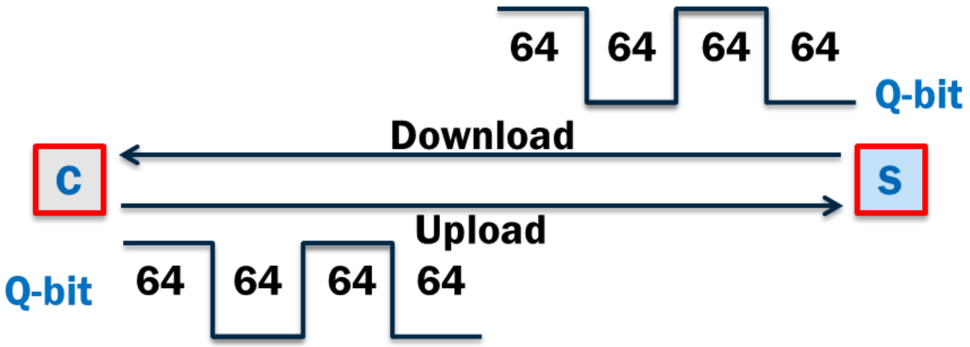
- These are described in **draft-ietf-ippm-explicit-flow-measurements (in Last Call)**

Spin Bit and sQuare Bit

- The **Spin bit** idea is to create a square wave signal on the data flow, using a bit, whose length is equal to RTT. It is optional in QUIC (RFC 9000 and RFC 9312)

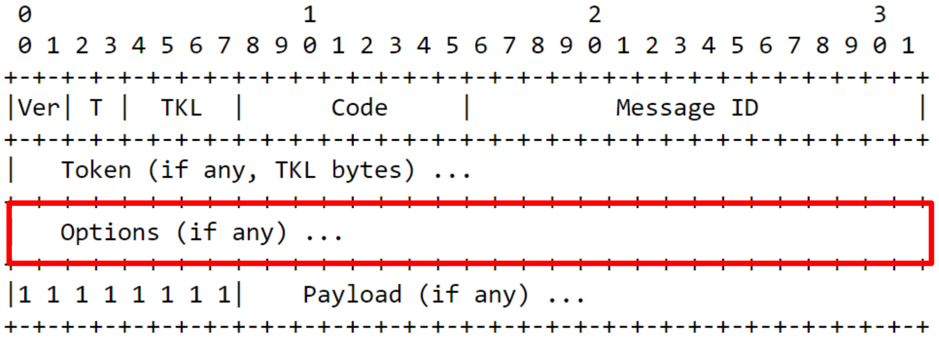


- The **sQuare bit** creates square waves of a known length as defined in the Alternate Marking (RFC 9341). This can be used for packet loss (and delay) measurements.



COAP PM Option

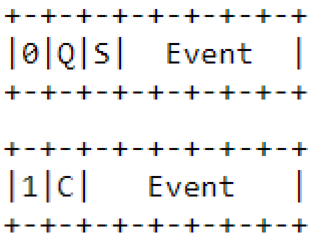
- A new option for CoAP carrying PM bits (Spin bit and sQuare Bit) can be introduced



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver| T |  TKL  |      Code     |          Message ID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Token (if any, TKL bytes) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 1 1 1 1 1|    Payload (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- The PM Option Value can be defined with 1 bit or 2 bits, which are defined as follows:
  - sQuare Bit (Q) for Packet Loss measurement in both directions.
  - Spin Bit (S) for RTT measurement.
  - Combined sQuare Bit (C) can reinforce Q with Delay information.

The Option value is a 1 byte unsigned integer, and two patterns are currently defined:



```
+-+-+-+-+-+-+-+-+
|0|Q|S|  Event  |
+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+
|1|C|   Event   |
+-+-+-+-+-+-+-+-+
```

The Event bits can be used to communicate loss and delay events.

- An on-path observer may know the network condition also by reading the Event bits.

New patterns may be added based on the methods in draft-ietf-ippm-explicit-flow-measurements

CoAP PM: Use Cases

The CoAP PM Option allows end-to-end measurements between the client and the server

Split measurements are also allowed. The intermediaries or on-path observers could be:

- Probes that must be able to see deep into application.
- Proxies, tasked by CoAP clients to perform requests on their behalf (RFC 7252)

Different application scenarios are considered:

- ➢ Non-proxying endpoints
- ➢ Collaborating proxies
- ➢ Non-collaborating proxies
- ➢ Caching or non-caching proxies
- ➢ DTLS
- ➢ OSCORE

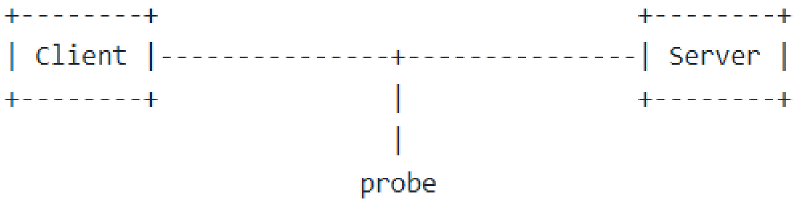
Application Scenarios (1/2)

➢ Non-proxying endpoints

The CoAP PM Option can be applied end-to-end between client and server and, since it is Elective, it can be ignored by an endpoint that does not understand it.

Measurements:

- e2e (Client-Server)
- on-path upstream and downstream (Probe)
- on-path intra-domain portion (with more Probes)



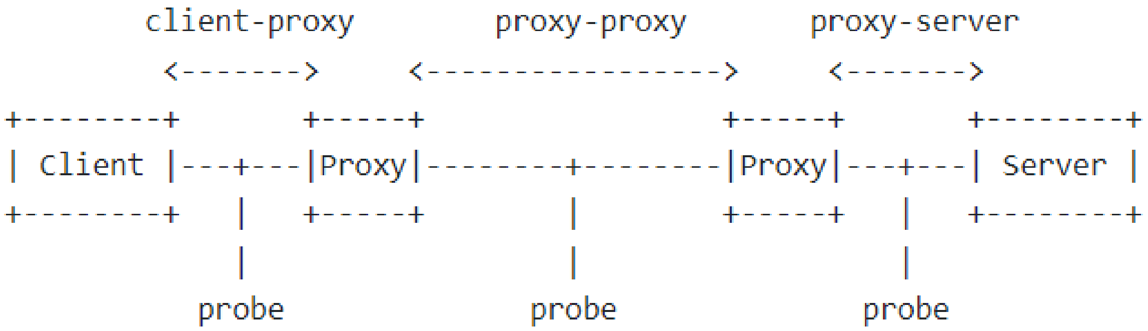
```
+--------+                             +--------+
| Client |--------------+--------------| Server |
+--------+              |              +--------+
                        |
                      probe
```

➢ Collaborating proxies

The CoAP PM Option can be applied end-to-end between client and server (or between collaborating Proxies).

Measurements *in case of collaborating proxies:*

- between Client-Server, Proxy-Proxy, Proxy-Server
- on-path upstream and downstream (Probe and/or Proxy)
- on-path intra-domain portion



```
           client-proxy       proxy-proxy       proxy-server
            <------->     <----------------->     <------->
+--------+         +-----+                 +-----+      +--------+
| Client |---+---|Proxy|--------+--------|Proxy|---+---| Server |
+--------+   |     +-----+      |          +-----+   |  +--------+
             |                  |                    |
           probe              probe                probe
```

Application Scenarios (2/2)

➢ Non-collaborating proxies

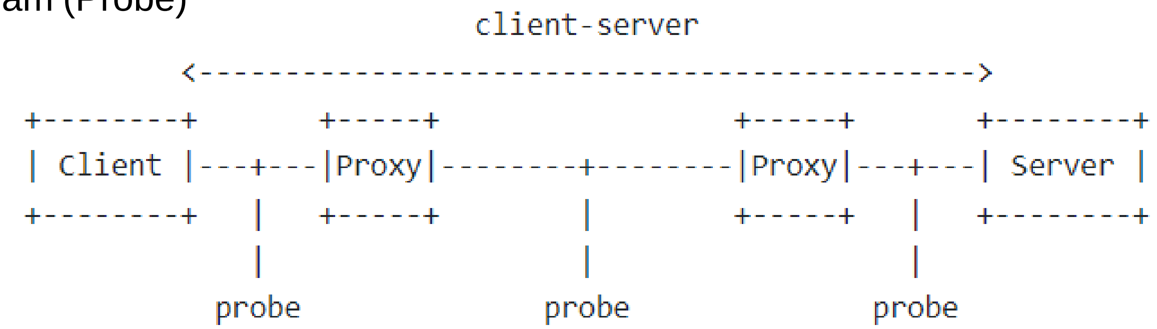
The PM Option is Proxy Unsafe and is unsafe for forwarding by a proxy that does not understand it.

- If there are non-collaborating and caching proxies, the measurements would not be possible.

An implementation MAY consider the PM Option as Safe-to-Forward if the proxies are non-caching

Measurements *in case of non-collaborating and non-caching proxies:*

- e2e (Client-Server)
- on-path upstream and downstream (Probe)
- on-path intra-domain portion



```
                         client-server
           <-------------------------------------------->
+--------+       +-----+                 +-----+       +--------+
| Client |---+---|Proxy|--------+--------|Proxy|---+---| Server |
+--------+   |   +-----+        |        +-----+   |   +--------+
             |                  |                  |
           probe              probe              probe
```

➢ DTLS

When a client uses a collaborating proxy the separated sessions are secured using DTLS but can still be measured. An on-path probe cannot perform the measurements in any case.

➢ OSCORE

If an OSCORE endpoint sends both outer and inner option, the inner is for measuring the connection to the end-to-end peer, and the outer can be used for measuring the connection to next proxy.

Changes in -03 and -04

It was presented during the Interim meeting in February

The comments received from Christian Amsüss, Marco Tiloca and Carsten Bormann have been addressed, in particular:

- Defined the Option as Proxy Unsafe instead of Safe-to-Forward
- Revised application scenarios by including the case of caching and non-caching proxies
- Reviewed DTLS and OSCORE cases
- Editorial Changes

Next Steps

- This draft is based on well-known methodologies applied in RFC9000 (SpinBit) and RFC9341 (sQuare Bit).
- It aims to meet the limited resources of constrained environment.

Evaluate WG Adoption

Welcome questions, comments

Thank you