Constrained Application Protocol (CoAP) Performance Measurement Option

draft-fz-core-coap-pm-03

Online, Feb 2023, interim meeting

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

Motivation

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

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

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

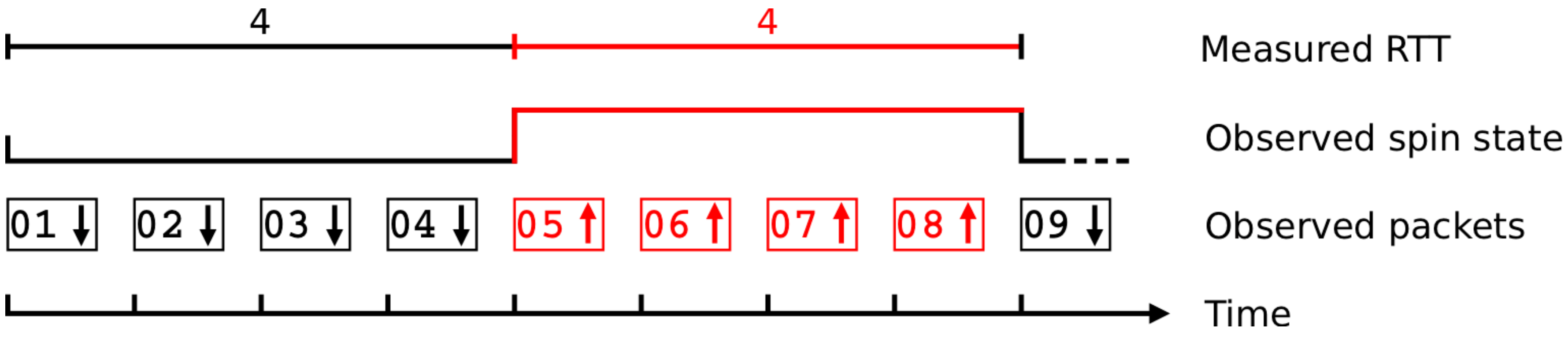
- Performance Measurement in constrained environment needs straightforward methodologies!

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**

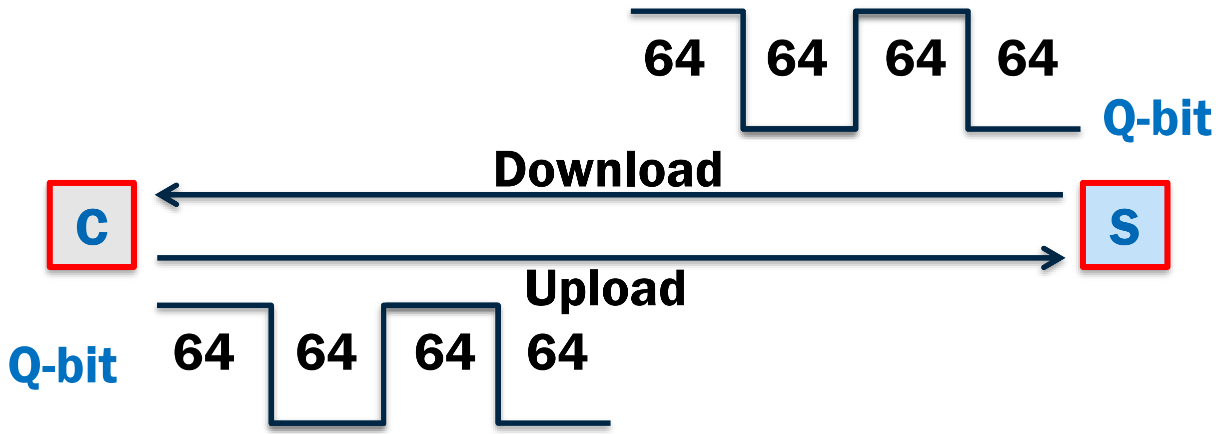
Round Trip Time: Spin bit

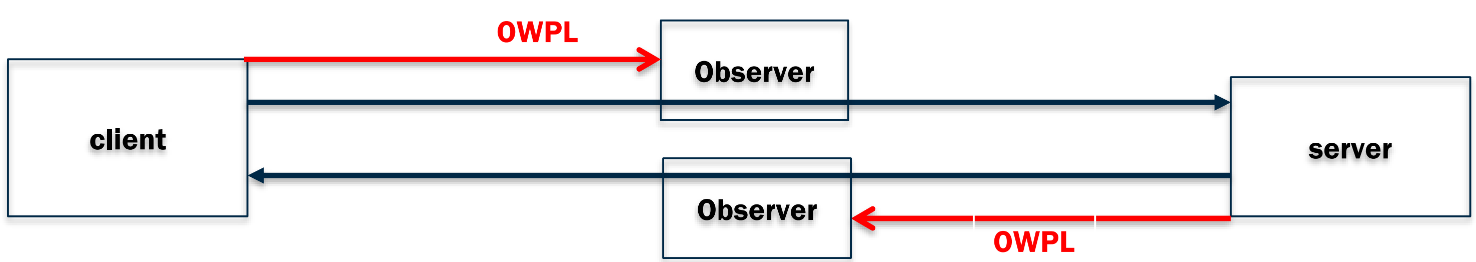
- Spin bit for RTT measurement was the first case of Explicit PM.
- It’s implemented, optionally, in QUIC protocol (https://www.ietfjournal.org/enabling-internet-measurement-with-the-quic-spin-bit/)
- The spinbit idea is to create a square wave signal on the data flow, using a bit, whose length is equal to RTT.
- An observer in the middle (wherever is located) can measure the end-to-end RTT only watching the spinbit.



One-Way Packet Loss: sQuare bit (Q-bit)

- The Q-bit (firstly described in draft-ferrieuxhamchaoui-quic-lossbits) creates square waves of a known length (e.g. 64 packets) as defined in the Alternate Marking RFC 9341





OWPL: One Way Packet Loss

COAP PM Option

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

```
 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. 2 bits 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.

```
  0   1   2   3   4   5   6   7
+---------------+---------------+
|               |               |
|  Option Delta | Option Length |   1 byte
|               |               |
+---------------+---------------+
\                               \
/         Option Delta          /   0-2 bytes
\          (extended)           \
+-------------------------------+
\                               \
/         Option Length         /   0-2 bytes
\          (extended)           \
+-------------------------------+
\                               \
/                               /
\                               \
/         Option Value          /   0 or more bytes
\                               \
/                               /
\                               \
+-------------------------------+
```

```
+-+-+-+-+-+-+-+-+
|0|C|   Event   |
+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+
|1|Q|S|  Event  |
+-+-+-+-+-+-+-+-+
```

Example: the Event bits can be divided into two parts: loss event bits and delay event bits.

- An end point can define different levels of thresholds and set the delay/loss event accordingly.

An on-path observer (Proxy or Gateway) knows the network condition by reading the Event bits.

- It MAY communicate with Client and Server to set some parameters based on the performance.

CoAP PM: Use Cases

The main usage of the CoAP PM Options is for end-to-end measurement between the client and the server

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

- Network Functions or Probes that must be able to see deep into application.
- Gateway or Proxies, tasked by CoAP clients to perform requests on their behalf (RFC7252)

Different cases:

- ➢ Non-proxying endpoints
- ➢ Collaborating or Non-collaborating proxies
- ➢ 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 (Observer)
- on-path intra-domain portion

```
+--------+                                  +--------+
| Client |---------------+----------------| Server |
+--------+               |                  +--------+
                         |
                     observer
```

➢ **OSCORE**

- If the CoAP PM Option is sent as Outer Option, it allows both end-to-end and on-path measurements

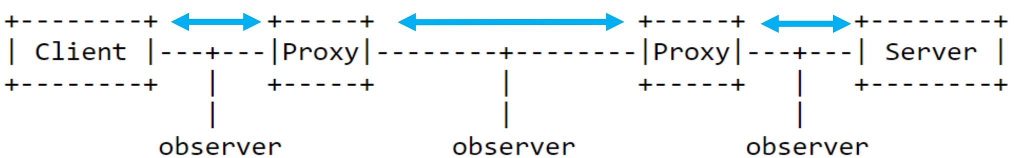
Application Scenarios (2/2)

➢ **Collaborating or Non-collaborating proxies**

The CoAP PM Option can be applied end-to-end between client and server (or between collaborating Proxies). Since it is Safe-to-Forward, it is intended to be safe for forwarding by a non-collaborating proxy.

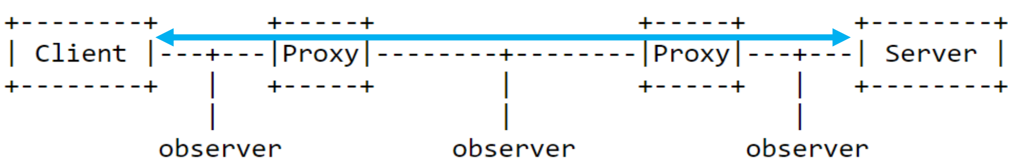
Measurements *in case of collaborating proxies*

- e2e (Client-Server or Proxy-Proxy)
- on-path upstream and downstream (Observer and/or Proxy)
- on-path intra-domain portion



Measurements *in case of non-collaborating proxies*

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



Additional EFM Methods

- draft-ietf-ippm-explicit-flow-measurements includes additional methods which are described hereinafter
- The Explicit Flow Measurement Techniques can be used alone or in combination
- Each technique uses a small number of bits and exposes a specific measurement (Delay or Losses)

The Delay bit

- The Delay bit is a single bit RTT measurement (like the Spin bit).

How Delay bit works

The marking bit is the Delay bit (D-bit) and a packet with D-bit=1 is a Delay bit Sample (DbS).

- The idea is to have a single marked packet, the DBS, generated by the Client, that bounces between Client and Server, using the production traffic.
- When the Client doesn’t detect the DbS for more then a specific time, Tmax (greater than maximum RTT: e.g. 1000 ms.), the DbS is declared lost; so the Client regenerates the DbS.
- Client and Server don’t reflect the DbS if the reflection time is more than 1 ms. (the application delay threshold: «E»).
- Observer:
    - It knows Tmax
    - Every two consecutive DbS it calculates RTT:
        - «Valid RTT» Rule => RTT< Tmax (in practice we use RTT< 90% Tmax)

“Delay Bits” Summary

	Bits	Unidirectional Observer	Bidirectional Observer	# of Measurements	Impairment resiliency
S: Spin Bit	1	RTT	x2 Half-RTT	Very High	Low
D: Delay bit	1	RTT	x2 Half-RTT	Medium °	High
D^: Hidden Delay bit	1	RTT^	x2 Left Half-RTT^ Right Half-RTT	High ~	High
SD: Spin bit + Delay bit *	2	RTT	x2 Half-RTT	Very High	High

° It depends on the “application delay” threshold (e.g. E=1 ms.), causing DbS discarded, and on DbS losses. But many of these missing measurements are “errored” measurements.

~ The “application delay” threshold (e.g. E=1 ms.) is only on the Server (see previous note).

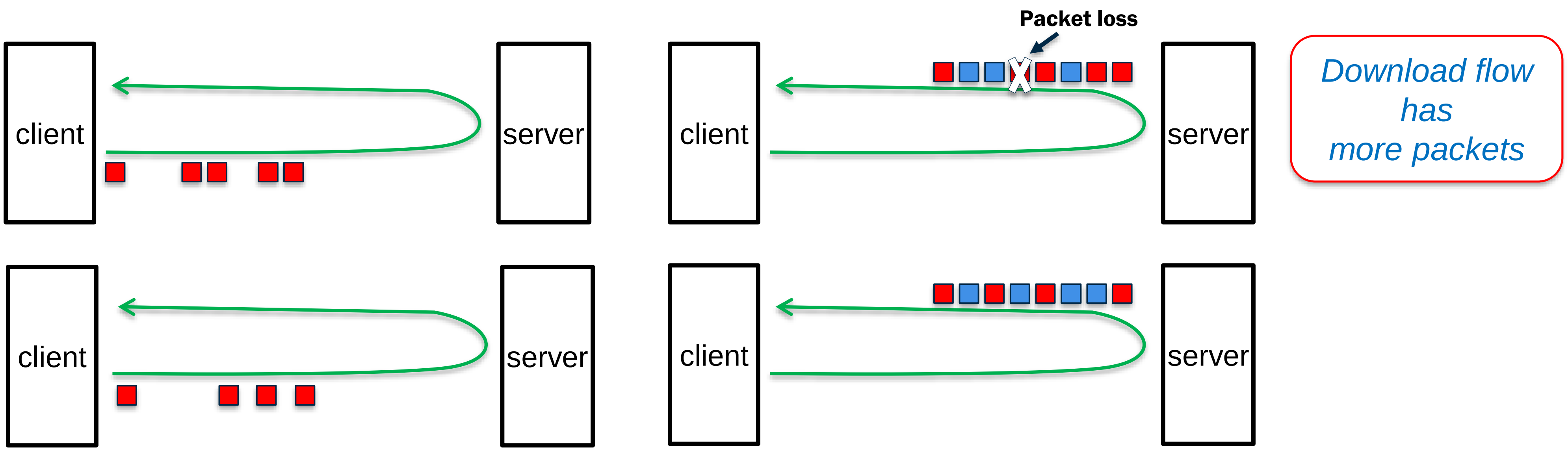
* Both algorithms work independently; an observer could use approximate spin bit measures when delay bit ones aren't available.

X2 Same metric for both directions.

^ Masked metric (real value can be calculated only by those who know the Additional Delay).

round Trip packet loss: T-bit

- The Client generate a «train» of market packets (using the T-bit)
- The Server «reflects» these packets (marking production packets flowing in the opposite direction). The Server inserts some not marked packets if download flow has more packets than upload flow.
- The Client reflects the marked packets.
- The Server again reflects the marked packets (two complete Client- Server rounds, so an intermediate Observer can see the «train» twice and compare the marked packets number to measure the RT Packet Loss).
- The Client generate a new train of market packets and so on.



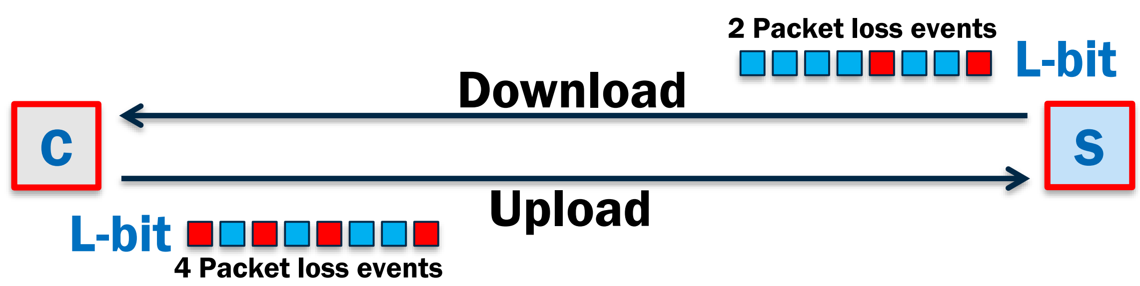
Marked packets: red,
Not Market packets: blue

One-Way Packet Loss: Q-bit+L-bit (Loss event bit)

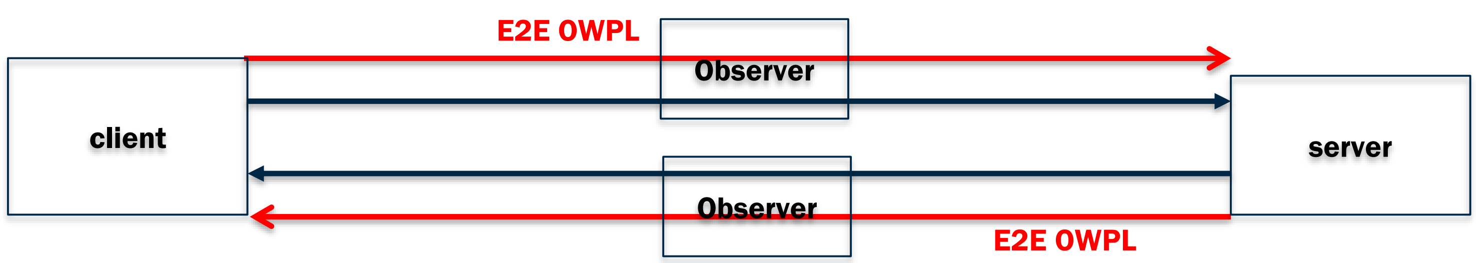
- This method uses 2 bits: the sQuare bit (Q-bit) and Loss event bit (L-bit).

- The L-bit (firstly described in draft-ferrieuxhamchaoui-quic-lossbits) marks a packet each time the protocol detect a loss packet event.

L-bit Marked packets: red,
L-bit Not Market packets: blue



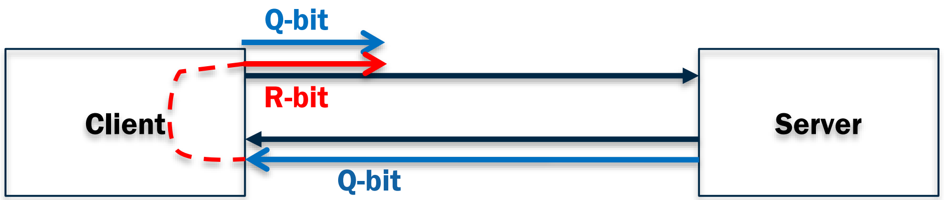
- L-bit measurement:



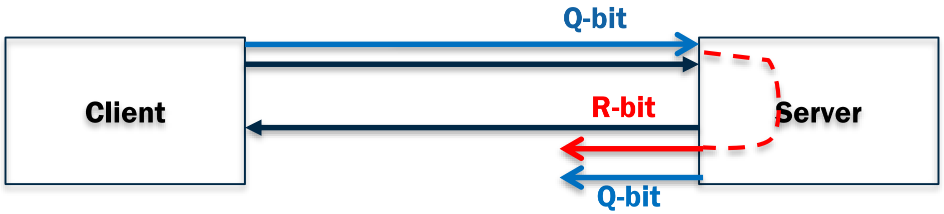
OWPL: One Way Packet Loss

OW Packet Loss: Q-bit+R-bit (Reflection square bit)

- This method uses 2 bits: the sQuare bit (Q-bit) and Reflection square bit (R-bit).
- The idea is to reflect the Q-bit in the opposite direction using the R-bit.
- The sizes of the transmitted R-bit blocks are the “average sizes” of the received Q-bit blocks.
- This idea allows to have continuous alternate marked packet blocks in both directions.
- The Client generates the Q-bit signal and reflects the received Q-bit signal using the R-bit signal:



- The Server does the same in the opposite direction:



“Loss Bits” Summary

Method	Bits	Unidirectional Observer	Bidirectional Observer	Proto	Measurement Fidelity	Measurement Delay
T round Trip loss bit	1+spin	Round Trip	Round Trip Half-RT x2	*	Rate by sampling $\frac{1}{3}$ to $\frac{1}{3*ppa}$ packets over 2 RTT	~6 RTT
Q sQuare bit	1	Upstream	Upstream x2	*	Rate over N packets (e.g. N=64)	N packets (e.g. B-64)
L Loss event bit	1	End-to-End	End-to-End x2	#	Loss shape and rate	Min: RTT Max: RTO
QL sQuare + Loss event bits	2	Upstream Downstream End-to-End	Upstream x2 Downstream x2 End-to-End x2	#	→ see Q → see Q|L → see L	→ see Q → see L → see L
QR sQuare + Reflection square bit	2	Upstream “3/4 RT” Opp. Dir. E2E	Upstream x2 “3/4 RT” x2 End-to-End x2 Downstream x2 Half-RT x2	*	Rate over $N * ppa$ packets (e.g. N=64)	Upstream: see Q Others: $N * ppa$ pkts (e.g. N=64)

*	All protocols	x2	Metric in both directions	ppa	Packets-per-Ack
#	Protocols with loss detection (w/ or w/o pure ACK loss detection)			Q|L	See Q if Upstream loss is significant; L otherwise

EFM Probes on user devices/hosts

- draft-cnbf-ippm-user-devices-explicit-monitoring proposes to put the Explicit Flow Measurements probe also on the user device (e.g. mobile phones, PCs) or on the server.
- “User device EFM rules”:
  1. The device owner decides whether to mark his traffic.
  2. The device owner decides whether to share his performance data.
- Strenghts:
  1. Scalability: On the user device there are few connections to monitor.
  2. More precise measurements: Client application delay can be measured.
  3. Both directions monitoring.
  4. Network monitoring equipment savings: Network probes can monitor only impaired connections through “user device and network probes coordination”. It’s possible to set alarm thresholds on the user device. In this case network probes, also embedded into network nodes, need to monitor only a limited number of connections.

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