DOTS Q-Block
interop testing report

IETF111 DOTS WG
Kaname Nishizuka/NTT Communications
Jon Shallow
Interop Plan

• Target Draft
  – draft-ietf-core-new-block-14

• 2 independent implementations
  – go-dots (https://github.com/nttdots/go-dots) by NTT
  – Jon’s Implementation

• 2 days of preliminary interop testing of Q-Block2
  – DOTS server to DOTS client only (Section 10.2)

• Quick recap
  – Both ends use libcoap library forked by Jon
  – Successfully communicated each other
  – even with certain rate of packet loss
Design of Q-Block

Purpose
• supporting Non-confirmable messages
  – Needed for (large) telemetry updates
  – NON because of DDoS uni-directional network pipe loss
• fewer packet interchanges
• faster recovery (should any of the blocks get lost in transmission)
Testing Environment
(without loss)

DOTS client => CoAP Client

DOTS Server => CoAP Server

Figure 8: Example of NON Notifications with Q-Block2 Option (Without Loss)
Testing Environment
(with loss)

Asymmetric random packet loss from server
Induced by iptables

DOTS client
=>CoAP Client
=>CoAP Server

Figure 10: Example of NON Notifications with Q-Block2 Option (Blocks Recovery)
Result

Without loss
• Successfully received entire body larger than MAX_PAYLOADS payloads
  • Every MAX_PAYLOADS count gives Congestion Control pause if no ‘Continue’ response
• Fewer packets (compared with Block2 (which normally requires Confirmable))

With loss
• Successfully recovered entire body even with 1%, 3%, 5%, 10% packet loss rate
• Fewer packets: Reclaim of missing blocks in one go
• Found a few libcoap bugs (now fixed)
libcoap bugs and discussion (1/2)

Issue:
It was not clear before the interop which request method should be used for asking for missing blocks (e.g. PUT triggered a large blocked response)

Conclusion:
When requesting the additional blocks (Block2) or requesting the missing blocks (Q-Block2) then the request method (+ appropriate (Q-)Block2 options) is the same for the next blocks as the initial request - even if it was a PUT
libcoap bugs and discussion (2/2)

Issue:
Initial GET for the entire body (NUM is zero and Mbit is set) could be misunderstood by the CoAP server as a request for the remaining missing blocks of the previous blocked response.

Conclusion:
GET (NUM is zero and Mbit is set) should be treated as a request for the entire (new or refreshed) body.
[draft 4.4 Using the Q-Block2 Option
   NUM is zero: This is a request for the entire body.]
Questions?
Thank You