Constrained RESTful Environments WG (core)

Chairs:

Andrew McGregor <andrewmcgr@gmail.com>

Carsten Bormann <cabo@tzi.org>

Mailing List:

core@ietf.org

Jabber:

core@jabber.ietf.org

- **We assume people have read the drafts**
- **Meetings serve to advance difficult issues by making good use of face-to-face communications**
- **Note Well: Be aware of the IPR principles, according to RFC 3979 and its updates**

✓Blue sheets
✓Scribe(s):
http://tools.ietf.org/wg/core/minutes

Note Well

Any submission to the IETF intended by the Contributor for publication as all or part of an IETF Internet-Draft or RFC and any statement made within the context of an IETF activity is considered an "IETF Contribution". Such statements include oral statements in IETF sessions, as well as written and electronic communications made at any time or place, which are addressed to:

- The IETF plenary session
- The IESG, or any member thereof on behalf of the IESG
- Any IETF mailing list, including the IETF list itself, any working group or design team list, or any other list functioning under IETF auspices
- Any IETF working group or portion thereof
- Any Birds of a Feather (BOF) session
- The IAB or any member thereof on behalf of the IAB
- The RFC Editor or the Internet-Drafts function

All IETF Contributions are subject to the rules of RFC 5378 and RFC 3979 (updated by RFC 4879).

Statements made outside of an IETF session, mailing list or other function, that are clearly not intended to be input to an IETF activity, group or function, are not IETF Contributions in the context of this notice. Please consult RFC 5378 and RFC 3979 for details.

A participant in any IETF activity is deemed to accept all IETF rules of process, as documented in Best Current Practices RFCs and IESG Statements.

A participant in any IETF activity acknowledges that written, audio and video records of meetings may be made and may be available to the public.

Wednesday

All times are in time-warped EDT

- **15:20–15:23 Intro I**
- **15:23–15:43 CoAP Congestion Control (CG)**
- **15:43–15:50 Intro II**
- **15:50–16:00 Block post WGLC (CB)**
- **16:00–16:15 HTTP mapping (SL)**
- **16:15–16:20 Links-JSON (chairs)**
- **16:20–16:25 Core-Interfaces (chairs)**
- **16:25–16:30 Resource Directory (chairs)**
- **16:30–16:35 comi, alt trans (chairs)**
- **16:35–16:45 No-Response (AB)**
- **16:45–16:50 Flextime**

CoAP Simple Congestion Control/Advanced (CoCoA)

draft-bormann-core-cocoa-02

Carsten Bormann – Universität Bremen TZI
cabo@tzi.org
August Betzler, Carles Gomez, Ilker Demirkol
Universitat Politècnica de Catalunya (UPC)/Fundació i2cat
carlesgo@entel.upc.edu

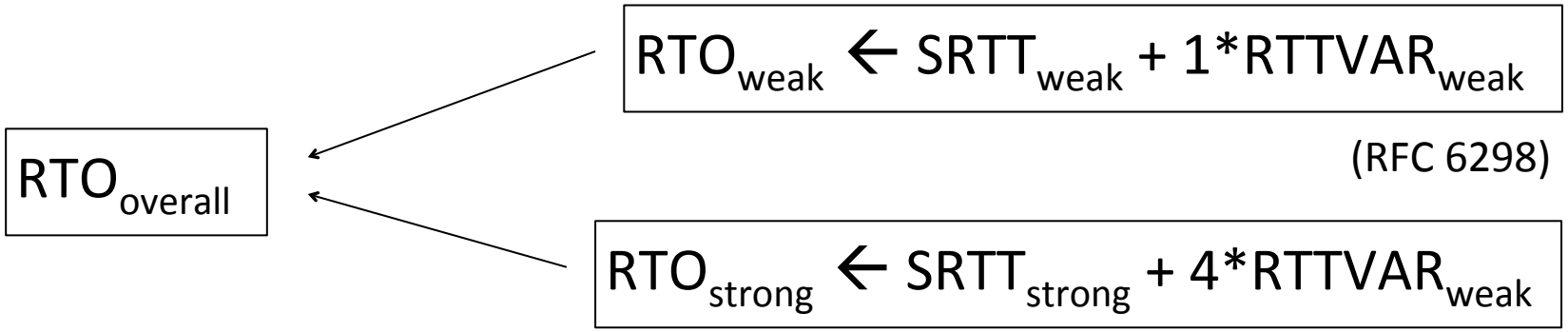
UPC members have been partly supported by the Spanish Government through project TEC2012-32531, and FEDER

Context (I)

- Default CoAP congestion control for CONs (RFC 7252)
  - RTO chosen from a fixed interval
  - Binary Exponential Backoff (BEB)
- CoCoA
  - Simple mechanism for advanced congestion control
    - Using RTT measurements for CONs
    - Rules for NONs

Context (II)

- RTO estimator
  - Input from *weak* and *strong* estimators
  - $RTO_{overall}$ is evolved from the estimator that made the most recent contribution



Updates in -02 (1/3)

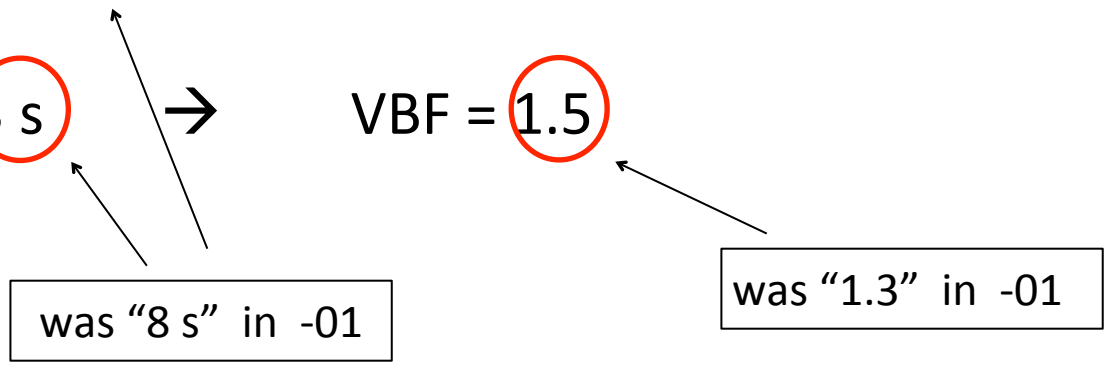
- Reduced RTO_{weak} contribution:
  - $RTO_{overall} := 0.25*RTO_{weak} + 0.75*RTO_{overall}$
  - $RTO_{overall} := 0.5*RTO_{strong} + 0.5*RTO_{overall}$
  - Only responses obtained before the 3rd retransmission update RTO_{weak}
- Aging mechanism for high RTO values
  - If RTO > 3 s, and not updated for 4*RTO , then
  - RTO = 1 + 0.5*RTO s
    - Converge towards default RTO values

Updates in -02 (2/3)

- Blind RTO estimate
  - If only the initial RTO estimate is available
    - In -01: RTO = $2^{\text{Nparallel exchanges}}$ s
    - In -02: RTO = $N_{\text{parallel exchanges}} * 2$ s
  - Changed from exponential to linear approach
  - Benefit
    - Avoids too large RTO estimates

Updates in -02 (3/3)

- Variable Backoff Factor
  - Parameter updates
    - RTO < 1 s → VBF = 3
    - 1 ≤ RTO ≤ 3 s → VBF = 2
    - RTO > 3 s → VBF = 1.5



- Minor editorial updates and clarifications

Running code

- cocoa-02 has been implemented for Californium
  - Except for a parameter of the aging mechanism
    - Applied when large RTO is not updated after 1 minute
  - Optional congestion control “layer”
  - Unconstrained platform
    - Remote endpoints identified at the highest granularity: IP address and port number
- Note: Californium is publicly available
  - The cocoa-02 implementation for Californium will also be available soon (August/September)

Experiments and results (1/4)

- PC
  - Running multiple instances of Californium client
    - cocoa-02

- ETH Zürich Flocklab multihop testbed
  - 30 Tmote Sky motes
  - Full Contiki OS 6LoWPAN stack
    - Erbium CoAP implementation
      - Without CoCoA
    - Two different link layer mechanisms
      - ContikiMAC
      - Null RDC

Experiments and results (2/4)

- Throughput (req/s)
  - 1-to-1 scenario:

	No RDC	ContikiMAC
Default CoAP	0.67	0.56
CoCoA	1.18	0.89

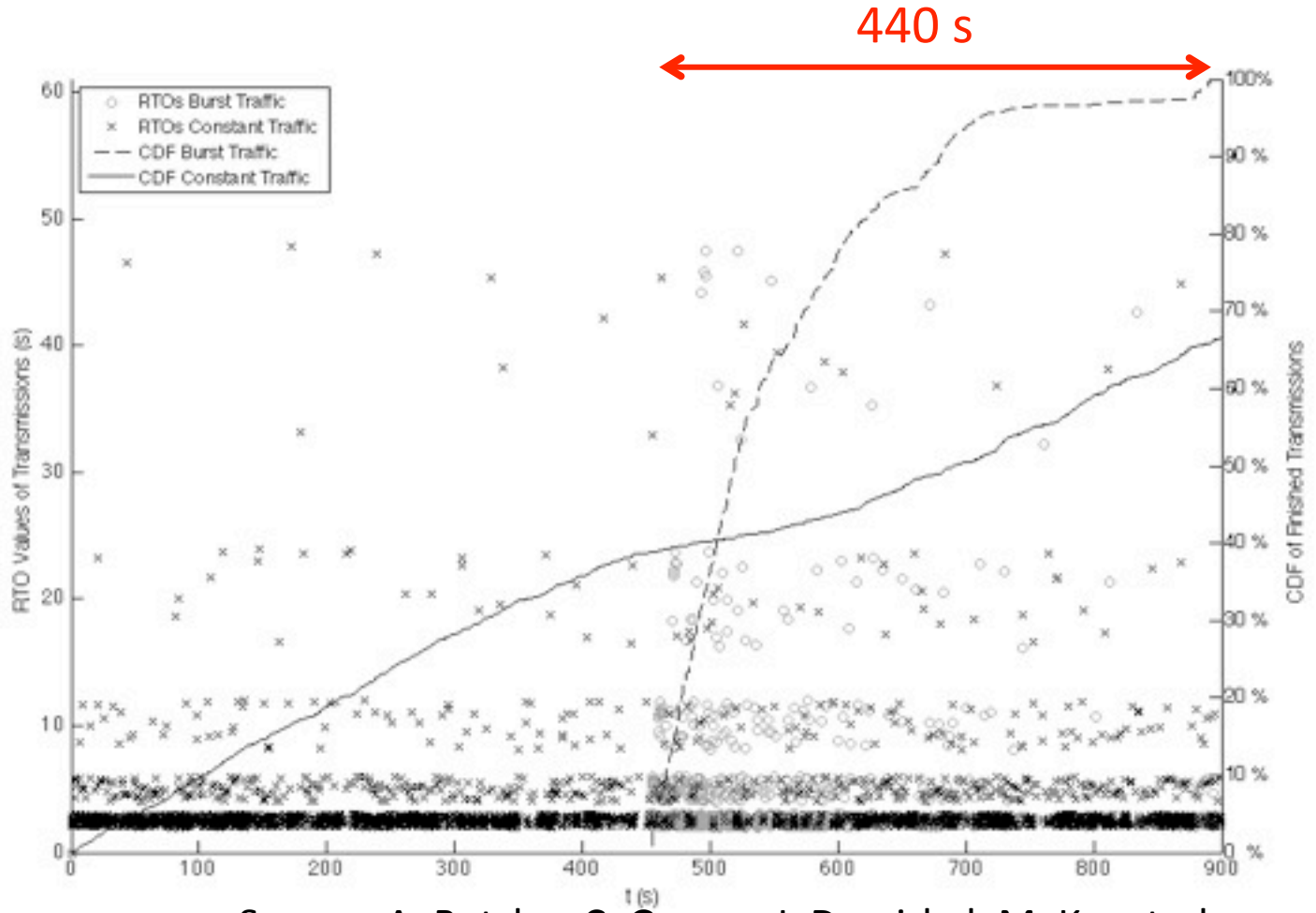
  - many-to-many scenario:
    - Note: many ϵ [25, 30]

	No RDC	ContikiMAC
Default CoAP	4.23	1.60
CoCoA	5.34	1.79

Source: A. Betzler, C. Gomez, I. Demirkol, M. Kovatsch, "Congestion Control for CoAP cloud services", SOCNE 2014.

Experiments and results (3/4)

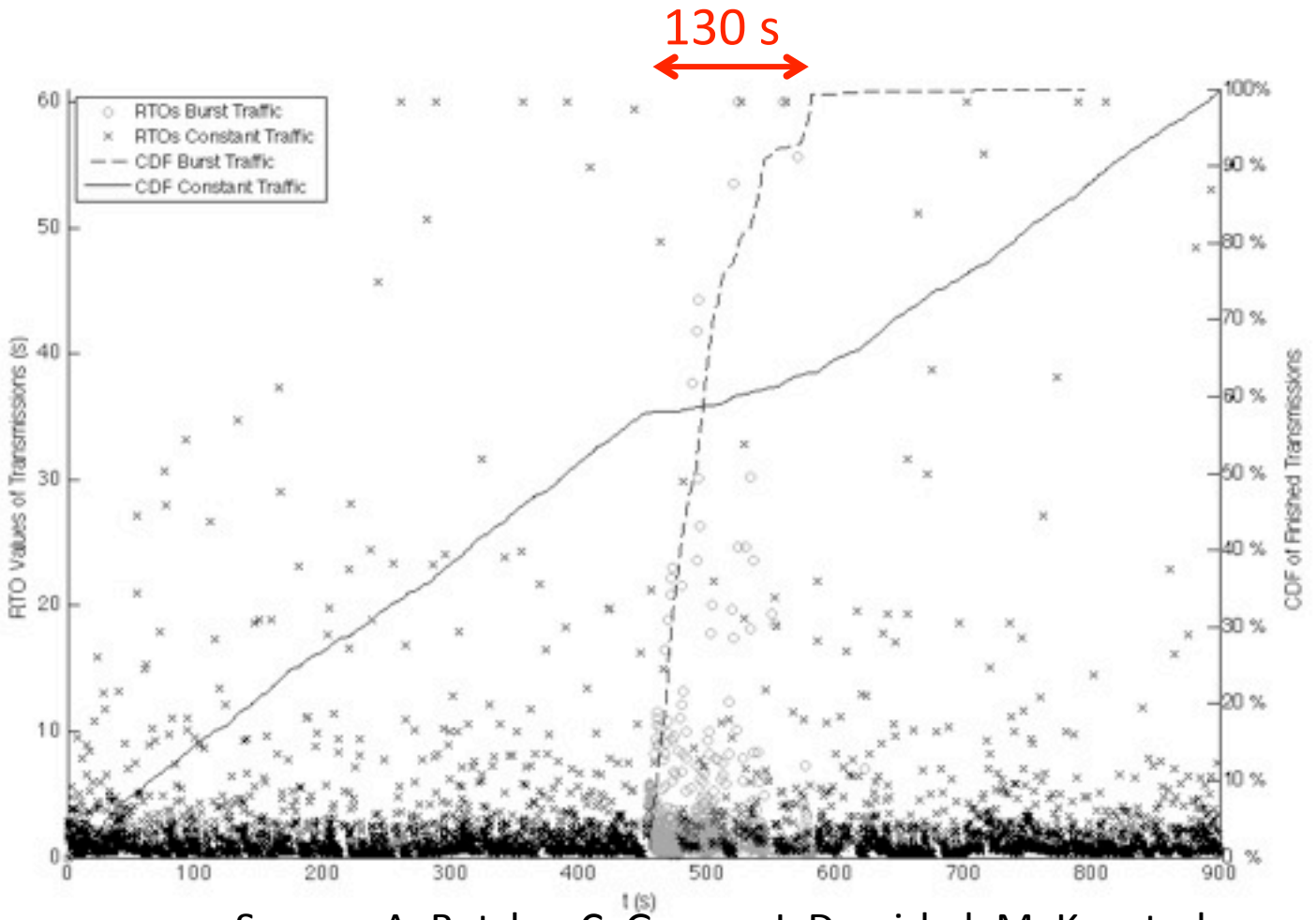
- Burst traffic scenario: default CoAP



Source: A. Betzler, C. Gomez, I. Demirkol, M. Kovatsch, "Congestion Control for CoAP cloud services", SOCNE 2014.

Experiments and results (4/4)

- Burst traffic scenario: CoCoA



Source: A. Betzler, C. Gomez, I. Demirkol, M. Kovatsch, "Congestion Control for CoAP cloud services", SOCNE 2014.

GPRS scenario: preliminary results

Conclusions

- CoCoA is becoming stable
  - Different types of evaluations
    - Simulation of 802.15.4 networks
    - Experiments in an 802.15.4 testbed
    - (Preliminary) experiments in a GPRS scenario
  - Running code
    - Implemented for Californium
    - Implemented for Zolertia Z1 (Cooja evaluations)

Call to Action

- Please experiment with Californium+cocoa-02
  - We'll announce availability on the list
- Please implement cocoa-02
- Please provide feedback before IETF91

Questions from the agenda

- Is this getting close to being useful?
- Should we continue to work on this?
- When do we want to adopt this?

Milestones (from WG charter page)

http://datatracker.ietf.org/wg/core/charter/

Document submissions to IESG:

- **Done** CoAP protocol specification with mapping to HTTP Rest API to IESG
- **Oct 2013** Blockwise transfers in CoAP to IESG
- **Done** Observing Resources in CoAP to IESG
- **Done** Group Communication for CoAP to IESG
- **Jan 2014** BP for HTTP-CoAP Mapping Impl to IESG
- **Jan 2014** CoRE Link Collections in JSON to IESG
- **May 2014** CoRE Interfaces to IESG
- **Dec 2099** HOLD (date TBD) Constrained security bootstrapping specification to IESG

RFC 7252

~~draft-ietf-core-coap-18~~

- Was approved 2013-07-11
- Waited some more for AUTH48 until 2014-06-26
- draft-mcgrew-tls-aes-ccm-ecc
  - Defines the DTLS ciphersuites for RPK and Cert mode: TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
  - (alongside with TLS_PSK_WITH_AES_128_CCM_8 for PSK mode, RFC 6655)
  - IESG state “Approved-announcement to be sent::Point Raised – writeup needed”
- draft-ietf-tls-oob-pubkey (in queue since 2014-02-06)
  - Defines RPK for DTLS
  - IESG state “Waiting for Writeup” (post IETF last call)

Submitted to IESG

- **draft-ietf-core-observe-14 (2014-07-20) for Standards-Track (Proposed Standard)**
- **draft-ietf-core-groupcomm-20 (2014-07-21) for Informational**

WG documents

- draft-ietf-core-block — 2nd WGLC passed
  - waiting for revision
- draft-ietf-core-http-mapping

- draft-ietf-core-links-json
  - done, but waiting for more implementation experience
- draft-ietf-core-resource-directory
  - charter work needed, to resume activity!
- draft-ietf-core-interfaces
  - to resume activity!

Related Work Reports

Quiet Period for IoT Plugtests

- Next: **ETSI-M2M** Plugtest
  - Testing the HTTP binding, though
- Where?
  - SOPHIA ANTIPOLIS, FRANCE
- When?
  - 10–12 SEPTEMBER 2014
- How Much?
  - Free!
- Tests:
  - ETSI-M2M Devices, DA, GSCL, NSCL, NA…

Agenda Bashing

Wednesday

- **15:20–15:23 Intro I** All times are in time-warped EDT
- **15:23–15:43 CoAP Congestion Control (CG)**
- **15:43–15:50 Intro II**
- **15:50–16:00 Block post WGLC (CB)**
- **16:00–16:15 HTTP mapping (SL)**
- **16:15–16:20 Links-JSON (chairs)**
- **16:20–16:25 Core-Interfaces (chairs)**
- **16:25–16:30 Resource Directory (chairs)**
- **16:30–16:35 comi, alt trans (chairs)**
- **16:35–16:45 No-Response (AB)**
- **16:45–16:50 Flextime**

Thursday

- 17:30–17:33 Intro
- 17:33–17:53 CoAP-MQ
- 17:53–18:15 other new work
- 18:15–18:30 Flextime

Group 1: WG docs

Wednesday

All times are in time-warped EDT

- **15:20–15:23 Intro I**
- **15:23–15:43 CoAP Congestion Control (CG)**
- **15:43–15:50 Intro II**
- **15:50–16:00 Block post WGLC (CB)**
- **16:00–16:15 HTTP mapping (SL)**
- **16:15–16:20 Links-JSON (chairs)**
- **16:20–16:25 Core-Interfaces (chairs)**
- **16:25–16:30 Resource Directory (chairs)**
- **16:30–16:35 comi, alt trans (chairs)**
- **16:35–16:45 No-Response (AB)**
- **16:45–16:50 Flextime**

-block-15

- editorial update from -14
  - -14 was plugtested in London and seemed to work well
- WGLC 2014-07-04..-18
  - #368 (editorial)
- Older ticket #367 (Content-Format mismatch)
  - clearly, abort transfer on mismatch
  - similarly, could add more text for slide 25–27 of IETF88 slides
- Is this done now?

Wednesday

All times are in time-warped EDT

- **15:20–15:23 Intro I**
- **15:23–15:43 CoAP Congestion Control (CG)**
- **15:43–15:50 Intro II**
- **15:50–16:00 Block post WGLC (CB)**
- **16:00–16:15 HTTP mapping (SL)**
- **16:15–16:20 Links-JSON (chairs)**
- **16:20–16:25 Core-Interfaces (chairs)**
- **16:25–16:30 Resource Directory (chairs)**
- **16:30–16:35 comi, alt trans (chairs)**
- **16:35–16:45 No-Response (AB)**
- **16:45–16:50 Flextime**

Guidelines for HTTP-CoAP Mapping Implementations

Angelo Castellani, Salvatore Loreto, Akbar Rahman, Thomas Fossati, Esko Dijk

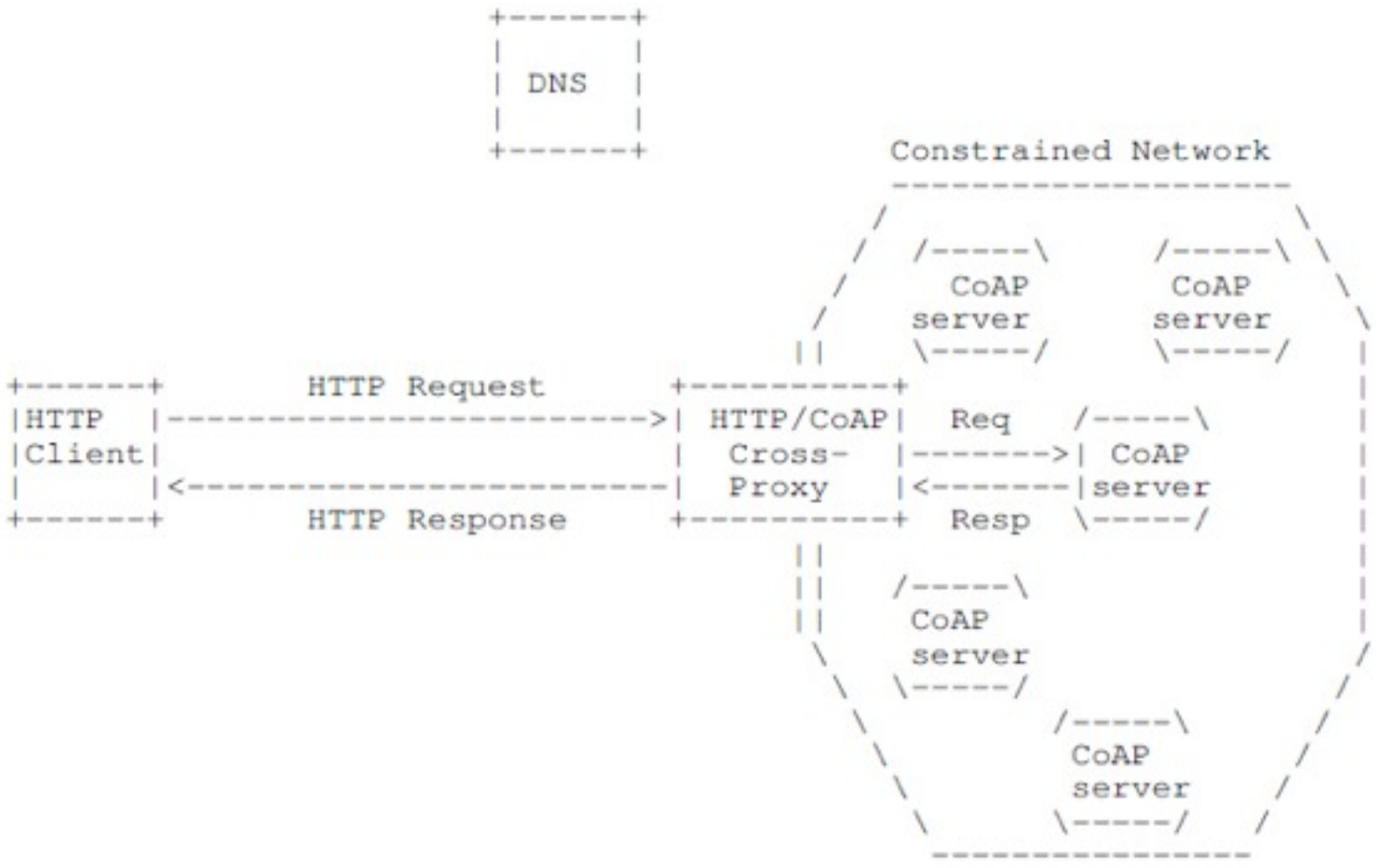
IETF-90, July 2014

http://tools.ietf.org/html/draft-ietf-core-http-mapping-04

Main Changes (from IETF-89 London)

- Changes from ietf-03 to ietf-04:
  - Expanded use case descriptions in Section 4
  - Fixed/enhanced discovery examples in Section 5.4.1
  - Addressed Ticket #365 (Media-type conversion by HTTP- CoAP proxy) in new:
    - Section 6.3.1 (Generalized media-type mapping)
    - Section 6.3.2 (Content translation)
  - Updated HTTPBis WG draft references to recently published RFC numbers
  - Various editorial improvements

Reverse Cross-Protocol Proxy Deployment Scenario



Reminder: Focus of I-D is reverse HTTP-CoAP (HC) Cross Proxy

Closed Ticket #365 (1/4)

- Ticket #365 (Media-type conversion by HTTP-CoAP proxy)
  - Need to specify that media-type conversion is needed in HC proxy
  - Conversion between the CoAP Content Format and HTTP media type is needed, in both directions

Closed Ticket #365 (2/4)

- Overview (Section 6.3)
  - A HC Proxy translates HTTP media types [RFC7231] and content encodings [RFC7231] into CoAP content formats (Section 12.3 of [RFC7252])
  - Media type translation can happen in:
    - GET, PUT or POST requests going from HTTP to CoAP,
    - and in 2.xx (i.e. successful) responses going from CoAP to HTTP
  - Specifically:
    - PUT and POST need to map the Content- Type and Content-Encoding HTTP headers into a CoAP Content-Format option,
    - whereas GET needs to map Accept and Accept-Encoding HTTP headers into a CoAP Accept option.
    - On the way back, the CoAP Content-Format option is renormalised into a suitable HTTP Content- Type and Content-Encoding combination

Closed Ticket #365 (3/4)

- Generalized media-type mapping (Section 6.3.1)
  - Is needed whenever one of the **many** HTTP media-types do not exist in the **short** CoAP Content-Format table
  - Prefer loosing some information about the content instead of returning a failure (i.e. "*415 Unsupported media-type" for POST/PUT response*)
  - Table 2 defines the default lookup table for the "loose" media-type mapping.
  - Given an input media-type, the table returns its best generalised media-type using longest prefix match
  - OPTIONAL feature - Implementations supporting this mapping SHOULD provide a flexible way to define the set of media-type generalisations allowed

Internet media-type	Generalised media-type
application/*+xml	application/xml
application/*+json	application/json
text/xml	application/xml
text/*	text/plain
/	application/octet-stream

Table 2: Media type generalisation

Closed Ticket #365 (4/4)

- Media-type mapping algorithm (Section 6.3.2)
  - This section defines the algorithm used to map an Internet media type to its correspondent CoAP Content-Format.

```
INPUT:  C-T and C-E
OUTPUT: C-F or Fail

1.  if no C-T: return Fail
2.  C-F = MAP[C-T, C-E]
3.  if C-F is not None: return C-F
4.  if C-E is not "identity":
5.    if C-E is supported (e.g. gzip):
6.      decode the representation accordingly
7.      set C-E to "identity"
8.    else:
9.      return Fail
10. repeat steps 2. and 3.
11. if C-T allows a non-lossy transformation into \
12.    one of the supported C-F:
13.      transcode the representation accordingly
14.      return C-F
15. if GMAP is defined:
16.   C-F = GMAP[C-T]
17.   if C-F is not None: return C-F
18. return Fail
```

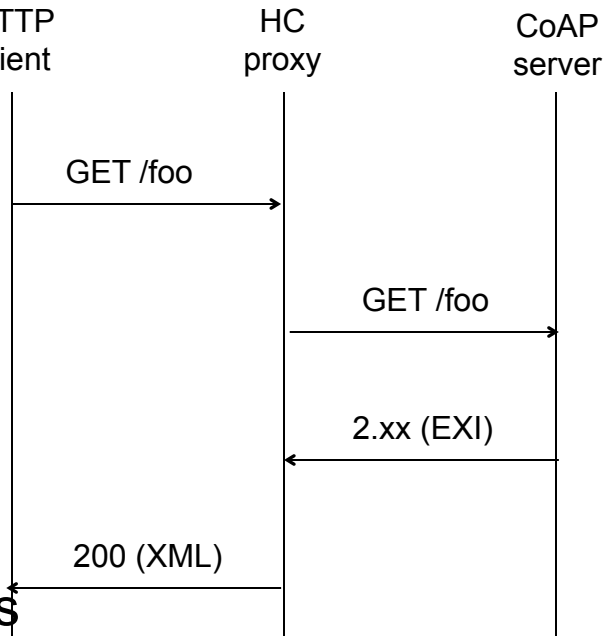
Figure 2

Where:

- C-T, C-E, and C-F stand for the values of the Content-Type (or Accept), Content-Encoding (or Accept-Encoding) HTTP headers, and Content-Format CoAP option respectively.
- If C-E is not given it is assumed to be "identity".
- MAP is the mandatory lookup table, GMAP is the optional generalised table.

Content translation (Section 6.3.3)

- Some content formats specifically target constrained environments
  - Various web formats have their binary counterpart
  - e.g.: XML / EXI, JSON / CBOR
- Binary formats (EXI/CBOR) right now have still a low adoption rate on the “unconstrained “ web
  - Human-readable protocols/formats are preferred HTTP/XML/JSON
- Cross-proxy MAY translate these formats to its well-known, well-supported counterpart (preferably through external modules).
  - Access control and black-listing/white-listing rules SHOULD be provided together with this feature



<u>This is an OPTIONAL feature, HC proxy MUST be specifically configured to this kind of translations</u>

Last Open Ticket #366 (Mapping of Link Format payloads to be valid in HTTP domain)

- If a HTTP client retrieves a CoRE Link Format document via an HC proxy, the relative URI references may not be valid anymore.
- Example HTTP request to HC proxy:
  - GET http://hcproxy.example.com/hc/coap://mysensor.example.com/.well-known/core
- Example Link Format resource /.well-known/core retrieved by the HC proxy:
  - </sensors/temp>;rt="temperature-c";if="sensor"
- Which indicates a relative path on the CoAP server. While, to be fully correct, it should perhaps read:
  - <coap://mysensor.example.com/sensors/temp>;rt="temperature-c";if="sensor"
- That means the HC Proxy would need to modify the payload before returning it to the HTTP client. See for background: http://tools.ietf.org/html/rfc6690#section-2.2
- Is this desirable? Or should the HC proxy not change any payloads retrieved.
  Are there other content formats that need to be adapted in a similar way?

Flextime

- **We assume people have read the drafts**
- **Meetings serve to advance difficult issues by making good use of face-to-face communications**
- **Note Well: Be aware of the IPR principles, according to RFC 3979 and its updates**

✓Blue sheets
✓Scribe(s)

Note Well

Any submission to the IETF intended by the Contributor for publication as all or part of an IETF Internet-Draft or RFC and any statement made within the context of an IETF activity is considered an "IETF Contribution". Such statements include oral statements in IETF sessions, as well as written and electronic communications made at any time or place, which are addressed to:

- The IETF plenary session
- The IESG, or any member thereof on behalf of the IESG
- Any IETF mailing list, including the IETF list itself, any working group or design team list, or any other list functioning under IETF auspices
- Any IETF working group or portion thereof
- Any Birds of a Feather (BOF) session
- The IAB or any member thereof on behalf of the IAB
- The RFC Editor or the Internet-Drafts function

All IETF Contributions are subject to the rules of RFC 5378 and RFC 3979 (updated by RFC 4879).

Statements made outside of an IETF session, mailing list or other function, that are clearly not intended to be input to an IETF activity, group or function, are not IETF Contributions in the context of this notice. Please consult RFC 5378 and RFC 3979 for details.

A participant in any IETF activity is deemed to accept all IETF rules of process, as documented in Best Current Practices RFCs and IESG Statements.

A participant in any IETF activity acknowledges that written, audio and video records of meetings may be made and may be available to the public.

Thursday

- 17:30–17:33 intro
- 17:33–17:53 CoAP-MQ
- 17:53–18:15 other new work
- 18:15–18:30 Flextime

Thursday

- 17:30–17:33 Intro
- 17:33–17:53 Wednesday * 2 / 3
- 17:53–18:13 CoAP-MQ
- 18:13–18:28 other new work
- 18:28–18:30 Flextime

Wednesday

All times are in time-warped EDT

- **15:20–15:23 Intro I**
- **15:23–15:43 CoAP Congestion Control (CG)**
- **15:43–15:50 Intro II**
- **15:50–16:00 Block post WGLC (CB)**
- **16:00–16:15 HTTP mapping (SL)**
- **16:15–16:20 Links-JSON (chairs)**
- **16:20–16:25 Core-Interfaces (chairs)**
- **16:25–16:30 Resource Directory (chairs)**
- **16:30–16:35 comi, alt trans (chairs)**
- **16:35–16:45 No-Response (AB)**
- **16:45–16:50 Flextime**

draft-ietf-core-links-json-00.txt

- RFC 6690 (link-format) documents are somewhat foreign to many web app developers
  - would prefer to have them in JSON format
- There is no standard way to represent link-format documents in applications
  - but everyone knows how to handle JSON

➔ Define a standard JSON translation for link-format

```
</sensors>;ct=40;title="Sensor Index",
</sensors/temp>;rt="temperature-c";if="sensor",
</sensors/light>;rt="light-lux";if="sensor",
<http://www.example.com/sensors/t123>
  ;anchor="/sensors/temp";rel="describedby",
</t>;anchor="/sensors/temp";rel="alternate"
```

➔

```
[{"href":"/sensors","ct":"40","title":"Sensor Index"},
  {"href":"/sensors/temp","rt":"temperature-c","if":"sensor"},
  {"href":"/sensors/light","rt":"light-lux","if":"sensor"},
  {"href":"http://www.example.com/sensors/t123",
   "anchor":"/sensors/temp","rel":"describedby"},
  {"href":"/t","anchor":"/sensors/temp","rel":"alternate"}]
```

Potential Issue: How to update

- Structure: Array of links
- RD update might
  - add links: trivial
  - change links: replace on href as key?
  - remove links (how to indicate this?)
- draft-ietf-appsawg-json-merge-patch was defined to solve problems like this
  - but does not fit: only can update object (map), not array
- restructure links-json as an object (map)?
  - are hrefs really unique in real-life link sets???

Wednesday

All times are in time-warped EDT

- **15:20–15:23 Intro I**
- **15:23–15:43 CoAP Congestion Control (CG)**
- **15:43–15:50 Intro II**
- **15:50–16:00 Block post WGLC (CB)**
- **16:00–16:15 HTTP mapping (SL)**
- **16:15–16:20 Links-JSON (chairs)**
- **16:20–16:25 Core-Interfaces (chairs)**
- **16:25–16:30 Resource Directory (chairs)**
- **16:30–16:35 comi, alt trans (chairs)**
- **16:35–16:45 No-Response (AB)**
- **16:45–16:50 Flextime**

Wednesday

All times are in time-warped EDT

- **15:20–15:23 Intro I**
- **15:23–15:43 CoAP Congestion Control (CG)**
- **15:43–15:50 Intro II**
- **15:50–16:00 Block post WGLC (CB)**
- **16:00–16:15 HTTP mapping (SL)**
- **16:15–16:20 Links-JSON (chairs)**
- **16:20–16:25 Core-Interfaces (chairs)**
- **16:25–16:30 Resource Directory (chairs)**
- **16:30–16:35 comi, alt trans (chairs)**
- **16:35–16:45 No-Response (AB)**
- **16:45–16:50 Flextime**

CoRE Resource Directory

draft-ietf-core-resource-directory-02 (in progress)

Z. Shelby, C. Bormann

CoRE WG, IETF-90 Toronto

Open issues in progress

- #369 – Semantic catalogue use case addition
- #370 – Integrate the DNS-SD mapping
  - Together with Kerry Lynn
- #371 – DDoS security consideration
- #372 – Example section with use cases
  - Help from the WG welcome
- #373 – Fix the registration update interface

Registration update interface

When are we done?

1. Close the current set of tickets in -02
2. Do one more editing round for -03
3. Get 1-2 expert reviews, at least one from OMA
4. Oh yes, add this to our charter ☺

Wednesday

All times are in time-warped EDT

- **15:20–15:23 Intro I**
- **15:23–15:43 CoAP Congestion Control (CG)**
- **15:43–15:50 Intro II**
- **15:50–16:00 Block post WGLC (CB)**
- **16:00–16:15 HTTP mapping (SL)**
- **16:15–16:20 Links-JSON (chairs)**
- **16:20–16:25 Core-Interfaces (chairs)**
- **16:25–16:30 Resource Directory (chairs)**
- **16:30–16:35 comi, alt trans (chairs)**
- **16:35–16:45 No-Response (AB)**
- **16:45–16:50 Flextime**

comi: draft-vanderstok-core-comi-04

- The draft seems to be maturing
- This would be great functionality to have
- Needs more coordination with coman/opsawg
  - BTW, please help review coman drafts!
- Where do we go from here?

- Hallway-Meeting Fri 0900–1200, Room TBA

alt-trans

- Increasing implementer pull for CoAP-over-TCP and CoAP-over-websockets
  - draft-bormann-core-coap-tcp-01
  - draft-savolainen-core-coap-websockets-02
- CoAP-over-SMS is in LWM2M, but somewhat underspecified
  - expired: draft-becker-core-coap-sms-gprs-04
- URI issues seem to settle
  - draft-silverajan-core-coap-alternative-transports-06
- Tackle this package now?

Wednesday

All times are in time-warped EDT

- **15:20–15:23 Intro I**
- **15:23–15:43 CoAP Congestion Control (CG)**
- **15:43–15:50 Intro II**
- **15:50–16:00 Block post WGLC (CB)**
- **16:00–16:15 HTTP mapping (SL)**
- **16:15–16:20 Links-JSON (chairs)**
- **16:20–16:25 Core-Interfaces (chairs)**
- **16:25–16:30 Resource Directory (chairs)**
- **16:30–16:35 comi, alt trans (chairs)**
- **16:35–16:45 No-Response (AB)**
- **16:45–16:50 Flextime**

TATA CONSULTANCY SERVICES
Experience certainty.

The No-Response option for CoAP

draft-tcs-coap-no-response-option-06

CoRE WG meeting @ IETF 90

Abhijan Bhattacharyya, Soma Bandyopadhyay, Arpan Pal
TCS Innovation Labs

From meeting #89, London

- Gained good interest
- Received traction from multicast
- One technically important comment received
  - How to handle token reuse

Major updates in draft in a nutshell

- Added a new section on token reuse (section 5.2 in the new draft)
  - Discussed in detail later
- Modified the section 3 (exemplary application scenario) to make it more compact
  - Incorporated an example of sending actuation commands through multicast with No-response
- Applicability to request methods is modified (Ref. Table 2)
  - GET request for cancellation of observe (‘observe’ with value ‘1’ – ‘deregister’)
    - Client may express disinterest in getting response from the server against a cancellation request
      - Only possible application along with GET identified so far
      - No-response with normal GET or GET with observe (start an observe session) request should have no effect of No-response, if present

Option properties for quick reference

Number	C	U	N	R	Name	Format	Length	Default
TBD		X	-		No-Response	uint	1	0

Option Properties

Value	Binary Representation	Description
0	00000000	Suppress all responses (same as empty value).
2	00000010	Allow 2.xx success responses.
8	00001000	Allow 4.xx client errors.
16	00010000	Allow 5.xx server errors.

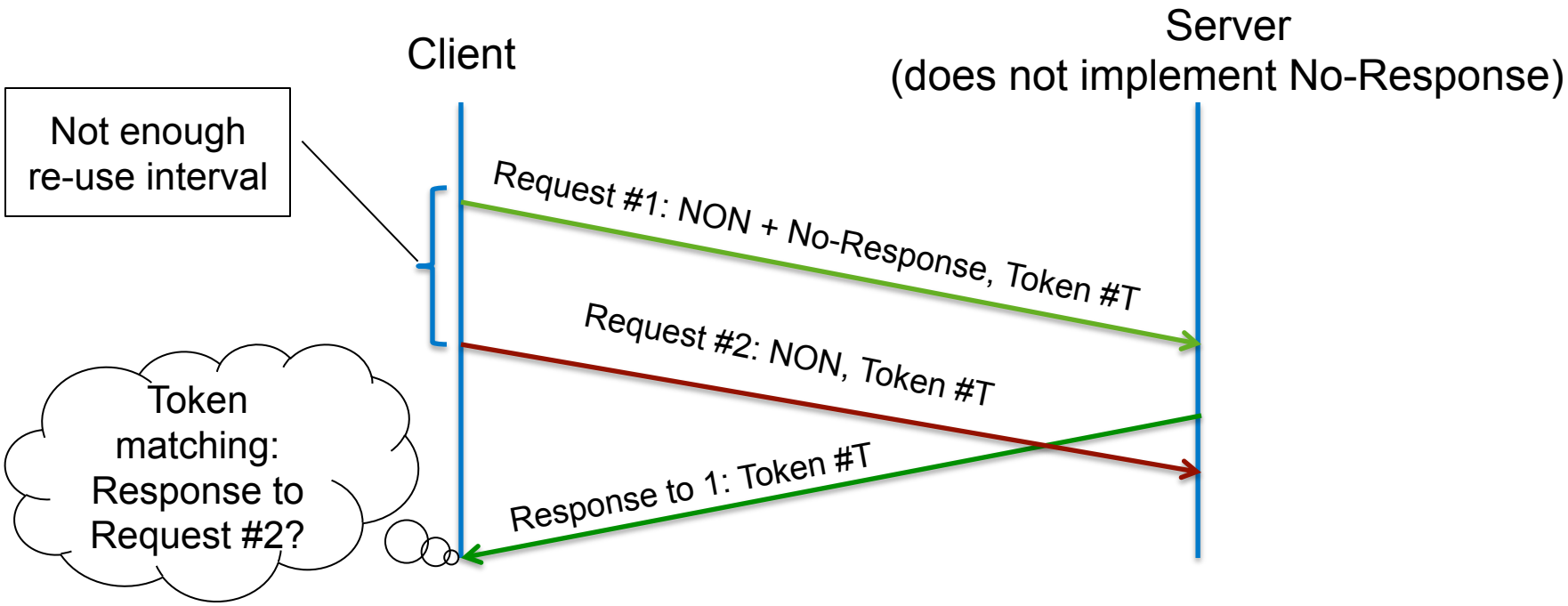
Option values

Token re-use : The problem

- Tokens provide a matching criteria between request and response
- Life of a token starts when it is assigned to a request and ends when the final matching response is received
  - Can be reused again
- NON request with No-Response has no response path – no token matching
  - Client has to decide a suitable time for retiring the token and reuse
- Problem is, the option is elective – server may not implement and actually send a response
  - Uncertainty about the server side response
    - True for granular suppression of response as well

Token re-use : The problem (contd.)

- Simple situation: Client is never going to care about any response coming back or about relating the response to the original request
  - MAY reuse the token value at liberty
- But what happens if we have something like this:



Token re-use: Solution approach

- Had quite a bit of discussion in the mailing list
- In most practical cases client should have an application specific ‘patience’ time till when it can re-use a token
  - Appendix-B.4.1 of draft-bormann-coap-misc defines ‘patience’ option which in effect puts a deadline to the server to respond back
  - However, ‘patience’ is not exposed to the protocol level at present
- A TOKEN_REUSE_TIME is defined similar to Section 2.5 of groupcomm draft

```
TOKEN_REUSE_TIME = NON_LIFETIME + MAX_SERVER_RESPONSE_DELAY + MAX_LATENCY
```

- Similar interpretation for multicast
- Unicast specific modifications:
  - MAX_SERVER_RESPONSE_DELAY : simply the expected maximum response delay from the server to which client sent the request
  - Includes ‘Leisure’ period (Section 8.2 of RFC 7252) but G = 1 for unicast.

Thank you

What next?

Wednesday

All times are in time-warped EDT

- **15:20–15:23 Intro I**
- **15:23–15:43 CoAP Congestion Control (CG)**
- **15:43–15:50 Intro II**
- **15:50–16:00 Block post WGLC (CB)**
- **16:00–16:15 HTTP mapping (SL)**
- **16:15–16:20 Links-JSON (chairs)**
- **16:20–16:25 Core-Interfaces (chairs)**
- **16:25–16:30 Resource Directory (chairs)**
- **16:30–16:35 comi, alt trans (chairs)**
- **16:35–16:45 No-Response (AB)**
- **16:45–16:50 Flextime**

Thursday

- 17:30–17:33 Intro
- 17:33–17:53 CoAP-MQ
- 17:53–18:15 other new work
- 18:15–18:30 Flextime

CoAP-MQ: Publish-Subscribe Extensions For CoAP

Extensions to enable a publish-subscribe interaction model between CoAP endpoints and CoAP services with asynchronous notifications and supporting sleeping and partially reachable endpoints

Use Cases and Design Overview

- Enable Publish-Subscribe model for interaction between CoAP endpoint and gateway or server
- Enable connection to a partially reachable endpoint, e.g. sleeping sensor or behind firewall
- Uses CoRE Resource Directory to register MQ endpoints
- Uses new core.mq server attribute and core.pubsub registration parameters
- Topics map to resource path
- CoAP-MQ Broker becomes the origin server
- Clear distinction between client and server endpoint roles

Documents Referred To In This Draft

- RFC 7252 (CoAP)
- CoRE Resource Directory
- Observing Resources in CoRE
- RFC 6690 (Core Link-Format)
- OMA LWM2M (queue mode)

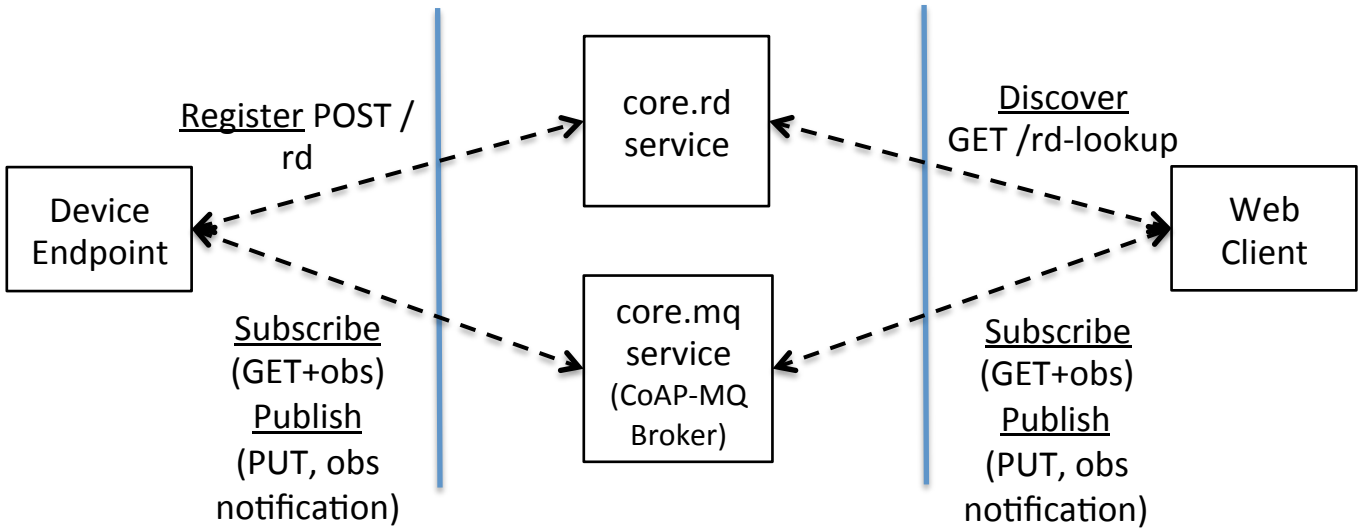
Terminology

- Topic – a string, similar to a path, that uniquely identifies an item or object being subscribed or published to
- CoAP-MQ Broker – A server node that stores information published to it, referenced by topic, and published said information to all subscribed entities
- Client role endpoint – an endpoint that is responsible for initiating all interactions, capable of sleeping
- Server role endpoint – an endpoint that listens for requests and responds to commands

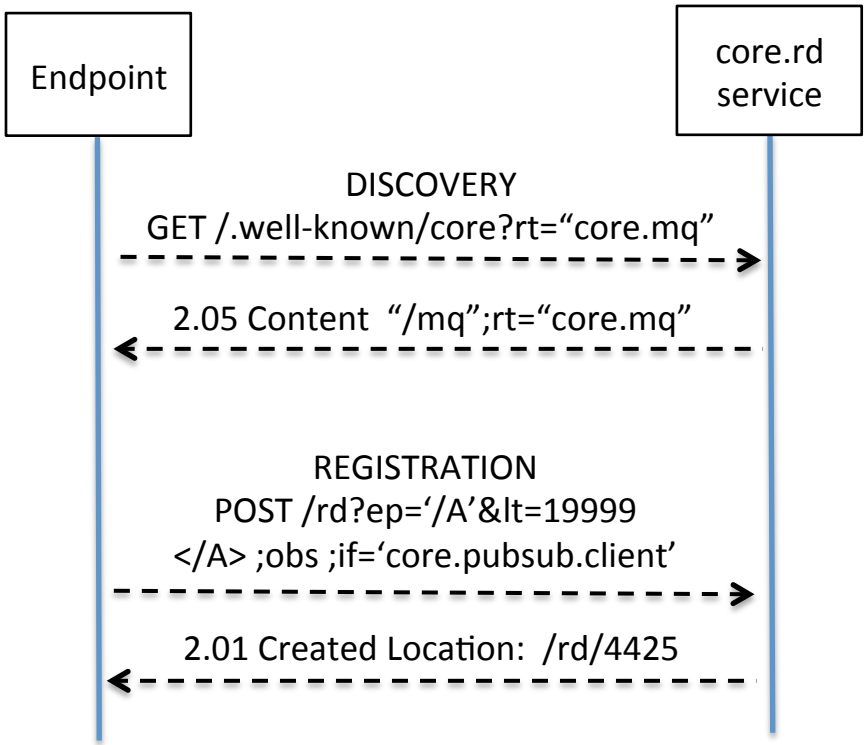
Identifiers and Attributes

- “core.mq” discoverable attribute of a service, indicates CoapMQ function set is supported
- “if=core.pubsub.client” indicates that interface to a resource is expected to support pubsub interaction model in a client role
- “if=core.pubsub.server” indicates that interface to a resource is expected to support pubsub interaction model in a server role
- “b=Q” LWM2M Transport Binding with Queue mode to support endpoint sleep/wakeup cycle with queueing

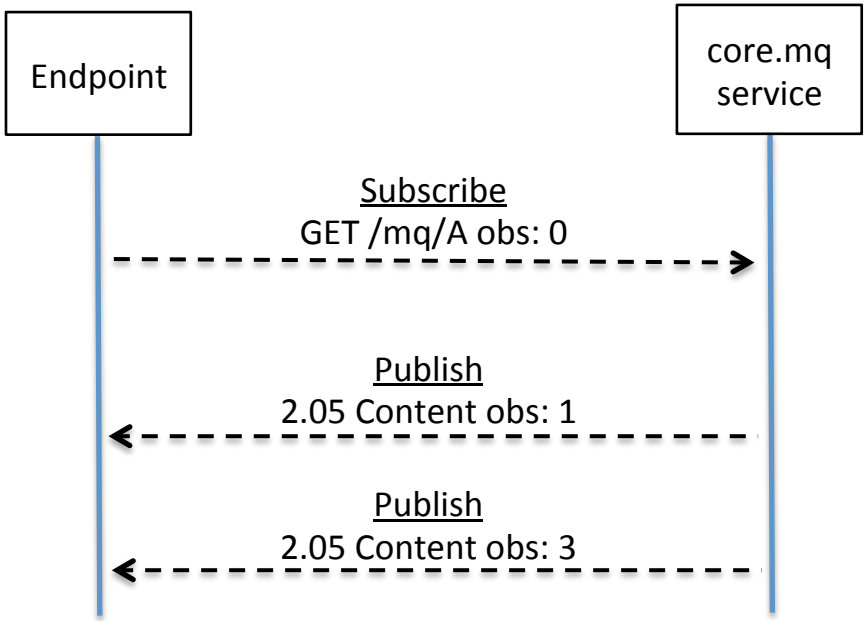
Architecture



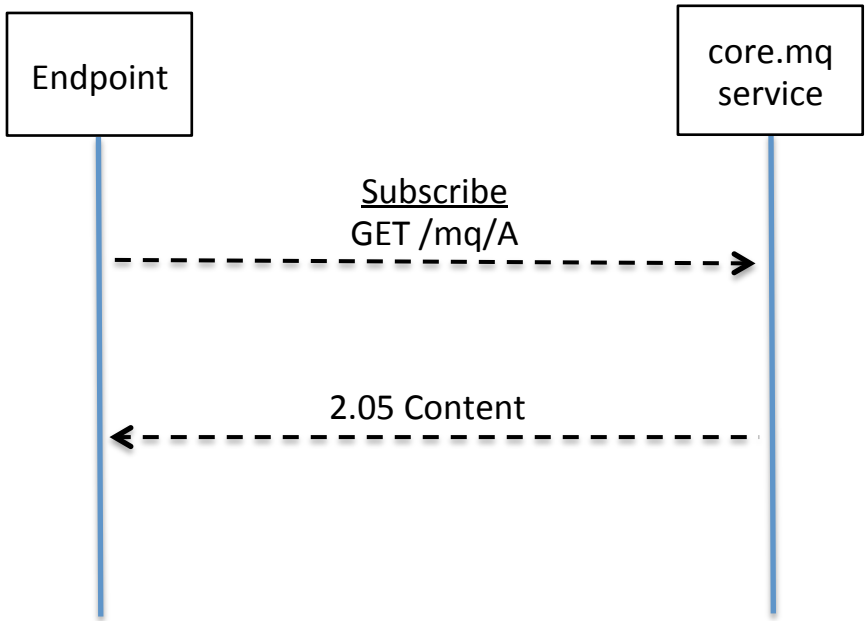
CoAP-MQ Broker Discovery and Registration Using Resource Directory Service



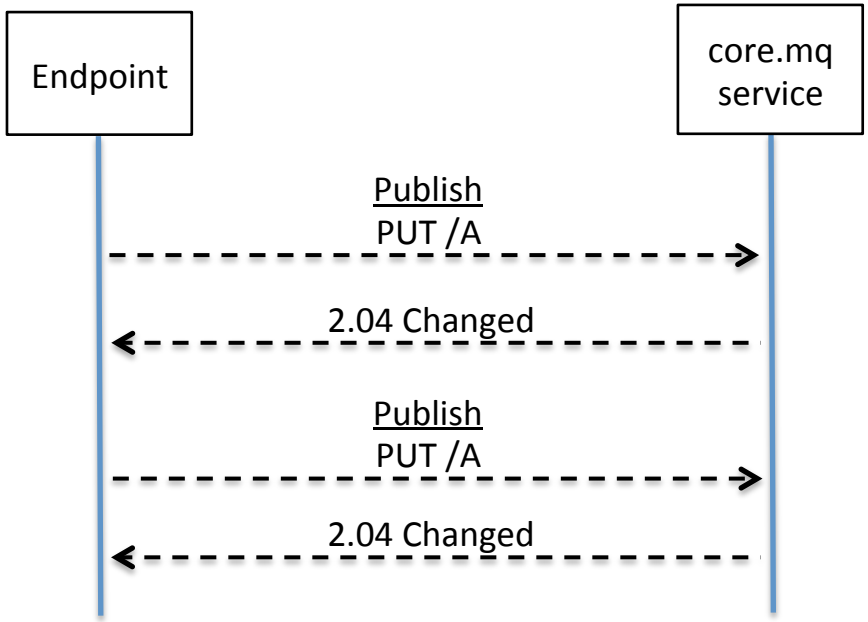
Client Endpoint Subscribes To CoAP-MQ Broker, CoAP-MQ Broker Publishes To EP



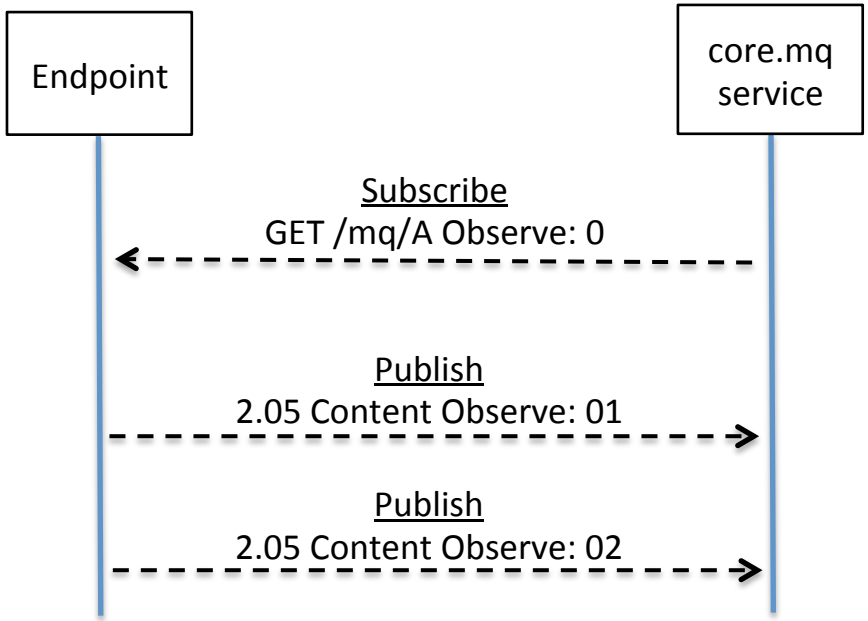
Client Endpoint Obtains Most Recent Representation of Topic Value: Like Retained



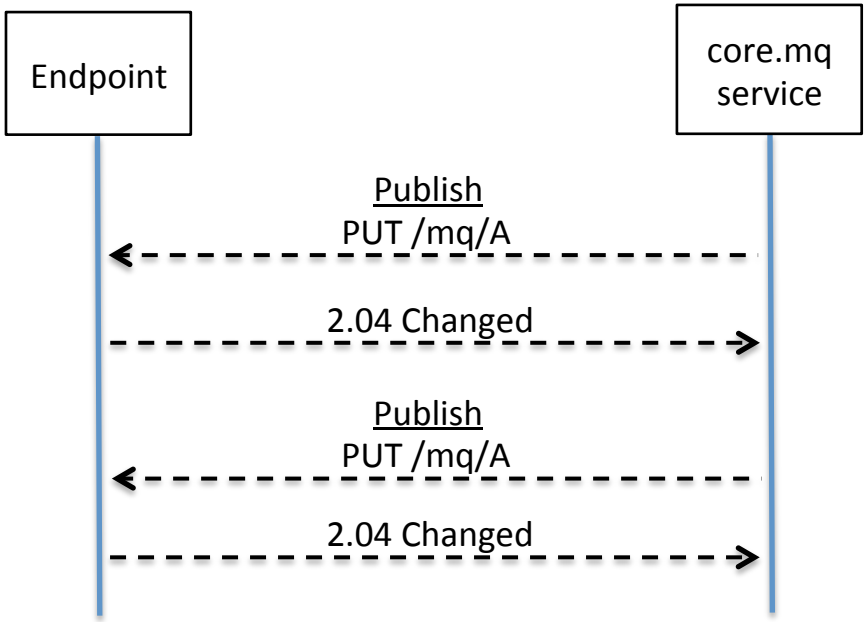
Client endpoint Publishes To CoAP-MQ Broker



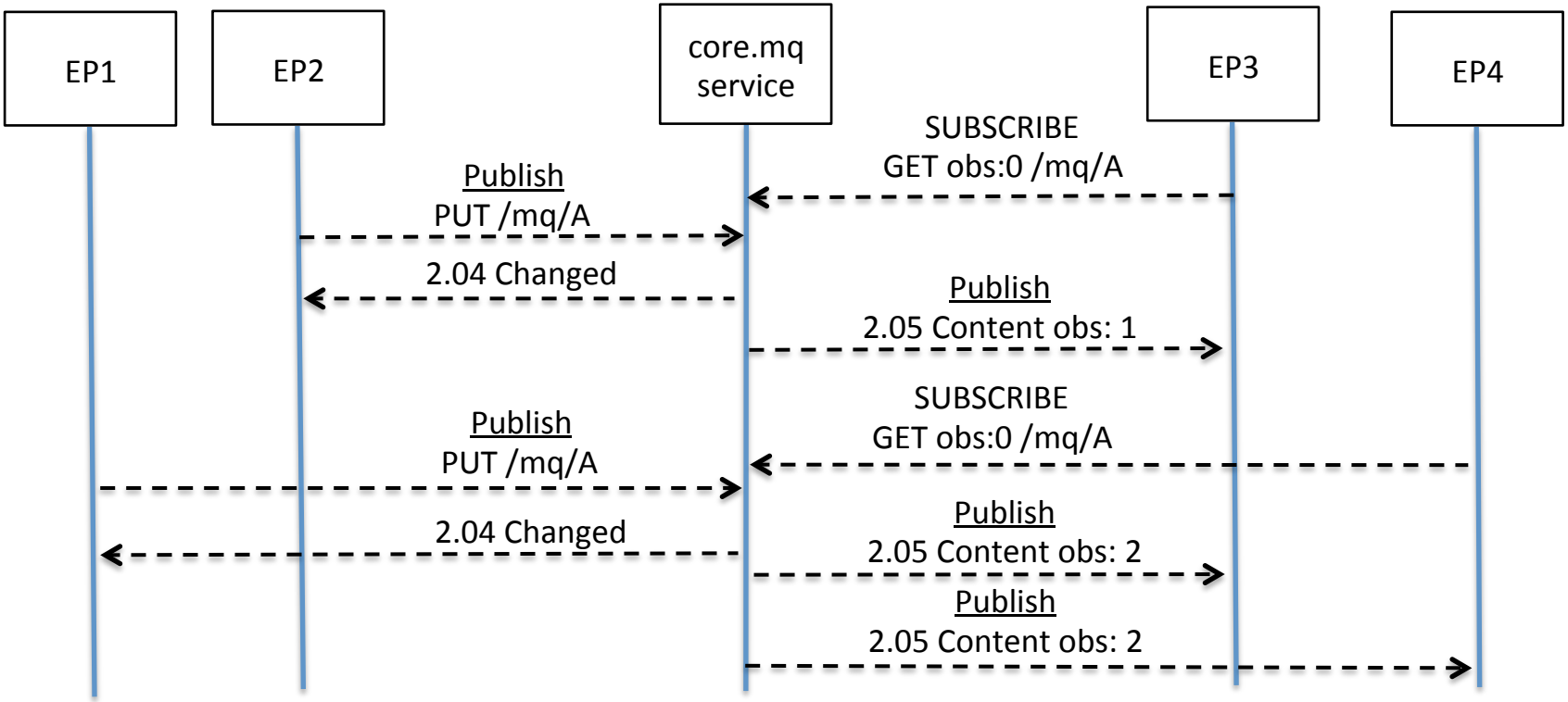
Server endpoint accepts subscriptions from CoAP-MQ broker



CoAP-MQ Broker Publishes to Server endpoint



Multiple Publishers and Subscribers(2)



Differences and Issues

- Topics are registered in addition to endpoints; need to use context parameter to point to topic location
- Nonexistent topic can not be subscribed to – does removal of last EP remove topic?
- Server role endpoints may only be single publishers
- Server EP subscribe mechanism is out of band, needs registration parameter or something
- How do endpoint roles interoperate on a topic?
- Rules for path construction and wildcards needed
- CoAP-MQ can act as a Pub-Sub to REST bridge

Thursday

- 17:30–17:33 Intro
- 17:33–17:53 CoAP-MQ
- 17:53–18:15 other new work
- 18:15–18:30 Flextime

“new work” (continued)

CoAP NodeId Option Extension

draft-li-core-coap-node-id-option-01 (***Kepeng Li***, Gengyu Wei)

draft-hong-core-coap-endpoint-unit-id-00 (Yong-Geun Hong,etc)

draft-kleine-core-coap-endpoint-id-00 (Oliver Kleine)

Motivation

- ✓The proposed “NodeId” option
  - ✓To identify the Node (Client or Server)
    - ✓ IP address is easily to be changed
    - ✓Useful to correlate request and response in case of IP address change
    - ✓Useful for alternative transport, e.g. SMS
    - ✓Useful for multicast and observe use cases
  - ✓To be used for authentication and authorization

NodeId Option

✓ Definition

Type	C	U	N	R	Name	Format	Length	Default
TBD	-	-	-	-	NodeId	string	1-255 B	(none)

Or

Type	C	U	N	R	Name	Format	Length	Default
124	E	U	-	-	ENDPOINT_ID_1	opaque	0-4 B	(none)
189	C	U	-	-	ENDPOINT_ID_2	opaque	0-4 B	(none)

Example Usage

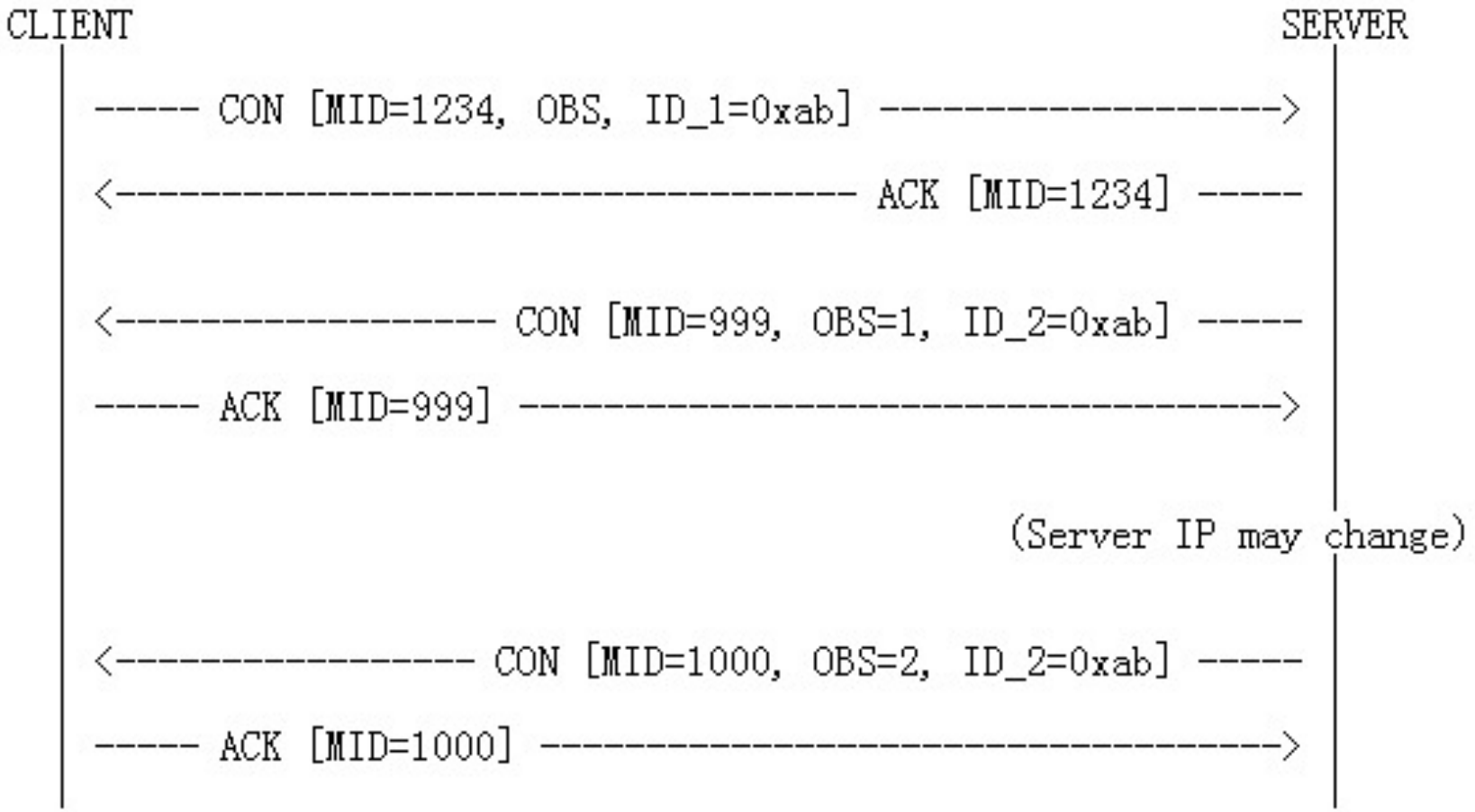


Figure 11: Server IP address changes during observation

Another Proposal: unit Id

- Usage: Identify resources.
- Benefit: Reduction in message transmission because it is shorter than Resource URI.

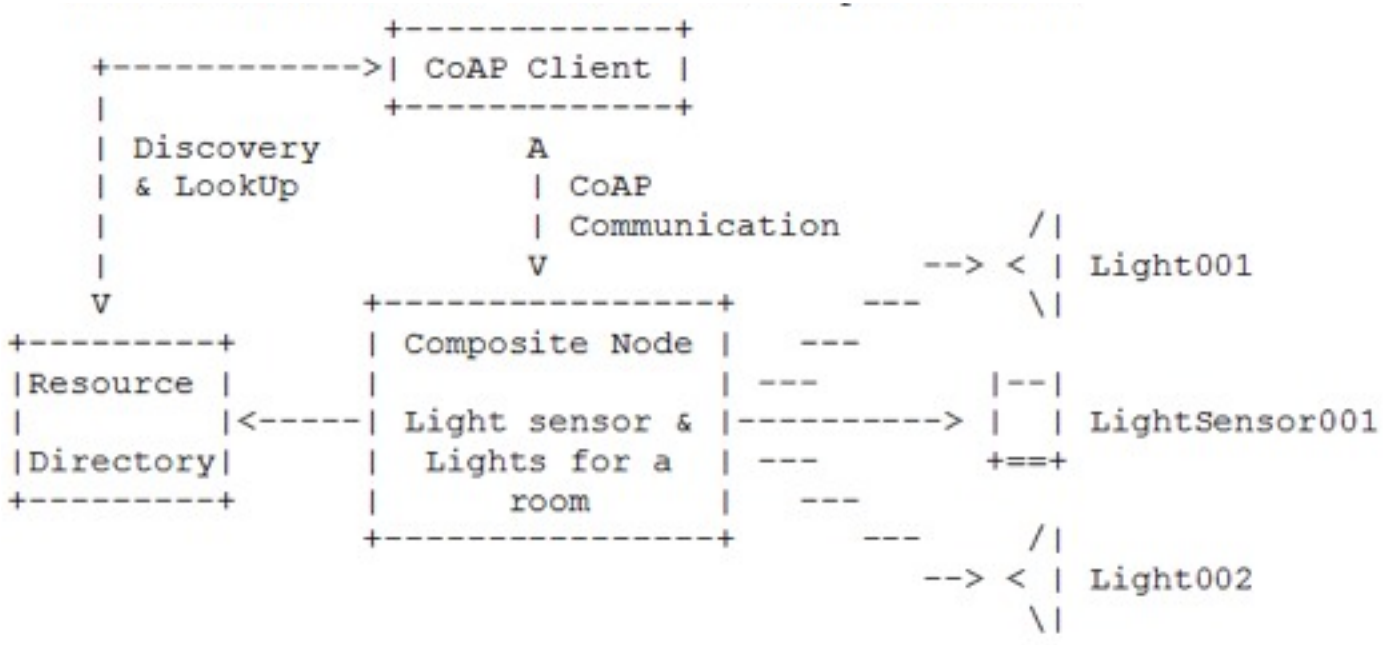


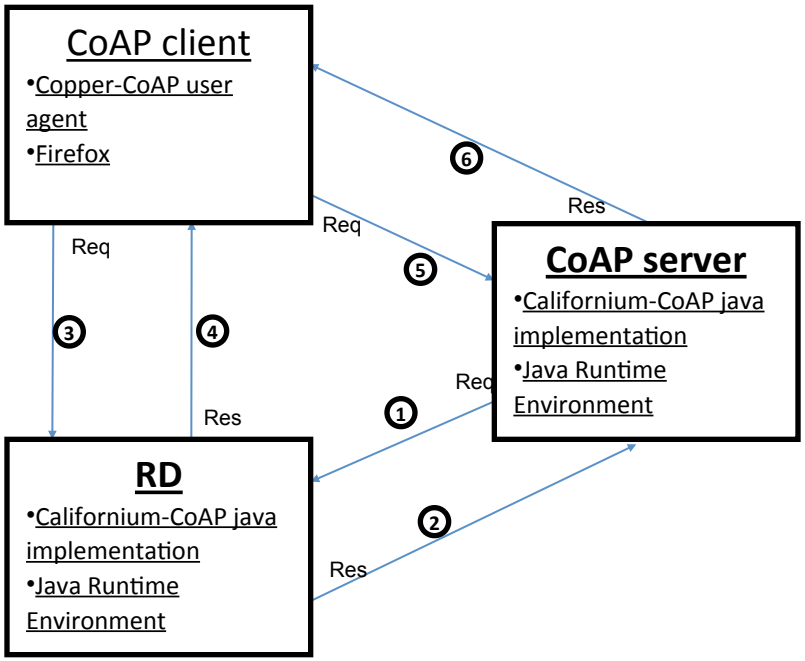
Figure 2: Multiple UnitID based composite CoAP node interaction use Case

Benefits of unit id

- Multiple resources can be uniquely identified by using just multiple unit IDs.
- Single CoAP message can be used to control multiple unit resources by using special characters in conjunction with multi-ID CoAP protocol.
- The reduction in message transmission results in reduced traffic and hence energy conservation in constrained resources.

Prototype Development Environment

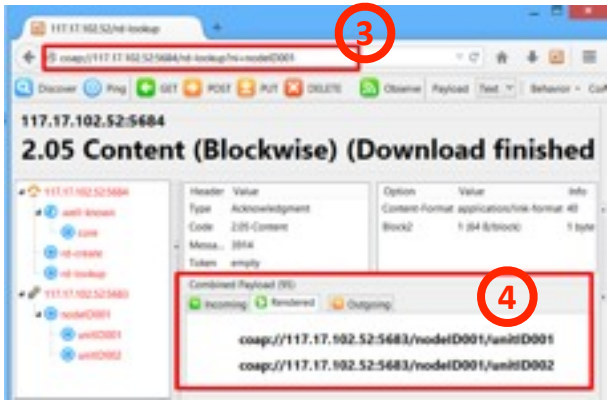
- Prototype development environment for CoAP endpoint unit identification

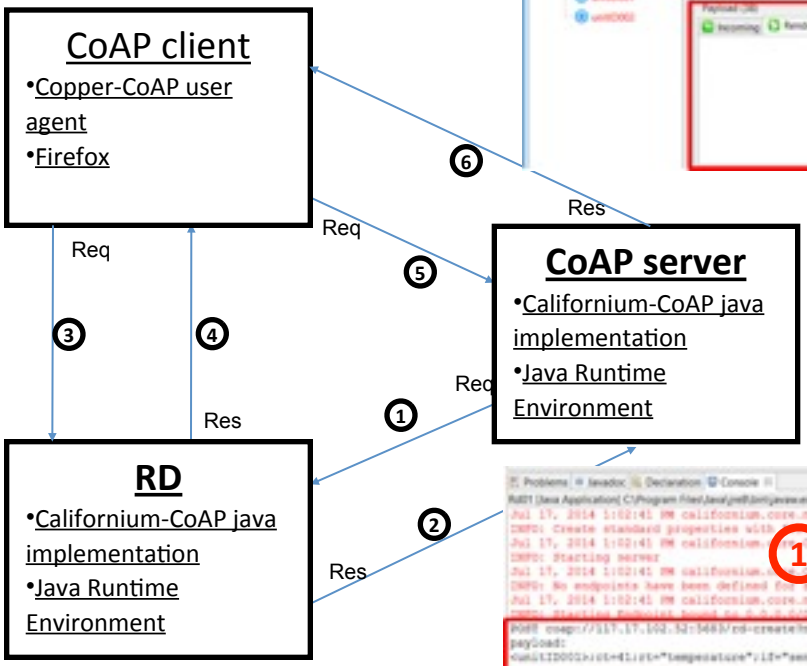


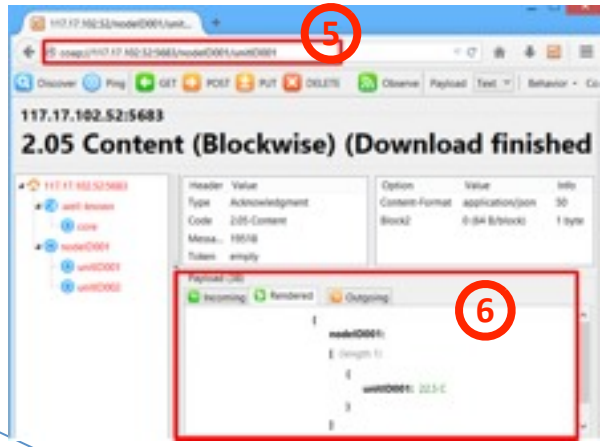
1. Req: POST coap://{rd-ip:port}/rd-create?ni=nodeID001
   Payload:
   <unitID001>;ct=41;rt="temperature";if="sensor",
   <unitID002>;ct=41;rt="luminance";if="sensor"
   ---
2. Res: 2.01 Created
   ==========
3. Req: GET coap://{rd-ip:port}/rd-lookup/ni=nodeID001
   ---
4. Res: 2.05 Content
   <coap://{node-ip:port}/nodeID001/unitID001>,
   <coap://{node-ip:port}/nodeID001/unitID002>
   ==========
5. Req: GET coap:// {node-ip:port}/nodeID001/unitID001
   ---
6. Res: 2.05 Content
   {nodeID001:[{"unitID001":"22.5 C"}]}
   ==========
5. Req: GET coap://{node-ip:port}/nodeID001?
   unitID001&unitID002
   ---
6. Res: 2.05 Content
   {"nodeID001":[{"unitID001":"22.5 C","unitID002":"1000 LUX"}]}
   ==========

Prototype Development Results

- Prototype development results for CoAP endpoint unit identification







Next Steps

- ✓ Socialize the idea with ACE WG about the usage in Authentication and Authorization
- ✓ Cover more usages in the draft, e.g. SMS transport
- ✓ Get implementation experience

CoAP Patience Option Extension

draft-li-core-coap-patience-option-04

Kepeng Li
Bert Greevenbosch
Esko Dijk
Salvatore Loreto

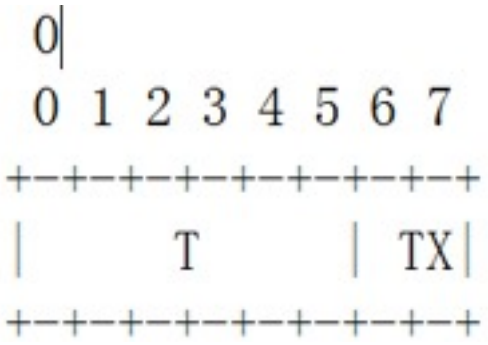
Motivation

- ✓Current “Leisure” defined in RFC7252
  - ✓The server SHOULD then pick a random point of time within the chosen leisure period to send back the unicast response to the multicast request.
- ✓The proposed “Patience” option
  - ✓In multicast case, the Patience option in a CoAP request can be used as an upper bound for the Leisure.
  - ✓In unicast case, used by a CoAP client to indicate the maximum time a requester is prepared to wait for a response.
  - ✓In observe case, the Patience Option MAY be used in a notification to indicate the maximum time an observer should wait before starting any observation relationship recovery.

Patience Option

✓ Definition

No.	C	U	N	R	Name	Format	Length	Default
28			x		Patience	see below	1 B	(none)



T = Time
TX = Time Exponent

Patience time = 2^(TX * 4 + 3) * T

Usage 1: in unicast request

- ✓Use case
  - Client sends a request to a server, but receives a response much later than expected. The response is discarded.
- ✓Proposal
  - Use Patience option to indicate the maximum time a requester is prepared to wait for a response.
- ✓Benefit
  - It can avoid that the recipient wastes resources by sending a response which already exceeds the set patience timeout.

Usage 2: in multicast request

- ✓Use case
  - ✓ Client sends multicast request to many servers, and receives many responses almost at the same time, which introduces congestion control issues.
- ✓Proposal
  - ✓Use Patience option in a request to indicate that the response SHOULD be replied with a dithered delay, i.e. a randomly chosen delay between 0 and the time indicated in the option.
- ✓Benefit
  - ✓Helps avoiding congestion i.e. multicast response storms in constrained networks.

Usage 3: in observe notification

- ✓Use case:
  - ✓ When the Max-Age of the observed resource state expires, client issues a new GET request to refresh observation relationship. This may introduce congestion issues.
- ✓Proposal:
  - ✓Server to use Patience option to indicate that, after the period of time in the Max-Age option has expired, a new notification will be sent within the time interval.
- ✓Benefit:
  - ✓Maintain a robust observation relationship; avoid network congestion issues.

Next Steps

- ✓ Remove multicast usage and observe usage?
  - ✓ Keep only one semantics for unicast usage
  - ✓ Make this option simpler
- ✓ Improve the algorithm to calculate the patience time

Thursday

- 17:30–17:33 Intro
- 17:33–17:53 CoAP-MQ
- 17:53–18:15 other new work
- 18:15–18:30 Flextime

Flextime