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

http://www.ietf.org/about/note-well.html

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.

Agenda Bashing

All times are in time-warped JST

Tuesday

- 09:00–09:10 Intro, WG status
- 09:10–09:25 CoAP over reliable (chairs)
- 09:25–09:30 HTTP Mapping (chairs)
- 09:30–09:50 Resource Directory (MK)
- 09:50–10:05 CoRE Interfaces (MK)
- 10:05–10:35 SenML (AK)
- 10:35–10:55 COMI (PV)
- 10:55–11:10 PATCH (PV)
- 11:20–11:20 FETCH (CB), COOL (AP)
- 11:20–11:30 Pubsub (MK)

All times are in time-warped JST

Friday

- 09:00–09:05 Intro
- 09:05–09:25 Delay Attacks (JM)
- 09:25–10:00 Object Security (GS)
- 10:00–10:30 CoCoA (CG, IJ)
- 10:30–10:40 Mostly-offline nodes (PV)
- 10:40–11:30 Flextime

Observe ✓

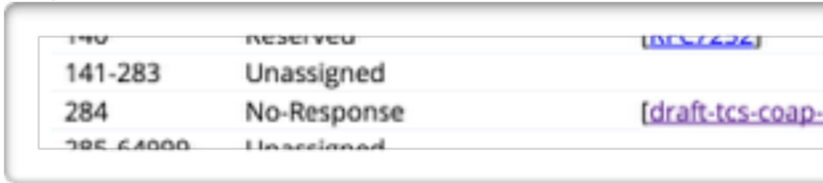
RFC 7641

WG documents

- draft-ietf-core-block — 3rd WGLC completed
  - data tracker confusion… Now in AD to clear
- draft-ietf-core-http-mapping
  - WGLC very soon
- draft-ietf-core-links-json
  - merged with draft-li-core-cbor-equivalents
- draft-ietf-core-resource-directory
  - charter work needed (today), added authors
- draft-ietf-core-interfaces
  - to resume activity!

Option 284: No-Response

- Started out as a contribution to CoRE
- Received considerable WG review
- Now a registered option in Specification Required space



140	Reserved	[RFC7252]
141-283	Unassigned	
284	No-Response	[draft-tcs-coap-
285-64999	Unassigned	

- Points to draft-tcs-coap-no-response-option-13
- Plan: make this an RFC via ISE submission
- Review from WG experts is still useful

All times are in time-warped JST

Tuesday

- **09:00–09:10 Intro, WG status**
- **09:10–09:25 CoAP over reliable (chairs)**
- **09:25–09:30 HTTP Mapping (chairs)**
- **09:30–09:50 Resource Directory (MK)**
- **09:50–10:05 CoRE Interfaces (MK)**
- **10:05–10:35 SenML (AK)**
- **10:35–10:55 COMI (PV)**
- **10:55–11:10 PATCH (PV)**
- **11:20–11:20 FETCH (CB), COOL (AP)**
- **11:20–11:30 Pubsub (MK)**

A TCP and TLS Transport for the Constrained Application Protocol (CoAP)

Carsen Borman

Simon Lemay

Hannes Tschofenig

Recap From and since last IETF

- Most of the Confusion around CON/NON was resolved
- Proposed 3 different length delimiter approach
- Explored the different problems if using a UDP/TCP proxy
- Attempt to have a discussion about the length delimiter online
  - Cancel due to lack of attendence
- Thank you for all who expressed opinions or concerns

Length delimiter

- L2 can safely be removed from the choices
- L1 implemented by 2 projects
- L3 implemented by 1 project
- Others?

- NEED TO MAKE A CHOICE

Pending work on L1 (if chosen)

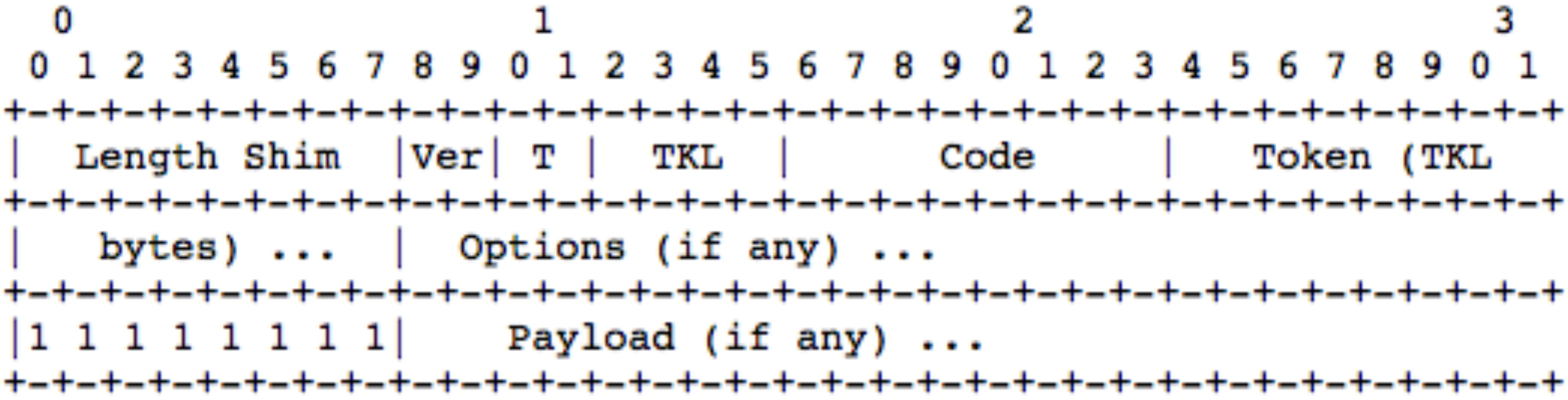
- If L1 is chosen, if delimiter is 2 bytes, we need a protocol specific block size)
  - (does not make sense that any packet larger than 0xFFFF would then be chunk into 1024byte packet)

Proxy (open issue)

- Need an implementation to better understand the consequences of the choices we are making.
- Draft need guideline on how to handle the case

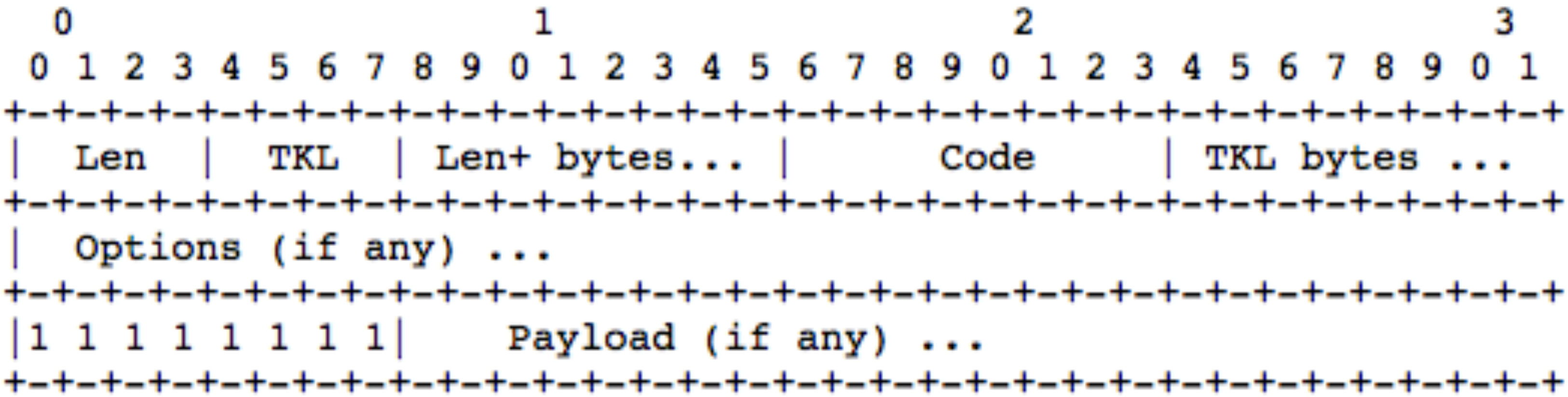
Call for adoption!

Shim



```
  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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Length Shim  |Ver| T |  TKL  |      Code     |   Token (TKL
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   bytes) ...  |  Options (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 1 1 1 1 1|    Payload (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Like



```
  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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Len  |  TKL  | Len+ bytes... |      Code     | TKL bytes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Options (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 1 1 1 1 1|    Payload (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

All times are in time-warped JST

Tuesday

- **09:00–09:10 Intro, WG status**
- **09:10–09:25 CoAP over reliable (chairs)**
- **09:25–09:30 HTTP Mapping (chairs)**
- **09:30–09:50 Resource Directory (MK)**
- **09:50–10:05 CoRE Interfaces (MK)**
- **10:05–10:35 SenML (AK)**
- **10:35–10:55 COMI (PV)**
- **10:55–11:10 PATCH (PV)**
- **11:20–11:20 FETCH (CB), COOL (AP)**
- **11:20–11:30 Pubsub (MK)**

All times are in time-warped JST

Tuesday

- **09:00–09:10 Intro, WG status**
- **09:10–09:25 CoAP over reliable (chairs)**
- **09:25–09:30 HTTP Mapping (chairs)**
- **09:30–09:50 Resource Directory (MK)**
- **09:50–10:05 CoRE Interfaces (MK)**
- **10:05–10:35 SenML (AK)**
- **10:35–10:55 COMI (PV)**
- **10:55–11:10 PATCH (PV)**
- **11:20–11:20 FETCH (CB), COOL (AP)**
- **11:20–11:30 Pubsub (MK)**

CoRE Resource Directory

draft-ietf-core-resource-directory-05

Overview

- CoRE Resource Directory enables a service based registration and discovery model for endpoint devices and resources exposed by network services
- CoRE RD is used by OMA LWM2M in the Registration Interface

Updates

- Added explicit HTTP mapping and examples
- Added support for updating links in the collection using PATCH
  - Using query filtering to function as link selector for PATCH target
  - Enables adding or removing links, and adding, updating, or removing link attributes
- Uses RFC 7396 JSON Merge Patch on selected links – assumes a JSON Links representation

Open Issues, Questions

- Dependence on JSON Link-Format for model of the document to patch using RFC 7396
- Status of the work, what is left to do?

All times are in time-warped JST

Tuesday

- 09:00–09:10 Intro, WG status
- 09:10–09:25 CoAP over reliable (chairs)
- 09:25–09:30 HTTP Mapping (chairs)
- 09:30–09:50 Resource Directory (MK)
- 09:50–10:05 CoRE Interfaces (MK)
- 10:05–10:35 SenML (AK)
- 10:35–10:55 COMI (PV)
- 10:55–11:10 PATCH (PV)
- 11:20–11:20 FETCH (CB), COOL (AP)
- 11:20–11:30 Pubsub (MK)

Reusable Interface Definitions for Constrained RESTful Environments

draft-ietf-core-interfaces-04

Overview

- Defines Interface Types to be used with the “if” link attribute to describe how to interact with a resource - methods, content formats, resource behavior
- Defines Observe Attributes pmin, pmax, st, lt, gt which are set on a resource using query parameters
- Defines “link bindings” to resources which synchronize the state of resources in different endpoints, through the exchange of resource representations
  - A link binding implements the client role and associates a source resource with a destination resource
  - Polling, Observe, and Push type bindings
  - Bindings are configured with observe attributes
- Defines some interface types for simple machine interactions
  - sensor, actuator, batch, link list, linked batch, parameter, and binding
- Link list, batch, and linked batch are Collection types

Updates

- Ticket 385 – Change POST from toggle to POST
- Ticket 386 - `%s/content-type/content-format/g`
- New abstract and intro, more descriptive
- Reorganized order of sections
- Add a description section for collections
- Reference to the collection types in this document in OIC Sec. 7 – Resource Model
- Added a new Hypermedia Collection type

Hypermedia Collection

- Generalization of ll, b, lb types
- Adds group update and actuation
- Adds dynamic item creation with link
- Enables hypermedia based resource structure defined by links instead of URI paths
- Content format selects representation of links (ex. link-format), items(ex. SenML), or collection(new content format)
- Embedding link processing controls are defined for bulk read, batch, and group updates

Collection Representation

```
{
  "bn":"/sen/temp/1/",
  "e":[
    {"n":"currentValue","v":"31.3","u":"C"},
    {"n":"maxValue","v":"37.1","u":"C"},
    {"n":"minValue","v":"18.3","u":"C"},
    {"n":"resetMaxMin"},
    {"n":"minScale","v":"0","u":"C"},
    {"n":"maxScale","v":"100","u":"C"},
    {"n":"appType","sv":"Inboard Bearing”}
  ],
  "l":[
    {"href":"","rel":"self","rt":"temperature","u":"C"},
    {"href":"currentValue","rt":"currentValue","u":"C"},
    {"href":"maxValue","rt":"maxValue","u":"C"},
    {"href":"minValue","rt":"minValue","u":"C"},
    {"href":"resetMaxMin","rt":"resetMaxMin"},
    {"href":"minScale","rt":"minScale","u":"C"},
    {"href":"maxScale","rt":"maxScale","u":"C"},
    {"href":"appType","rt":"appType”}
  ]
}
```

Hypermedia Interaction Example

```
GET /light/?rt=brightness content-format=22003 <= links
[{"href":"level/","rt":["thing","brightness"],"ct":
[22001,22002,22003]}]

GET /light/level/?rt=actions content-format=22003
[{"href":"actions/","rt":"actions","ct":[22004]}]

GET /light/level/actions/?rt=change content-format=22004 <= form
[{"href":"../actuations","rel":"action","rt":"change","method":"post",
       "ct":22001,"params":[{"name":"targetValue","range":"0-100"},
                            {"name":"transitionTime"},"units":"s"],
                "template":"{"e":[{"n":"change","v":"$targetValue"},

{"n":"ttime","v":"$transitionTime"}]}”}]

POST /light/level/actuations content-format=22001 <= collection
 {"e":[{"n":"change","v":"50"},{"n":"ttime","v":"10"}]}

201 Created Location:/light/level/actuations/19934577

DELETE /light/level/actuations/19934577
```

Open Issues

- Should this be moved to standards track?
  - Interface types are referenced by OIC and other work in progress
  - Observation attributes are used in OMA LWM2M
  - Collection pattern is used in OIC, IPSO, LWM2M

All times are in time-warped JST

Tuesday

- 09:00–09:10 Intro, WG status
- 09:10–09:25 CoAP over reliable (chairs)
- 09:25–09:30 HTTP Mapping (chairs)
- 09:30–09:50 Resource Directory (MK)
- 09:50–10:05 CoRE Interfaces (MK)
- 10:05–10:35 SenML (AK)
- 10:35–10:55 COMI (PV)
- 10:55–11:10 PATCH (PV)
- 11:20–11:20 FETCH (CB), COOL (AP)
- 11:20–11:30 Pubsub (MK)

Media Types for Sensor Markup Language (SenML)

draft-jennings-core-senml-02

IETF 94, Yokohama, Japan
November 3rd, 2015

Ari Keränen
ari.keranen@ericsson.com

Valid characters in JSON (UTF-8 vs. ASCII)

- Currently: only ASCII text allowed
- Proposal: “UTF-8 without Surrogates or Noncharacters” [I-JSON: RFC7493]
  - exclude all control characters?
- Should exclude quote (“) from string values for simpler processing?
- Name attribute
  - Currently: only URI characters

Base value for value

- Currently: base name, unit, and time
- Proposal: add base/default value
  - add (sum) numeric values
  - append string values
  - default value for rest (Boolean & binary)

- Use case: set of many measurements with exact same or only slightly different value

Location of base values

- No fixed order for members (name/value pairs) in JSON object
- When parsing, don't know full name/time/units before base values reached
  - Need full structure to memory or parse it twice; can't do stream processing
  - (for example) block transfer: may not have the end or full structure easily accessible

Location of base values

- Current solution: array root and base first

```
[{
   "bn": "urn:dev:mac:0024befffe804ff1/",
   "bt": 1276020076,
   "bu": "A"
 },
 [{ "n": "voltage", "u": "V", "v": 120.1 },
  { "n": "current", "t": -3, "v": 0.14e1 },
  { "n": "current", "t": -2, "v": 1.5 },
  { "n": "current", "t": -1, "v": 1.6 },
  { "n": "current", "t": 0, "v": 1.7 }]
]
```

Multiple bases

```
[{
   "bn": "urn:dev:mac:0024befffe804ff1/voltage",
   "bt": 1276020076,
   "bu": "V"
 },
   [{"v": 120.1 },
     ...
   ],
 {"bn": "urn:dev:mac:0024befffe804ff1/current",
   "bu": "A" },
   [{"t": -3, "v": 0.14e1 },
     {"t": -2, "v": 1.5 },
     {"t": -1, "v": 1.6 },
     {"t": 0, "v": 1.7 }]
]
```

JSON Merge Patch format
RFC 7396

JSON syntax proposal

- Currently: nested array for measurements
- Proposal: all measurement entries in the root array. Base value objects mixed in between.

```
[
 {"bn": "urn:dev:mac:0024befffe804ff1/current",
  "bt": 1276020076, "bu": "A"},
  {  "t": -1, "v": 1.6 },
  {  "t": 0,  "v": 1.7 },
 {"bn": "urn:dev:mac:0024befffe804ff1/voltage", "bu":"V”},
  {  "t": -2, "v": 120.05 },
  {  "t": -1, "v": 120.1 }
]
```

no embedded array

New value type: binary blob

- Currently: only numeric, string, and boolean
- Proposal: “binary blob” data type
  - e.g., base64url encoded string in JSON and XML
  - CBOR representation supports binary

- Do we need string type anymore?
  - Or do we need this while we have string?

All times are in time-warped JST

Tuesday

- **09:00–09:10 Intro, WG status**
- **09:10–09:25 CoAP over reliable (chairs)**
- **09:25–09:30 HTTP Mapping (chairs)**
- **09:30–09:50 Resource Directory (MK)**
- **09:50–10:05 CoRE Interfaces (MK)**
- **10:05–10:35 SenML (AK)**
- **10:35–10:55 COMI (PV)**
- **10:55–11:10 PATCH (PV)**
- **11:20–11:20 FETCH (CB), COOL (AP)**
- **11:20–11:30 Pubsub (MK)**

CoRE working group

CoAP Management Interface
draft-vanderstok-core-comi-08

P. van der Stok, A. Bierman, J. Schoenwalder, A. Sehgal

State with respect to version 7

Thanks to Michel Veillette and Alexander Pelov

Current version 8

- Use module name instead of prefix in hash calculation
- Rehashed object hashes have bit 31 set to 1
- YANG lists transported as CBOR map of maps
- CBOR content examples added in appendix
- Appendices on hash clash probability and server storage overhead

Rehash error return, description

Hash collision occurs when two names have the same hash in a given server

With 30.000 names in one server probability about 10%

The conflicting names have to be rehashed in the server.

When name is rehashed, bit 31 of hash is set to one by server.

This alerts the client to identify to object and module.

Client may set bit 31 to one for rehashed names (not necessary)

YANG lists

Yang lists may contain keys which uniquely identify list instances.
Transport only selected list instances.
Identify the instances with their key values.

```
REQ: PATCH example.com/mg/interfaces/interface/ipv6/neighbor
  { "neighbor" :{
          {"ip" : "fe80::200:f8ff:fe21:67cf"}:
          {"link-layer-address" : "00:00::10:01:23:45"}
   } }
```

Key

Value

No standard JSON, but CBOR diagnostic notation corresponding with legal CBOR

RPC addition

PRO:

- RPC is part of YANG
- Modelling: Control uses action models instead of data models
- Use of IN and OUT parameters (request sets and returns data)

CONTRA:

- Not REST

WG opinion?

Way forward

No further issues identified, apart CoOL extension: more features, and numbering

- Balanced functionality?
- WG acceptance?

Document split

- CoMI function set specification
- Hashing specification
- CBOR content format

WG opinion?

- 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://www.ietf.org/about/note-well.html

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.

All times are in time-warped JST

Friday

- **09:00–09:05 Intro**
- **09:05–09:25 Delay Attacks (JM)**
- **09:25–10:00 Object Security (GS)**
- **10:00–10:30 CoCoA (CG, IJ)**
- **10:30–10:40 Mostly-offline nodes (PV)**
- **10:40–11:30 Flextime**

Insert Tuesday here

All times are in time-warped JST

~~Tuesday~~ Friday

- 09:00–09:10 Intro, WG status
- 09:10–09:25 CoAP over reliable (chairs)
- 09:25–09:30 HTTP Mapping (chairs)
- 09:30–09:50 Resource Directory (MK)
- 09:50–10:05 CoRE Interfaces (MK)
- 10:05–10:35 SenML (AK)
- 10:35–10:55 COMI (PV)
- 10:55–11:10 PATCH (PV)
- 11:20–11:20 FETCH (CB), COOL (AP)
- 11:20–11:30 Pubsub (MK)

All times are in time-warped JST

Friday

- **09:00–09:05 Intro**
- **09:05–09:25 Delay Attacks (JM)**
- **09:25–10:00 Object Security (GS)**
- **10:00–10:30 CoCoA (CG, IJ)**
- **10:30–10:40 Mostly-offline nodes (PV)**
- **10:40–11:30 Flextime**

SECURITY FOR COAP (ACTUATORS)

DRAFT-MATTSSON-CORE-COAP-ACTUATORS
MATTSSON, FORNEHED, SELANDER, PALOMBINI
IETF94 CORE, NOV 6 2015

INTRODUCTION

- Analysis of whether CoAP has all properties needed to securely control actuators, or if something is missing. Findings:
  - Guidelines to implementers needed.
  - New mechanism needed to mitigate delay attacks.

- Somewhat surprisingly we also discovered that while HTTPS is secure, CoAP over DTLS (or IPsec) is vulnerable to mismatch attacks where the client matches the wrong response to the request.
  - This is not expected, has serious consequences, and must be fixed.
  - Luckily, very easy to fix on the CoAP layer.

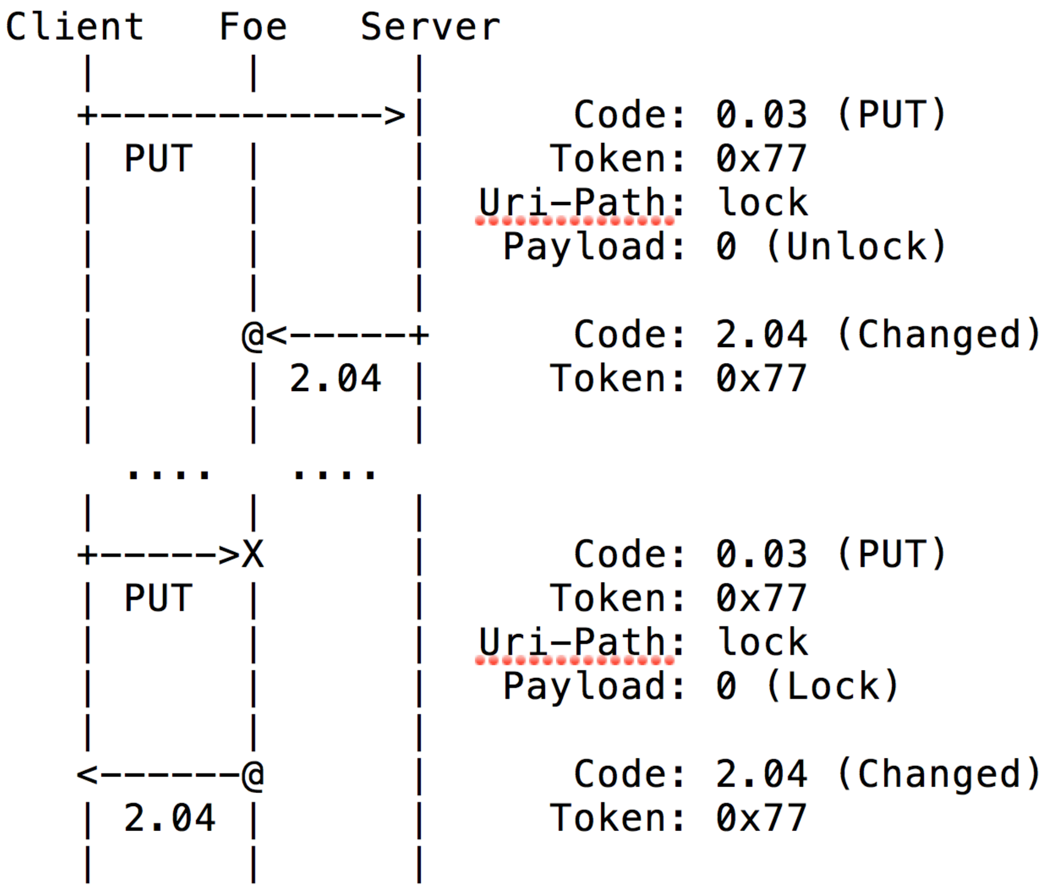
RESPONSE DELAY AND MISMATCH ATTACK

Assumptions:

- Attacks are performed by an on-path attacker, not a “trusted” middle box.
- CoAP is protected with a security protocol such as DTLS or IPsec.

MISMATCH ATTACK (1/3)

- Possible if CoAP is protected with DTLS or IPsec.
- The client needs to reuse the same token.
- If encryption is used the attacker cannot see the token. Attacker must use implementation knowledge or guess.
- The attacker can always control the replay window.
- Example: Attacker can e.g. fool client to believe a door has been locked.



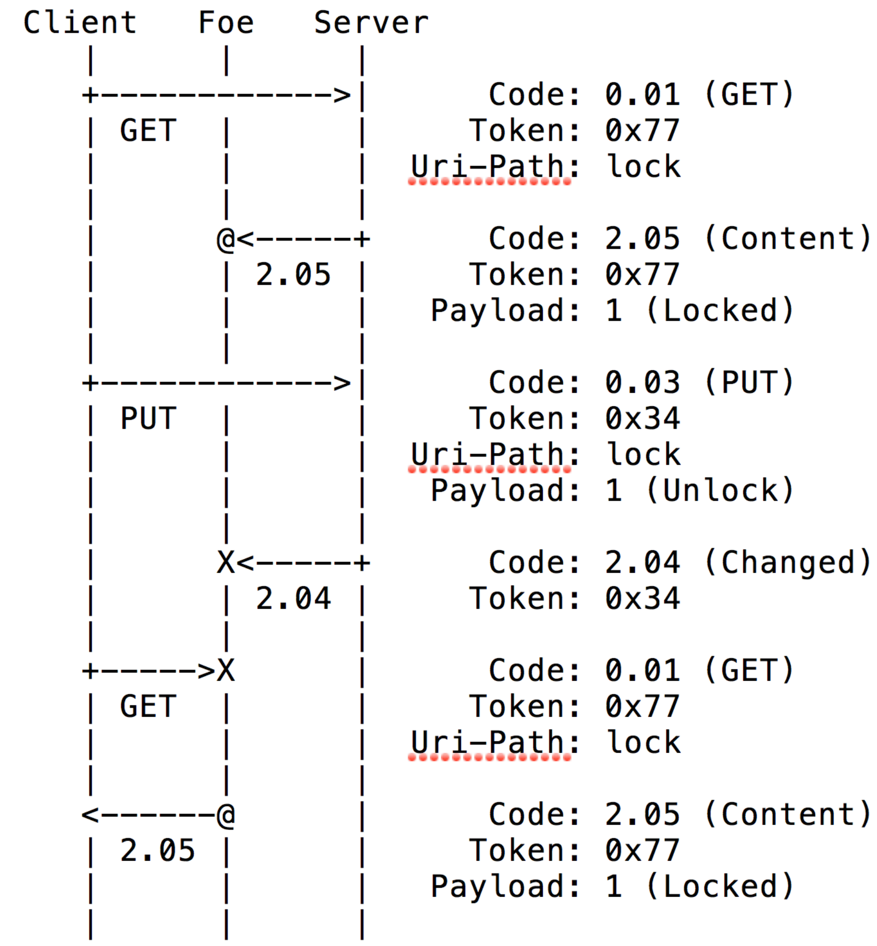
```
Client   Foe   Server
   |      |      |
   +------------->|      Code: 0.03 (PUT)
   | PUT  |      |     Token: 0x77
   |      |      |  Uri-Path: lock
   |      |      |   Payload: 0 (Unlock)
   |      |      |
   |      @<-----+      Code: 2.04 (Changed)
   |      | 2.04 |     Token: 0x77
   |      |      |
     ....   ....
   |      |      |
   +----->X      |      Code: 0.03 (PUT)
   | PUT  |      |     Token: 0x77
   |      |      |  Uri-Path: lock
   |      |      |   Payload: 0 (Lock)
   |      |      |
   <------@      |      Code: 2.04 (Changed)
   | 2.04 |      |     Token: 0x77
   |      |      |
```

Figure 5: Mismatching Response to PUT

MISMATCH ATTACK (2/3)

- Attack is also valid for sensors. Also here with serious consequences.

- Example: Attacker can e.g. fool client to believe a door is locked.

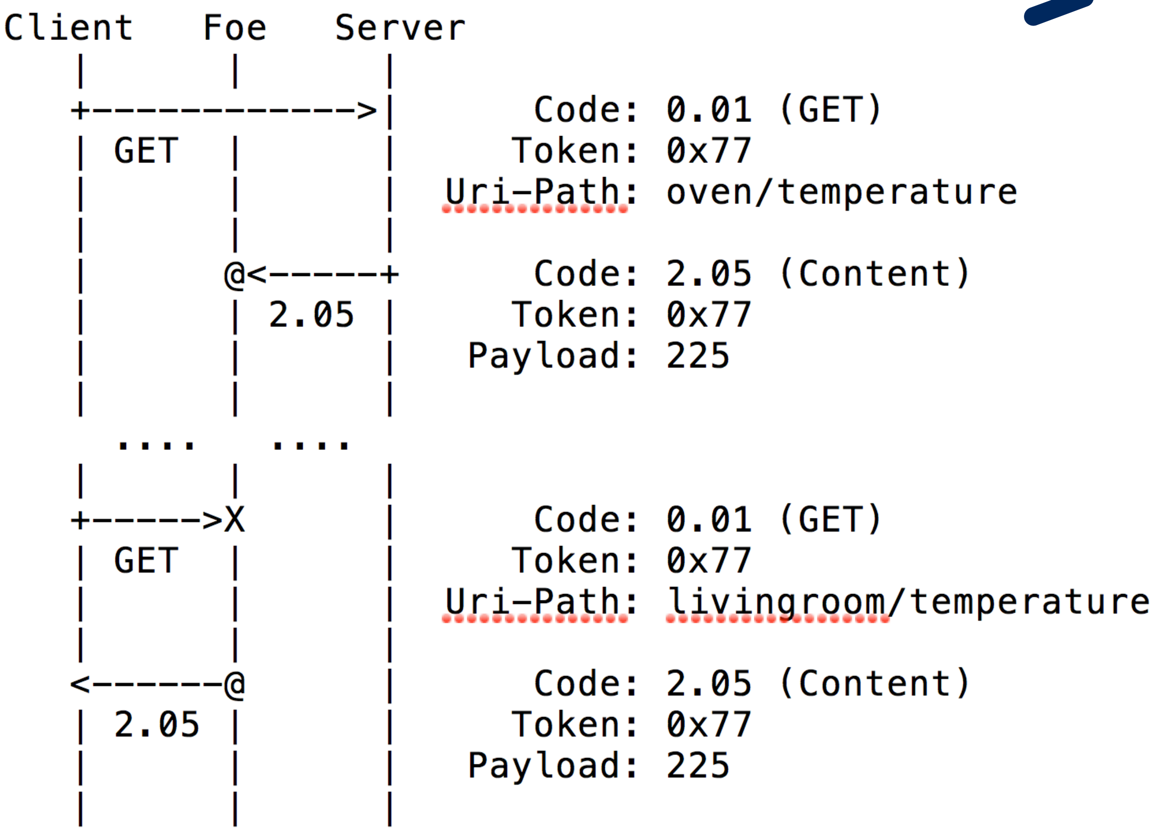


```
 Client   Foe   Server
    |      |      |
    +------------>|      Code: 0.01 (GET)
    | GET  |      |     Token: 0x77
    |      |      |  Uri-Path: lock
    |      |      |
    |      @<-----+      Code: 2.05 (Content)
    |      | 2.05 |     Token: 0x77
    |      |      |   Payload: 1 (Locked)
    |      |      |
    +------------>|      Code: 0.03 (PUT)
    | PUT  |      |     Token: 0x34
    |      |      |  Uri-Path: lock
    |      |      |   Payload: 1 (Unlock)
    |      |      |
    |      X<-----+      Code: 2.04 (Changed)
    |      | 2.04 |     Token: 0x34
    |      |      |
    +----->X      |      Code: 0.01 (GET)
    | GET  |      |     Token: 0x77
    |      |      |  Uri-Path: lock
    |      |      |
    <------@      |      Code: 2.05 (Content)
    | 2.05 |      |     Token: 0x77
    |      |      |   Payload: 1 (Locked)
    |      |      |
```

Figure 6: Mismatching Response to GET

MISMATCH ATTACK (3/3)

- Attacker can even mix responses from different resources if the resources share the same DTLS connection **on some part of the path.**
  - Located behind a common gateway
  - Served by the same CoAP proxy.
- Example: Attacker makes client believe the house is on fire.



```
Client    Foe    Server
   |       |       |
   +-------------->|      Code: 0.01 (GET)
   | GET   |       |     Token: 0x77
   |       |       |  Uri-Path: oven/temperature
   |       |       |
   |       @<------+      Code: 2.05 (Content)
   |       | 2.05  |     Token: 0x77
   |       |       |   Payload: 225
   |       |       |
     ....    ....
   |       |       |
   +------>X       |      Code: 0.01 (GET)
   | GET   |       |     Token: 0x77
   |       |       |  Uri-Path: livingroom/temperature
   |       |       |
   <-------@       |      Code: 2.05 (Content)
   | 2.05  |       |     Token: 0x77
   |       |       |   Payload: 225
   |       |       |
```

Figure 7: Mismatching Response from other resource

MISMATCH MITIGATION

- This has serious consequences and must be fixed. The earlier the better.
- **Which layer**?
  - DTLS (including CoAP-DTLS interworking)
  - CoAP
- **Where?**
  - Errata
  - New RFC
- **Proposed solution:** *A CoAP client MUST NOT reuse any tokens for a given source/destination which the client has not received responses to. The easiest way to accomplish this is to implement the token as a counter and never reuse any tokens at all, this approach SHOULD be followed.*

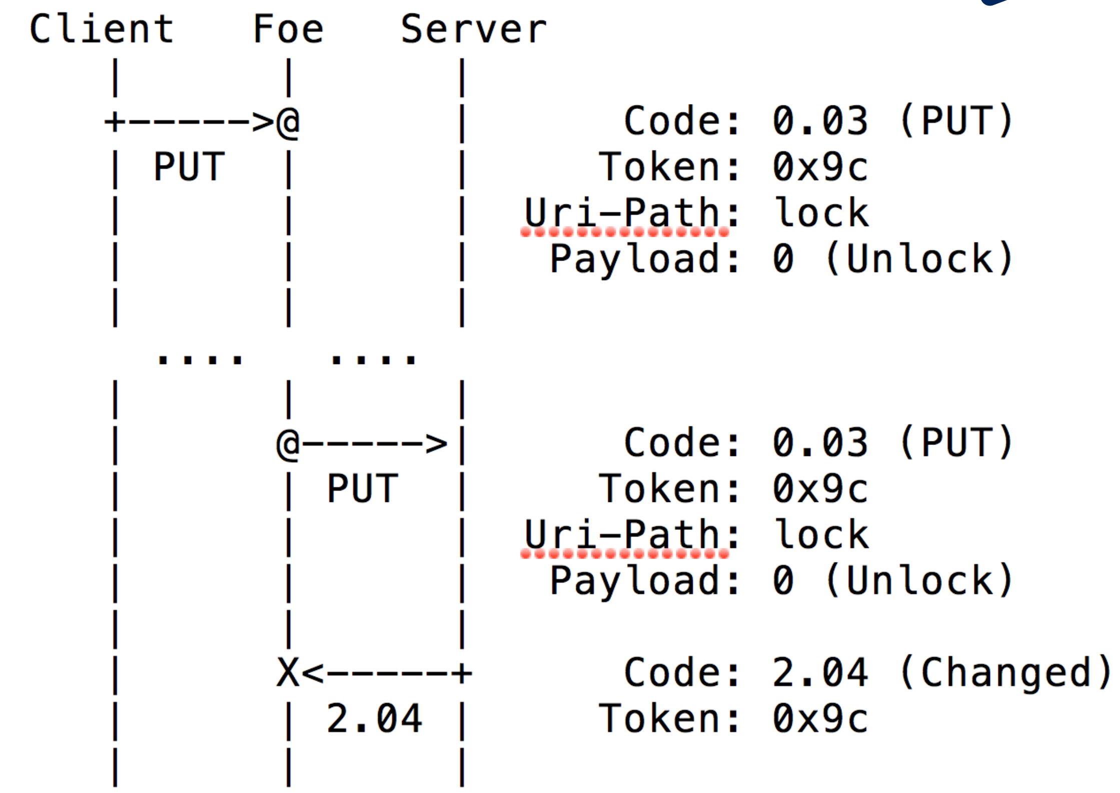
REQUEST DELAY ATTACK

Assumptions:

- Attacks are performed by an on-path attacker, not a “trusted” middle box.
- CoAP is protected with a security protocol such as DTLS, IPsec, or OSCOAP.

DELAY ATTACK (1/2)

- An attacker can delay the delivery of any packet (request or response) by a chosen amount of time.
- The attacker can control the replay window.
- Example: Attacker can e.g. unlock a door at a given time.



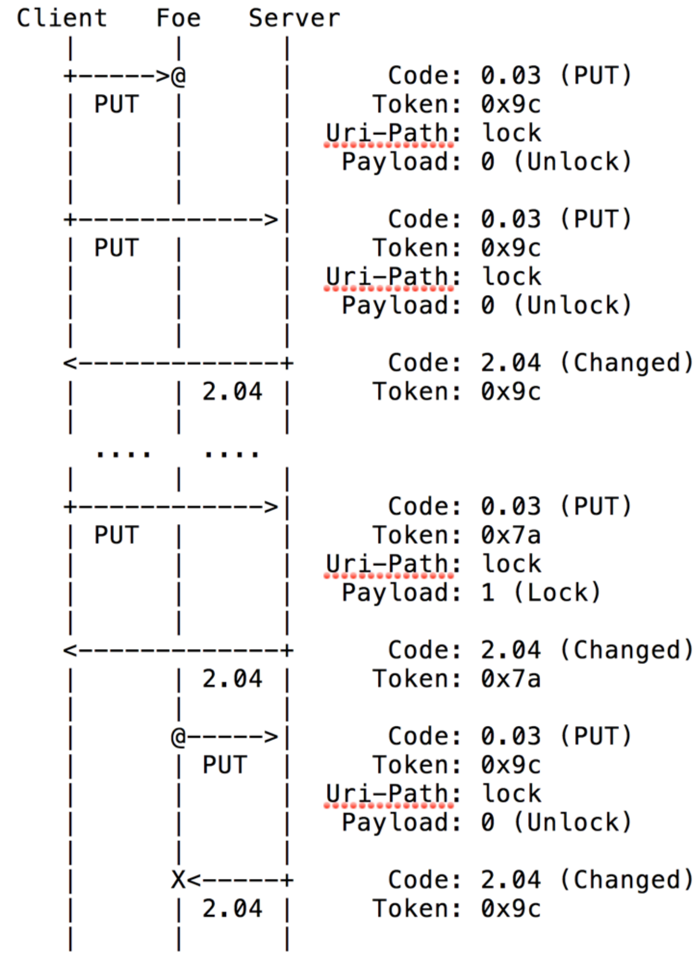
```
Client   Foe   Server
   |      |      |
   +----->@      |      Code: 0.03 (PUT)
   | PUT  |      |     Token: 0x9c
   |      |      |  Uri-Path: lock
   |      |      |   Payload: 0 (Unlock)
   |      |      |
     ....   ....
   |      |      |
   |      @----->|      Code: 0.03 (PUT)
   |      | PUT  |     Token: 0x9c
   |      |      |  Uri-Path: lock
   |      |      |   Payload: 0 (Unlock)
   |      |      |
   |      X<-----+      Code: 2.04 (Changed)
   |      | 2.04 |     Token: 0x9c
   |      |      |
```

Figure 3: Delaying a Request

DELAY ATTACK (2/2)

- **Reordering**: If a non-zero replay window is used, the attacker can let the client interact with the actuator before delivering the delayed request.

- Server needs to verify freshness. Cannot be put on implementers. So how?
  - Layer 2 (not possible on Internet)
  - CoAP option
  - CoAP payload



```
Client   Foe   Server
   |      |      |
   +----->@      |      Code: 0.03 (PUT)
   | PUT  |      |     Token: 0x9c
   |      |      |  Uri-Path: lock
   |      |      |   Payload: 0 (Unlock)
   |      |      |
   +------------>|      Code: 0.03 (PUT)
   | PUT  |      |     Token: 0x9c
   |      |      |  Uri-Path: lock
   |      |      |   Payload: 0 (Unlock)
   |      |      |
   <-------------+      Code: 2.04 (Changed)
   |      | 2.04 |     Token: 0x9c
   |      |      |
     ....   ....
   |      |      |
   +------------>|      Code: 0.03 (PUT)
   | PUT  |      |     Token: 0x7a
   |      |      |  Uri-Path: lock
   |      |      |   Payload: 1 (Lock)
   |      |      |
   <-------------+      Code: 2.04 (Changed)
   |      | 2.04 |     Token: 0x7a
   |      |      |
   |      @----->|      Code: 0.03 (PUT)
   |      | PUT  |     Token: 0x9c
   |      |      |  Uri-Path: lock
   |      |      |   Payload: 0 (Unlock)
   |      |      |
   |      X<-----+      Code: 2.04 (Changed)
   |      | 2.04 |     Token: 0x9c
   |      |      |
```

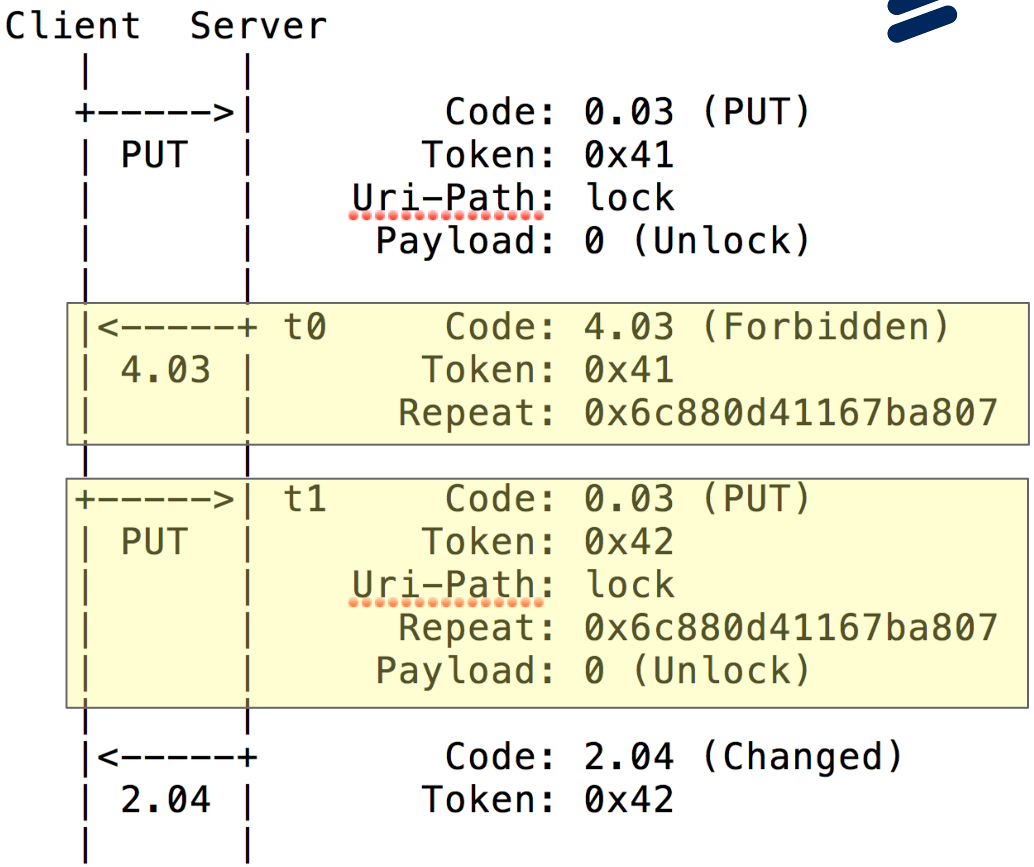
Figure 4: Delaying Request with Reordering

REPEAT OPTION

- New e2e challenge-response mechanism for CoAP, binding a request to an earlier response.

- The challenge (for the client) is simply to echo the value.

- Enables the server to verify the freshness of a request: (t1 - t0) = RTT < Threshold, thus mitigating delay attacks.



```
Client  Server
   |      |
   +----->|        Code: 0.03 (PUT)
   | PUT  |       Token: 0x41
   |      |    Uri-Path: lock
   |      |     Payload: 0 (Unlock)
   |      |
   |<-----+ t0     Code: 4.03 (Forbidden)
   | 4.03 |       Token: 0x41
   |      |      Repeat: 0x6c880d41167ba807
   |      |
   +----->| t1     Code: 0.03 (PUT)
   | PUT  |       Token: 0x42
   |      |    Uri-Path: lock
   |      |      Repeat: 0x6c880d41167ba807
   |      |     Payload: 0 (Unlock)
   |      |
   |<-----+        Code: 2.04 (Changed)
   | 2.04 |       Token: 0x42
   |      |
```

Figure 9: The Repeat Option

REPEAT OPTION

- Gives the server freshness guarantee independently of what the client does.

- A pseudorandom value requires the server to save the option value and time t0.

- Instead of a pseudorandom value send an CCM-encrypted timestamp:

  option value = AEAD(key, t0, NULL)
  or AEAD(key, t0, parts of request)

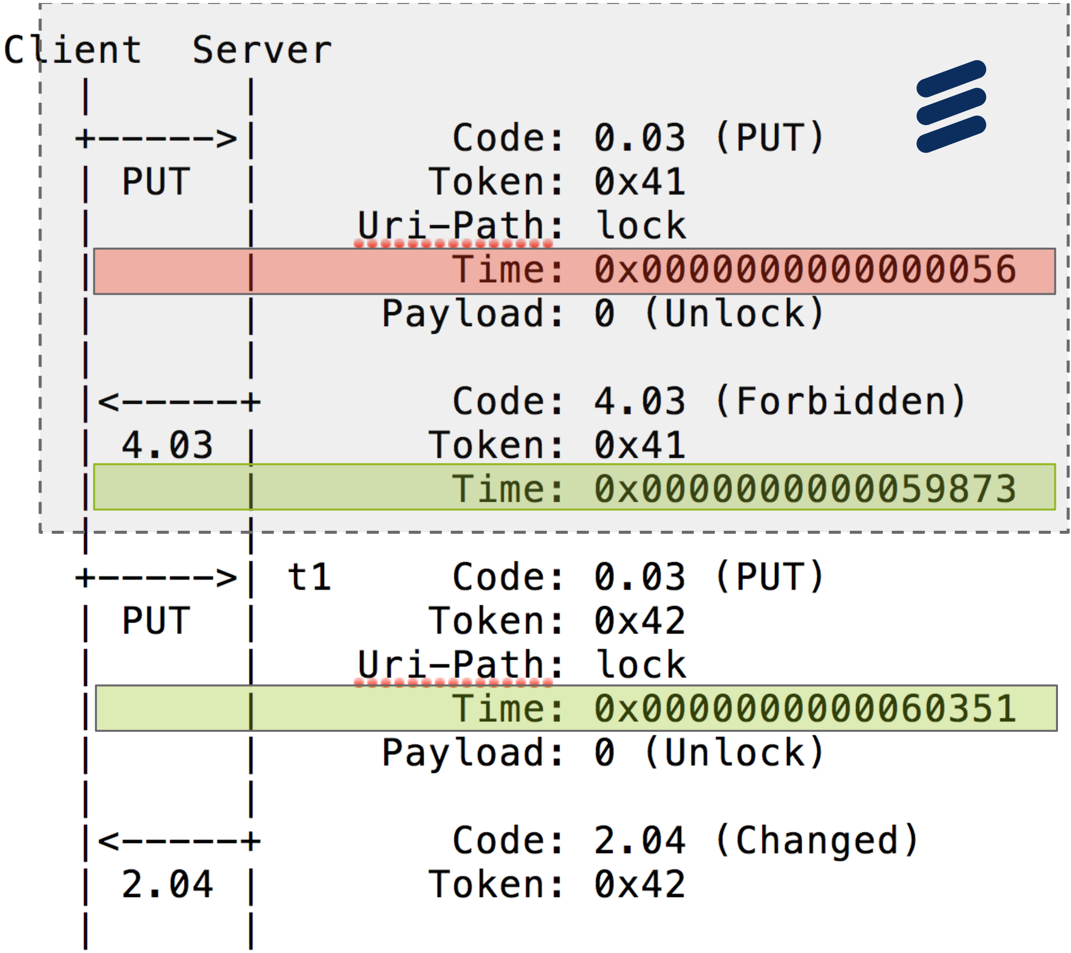
  This eliminates server state but still requires two round-trips.

TIME OPTION?

- New CoAP option for sending time.

- Time sync could be done in same option, but is probably better done via other solution (e.g. NTP)

- Server verifies freshness:
  t1 >> option value => reject



```
Client  Server
  |       |
  +------>|           Code: 0.03 (PUT)
  | PUT   |          Token: 0x41
  |       |       Uri-Path: lock
  |       |           Time: 0x0000000000000056
  |       |        Payload: 0 (Unlock)
  |       |
  |<------+           Code: 4.03 (Forbidden)
  | 4.03  |          Token: 0x41
  |       |           Time: 0x0000000000059873
  |       |
  +------>| t1        Code: 0.03 (PUT)
  | PUT   |          Token: 0x42
  |       |       Uri-Path: lock
  |       |           Time: 0x0000000000060351
  |       |        Payload: 0 (Unlock)
  |       |
  |<------+           Code: 2.04 (Changed)
  | 2.04  |          Token: 0x42
  |       |
```

Figure 9: The Time Option

ERICSSON

All times are in time-warped JST

Friday

- **09:00–09:05 Intro**
- **09:05–09:25 Delay Attacks (JM)**
- **09:25–10:00 Object Security (GS)**
- **10:00–10:30 CoCoA (CG, IJ)**
- **10:30–10:40 Mostly-offline nodes (PV)**
- **10:40–11:30 Flextime**

Object Security of CoAP Messages (OSCOAP)

draft-selander-ace-object-security-03

Göran Selander, Ericsson
John Mattsson, Ericsson
Francesca Palombini, Ericsson
Ludwig Seitz, SICS Swedish ICT

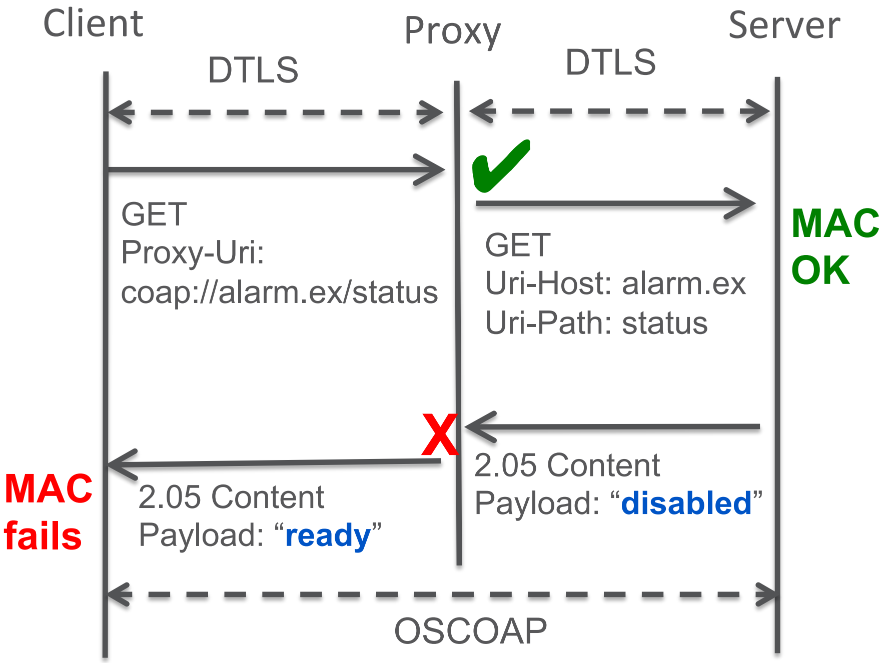
IETF 94, CORE WG, Yokohama, Nov 6, 2015

Recap by means of an example

- CoAP defines proxy operations
- Proxy needs access to CoAP message

- DTLS hop-by-hop
  - CoAP message unprotected in proxy
  - **With DTLS, the proxy is trusted . . .**
    - **with forwarding (OK)**
    - **with everything else also (not OK)**

- OSCOAP protects selected parts of CoAP message end-to-end*)
  - **Allows legitimate proxy operations**
  - **Detects illegitimate proxy operations**



Client

Proxy

Server

DTLS

DTLS

✓

GET
Proxy-Uri:
coap://alarm.ex/status

GET
Uri-Host: alarm.ex
Uri-Path: status

MAC OK

X

2.05 Content
Payload: "**disabled**"

2.05 Content
Payload: "**ready**"

MAC fails

OSCOAP

*) "end-to-end" := between origin client and origin server

Updates in version -03

Review comments:

- CoAP header field Version (must be integrity protected)
- CoAP options Block1, Block2 (must not be integrity protected)

Other changes:

- New CoAP option "Object-Security" (replacing options "Enc" and "Sig")
- Replaced "Mode" terminology:
  - OSCOAP: Object security of CoAP messages
  - OSCON: Object security of content/payload only (moved to an appendix)
- Removed all construction built on JOSE
- Updated to cose-msg-05; optimizations; presented in the COSE WG
- https://www.ietf.org/proceedings/94/slides/slides-94-cose-1.pdf

What security properties does CoAP need?

- So far the work with OSCOAP has been **reactive**
- It has gone surprisingly well!
- Not all parts of CoAP message can be protected end-to-end, but it isn't necessarily an issue
  - CoAP message layer
  - Token may be changed by a proxy, we introduced Transaction Identifier (TID)
  - Observe required separate TIDs in request and response
- Blockwise blocks can't be protected end-to-end – we think this is an issue
  - Easy DoS attack
  - End-to-end option would be an efficient protection of blocks **and** whole message

What security properties does CoAP need?

- We propose CORE WG should be more **proactive**
- Analysis of end-to-end security properties before becoming standard
- Consider errata to RFCs

- To support this work we propose a **separate draft** on required end-to-end security properties and what part of CoAP message needs to be protected
- The OSCOAP draft has currently 3 parts
  1. Security objectives ("why?")
  2. Protected CoAP metadata ("what?")
  3. Protection mechanism ("how?")
- Work started joint between University of Bremen, SICS and Ericsson

ERICSSON

Thank you!

Comments/questions?

All times are in time-warped JST

~~Tuesday~~ Friday

- **09:00–09:10 Intro, WG status**
- **09:10–09:25 CoAP over reliable (chairs)**
- **09:25–09:30 HTTP Mapping (chairs)**
- **09:30–09:50 Resource Directory (MK)**
- **09:50–10:05 CoRE Interfaces (MK)**
- **10:05–10:35 SenML (AK)**
- **10:35–10:55 COMI (PV)**
- **10:55–11:10 PATCH (PV)**
- **11:20–11:20 FETCH (CB), COOL (AP)**
- **11:20–11:30 Pubsub (MK)**

CoRE working group

Patch method for CoAP
draft-vanderstok-core-patch-02

P. van der Stok, A. Sehgal

Subjects

- Atomicity
- Return code 4.09 and 5.03
- Idempotent iPatch

Atomicity

Patch request	legal	Original server state	New server state
(b->B)	success	a.b.c.d	a.B.c.d
	No success	a.b.c.d	a.b.c.d
	Not allowed	a.b.c.d	A.b/B.c.d
(b->B, f->F)	success	a.b.c.d.e	a.B.c.d.e
<b->B, f->F)	success	a.b.c.d.f	a.B.c.d.F

Use case; changing manufacturer extensions without case testing

Multicast: atomicity for each individual server, not for all destinations together

Proposal accepted?

Return codes

Thanks to Christian Amsuss

4.09: when resource goes to inconsistent state which cannot be solved by server

5.03: concurrent request cannot be queued
too long waiting times (e.g. resource reserved too long)

4.09 needs additional work by client
5.03 suggests “try later” to client

Question: Is this satisfactory (refinement of HTTP Patch error codes)

Idempotency

CoAP Patch is NOT idempotent like HTTP Patch

iPatch is idempotent version of Patch

Question: do we keep iPatch?

Open issues

- Suggestions?

Ready for WG adoption?

More drafts refer to patch draft

All times are in time-warped JST

~~Tuesday~~ Friday

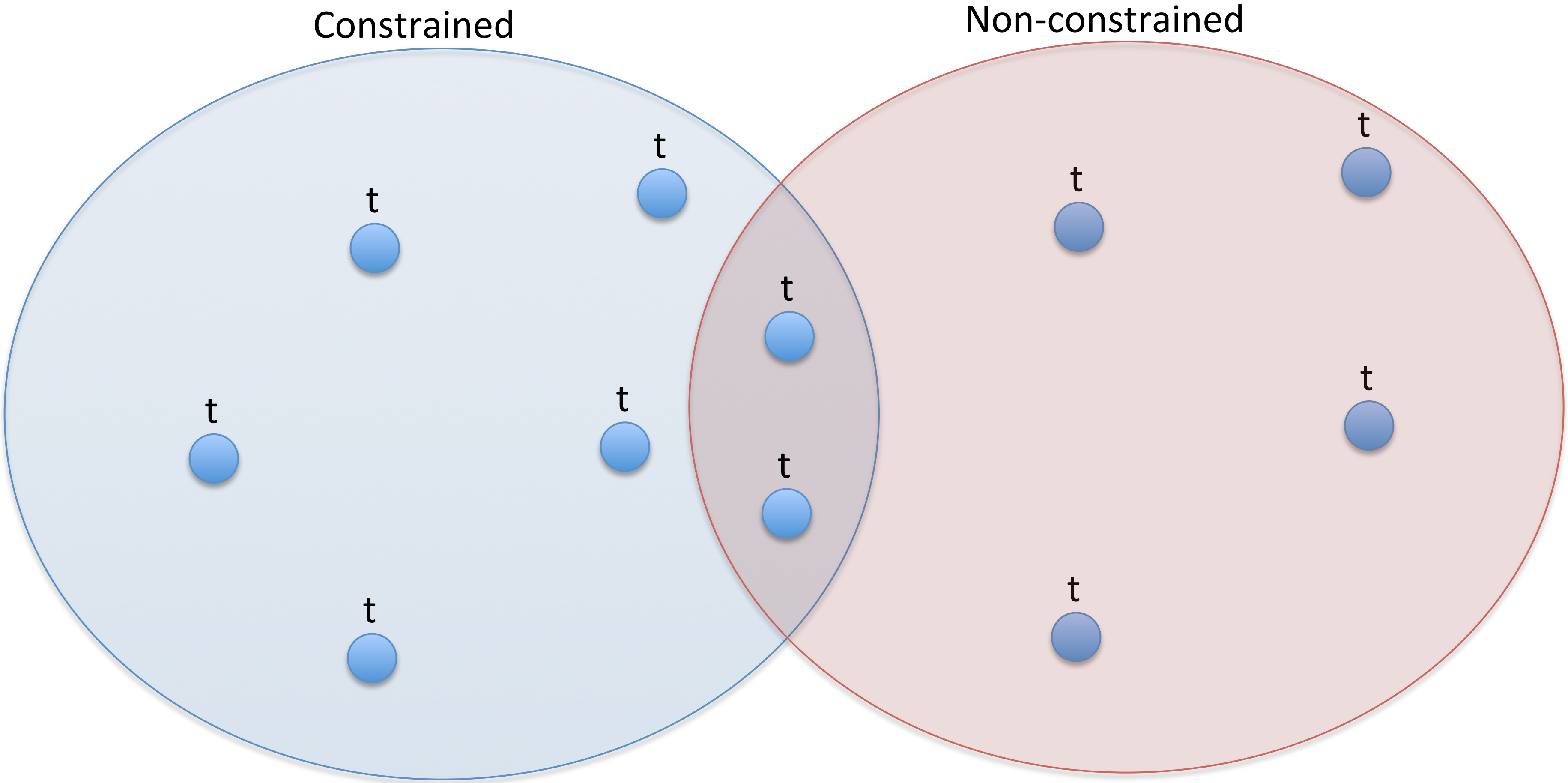
- 09:00–09:10 Intro, WG status
- 09:10–09:25 CoAP over reliable (chairs)
- 09:25–09:30 HTTP Mapping (chairs)
- 09:30–09:50 Resource Directory (MK)
- 09:50–10:05 CoRE Interfaces (MK)
- 10:05–10:35 SenML (AK)
- 10:35–10:55 COMI (PV)
- 10:55–11:10 PATCH (PV)
- 11:20–11:20 FETCH (CB), COOL (AP)
- 11:20–11:30 Pubsub (MK)

Constrained Objects Language

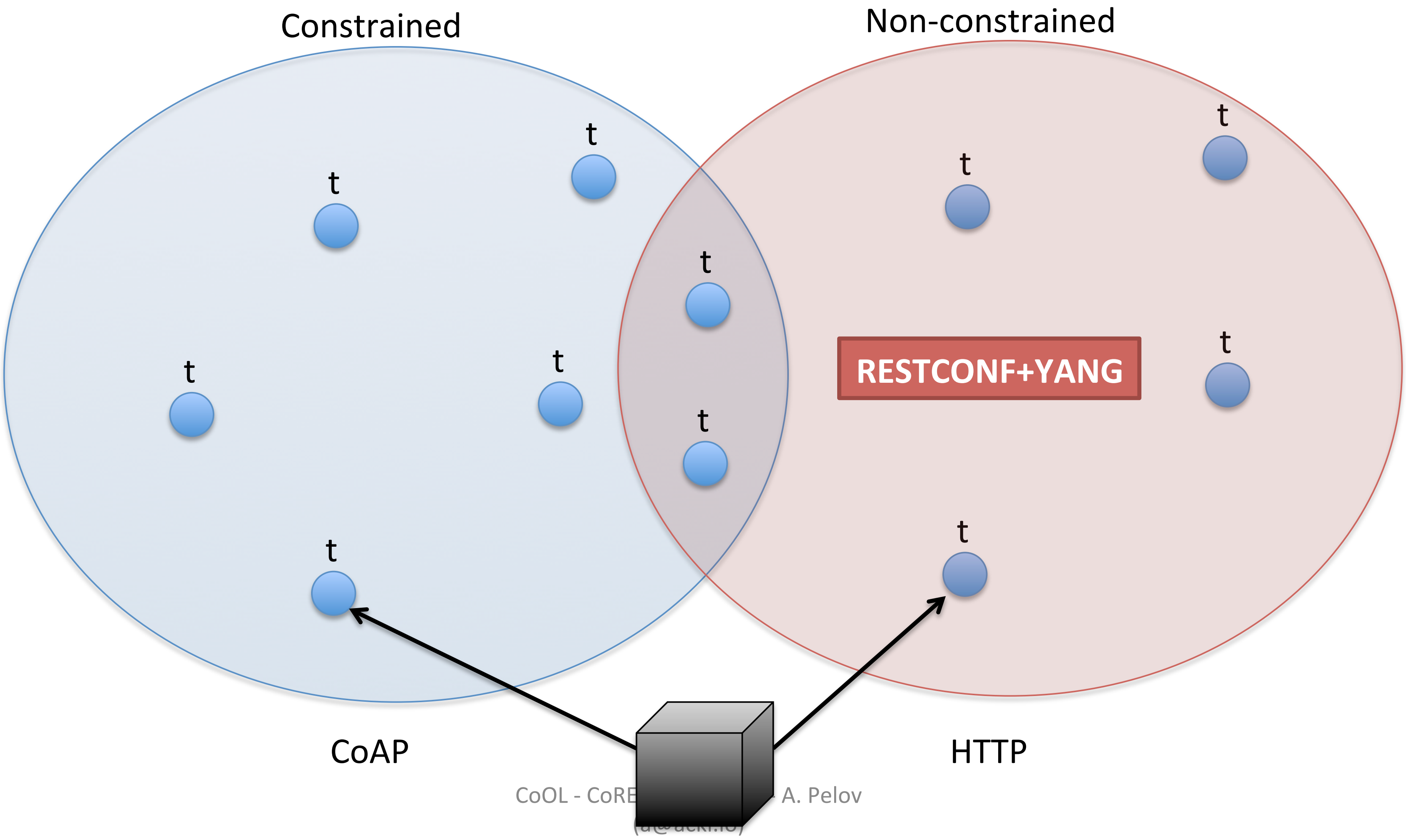
CoOL

Michel Veillette
Alexander Pelov (a@ackl.io)
Abhinav Somaraju
Randy Turner

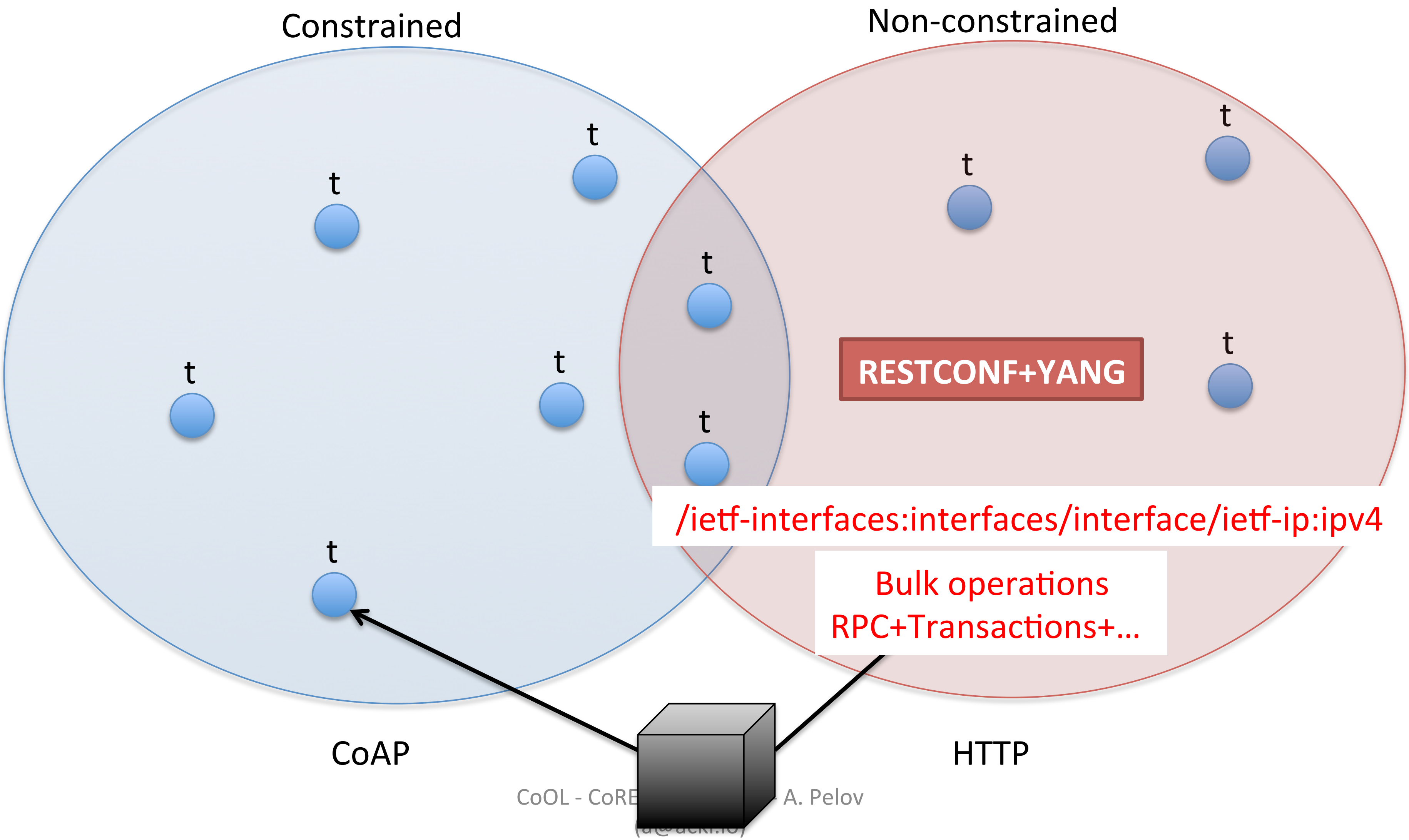
You want to manage things



You want to manage things



You want to manage things



Constrained

Non-constrained

RESTCONF+YANG

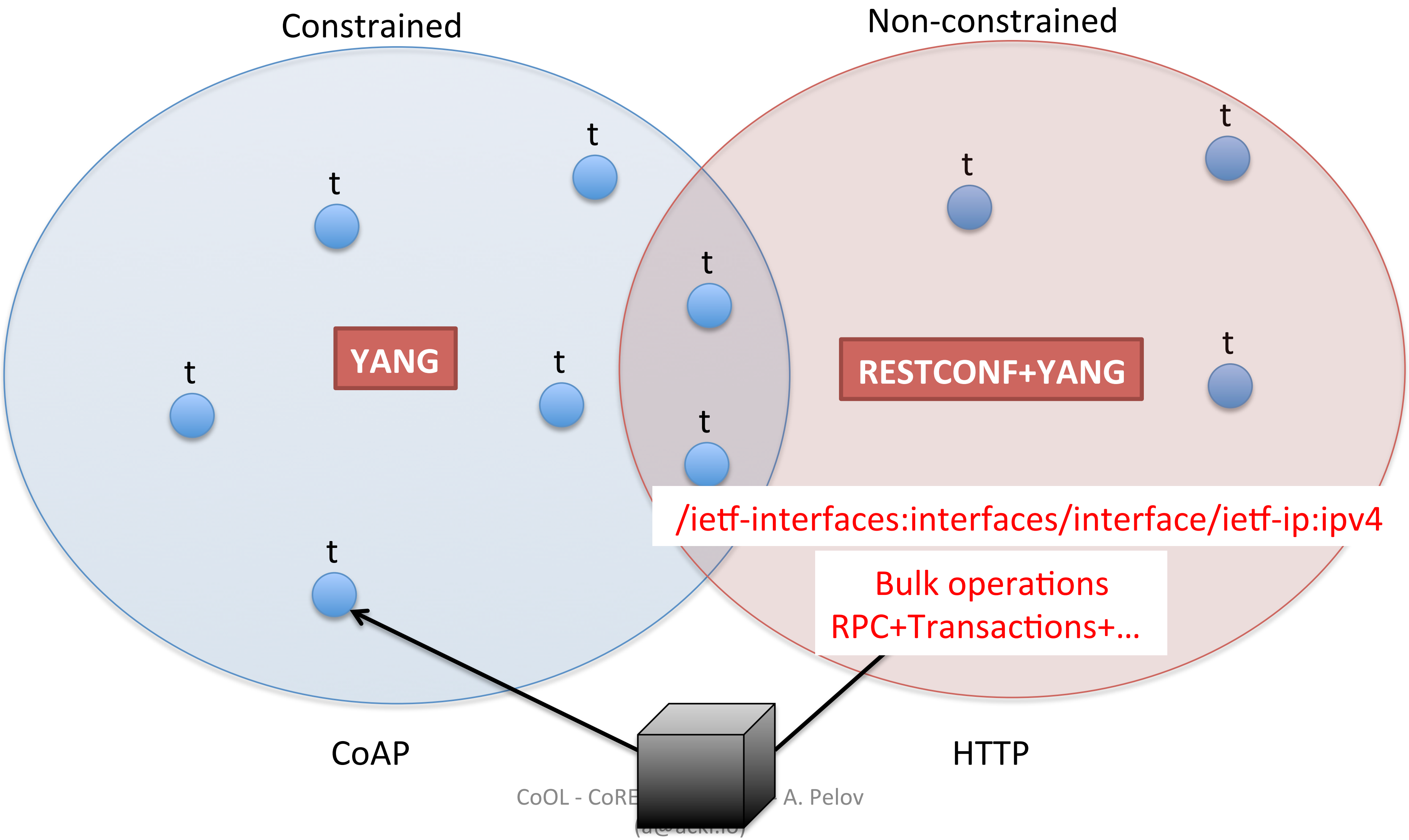
/ietf-interfaces:interfaces/interface/ietf-ip:ipv4

Bulk operations
RPC+Transactions+…

CoAP

HTTP

You want to manage things



Constrained

Non-constrained

YANG

RESTCONF+YANG

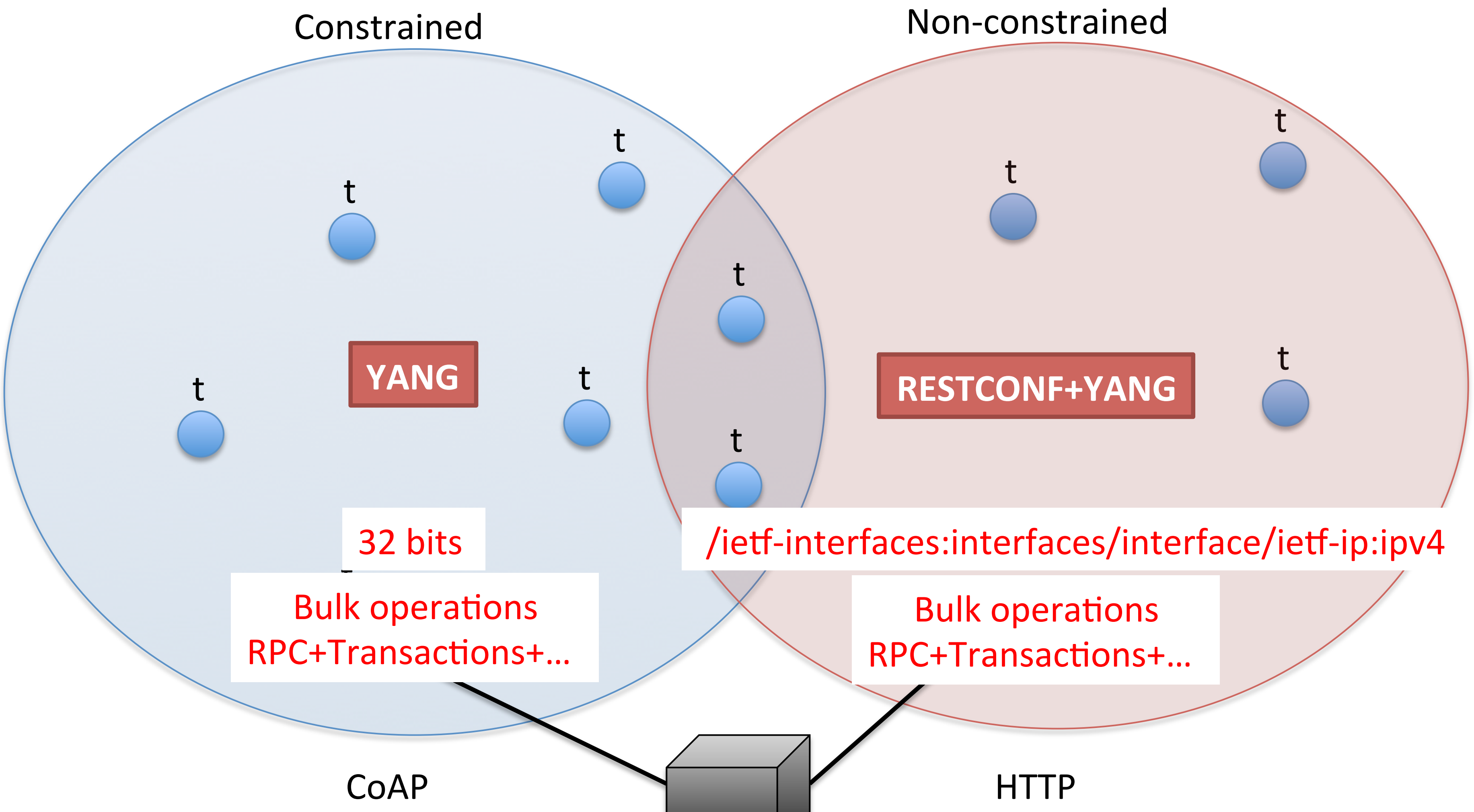
/ietf-interfaces:interfaces/interface/ietf-ip:ipv4

Bulk operations
RPC+Transactions+…

CoAP

HTTP

You want to manage things



Constrained

Non-constrained

YANG

RESTCONF+YANG

32 bits

/ietf-interfaces:interfaces/interface/ietf-ip:ipv4

Bulk operations
RPC+Transactions+…

Bulk operations
RPC+Transactions+…

CoAP

HTTP

CoMI

- Bringing NETCONF/RESTCONF to constrained devices
  - CoAP, CBOR, etc.
  - Compressing identifiers is key
- Based on a beautiful, but broken principle - hashes
  - There can always be collisions
  - Put burden on servers, clients, and networks
  - Difficult to test
    - A hash clash 5 years down the road can break your network
  - Hashes are not simple
    - 20% of the draft deal with hashes, and they are still underspecified
  - No CBOR magic
    - Hashes always take 5 bytes

Where should we compromise?

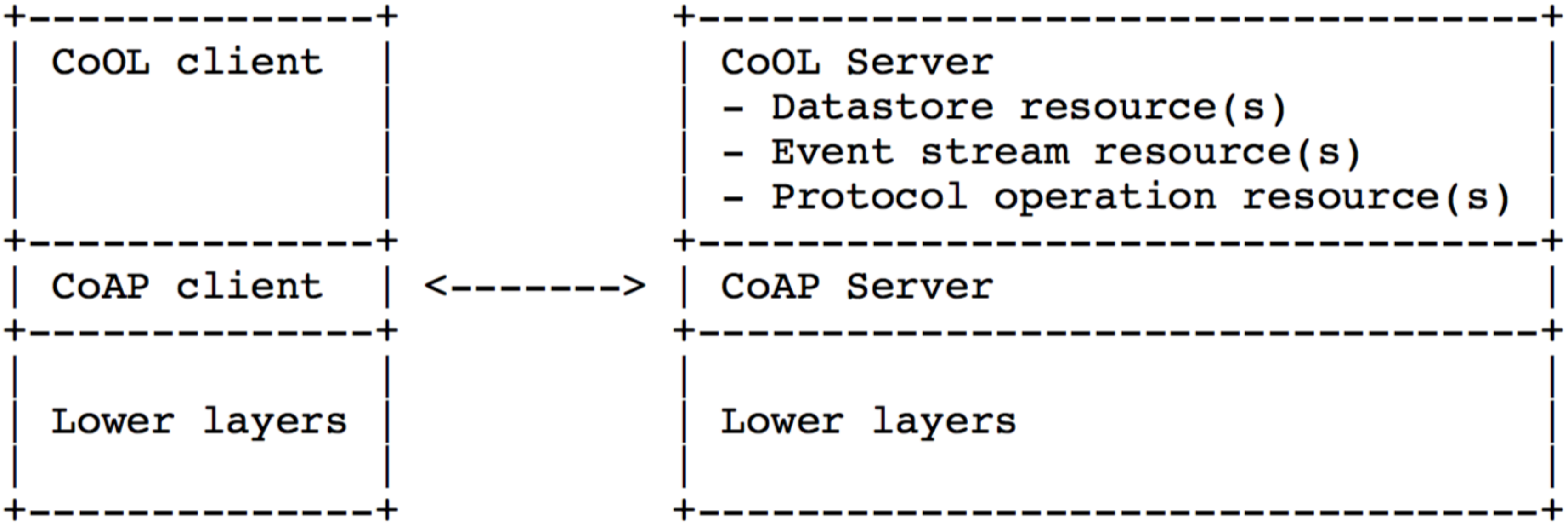
CoMI

- Focus
  - Simplify peoples’ lives by using unmanaged IDs
- Compromise
  - Subset of the YANG
  - Larger message size
  - Extra handshake, logic and storage requirements
  - Non-deterministic protocol behavior

CoOL

- Focus
  - Full coverage of YANG
  - Smaller message size, and implementation footprint
  - Deterministic behavior
- Compromise
  - YANG module IDs need to be registered

CoOL architecture



```
+--------------+           +----------------------------------+
| CoOL client  |           | CoOL Server                      |
|              |           | - Datastore resource(s)          |
|              |           | - Event stream resource(s)       |
|              |           | - Protocol operation resource(s) |
+--------------+           +----------------------------------+
| CoAP client  | <-------> | CoAP Server                      |
+--------------+           +----------------------------------+
|              |           |                                  |
| Lower layers |           | Lower layers                     |
|              |           |                                  |
+--------------+           +----------------------------------+
```

CoOL

- Managed Identifiers (30 bits)
  - **Module ID (20 bit) + Node ID (10 bits)**
    - **Module ID** is allocated (Module IDs for private modules and dev modules)
    - **Node ID** is automatically allocated (could be overridden)

- Central repository for modules
  - Need IANA action
    - A company can obtain a bundle, and allocate the IDs from this bundle at will
  - Distributed scenarios (t2t!)
  - Reflection for free (meta-data, such as *module discovery, module versioning, aliases* for frequently used data nodes, …)

- CBOR magic (most identifiers take 1 byte instead of 5)
  - **Module ID** can be omitted in collections
  - Collections (e.g. selection of multiple rows, multiple columns)

- ID of DATA, EVENTS and RPCs

CoOL

- Perform on a single resource (e.g. GET /cool)
  - *“Fields”* option contains the list of nodes selected, encoded using a CBOR array

Qualified

REQ: GET /cool Fields([14337, 18, 19])

CBOR

Unqualified

RES: 2.05 Content (Content-Format: application/cbor)

```
{
  14337: 57,
  18 : 76,
  19 : 837
}
```

CBOR

Recap on CoOL

- Managed IDs
  - Data nodes, notification streams, protocol operations (RPC)
  - Very efficient for single node operations
    - Extremely efficient for collection operations
- Straightforward, deterministic protocol behavior
- Full draft, specifying the complete YANG<->CBOR mapping
- Explicit PATCH, support ordered-by user list.
- Full coverage of YANG, no restrictions on data types, notification, RPC, …
- Enhanced list access (multi-rows, specific columns)

Next steps

- Deterministic multimaps
- PubSub
- Variable datastores
- Distributed transactions

CoAP FETCH

2015-11-03, CoRE WG
Carsten Bormann cabo@tzi.org

- https://maps.google.com/maps?f=d&source=s_d&saddr=Millbrae+Caltrain+Station&daddr=&hl=en&geocode=&mra=ls&dirflg=r&ttype=dep&noexp=0&noal=0&sort=&sll=37.510543,-122.259507&sspn=0.012357,0.015235&ie=UTF8&lci=transit&start=0&ll=37.603641,-122.386107&spn=0.013566,0.018046&z=16&iwloc=lyrftr:m,0x808f77b091ff6be5:0x725e536abba2f0c6,37.599829,-122.386537

What if > ~ 1KiB?

- Switch to POST
  - can send detailed parameters in payload instead
- Lose GET properties
  - safe
  - idempotent

HTTP SEARCH

- Like GET
- Add a body
- No longer need to POST a > 1KiB search

CoAP FETCH

- Similar to HTTP SEARCH
    - Add request payload to a GET
- Slightly different semantics: cacheable

FETCH and collections

- FETCH request payload has a media type
  - Can define application-specific formats
- Addressing Collections: e.g.,
  `[* selector]`

Caveat

- GET operates on a link
- FETCH additionally requires guidance how to construct payload
  - form relations!

FETCH rhymes with PATCH

- GET, PUT, POST, DELETE
- FETCH, iPATCH, PATCH
    - Patch payload, e.g.:
      `{ * selector => action }`

CoAP Methods

Code	Name	Code	Name	safe	idempotent
0.01	GET	0.05	FETCH	yes	yes
0.02	POST	0.06	PATCH	no	no
0.03	PUT	0.07	iPATCH	no	yes
0.04	DELETE			no	yes

```
s/iPATCH/???/?
```

All times are in time-warped JST

~~Tuesday~~ Friday

- **09:00–09:10 Intro, WG status**
- **09:10–09:25 CoAP over reliable (chairs)**
- **09:25–09:30 HTTP Mapping (chairs)**
- **09:30–09:50 Resource Directory (MK)**
- **09:50–10:05 CoRE Interfaces (MK)**
- **10:05–10:35 SenML (AK)**
- **10:35–10:55 COMI (PV)**
- **10:55–11:10 PATCH (PV)**
- **11:20–11:20 FETCH (CB), COOL (AP)**
- **11:20–11:30 Pubsub (MK)**

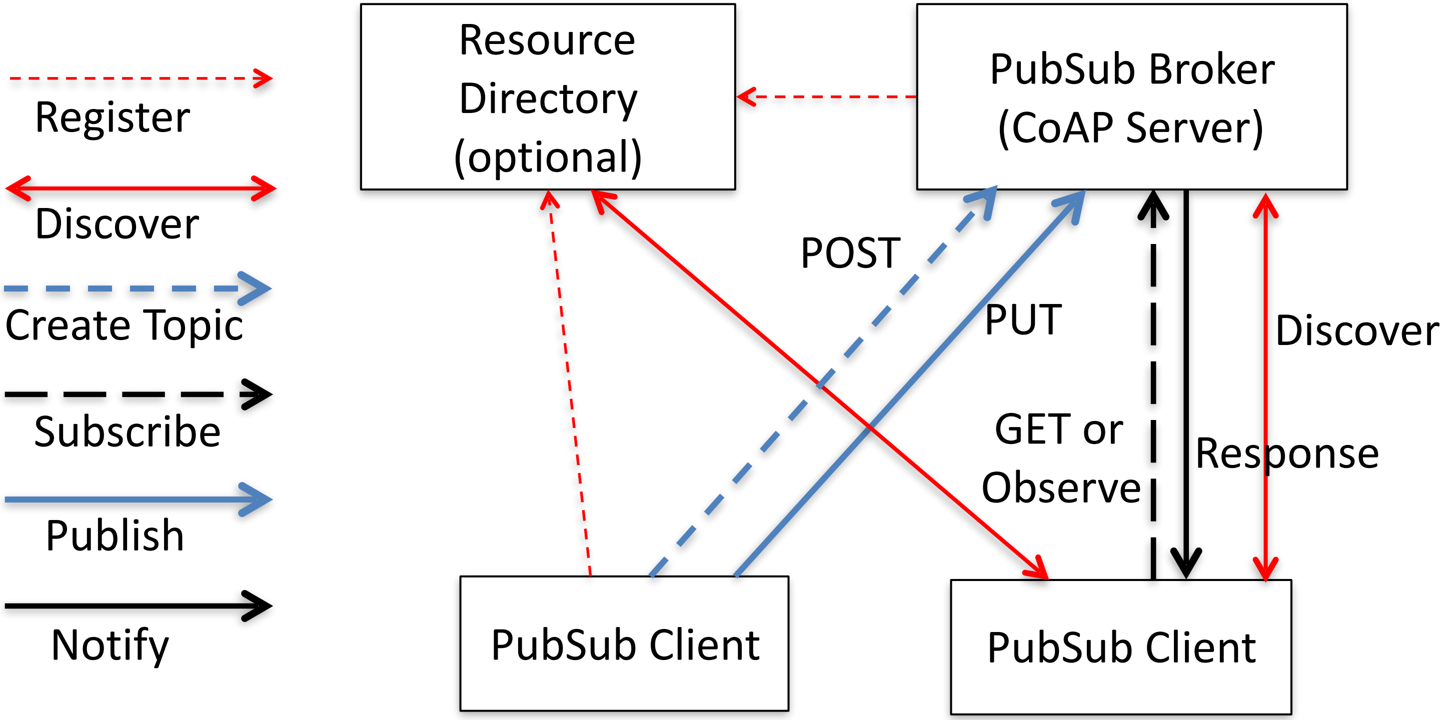
Publish-Subscribe Broker for the Constrained Application Protocol

draft-koster-core-coap-pubsub-04

Overview

- Pub/sub pattern enables device to act as a CoAP client and initiate transactions
  - devices can operate with limited reachability
- Broker becomes the state origin of the system
- Application can GET, Observe, and PUT to interact with topics on the broker as it would with resources, using Observe to Subscribe and PUT to Publish

Architecture Review



Updates

- Clarified some operations and interactions
- Decided not to significantly change how it works, keep it simple, CoAP flavored, not trying to replicate MQTT
- Doubt that significant QoS can be achieved without an end to end application handshake
- REST broker with publication resources and delivery receipts can be developed in a new draft
- Focus on cleaning up, correcting errors and underspecified elements

Updates

- Removed discussion of CON/NON/ACK
- Specified what happens when topics and subscribers are removed
- Added topic discovery using link-format GET
- Added discussion of sub-topic creation using links

Topic Discovery

```
Client                                          Broker
  |                                               |
  | ------ GET /.well-known/core?rt=core.ps ----->|
  | ---Content-Format: application/link-format----|
  |                                               |
  | <-----2.05 Content “</ps/>;rt="core.ps"-------|
  |                                               |

Client                                          Broker
  |                                               |
  | ---------- GET /ps/?rt="temperature" -------->|
  |    Content-Format: application/link-format    |
  |                                               |
  | <---2.05 Content                              |
  |   </ps/currentTemp>;rt="temperature";ct=50 ---|
  |                                               |
```

Sub-topics

```
Client                                          Broker
  |                                               |
  | ------- POST /ps/ "<mainTopic>;ct=40" ------->|
  |                                               |
  | <---------------- 2.01 Created ---------------|
  |             Location: /ps/mainTopic/          |
  |                                               |
  |                                               |
  | --- POST /ps/mainTopic/ "<subTopic>;ct=50" -->|
  |                                               |
  | <---------------- 2.01 Created ---------------|
  |        Location: /ps/mainTopic/subTopic       |
  |                                               |
  |                                               |
```

Open Issues

- Does a broker maintain an internal representation or does it buffer payloads?
  - Should do queuing?
- Should a broker convert content formats?
  - Advertise alternate formats in links?
- POST vs. PUT to publish?
  - Suggest only PUT
- Default Max-Age if not specified?
  - No expiration

https://svn.tools.ietf.org/svn/wg/core/charter-ietf-core.txt

The WG will perform maintenance on its first four standards-track specifications (RFC 6690, RFC 7252, -observe, -block) and will continue to evolve the experimental group communications support (RFC 7390). The working group will not develop a reliable multicast solution.

CoAP today works over UDP and DTLS. The WG will define transport mappings for alternative transports as required, both IP (starting with TCP and a secure version over TLS) and non-IP (e.g., SMS, working with DICE on potentially addressing the security gap); this includes defining appropriate URI schemes. Continued compatibility with CoAP over SMS as defined in OMA LWM2M will be considered.

…

https://svn.tools.ietf.org/svn/wg/core/charter-ietf-core.txt

CoRE will continue and complete its work on its resource-directory, as already partially adopted by OMA LWM2M. Interoperability with DNS-SD (and the work of the dnssd working group) will be a primary consideration. The WG will also work on a specification enabling broker-based publish-subscribe-style communication over CoAP.

CoRE will work on related data formats, such as alternative representations of RFC 6690 link format and RFC 7390 group communication information. The WG will complete the SenML specification, again with consideration to its adoption in OMA LWM2M.

RFC 7252 defines a basic HTTP mapping for CoAP, with further discussion in -http-mapping. This mapping will be evolved and supported by further documents.

https://svn.tools.ietf.org/svn/wg/core/charter-ietf-core.txt

Beside continuing to examine operational and manageability aspects of the CoAP protocol itself, CoRE will also develop a way to make RESTCONF-style management functions available via CoAP that is appropriate for constrained node networks. This will require very close coordination with NETCONF and other operations and management WGs.

The WG has selected DTLS as the basis for the communications security in CoAP. CoRE will work with DICE on the efficiency of this solution. The preferred cipher suites will evolve in cooperation with the TLS working and CFRG research groups. ACE is expected to provide solutions to authorization that may need complementary elements on the CoRE side. Object security as defined in JOSE and being adapted to the constrained node network requirements in COSE also may need additions on the CoRE side.

…

https://svn.tools.ietf.org/svn/wg/core/charter-ietf-core.txt

The WG will coordinate on requirements from many organizations and SDO. The WG will closely coordinate with other IETF WGs, particularly of the constrained node networks cluster (6Lo, 6TiSCH, LWIG, ROLL, ACE, COSE, DICE), and appropriate groups in the IETF OPS and Security areas. Work on these subjects, as well as on interaction models and design patterns (including follow-up work around the CoRE Interfaces draft) may benefit from close cooperation with the proposed Thing-to-Thing Research Group.

All times are in time-warped JST

Friday

- 09:00–09:05 Intro
- 09:05–09:25 Delay Attacks (JM)
- 09:25–10:00 Object Security (GS)
- 10:00–10:30 CoCoA (CG, IJ)
- 10:30–10:40 Mostly-offline nodes (PV)
- 10:40–11:30 Flextime

CoAP Simple Congestion Control/Advanced (CoCoA)

draft-bormann-core-cocoa

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

IETF 94 – Yokohama, Nov 2015

Outline

- 1. Emulation results on NONs
  - Based on -02

A. Betzler, C. Gomez, I. Demirkol, "Evaluation of Advanced Congestion Control Mechanisms for Unreliable CoAP Communications", PE-WASUN, Cancún, Mexico, Nov 2015

- 2. Updates in -03

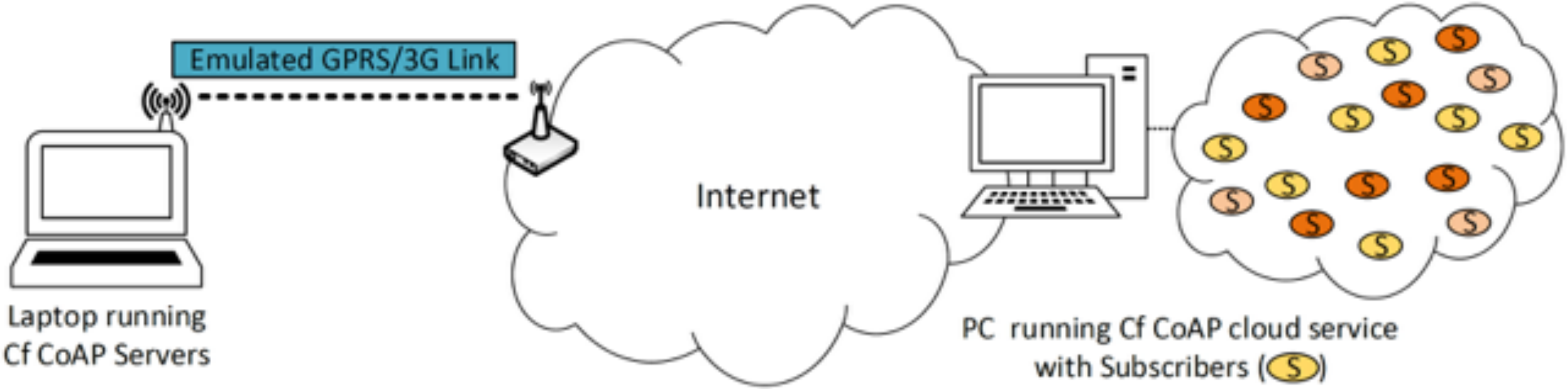
- 3. Plan for -04

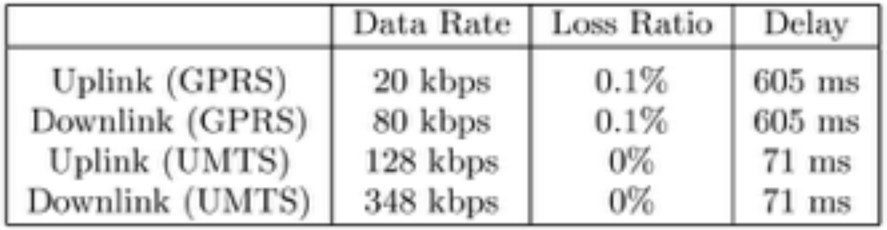
1. Emulation results on NONs

- Default CoAP
  - No rate limitation for NONs that do not trigger a response
- Observe:
  - NON rate SHOULD NOT exceed 1 / 3 s
- CoCoA on NONs
  - NON rate cannot exceed 1 / RTO
    - RTO obtained as per CoCoA (Section 4)
  - 2 out of 16 consecutive NONs must be CONs
    - To get RTT samples

1. Emulation results on NONs

- Evaluation setup

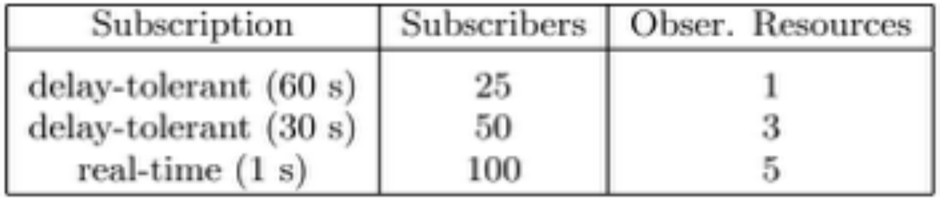

Emulated GPRS/3G Link
Internet
Laptop running Cf CoAP Servers
PC running Cf CoAP cloud service with Subscribers (S)
S



	Data Rate	Loss Ratio	Delay
Uplink (GPRS)	20 kbps	0.1%	605 ms
Downlink (GPRS)	80 kbps	0.1%	605 ms
Uplink (UMTS)	128 kbps	0%	71 ms
Downlink (UMTS)	348 kbps	0%	71 ms

→ Parameters for the link emulator

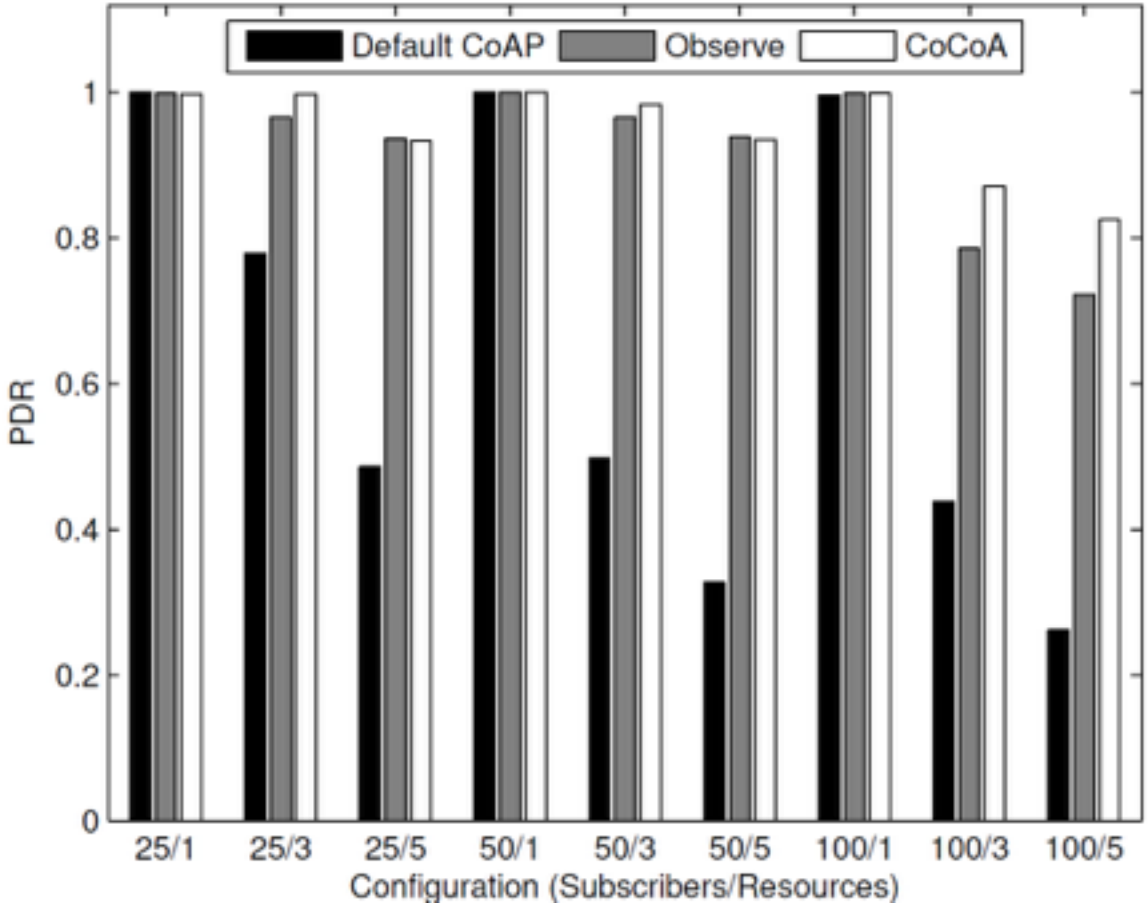
Combinations of use cases, # of subscribers, and observed resources →



Subscription	Subscribers	Obser. Resources
delay-tolerant (60 s)	25	1
delay-tolerant (30 s)	50	3
real-time (1 s)	100	5

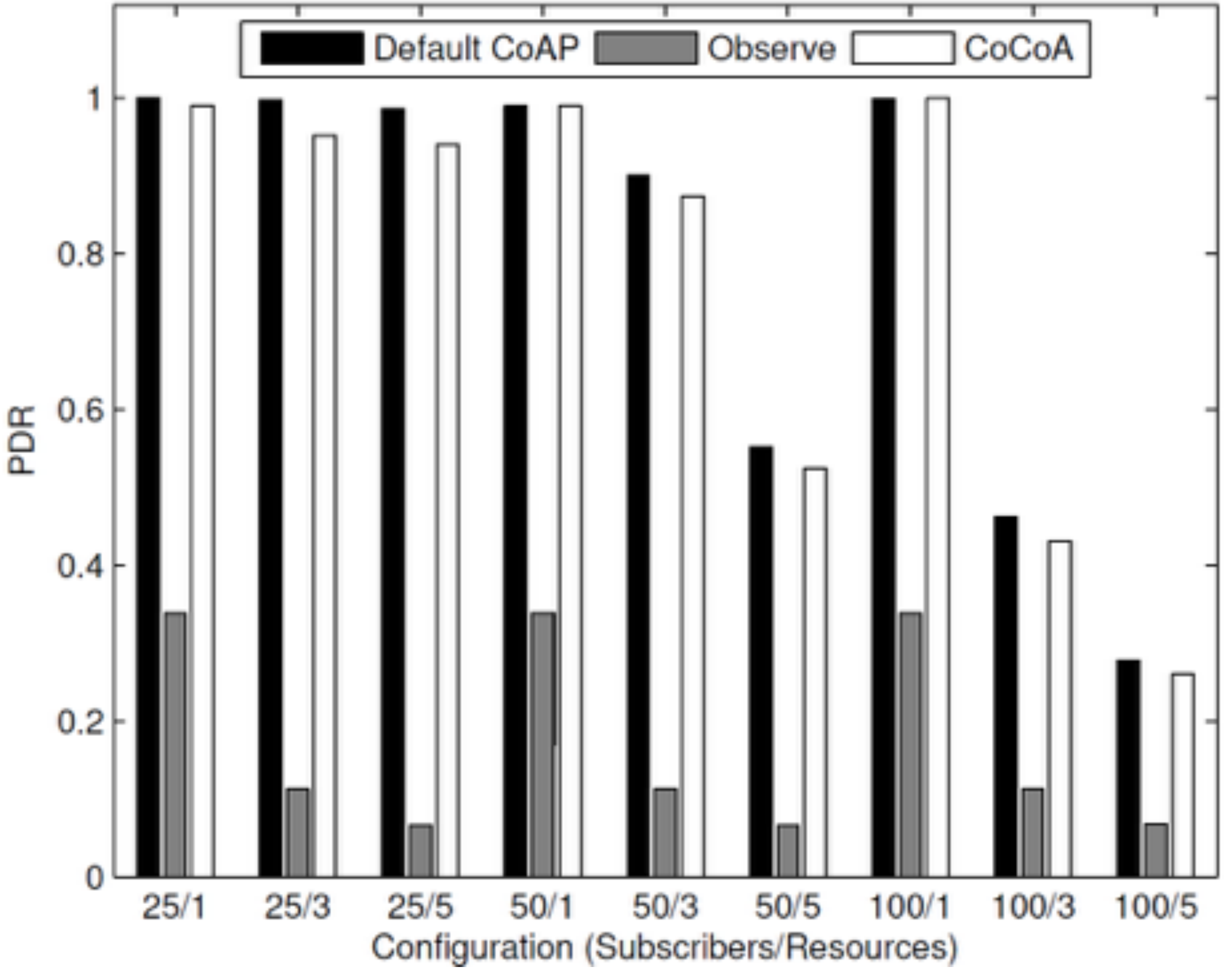
1. Emulation results on NONs

- Notification round every 60 s
  - Packet Delivery Ratio (PDR)
  - GPRS



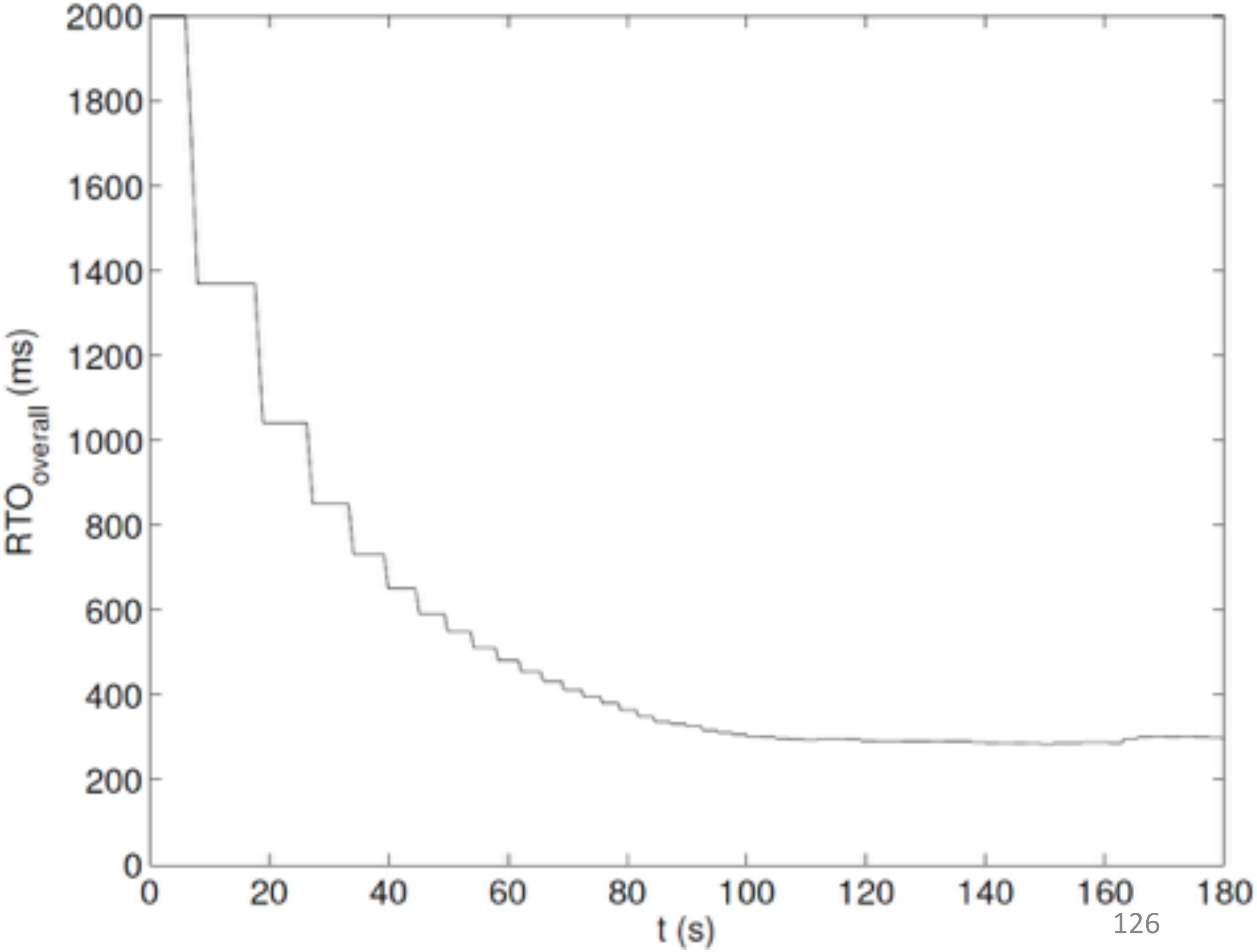
1. Emulation results on NONs

- Notification round every 1 s
  - UMTS



1. Emulation results on NONs

- RTO evolution for CoCoA
  - Example
  - Test run
    - 25 subscribers
    - 3 resources



2. Updates in -03 (I/III)

- New Section 3: Area of Applicability
  - Intended to be generally applicable
  - Better goodput, latency, recovery from bursts, etc., without sacrificing safety (vs Default CoAP)
  - State per scope (caveat: class 1 devices)

- Terminology alignment with relevant RTO-related RFCs (e.g. RFC 6298)
  - $RTO_{overall} := 0.25*RTO_{weak} + 0.75*RTO_{overall}$
  - $RTO_{overall} := 0.5*RTO_{strong} + 0.5*RTO_{overall}$

  OLD

  - RTO := 0.25*E_weak_ + 0.75*RTO
  - RTO := 0.5*E_strong_ + 0.5*RTO

  NEW

2. Updates in -03 (II/III)

- New Section 6: Aggregate congestion control
  - Motivation
    - Need to control all the traffic generated by a node including parallel interactions
      - RFC 5405
      - draft-eggert-core-congestion-control-01
      - draft-bormann-core-cc-qq-01
  - Definition
    - PLIMIT: maximum number of outstanding interactions towards different destinations in parallel
    - If RTO estimate exists for a destination
      - PLIMIT = max(LAMBDA, LAMBDA*ACK_TIMEOUT)/mean(RTO))
    - Otherwise
      - PLIMIT = LAMBDA
    - Default LAMBDA = 4
      - Is this too conservative?

2. Updates in -03 (III/III)

- New Section 6: Aggregate congestion control
  - Discussion
    - Reuse available information, not add much complexity
    - High RTO as a sign of congestion
    - RTOs for some destinations may inadequately influence the transmission to other „uncorrelated“ destinations
    - LAMBDA is a critical parameter
    - Could be more conservative
      - E.g. Use max (RTO) instead of mean(RTO)
  - Question to the WG: is this useful?

3. Plan for -04

- Get feedback, stabilize aggregate congestion control proposal

- Fulfill editorial holes

- Then ask for WG adoption

Call to Action

- Please implement cocoa-03
  - Is the draft specification clear?
- Please experiment with cocoa-03
  - Performance issues?
  - Improvement possibilities?
- Please provide feedback

Evaluation of CoAP, CoCoA, and TCP-based Congestion Control Approaches

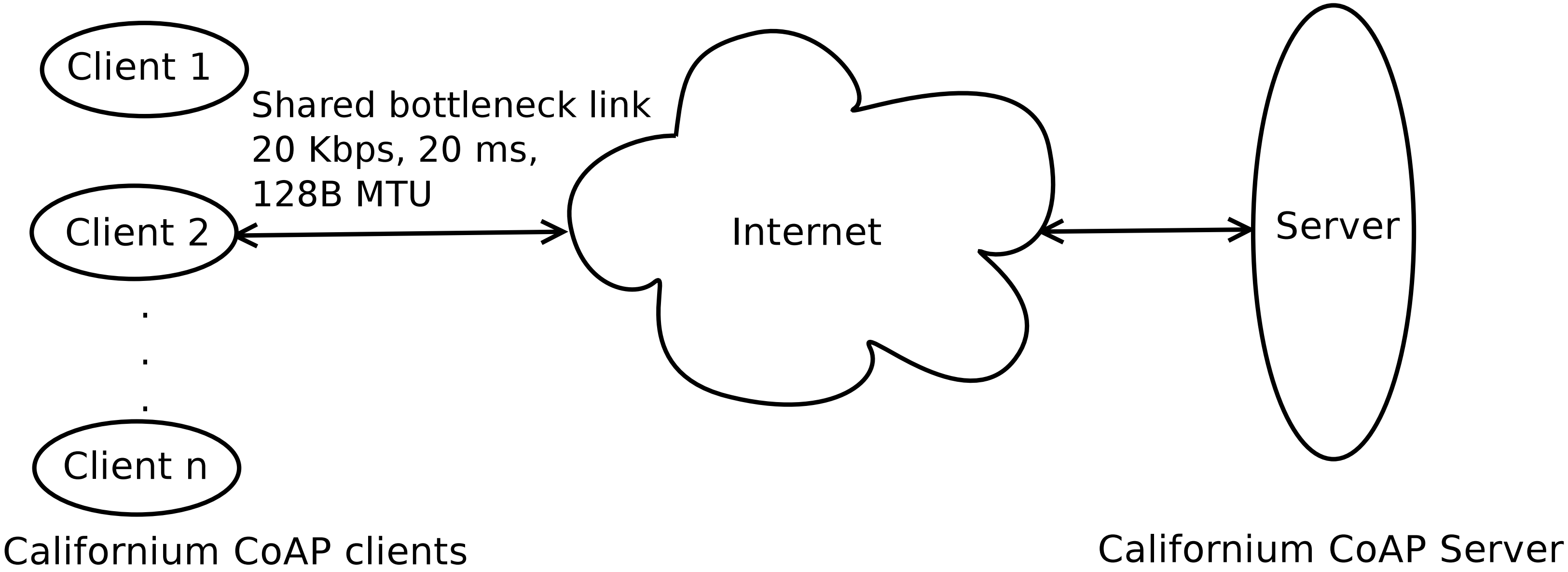
Ilpo Järvinen, Laila Daniel, and Markku Kojo

Department of Computer Science
University of Helsinki

UNIVERSITY OF HELSINKI

IETF94 / core WG
Nov 6th 2015

Test Setup



- 10, 20, 30, 40, and 80 clients talking with 1 CoAP server
- 30B payload
- Bottleneck link: 20kbps / 20ms emulated using dual-queue netem[1]
- 4, 10, and 50 packets buffer in front of the bottleneck

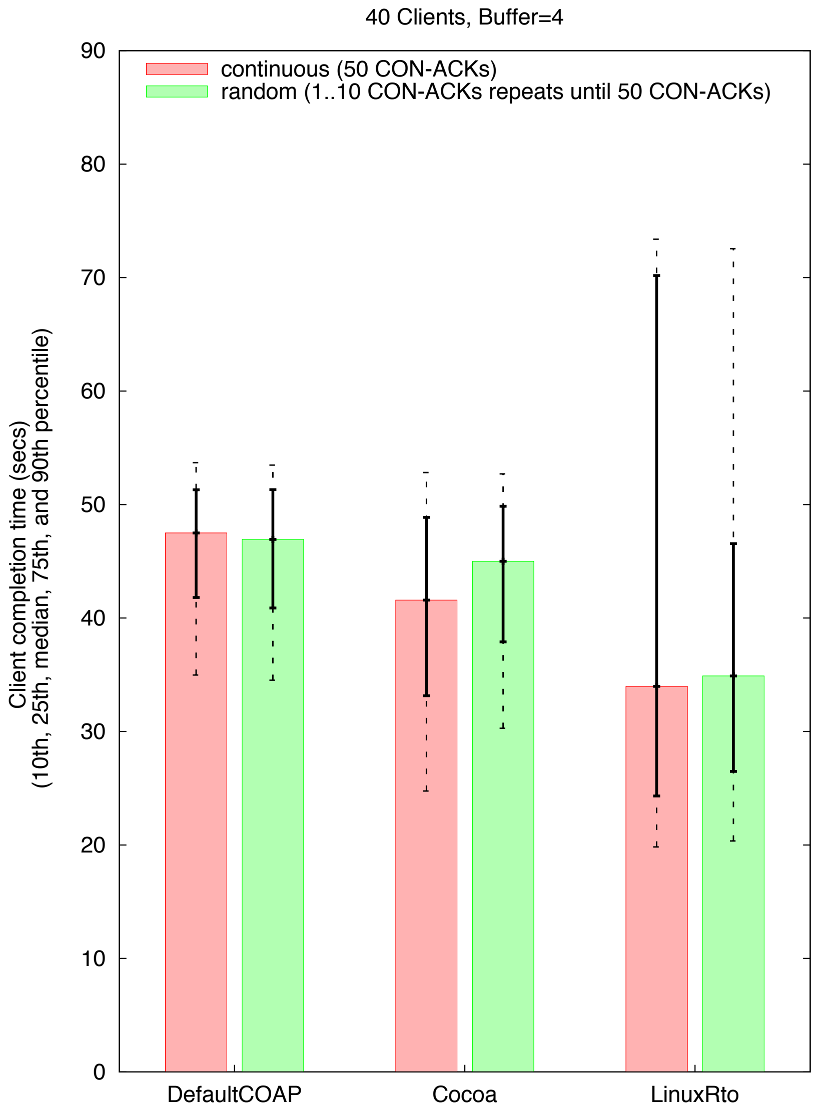
[1] Thanks to Simula Research Laboratory, Norway for sharing their implementation

Test Setup (continued)

- Californium[2]CoAP CC Variants
  - Default CoAP
  - CoCoA
  - LinuxRto: 60 secs max RTO as per Linux implementation and TCP specs
- We increased MAX_RETRANSMIT from 4 (more about this later)
- Workload types
  - Continuous: 50 CON-ACKs
  - Random: 1-10 CON-ACKs after which the CoAP endpoint is changed to reset congestion control state, the process repeated until the total of 50 CON-ACKs
  - Competition: half of the clients using Default CoAP, other half using other CC algorithm (with both Continuous and Random)

[1] Based on Release 1.0.0-M3 / commit id 0f2fe48d5

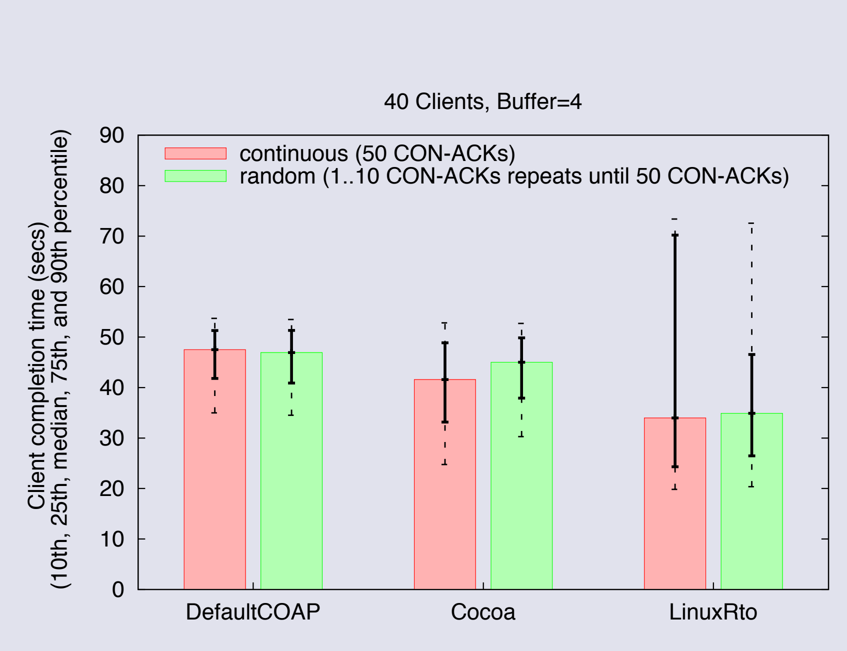
Results: Client Completion Time



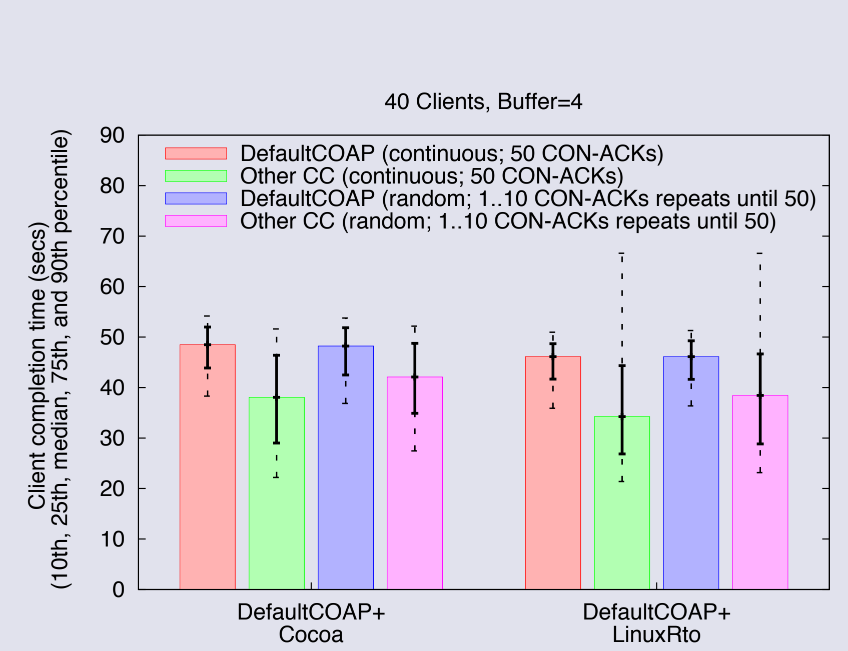
- Both CoCoA and LinuxRto have better client completion time than DefaultCOAP
- LinuxRto has the best median client completion time, however, it is also less stable than the others
- We also tried LinuxRto with MinRTO=1 sec
  - More conservative approach increases client completion time slightly
  - Better stability of client completion time

Results: Competition

CC Algorithm Alone



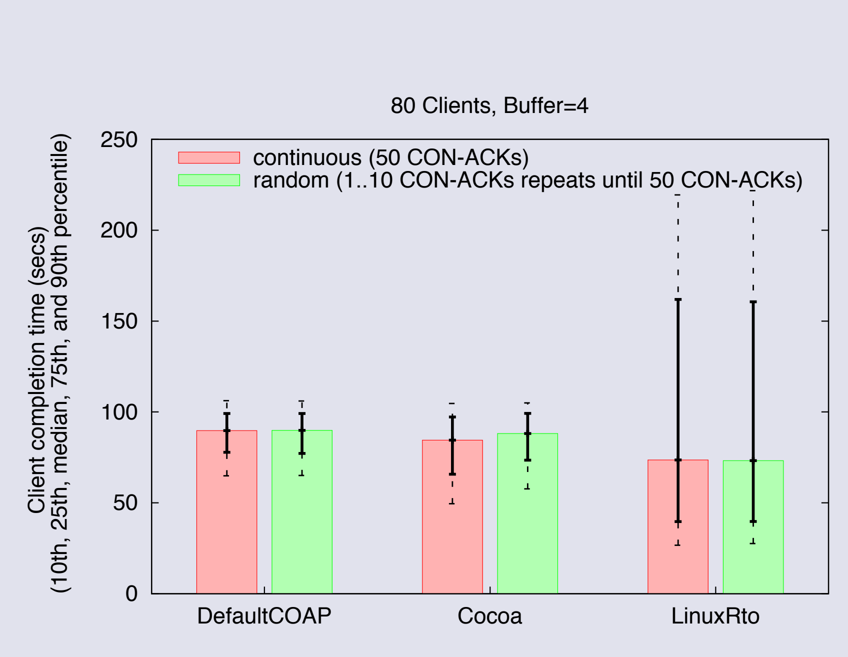
CC Algorithms Competing



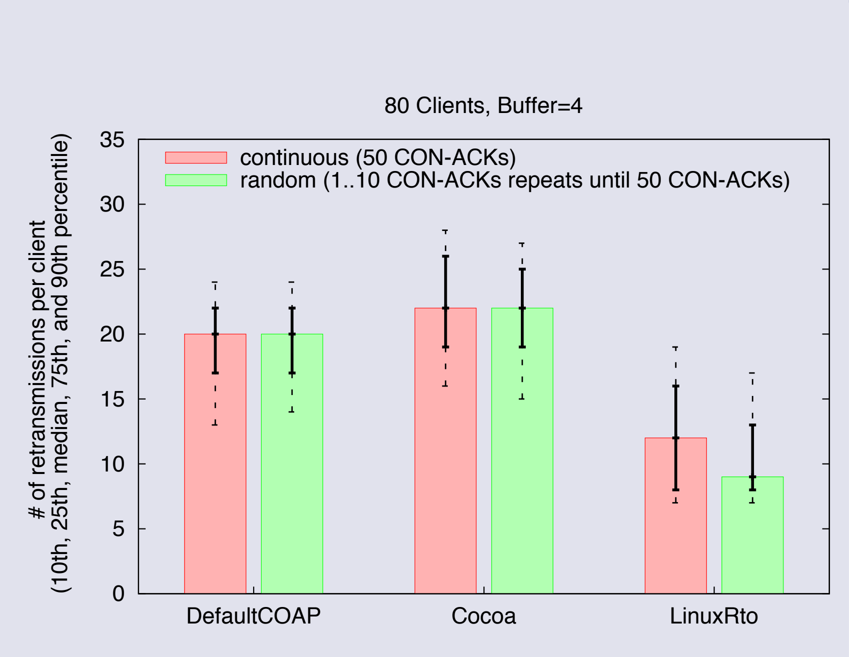
- CoCoA and LinuxRto have smaller client completion time when competing DefaultCOAP, as expected
- No notable harm done for DefaultCOAP by the competing, more aggressive CoCoA or LinuxRto clients

Results: Number of Rexmits

Client Completion Time



Number of Rexmits per Client



- LinuxRto uses less retransmission than CoCoA, and has lower median client completion time
- With larger buffers the differences reduce

Results: Frequency of Rexmits

80 Clients, buffer=4										
Workload	CC algo	Orig.	Retransmissions							
		trans	1st	2nd	3rd	4th	5th	6th	7th	8th
continuous	Def.COAP	117880	28663	9276	2953	897	285	46	0	0
	Cocoa	112454	31757	10503	3546	1263	373	95	9	0
	LinuxRto	133643	18403	3182	1095	614	503	385	282	157
random	Def.COAP	117741	28646	9452	2920	918	276	47	0	0
	Cocoa	113466	31467	10240	3366	1051	340	67	3	0
	LinuxRto	137453	15419	2548	902	558	524	395	306	159

- With CoCoA, lower end of rexmits is larger than with LinuxRto and vice-versa for the higher end
- The default MAX_RETRANSMIT=4 is clearly too small
    - Even with 20 clients, too small for a few CONs

Thoughts / Discussion Items

- Larger MAX_RETRANSMIT
  - Tradeoff for better reliability when heavily congested
  - Less need for adhoc retry workarounds
- Does CoCoA seem justified against well proven Linux RTO
  - Weak estimator, for what purpose?
    - Increases complexity slightly
    - Longer RTO in case of non-congestion losses (untested)?
  - Even if RTO is more convervative, more total retransmissions when congested!
  - Backoff * 1..ACK_RANDOM_FACTOR necessary?

MAX_RETRANSMIT: Californium Implementation Issues

- Increasing MAX_RETRANSMIT
  - In Californium other time related configs also needed to be increase for the increase to have any effect
  - We also tried retry after error on client side using same the same CoAP endpoint
    - Californium congestion control alternatives do not handle such retry consistently
- Increasing MAX_RETRANSMIT beyond 9 triggers yet another set of issues with Californium LinuxRto
  - No timeout but retransmits all remaining retransmits immediately
  - Perhaps with other CCs too, untested as tests had too little congestion to trigger enough retransmissions with CoCoa or Default COAP

Thank you

See also

I. Järvinen, L. Daniel, and M. Kojo, Experimental Evaluation of Alternative Congestion Control Algorithms for Constrained Application Protocol (CoAP), In the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). December 2015. (to appear).

All times are in time-warped JST

Friday

- **09:00–09:05 Intro**
- **09:05–09:25 Delay Attacks (JM)**
- **09:25–10:00 Object Security (GS)**
- **10:00–10:30 CoCoA (CG, IJ)**
- **10:30–10:40 Mostly-offline nodes (PV)**
- **10:40–11:30 Flextime**

CoRE working group

Sleepy Nodes
draft-zotti-core-sleepy-nodes-04

T. Zotti, P. van der Stok, E. Dijk

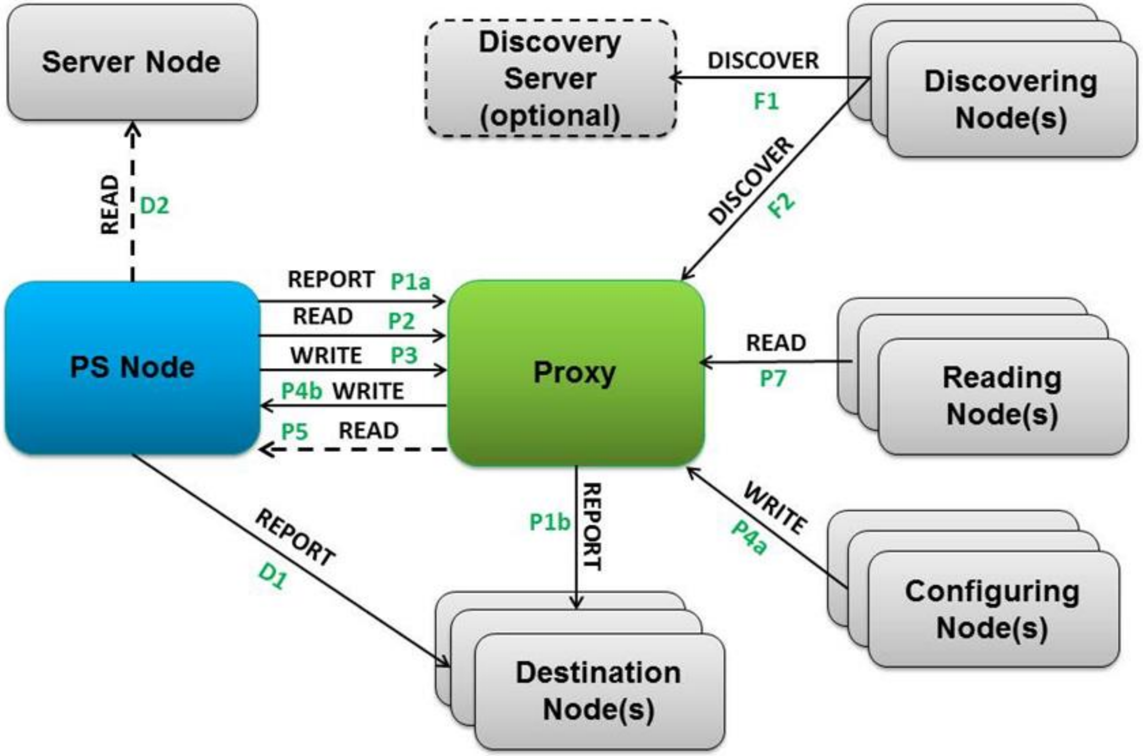
Changes with respect to 03

Many thanks to Matthieu Vial who allowed us to copy large parts of text and examples from draft-vial-core-mirror-server-01

Thanks to Rahman Akbar for review

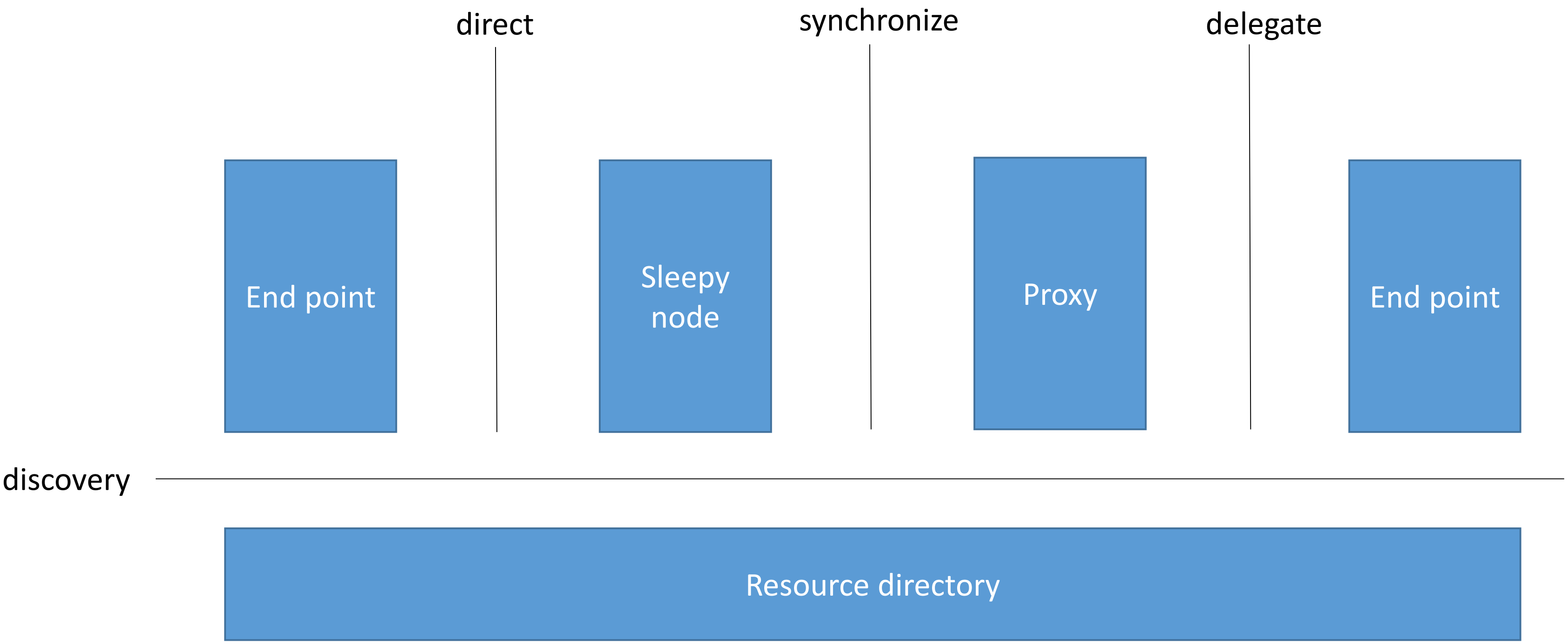
- Comparison with PubSub Broker completed.
- Mistakes in examples removed.
- Less dependence on 6LowPAN networks.
- Added Design motivation section.

Role of nodes around sleepy node



November 6, 2015

Interfaces around sleepy node



direct

synchronize

delegate

End point

Sleepy node

Proxy

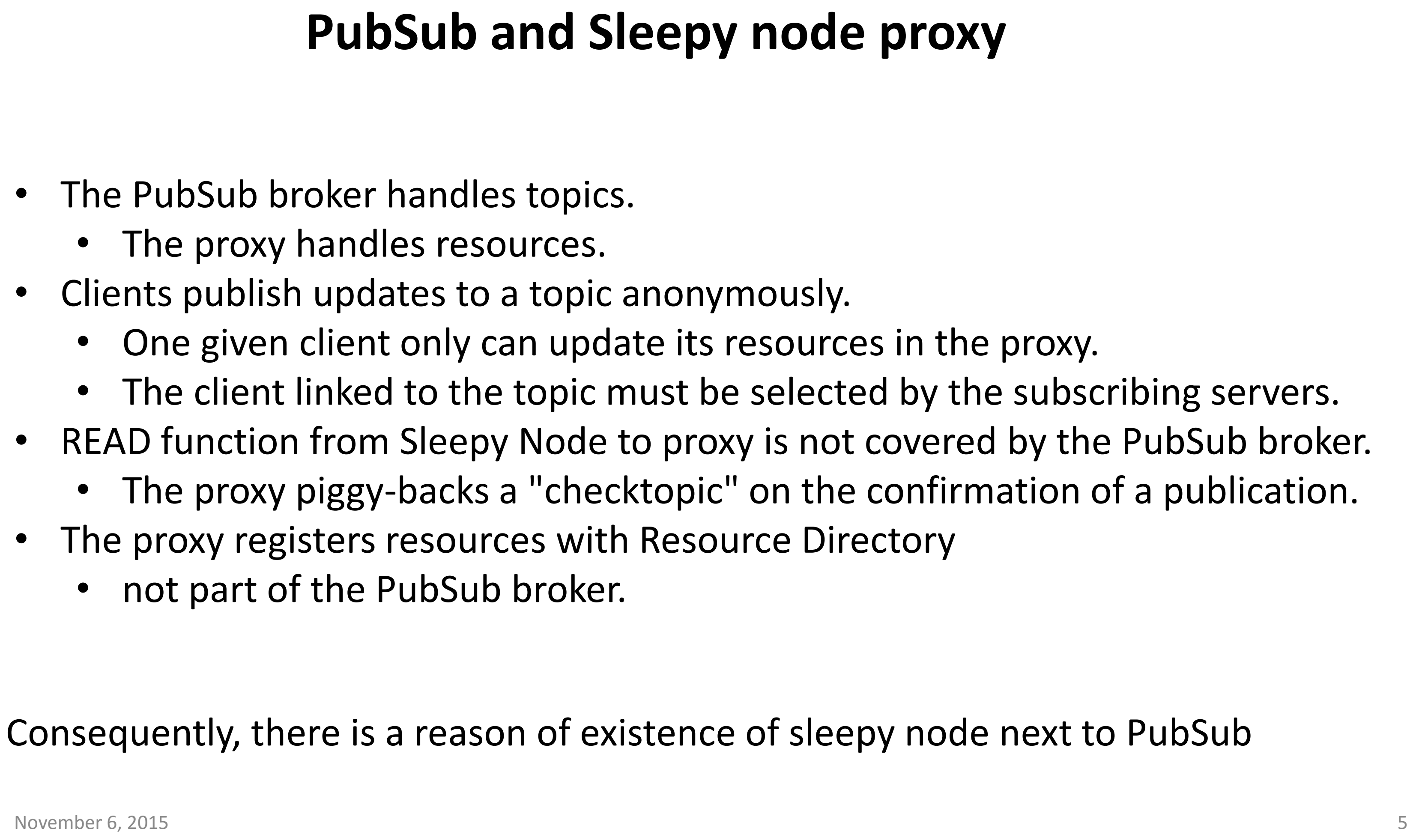
End point

discovery

Resource directory

November 6, 2015

4



PubSub and Sleepy node proxy

- The PubSub broker handles topics.
  - The proxy handles resources.
- Clients publish updates to a topic anonymously.
  - One given client only can update its resources in the proxy.
  - The client linked to the topic must be selected by the subscribing servers.
- READ function from Sleepy Node to proxy is not covered by the PubSub broker.
  - The proxy piggy-backs a "checktopic" on the confirmation of a publication.
- The proxy registers resources with Resource Directory
  - not part of the PubSub broker.

Consequently, there is a reason of existence of sleepy node next to PubSub

What now?

Does the WG agree with the approach of this I-D?

Flextime