

Constrained RESTful Environments WG (core)

Chairs:

Andrew McGregor <andrewmcgr@gmail.com> *) NEW

Carsten Bormann <cabo@tzi.org>

Mailing List:

core@ietf.org

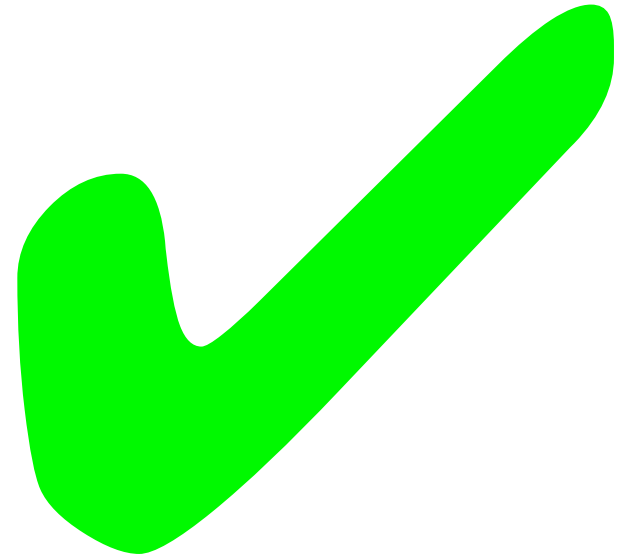
Jabber:

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)

RFC 6690



link-format was published on
2012-08-07

Milestones (from WG charter page)

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

Document submissions to IESG:

- **Dec 2012 CoAP protocol specification** with mapping to HTTP Rest API **to IESG**
- **Feb 2013 Blockwise transfers in CoAP to IESG**
- **Feb 2013 Observing Resources in CoAP to IESG**
- **Apr 2013 Group Communication for CoAP to IESG**
- **Dec 2009 HOLD (date TBD) Constrained security bootstrapping specification to IESG**

SOLACE

- After kicking around the { security bootstrapping / key management / commissioning } problem around between WGs for half a decade
- let's approach the actual problem
 - it's not just a single layer!
- solace@ietf.org
→ <https://www.ietf.org/mailman/listinfo/solace>
- Discuss this in [SAAG, Thu 1510–1710](#) and on Friday
- Somewhat related: [COMAN \(Friday\)](#)

LWIG

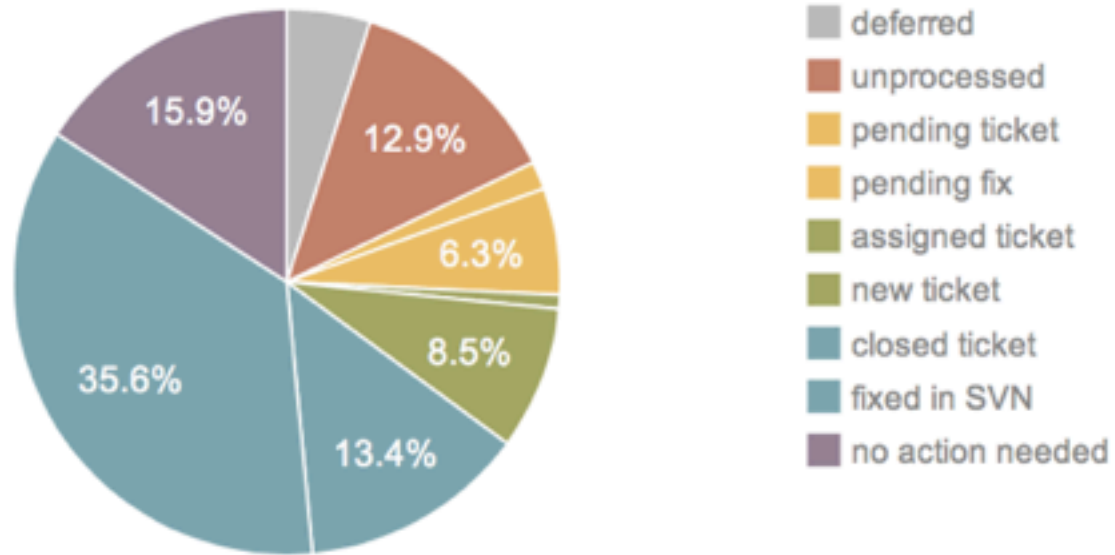
- **Met Monday, 17–19**
- **Planning to extract terminology into a separate, expedited document**
- **draft-ietf-lwig-guidance-02.txt**
- **draft-kovatsch-lwig-class1-coap-00.txt**
- **draft-tschofenig-lwig-tls-minimal-01.txt**

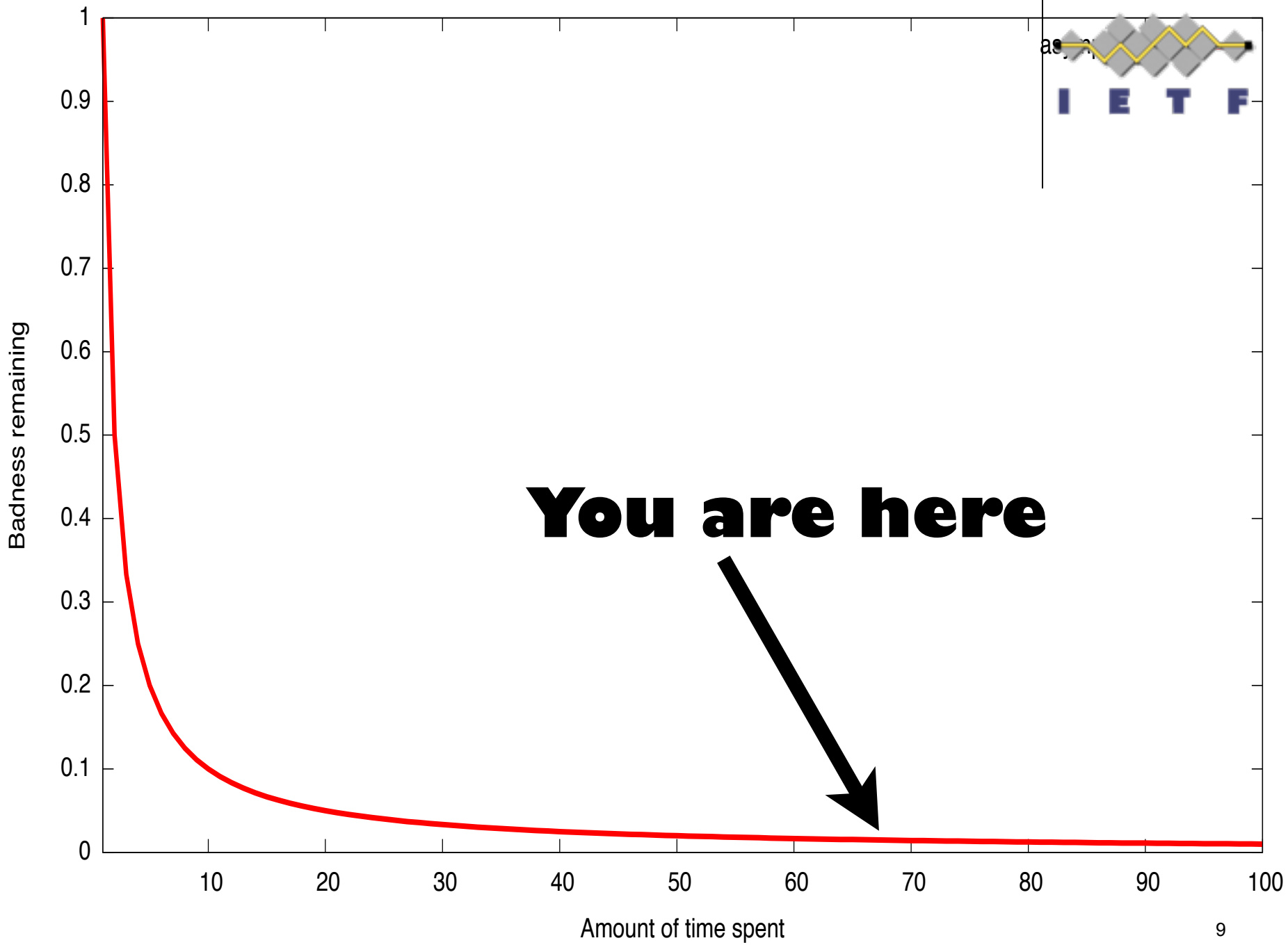
coap-12, block-10, observe-07 post WGLC

- **over 300 comments**
- **most important core-coap comments are covered in coap-12**
 - including one more breaking change we made in Vancouver
 - some work does remain
- **observe-07 mostly updated**
 - need to finish optimism model
- **block-10 mostly updated, but**
 - waiting for grand editorial rewrite
 - one last issue raised: initiative

Summary: 365 items in total | 18 are deferred | 47 are unprocessed | 6 need a ticket | 23 need to be fixed

32 x coap-09 | 15 x block-08







COAP PLUGTEST OVERVIEW

ETSI Centre for Testing and Interoperability

- ETSI, IPSO Alliance and Probe-IT are organising a 2nd IoT CoAP Plugtest and a workshop
 - Workshop on 27 November 2012
 - Interop event : 28 to 30 November 2012
 - Location : Sophia-Antipolis (France, between Nice and Cannes)
- Interop testing scope:
 - CoAP-12 Testing based on updated base specifications (CORE, LINK, OBSERV, BLOCK)
 - Additional features
 - Proxy Caching
 - Security DTLS
 - IPSO Application Framework
 - Full set of options
 - More conformance sessions during the Interop Plugtests event

groupcomm-03

- **Informational**
- **Background**
- **Group Definition and Naming**
- **Group Discovery and Member Discovery**
- **Group Resource Manipulation**
- **Configuring Group Membership In Endpoints**
- **Congestion Control**
- **CoAP Multicast and HTTP Unicast Interworking**
- **Use Cases and Corresponding Protocol Flows**
- **Deployment Guidelines**







draft-ietf-core-block	-10	2012-10-21	Active
draft-ietf-core-coap	-12	2012-10-01	Active
draft-ietf-core-groupcomm	-03	2012-10-19	Active
draft-ietf-core-observe	-07	2012-10-22	Active
Published:			
Draft name	Rev.	Dated	Status
draft-ietf-core-link-format	-14	2012-06-01	RFC 6690

Related Active Documents (not working group documents):

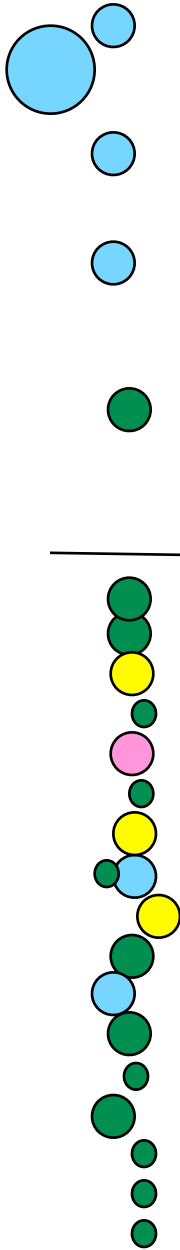
(To see all core-related documents, go to [core-related drafts in the ID-archive](#))

draft-arkko-core-cellular	-00	2012-07-09	
draft-arkko-core-dev-urn	-03	2012-07-09	
draft-becker-core-coap-sms-gprs	-02	2012-07-15	
draft-bormann-core-coap-misc	-21	2012-10-02	
draft-bormann-core-coap-block	-01	2010-10-24	replaced by draft-ietf-core-block
draft-bormann-core-cocoa	-00	2012-08-13	
draft-bormann-core-congestion-control	-02	2012-08-01	
draft-bormann-core-links-json	-01	2012-07-14	
draft-bormann-core-roadmap	-03	2012-10-22	
draft-cao-core-pd	-02	2012-07-16	
draft-castellani-core-advanced-http-mapping	-00	2012-07-04	
draft-castellani-core-http-mapping	-06	2012-10-22	
draft-dijk-core-groupcomm-misc	-02	2012-10-19	
draft-doi-core-parameter-option	-01	2012-10-15	
draft-fossati-core-fp-link-format-attribute	-00	2012-07-09	
draft-fossati-core-monitor-option	-00	2012-07-09	
draft-fossati-core-multipart-ct	-01	2012-10-02	
draft-fossati-core-publish-option	-00	2012-07-09	
draft-greevenbosch-core-minimum-request-interval	-00	2012-09-25	ipr
draft-greevenbosch-core-profile-description	-01	2012-10-22	
draft-hartke-core-observe	-02	2010-08-24	replaced by draft-ietf-core-observe
draft-hartke-core-codds	-02	2012-07-16	
draft-he-core-energy-aware-pd	-01	2012-07-16	
draft-jennings-core-transitive-trust-enrollment	-01	2012-10-13	
draft-li-core-coap-patience-option	-01	2012-10-22	
draft-li-core-coap-payload-length-option	-00	2012-05-26	
draft-li-core-conditional-observe	-03	2012-10-22	
draft-lynn-core-discovery-mapping	-02	2012-10-22	
draft-ma-core-stateful-observe	-00	2012-07-30	
draft-rahman-core-groupcomm	-07	2011-10-12	replaced by draft-ietf-core-groupcomm
draft-rahman-core-sleepy	-01	2012-10-16	ipr
draft-rahman-core-sleepy-problem-statement	-01	2012-10-21	
draft-sarikaya-core-shootstrapping	-05	2012-07-10	
draft-shelby-core-comp	-01	2010-05-10	replaced by draft-ietf-core-coap
draft-shelby-core-interfaces	-03	2012-07-11	
draft-shelby-core-link-format	-00	2010-09-28	replaced by draft-ietf-core-link-format
draft-shelby-core-resource-directory	-04	2012-07-16	
draft-sundstok-core-dna	-02	2012-07-16	
draft-vial-core-mirror-proxy	-01	2012-07-13	
draft-vial-core-mirror-server	-00	2012-10-08	

Can't discuss all of them

-  Tue
-  Fri
-  Not discussed
-  LWIG (Mon)

+ IPSO stuff



CoRE @ IETF85

- **WGLC**

- draft-ietf-core-coap-12
- draft-ietf-core-block-10
- draft-ietf-core-observe-07

- **WG documents**

- draft-ietf-core-groupcomm-03 (15)
 - (Background reading: draft-dijk-core-groupcomm-misc-02)

- **Companions:**

- draft-castellani-core-http-mapping-06 (15)
 - (Background reading: draft-castellani-core-advanced-http-mapping-00)

- draft-bormann-core-cocoa
 - (Background reading: draft-bormann-core-congestion-control)

- draft-doi-core-parameter-option (10)

- **New work**

- draft-bormann-core-roadmap-03
- draft-rahman-core-sleepy-problem-statement-01 (15)
- draft-rahman-core-sleepy-01 (05)

Fri:

- Continue this
- IANA policy review
- MUST/SHOULD review

Tue/Fri scheduling

- **Original plan was**
 - **Tue = WGLC documents, Fri = new stuff**
- **Travel plans → a bit more entropy...**

Group 1:WGLC

coap-12, block-10, observe-07

Tickets

- ... are our way to make the steps forward
- are at:
- <http://tools.ietf.org/wg/core>
- Updates are sent to the mailing list
 - **Please review!**
- When we close a ticket, **please review once more!**

Constrained Application Protocol

draft-ietf-core-coap-12

Z. Shelby, K. Hartke, C. Bormann, B. Frank

Progress Since WGLC

- Three revisions of the draft (-10, -11 and -12)
- Closed all major WGLC tickets
- Many editorial improvements
- CoAP-12 is now the stable implementation baseline
- What is left to completion?
 - Process remaining WGLC comments
 - <http://trac.tools.ietf.org/wg/core/trac/browser/wglc/issues.html>
 - Execute a 2nd WGLC
 - Update and ship to the IESG

Tickets Closed in -12

- Added new Jump mechanism for options and removed Fenceposting (#214)
- Added Proxy Unsafe/Safe and Cache-Key masking to option numbers (#241)
- Re-numbered option numbers to use Unsafe/Safe and Cache-Key compliant numbers (#241)
- Added new IANA option number registration policy (#214)
- Defined NSTART and restricted the value to 1 with a MUST (#215)
- Defined PROBING_RATE and set it to 1 Byte/second (#215)
- Defined DEFAULT_LEISURE (#246)

Proposed issues to close in -13

- **IANA multicast update (#247)**
- **Authority Name issues with SNI and X.509 certificate (#255)**
- **Conflicting security requirements in groupcomm/core-coap (#252)**
- Max-Age, Etag MUST IMPLEMENT for proxies (#254)
- Caching text needs to be updated (#256)
- URI references and multicast requests (#257)
- Standardize a workaround for HTTP library limitations in talking to forward HTTP-COAP cross-proxies? (#259)
- **IANA Policy Review (#260)**
- SHOULD Review (#261)
- Split out IPsec details into a separate draft (#262)

IPv4/IPv6 Multicast IANA Issues (247)

- <http://tools.ietf.org/wg/core/trac/ticket/247>
- coap-12 requests
 - Internetwork Control Block for IPv4
 - Variable Scope for IPv6
- IANA has concerns about “All CoAP Nodes” IPv4 multicast
 - IPv6 also on hold to align with IPv4 considerations
- 1. How does this scale when forwarded?
 - Proposed Action: Change IPv4 request to Local Network Control Block. Change IPv6 request to link-local and site-local scope only. Use application specific multicast address for larger scope multicast needs.
- 2. Are there any security considerations?
 - Proposed Answer: Point to existing text in 11.3

#252: multicast security

- **core-coap-12: "CoAP servers SHOULD NOT accept multicast requests that can not be authenticated".**
 - but we don't know how to do that.

#255: Authority names, SNI, and EUI-64

- **core-coap-12: "The Authority Name in the certificate is the name that would be used in the Authority part of a CoAP URI".**
 - but we don't always want to tie authentication to that.
 - e.g., we might use EUI-64 or similar for device IDs.

IANA Policy Review (260)

Registry	Sub-Registry	Range	Policy
CoAP Code	Method	1-31	IETF Review
		32-63	Reserved
	Response Code	64-191	IETF Review
		192-255	Reserved
Option Number		0-255	IETF Review
		256-2047	Specification Required
		2048-65535	Designated Expert
Content Format		0-200	Expert Review
		201-255	“Private Use” → “Experimental”?
		256-65535	First Come First Served

A Use Case for the Content Parameter Option

Yusuke DOI

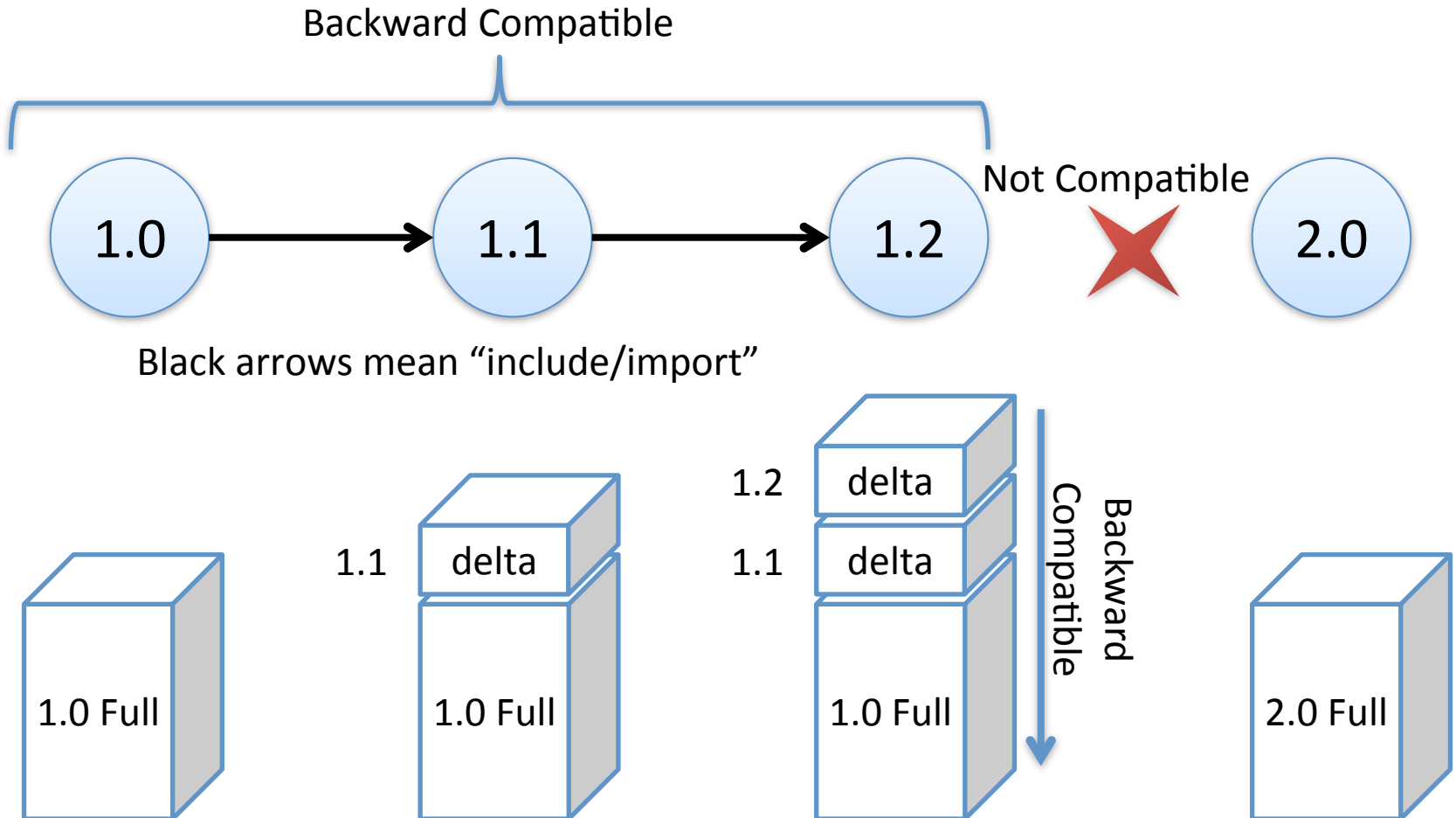
<yusuke.doi@toshiba.co.jp>

Why?

- In CoAP, content-type is a number (Content-Format in core-12)
- But, a parameter is not a format (sometimes)
 - may have *relationships* (e.g. version numbers)
 - may have *range* (e.g. frequency)
- Parameter semantics are up to the media type
 - Is this suitable to manage them (including proprietary parameters) in IANA?

A Use Case: Version Match in EXI

```
Accept: foo-exi; version=(major).(minor)
```



Simple Comparison

- Content-Format per parameters
 - `Accept: foo-exi;`
`version=1.2`
 - `Accept: foo-exi;`
`version=1.1`
 - `Accept: foo-exi;`
`version=1.0`
- Content-Format plus optional parameter
`Accept: foo-exi`
`Accept-CT-Option:`
`(0,version,"1.2")`
- May not be able to have too many revisions
- May have large number of revisions

(if time allows...)

PROPOSALS

Content-Type Parameter Option

- The application has responsibility to handle the semantics of content-type parameter
 - attribute=value
 - *aid*: attribute will be in a small limited set (ID)
 - *value* could be in opaque 8-bit for applications
- CT-Parameter: for Content-Type^wFormat
 - concat(aid,value) // aid=value
- Accept-CT-Parameter: for Accept[idx]
 - concat(idx, aid, value) // idx'th accept option; aid=value

To avoid semantics confusion

- Expected problem: a node need to expect both content-format and parameter for exactly same content type? -> maybe no.
- If there's a matching content-format ID, a node **MUST** use content-format ID.
- If there's no matching content format ID of given content-type+parameter **AND** the parameter is not just an identifier, a node **MAY** use CT-Parameter or Accept-CT-Parameter option in most compact form to describe the parameter.

Choices?

- Option 1: content-format ID per every combination of parameters, forget EXI.
- Option 2: do block assignment on content-format ID to describe revisions.
 - Note: It may require more space on content-format ID
- Option 3: option 1 plus content-format parameter option (or something like that)
- Option 4: content-TYPE ID without parameter and move all parameters to content-format parameter option
 - I don't push the idea, but it's a choice to have a cleaner design on ID/parameter separation.
- Option 5: let applications/media-types to have an option to describe their parameter (e.g. foo-exi to have an option FOO-EXI-VERSION option, bar-exi needs BAR-EXI-VERSION option)
 - If they are local to specific applications, core spec could be untouched
-

-block Tickets

- **Essentially done (editorial work remains):**
 - **#210 Disentangle Block and Token**
 - cleanup
 - **#206 Clarify that atomic Block1 transfers match ~~per-token~~ *and* endpoint**
 - actually, Token is out... (#210)
clarify buffered transfers depend on endpoint, too.
 - **#209 Add potential attacks to security considerations**
 - **#245 Compression vs. Block**
- **“Good idea”**
 - **#211 Signal provisional responses (atomic Block1) in the response code**
 - Special response code for non-committal “I stored this, go on”?
 - 2.xx vs. new class 1.xx? [cf. 100 continue]
- **Initiative issue (next slide):**
 - **#253 Block2 vs. Initiative (Don't call us, we'll call you)**
 - #203 Restrict the potential combinations of Block1 and Block2

The block option: (NR, M, SZ)

- Some resource representations are > MTU bytes
- Transfer in blocks

```

0
0 1 2 3 4 5 6 7
+---+---+---+---+---+---+
|blocknr|M| szx |
+---+---+---+---+---+

```

M: More Blocks

szx: \log_2 Blocksize - 4

```

0                               1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           block nr           |M| szx |
+---+---+---+---+---+---+---+---+---+---+---+---+

```

```

0                               1                               2
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           block nr           |M| szx |
+---+---+---+---+---+---+---+---+---+---+---+---+

```

Decisions:

- Block size is power of 2
- $16 \leq \text{Block size} \leq 2048$

The block protocol: Implementation

- Transfer large representations using **-block**: take apart into multiple exchanges
- **Stateless** transfer:
Server can act on each exchange individually
 - GET for static resource: just get the slice and send it
 - PUT for idle resource: just update the slice in place
- **Buffered** transfer:
Server creates some form of buffer
 - GET for dynamic resource: cache current state (@ block 0)
 - “atomic” PUT: collect data, update on final PUT (M=0)
 - indicated by M bit in response

Look, Ma, no transport!

Block2 vs. Initiative

- In **-block**, *initiative* is with the client
 - Natural for Block1
 - Most useful for stateless Block2
 - Switch to server initiative for Block1, Observe
- **Server initiative is also natural for buffered Block2**
- **Add a way for Server to take initiative?**

-observe Tickets

- **Essentially done (editorial work remains):**
 - **#221 Occasionally sending CON is not just a security consideration**
 - ties in to congestion control changes in core-coap
 - **#242 Wait for acknowledgement before sending new notification**
 - **#234 Editorial updates to -observe examples**
 - **#235 Avoid extending the base standard retransmission rules**
 - **#237 Multicast → reference groupcomm draft**
- **“Good point”**
 - **#258 Be explicit about the "observe key" [Accept]**
- **Optimism issue (take that offline once more):**
 - **#204 Introduce a minimal version of Pledge**

Group 2: groupcomm

Group Communication for CoAP

Akbar Rahman
Esko Dijk



IETF 85, November 2012

<http://www.ietf.org/id/draft-ietf-core-groupcomm-03.txt>

Summary of Changes (1/2)



- Clarified that a group resource manipulation may return back a mixture of successful and unsuccessful responses (section 3.4 and Figure 6) (#251)
- Clarified that security option for group communication must be NoSec mode (section 6) (#250)
- Added mechanism for group membership configuration (section 3.5) (#249)
- Removed IANA request for multicast addresses (section 7) and replaced with a note indicating that the request is being made in [I-D-ietf-core-coap] (#248)
- Made the definition of 'group' more specific to group of CoAP endpoints and included text on UDP port selection (#186)

Summary of Changes (2/2)



- Added explanatory text (section 3.4) regarding why not to use group communication for non-idempotent messages (i.e. CoAP POST) (#186)
- Changed link-local RD discovery to site-local in RD discovery use case to make it more realistic
- Fixed lighting control use case CoAP proxying; now returns individual CoAP responses as in [I-D-ietf-core-coap]
- Replaced link format I-D with RFC6690 reference
- Various editorial updates for improved readability
- Big thanks to my co-author (Esko Dijk) for diligently generating and updating the Tickets (for all multicast related items across the various I-Ds)

CoAP Group Communication Overview



- Now that we have addressed most of the issues for CoAP Group Communication it will be useful to walk through a detailed use case (and more importantly the corresponding protocol flow)
 - Turning on Lights in a large conference room
 - As detailed in the I-D

Remaining Open Issues



- 1) The [I-D-ietf-core-coap-12] generally indicates that IP Multicast should be done in NoSec mode as in section 9.1 (DTLS-secured CoAP). However in section 11.3 (Risk of Amplification) it still says:
 - o “A server SHOULD NOT accept multicast requests that can not be authenticated”

- 2) In slide10, note that the 3 responses from Router-1 to Light-Switch cannot be distinguished by the Light-Switch. That implies that the Light-Switch doesn't know which of the Lights gave a 5.00 error. Because the response Token will be the same in all 3 responses. And the source IP address of responses will always be address of Router-1.

- 3) Select between the different options in Configuring Group Membership In Endpoints (section 3.5 of Groupcomm)

Request for Volunteer Reviewers



- Can we have 3 (or more) volunteer reviewers to review the Groupcomm I-D before the next IETF?



BACKUP

Definition



- CoAP Group Communication :
 - A source node sends a single CoAP message which is delivered to multiple destination nodes, where all destinations are identified to belong to a specific group
 - The source node may or may not be part of the group
 - The underlying mechanism for group communication is IP multicast
 - The network where the group communication takes place can be either:
 - a constrained network or
 - a regular (un-constrained) network

Protocol Mechanism Summary (1/6)



- IP Multicast can be either Link-Local (LL) or across subnets:
 - LL multicast is supported directly by underlying link layer (e.g. WiFi, Ethernet, etc.)
 - Across subnets, an IP multicast routing protocol needs to be active on routers, and receivers need to subscribe
 - The RPL protocol [RFC6550] for example is able to route multicast traffic in constrained LLNs
 - While PIM-SM [RFC4601] is often used for multicast routing in un-constrained networks
 - Receiver nodes use MLD [RFC3810] to subscribe (and receive) any messages sent to selected IP multicast group

Protocol Mechanism Summary (2/6)



- Group Methods:
 - Group communications SHALL only be used for idempotent messages (i.e. CoAP GET, PUT, DELETE)
 - Group communications SHALL NOT be used for non-idempotent messages (i.e. CoAP POST)
 - The CoAP messages that are sent via group communications SHALL be Non-Confirmable
 - A unicast response MAY be sent back to answer the group request (e.g. response "2.05 Content" to a group GET request)

Protocol Mechanism Summary (3/6)



- All nodes in a given group must be able to process the group communication request. **This will not be the case if there is diversity in the authority port (i.e. a diversity of dynamic port addresses across the group) or if the targeted resource is located at different paths on different nodes.**

Protocol Mechanism Summary (4/6)



- Group URIs:
 - All CoAP multicast requests SHOULD operate only on URIs (links) which were retrieved either from:
 - A `"/.well-known/core"` lookup on at least one group member node
 - Or from equivalent service discovery lookup
 - A group URI must be mappable to a site-local or global multicast IP address via DNS resolution (e.g. `[I-D:vanderstok-core-dna]`)

Protocol Mechanism Summary (5/6)



- Group Ports:
 - All CoAP multicast requests **MUST** be sent either to the **default CoAP port** (i.e. default Uri-Port as defined in [I-D.ietf-core-coap])
 - Or to a port number obtained via a **service discovery lookup** operation being a valid CoAP port for the targeted multicast group

Protocol Mechanism Summary (6/6)



- Congestion Control:
 - Multicast CoAP requests may result in a multitude of replies from different nodes, potentially causing congestion. Therefore **sending multicast requests should be conservatively controlled**:
 - A server MAY choose not to respond to a multicast request if there's nothing useful to respond (e.g. error or empty response)
 - A server SHOULD limit the support for multicast requests to specific resources where multicast operation is required
 - A multicast request MUST be Non-Confirmable
 - A server does not respond immediately to a multicast request, but SHOULD first wait for a time that is randomly picked within a predetermined time interval called the Leisure

CoRE: Constrained RESTful environments

Friday = Casual Day!

The
“how many engineers
does it take
to light up a light bulb”

WG

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

- ✓ Blue sheets
- ✓ Scribe(s)

IPSO Booth @ IETF85
(Bits'n Bites)



CoRE @ IETF85

- **WGLC**

- draft-ietf-core-coap-12
- draft-ietf-core-block-10
- draft-ietf-core-observe-07

- **WG documents**

- draft-ietf-core-groupcomm-03 (15)
 - (Background reading: draft-dijk-core-groupcomm-misc-02)

- **Companions:**

- draft-castellani-core-http-mapping-06 (15)
 - (Background reading: draft-castellani-core-advanced-http-mapping-00)
- draft-bormann-core-cocoa
 - (Background reading: draft-bormann-core-congestion-control)
- draft-doi-core-parameter-option (10)

- **New work**

- draft-bormann-core-roadmap-03
- draft-rahman-core-sleepy-problem-statement-01 (15)
- draft-rahman-core-sleepy-01 (05)
- more...

Fri:

- Continue this
- IANA policy review
- MUST/SHOULD review

Constrained MANagement (COMAN)

Management of Networks with Constrained Devices: Use Cases and Requirements

[draft-ersue-constrained-mgmt-02](#)

IETF #85, Atlanta, USA

mehmet.ersue@nsn.com

dromasca@avaya.com

j.schoenwaelder@jacobs-university.de

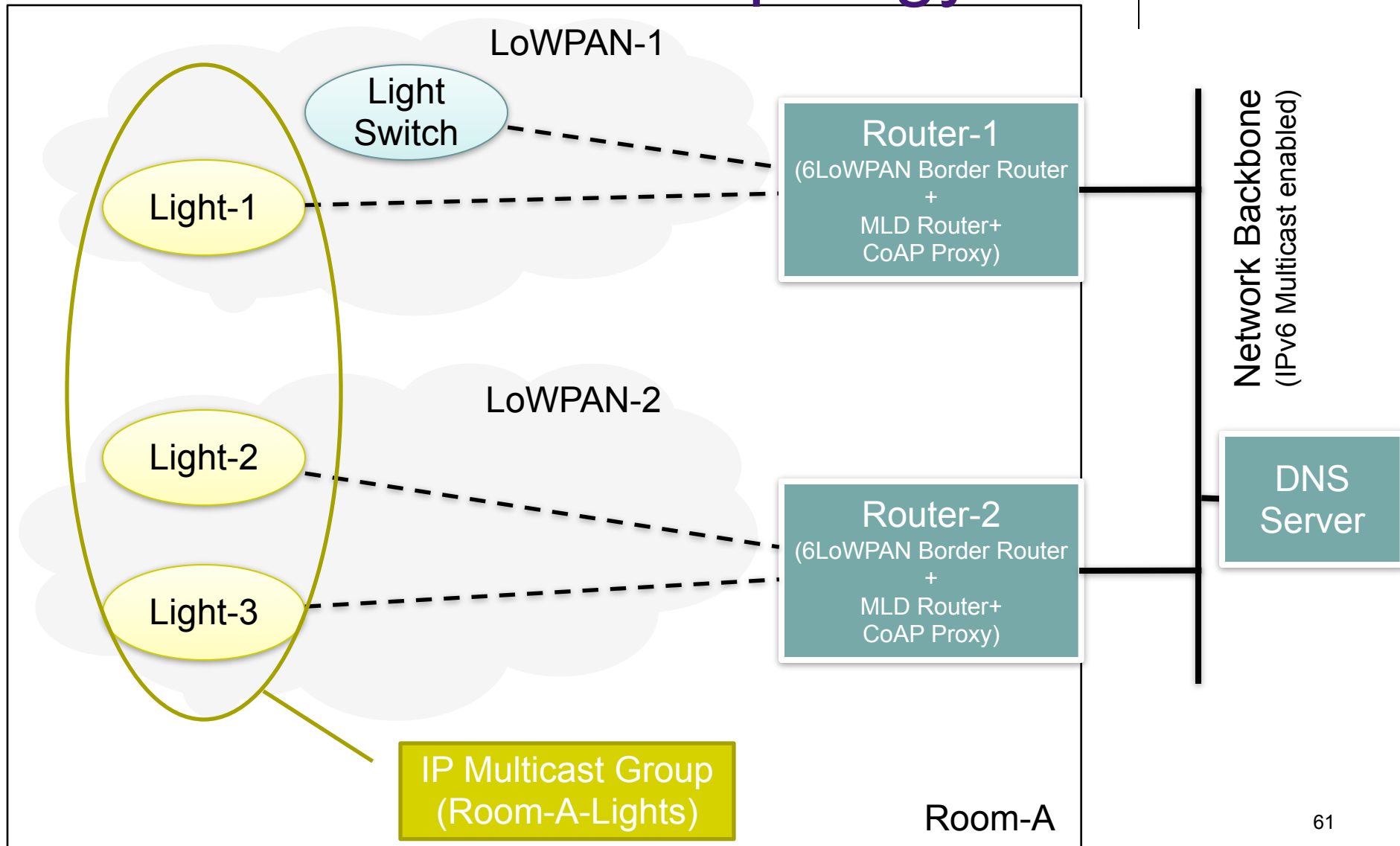
Important new activity

COstrained MANagement (COMAN)

- The aim of the COMAN activity is to . . .
 - provide a problem statement on the issue of the management of constrained devices and the networks with constrained devices.
 - discuss the constrainedness of a network and how it influences the management of devices.
 - raise the questions on and understand the use cases, requirements and the required solution space for the management of constrained devices and the networks with constrained devices.
 - highlight gaps and propose potential new work.
- The current draft provides a long list of requirements as input for discussion.
- The draft will be divided into three pieces after the IETF 85 meeting as the problem statement, use cases and requirements documents.
- We need more constrained network experts as reviewers. Your input is appreciated.
- The discussion is ongoing on the non-wg maillist 'coman': <https://www.ietf.org/mailman/listinfo/coman>
 - PLEASE subscribe to the coman maillist and REVIEW&CONTRIBUTE.

Group 2: groupcomm, part deux

Room-A Network Topology



Turning on lights in Room-A (1/5)



Light-1 Light-2 Light-3 Light switch Router-1 (CoAP Proxy) Router-2 (CoAP Proxy)

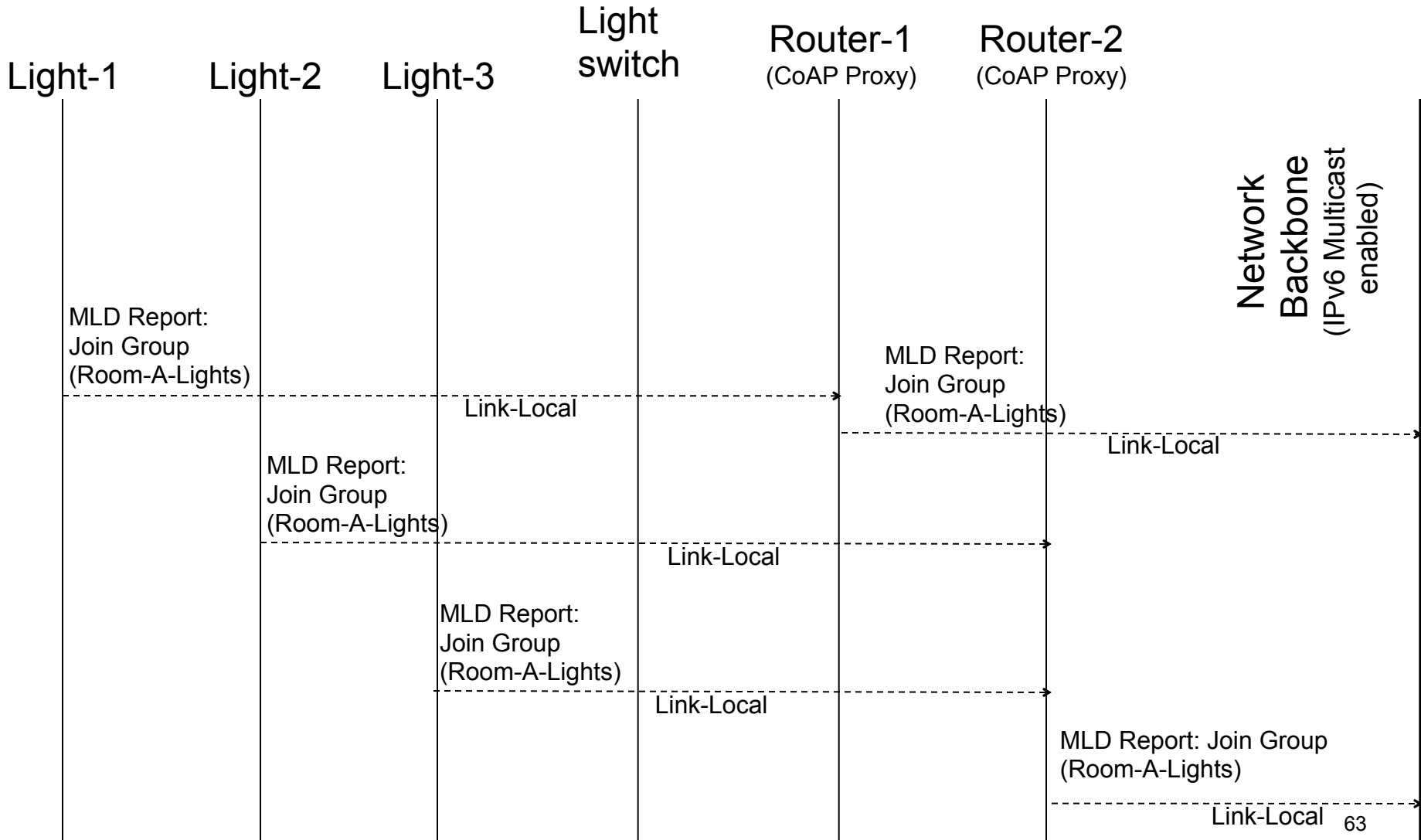
Startup phase

- 6LoWPANs formed
- IPv6 addresses assigned
- CoAP network formed
- Etc.

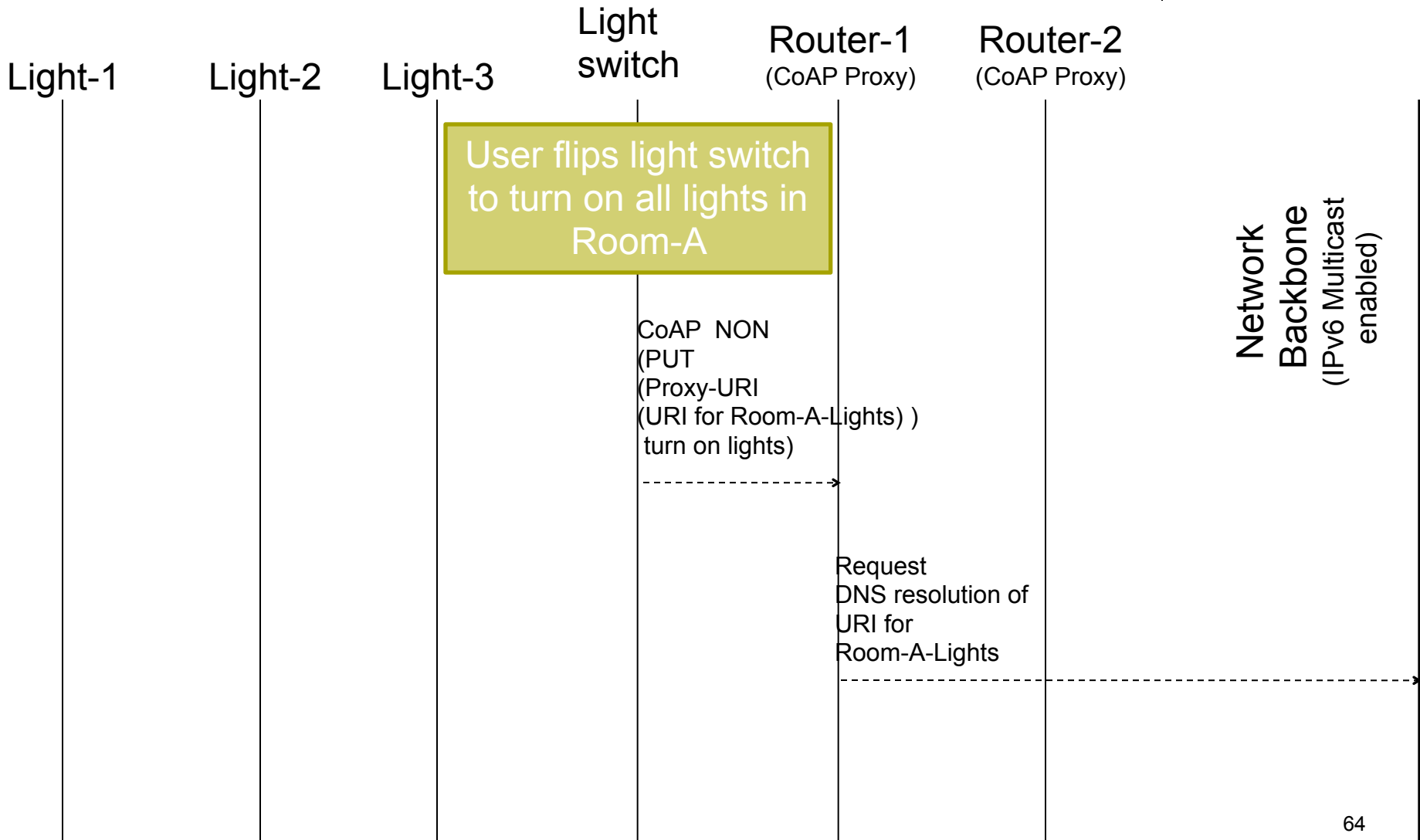
Commissioning phase (by applications)

- Light Switch: URI of group has been set
- Lights: IP multicast address of group has been set
- DNS: AAAA record has been set for the group
- Etc.

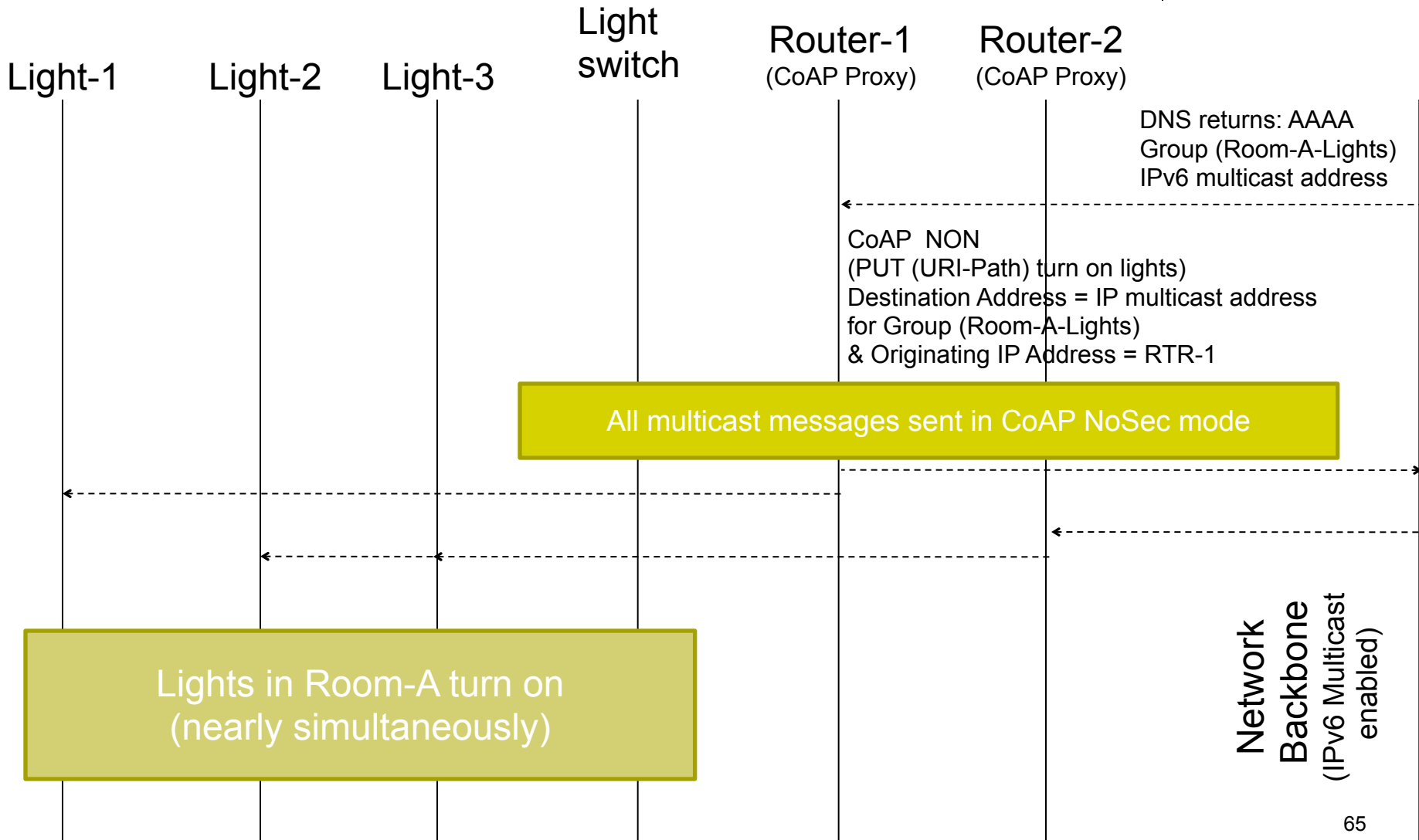
Turning on lights in Room-A (2/5)



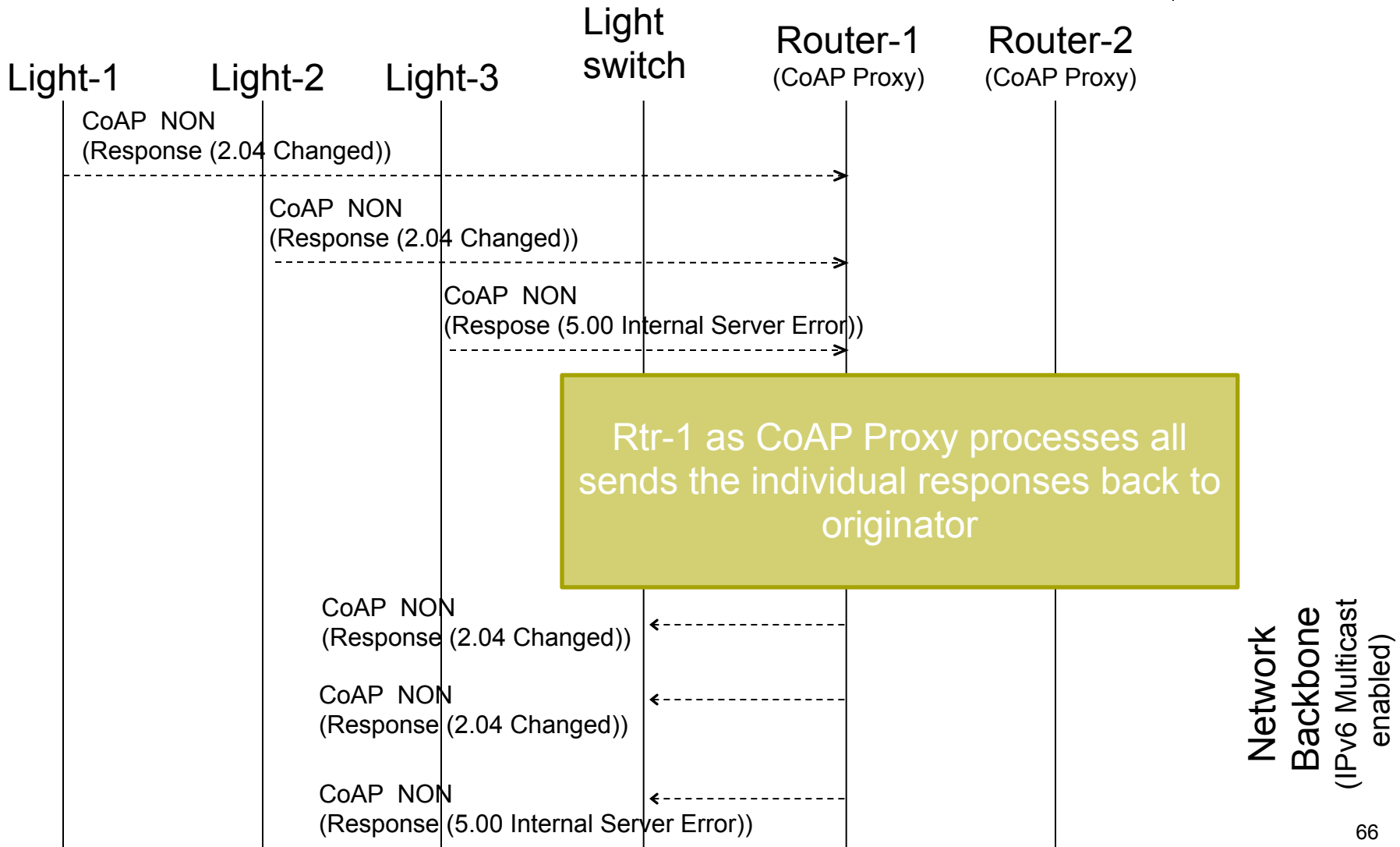
Turning on lights in Room-A (3/5)



Turning on lights in Room-A (4/5)



Turning on lights in Room-A (5/5)



Best Practices for HTTP-CoAP Mapping Implementation

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



IETF 85, November 2012

<http://www.ietf.org/id/draft-castellani-core-http-mapping-06.txt>

Summary of Changes (from -05 rev)



- Addressed detailed review comments from Peter Van der Stok
 - <http://www.ietf.org/mail-archive/web/core/current/msg03758.html>
- Other assorted changes including:
 - Clarified that focus is on Reverse Proxy case
 - Added HTTP-CoAP Response Code translation table
 - Etc.

Goals of I-D



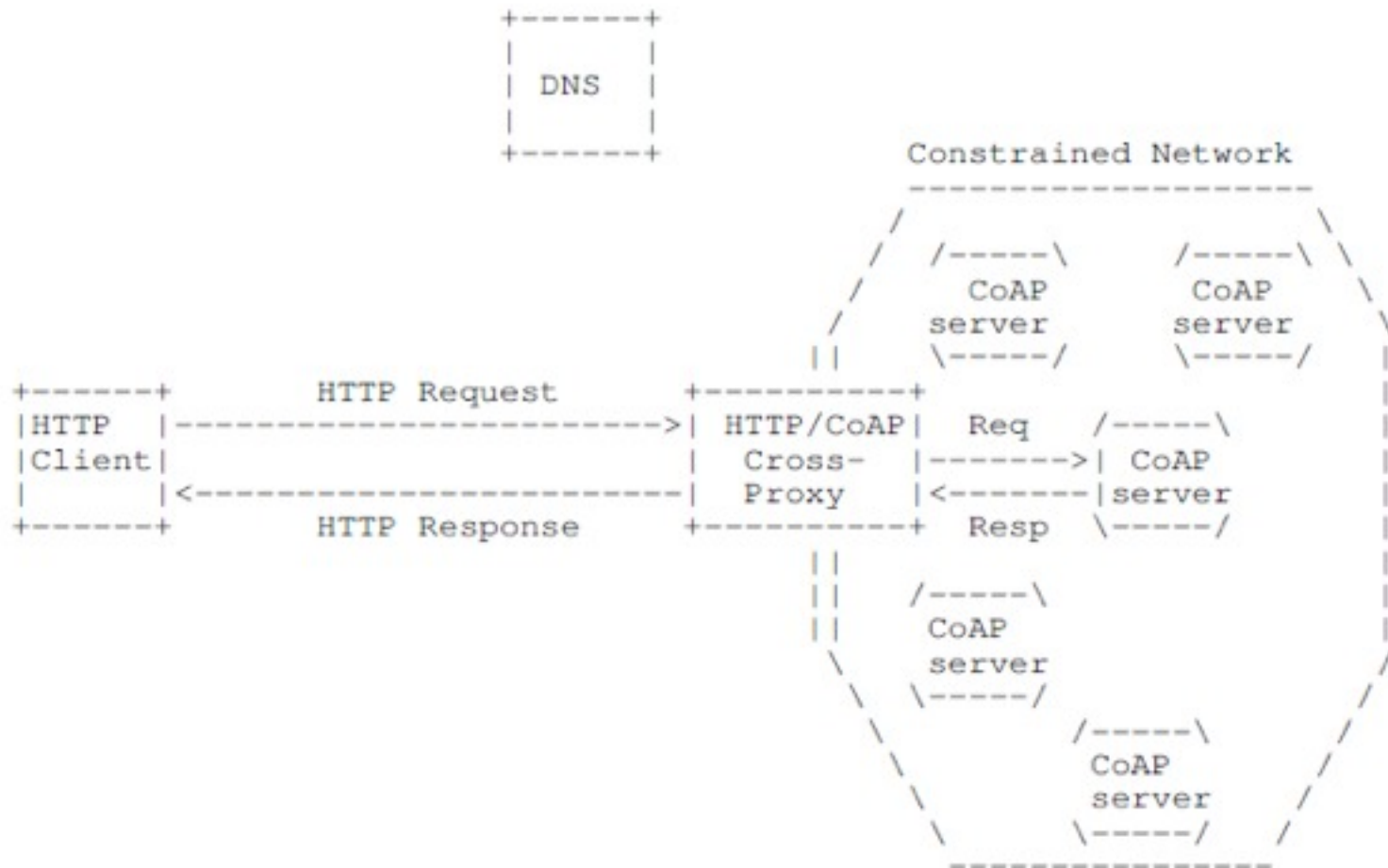
- For Reverse HTTP-CoAP Cross Protocol Proxy:
 - Provide more detailed information to proxy designers (beyond Section 10 of [I-D.ietf-core-coap]), to help implement proxies that correctly inter-work with other CoAP and HTTP client/server implementations that adhere to the specifications
 - Define a consistent set of guidelines that a HTTP-to-CoAP proxy implementation MAY adhere to. The main reason of adhering to such guidelines is to reduce arbitrary (coincidental) variation in proxy implementations, thereby increasing interoperability.
 - As an example use case, a proxy conforming to these guidelines made by vendor A can be easily replaced by a proxy from vendor B that also conforms to the guidelines

I-D Outline



- Guidance on HTTP to CoAP URI mapping
- HTTP-CoAP Reverse cross-protocol proxy implementation
 - Placement
 - Caching and congestion control
 - Response code & media type translations
 - Cache refresh via Observe
 - Use of CoAP blockwise transfer
 - Security translation
- Security Considerations
 - Traffic overflow
 - Handling secured exchanges

Reverse Cross-Protocol Proxy Deployment Scenario



HTTP-CoAP Response Code Mapping



CoAP Response Code	HTTP Status Code	Notes
2.01 Created	201 Created	1
2.02 Deleted	200 OK	2
2.03 Valid	204 No Content	2
	304 Not Modified	3
2.04 Changed	200 OK	4
	200 OK	2
2.05 Content	204 No Content	2
	200 OK	
4.00 Bad Request	400 Bad Request	
4.01 Unauthorized	400 Bad Request	5
4.02 Bad Option	400 Bad Request	6
4.03 Forbidden	403 Forbidden	
4.04 Not Found	404 Not Found	
4.05 Method Not Allowed	400 Bad Request	7
4.06 Not Acceptable	406 Not Acceptable	
4.12 Precondition Failed	412 Precondition Failed	
4.13 Request Entity Too Large	413 Request Repr. Too Large	
4.15 Unsupported Media Type	415 Unsupported Media Type	
5.00 Internal Server Error	500 Internal Server Error	
5.01 Not Implemented	501 Not Implemented	
5.02 Bad Gateway	502 Bad Gateway	
5.03 Service Unavailable	503 Service Unavailable	8
5.04 Gateway Timeout	504 Gateway Timeout	
5.05 Proxying Not Supported	502 Bad Gateway	9

Implementation Experience



- Direct experience from the draft authors:
 - Squid HTTP-CoAP mapping module
 - University of Padova
 - <http://telecom.dei.unipd.it/iot>
 - Both Forward and Interception operation supported
 - HTTP-CoAP proxy based on EvCoAP
 - KoanLogic, University of Bologna and Salvatore Loreto (as individual)
 - <https://github.com/koanlogic/webthings/tree/master/bridge/sw/lib/evcoap>
- The document is open to input from other implementations

Next Steps



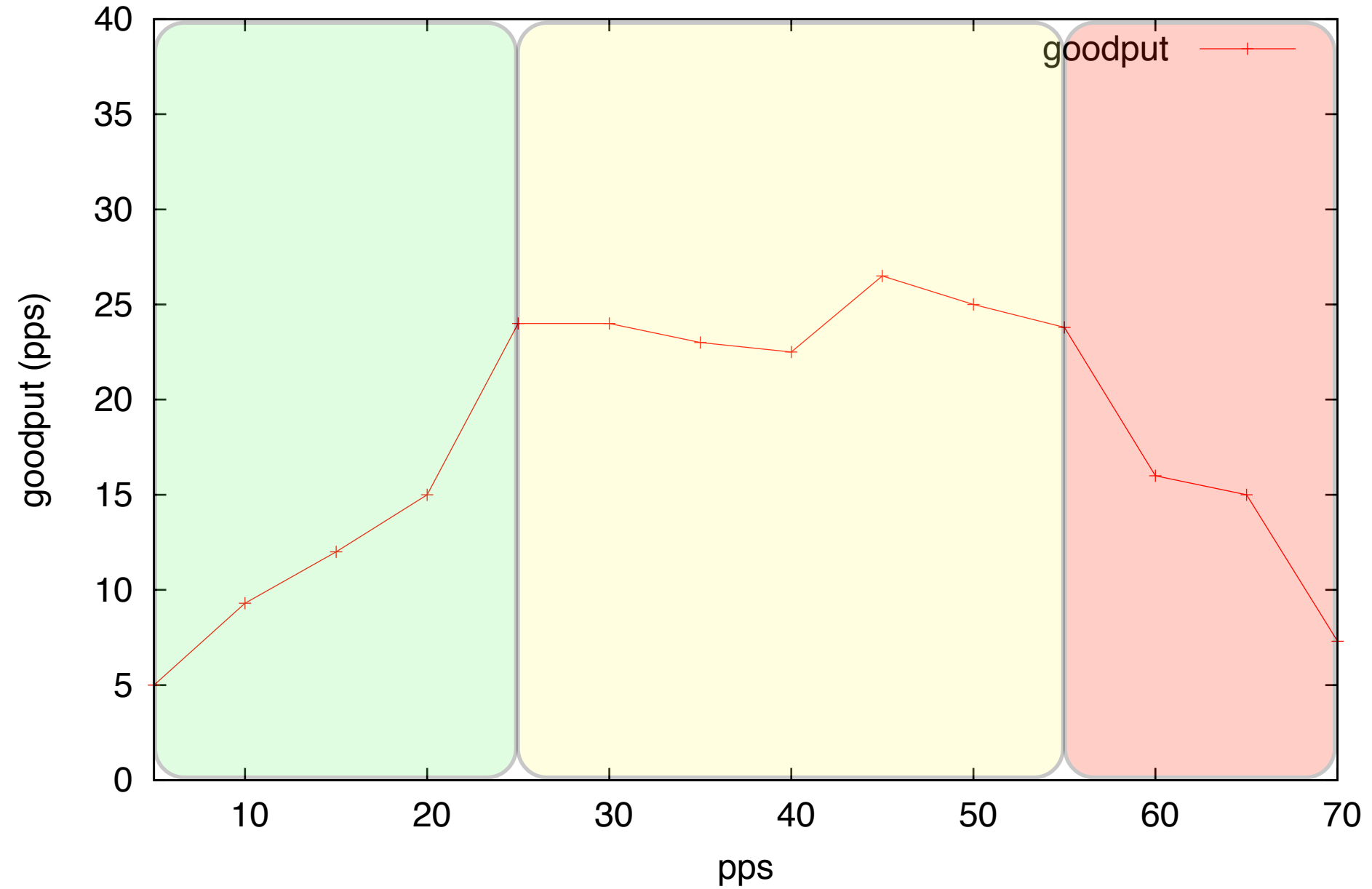
- Does the WG recommend adoption?
 - Intended status: Informational Best Practice
 - Purpose: Reduce arbitrary variation of behavior of proxy implementations

Dealing with congestion issues in CoAP

Carsten Bormann, IETF-85, 2012-11-09



Goodput vs pps (Econotags)



coap-12 (section 4.7): minimal basic congestion avoidance

- **Summary: Can only have 1 outstanding exchange**
- **Rely on existing binary exponential backoff**
- **Ensure RFC 5405 compliance**

- **Sender (usu. client) can add “advanced” cc**
- **Receiver (usu. server) provides enough feedback to enable multiple cc schemes**
- **Experimenting with a couple of them**
- **But do they exist?**

draft-bormann-cocoa-00.txt

- **First shot at “advanced” cc**
- **Idea: Combine BEBO with RTO → one variable**
 - **actually, need to split into weak and strong estimator (ACK ambiguity), ouch**
- **Not yet clear whether that is good enough**
 - **Work continues**
- **Count this as a feeble existence proof?**

Apart from that: Are we done for the base draft?

- **When can we close #215?**
- **Review needed.**

SOLACE

Smart Object Lifecycle Architecture
for Constrained Environments

moved here so Alper can make it

Where do I get my keys?

- IEEE 802.15.4 needs keys
- RPL needs keys
- CoAP/DTLS needs keys

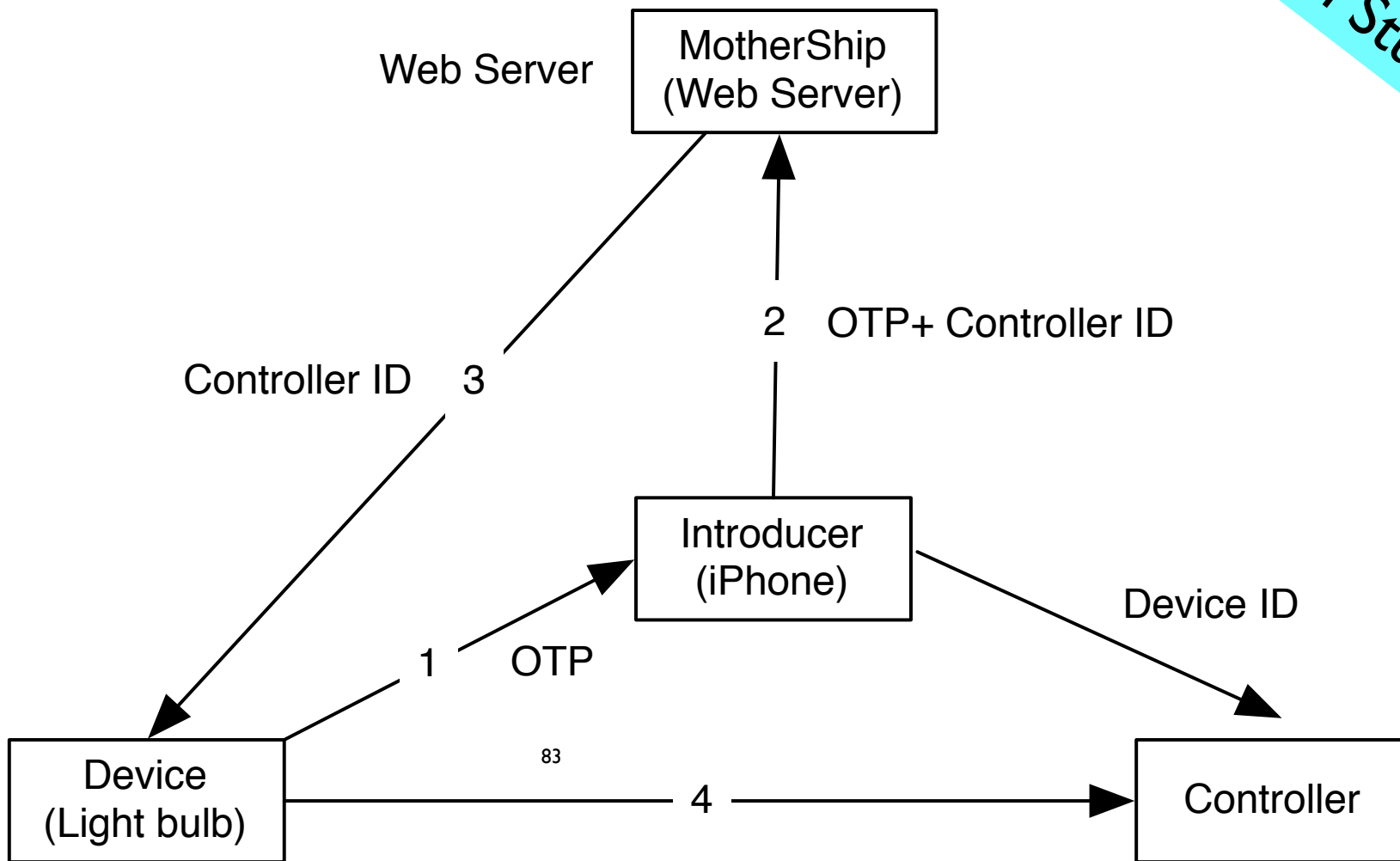
- Lots of desire for key management protocols

Secure Bootstrapping Protocol



- ❑ We have a solution based on EAP-TLS and raw public keys as certificates
- ❑ Based on EAP authentication framework of RFC 5247 (covered in Annex C)
- ❑ EAP-TLS (RFC5216) certificate-based mutual authentication and key derivation protocol that uses TLS
- ❑ draft-ietf-tls-oob-pubkey extends TLS with raw public key support
- ❑ For CoAP devices the usage of X.509-based PKIX certificates is an unnecessary burden
- ❑ CoAP device can be configured with a client public key aka raw public key and use it as certificate
- ❑ Result: simplified authentication, no need for CAs, reduced code size

Cool Stuff



draft-jennings-core-transitive-trust-enrollment-01.txt

What do the keys do?

- Where can I use them?
- What do they authenticate? authorize?
- How do I re-key? get rid of their power?

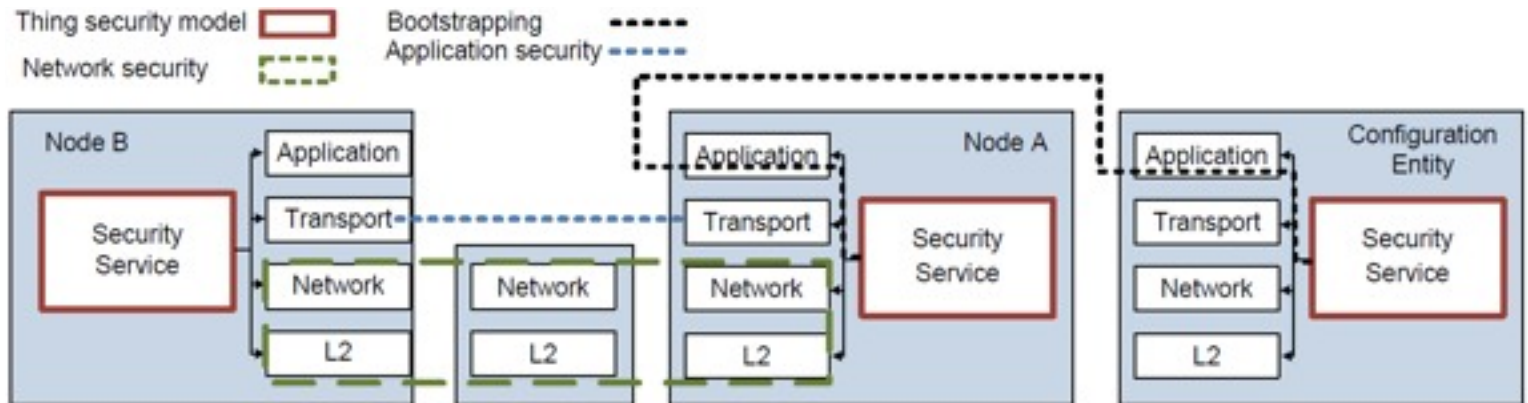
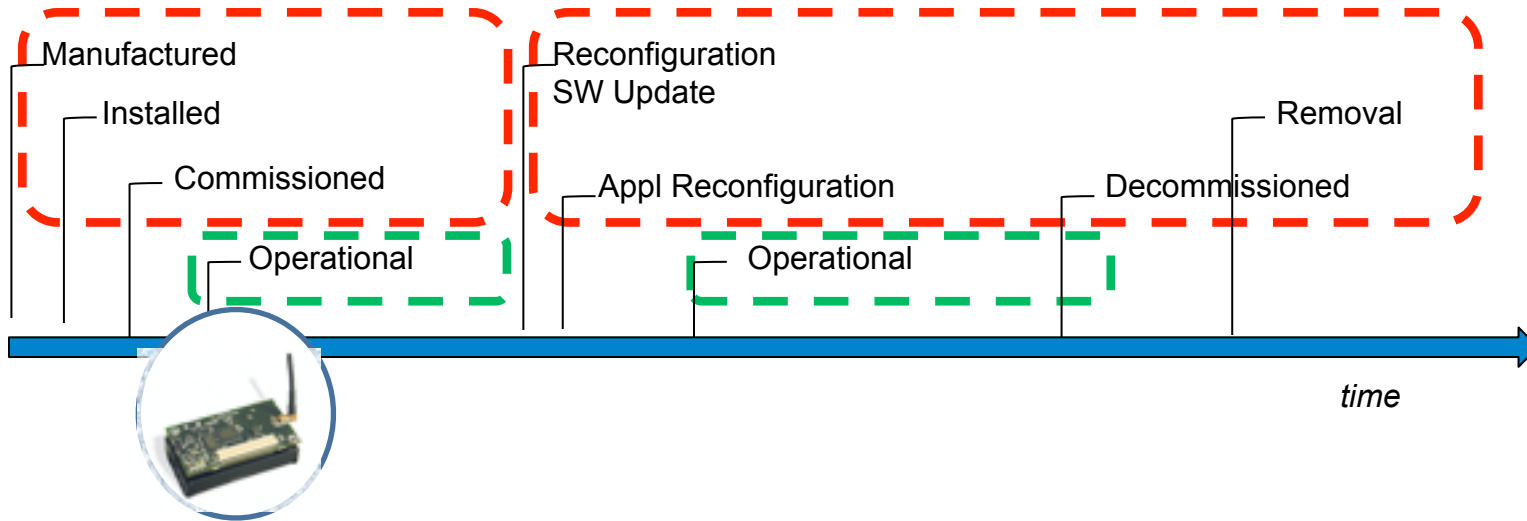
What are my security objectives, anyway?

- There is no security without security objectives
- Who tells us those? When? How?
- Who is authorized to make these decisions? Who did they authorize?
- Who owns stuff? data?

General security objectives

- Not subject to a mass attack
- Usable (yes, Virginia, that is a security objective)
- Channel security
- Authentication of participating entities
- Authorization of access to resources
- Maintains security over a **lifecycle**
- ...

Thing lifecycle and security framework



Objective

- Define enough of the **architecture** so:
 - we know what we are **talking about**
 - and have **terminology** for the components
 - we know when we have the **technology pieces** we need

Technology pieces

- **Cryptographic algorithms:** hash functions, keyed message digest, encryption functions, ...
- **Enrollment:** leap of faith, PAKE, out-of-band provisioning, ...
 - probably most relevant from **usability** p.o.v.
 - stay reasonable/**lightweight** per application
- Security **protocols:** TLS/DTLS, IKEv2, EAP-TLS, ...
- **Credentials:** Raw Public Keys, PSK Identity, X.509 certificates, passwords, ...

SOLACE: Where?

- We bounced it around IETF WGs for half a decade or so
- We got focused again in two **workshops**:
 - IAB Smart Object workshop **2011** <http://tools.ietf.org/html/rfc6574>
 - Smart Object Security workshop **2012** <http://tools.ietf.org/html/draft-gilger-smart-object-security-workshop-00>
- Where to do the work?
 - Start in the **IRTF**, and then do the missing pieces in the **IETF**

SOLACE:

How to start it

- Define one (1) **usage scenario**/use case
- Solicit **contributions** that
 - **spec out** the smart object lifecycle,
from manufacturing via initial keying, establishment of security associations, authorization, configuration, changes to all these (including re-keying), decommissioning (and de-authorization), and recycling/re-use.
 - **considering** network access, routing, and application **layers**
- Discuss and **extract** structure, elements of an architecture

Group 3: “new work”



draft-ietf-core-block	-10	2012-10-21	Active
draft-ietf-core-coap	-12	2012-10-01	Active
draft-ietf-core-groupcomm	-03	2012-10-19	Active
draft-ietf-core-observe	-07	2012-10-22	Active

Published:

Draft name	Rev.	Dated	Status
draft-ietf-core-link-format	-14	2012-06-01	RFC 6690

Related Active Documents (not working group documents):

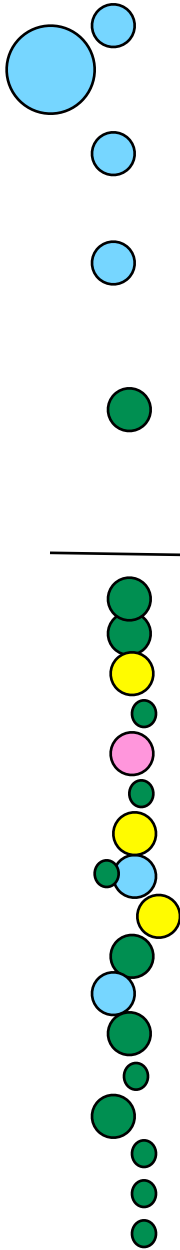
(To see all core-related documents, go to [core-related drafts in the ID-archive](#))

draft-arkko-core-cellular	-00	2012-07-09	
draft-arkko-core-dev-urn	-03	2012-07-09	
draft-becker-core-coap-sms-gprs	-02	2012-07-15	
draft-bormann-core-coap-misc	-21	2012-10-02	
draft-bormann-core-coap-block	-01	2010-10-24	replaced by draft-ietf-core-block
draft-bormann-core-cocoa	-00	2012-08-13	
draft-bormann-core-congestion-control	-02	2012-08-01	
draft-bormann-core-links-json	-01	2012-07-14	
draft-bormann-core-roadmap	-03	2012-10-22	
draft-cao-core-pd	-02	2012-07-16	
draft-castellani-core-advanced-http-mapping	-00	2012-07-04	
draft-castellani-core-http-mapping	-06	2012-10-22	
draft-dijk-core-groupcomm-misc	-02	2012-10-19	
draft-doi-core-parameter-option	-01	2012-10-15	
draft-fossati-core-fp-link-format-attribute	-00	2012-07-09	
draft-fossati-core-monitor-option	-00	2012-07-09	
draft-fossati-core-multipart-ct	-01	2012-10-02	
draft-fossati-core-publish-option	-00	2012-07-09	
draft-greevenbosch-core-minimum-request-interval	-00	2012-09-25	ipr
draft-greevenbosch-core-profile-description	-01	2012-10-22	
draft-hartke-core-observe	-02	2010-08-24	replaced by draft-ietf-core-observe
draft-hartke-core-codds	-02	2012-07-16	
draft-he-core-energy-aware-pd	-01	2012-07-16	
draft-jennings-core-transitive-trust-enrollment	-01	2012-10-13	
draft-li-core-coap-patience-option	-01	2012-10-22	
draft-li-core-coap-payload-length-option	-00	2012-05-26	
draft-li-core-conditional-observe	-03	2012-10-22	
draft-lynn-core-discovery-mapping	-02	2012-10-22	
draft-ma-core-stateful-observe	-00	2012-07-30	
draft-rahman-core-groupcomm	-07	2011-10-12	replaced by draft-ietf-core-groupcomm
draft-rahman-core-sleepy	-01	2012-10-16	ipr
draft-rahman-core-sleepy-problem-statement	-01	2012-10-21	
draft-sarikaya-core-shootstrapping	-05	2012-07-10	
draft-shelby-core-comp	-01	2010-05-10	replaced by draft-ietf-core-coap
draft-shelby-core-interfaces	-03	2012-07-11	
draft-shelby-core-link-format	-00	2010-09-28	replaced by draft-ietf-core-link-format
draft-shelby-core-resource-directory	-04	2012-07-16	
draft-sundstrom-core-dna	-02	2012-07-13	
draft-vial-core-mirror-proxy	-01	2012-07-13	
draft-vial-core-mirror-server	-00	2012-10-08	

Can't discuss all of them

- Tue
- Fri
- Not discussed
- LWIG (Mon)

+ IPSO stuff



Roadmap

- **Give an overview over WG documents**
 - Explain how they fit together
 - Summarize status
- **Document rough edges**
 - Might become an RFC4815-style document (“Corrections and Clarifications”)
- **Try to provide an overview over individual drafts**
 - This doesn’t work without your help!
 - Draft authors: Please review and send comments!
- **Interest???**

Sleepy devices: Problem statement

Salvatore Loreto, Thomas Fossati,
Matthieu Vial, Akbar Rahman



IETF 85, November 2012

<http://www.ietf.org/id/draft-rahman-core-sleepy-problem-statement-01.txt>

Introduction



- The I-D analyzes the COAP protocol issues related to sleeping devices
- The goal of this I-D is to trigger discussions in the CORE WG so that all relevant considerations for sleeping devices are taken into account when designing CoAP (in the various separate solutions I-Ds)

What is a Sleeping Device?



- A Sleeping End Point (SEP) is a device able to cut power to unneeded subsystems and so significantly reduce battery consumption with the important side-effect of reduced CoAP protocol operation during these times
- Different Sleep Modes:
 - Always-On
 - Any sleep is so short as to be invisible to IP layer and above
 - Usual Client-Server model is efficient
 - Intermittent Presence
 - Longer and possibly non-deterministic sleep periods which affects IP layer and above functionality
 - Client-Server model not applicable and new model required

Assumptions



- We assume a Sleeping End Point (SEP) understands and implements the following in a way that is conformant with the CoAP protocol. This will provide the common ground on which to build their integration into the hosting CoAP domain:
 - The concept of information resource and its representational state
 - The semantics and syntax of CoAP URIs
 - The semantics associated with the CoAP methods

Objectives



- The ideal Sleeping End Point (SEP) solution should:
 - Make the set of resource owned and hosted by any SEP available to all the other participants, in the same constrained RESTful environment, without making any assumption on the presence of specific or special entities neither on the network topology
 - Provide the possibility to use Client or Observer Model to access resources owned and hosted by a SEP
 - Allow the (Secure) delegation of resource handling while retaining ownership
 - Minimize the configuration needs to bootstrap a SEP within an existing CoRE domain
 - Maximize the integration with base CoRE Features (i.e. Resource Discovery, Multicast, Observer, Block)
 - Reuse already available CoAP mechanisms as much as possible

Feedback



- Any questions or comments?

Enhanced Sleepy Node Support for CoAP



Akbar Rahman

IETF 85, November 2012

<http://www.ietf.org/id/draft-rahman-core-sleepy-01.txt>

Introduction



- It is expected that in CoAP networks there will be a certain portion of devices that are "sleepy" and which may occasionally go into a sleep mode (i.e. go into a low power state to conserve power) and subsequently have reduced CoAP protocol communication ability
- This I-D proposes a minimal and efficient mechanism building on the Resource Directory concept to enhance sleepy node support in CoAP networks

Current CoAP Support of Sleepy Node (1/2)



- CoAP proxies can use a previously cached response to service a new GET request for a sleepy origin server (as in HTTP)
 - But if no valid cache then proxy has to attempt to retrieve and may fail if origin server is sleeping
 - [I-D.ietf-core-coap]
- Clients can discover list of resources from RD (GET /rd-lookup/...) for sleepy servers
 - But attempt to GET resource from sleepy origin server may fail if origin server is sleeping
 - [I.D.ietf-core-link-format & I.D.shelby-core-resource-directory]

Current CoAP Support of Sleepy Node (2/2)



- Lower layer support for sleepy nodes in most wireless technologies (e.g. WiFi, ZigBee).
 - But limited to MAC packet scheduling for sleepy nodes and not aware of specific needs of IP applications (like CoAP)

Proposal – RD Based Sleep Tracking (1/4)



- The current CoAP approach to support sleepy nodes can be significantly improved by introducing RD based mechanisms for a CoAP client to determine whether:
 - A targeted resource is located on a sleepy server
 - A sleepy server is currently in sleep mode or not
- There is any associated caching Proxy (possibly the RD itself) for a sleepy server

Proposal – RD Based Sleep Tracking (2/4)



- We define the following new RD attributes to characterize the properties of a sleepy node:
 - SleepState - Indicates whether the node is currently in sleep mode or not (i.e. Sleeping or Awake)
 - SleepDuration - Indicates the maximum duration of time that the node stays in sleep mode
 - TimeSleeping - Indicates the length of time the node has been sleeping (i.e. if Sleep State = Sleeping)
 - NextSleep - Indicates the next time the node will go to sleep (i.e. if Sleep State = Awake)
- CachingProxy – Indicates the caching proxy of the sleepy node (i.e. the RD itself or another node)

Proposal – RD Based Sleep Tracking (3/4)



- These attributes are all server (node) level and are new parameters added to the RD URI Template Variables
- Finally, we also define a new lookup-type ("ss") for the RD lookup interface specified in [[I-D.shelby-core-resource-directory](#)].
 - This new lookup-type supports looking up the “SleepState” (ss) of a specified end-point

Proposal – RD Based Sleep Tracking (4/4)



- The three time based parameters (SleepDuration, TimeSleeping, NextSleep) can be based on either an absolute network time (for a time synchronized network) or a relative local time (measured at the local node)
- Following the approach of [[I-D.ietf-core-link-format](#)] and [[I-D.shelby-core-resource-directory](#)], sleep parameters for sleepy servers can be stored by the server in the RD and accessed by all interested clients
- Examples of using these parameters in a synchronous or asynchronous manner are shown in the I-D

Feedback



- Any questions or comments?

draft-greevenbosch-core-profile- description-01

Bert Greevenbosch

Jeroen Hoebeke

Isam Ishaq

Description

- JSON format for signalling the server's capabilities.
- Signalling of supported options, media types and block size.
 - Other items can be added.
- Filtering on specific fields through URI-Query.
- Link: <http://datatracker.ietf.org/doc/draft-greevenbosch-core-profile-description/>

.well-known/profile

- To acquire profile data of resources on a particular service, the [.well-known/profile](#) URI-path is introduced.
- For example, to get all information from sensors served by www.example.org, we can do a GET to <http://www.example.org/.well-known/profile>.
- Filtering on particular resources is done through URI-Queries.

Format

- Currently the following fields are defined:
 - “path”: contains the URI-path associated with a resource;
 - “op”: a numerical list of supported option numbers;
 - “cf”: a numerical list of supported content-format numbers;
 - “b1s”, “b2s”: supported block sizes for Block1 and Block2, respectively.

Example

- On the right is an example of a camera sensor at "coap://www.example.org/cam", that supports the "Uri-Host" (3), "ETag" (4), "Uri-Port" (7), "Uri-Path" (11), "Content-Format" (12), "Token" (19), "Block2" (23) and "Proxy-Uri" (35) options.
- The supported content formats are "text/plain" (0), "application/link-format" (40) and "application/json" (50).
- The supported Block2 can use 256 or 512 byte blocks.

```
Req:
GET coap://www.example.org/.well-known/profile
Res:
2.05 Content (application/json)
{
  "profile":
  {
    "path": "cam",
    "op": [3,4,7,11,12,19,23,35],
    "cf": [0,40,50]
    "b2s": [4,5]
  }
}
```

Filtering through use of URI-Query

- Filtering can be done through the URI-Query.
- Query format: $N=V$
 - N is a profile field;
 - V is the desired value.
- Examples:
 - To find resources that support content format “application/json”:

GET www.example.org/.well-known/profile?cf=50

- To get information about the camera:

GET www.example.org/.well-known/profile?path=cam

Open issues

- Which other profile data needs signalling?
- Fix the order in which the profile fields must appear?
- Inheritance of a profile description?
- Extend usage to signal the client profile?

Thank you!

Questions?

draft-greevenbosch-core- minimum-request-interval-00

Bert Greevenbosch

Description

- The “MinimumRequestInterval” option can be used to indicate the minimum time between two requests in a transaction.
- Originally intended for Block, but also usable for other transactions.
 - Example: browsing a server.
- The goal is to reduce the server load and network traffic.
- Link: <https://datatracker.ietf.org/doc/draft-greevenbosch-core-minimum-request-interval/>

Usage

- The client keeps the server informed about its request speed through inclusion of the “MinimumRequestInterval” option.
- In responses, the server fixes the minimum request interval through the same option.
- The client obeys the speed indicated by the server.
- The client can send slower, but not faster.

Advantages

- The server has means to limit the amount of incoming traffic.
- The server can prevent to become overloaded with too many tasks.
- The server does not need artificially slow down the client by sending late ACKs.
 - No need to keep track of delayed ACKs.
 - The server can perform other tasks instead.
- Reduced network traffic.

Thank you!

Questions?