Constrained RESTful Environments WG (core)

Chairs:

Cullen Jennings <fluffy@cisco.com>

Carsten Bormann <cabo@tzi.org>

With some help from Peter Saint-Andre...

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**
- **Be aware of the IPR principles, according to RFC 3979 and its updates**

✓Blue sheets
✓Scribe(s)

Milestones (from WG charter page)

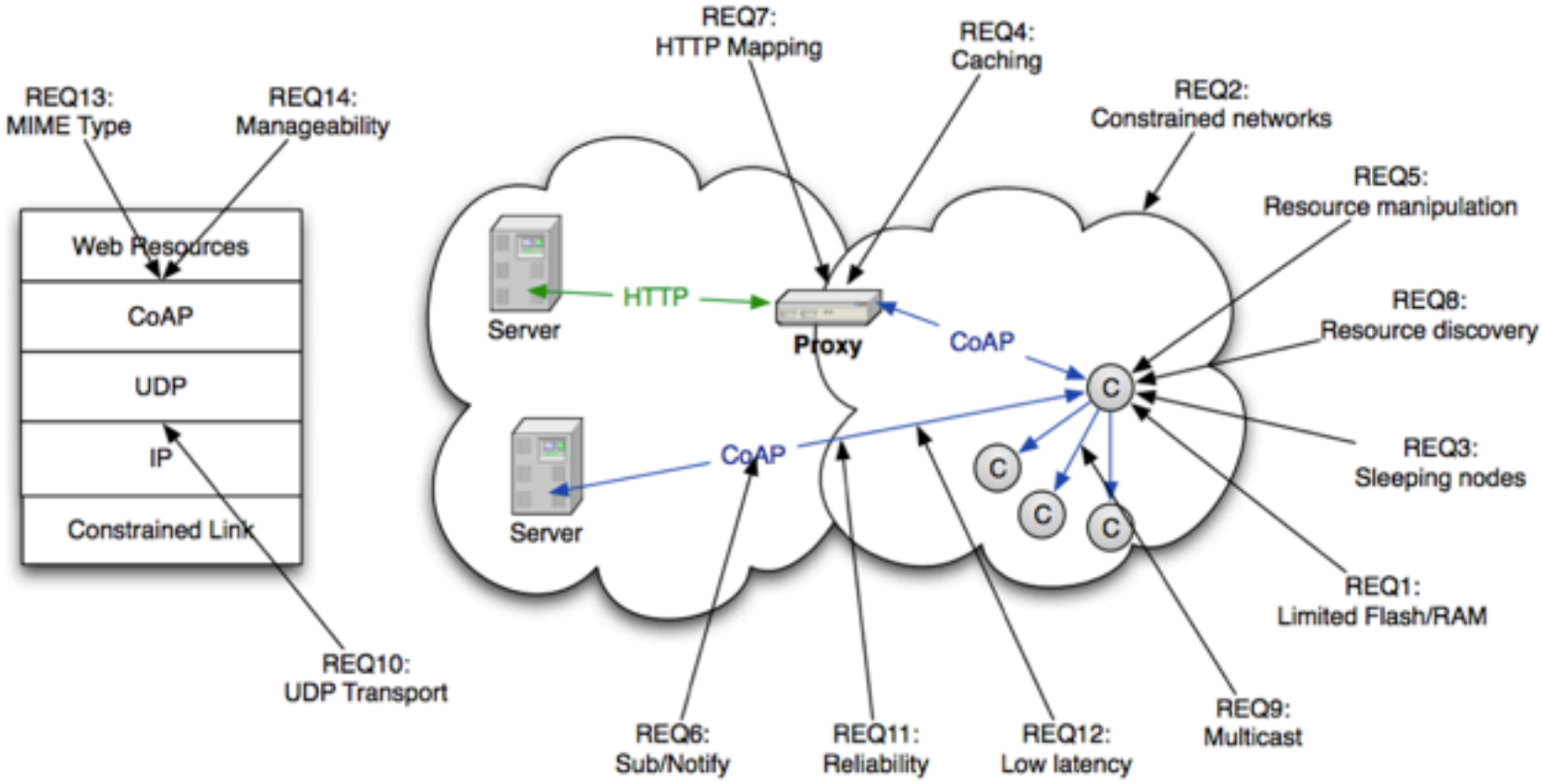
http://datatracker.ietf.org/wg/**core**/charter/

Document submissions to IESG:

- **Apr 2010 Select WG doc for basis of CoAP protocol**
- **Dec 2010 1 – CoAP spec+ with mapping to HTTP REST submitted to IESG as PS**
- **Dec 2010 2 – Constrained security bootstrapping spec submitted to IESG as PS**

- **Jan 2011 Recharter to add things reduced out of initial scope**

CoAP: Meeting the requirements



http://6lowapp.net

Drafts

http://tools.ietf.org/wg/core/

Draft name	Rev.	Dated	Status	Comments, Issues
Active:				
draft-ietf-core-link-format	-01	2010-10-25	Active	4/4
draft-ietf-core-coap	-03	2010-10-25	Active	11/37
draft-ietf-core-observe	-00	2010-10-18	Active	9/9
draft-ietf-core-block	-00	2010-10-18	Active	4/4
Related Active Documents (not working group documents):				
(To see all core-related documents, go to core-related drafts in the ID-archive)				
draft-vanderstok-core-bc	-02	2010-10-25		
draft-rahman-core-groupcomm	-01	2010-10-25		
draft-oflynn-core-bootstrapping	-02	2010-10-19		
draft-shelby-core-coap-req	-02	2010-10-18		
~~draft-shelby-core-link-format~~ replaced by *draft-ietf-core-link-format*	-00	2010-09-28		
~~draft-hartke-coap-observe~~ replaced by *draft-ietf-core-observe*	-02	2010-08-24		
draft-bormann-core-coap-block	-00	2010-08-24		
draft-bormann-coap-misc	-06	2010-08-24		
draft-martocci-6lowapp-building-applications	-01	2010-07-08		
draft-rahman-core-sleeping	-00	2010-06-29		
draft-eggert-core-congestion-control	-00	2010-06-23		
draft-moritz-6lowapp-dpws-enhancements	-01	2010-06-16		
~~draft-shelby-core-coap~~ replaced by *draft-ietf-core-coap*	-01	2010-05-10		

http://6lowapp.net

CoAP Plugfest Sunday, Nov 07, 2010

- Testing core-coap-03
  - focusing on newcomers
  - most physically present, some via Internet
- Basic interoperability done
  - message format, options encoding, transaction model
  - GET, PUT, POST, DELETE, link-format
- Continue testing on specific features
  - Block (nearly universal now)
  - Asynchronous transactions, observe (3 interoperable)
- Followup plugfest 1600–1800 Thursday
  - let's just hijack the terminal room

79th IETF: core WG Agenda

15:10	Introduction, Agenda, Status	Chairs (10)
15:20	1 – core CoAP	ZS (30)
15:50	1 – slicing/(-block)	CB (20)
16:10	retire to Wednesday, 09:00 Intro	Chairs (05)
09:05	1 – (-link-format)	ZS (15)
09:20	1 – (-observe)	ZS (35)
09:55	1 – Review of Requirements	ZS (10)
10:05	2 – Bootstrapping	BS (15)
10:35	1/2 – Group Communication	AR (15)
11:05	1/2 – CoAP Usage	PV (15)
11:20	Next Steps	Chairs (10)

79th IETF: core WG Agenda

15:10	Introduction, Agenda, Status	Chairs (10)
15:20	1 – core CoAP	ZS (30)
15:50	1 – slicing/(-block)	CB (20)
16:10	retire to Wednesday, 09:00 Intro	Chairs (05)
09:05	1 – (-link-format)	ZS (15)
09:20	1 – (-observe)	ZS (35)
09:55	1 – Review of Requirements	ZS (10)
10:05	2 – Bootstrapping	BS (15)
10:35	1/2 – Group Communication	AR (15)
11:05	1/2 – CoAP Usage	PV (15)
11:20	Next Steps	Chairs (10)

Constrained Application Protocol (CoAP)

draft-ietf-core-coap-03

Z. Shelby, B. Frank, D. Sturek

Didn’t read the draft?
See CoAP tutorial at the end

CoRE WG, IETF-79 Beijing

Progress since Maastricht

- coap-02 released
  - Link-format to *draft-ietf-core-link-format*
  - Use of Uri-Authority defined more completely
  - Uri-Scheme option removed
- coap-03 released
  - Token option added
  - CoAP specific error codes added
  - Uri-Query option added
  - Security section completed
- coap-03 plugfest event held yesterday

Editorial Tickets

- **#29** Section 2.1.2 error
  - Token error in the Section 2.1.2. example text
- **#31** Variable uint
  - Add section defining variable length uint (from coap-observe)
- **#51** Section 2 Organization
  - Separate by transaction model and req/res model sections
- **#56** Distinguishing CoAP and DTLS
  - Section 10.2 need a discussion on how to tell the difference between DTLS and CoAP messages (solved on mailing list)

Technical Tickets

- **#30** Max-age 0-4 bytes
  - Change Max-age length to 0-4 bytes from 1-4 bytes, allowing for 0s to be indicated by 0 length
- **#50** Human readable error payloads
  - Define error payloads to be human readable in Section 11.1.
- **#52** How strict to define POST
  - Was a comment that POST text may be too restrictive.
  - Proposal to include the language:

  “The actual function performed by the POST method is determined by the server and is usually dependent on the request URI”
- **#63** Verify all synchronous and asynchronous interactions

- And **#28, #48, #53, #62 ...**

#53 Token length

- A token was added to coap-03 to match requests with responses
- Very conservative length of 1-2 B was chosen
  - Reason: minimize overhead and server state
  - Problem: not sufficient for a client to store context in token (and protect the content)
- Proposal to define 1-8 B length
  - Sufficient for some token context and protection
  - Reasonable amount of state for a server

#28 Clarification on retransmission

- Should retransmits of responses transmit the current state of resource, rather than a snapshot of the state at the time of the first attempt?
  - Assumption in coap-03: **the snapshot**
  - Assumption in observe-00: **current recommended**
- Conclusion of mailing list discussion:
  - In some cases more memory efficient to send current state (rather than saving snapshot)
  - Proposal: Change Section 4.3 with “MAY include the current snapshot” and an explanation

#45 Block needs redirect

- (Ticket listed under ietf-block-00)
- Block transfer can support large representation in a POST request (or response) but not in both at the same time.
- This ticket proposes adding redirect support to coap-04 for redirecting a client to use a GET for retrieving the response
- Restricted to same-host only redirection for security reasons

#62 Uri-Scheme Option

- In draft-ietf-core-coap-01 and earlier we had a Uri-Scheme option
- Recent discussion has indicated that some people would find Uri-Scheme useful for a client to indicate the protocol to proxy to when using a multi-protocol proxy
- Discussion is needed to determine if we want to add Uri-Scheme back to the protocol

Security Tickets

- **#58** Define trust model
  - What's the general trust model in terms of the relationship between the servers and clients?
- **#59** Assumed device capabilities
  - What are the assumed capabilities of the devices in question?
- **#61** Cross-protocol attacks
  - Add some discussion of cross-protocol attacks, which seem likely with the NoSec mode.

- And **#54, #55, #60** ...

#55 AES-CCM ciphers

- Section 10.2 defines using CoAP with DTLS
- coap-03 currently defines SHOULD support for AES-CBC ciphers with DTLS
  - Problem: AES-CBC not possible on all constrained hardware
  - But: RFC4347 is based on TLS 1.1 and does not support AES-CCM
- Solution:
  - Wait for RFC4347bis, just passed WGLC
    - Supports AEAD, but only GCM defined
  - Define a separate CCM cipher suite, or use draft-mcgrew-tls-aes-ccm-00.txt? (hash?)

#60 Access control

- Eric (and Adam) brought up an issue:
  - “How is access control expected to behave with respect to proxy caches? (The HTTP story is clear but you've stripped out the HTTP access control mechanisms). I don't see how a server even verifies a client who goes through a cache.”

#54 IPsec and multicast

- CoAP supports multicast requests
- How to secure them?

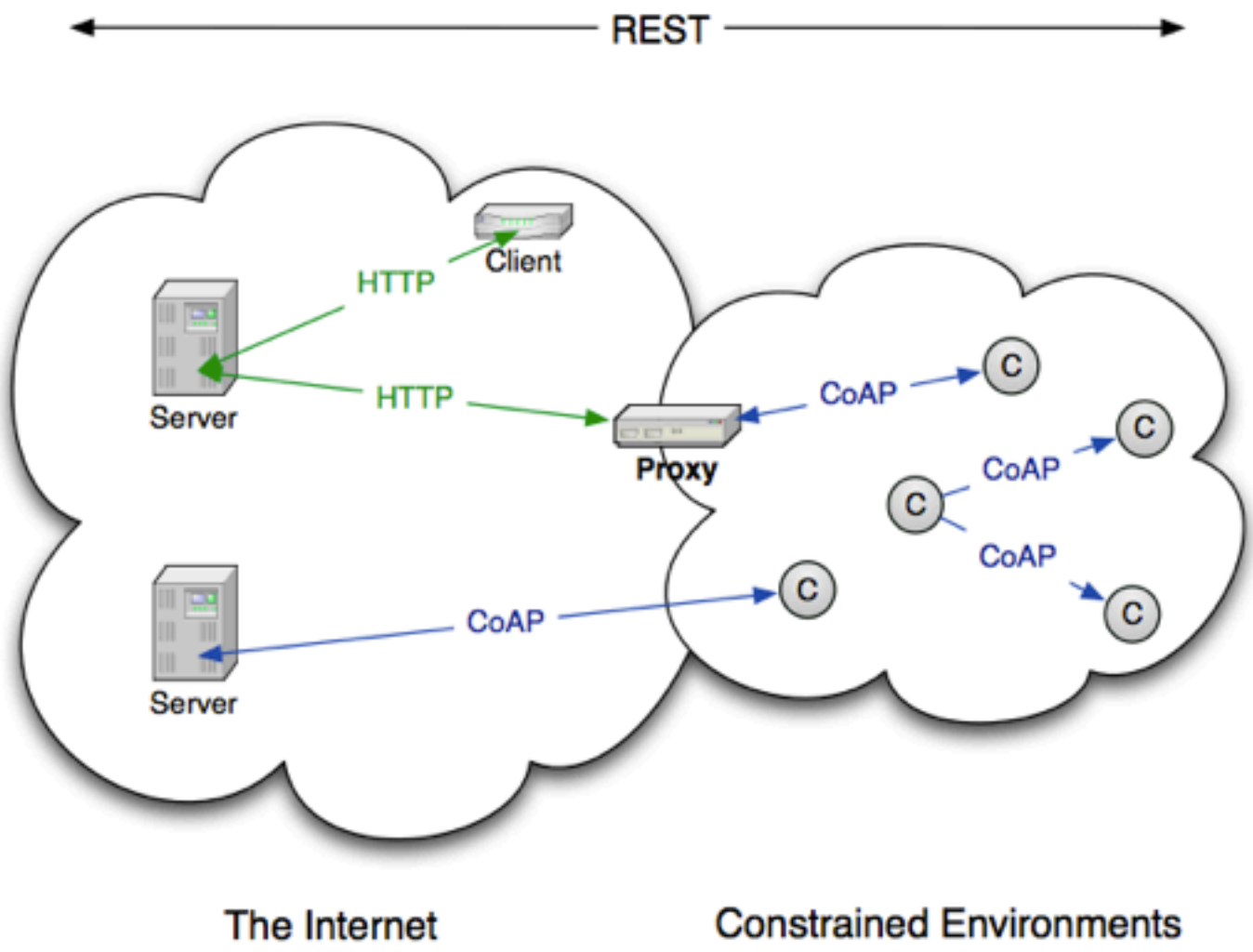
- Section 10.1 needs to be extended with a discussion on the use of IPsec with multicast

Next Steps

- Repeat the plugfest event this Thursday
- Close these tickets
  - Main focus on security
- Submit coap-04 within 3 weeks
- Go to last call
- WG goal to submit in December

CoAP Tutorial

The CoRE Architecture



CoAP Features

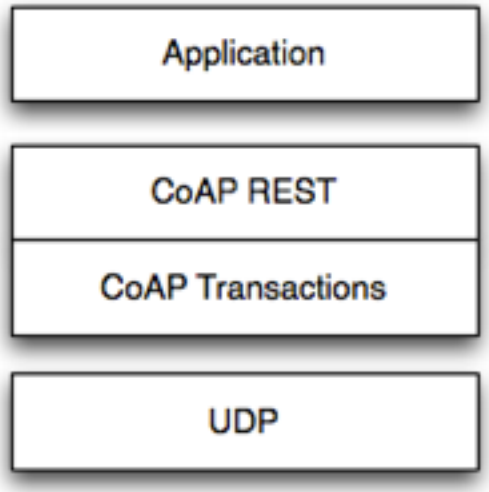
- Asynchronous transaction model
- UDP binding with reliability and multicast support
- GET, POST, PUT, DELETE
- Small, simple header < 10 bytes
  - 4 byte base header
  - TLV options, typically 2-4 bytes per option
- URI support
- Subset of IANA Internet media types
- Subset of HTTP-compatible response codes
- coap:// scheme
- Optional observation and discovery

What CoAP is (and is not)

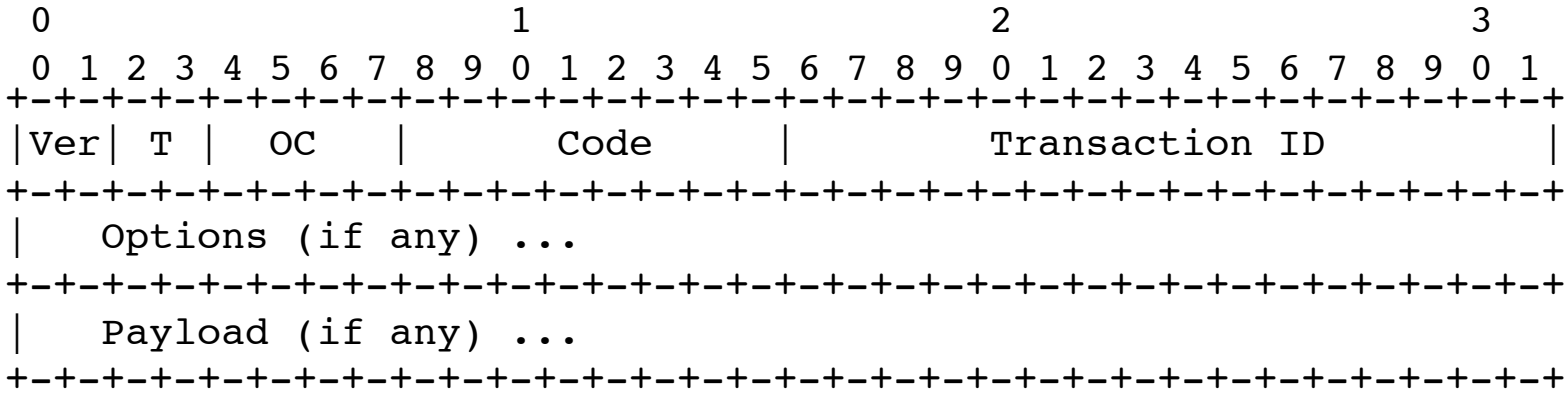
- CoAP is
  - A RESTful protocol
  - Both synchronous and asynchronous
  - For constrained devices and networks
  - Specialized for M2M applications
  - Easy to proxy to/from HTTP
- CoAP is not
  - A replacement for HTTP
  - General HTTP compression
  - Separate from the web

The Transaction Model

- Transport
  - CoAP is defined for UDP
- Transaction
  - Single message exchange between end-points
  - CON, NON, ACK, RST
- REST
  - Piggybacked on transaction messages
  - Method, Response Code and Options (URI, content-type etc.)



Message Header



Ver - Version (1)

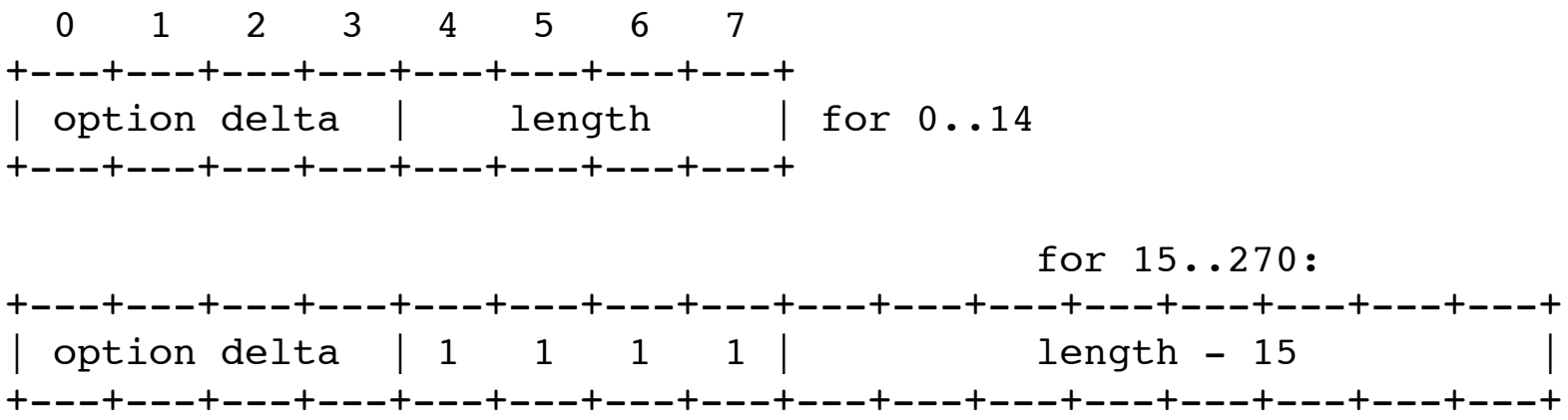
T - Transaction Type (Confirmable, Non-Confirmable, Acknowledgement, Reset)

OC - Option Count, number of options after this header

Code - Request Method (1-10) or Response Code (40-255)

Transaction ID - Identifier for matching responses

Option Header



Option Delta - Difference between this option type and the previous

Length - Length of the option value (0-270)

Value - The value of Length bytes immediately follows Length

Options

Type	C/E	Name	Data type	Length	Default
0	-	Reserved	-	-	-
1	C	Content-type	8-bit unsigned integer	1 B	0 (text/plain)
2	E	Max-age	Variable length unsigned integer	1-4 B	60 seconds
4	E	Etag	Sequence of bytes	1-4 B	-
5	C	Uri-Authority	String	1-270 B	""
6	E	Location	String	1-270 B	-
7	-	Reserved	-	-	-
9	C	Uri-Path	String	1-270 B	""
11	C	Token	Sequence of bytes		-
15	C	Uri-Query	String	1-270	-

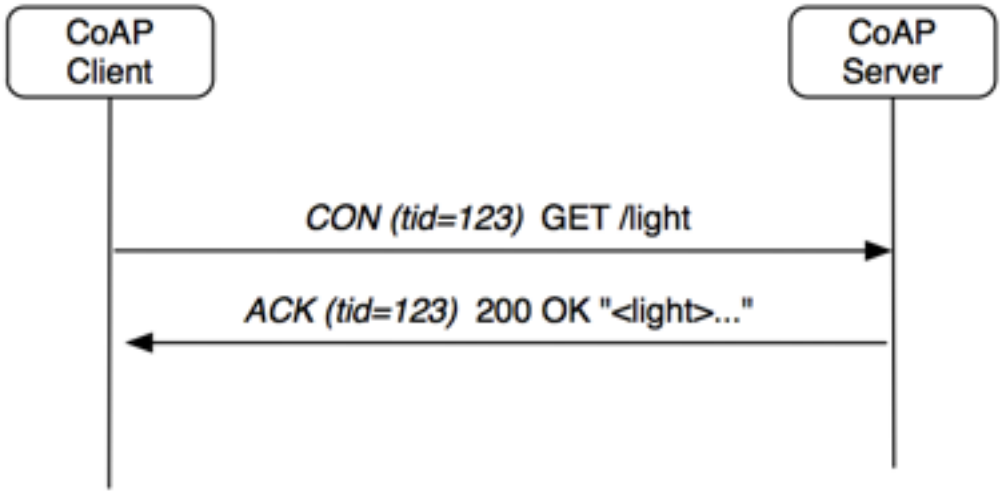
Response Codes

Code	HTTP Name
40	100 Continue
80	200 OK
81	201 Created
124	304 Not Modified
160	400 Bad Request
164	404 Not Found
165	405 Method Not Allowed
175	415 Unsupported Media Type
200	500 Internal Server Error
202	502 Bad Gateway
203	503 Service Unavailable
204	504 Gateway Timeout
240	Token Option required by server
241	Uri-Authority Option required by server
242	Critical Option not supported

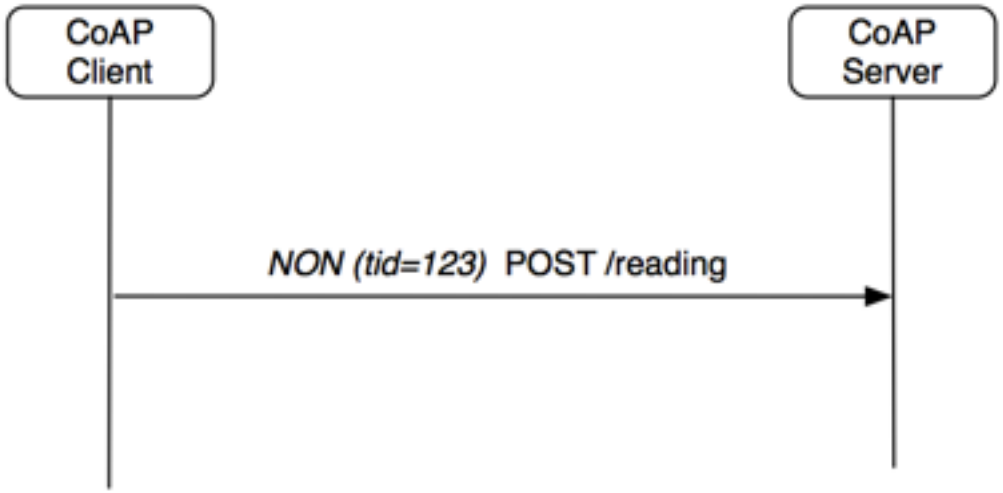
Internet Media Types

Internet media type	Identifier
text/plain (UTF-8)	0
text/xml (UTF-8)	1
text/csv (UTF-8)	2
text/html (UTF-8)	3
image/gif	21
image/jpeg	22
image/png	23
image/tiff	24
audio/raw	25
video/raw	26
application/link-format [IANA_TBD_LINK]	40
application/xml	41
application/octet-stream	42
application/rdf+xml	43
application/soap+xml	44
application/atom+xml	45
application/xmpp+xml	46
application/exi	47
application/x-bxml	48
application/fastinfoset	49

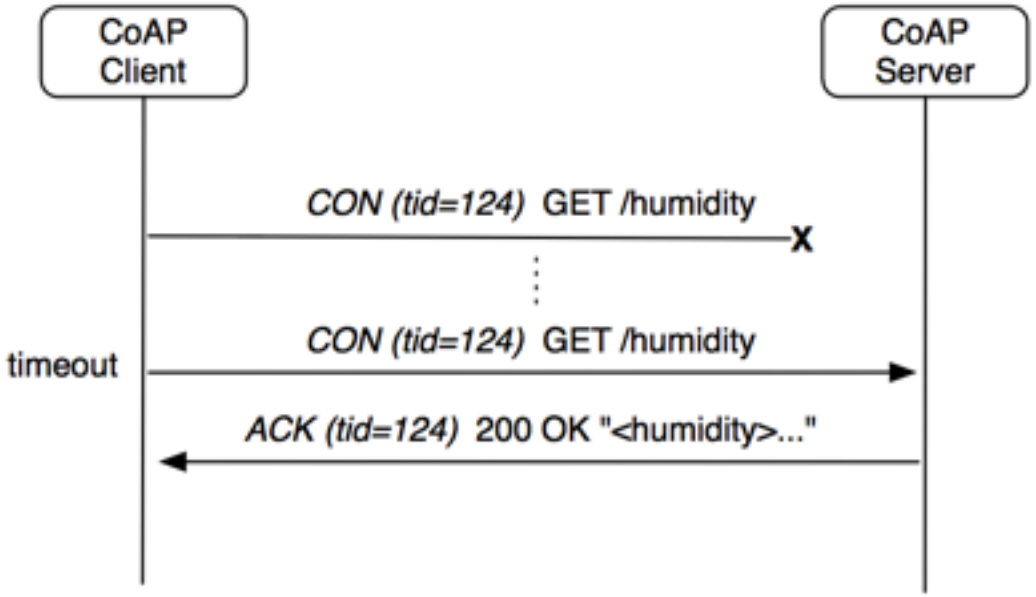
Confirmable Request



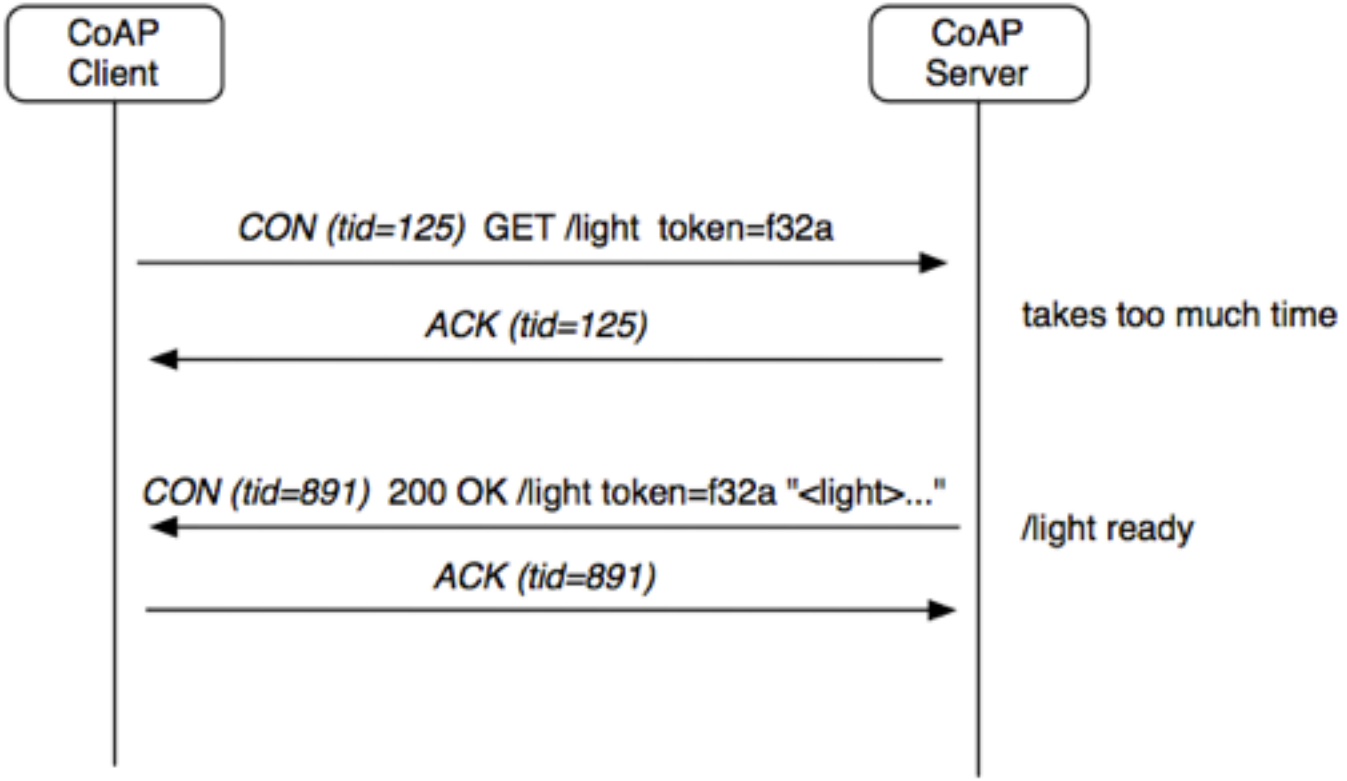
Non-Confirmable Request



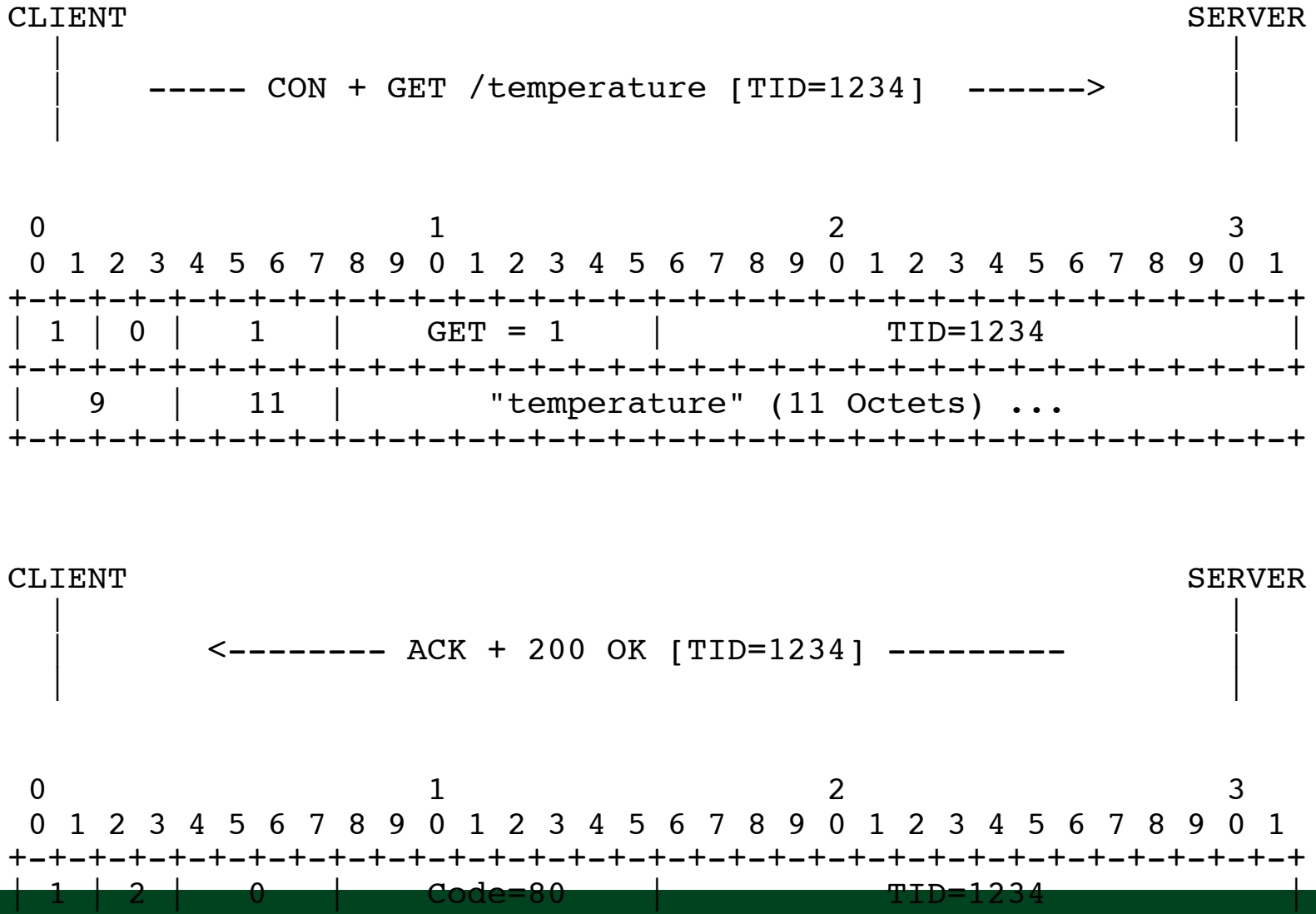
Dealing with Packet Loss



Asynchronous Response



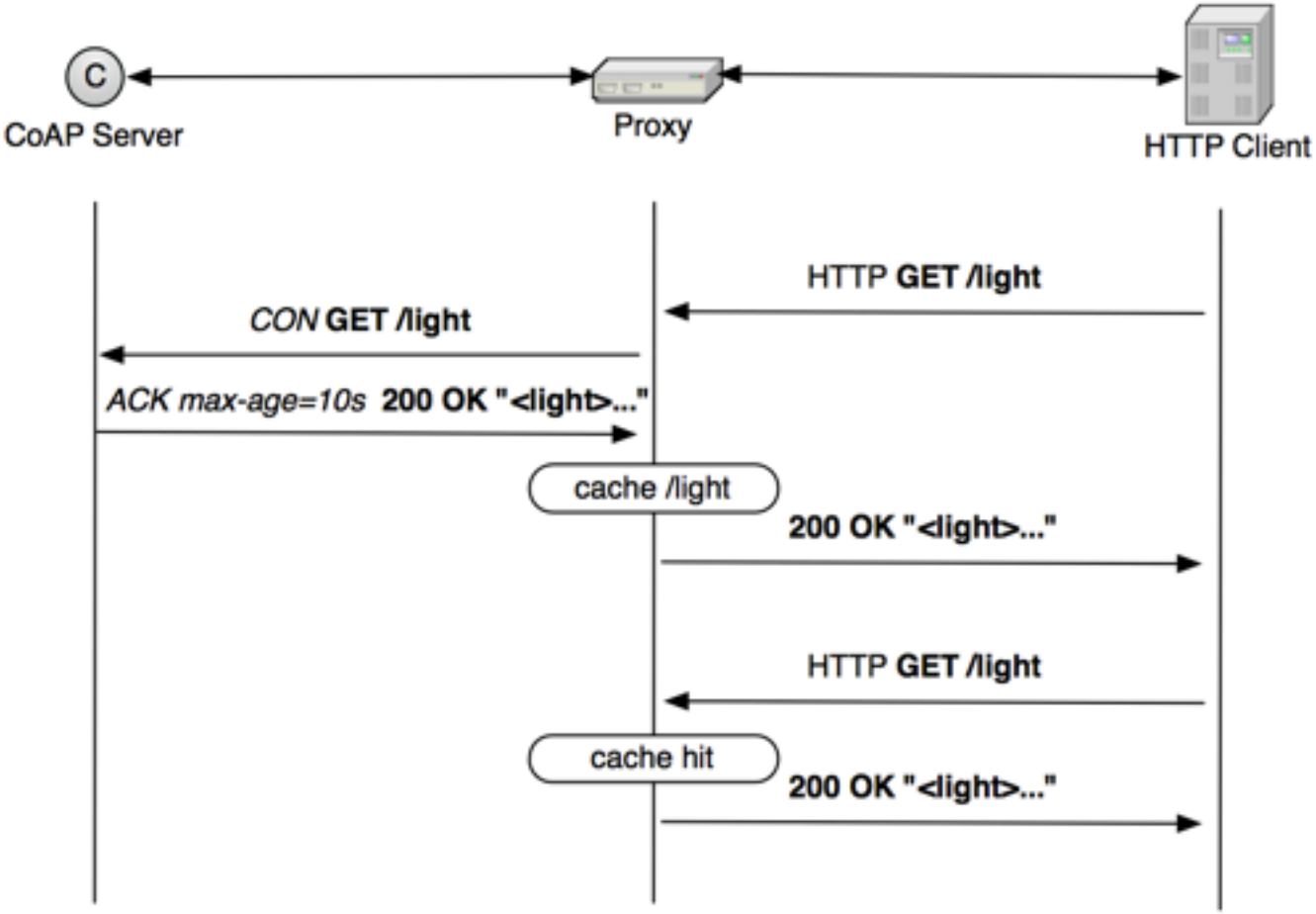
Bits and bytes...



Caching

- CoAP includes a simple caching model
  - Current only for the GET method
- Cache life
  - Controlled by the Max-Age Option
- Cache refresh and versioning
  - Using the Etag Option
- A proxy may participate in caching
  - Usually on behalf of a sleeping node

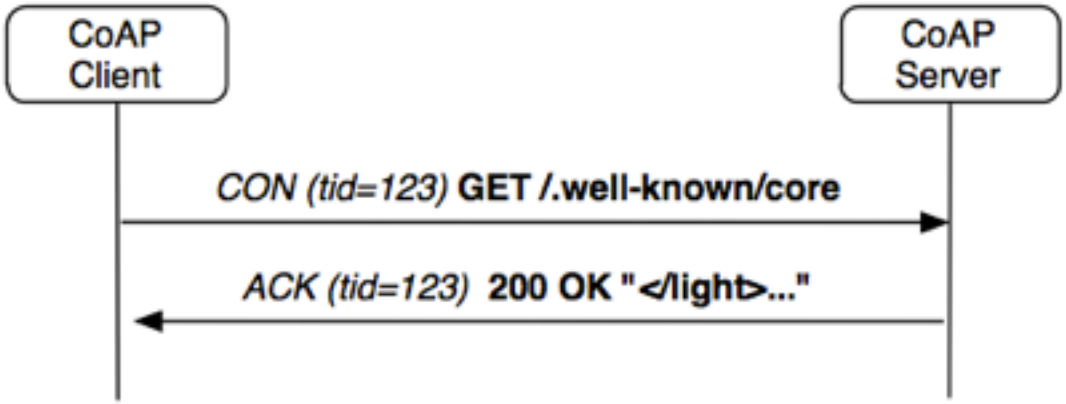
Proxying and caching



Resource Discovery

- Service Discovery
  - Leave this to e.g. DNS-SD
- Resource Discovery
  - Retrieving the links offered by CoAP servers
  - GET /.well-known/core
  - Returns a link-header style format
    - URL, name, description, content-type, short-url, id
- See *draft-ietf-core-link-format-01*

Resource Discovery



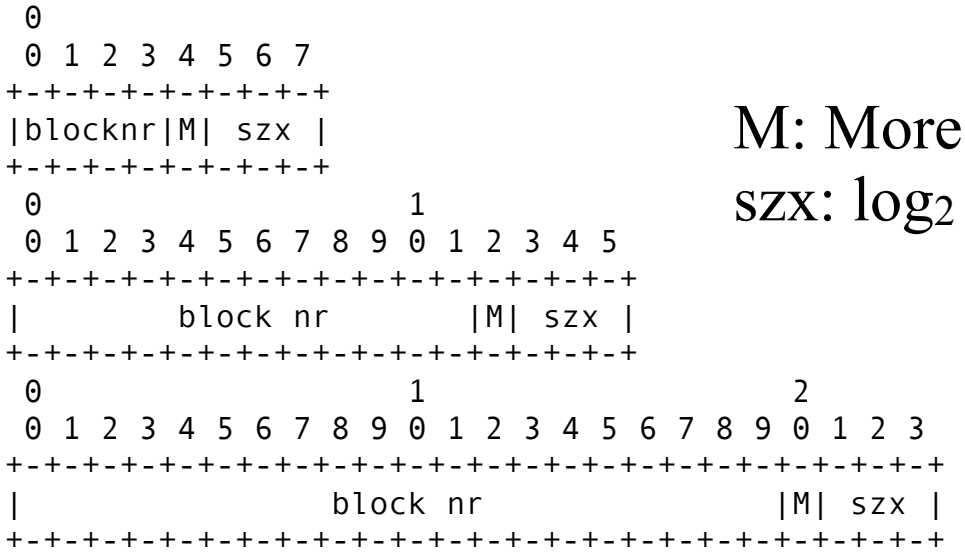

```
</light>;n="Illuminance";ct=0;sh=/i,
</s/maastr.xml>;n="Maastricht weather";ct=1,
</s/maastr/temp>;n="Temperature in Maastrich";ct=1;sh=/m,
</s/oulu.xml>;n="Oulu weather";ct=1,
</s/oulu/temp>;n="Temperature in Oulu";ct=1;sh=/o,
</s/temp>;n="Temperature";ct=0;sh=/t,
</test>;n="test";ct=0
```

79th IETF: core WG Agenda

15:10	Introduction, Agenda, Status	Chairs (10)
15:20	1 – core CoAP	ZS (30)
15:50	1 – slicing/(-block)	CB (20)
16:10 retire to Wednesday, 09:00	Intro	Chairs (05)
09:05	1 – (-link-format)	ZS (15)
09:20	1 – (-observe)	ZS (35)
09:55	1 – Review of Requirements	ZS (10)
10:05	2 – Bootstrapping	BS (15)
10:35	1/2 – Group Communication	AR (15)
11:05	1/2 – CoAP Usage	PV (15)
11:20	Next Steps	Chairs (10)

The block option

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



M: More Blocks
szx: $\log_2$ Blocksize – 4

Decisions:

- Block size is power of 2
- $16 \leq$ Block size ≤ 2048

The block option vs. methods

- **GET: trivial**
  - **Receiver: watch Etag to obtain parts of same resource repr.**
  - **Also works for asynchronous responses (subscriptions)**
    - initiative is with responder, then!
- **PUT, POST: trigger actual update on M=0**
  - **manage parallel operations based on token option**
- **Block is CRITICAL**

draft-bormann-core-coap-block-01.txt

- **Thought experiment**
  - **develops Size-Estimate option (see below)**
  - **develops "semantic segmentation" ("Slicing")**
- **Instead of using numeric block numbers, use semantic continuation tokens**
  - **continuation-response option: this is not all, more can be had by handing back the token given**
  - **continuation-request option: hand back the token**
  - **continuation-required: ask for a token (POST/PUT)**
  - **mesage-size: aid in agreement on a good slice size**
- **Is this better than Block?**

Semantic Slicing

- **Advantages:**
  - **Enables certain stateless proxies (for device enumeration)**
    - those could be done using REST means
    - requires putting continuation token in response body
  - **Enables application-oriented slice boundaries**
  - **Handles large POST/PUT responses**
  - **More flexibility**
- **Disadvantages:**
  - **More flexibility (behavior harder to predict)**
  - **More complexity**
  - **Harder to debug (less self-describing)**
  - **No random-access semantics (+/–?)**

Block-00 Tickets: Editorial work (no tech change)

- #47 Move discussion of benefits to introduction
- #48 Add example interactions
- #49 Expand security considerations

#44 estimate the size

- One solution in section 3 of bormann-block-01:
  - new option Size-Estimate
  - “should” be sent with first slice
- Alternative/additional solution:
  - Add size relationship attribute to link-format
- Exact size or estimate?
  - use cases not quite clear

#45 large responses to POST/PUT

- **Block can be used either on request body (POST/PUT) or on response body (GET), not both**
  - **do we need large POST/PUT responses? If yes:**
  - **add a second option?**
  - **use redirects to GET to retrieve large POST/PUT responses?**
- **Redirects currently not available in base CoAP**

#46 Error Codes for Block

- **Reaction to PUT or POST where previous segments aren't available**
- **Also possibly errors:**
  - **GET to block number that is beyond end**
    - could return empty payload instead
  - **GET with block number ≠ 0 and unsupported block size**
    - could reduce block size and shift block number instead

79th IETF: core WG Agenda

15:10	Introduction, Agenda, Status	Chairs (10)
15:20	1 – core CoAP	ZS (30)
15:50	1 – slicing/(-block)	CB (20)
16:10	retire to Wednesday, 09:00 Intro	Chairs (05)
09:05	1 – (-link-format)	ZS (15)
09:20	1 – (-observe)	ZS (35)
09:55	1 – Review of Requirements	ZS (10)
10:05	2 – Bootstrapping	BS (15)
10:35	1/2 – Group Communication	AR (15)
11:05	1/2 – CoAP Usage	PV (15)
11:20	Next Steps	Chairs (10)

CoRE Link Format

draft-ietf-core-link-format-01

Z. Shelby, with a lot of help from Peter Bigot

What is the CoRE Link Format

```
GET 184.106.150.250:61618/.well-known/core

</.well-known/core>;n="Resource discovery";ct=40,
</draft>,
</s/oulu.xml>;n="Oulu weather";ct=1,
</s/oulu/temp>;sh="o";n="Temperature in Oulu";ct=1,
</s/rand>;sh="r";n="Random number";ct=0,
</test>;n="test";ct=0,
</time/china>;n="Current time in China",
</time/euro>;n="Current time Central Europe",
</time/finland>;n="Current time in Finland”
```

You can try yourself:

http://184.106.150.250/coap/%5B0:0:0:0:0:0:0:1%5D:61618/.well-known/core

http://184.106.150.250

Progress since Maastricht

- link-format-00 split off from coap draft
  - Fixed the ABNF link-extension format
  - Clarified how filtering is optional
  - Required support of wildcard * processing when filtering is supported
- link-format-01 released
  - Formal definition for filter query string
  - Removed URI-reference from "n" and "id"
  - Added security text about multicast requests

Current Status

- RFC5988 “Web Linking” published recently
  - The CoRE link format is derived from this
- Tested in two plugfest events
  - Only trivial issues found in Beijing
  - Has been universally implemented
- 5 tickets currently identified
- Interest from memento.org to reference the CoRE Link Format

Known Issues

- (**#41** Update link-header ref to RFC5988)
- **#42** Finalize the link-extensions to define
  - Separate slide
- **#43** More examples needed
- **#57** Cyclical links
  - Clients parsing the link-format should be aware that /.well-known/core could include a link to itself or other cycles
- **#70** Query string filter definition

Finalizing the link-extensions

Extension	Key	Type	
Description	d	URI-reference	
Short URL	sh	URI-reference	**remove?**
Name	n	quoted-string	
Content-type	ct	integer	
Identifier	id	quoted-string	
Observable	obs	-	**proposed**
Size maximum	sz	integer	**proposed**

Next Steps

- Close these tickets
- Submit link-format-02 within 2 weeks
- Go to last call
- WG goal to submit in December

79th IETF: core WG Agenda

15:10	Introduction, Agenda, Status	Chairs (10)
15:20	1 – core CoAP	ZS (30)
15:50	1 – slicing/(-block)	CB (20)
16:10	retire to Wednesday, 09:00 Intro	Chairs (05)
09:05	1 – (-link-format)	ZS (15)
09:20	1 – (-observe)	ZS (35)
09:55	1 – Review of Requirements	ZS (10)
10:05	2 – Bootstrapping	BS (15)
10:35	1/2 – Group Communication	AR (15)
11:05	1/2 – CoAP Usage	PV (15)
11:20	Next Steps	Chairs (10)

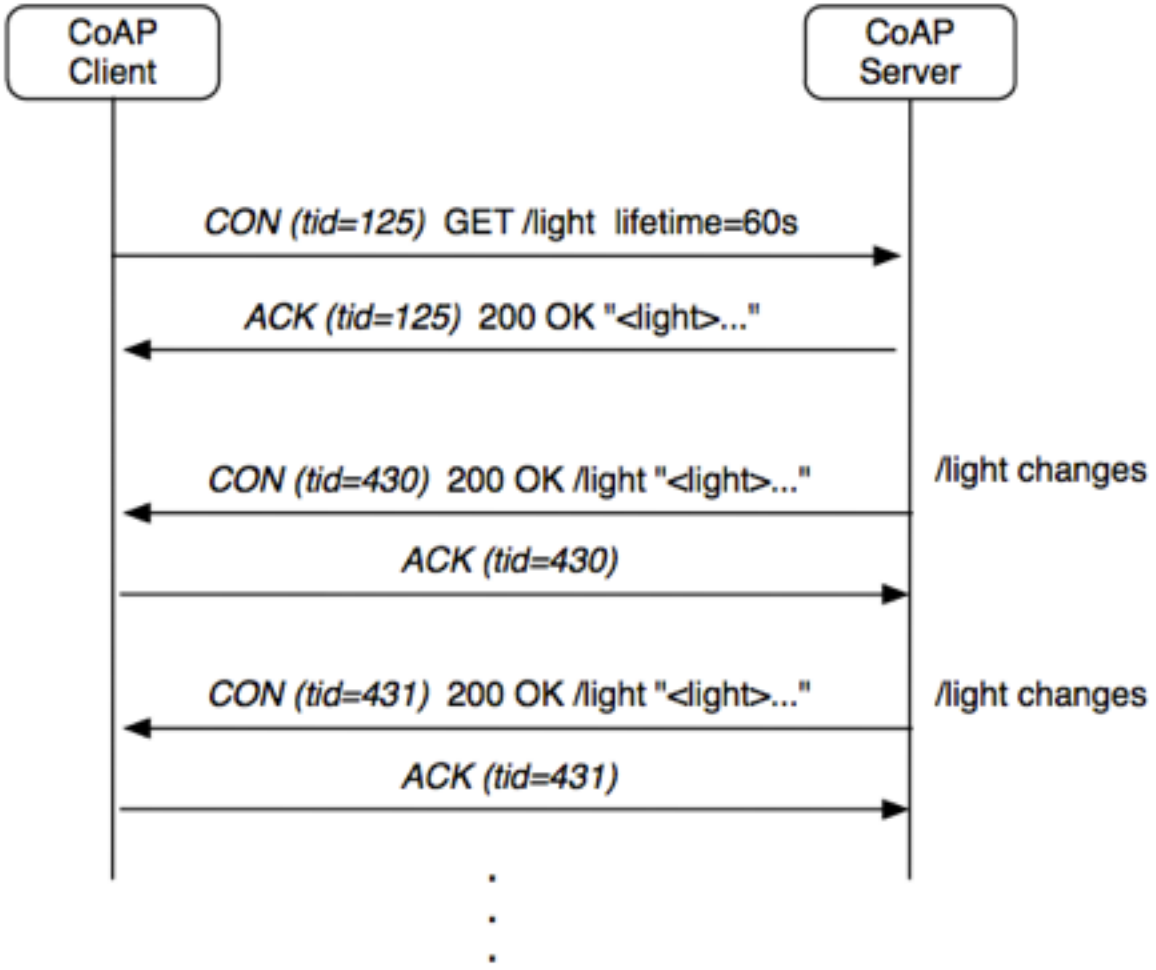
CoRE Observation

draft-ietf-core-observe-00

K. Hartke, Z. Shelby

CoRE WG, IETF-79 Beijing

What is CoRE Observe?



Progress since Maastricht

- observe-00 (submitted as WG document)
  - Removed the explanatory appendix
  - Removed the HTTP mapping
  - Removed the caching explanation
  - Omit URI from notifications if Token is present
  - Subscription option as variable length uint

Current Status

- First working group version
  - Needs plenty of editing
  - Needs considerations for coap-04 and block
- Tested in Maastricht and Beijing plugfests
  - 4+ implementations
- 15 tickets currently identified
  - Many are placeholders

Main Technical Tickets

- **#34** Canceling a subscription
- **#36** Add consideration of core-block
- **#38** Example on proxy interaction
- **#39** Caching (validation model)
- **#40** Security section needed
- **#65** Normal requests should not affect any ongoing observation
- **#66** Identifying observations
- **#67** Clarify rules for notifications
- **#69** Notifying temporarily unresponsive clients

Next Steps

- Close these tickets
- Submit observe-01 soon after Beijing
- More implementation testing and feedback

79th IETF: core WG Agenda

15:10	Introduction, Agenda, Status	Chairs (10)
15:20	1 – core CoAP	ZS (30)
15:50	1 – slicing/(-block)	CB (20)
16:10	retire to Wednesday, 09:00 Intro	Chairs (05)
09:05	1 – (-link-format)	ZS (15)
09:20	1 – (-observe)	ZS (35)
09:55	1 – Review of Requirements	ZS (10)
10:05	2 – Bootstrapping	BS (15)
10:35	1/2 – Group Communication	AR (15)
11:05	1/2 – CoAP Usage	PV (15)
11:20	Next Steps	Chairs (10)

79th IETF: core WG Agenda

15:10	Introduction, Agenda, Status	Chairs (10)
15:20	1 – core CoAP	ZS (30)
15:50	1 – slicing/(-block)	CB (20)
16:10	retire to Wednesday, 09:00 Intro	Chairs (05)
09:05	1 – (-link-format)	ZS (15)
09:20	1 – (-observe)	ZS (35)
09:55	1 – Review of Requirements	ZS (10)
10:05	2 – Bootstrapping	BS (15)
10:35	1/2 – Group Communication	AR (15)
11:05	1/2 – CoAP Usage	PV (15)
11:20	Next Steps	Chairs (10)

Security Bootstrapping

Colin O'flynn
Behcet Sarikaya (presenter)
Yoshi Ohba
Zhen Cao
Robert Cragie

draft-oflynn-core-bootstrapping-03 at IETF 79

Architecture

- 6lowPAN ND or Zigbee SE 2.0 architecture/ topology adopted
- Root node is coordinator/6LBR
- Interior routers/6BR
- Leaf nodes
- Bootstrapping keys based on layers
- Lower layer protocols: **802.15.4 MAC & LowPAN adaptation Layer**
- Higher layer protocols: **IP and above**

Protocols

- **Security Objectives**
- **EAP**: EAP authentication framework based on RFC 5247
- **Available Methods**:
  - PANA
  - HIP-DEX
  - 802.1X
- **Emphasis in the draft is** on the requirements on each of the available methods and meeting the objectives

Next Steps

- The draft has gone through major revision on -02
- Presented -02 in the last Interim
- Comments since then incorporated into -03
- Ready to become WG draft
- We ask for WG draft adoption

79th IETF: core WG Agenda

15:10	Introduction, Agenda, Status	Chairs (10)
15:20	1 – core CoAP	ZS (30)
15:50	1 – slicing/(-block)	CB (20)
16:10	retire to Wednesday, 09:00 Intro	Chairs (05)
09:05	1 – (-link-format)	ZS (15)
09:20	1 – (-observe)	ZS (35)
09:55	1 – Review of Requirements	ZS (10)
10:05	2 – Bootstrapping	BS (15)
10:35	1/2 – Group Communication	AR (15)
11:05	1/2 – CoAP Usage	PV (15)
11:20	Next Steps	Chairs (10)

Group Communication for CoAP

Akbar Rahman (Editor)
(with much input from Kerry Lynn, Peter Bigot,
Peter van der Stok, and others)

IETF 79, November 2010

http://tools.ietf.org/html/draft-rahman-core-groupcomm-01

Background

- This draft is a follow up to our previous draft on "Sleeping and Multicast Considerations for CoAP" which was in a problem statement format:
  - http://tools.ietf.org/html/draft-rahman-core-sleeping-00

- During the previous CORE Webex calls, we were asked to produce satellite drafts to more precisely identify the problems and provide some initial solution proposals for:
  - Group Communications (as the more general problem of multicast) – This draft
  - Sleeping Nodes – TBD draft (but in progress)

Potential Approaches for Group Communication

- There are three alternative approaches for CoAP group communications each with associated pros/cons:
  - IP Multicast
  - Overlay (Proxy based) Multicast
  - CoAP Application level Group Management

IP Multicast

- Concept:
  - CoAP sub-networks to be connected directly to IP multicast enabled routers (e.g. running PIM-SM [RFC4601]).
  - Sending CoAP node can directly transmit group messages by setting IP address to selected multicast IP group address
  - Receiver CoAP nodes use MLD [RFC3810] to subscribe (listen) to any messages sent to selected IP multicast group
- Pros
  - Most efficient solution since done at IP layer
  - ROLL [draft-ietf-roll-rpl-14] assumes IP multicast supported
  - CoAP-03 draft [section 4.1] assumes IP multicast supported
- Cons
  - IP multicast is not generally deployed outside of corporate LANs and a few ISPs. So we may specify IP multicast support but practically it may often not be deployed

Overlay (Proxy based) Multicast (1/2)

- Concept:
  - We define overlay multicast as one that utilizes an infrastructure based on proxies (rather than an IP router based multicast backbone) to deliver IP multicast packets to an end device
  - Since ROLL and CoAP drafts already support MLD (see pg. 4), we propose MLD Proxy [RFC3810] to be used as the overlay multicast approach
  - Specifically, the CoAP proxy node will also support Proxy MLD
  - Receiver CoAP nodes use MLD Proxy signaling to subscribe (listen) to any messages sent to selected IP multicast group
  - The CoAP (MLD) proxy node would be responsible for delivering any IP multicast message to the subscribed CoAP devices
  - Note that the CoAP (MLD) proxy need not necessarily be connected to an external multicast backbone

Overlay (Proxy based) Multicast (2/2)

- Pros
  - Ties well into existing CoAP proxy concept
- Cons
  - It is not obvious that existing MLD Proxy [RFC 3810] allows the specific scenario we are proposing. Further investigation required.

CoAP Application level Group Mgmt

- Concept:
  - Perform all group communications at the CoAP application level
  - Expand CoAP headers to allow simple group mgmt functions (Join, Leave, etc.)
  - The CoAP proxy node would be responsible for group mgmt
  - Any CoAP node that wanted to send a message to a CoAP group would first send the CoAP message to the proxy. The proxy would then explode it out to the group
- Pros
  - Functionality fully within the CoAP protocol (and CORE WG control)
  - Analogous approach as Email group management (and other Apps)
- Cons
  - Has high overhead compared to lower layer solutions

Group Resource Manipulation (1/3)

- Needed to replicate functionality of existing standards, e.g. BACnet's Alarm and Event Notification service

- Two forms of group resource manipulation should be supported:
  - Push (PUT or MPUT) as for example "turn off all lights simultaneously"
  - Pull (GET or MGET) as for example "return all the resources matching a well known URI"

- Conceptually, the result of a MGET or MPUT should be the same as if the client had unicast them serially

Group Resource Manipulation (2/3)

- Limit manipulation to idempotent methods (PUT/GET/DEL)
  - Repeat requests can then be used to increase reliability of receipt
- Requires a consistent naming and addressing scheme for groups
  - Multicast is the easy case; can use DNS to resolve FQDN in authority to multicast or unicast address
- Can a group be represented by a list of addresses as well?
  - If so, perhaps this argues for a group scheme, e.g. “coapm” to signal a proxy to do fan-out task

Group Resource Manipulation (3/3)

- Target resource must be located at same port and path for all group members
  - Suggests a need to advertise path, port or have a priori agreement

79th IETF: core WG Agenda

15:10	Introduction, Agenda, Status	Chairs (10)
15:20	1 – core CoAP	ZS (30)
15:50	1 – slicing/(-block)	CB (20)
16:10	retire to Wednesday, 09:00 Intro	Chairs (05)
09:05	1 – (-link-format)	ZS (15)
09:20	1 – (-observe)	ZS (35)
09:55	1 – Review of Requirements	ZS (10)
10:05	2 – Bootstrapping	BS (15)
10:35	1/2 – Group Communication	AR (15)
11:05	1/2 – CoAP Usage	PV (15)
11:20	Next Steps	Chairs (10)

CoAP Utilization for Building Control

draft-vanderstok-core-bc-02

Naming/Discovery/Legacy

Peter van der Stok
Kerry Lynn

November 10, 2010

Motivating Naming/Discovery

A typical BC installation may have 1000s of “points”

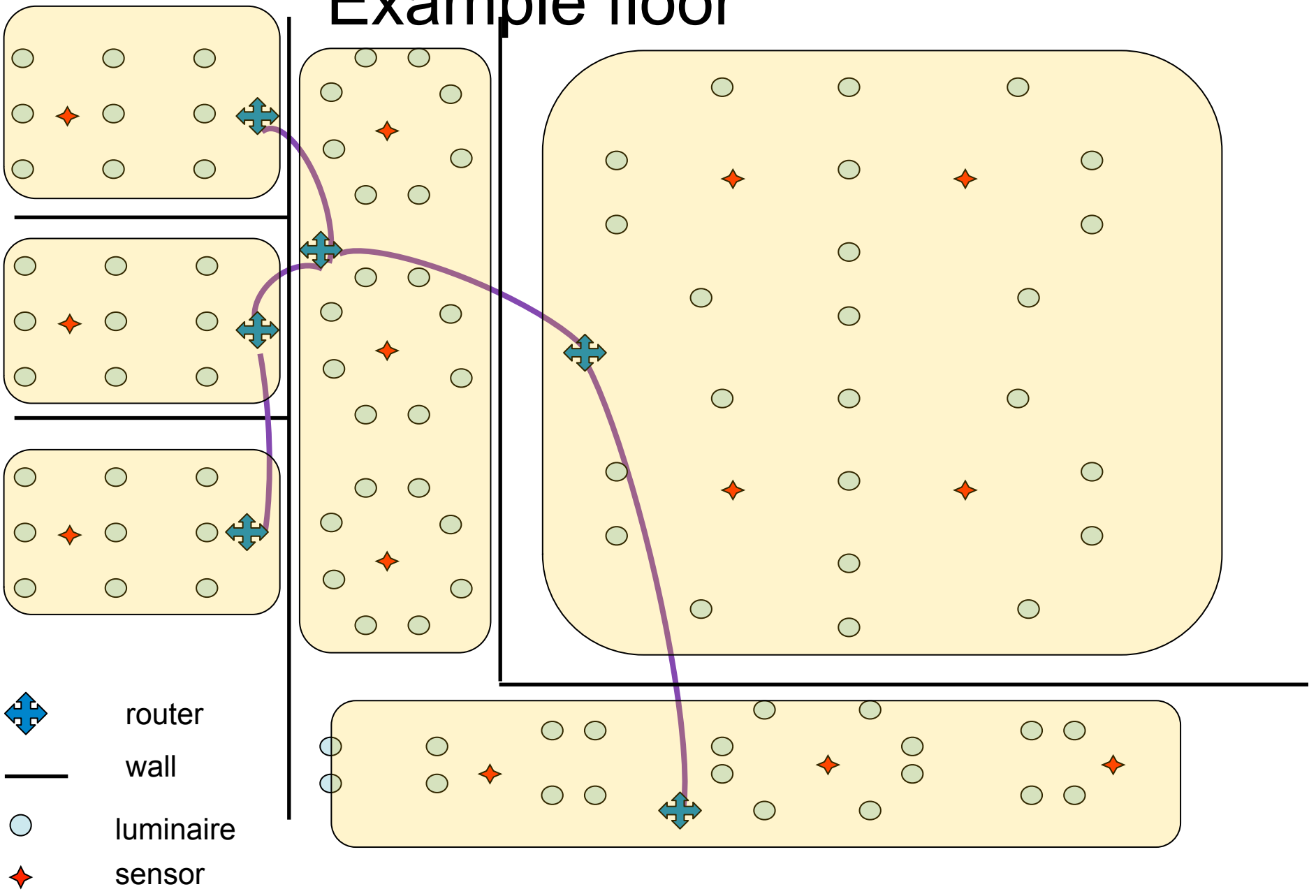
Authority:
- Node (host [:socket])
- Group (set of nodes)

Service:
- CoAP resource (URI)
- /.well-known/core/type/device/... (RFC 5785)
- legacy standards (e.g. dali, ZigBee, BACnet)

CoAP exposes: list of resources for a given node (functional entry points)

Additionally needed:
- Definition/discovery of groups
- Discovery of all nodes in a scope (authority)
- Discovery of resources with given characteristics (type, etc.)

Example floor



core-bc works out use of DNS

Central server solves:

- Large set > 100 nodes per domain
- Grouping (over subnets)

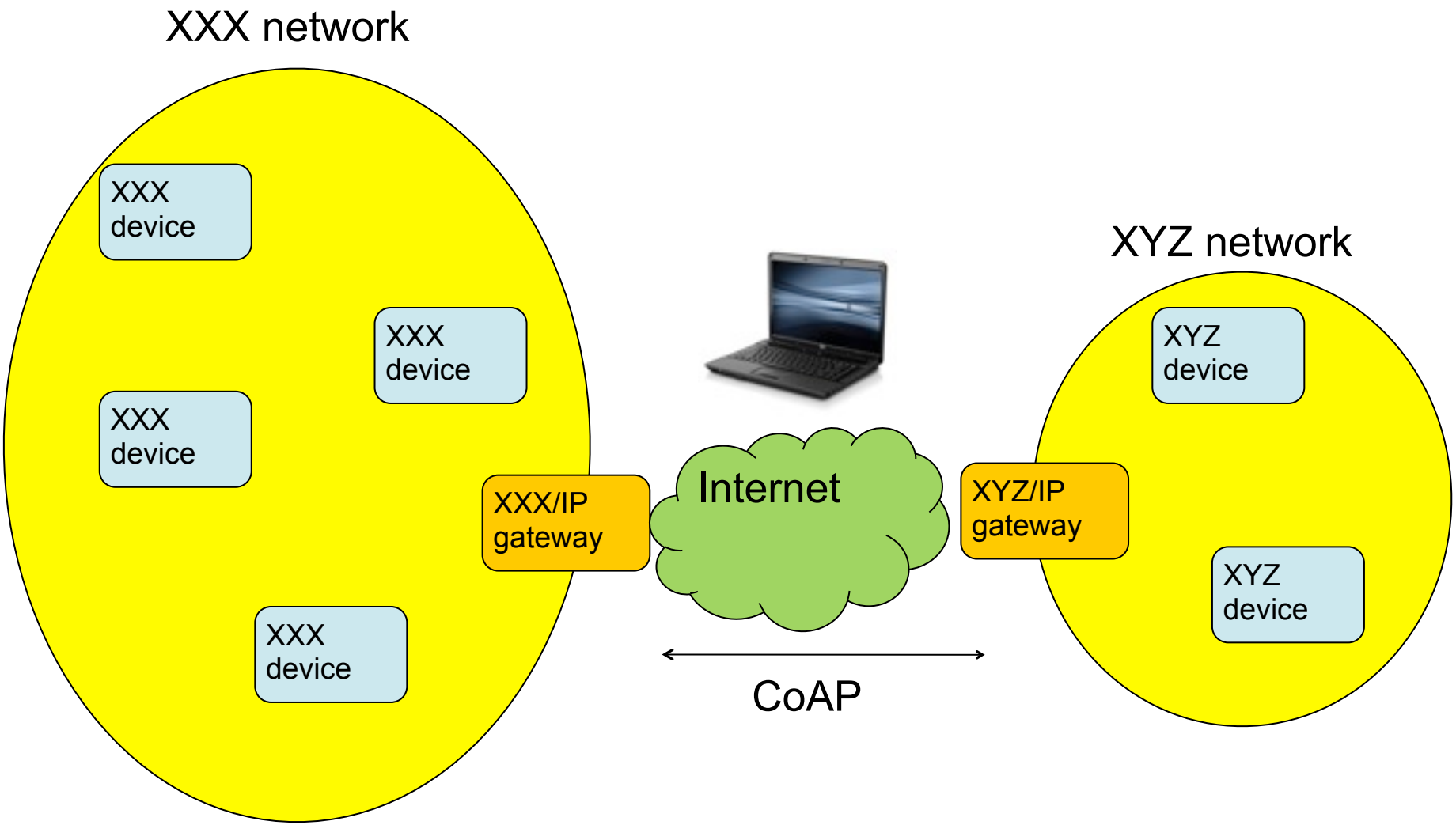
DNS-SD:

- Based on mature, well-known technology
- Hosts, dynamic ports via SRV records
- Path (functional entry points) via TXT records

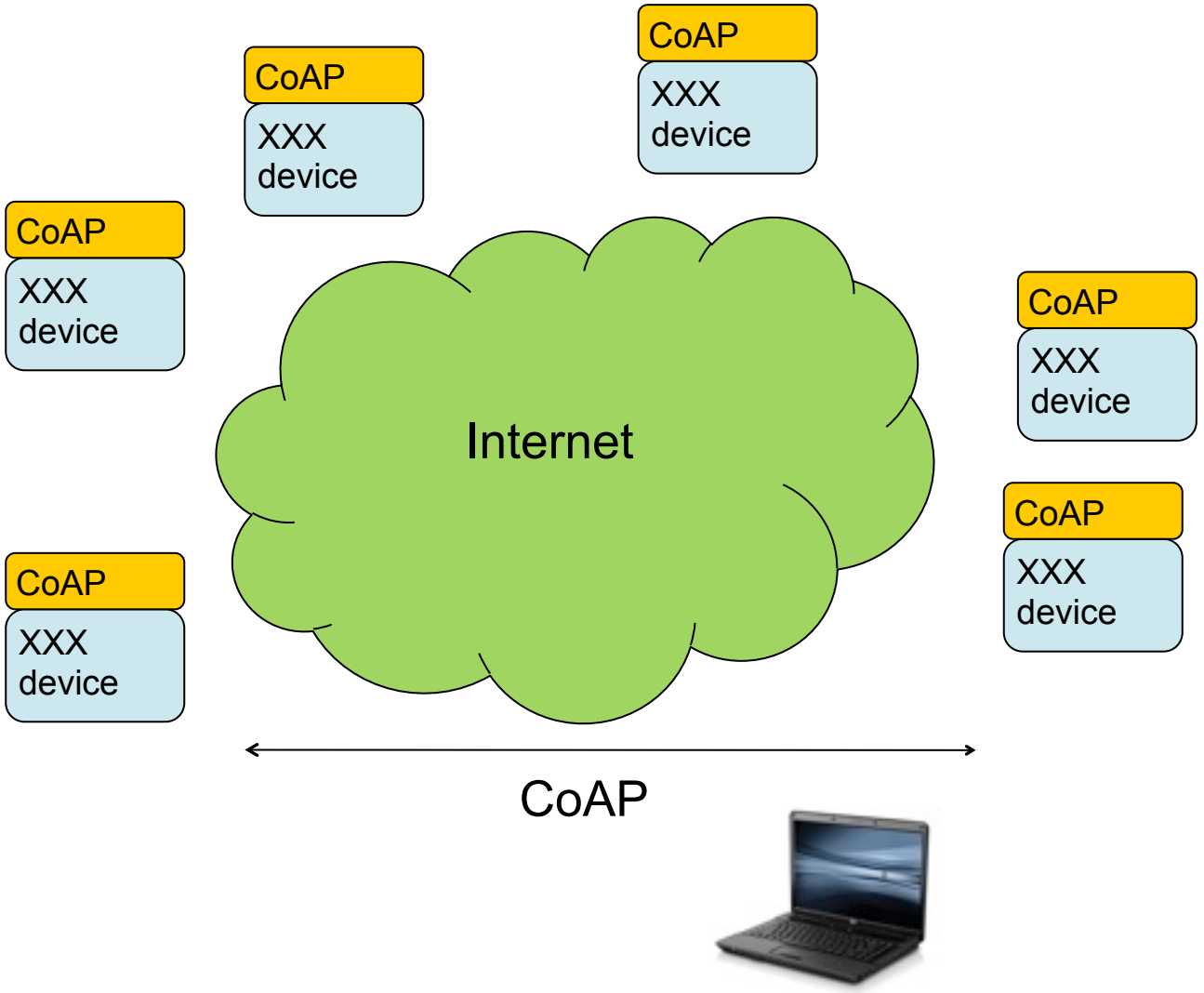
Future work:

Smooth transition during installation/commissioning
FROM local isolated networks without IT services (mDNS?)
TO DNS providing global name registry

XXX/XYZ legacy network and IP networks



XXX legacy naming with IP connectivity



Legacy and DNS-SD

Mapping of legacy standard “xxx” to DNS-SD naming conventions:
Assume that standardization body decides independent of CoAP

Proposed names inserted in DNS for legacy devices:
identifier._xxx._udp.domain
e.g. name._zigbee._udp.domain, where “name” is based on “n=”

Possibly also by subtype:
identifier._type._sub._xxx._coap._udp.domain
e.g. name._light._sub._dali._udp.domain

In TXT records, additional information like:
type=dali
dalitype=4
sh=/dl20

DRAFT **Example** of Installation

Assume an installation tool, DNS server on-line
DNS is initialized with domains

Devices connected to network and switched on
Tool communicates to device:
identifier: e.g: xyz0054ba
domain: bu036.floor1...
Tool reads from device: IP address, service, resources, short url

Tool updates DNS server (port number?)
Tool defines groups

Maps id._dalitype._sub._dali._udp. bu036.floor1.building.org
To coap://id.bu036.floor1.building.org/short_url

Proposals

- Use DNS-SD and mDNS for service/resource discovery
- Base coap authority on canonical host name (A or AAAA record name)
- Create a limited structured namespace for functional entry points at /.well-known/core/type/function
- Naming convention to discover services with legacy naming

- Continue with:
  - mDNS to DNS transition
  - CoAP gateway to legacy

79th IETF: core WG Agenda

15:10	Introduction, Agenda, Status	Chairs (10)
15:20	1 – core CoAP	ZS (30)
15:50	1 – slicing/(-block)	CB (20)
16:10	retire to Wednesday, 09:00 Intro	Chairs (05)
09:05	1 – (-link-format)	ZS (15)
09:20	1 – (-observe)	ZS (35)
09:55	1 – Review of Requirements	ZS (10)
10:05	2 – Bootstrapping	BS (15)
10:35	1/2 – Group Communication	AR (15)
11:05	1/2 – CoAP Usage	PV (15)
11:20	Next Steps	Chairs (10)