Constrained RESTful Environments WG (core)

Chairs:

Andrew McGregor <andrewmcgr@gmail.com>

Carsten Bormann <cabo@tzi.org>

Mailing List:

core@ietf.org

Jabber:

core@jabber.ietf.org

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

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

Note Well

This summary is only meant to point you in the right direction, and doesn't have all the nuances. The IETF's IPR Policy is set forth in BCP 79; please read it carefully.

The brief summary:

- **By participating with the IETF, you agree to follow IETF processes.**
- **If you are aware that a contribution of yours (something you write, say, or discuss in any IETF context) is covered by patents or patent applications, you need to disclose that fact.**
- **You understand that meetings might be recorded, broadcast, and publicly archived.**

For further information, talk to a chair, ask an Area Director, or review the following:

BCP 9 (on the Internet Standards Process)

BCP 25 (on the Working Group processes)

BCP 78 (on the IETF Trust)

BCP 79 (on Intellectual Property Rights in the IETF)

Milestones (from WG charter page)

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

Document submissions to IESG:

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

draft-ietf-core-coap-18

- Was approved 2013-07-11
- Still in RFC editor queue, waiting for two MISSREFs:
- draft-mcgrew-tls-aes-ccm-ecc
  - Defines the DTLS ciphersuites for RPK and Cert mode: TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
  - (alongside with TLS_PSK_WITH_AES_128_CCM_8 for PSK mode, RFC 6655)
  - IESG state “Approved-announcement to be sent::Point Raised – writeup needed”
- draft-ietf-tls-oob-pubkey
  - Defines RPK for DTLS
  - IESG state “Waiting for Writeup” (post IETF last call)

We are now in a phase change

- From
  - Oh I have this cool idea, how about that
- To
  - I have a deployment with a problem to solve
  - Here is how we solved it

WG documents

- draft-ietf-core-observe — after 2nd WGLC
- draft-ietf-core-block — before 2nd WGLC
- draft-ietf-core-groupcomm — before WGLC
- draft-ietf-core-http-mapping

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

Monday

All times are in time-warped PST

- **14:50–14:55 Intro**
- **14:55–15:25 -observe (KH)**
- **15:25–15:45 -block (CB)**
- **15:45–16:00 Groupcomm (AR)**
- **16:00–16:15 HTTP mapping (SL)**
- **16:15–16:30 Resource Directory (ZS)**

} WG docs

- ~~**16:30–17:00 Alternative Transports (KH)**~~ **Thu**
- **17:00–17:20 No-Response (AB)**

Between the slots

- **LWIG meeting Wed 13:00–15:30**
- **6Lo Tue 16:10–18:40 (preceded by 6TiSCH)**

- **CoRE AA proposed work going on**

Thursday

- 15:20–15:25 Intro
- 15:25–16:10 Access Control/Authorization in CoAP
  - Report from ad-hoc meetings
  - Way forward (discussion, chairs)
- 16:10–16:15 External updates (OMA, plugtest)
- 16:15–16:30 What do we need for sleepy nodes (AR)
- 16:30–16:40 CoAp Management Interfaces (PV)
- 16:40–17:20 Flextime
  - Spillover from Monday
  - Implementer info, e.g., draft-bormann-core-roadmap, LWIG
  - Open Mike

Group 1: WG docs

Observing Resources in CoAP

draft-ietf-core-observe

IETF 88

Klaus Hartke

Cancellation

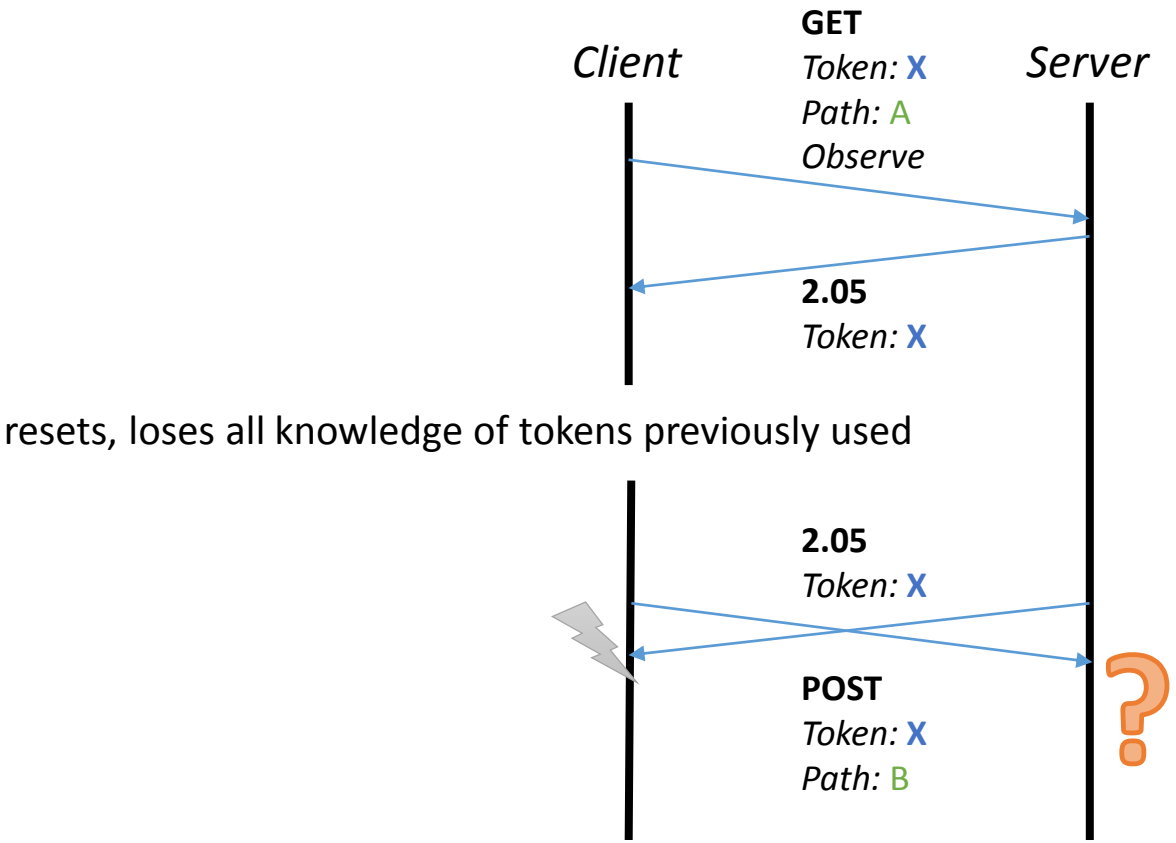
- Current: *Reactive Cancellation*
  An entry is removed from the list of observers when
  - the server sends a non-2.xx notification,
  - the server reboots and loses the state,
  - the server receives a Reset message in reaction to a notification, or
  - the last attempt to transmit a confirmable notification times out
- Double-check: Do we need *Proactive Cancellation*?
- We need an alternative cancellation mechanism for some transports anyway (such as *CoAP over WebSockets*)
- Note: The mechanism would be in addition to the current mechanism in *CoAP over UDP/DTLS*, not instead

Token Lifetime

- “The client SHOULD generate tokens in such a way that tokens currently in use for a given source/destination endpoint pair are unique”
- Current: A Token is in use when observing a resource
  - while the client expects further notifications
  - as long as the token is in the list of observers at the server

  whichever takes longer
- Problem: It may not entirely clear for a client when a token can be reused
- Simple solution: Don’t reuse tokens
  - Keep track of tokens used; randomize tokens if not possible
- Do we need something more elaborate than this?

Inadvertent Token Reuse



- Do we need to place any explicit requirements on the server here?

Intentional Token Reuse

- Current: A client may reuse a token exclusively to reinforce its interest in a resource
- If we place requirements on a server that experiences a client using an active token for a new exchange:
- Are these requirements powerful enough to make reuse of active tokens a generally acceptable technique?

Notification Supersedure (i)

- The purpose of the protocol is to let a client interested in a resource always have a representation of the current resource state
  - This is not perfectly possible because of network latency and message loss
  - The idea is to have client and server constantly work towards this goal – limited by congestion control
- If the state of an observed resource changes while the transmission of a previous notification has not completed yet, the server cannot send a new notification immediately
- Approaches:
  A. Wait until transmission of the old notification has completed
  B. Abort the transmission of the old notification (but not before the current transmission attempt completed) and start a new transmission for the new notification (but with the timer and counter of the aborted transmission)

Notification Supersedure (ii)

- Approaches:
  A. Wait until transmission of the old notification has completed
  B. Abort the transmission of the old notification
     (but not before the current transmission attempt completed)
     and start a new transmission for the new notification
     (but with the timer and counter of the aborted transmission)
- Current: B is REQUIRED
  - Pros
    - The server starts working on updating the client to the latest resource state sooner
    - The server needs to remember only latest representation, not the representation in transmission and the latest representation
  - Cons
    - Needs extension of the CoAP transmission algorithm
    - Current text in the draft has room for improvement
- Change
  - Allow choice between A and B
    – or –
  - Remove B and support only A

-block:
Blockwise Transfer

-block-14

- The good news: Block has been stable for a long time for the “GET large static resource” use case.
- The bad news: it seems a bit harder to get the other cases right.
- Resolving #211: Introduce 2.31 Continue if a server doesn’t know the final response. Now redundant with More bit (which also is set).
- Resolving #253: get rid of server initiative. It is now always the responsibility of the client to get more blocks.

Interaction with -observe

- For a large resource, server now only notifies the first block (Block2 option, NUM=0).
  - client can advise desired block size in GET(Observe).
  - client cannot GET(Observe) with Block2(NUM≠0).
- Client then makes its own GET requests for the blocks with NUM≠0. (Different token, no Observe option.)
- Correlate using ETag, which therefore SHOULD be sent with initial notification when a Block2 option is used.

Lifetime issues

- How long does the server keep the response body around?
  - Can discard EXCHANGE_LIFETIME after last block was retrieved.
  - Client should probably retrieve without undue delay...
  - Server might want to discard EXCHANGE_LIFETIME after any block was retrieved.
- How long does the server keep partial request bodies around?
  - Client should probably send without undue delay...
  - Server might want to discard after EXCHANGE_LIFETIME of inactivity (send 4.08 if further blocks NUM≠0 come in).
- Should there be a way to cancel a blockwise transfer before the server times it out?
  - No.

Ordering (#334)

- –14 clarifies that a server can send 4.08 for out-of-order request blocks, and admonishes clients that don't want that to happen to send request blocks in order.
- It is completely the decision of the server if it wants to enforce (or even check) NUM order.
- Client can try out-of-order and receives unambiguous signal if that isn't supported by server.

The critical section problem

- For one cache-key (URI etc.), only one POST or PUT can be active from the same client.
- Starting another POST/PUT with Block1(NUM=0) kills the reassembly buffer for atomic requests.
- A Block2 response is collected by sending the POST/PUT request with Block2(NUM≠0).
  - client cannot indicate for which of several requests for the same cache-key the response is being collected.
  - server might want to support proper correlation by including ETag response option even for 4.xx responses.
- Workaround: vary cache key (e.g., in URI or a safe elective option).

No text about caching

- Can blocks be individually cached or is it expected that a cache obtains the complete representation before it serves parts of it?
  - Implementation dependent. It is more efficient to enable partial caching (equivalent to a streaming HTTP proxy).
- What are the rules for using a cached block to satisfy a request that is presented to a cache?
  - A partial cached response can be used in place of a complete one.
- Can different blocks have different Max-Age values?
  - Yes. This is hard to avoid as they are transferred at different times.
- What happens when blocks overlap? Does a response with a block update the freshness of the complete representation?
  - Yes.
- Can individual blocks be validated? Does validating a single block validate the complete representation?
  - Yes.
- Does a response with Block1 Option in control usage with the M bit set invalidate cached responses for the target URI?
  - Yes.

Proxying blocks

- **Pretty much obvious for GET.**
- **PUT, POST: origin server may be stateless**
  - **How to combine the 2.01/2.04 responses and their response payloads?**
  - **(Easy answer: hand through the responses. Only works if the block sizes are the same. Does not work for cross-proxy.)**
- **Proxying exacerbates the critical section and makes using the workaround harder.**

If-None-Match and Block1

- **If-None-Match only works on Block1(NUM=0).**
  - **Do not use If-None-Match with Block1(NUM≠0).**

Group Communication for CoAP

Akbar Rahman
Esko Dijk

IETF 88, November 2013

http://www.ietf.org/id/draft-ietf-core-groupcomm-16.txt

Summary of Main Changes

- I-D had multiple updates (Rev. 11 to Rev.16) after IETF-87 (Berlin)

- Extensive editorial and technical updates based on Chair's review (by Carsten Bormann)
  - restructured Section 2.6 (Configuring Group Memberships)
  - fixes & detailing of group membership API Section 2.6;
    now uses RFC 3986 syntax for group IP addresses
  - restructured Section 4 (Deployment Guidelines)

- In section 2.2, provided guidance on how implementers should parse URIs for group communication (#339).

- *(For a more detailed change report per revision: see Backup section)*

Open Ticket #354 (1/3)

- #354: Group configuration API should enable deletion of individual group memberships

- Review input from Peter, Carsten and others was that preferably the group configuration API in section 2.6.2 should enable deletion of individual group memberships.
  - This allows configuration and reconfiguration of a CoAP server by potentially multiple configuring endpoints (e.g. a commissioning tool and a back-end application).

- To accomplish this, some changes are needed in the defined media type "application/coap-group+json" and also in the REST interaction with the group configuration resource.

Open Ticket #354 (2/3)

- Proposed solution – shown by examples below
- Create group membership:

```
POST /coap-group (Content-Format: application/coap-group+json)
     { "n": "All-Devices.floor1.west.bldg6.example.com","a":
     "[ff15::4200:f7fe:ed37:abcd]:4567" }
2.01 Created  (Location: /coap-group/12)
```

- Delete group membership:

```
DELETE /coap-group/12
2.02 Deleted
```

- Update group membership:

```
PUT /coap-group/12
     {"n":"newname.example.com","a":"[ff15::4200:f7fe:ed37:abcd]"}
Res: 2.04 Changed
```

Open Ticket #354 (3/3)

- Reading all group memberships:

```
GET /coap-group
2.05 Content (Content-Format: application/coap-group+json)
{
"8":{"a": "[ff15::f7fe:ed37:14ca]"},
"11":{"n":"floor1.bld6.example.com","a":[ff15::f7fe:ed37:25cb]"},
"12":{"n": "All-Devices.floor1.west.bldg6.example.com","a":
"[ff15::4200:f7fe:ed37:abcd]:4567"}
}
```

- Structure of each group membership is:
  “<membership-ID>”: { JSON membership object}
- This allows GETs / DELETEs by other CoAP clients than those that originally registered the group membership.

Next Steps

- Any other updates that the WG would like to see?
- Is the I-D (finally) ready for WGLC?

Backup

Summary of Changes (1/4)

- I-D had multiple updates (Rev. 11 to Rev.16) after IETF-87 (Berlin)

- Changes from ietf-11 to ietf-12:
  - Removed reference to "CoAP Ping" in Section 3.5 (Group Member Discovery) and replaced it with the more efficient support of discovery of groups and group members via the CORE RD as suggested by Zach Shelby.
  - Various editorial updates for improved readability.

- Changes from ietf-12 to ietf-13:
  - Extensive editorial updates due to comments from the Chair's review (by Carsten Bormann) of the draft. The best way to see the changes will be to do a -Diff with Rev. 12.
  - (The technical comments from the Chair's review were addressed in future revisions)

Summary of Changes (2/4)

- Changes from ietf-13 to ietf-14:
  - Update to address final editorial comments from the Chair's review (by Carsten Bormann) of the draft. This included restructuring of Section 2.6 (Configuring Group Memberships) and Section 4 (Deployment Guidelines) to make it easier to read. Also various other editorial changes.
  - Changed "ip" field to "a" in Section 2.6 (#337)

Summary of Changes (3/4)

- Changes from ietf-14 to ietf-15:
  - In section 2.2, provided guidance on how implementers should parse URIs for group communication (#339).
  - In section 2.6.2.1, specified that for group membership configuration interface the "ip" (i.e. "a" parameter) key/value is not required when it is unknown (#338).
  - In section 2.6.2.1, specified that for group membership configuration interface the port configuration be defaulted to standard CoAP port 5683, and if not default then should follow standard notation (#340).
  - In section 2.6.2.1, specified that notation of IP address in group membership configuration interface should follow standard notation (#342).
  - In section 6.2, "coap-group+json" Media Type encoding simplified to just support UTF-8 (and not UTF-16 and UTF-32) (#344).
  - Various editorial updates for improved readability.

Summary of Changes (4/4)

- Changes from ietf-15 to ietf-16:
  - In section 2.6.2, changed DELETE in group management interface to a PUT with empty JSON array to clear the list (#345).
  - In section 2.6.2, aligned the syntax for IP addresses to follow RFC 3986 URI syntax, which is also used by coap-18. This allows re-use of the parsing code for CoAP URIs for this purpose (#342).
  - Addressed some more editorial comments provided by Carsten Bormann in preparation for WGLC.
  - Various editorial updates for improved readability.

Guidelines for HTTP-CoAP Mapping Implementations

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

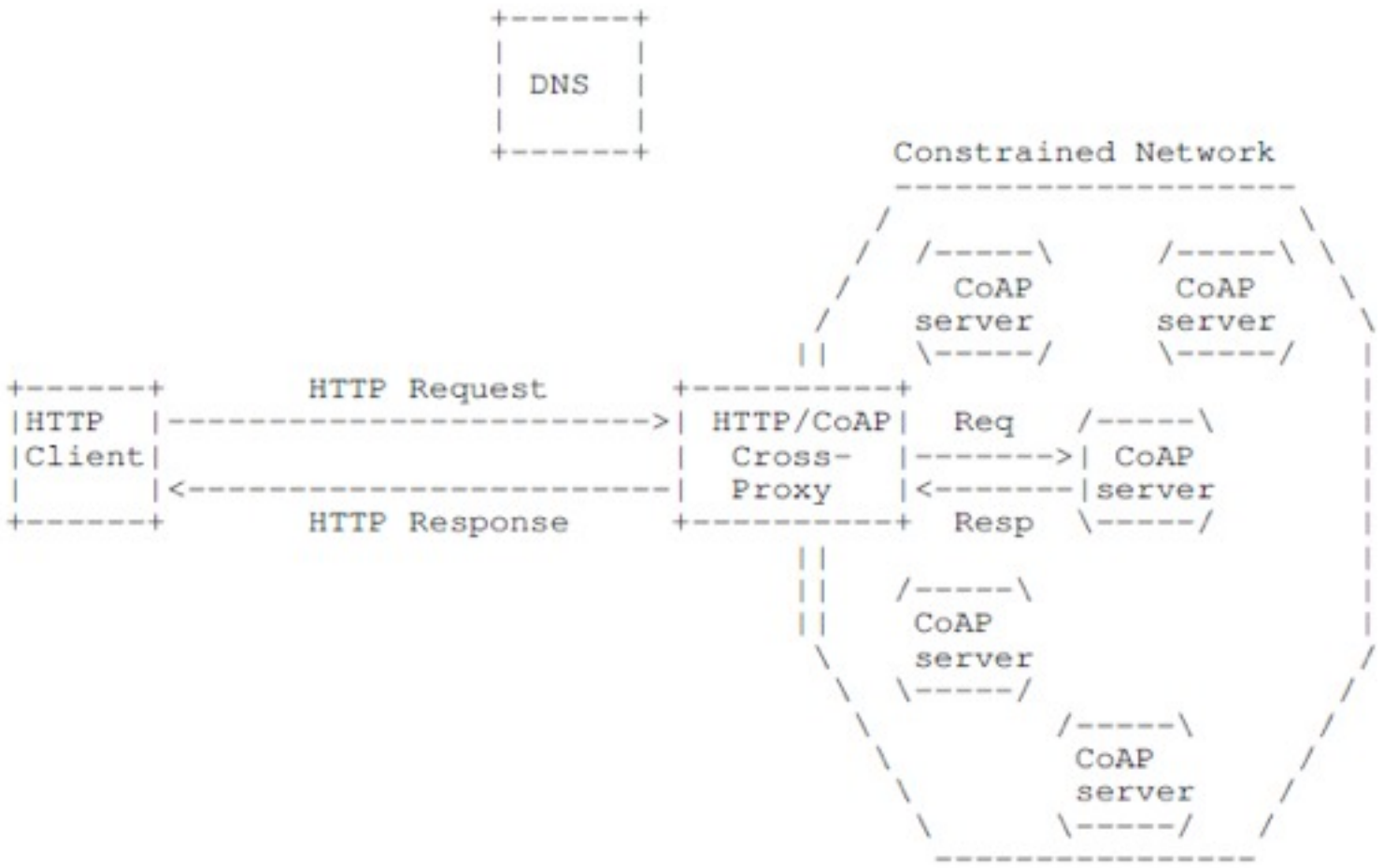
IETF 88, November 2013

http://www.ietf.org/id/draft-ietf-core-http-mapping-02.txt

Main Changes (from IETF-87 Berlin)

- Changed I-D title from “Best Practices” to “Guidelines”
  - As I-D is Informational and not BCP
- After discussions on WG list, selected a HTTP-to-CoAP URI mapping algorithm
  - From the original 5 proposals on the table
- Added simple use cases to illustrate HTTP-to-CoAP mapping
- Added some clarifying text to Security section

Reverse Cross-Protocol Proxy Deployment Scenario



Selected HTTP-to-CoAP URI Mapping Algorithm (1/2)

- From a high level viewpoint, the URI mapping is implemented in a HTTP client by appending the CoAP URI to a HTTP proxy base URI

- For example:
  - Proxy base URI may be http://p.example.com/.well-known/core/
  - CoAP URI may be coap://s.coap.example.com/foo.
  - The mapping embeds the COAP URI into a HTTP URI by appending it, as follows:
  - http://p.example.com/.well-known/core/coap://s.coap.example.com/foo

- Alternative embedding, being discussed among authors (see Appendix):
  - Use '?' instead of '/' to demarcate the start of the coap URI
  - http://p.example.com/.well-known/core?coap://s.coap.example.com/foo

Selected HTTP-to-CoAP URI Mapping Algorithm (2/2)

- The URI template is one of the following two expressions, using the notation of [RFC6570].
  - {+proxy-origin}/.well-known/core/{scheme}://{+authority}{+path-absolute}
  - {+proxy-origin}/.well-known/core/{scheme}://{+authority}{+path-absolute}{+?query}
- Where:
  - proxy-origin identifies the proxy HTTP side as usual scheme "://" authority;
  - scheme is the scheme of the embedded CoAP URI, either 'coap' or 'coaps';
  - authority is the CoAP URI authority. If the host is in IPv6address format, then the '[' and ']' characters MUST be percent-encoded, in order to comply with the syntax defined in Section 3.3. of [RFC3986] for a URI path segment;
  - path-absolute (defined by [RFC3986]) is the absolute CoAP URI path;
  - query (defined by [RFC3986]) is the optional query component of the CoAP URI.

Open Ticket #349

- **Can we use "/.well-known/core" for HTTP-CoAP mapping resource?**
  - All guiding documents (e.g. coap-18, RFC 5785, etc.) imply that "/.well-known/" should only be used for discovery

- Possible resolutions:
  1. **Do it anyway**
     For ease of use and to avoid extra discovery step
  2. **Use an example path that proxy implementations MAY use:**
     `/hc` or `/http-coap`
  3. **Don't define a default mapping resource**
     Make discovery of the http-coap mapping entry point (resource) a first step in the process. This would be done e.g. by HTTP query of /.well-known/core for the resource type (rt) of the mapping function.

- In any case a resource type for http-coap mapping needs to be defined to allow the discovery-based approach 3) above

Open Ticket #351

- Add security implications of proposed default HC URI mapping

- Possible resolution of this issue:
  - Add security implications of proposed default HC URI mapping

  - We may want to take into consideration the security/privacy implications of exposing the HC proxy function to the public, especially given the URI format that we are proposing which exposes details on the internal network.

  - Proposal: add a section 8.3 to the Security Considerations to cover above

Appendix – Further Reducing Parsing Complexity

#1 Target URI encoded in the path (current proposal)

- Host is reg-name:

  Target URI: coap://s.coap.example.com:4567/p?q

  Mapped URI: http://p.example.com/h2c/coap://s.coap.example.com:4567/p?q

- Host is IPv4address:

  Target URI: coap://192.0.2.1:4567/p?q

  Mapped URI: http://p.example.com/h2c/coap://192.0.2.1:4567/p?q

- Host is IP-literal (IPv6address)

  Target URI: coap://[2001:db8::1]:4567/p?q

  Mapped URI: http://p.example.com/h2c/coap://%5B2001:db8::1%5D:4567/p?q

#2 Target URI encoded in the query (new proposal)

- Use '?' instead of '/' to demarcate the start of the coap URI
- Host is reg-name:

  Target URI: coap://s.coap.example.com:4567/p?q

  Mapped URI: http://p.example.com/h2c?coap://s.coap.example.com:4567/p?q

- Host is IPv4address:

  Target URI: coap://192.0.2.1:4567/p?q

  Mapped URI: http://p.example.com/h2c?coap://192.0.2.1:4567/p?q

- Host is IP-literal (IPv6address)

  Target URI: coap://[2001:db8::1]:4567/p?q

  Mapped URI: http://p.example.com/h2c?coap://%5B2001:db8::1%5D:4567/p?q

#1 Parsing Complexity

A bit tricky:

1. Split Mapped URI into components;
2. Remove base path (e.g. `h2c/`) and retrieve the first part of the Target URI up to the optional query;
3. Concatenate it with the query isolated in 1.

#2 Parsing Complexity

Trivial, no copy needed:

1. Search for the first ‘?’: everything after that is the target URI.

NOTE: in both cases the pct-encoding of square brackets surrounding an IP-literal must be removed.

CoRE Resource Directory

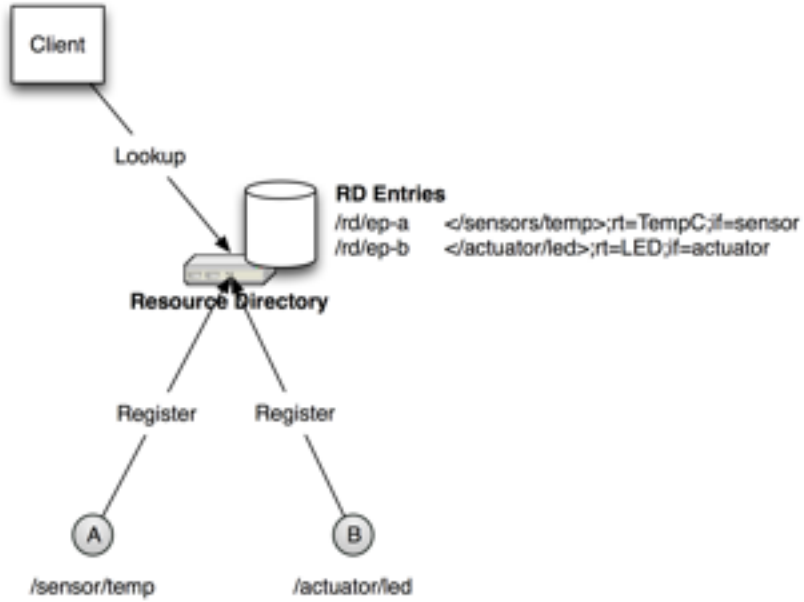
draft-ietf-core-resource-directory-00

Z. Shelby, C. Bormann, S. Krco

CoRE WG, IETF-88 Vancouver

Background

- Not a new concept
  - think web search engine or any link directory
- Defines the interfaces to a Resource Directory
- Based on Web Linking framework and the CoRE Link Format
- Generic REST design for use over HTTP and CoAP
- Part of the OMA Lightweight M2M standard
- Has already been deployed
  - In traffic monitoring systems
  - In street lighting systems
  - For vehicular asset tracking
  - By a major Cellular M2M operator



When are we done?

- We are defining an interface…. let's keep it simple
- Close the WG adoption comments
- Get (even) more implementation experience
- Complete the access control and security considerations
- Integrate a DNS-SD mapping section
- Maintain compatibility with OMA Lightweight M2M
  - Registration, Update and De-registration interfaces
  - Queue mode interface?
- Registry for REST interface parameters

Comments and Known Issues

- Remove the ETag “Validation” feature
- Add a DNS-SD mapping section based on draft-lynn-core-discovery-mapping-02
- More clarification in the Simple Discovery and Discovery sections
- Disallow a GET on the EP entry location e.g. /rd/1234
- Cross-reference the Groupcomm draft WRT the RD group functionality
- Further security considerations on access control
- Improve the lookup function set
  - Some link responses are awkward
  - Separate types of lookups into separate function sets?

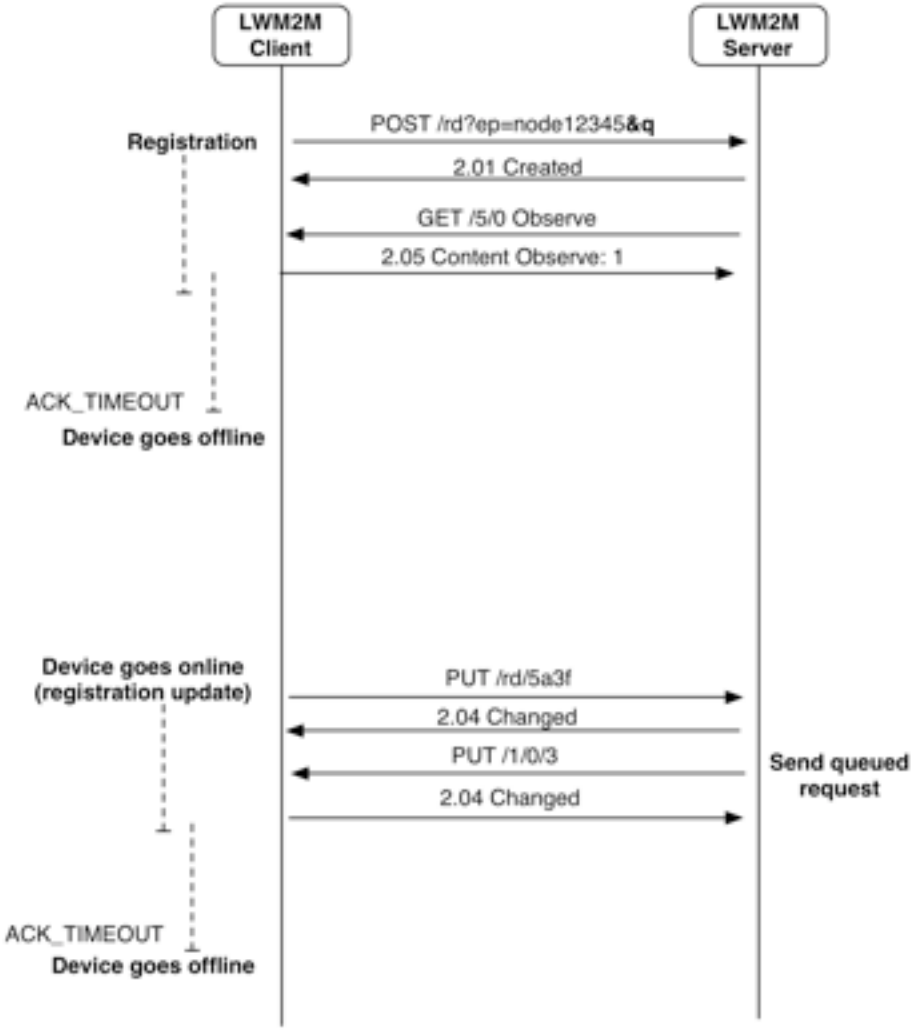
New: Web application support

- Open IoT [www.openiot.org] project has requested better HTTP web application support
  - Large consortium of companies in the UK
  - Interested to move to standard IETF RD solution
- Use case: Resource Directory used by web applications to lookup and store links
- Changes needed for alignment:
  - Add a use case for web applications
  - Explicit support and examples for draft-ietf-core-links-json
  - Explicit examples of more complex links
  - Improvements to the RD lookup interface may be needed

New: Document OMA queue mode

- Lots of WG discussion on “sleepy node” support
- OMA Lightweight M2M has used the RD registration to enable a generic “queue mode” feature
- Uses RD to register and trigger queuing of a Proxy
- Registration: POST /rd?ep={endpoint}&q
  - Indicates endpoint requests queue mode
- Standard RD update to poll for queued requests
- Proposal
  - Document this as an optional feature
  - Focus on RD interface

New: Document OMA queue mode



ACK_TIMEOUT = Constant defining the time the LWM2M Client waits for requests from the server since the last packet sent to the Server. Same value as the CoAP ACK_TIMEOUT.

resource discovery: the charter issue

- »The working group [...] will not develop a general service discovery solution. There is a desire for discovery and configuration features, but the working group has not yet closed in on an specific approach. Thus, the WG may explore these topics and adopt drafts that define requirements or set problem statements, but will not adopt implementable specifications without a recharter.«
- Oops.
- No hurry, but will need a focused charter discussion.

Group 2:“new work”

No Response Needed

TATA CONSULTANCY SERVICES
Experience certainty.

The No-Response option for CoAP

draft-tcs-coap-no-response-option

CoRE WG meeting@IETF 88

Abhijan Bhattacharyya, Soma Bandyopadhyay, Arpan Pal
TCS Innovation Labs

WHY?

- There might be typical M2M scenarios where any response from the receiver to the sender against a request might be considered redundant
  - e.g., frequent update through PUT/POST
- CoAP already provides a non-confirmable (NON) mode of exchange
  - The receiving end-point does not respond with ACK
  - The receiving end-point responds the sender with a status code
- Each NON request or notification triggers 4 Bytes of response on the reverse channel – same as piggybacked ACK for CON
  - N * 4 Bytes of response for N updates in a system over a period of time (that's only the application layer; more bytes are following up from the lower end of the stack)
- N can be quite large for large number of sensors with each one posting frequently
  - Network clogging
  - Server loading
  - Draining of energy if receiving end-point is a constrained node

WHAT?

- A header option: 'No-Response' to suppress responses from the receiver
- The option provides granularity by allowing suppressing a typical class or a combination of classes of responses as additional benefit

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

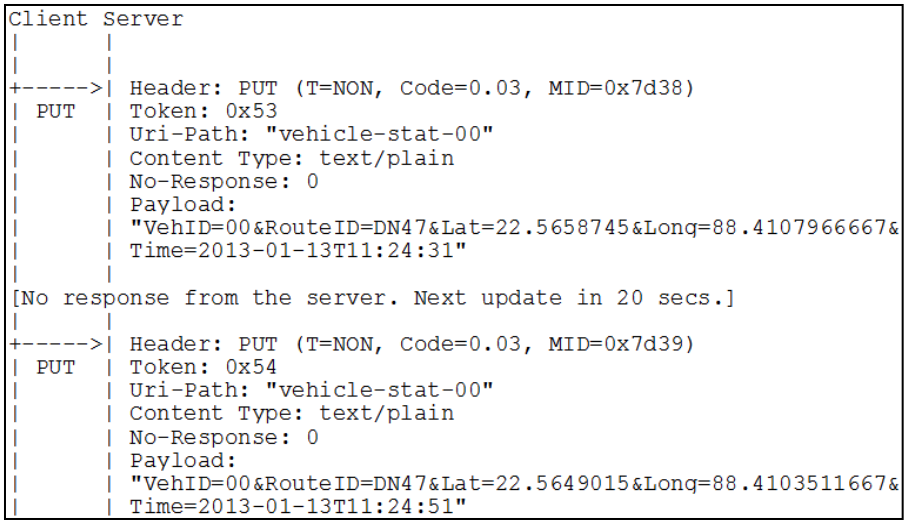
Value	Binary Representation	Description
0	00000000	Suppress all responses (same as empty value).
2	00000010	Allow 2.xx success responses.
8	00001000	Allow 4.xx client errors.
16	00010000	Allow 5.xx server errors.

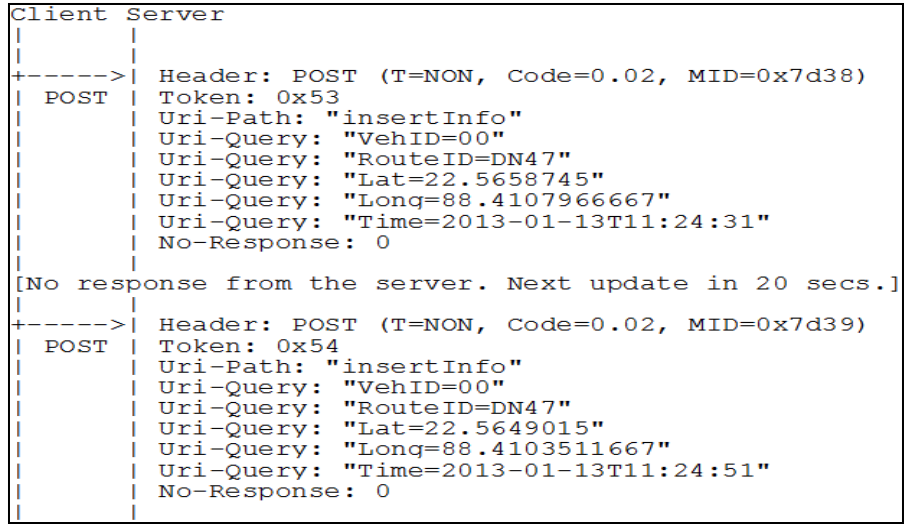
Applicability

- Defined for NON updates when the application permits to be ‘careless’ about the status of the update request
  - NON - PUT/POST
- Obviously not applicable for GET
- Not applicable for DELETE. The requester would always ensure deletion was done

Some example application scenarios

- Frequent update of geo-location from vehicles to backend
  - Imagine updates coming to the backend from thousands of vehicles frequently
  - Using ‘No-Response’ reduces clogging and server load
- Multicasting traffic congestion information to PDAs/ smart-phones in a city
  - The backend not necessarily interested in the delivery status of these updates
  - No-Response may reduce clogging and save energy at the PDA side by reducing the transmission due to responses



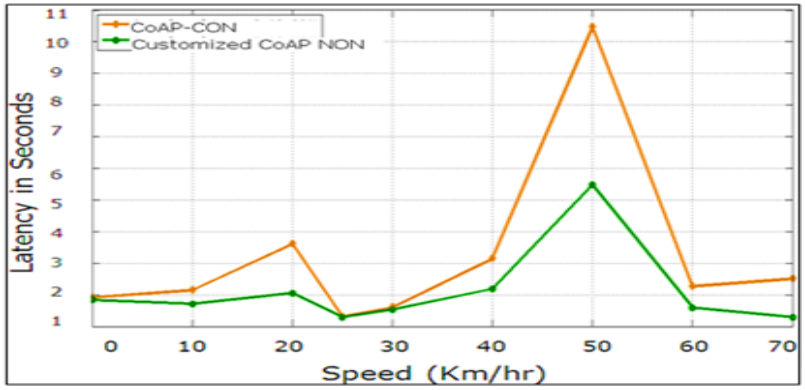


Implementation considerations

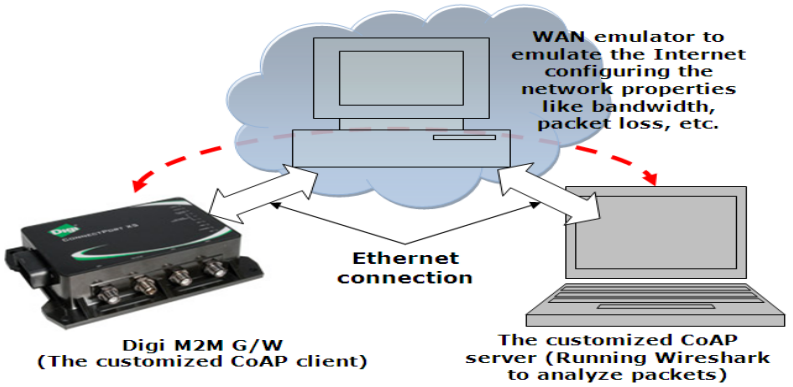
- Use is very much driven by the application scenario and the characteristics of the information to be updated
- How to confirm if the recipient is actually listening?
  - Requests without response suppression may be interweaved
- Granular suppression: The client must wait for a certain application-specific time out for each request
  - If success responses are suppressed it will still have to wait in case a failure occurs
  - Latency will have an upper limit set by application defined time-out
- What if response is allowed only for errors and an error response gets lost
  - Should not be treated as an indication of success
  - Interweaved requests with responses might help revealing these occasions

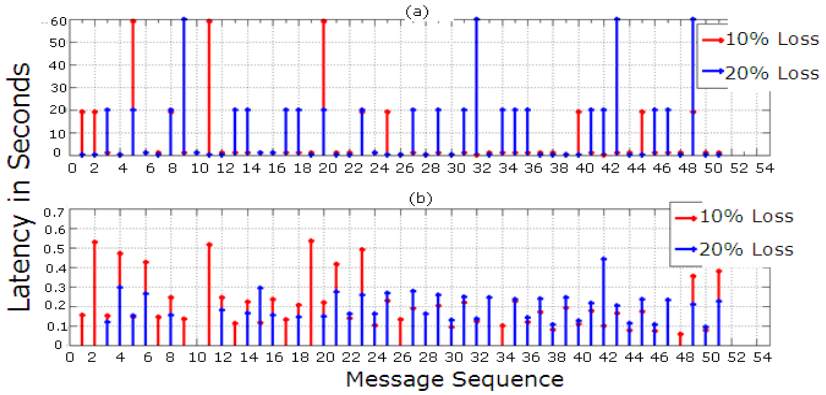
Experiments performed - Location tracking application

On the field



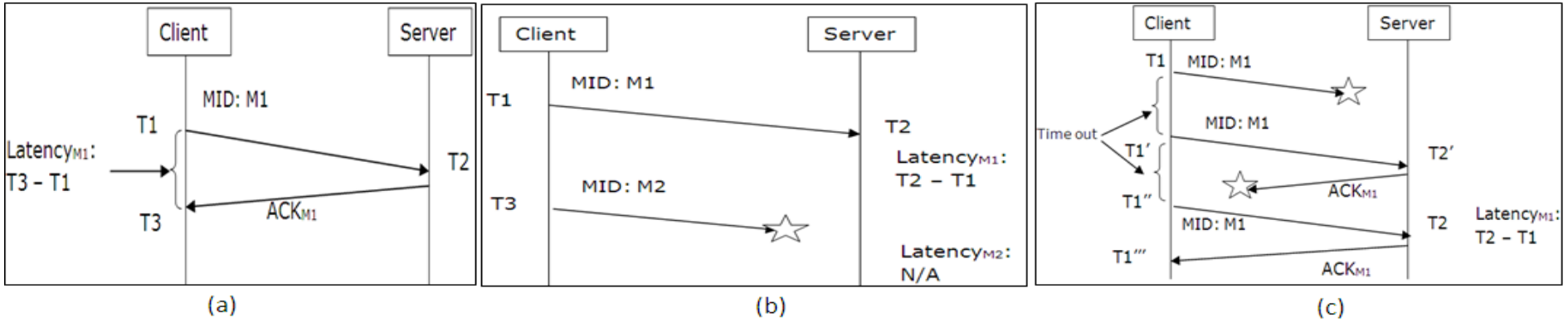
In the lab with emulated network





- Comparison against NON is not displayed
  - The closed-loop latency becomes same as the TIME-OUT set by the application when the response is lost
- CON does not perform well due to re-transmission and duplicate transmissions

Experiments performed – Considerations for Latency Measurements



(a) Closed-loop latency measurement for CoAP-CON. Open-loop latency measurement for (b) customized CoAP-NON (c) CoAP-CON.

Few open issues

- The granularity feature: useful enough to be considered?
- Should Response-filter be a better name for this option?
  - Probably; considering granular suppression.
- What should be the safe-to-reuse interval for a message ID used with a request with No-Response?
- At present this option is unsafe to forward for proxies. Do we need to open it up?
  - Unsafe is probably OK if only the leg between the constrained client and the proxy is constrained
- What if an HTTP non-constrained client frequently updating a constrained CoAP server through a cross-proxy?
  - HTTP will need a response. But any mechanism to intelligently use No-Response yet satisfy the HTTP client?
- Can we temporarily assign some available option number to allow trying with this option?

Thank You

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

✓Blue sheets
✓Scribe(s)

Note Well

This summary is only meant to point you in the right direction, and doesn't have all the nuances. The IETF's IPR Policy is set forth in BCP 79; please read it carefully.

The brief summary:

- **By participating with the IETF, you agree to follow IETF processes.**
- **If you are aware that a contribution of yours (something you write, say, or discuss in any IETF context) is covered by patents or patent applications, you need to disclose that fact.**
- **You understand that meetings might be recorded, broadcast, and publicly archived.**

For further information, talk to a chair, ask an Area Director, or review the following:

BCP 9 (on the Internet Standards Process)

BCP 25 (on the Working Group processes)

BCP 78 (on the IETF Trust)

BCP 79 (on Intellectual Property Rights in the IETF)

Thursday

- 15:20–15:25 Intro
- 15:25–16:10 Access Control/Authorization in CoAP
  - Report from ad-hoc meetings
  - Way forward (discussion, chairs)
- 16:10–16:15 External updates (OMA, plugtest)
- 16:15–16:45 Alternative Transports (KH)
- 16:45–17:00 What do we need for sleepy nodes (AR)
- 17:00–17:10 CoAp Management Interfaces (PV)
- 17:10–17:20 Flextime
  - Implementer info, e.g., draft-bormann-core-roadmap, LWIG
  - Open Mike

Thu 1: Authorization

Authorization (BG, LS, SG, ZS)

Summary of Sundays CoAP AA informal meeting

Bert Greevenbosch
bert.greevenbosch@huawei.com

Introduction

- Since the Berlin meeting, several drafts concerning authentication and authorisation in CoAP have been submitted.
  - Use cases
  - Requirements
  - Solutions
- To discuss and scope the work, an informal meeting took place on Sunday.

Authentication

- **to authenticate**:
  - to prove that something is genuine, real or true
    (Oxford Advanced Learner's Dictionary)
- In the digital domain: prove that credentials (e.g. key or certificate) indeed belong to the asserted owner.
- Often achieved through the binding between a key and an identity, which is guaranteed by a third party.
  - An X.509 certificate contains the ID and public key, and is signed by a certificate authority.
  - Pretty Good Privacy (PGP) defines the web of trust, in which multiple trusted parties generate a new trusted party by signing the new key.
  - A SmartCard / Token solution provides a means for authentication for symmetric keys.

Authorisation

- **authorisation**:
  - (1) official permission or power to do something; the act of giving permission
  - (2) a document that gives somebody official permission to do something
    (Oxford Advanced Learner's Dictionary)
- In the digital domain: allow a party to perform certain actions, such as reading or manipulating a resource.
- Data structures and protocols to signal such permissions.
- Can come in different levels of granularity.
- Authorisation often requires prior authentication.

Current security modes in CoAP

- The current CoAP specification defines how CoAP works together with DTLS.
- Four security modes have been defined:
  - NoSec
    - Null-cipher. No security!
  - Pre-Shared Keys (PSK)
    - Not scalable, but knowledge of key authenticates.
  - Raw Public Key (RPK)
    - Missing authentication - but that can be added.
  - Public Key Infrastructure (PKI) based on X.509
    - Complex
- Authorisation (other than all or nothing) needs more.

Meeting summary

- Informal
- Sunday 12:15 - 16:15
- No. of attendees: 16
- No. of drafts: 9
  - 4 use case and requirements
  - 5 technical

Drafts

- Use cases
  - draft-garcia-core-security
  - draft-greevenbosch-core-authreq
  - draft-schmitt-two-way-authentication-for-iot
  - draft-seitz-core-security-modes
  - draft-seitz-core-sec-usecases
- Solutions
  - draft-gerdes-core-dcaf-authorize
  - draft-pporamba-dtls-certkey
  - draft-selander-core-access-control
  - draft-zhu-core-groupauth

Meeting proceedings

- First a discussion on the use cases and related requirements.
- Thereafter a discussion on the several technical solutions.
- The last part concerned the way forward.
- No official decisions were made during the informal meeting.

Commonality of technical solutions

- In most technical solutions, a third party helps the constrained devices perform the authentication and authorisation work.
- This third party is called AS (or AM).
- There may be two AS-es, one for the client and one for the server.
- Authentication or authorisation of a client to a server may be granted with a ticket provided

Issues

- Which use cases need to be considered at this stage?
- How much granularity in access permissions?
- Trade off between complexity and functionality.
- What does the constrained device need to delegate, and what can it handle by itself?

Work to be done

- Use cases and related requirements (Ludwig)
  - Scoping the work.
- Format of the ticket (Ludwig)
- Transportation of the ticket (Steffi)
- Revoking tickets/credentials/access rights (Bert)
- Authorisation payload (Olaf)
- Lifecycle management (Sandeep)
  - bootstrapping, expiration, revocation, provisioning
- Group authentication

Next steps

- Scoping the work
  - Consolidating the use cases
  - Extracting requirements
  - Making selection of what to do first

Recommendation

- To have a viable solution that can be used in praxis, I recommend to:
  - Consider how much resources and expertise we have.
  - Consider how to gather more expertise.
  - Consider how much meeting time we can spend on it.
  - Consider which will be the best way to create a solution with solid security, within a short timeframe.

CoRE security use cases

Ludwig Seitz
ludwig@sics.se
SICS Swedish ICT AB, Lund, Sweden

Overview

- Informal meeting on CoRE security use cases and resulting requirements
- 11 persons attending
- Scope:
  - Discussing use case drafts
  - How to proceed in the CoRE WG
- Not in scope: Solutions

Main discussion items

- Merging existing use case drafts and use cases from solution drafts
  - With quality control
  - General agreement that it's a good way to go
- Criticism:
  - Current use cases too complex
    - … and may differ a lot between market segments
  - Focus on easy, short term problems
    - Device2Service security management → OMA Lightweight
    - Device2Device would be useful to solve

Goals

- Gather relevant CoRE security use cases
  - Aiming for an Informational RFC
  - 9 persons willing to contribute
- Prevent *”solutions looking for a problem”*
  - Realistic problems, technical & business
- Categorize cases by time frame to solution
  - Concentrate on the simple, short term cases first
- Align with DICE use cases

Summary of Core-AA Transport Meeting on Tuesday, November 5

Stefanie Gerdes

IETF-88, CoRE WG, 2013-11-07

Core-AA Subgroups (work in progress)

- Use cases and requirements (coordinator: Ludwig Seitz)
- Ticket Format (Ludwig Seitz)
- **Ticket Transport (Stefanie Gerdes)**
  - **How to transport the access information to the Resource Server**
- Revoking tickets/credentials/access rights (Bert Greevenbosch)
- Authorization payload (Olaf Bergmann)

Status

- Drafts in this area
  - draft-gerdes-core-dcaf-authorize
  - draft-pporamba-dtls-certkey
  - draft-schmitt-two-way-authentication-for-iot
  - draft-seitz-core-security-modes
  - draft-selander-core-access-control

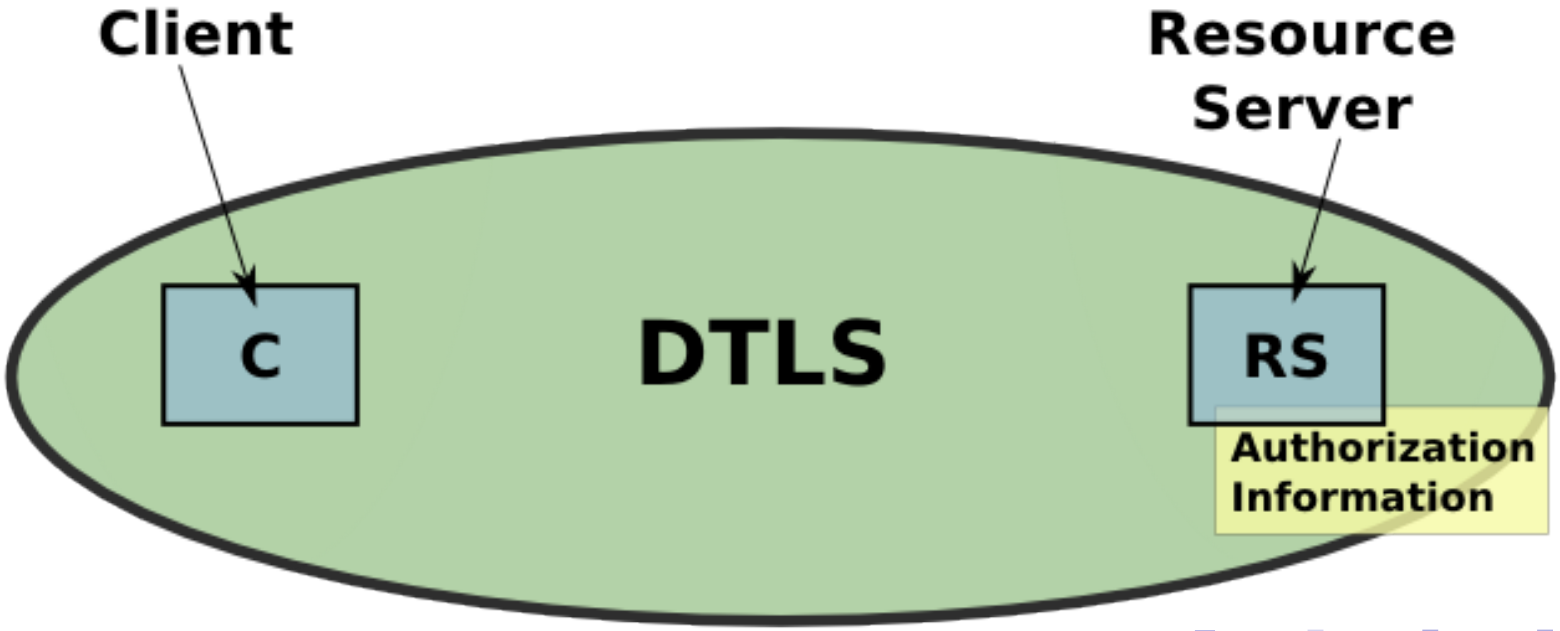
Who was there

- Eight attendees
- Four of the attendees willing to contribute
- Others are interested

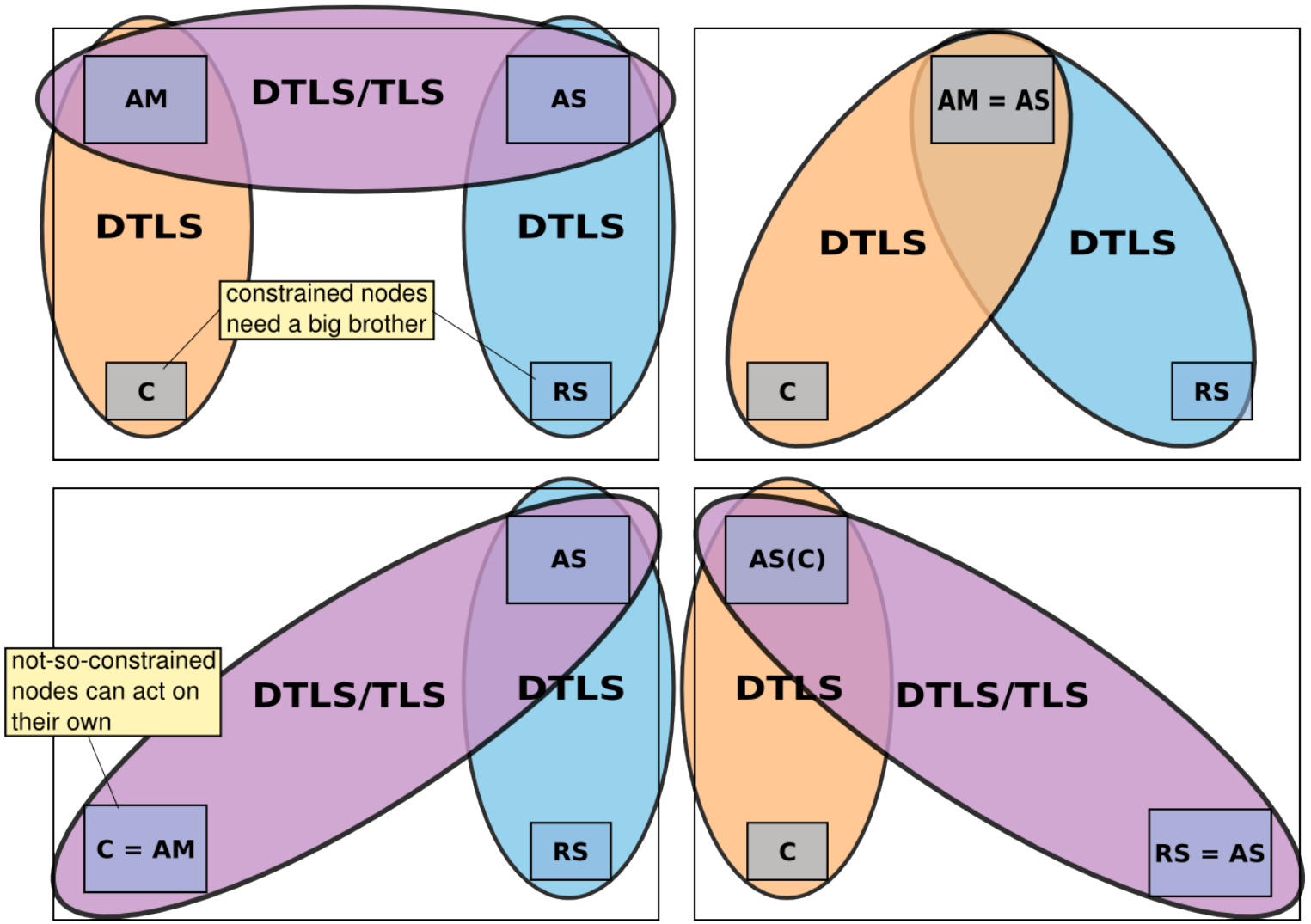
Goals

- Enable a constrained Resource Server to authenticate and authorize requesting client
- Enable a constrained Client to authenticate a Resource Server
- Enable authorized constrained device to constrained device communication
- Enable authorization that ranges from binary (all-or-nothing) to fine-grained
- Enforce Policy of resource owner
- Use DTLS whenever it's possible
- Use CoAP
- Focus on simple but extensible solutions
- Use one or more "big brothers"

What we want


Client
Resource Server
C
DTLS
RS
Authorization Information

Roles


AM
DTLS/TLS
AS
DTLS
DTLS
constrained nodes
need a big brother
C
RS
AM = AS
DTLS
DTLS
C
RS
not-so-constrained
nodes can act on
their own
AS
DTLS/TLS
DTLS
C = AM
RS
AS(C)
DTLS
DTLS/TLS
C
RS = AS

What is missing

- People have a hard time understanding what we are trying to achieve
  - What are the differences between the proposed solutions?
  - Which problem are we trying to solve?
  - Why don't we use existing approaches?

Example for a design choice to be taken

- Break the big problem into smaller pieces
- Present proposals for transferring the authorization information to the resource server
  - Resource Directory?
  - DTLS cookie?
  - PSK-identity?
  - well-known location?

How to proceed

- Merging drafts (if possible)
- Drafts using psk
  - draft-gerdes-core-dcaf-authorize
  - draft-seitz-core-security-modes
  - draft-selander-core-access-control
- Drafts using pk
  - draft-pporamba-dtls-certkey
  - draft-schmitt-two-way-authentication-for-iot
  - draft-seitz-core-security-modes

How to proceed (2)

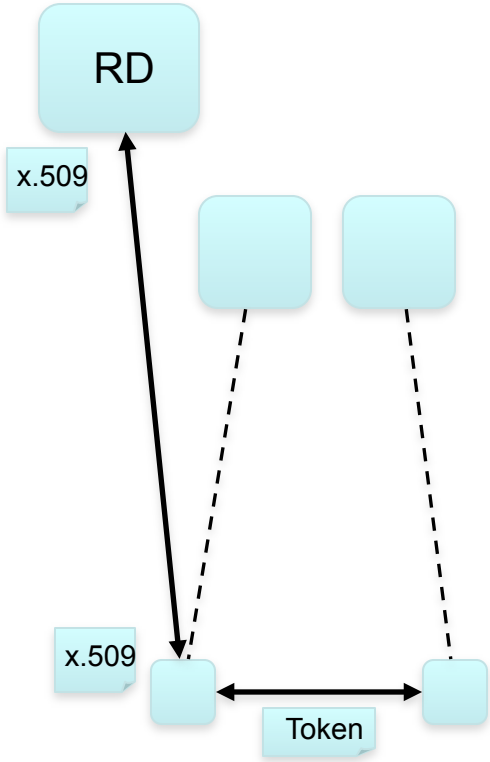
- Clarify problem we want to address
- Identify differences between approaches
- Take discussion about different approaches to the mailinglist
- Enable people to discuss things
- Don't try to solve everything at once (or at all)

CoRE AA: Low-hanging Fruit

1. Align use-cases with DICE

2. Specify X.509 for CoAP endpoints
   - Endpoint Name as Authority Name
   - New SubjectAltName OID
   - Needed by RD and DICE Profile

- Solve the peer-to-peer authorization problem
  - Suitable for use with DTLS
  - Extension/co-existence with existing security management (e.g. OMA Lightweight)



So what do we do? (Rough draft of what may need to be standardized)

1. Use cases, objectives (CoRE, informational)
2. Authorization payload, "ACL format" (CoRE*)
3. Authenticated multi-party transfer of authorization information (WGX?)
   - Note that there are older proposals for that such as draft-jennings-core-transitive-trust-enrollment
4. Cert (X.509) profile (DICE?? CoRE?? WGX??) (DTLS profiling already in DICE)
5. Get all this into the RD (CoRE*)

*) May require charter update

Current CoRE Charter (for reference)

- Security, particularly keying of new Devices, is very challenging for these applications. The WG will work to select approaches to security bootstrapping which are realistic given the constraints and requirements of the network. To ensure that any two nodes can join together, all nodes must implement at least one universal bootstrapping method.
- Security can be achieved using either session security or object security. For both object and session security, the WG will work with the security area to select appropriate security framework and protocol as well as selecting a minimal required to implement cipher suite. CoAP will initially look at CMS (RFC 5652), TLS/DTLS, and EAP.

Thu 2:
Related Work Reports

DICE WG report (ZS)

OMA LWM2M completion (ZS)

Reminder: Internet of Things Plugtest

- When?
  - November 19-22nd, 2013
- Where?
  - Las Vegas
- Who?
  - Implementers world-wide
- How Much?
  - Free!
- Tests:
  - CoAP (Mandatory)
  - Block, Observe (Optional)
  - DTLS for CoAP (Optional)
  - OMA Lightweight M2M (Optional)

Thu 3:“new work” (continued)

Alternate Transports

The need for alternate transports

- **Non-IP transports: e.g., SMS**
- **Alternate IP transport protocols: e.g., TCP, Websockets**
- **Combine non-IP and IP: e.g. SMS request, UDP response**

Issues

- **Encapsulation**
  - **Delimiting in stream transports (e.g., TCP)**
  - **Do we need CON/NON/ACK/RST for acknowledged transports?**
  - **Data transparency (Base64url…)**
- **URIs**
  - **Identifying the desired transport(s)**
  - **Enabling transport selection**
  - **Having multiple transports for the *same* resource**
- **Return Path**
  - **How to indicate the desired recipient for the response**

Endpoint identification

- **UDP: IP-Address + Port**
  - **Generalize to Address + Port + Transport (for TCP…)**
- **SMS: MSISDN**
- **Websockets: WS (WSS) URI**

- **Use indirection? DNS/SRV? DHTs?**

- **Do this in a way that can be used both in URIs and for Return-Path?**

Berlin and beyond: CoAP Alternative Transport URIs

draft-silverajan-core-coap-alternative-transports

Alternative Transports

draft-silverajan-core-coap-alternative-transports

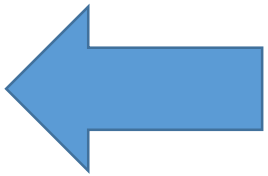
IETF 88

Bilhanan Silverajan

Teemu Savolainen

Klaus Hartke

Design Issues



- Identify the resource
  - CoAP resources are organized hierarchically;
  - Need to identify
    - the transport,
    - the server endpoint accessible over the transport,
    - the resource exposed by the server.
  - Tell proxy about a resource to retrieve or update
  - Follow links embedded in resource representations
- Identify the client
  - So the server can return a response to the client
  - E.g., UDP endpoint, TCP connection, …
- Describe how to encapsulate and transport CoAP requests and responses between client and server

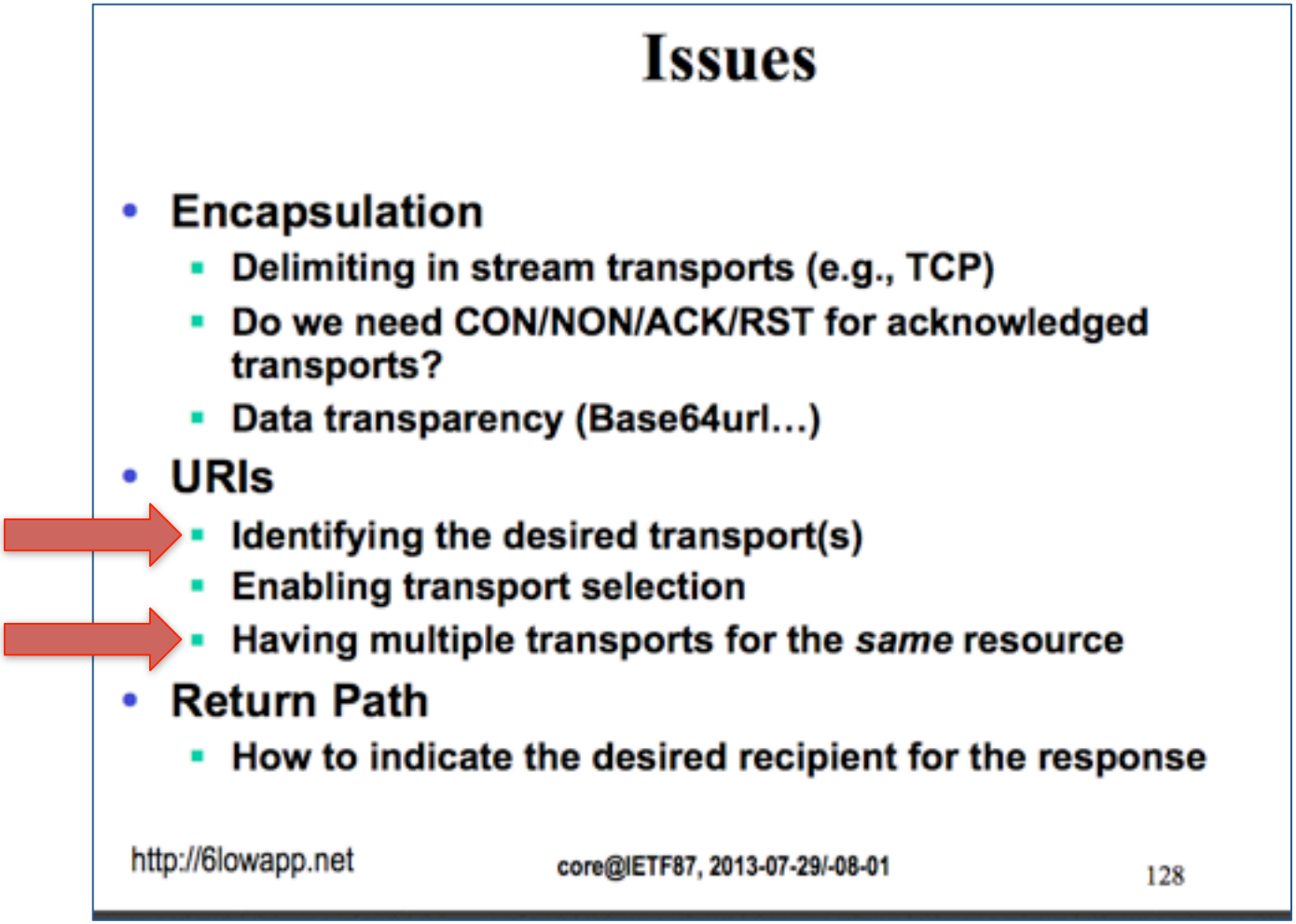
Challenges (i)

1. RFC 3986
   - URIs and (resolution of) URI references
   - scheme://authority/resource
2. May need to express wildly different authorities
   - *TCP* [2001:DB8::1234]:5683
   - *SMS* +1-800-555-0199
   - *Bluetooth* 01:23:45:67:89:ab
   - *WebSockets* ws://example.com/path?query
3. “Interesting” URI parsers
   - http://3571666038/ == http://212.227.76.118/
4. Can we find a unified syntax for all transports?

Challenges (ii)

4. Situation is different from HTTP
   - Transports may be very different (IP and non-IP)
   - Transports may not always be available
   - Cannot connect and upgrade
   - Probably cannot use DNS
5. May expose same resource over multiple transports
   - Server exposes multiple endpoints (cost, availability)
   - Share cache entry for different URIs (URI aliasing)
6. May return response over a different path then the request

Our Scope, From Berlin Meeting:



What happened in Berlin?

- We came to Berlin with draft -02 for Plan A:
  `coap+<transport>://authority/resource?query`

- Some other feedback / URL requirements:
  - Consult HTTP experts
  - Combine several of the presented URL ideas
  - Explore new ideas and requirements
    - One such idea pertains to relative URI references and to resist designing new resolution algorithms for any new URI format

Design Considerations

- Currently identified transports considered are
  - SMS, WebSockets, TCP and RS-232
- Conformance to RFC 3986 Generic URI Syntax
- Avoiding URI aliasing
- Properly resolve absolute and relative URIs
- Avoidance of reliance on DNS

URI Format 1 (Section 2.1)

Transport in Scheme

- Transport type within the scheme name
  - coap+sms://0015105550101/sensors/temperature
  - coap+tcp://example.org:5683/sensors/temperature
  - coap+ws://example.org:80/ws_endpoint?/sensors/temperature
- Main points:
  - Preferred place to put transport info, by some of the consulted HTTP experts
  - Multiple schemes, each alternative transport free to choose the syntax of its URI format
  - URI parsing with generic parsers proved to be straightforward
  - URI aliasing issue manifests with multiple transports for same resource
  - Relative URI parsing works as expected, if resource is in path component

URI Format 2 (Section 2.2)

Transport in Authority

- Prefix the authority with transport name
  - coap-at://sms-001510555010l/sensors/temperature
  - coap-at://tcp-example.org:5683/sensors/temperature
- Main points
  - Syntactically almost similar to keeping transport in scheme
  - Single scheme, each alternative transport must conform to the syntax of the URI format described
  - Special care to properly encode transport endpoint address in authority component, for parsers we tested
  - URI aliasing issue manifests with multiple transports for same resource
  - Relative URI parsing works as expected on described resource

URI Format 3 (Section 2.3.1)

Transport in path

- URI has a locator for transport endpoint, and a name (identifier) for CoAP resource

  coap-at:<transport>://<location>?resource_name

  - coap://example.org/sensors/temperature (UDP)
  - coap-at:tcp://example.org:1234?coap://example.org/sensors/temperature/ (TCP)
  - coap-at:ws://example.org:80/endpoint?coap://example.org/sensors/temperature (WebSockets)
- Provides the most flexible option for alternative transports among the three
- Single scheme, each alternative transport conforms to the syntax of the URI
- Easier to encode transport endpoint address in URI path component for generic parsing than proposal 2 (Section 2.2)
- Clean separation of transport location and resource naming overcomes URI aliasing
- Relative URI parsing does not work on CoAP resource (which is in query component)
  - We could replace '?' with '/' or another separator, so there are no query components, so that relative URI parsing works as expected on CoAP resource

URI Format 4 (New Work)

Transport in path

- URI has a locator for transport endpoint, and a name (identifier) for CoAP resource

  coap-at://authority/<transport_name>[:transport_path]/resource_name

  - coap://example.org/sensors/temperature (UDP)
  - coap-at://example.org:1234/tcp/coap://example.org/sensors/temperature/ (TCP)
  - coap-at://example.org:80/ws:ws_endpoint/coap://example.org/sensors/temperature (WebSockets)
- Provides the most flexible option for alternative transports among the three
- Single scheme, each alternative transport conforms to the syntax of the URI
- Encoding transport endpoint address is as easy as proposal 1
- Clean separation of transport location and resource naming overcomes URI aliasing
- Relative URI parsing works on CoAP resource (which is in path component)

Which design choices to omit?

- Overcoming URI aliasing
- Eliminating Locator/Identifier Split
- Relative URI references
- Differing request/response paths
- Non-reliance on DNS

Overcoming URI Aliasing

- URI aliasing refers to the discouraged practice of having multiple unique URIs for the same resource
- When applied to alternative transports, URI aliasing can occur for 1 resource available over multiple transports
- Impacts whether a proxy can serve a cached resource retrieved over a different transport

Eliminating Locator Identifer Split

- Last 2 formats advocate separating the locator from the identifier of the CoAP resource
- CoAP URI resides in a name space and is encapsulated with a Locator
- This decision impacts CoAP URIs described in Section 2.3

Relative URI references

- If a CoAP resource is available as a query component, the RFC 3986 resolution algorithm when a relative URI is encountered, fails
- This impacts URI describing CoAP over websockets as well as URI format proposed in section 2.3.1

Differing Request/Response Paths

- A request sent by Client A using Transport X is fulfilled by a Server B sending the response over Transport Y back to Client A
- This impacts the message exchange model and the CoAP options proposed for CoAP over SMS

Non-reliance on DNS

- DNS can be exploited as a naming infrastructure for discovery of
  - Different alternative transports available for a server or domain
  - Weights and priorities that can aid in transport selection when multiple options are present
- Clients might not possess enough storage and computation to use to perform DNS service records
- This impacts non-IP and M2M communication

Sleepy

Sleepy Devices: Do we need to Support them in CORE?

Akbar Rahman

IETF 88, August 2013

http://www.ietf.org/id/draft-rahman-core-sleepy-nodes-do-we-need-00.txt

Introduction

- At IETF-87 (Berlin), it was suggested to review/summarize the CORE WG interest on the topic of Sleepy Node support
- Specifically whether the WG feels that support of sleepy endpoints is required for the CoAP protocol, CORE Link Format, CORE Resource Directory, etc.
- Alternatively, whether the WG feels that Sleepy Node support can be completely done outside CORE such as in the lower Layer 2 (MAC) scheduling and/or in Layer 7 (application) logic

CORE I-Ds Related to Sleepy Nodes (1/3)

- There have been multiple drafts in the CORE WG related to the subject of Sleepy Nodes including:
  - [I-D.rahman-core-sleepy-problem-statement] summarizes the overall problem space of Sleepy Nodes
  - [I-D.cao-core-aol-req] defines requirements for Sleepy Nodes to behave as if they are "always on“
  - [I-D.dijk-core-sleepy-reqs] defines requirements for Sleepy Nodes based on home and building control use cases
  - [I-D.rahman-core-sleeping] defines general requirements for Sleepy Nodes
  - [I-D.bormann-core-roadmap] provides a classification and overview of CORE drafts (and features) including a section on Sleepy Nodes

CORE I-Ds Related to Sleepy Nodes (2/3)

- [I-D.arkko-core-sleepy-sensors] describes a sensor network implementation and shows how different communication models affect implementation complexity and energy consumption (including Sleepy Node support)
- [I-D.giacomin-core-sleepy-option] defines a proxy that acts as a store-and-forward agent for a Sleepy Node
- [I-D.castellani-core-alive] defines a new CoAP message type which the Sleepy Node multicasts to all interested devices when it wakes up
- [I-D.fossati-core-publish-option] allows an endpoint to temporarily delegate authority of its resources (when it is sleeping) to a proxy server that is always on
- [I-D.fossati-core-monitor-option] extends the Observe functionality to handle the scenario when both the server and clients are Sleepy Nodes

CORE I-Ds Related to Sleepy Nodes (3/3)

- [I-D.dijk-core-sleepy-solutions] defines an architectural approach to support Sleepy Nodes
- [I-D.rahman-core-sleepy] defines new parameters that describe an endpoint's sleepy characteristics and stores them in the Resource Directory
- [I-D.vial-core-mirror-server] defines a special type of Resource Directory from which endpoints can fetch the resource regardless of the (sleep) state of the server

WG Email List Poll for Sleepy Node Deliverable

- A pulse on interest on the topic was taken on the WG Email list by asking the following question: "Should we have a CORE deliverable for CoAP support of Sleepy Nodes?"

- The interesting (but non-normative) results were as follows:
  - Votes FOR: 9
  - Votes AGAINST: 3

Summary

- There have been over ten drafts related to the concept of CORE support of Sleepy Nodes
- The WG Email list pulse check on the topic showed that there was good interest in the topic of Sleepy Nodes.
  - However there were some important and high profile dissenters that argued against having such a topic in CORE
- Another point to consider is that during WG discussions, the CORE Mirror Server [I-D.vial-core-mirror-server] is sometimes referred to as the "existing" solution for CORE Sleepy Node support
  - However, this draft was never adopted as a WG draft

Next Steps

- Does the WG want to keep discussing Sleepy Nodes in CORE?
  - CoAP Protocol
  - CORE Link Format
  - CORE Resource Directory
  - Etc.
- Sub-Question:
  - Does the WG want to see some immediate development of the Mirror Server concept (which seems to have near unanimous support)?

CoRE working group

draft-vanderstok-core-comi-01

CoAP Management Interfaces

Peter van der Stok, Bert Greevenbosch

November 7,2013

Motivation

MIB and SNMP v.x are most popular management tools today

There is a large MIB installed base
Any new management approach should leverage this invested MIB effort

SNMP provides many aspects of CoAP
- Request response
- Preferably, one packet requests and responses
- Security

Wish exists:
- RESTful interface to MIB to reduce application development effort
- Integrate SNMP + CoAP to reduce code complexity and stack size

CoMI approach (1)

Interface specified as:

name	path	RT	Data type
management	/mg	Core.mg	n/a
MIB	/mg/mib	Core.mg	Application/json
MIB	/mg/mib	Core.mg	Application/cbor
MIB	/mg/mib	Core.mg	Application/xml
MIB	/mg/mib	Core.mg	Application/exi

Reasonable in size:

- cbor (but, table names indirection)
- exi (but, schema versions)

MIB specified with:

- Descriptor name: `sysUpTime`
- ASN.1 OBJECT-IDENTIFIER: `1.3.6.1.2.1.1.3`

CoMI approach (2)

For presentation: Payload in pseudo format instead of EXI, CBOR,..

Single MIB access

```
REQ: GET example.com/mg/mib/sysUpTime
RES: 2.05 Content (Content-Format:application/xxxx)
(sysUpTime, "123456")
```

Table, single row access

```
REQ: GET example.com/mg/mib/ipNetToMediaTable?row=1
RES: 2.05 Content (Content-Format: application/xxxx)
(ipNetToMediaIfIndex , 1)
(ipNetToMediaPhysAddress , "00:00::10:01:23:45")
(ipNetToMediaNetAddress, "10.0.0.51")
(ipNetToMediaType , "static")
```

Multiple MIB access

```
REQ: GET example.com/mg/mib
(sysUpTime, null)
(ipNetToMediaTable, null)
RES: 2.05 Content (Content-Format:application/xxxx)
(all following pairs)
```

Flextime