RESTful web services for IoT
whoami

- Bart Lemmens ([blemmens@etro.vub.ac.be](mailto:blemmens@etro.vub.ac.be))
- MSc Industrial Engineering @ EhB
- PhD @ VUB (ETRO-COMO)
  WSN & IoT (MAC, synchronization, ...)
- 6LoWPAN TETRA-project
  Implementation of web services

DISCLAIMER: I’m a researcher, not a professional web developer
what will I talk about

• Why RESTful IoT?
• CoAP: HTTP for IoT
• Erbium (Contiki) & Californium (Java)
• Our research work
• 6LoWPAN TETRA-project
Why RESTful IoT?
web app/service development

1. Get great idea
2. Design RESTful API
3. Implement
4. ...
5. Profit
Great idea #1
Great idea #2
Great idea #3

http://lifx.co
Great idea #3

http://nest.com
Great idea #3

http://zolertia.com
Great idea #3
Back-end communication

- MQTT
- ZigBee
- Z-Wave
- Wireless HART
- DASH7
- enocean
Back-end communication

Problems:

• Need knowledge of back-end message model
• How to combine different back-end technologies
• Intra-networks: on-premise application gateway required
Great idea #3 (version 2)
Competition

xively by LogMeIn

waylay.io

IFTTT
Best idea ever
Problems

• Constrained devices (100KB ROM, 10KB RAM)
• Constrained wireless network
  – Lossy
  – Low-bandwidth
• Connection to Internet (solved: 6LoWPAN)

We need RESTful protocols for IoT
Constrained RESTful Environments (core)
CoAP

- Constrained Application Protocol
- Lightweight HTTP
- Datagram-oriented transport (UDP)
  - No state
  - Less overhead
- Binary packed header
CoAP

- **Message** layer for reliable/asynchronous exchange
- **Request/Response** layer for resource manipulation
CoAP – Messages

4 message types
• Confirmable (CON)
• Non-confirmable (NON)
• Acknowledgement (ACK)
• Reset (RST)

Reliability/Deduplication
• Message ID (MID)
CoAP – Messages

Reliable transmission

Client → Server
- CON [0x7d34]
- ACK [0x7d34]

Unreliable transmission

Client → Server
- NON [0x01a0]
CoAP – Request/Response

Very similar to HTTP

• Method code / Response code
  – 0.01-0.31: request
  – 2.xx: successful
  – 4.xx: client error
  – 5.xx: server error

• Meta-data (including resource URL) in options

• Token (match response to request)
## CoAP – Messages

### Compact fixed-length header (4 bytes)

<table>
<thead>
<tr>
<th>Ver</th>
<th>T</th>
<th>TKL</th>
<th>Code</th>
<th>Message ID</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Token (if any, TKL bytes) ...</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Options (if any) ...</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Payload (if any) ...</th>
</tr>
</thead>
</table>
# CoAP – Options

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option Delta</td>
<td>Option Length</td>
<td>1 byte</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/</td>
<td>/</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option Delta</td>
<td>Option Length</td>
<td>0-2 bytes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(extended)</td>
<td>(extended)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/</td>
<td>/</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option Value</td>
<td>0 or more bytes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(extended)
## CoAP – Options

<table>
<thead>
<tr>
<th>No.</th>
<th>C</th>
<th>U</th>
<th>N</th>
<th>R</th>
<th>Name</th>
<th>Format</th>
<th>Length</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td>If-Match</td>
<td>opaque</td>
<td>0-8</td>
<td>(none)</td>
</tr>
<tr>
<td>3</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td>Uri-Host</td>
<td>string</td>
<td>1-255</td>
<td>(see below)</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td>ETag</td>
<td>opaque</td>
<td>1-8</td>
<td>(none)</td>
</tr>
<tr>
<td>5</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>If-None-Match</td>
<td>empty</td>
<td>0</td>
<td>(none)</td>
</tr>
<tr>
<td>7</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td>Uri-Port</td>
<td>uint</td>
<td>0-2</td>
<td>(see below)</td>
</tr>
<tr>
<td>8</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
<td>Location-Path</td>
<td>string</td>
<td>0-255</td>
<td>(none)</td>
</tr>
<tr>
<td>11</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>Uri-Path</td>
<td>string</td>
<td>0-255</td>
<td>(none)</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td>Content-Format</td>
<td>uint</td>
<td>0-2</td>
<td>(none)</td>
</tr>
<tr>
<td>14</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>Max-Age</td>
<td>uint</td>
<td>0-4</td>
<td>60</td>
</tr>
<tr>
<td>15</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>Uri-Query</td>
<td>string</td>
<td>0-255</td>
<td>(none)</td>
</tr>
<tr>
<td>17</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>Accept</td>
<td>uint</td>
<td>0-2</td>
<td>(none)</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>x</td>
<td></td>
<td>x</td>
<td>Location-Query</td>
<td>string</td>
<td>0-255</td>
<td>(none)</td>
</tr>
<tr>
<td>35</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td>Proxy-Uri</td>
<td>string</td>
<td>1-1034</td>
<td>(none)</td>
</tr>
<tr>
<td>39</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td>Proxy-Scheme</td>
<td>string</td>
<td>1-255</td>
<td>(none)</td>
</tr>
<tr>
<td>60</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>Size1</td>
<td>uint</td>
<td>0-4</td>
<td>(none)</td>
</tr>
</tbody>
</table>
# CoAP – Content-Format

<table>
<thead>
<tr>
<th>Media type</th>
<th>Encoding</th>
<th>Id.</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>text/plain; charset=utf-8</td>
<td>-</td>
<td>0</td>
<td>[RFC2046][RFC3676][RFC5147]</td>
</tr>
<tr>
<td>application/link-format</td>
<td>-</td>
<td>40</td>
<td>[RFC6690]</td>
</tr>
<tr>
<td>application/xml</td>
<td>-</td>
<td>41</td>
<td>[RFC3023]</td>
</tr>
<tr>
<td>application/octet-stream</td>
<td>-</td>
<td>42</td>
<td>[RFC2045][RFC2046]</td>
</tr>
<tr>
<td>application/exi</td>
<td>-</td>
<td>47</td>
<td>[EXIMIME]</td>
</tr>
<tr>
<td>application/json</td>
<td>-</td>
<td>50</td>
<td>[RFC4627]</td>
</tr>
</tbody>
</table>
CoAP – Message Size

• Avoid fragmentation (MTU = 127)
  – Block-wise transfer for larger messages

• Limited buffer size at constrained server
  – Response code 4.13 (Request entity too large)
  – Size1 option to indicate maximum acceptable size
CoAP – Cache & Proxy

Caching

• Important on constrained lossy networks
• Freshness: Max-Age (default 60)
• Validity: ETag

Proxy

• Security/Performance
• CoAP-HTTP & HTTP-CoAP cross-proxy
CoAP – Observations

Retrieve representation of resource up to date

• Request:
  – GET
  – Observe option (register/deregister)

• Response/Notification:
  – Notifications can be confirmable
  – Observe option (ordering)
CoAP – Observations

Client | Server
--- | ---
GET /temperature
Token: 0x4a
Observe: register

Registration

2.05 Content
Token: 0x4a
Observe: 12
Payload: 22.9 Cel

Notification of the current state

2.05 Content
Token: 0x4a
Observe: 44
Payload: 22.8 Cel

Notification upon a state change

2.05 Content
Token: 0x4a
Observe: 60
Payload: 23.1 Cel

Notification upon a state change
CoAP – Discovery

Service discovery

• all-coap-nodes multicast address
  – IPv4: 224.0.1.187
  – IPv6: FF02::FD (link-local)
    FF05::FD (site-local)

Resource discovery

• /.well-known/core
• CoRE link format (web linking)
CoAP – CoRE Link Format

Attributes

• **rt**: resource type (e.g. temperature-sensor)
• **if**: interface description (e.g. sensor)
• **sz**: maximum size estimate
• **ct**: possible content-type formats
• **obs**: observable resource
CoAP – CoRE Link Format

Example

REQ: GET /.well-known/core

RES: 2.05 Content
</temp>;rt="temp-sensor";if="sensor",
</light>; rt="light-sensor"; if="sensor“,
</lamp>;rt="lamp-actuator";if="actuator"
CoAP – CoRE Interfaces

Defines a set of **if** types

<table>
<thead>
<tr>
<th>Interface</th>
<th>if=</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link List</td>
<td>core.ll</td>
<td>GET</td>
</tr>
<tr>
<td>Batch</td>
<td>core.b</td>
<td>GET, PUT, POST (where applicable)</td>
</tr>
<tr>
<td>Linked Batch</td>
<td>core.lb</td>
<td>GET, PUT, POST, DELETE (where applicable)</td>
</tr>
<tr>
<td>Sensor</td>
<td>core.s</td>
<td>GET</td>
</tr>
<tr>
<td>Parameter</td>
<td>core.p</td>
<td>GET, PUT</td>
</tr>
<tr>
<td>Read-only Parameter</td>
<td>core,rp</td>
<td>GET</td>
</tr>
<tr>
<td>Actuator</td>
<td>core.a</td>
<td>GET, PUT, POST</td>
</tr>
<tr>
<td>Binding</td>
<td>core.bnd</td>
<td>GET, POST, DELETE</td>
</tr>
</tbody>
</table>
CoAP – Ontologies

SenML
• Media type for sensor measurements
• JSON, XML, EXI serialization

Semantic IoT
Many research proposals
Erbium & Californium
Erbium & Californium

Matthias Kovatsch (http://people.inf.ethz.ch/mkovatsc/)

• Copper (Firefox)
• Erbium (Contiki)
• Californium (Java)
Contiki

- OS for constrained devices
- 6LoWPAN network stack
- ContikiMAC RDC
- Cooja network simulator
Erbium

• C (Contiki)
• Client/Server
• Simple resource declaration
  – RESOURCE(…)
  – PERIODIC_RESOURCE(…)
  – EVENT_RESOURCE(…)
• Management of observations
• Creates /.well-known/core
Californium

- Java
- Maven dependency
- Client/Server
- Both blocking & non-blocking requests
- DTLS security
- Proxy (HTTP-CoAP)
Research work
Performance evaluation
Performance evaluation

Energy consumption

(CoAP server mote)

Energy consumption (mWs)

Server mote's mode

RX: CoAP > HTTP
TX: CoAP > HTTP
LPM: CoAP > HTTP
CPU: CoAP > HTTP
Performance evaluation

Response time

- CoAP 1-hop
- HTTP 1-hop
- CoAP 2-hop
- HTTP 2-hop

Client request

Response time (ms)
Bandwidth/Energy/Latency

- Bandwidth: Decreasing with sleep
- Energy: Increasing with sleep
- Latency: Increasing with sleep
Other

• Multicast in RPL
  – Service discovery
  – Energy-efficient algorithms

• Observation attributes
  – Threshold
  – Reduce communication

• CoAP on smart phones
6LoWPAN TETRA-project
Overview

• Combine all technologies
  – Embedded HW/SW: sensor/actuator
  – Infrastructure (edge routers)
  – Security
  – Web application
    • Administration
    • Visualization
    • Automation
Web application

• Google Web Toolkit
• Californium  
  used as CoAP client
• Async-IO Atmosphere  
  cross-platform server side push of observations
Observable actuators
In-network automation