M2M, IoT, DEVICE MANAGEMENT: ONE PROTOCOL TO RULE THEM ALL?

Julien Vermillard, Sierra Wireless
Who am I?

Software Engineer at Sierra Wireless, implementing various protocols for AirVantage cloud service.

Apache Software Foundation member. Initial Eclipse committer on Californium and Wakaama.

Disclaimer: opinions expressed are mine :o)
M2M/IoT protocols: MQTT and CoAP

Device management:
TR-069, OMA-DM, LWM2M

Application + DM => Protocol hell.
M2M/IoT Applications

My definition:

Large fleet of cloud connected devices *solving a business problem*.

(Ex: oil pump monitoring, truck fleet tracking)
M2M/IoT versatile protocols

Usual suspects:

MQTT
CoAP
XMPP
MQ Telemetry Transport

Very simple and light protocol on top of TCP.

Good fit for wireless applications.

Publish/Subscribe paradigm.

Websocket support.
On the wire messages:
- connect (with or without authentication)
- publish/puback
- subscribe/suback
- ping/pingack for keepalive
- disconnect

And voilà!
MQTT QoS levels

0 = fire and forget
1 = at least once
2 = exactly one time delivery
MQTT Topic example

- a sensor pushes telemetry values on some topics
  greenhouse/42/temperature
  greenhouse/42/humidity
  greenhouse/42/luminosity

- actions are on another topic
  greenhouse/42/open-the-roof

- payload format is free (json, binary, whatever..)
Uses SSL/TLS on top of the TCP stream.

Pre-shared key encryption is supported.
MQTT-SN

MQTT for Sensor Networks

A lighter MQTT for low bandwidth, high failure networks (Can use UDP/IP or plain Zigbee)

Security should be provided by the network (forget plain Internet!)
Paho for clients Java, C/C++, Python, Js, Go, Lua ...

Mosquitto feature full broker.

Ponte: Node.js server bridging MQTT, HTTP and CoAP.

Kura: M2M application framework with MQTT as default transport
CoAP: Constrained Application Protocol

Internet Eng. Task Force standard for Internet of Things.

Started in 2010!

Draft-18 is the final one.
Co: Constrained

Simple to encode: targets 8 bits MCU.

UDP based, targets low power IP networks.

Two level of QoS: confirmable message or not.

Simple observation mechanism.
CoAP: RESTful things!

REST paradigm for things:

URI: `coap://hostname/lamps/12/status`

HTTP like verbs:
- GET for reads
- POST, PUT, DELETE for mutation

But in a compact binary datagram.
## Co: Constrained

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

- Token (if any, TKL bytes) ...

- Options (if any) ...

- Payload (if any) ...

---
CoAP: Discoverability

GET coap://hostname/.well-known/core

Provides a list of all supported resources!
CoAP: Security

DTLS (TLS on UDP Datagrams)

Pre-shared key or not

DTLS is not really light :(
CoAP @ Eclipse IoT

Californium: Java CoAP server and client

Ponte
Device management

Definition:

Secure, monitor, manage fleet of deployed devices.
Device management

Configure the device.

Update the firmware (and maybe the app)

Monitor and gather connectivity statistics.
D.M. standards protocols

Usual suspects:

- TR-069
- OMA-DM
- Lightweight M2M

Goals: provide an application agnostic way to manage fleets of devices.
OMA-DM

An Open Mobile Alliance standard for Device Management.

Targets mobile phone terminals but can be used for M2M devices.

Mean to be used by mobile network operators.
OMA-DM features

Read, write configuration or monitoring nodes.

Trigger remote commands (Exec)

FUMO:
Firmware Update Management Object

SCOMO:
Software Component Management Object
OMA-DM

HTTP/XML based, with a binary XML (WBXML) encoding.

Weird phone features gets in the way: every communication the device gives its language (ex: EN_en)

Binary SMS for wakeup and bootstraping.
OMA-DM security

HMAC MD5: HTTP Header signing the payload

Use HTTPS if you need confidentiality
Lightweight M2M

A new Open Mobile Alliance standard

An OMA-DM successor for M2M targets
Lightweight M2M: CoAP

Built on top of CoAP:

Really lighter than OMA-DM or TRS-069.
LWM2M features

Firmware upgrades (in band or thru http)

Device monitoring and configuration

Server provisioning (bootstrapping)
SMS can be used for waking-up the device.

Or for any GET/POST/PUT!
LWM2M: standard objects

Device
Server
Connectivity monitoring
Connectivity statistics
Location
Firmware

The objects have a numerical identifier.
LWM2M the URLs

URLs:

/{object}/{instance}/{resource}

Ex:

/6/0      = whole position object (binary TLV)
/6/0/2    = only the altitude value
LWM2M @ Eclipse IoT

Wakaama (ex liblwm2m):

A C library for implementing LWM2M in your devices.
More LWM2M open-source

Leshan:
A Java based LWM2M server.

Based on Californium (Eclipse).

http://github.com/jvermillard/leshan
Architecture of your IoT application

Embedded Application

Cloud servers
Architecture of your IoT application

Embedded Application

Cloud servers

MQTT or CoAP for fun and profits!
A more realistic architecture

Cloud servers

OMA-DM

MQTT or CoAP

O/S

Supervisor

Application
A more realistic architecture

- Linux O/S
- Supervisor
- App1
- App2
- App3
- Low power App
- Ad-hoc
- Radio module
  - 2G/3G/LTE
  - ZigBee/6LowPan
  - WiFi, etc..
- Cloud servers
  - OMA-DM
- MQTT/CoAP
- Network Operator
  - OMA-DM
Protocol hell

Every processor and every application need to be configured, upgraded and monitored.

Device management is not an option!
Each protocol must be secured.

And synchronized:
You can’t trigger an update with a protocol, while you are rebooting the device using another.
M2M/IoT is not a simple problem.

Security and provisioning are really the hardest ones.

Try hard to reduce the number of protocols to make your life easier!
CoAP with LWM2M can provide a light device management and application protocol to rule them all!

But CoAP is still a newcomer in the field and not a one size fits all solution.

Let’s specify device management on top of MQTT!
Thanks!

Twitter: @vrmvrm

E-mail: jvermillard@sierrawireless.com

Creative Commons – Attribution (CC BY 3.0)
Microchip designed by Nicolò Bertoncin from the Noun Project
Cloud designed by James Fenton from the Noun Project
Secure by Charlene Chen from The Noun Project
Chat by Icomatic from The Noun Project
Microchip designed by Mario Verduzco from the Noun Project
Certificate designed by Charlene Chen from the Noun Project
Evaluate This Session

1. Sign-in: [www.eclipsecon.org](http://www.eclipsecon.org)
2. Select session from schedule
3. Evaluate: +1 0 -1