

CoAP an introduction

Bifrost workshop

2012-11-28

Robert Olsson KTH/UU

CoAP/overview

- Internet-Of-Things (IoT)
- CoAP (Constraint Application Protocol)
- IETF standard (Draft)

CoAP/overview

- Machine-to-Machine (M2M)
- Small footprint, RAM, ROM
- URI (Uniform Resource Identifier)
mapping considered

CoAP/overview

- RESTful client server compare http:
- Resource Discovery
- UDP
 - Reliable unicast
 - Best effort multicast
- Proxy and Caching is simple

CoAP/message type

- Confirmable message
- Non-confirmable message
- Ack message
- Reset message
 - Piggy-backed
 - Seperate

CoAP/transport

- Default UDP but required DTLS (Datagram TLS)
- TCP
- SCTP

CoAP/protocol

- 4 byte header
- Options
- Payload
 - uint (unsigned integer)
 - string
 - opaque
- Endpoint
 - IP addr, UDP port

CoAP/protocol header

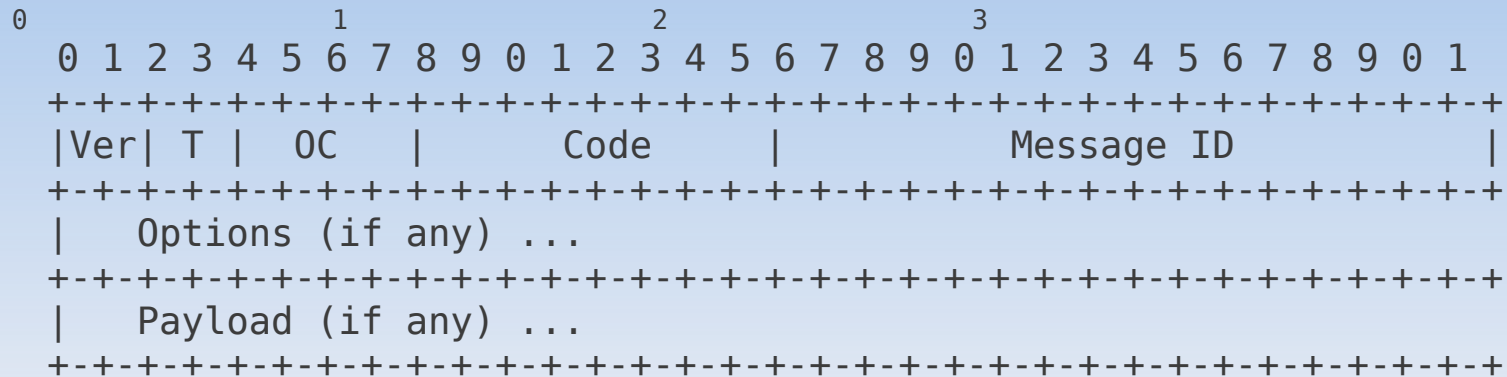


Figure 7: Message Format

3.1. Header Format

The fields in the header are defined as follows:

Version (Ver): 2-bit unsigned integer. Indicates the CoAP version number. Implementations of this specification MUST set this field to 1. Other values are reserved for future versions.

Type (T): 2-bit unsigned integer. Indicates if this message is of type Confirmable (0), Non-Confirmable (1), Acknowledgement (2) or Reset (3). See Section 4 for the semantics of these message types.

Option Count (OC): 4-bit unsigned integer. Indicates the number of options after the header (0-14). If set to 0, there are no options and the payload (if any) immediately follows the header. If set to 15, then an end-of-options marker is used to indicate the end of options and the start of the payload. The format of options is defined below.

CoAP/pkt-size

- Message size
 - Must fit in a single IP datagram
 - Default MTU 1280 bytes
 - 6LOWPAN 127 bytes
 - WSN based on IEEE 802.15.4 127 bytes

CoAP/RESTful

- CoAP Request/Response semantics
 - GET, POST, PUT, DELETE
 - Easy to map to HTTP
- Cache and Proxy possible
- Token to match. Request/Response pairs

CoAP/URI

coap URI

`coap://example.se:5683/~sensors./temp1.xml`

coaps URI

`coaps://example.se:XXXX/~sensors./temp1.xml`

CoAP/Resource Discovery

Endpoint resource discovery (IP addr is known)
(Knowing or learning URI namespace)

- ct (content type) web linking RFC5988
- Multicast: ALL-COAP-NODES
 - Some tricks to read.

CoAP/Secure

- DTLS (Datagram TLS)
- IPSEC alternative
 - Key sharing problems
 - Resource problems
 - Certificate problems

CoAP/HTTP mapping

- Request/Response model is mapped HTTP
 - Not messages, non-confirmable
 - Proxy CoAP → HTTP
 - Proxy HTTP → CoAP

 - Proxy is man-in-the middle
 - Security issues
 - Key sharing problems
- Caching. Consider.

CoAP/implementations

- Contiki-2.6
 - ETH Zurich
 - 8.5 kB ROM
 - 1.5 kB RAM
- Linux → libcoap
- TinyOs (libcoap)
- Firefox CoAP plugin

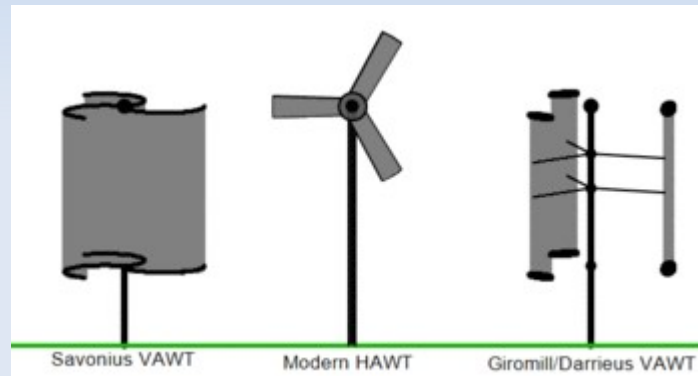
CoAP/additional standards

- Blockwise transfers in CoAP (16-1024 bytes)
draft-ietf-core-block-10
- CoRE Link Format (GET /.well known/core)
draft-shelby-core-link-format-14
- Observing Resources in CoAP (Observe option)
draft-ietf-core-observe-07

References

- The Contiki OS. <http://www.contiki-os.org/>
- draft-ietf-core-coap-12 <https://datatracker.ietf.org/doc/draft-ietf-core-coap/>
- draft-ietf-core-block-10 <https://datatracker.ietf.org/doc/draft-ietf-core-block/>
- draft-ietf-core-observe-07 <https://datatracker.ietf.org/doc/draft-ietf-core-observe/>
- draft-ietf-core-link-format-14 <https://datatracker.ietf.org/doc/draft-ietf-core-link-format/>
- M. Kovatsch, S. Duquennoy, and A. Dunkels, A Low-Power CoAP for Contiki in Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on, 2011, pp. 855-860, DOI:10.1109/MASS.2011.100.
- IANA: RFCUniform Resource Identifier (URI) Schemes. [RFC4395]
- R. Olsson and J. Laas, Sensd. <http://github.com/herjulf/sensd>.

Questions



- CoAP/HTTP
- Questions