Internet of Things: 802.15.4, 6LoWPAN, RPL, COAP

Jürgen Schönwälder

Presented on October 14, 2010

http://cnds.eecs.jacobs-university.de/
IEEE 802.15.4

1. IEEE 802.15.4
   - Radio Characteristics and Topologies
   - Frame Formats, Media Access Control, Security

2. IPv6 over IEEE 802.15.4 (6LoWPAN)
   - Header Compression
   - Fragmentation and Reassembly

3. RPL: IPv6 Routing Protocol for LLNs
   - Instances, DODAGs, Versions, Ranks
   - DODAG Construction and RPL ICMPv6 Messages

4. Constrained Application Protocol (CoAP)
   - Transactions and Methods
   - Message Formats
IEEE 802.15.4

The IEEE standard 802.15.4 offers physical and media access control layers for low-cost, low-speed, low-power wireless personal area networks (WPANs).

Application Scenarios

- Home Networking
- Automotive Networks
- Industrial Networks
- Interactive Toys
- Remote Metering
- ...
<table>
<thead>
<tr>
<th>IEEE 802.15.4 Standard Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>802.15.4-2003</strong></td>
</tr>
<tr>
<td>Original version using Direct Sequence Spread Spectrum (DSSS) with data transfer rates of 20 and 40 kbit/s</td>
</tr>
<tr>
<td><strong>802.15.4-2006</strong></td>
</tr>
<tr>
<td>Revised version using Direct Sequence Spread Spectrum (DSSS) with higher data rates and adding Parallel Sequence Spread Spectrum (PSSS)</td>
</tr>
<tr>
<td><strong>802.15.4a-2007</strong></td>
</tr>
<tr>
<td>Adding Direct Sequence Ultra-wideband (UWB) and Chirp Spread Spectrum (CSS) physical layers to the 2006 version of the standard (ranging support)</td>
</tr>
</tbody>
</table>
### Frequencies and Data Rates

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Channels</th>
<th>Region</th>
<th>Data Rate</th>
<th>Baud Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>868-868.6 MHz</td>
<td>0</td>
<td>Europe</td>
<td>20 kbit/s</td>
<td>20 kBaud</td>
</tr>
<tr>
<td>902-928 MHz</td>
<td>1-10</td>
<td>USA</td>
<td>40 kbit/s</td>
<td>40 kBaud</td>
</tr>
<tr>
<td>2400-2483.5 MHz</td>
<td>11-26</td>
<td>global</td>
<td>250 kbit/s</td>
<td>62.5 kBaud</td>
</tr>
</tbody>
</table>
## IEEE 802.15.4 Device Classes

<table>
<thead>
<tr>
<th>Full Function Device (FFD)</th>
<th>Reduced Function Device (RFD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Any topology</td>
<td>- Reduced protocol set</td>
</tr>
<tr>
<td>- PAN coordinator capable</td>
<td>- Very simple implementation</td>
</tr>
<tr>
<td>- Talks to any other device</td>
<td>- Cannot become a PAN coordinator</td>
</tr>
<tr>
<td>- Implements complete protocol set</td>
<td>- Limited to leafs in more complex topologies</td>
</tr>
</tbody>
</table>
### Network Device

An RFD or FFD implementation containing an IEEE 802.15.4 medium access control and physical interface to the wireless medium.

### Coordinator

An FFD with network device functionality that provides coordination and other services to the network.

### PAN Coordinator

A coordinator that is the principal controller of the PAN. A network has exactly one PAN coordinator.
IEEE 802.15.4 Star Topology

Star Topology
- All nodes communicate via the central PAN coordinator
- Leafs may be any combination of FFD and RFD devices
- PAN coordinator is usually having a reliable power source
IEEE 802.15.4 Peer-to-Peer Topology

Nodes can communicate via the central PAN coordinator and via additional point-to-point links

Extension of the pure star topology
IEEE 802.15.4 Cluster Tree Topology

Leafs connect to a network of coordinators (FFDs)

One of the coordinators serves as the PAN coordinator

Clustered star topologies are an important case (e.g., each hotel room forms a star in a HVAC system)
IEEE 802.15.4 Frame Formats

General Frame Format

<table>
<thead>
<tr>
<th>octets: 2</th>
<th>1</th>
<th>0/2</th>
<th>0/2/8</th>
<th>0/2</th>
<th>0/2/8</th>
<th>variable</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame control</td>
<td>Sequence number</td>
<td>Destination PAN identifier</td>
<td>Destination address</td>
<td>Source PAN identifier</td>
<td>Source address</td>
<td>Frame payload</td>
<td>Frame sequence check</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bits: 0–2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7–9</th>
<th>10–11</th>
<th>12–13</th>
<th>14–15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame type</td>
<td>Security enabled</td>
<td>Frame pending</td>
<td>Ack. requested</td>
<td>Intra PAN</td>
<td>Reserved</td>
<td>Dst addr mode</td>
<td>Reserved</td>
<td>Src addr mode</td>
</tr>
</tbody>
</table>

- IEEE 64-bit extended addresses (globally unique)
- 16-bit “short” addresses (unique within a PAN)
- Optional 16-bit source / destination PAN identifiers
- max. frame size 127 octets; max. frame header 25 octets
IEEE 802.15.4 Frame Formats

**Beacon Frames**
- Broadcasted by the coordinator to organize the network

**Command Frames**
- Used for association, disassociation, data and beacon requests, conflict notification, ...

**Data Frames**
- Carrying user data — this is what we are interested in

**Acknowledgement Frames**
- Acknowledges successful data transmission (if requested)
IEEE 802.15.4 Media Access Control

Carrier Sense Multiple Access / Collision Avoidance

Basic idea of the CSMA/CA algorithm:

- First wait until the channel is idle.
- Once the channel is free, start sending the data frame after some random backoff interval.
- Receiver acknowledges the correct reception of a data frame.
- If the sender does not receive an acknowledgement, retry the data transmission.
IEEE 802.15.4 Unslotted Mode

**Node → PAN, Node → Node**
- The sender uses CSMA/CA and the receiver sends an ACK if requested by the sender.
- Receiver needs to listen continuously and can’t sleep.

**PAN → Node**
- The receiver polls the PAN whether data is available.
- The PAN sends an ACK followed by a data frame.
- Receiving node sends an ACK if requested by the sender.
- Coordinator needs to listen continuously and can’t sleep.
A superframe consists of three periods:

1. During the Contention-Access-Period (CAP), the channel can be accessed using normal CSMA/CA.
2. The Contention-Free-Period (CFP) has Guaranteed Time Slots (GTS) assigned by the PAN to each node.
3. During the Inactive-Period (IP), the channel is not used and all nodes including the coordinator can sleep.

The PAN delimits superframes using beacons.
## IEEE 802.15.4 Security

### Security Services

<table>
<thead>
<tr>
<th>Security Suite</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null</td>
<td>No security (default)</td>
</tr>
<tr>
<td>AES-CTR</td>
<td>Encryption only, CTR Mode</td>
</tr>
<tr>
<td>AES-CBC-MAC-128</td>
<td>128 bit MAC</td>
</tr>
<tr>
<td>AES-CBC-MAC-64</td>
<td>64 bit MAC</td>
</tr>
<tr>
<td>AES-CBC-MAC-32</td>
<td>32 bit MAC</td>
</tr>
<tr>
<td>AES-CCM-128</td>
<td>Encryption and 128 bit MAC</td>
</tr>
<tr>
<td>AES-CCM-64</td>
<td>Encryption and 64 bit MAC</td>
</tr>
<tr>
<td>AES-CCM-32</td>
<td>Encryption and 32 bit MAC</td>
</tr>
</tbody>
</table>

- Key management must be provided by higher layers
- Implementations must support AES-CCM-64 and Null
IEEE.
IEEE Std 802.15.4-2003.

IEEE.
IEEE Std 802.15.4-2006.

IEEE.
IEEE Std 802.15.4a-2007.

MAC Security and Security Overhead Analysis in the IEEE 802.15.4 Wireless Sensor Networks.

Home Networking with IEEE 802.15.4: A Developing Standard for Low-Rate Wireless Personal Area Networks.

L. D. Nardis and M.-G. Di Benedetto.
Overview of the IEEE 802.15.4/4a standards for low data rate Wireless Personal Data Networks.

S. Labella M. Petrova, J. Riihijarvi, P. Mahonen.
Performance Study of IEEE 802.15.4 Using Measurements and Simulations.
Z. Sahinoglu and S. Gezici.
Ranging in the IEEE 802.15.4a Standard.
1. IEEE 802.15.4
   - Radio Characteristics and Topologies
   - Frame Formats, Media Access Control, Security

2. IPv6 over IEEE 802.15.4 (6LoWPAN)
   - Header Compression
   - Fragmentation and Reassembly

3. RPL: IPv6 Routing Protocol for LLNs
   - Instances, DODAGs, Versions, Ranks
   - DODAG Construction and RPL ICMPv6 Messages

4. Constrained Application Protocol (CoAP)
   - Transactions and Methods
   - Message Formats
The pervasive nature of IP networks allows use of existing infrastructure.

IP-based technologies already exist, are well-known, and proven to be working.

Open and freely available specifications vs. closed proprietary solutions.

Tools for diagnostics, management, and commissioning of IP networks already exist.

IP-based devices can be connected readily to other IP-based networks, without the need for intermediate entities like translation gateways or proxies.
6LowPAN Challenge

Header Size Calculation...

- IPv6 header is 40 octets, UDP header is 8 octets
- 802.15.4 MAC header can be up to 25 octets (null security) or \(25 + 21 = 46\) octets (AES-CCM-128)
- With the 802.15.4 frame size of 127 octets, we have
  - \(127 - 25 - 40 - 8 = 54\) octets (null security)
  - \(127 - 46 - 40 - 8 = 33\) octets (AES-CCM-128)
- of space left for application data!

IPv6 MTU Requirements

- IPv6 requires that links support an MTU of 1280 octets
- Link-layer fragmentation / reassembly is needed
The 6LowPAN protocol is an adaptation layer allowing to transport IPv6 packets over 802.15.4 links
Uses 802.15.4 in unslotted CSMA/CA mode (strongly suggests beacons for link-layer device discovery)
Based on IEEE standard 802.15.4-2003
Fragmentation / reassembly of IPv6 packets
Compression of IPv6 and UDP/ICMP headers
Mesh routing support (mesh under)
Low processing / storage costs
All LoWPAN encapsulated datagrams are prefixed by an encapsulation header stack.

Each header in the stack starts with a header type field followed by zero or more header fields.

<table>
<thead>
<tr>
<th>Bit Pattern</th>
<th>Short Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 xxxxxxx</td>
<td>NALP</td>
<td>Not A LoWPAN Packet</td>
</tr>
<tr>
<td>01 000001</td>
<td>IPv6</td>
<td>uncompressed IPv6 addresses</td>
</tr>
<tr>
<td>01 000010</td>
<td>LOWPAN_HC1</td>
<td>HC1 Compressed IPv6 header</td>
</tr>
<tr>
<td>01 010000</td>
<td>LOWPAN_BC0</td>
<td>BC0 Broadcast header</td>
</tr>
<tr>
<td>01 111111</td>
<td>ESC</td>
<td>Additional Dispatch octet follows</td>
</tr>
<tr>
<td>10 xxxxxxx</td>
<td>MESH</td>
<td>Mesh routing header</td>
</tr>
<tr>
<td>11 000xxx</td>
<td>FRAG1</td>
<td>Fragmentation header (first)</td>
</tr>
<tr>
<td>11 100xxx</td>
<td>FRAGN</td>
<td>Fragmentation header (subsequent)</td>
</tr>
</tbody>
</table>
**Uncompressed IPv6/UDP (worst case scenario)**

<table>
<thead>
<tr>
<th>preamble</th>
<th>802.15.4 MAC header</th>
<th>DSP</th>
<th>uncompressed IPv6 header</th>
<th>UDP</th>
<th>payload</th>
<th>FCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max. 23 / 44</td>
<td>1</td>
<td>40</td>
<td>8</td>
<td>up to 54 / 33</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

- Dispatch code (01000001\textsubscript{2}) indicates no compression
- Up to 54 / 33 octets left for payload with a max. size MAC header with null / AES-CCM-128 security
- The relationship of header information to application payload is obviously really bad
Compressed Link-local IPv6/UDP (best case scenario)

- Dispatch code (01000010₂) indicates HC1 compression
- HC1 compression may indicate HC2 compression follows
- This shows the maximum compression achievable for link-local addresses (does not work for global addresses)
- Any non-compressible header fields are carried after the HC1 or HC1/HC2 tags (partial compression)
Compression Principles (RFC 4944)

- Omit any header fields that can be calculated from the context, send the remaining fields unmodified
- Nodes do not have to maintain compression state (stateless compression)
- Support (almost) arbitrary combinations of compressed / uncompressed header fields

Ongoing Work

- Compression for globally routable addresses (HC1G)
- Stateful compression (IPHC, NHC)
Fragmentation and Reassembly

Fragmentation Principles (RFC 4944)

- IPv6 packets too large to fit into a single 802.15.4 frame are fragmented.
- A first fragment carries a header that includes the datagram size (11 bits) and a datagram tag (16 bits).
- Subsequent fragments carry a header that includes the datagram size, the datagram tag, and the offset (8 bits).
- Time limit for reassembly is 60 seconds.

Ongoing Work

- Recovery protocol for lost fragments (RFC 4944 requires to resend the whole set of fragments)
## Fragmentation Example (compressed link-local IPv6/UDP)

<table>
<thead>
<tr>
<th>preamble</th>
<th>802.15.4 MAC header</th>
<th>FRAG1</th>
<th>DSP</th>
<th>HC1</th>
<th>HC2</th>
<th>JSC</th>
<th>UDP</th>
<th>IPv6</th>
<th>1 34</th>
<th>payload</th>
<th>FCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>max. 23 / 44</td>
<td></td>
<td>4</td>
<td>1 1</td>
<td>1 1</td>
<td>1 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>max. 127 octets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>preamble</th>
<th>802.15.4 MAC header</th>
<th>FRAGN</th>
<th>payload</th>
<th>FCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>max. 23 / 44</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>max. 127 octets</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Homework Question (consult RFC 4944 first)

- How many fragments are created for an 1280 octet IPv6 packet with no / maximum compression and none / AES-CCM-128 link-layer security?
- How many fragmented datagrams can be in transit concurrently for a 802.14.5 source / destination pair?
### 6LowPAN Implementations

<table>
<thead>
<tr>
<th>Name</th>
<th>OS / License</th>
<th>Hardware</th>
<th>Maintained</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacobs</td>
<td>TinyOS / 3BSD</td>
<td>Telos B, ...</td>
<td>no</td>
</tr>
<tr>
<td>Berkley IP</td>
<td>TinyOS / 3BSD</td>
<td>Telos B, ...</td>
<td>active</td>
</tr>
<tr>
<td>Arch Rock</td>
<td>TinyOS / EULA</td>
<td>Raven, ...</td>
<td>active</td>
</tr>
<tr>
<td>SICS Slowpan</td>
<td>Contiki / 3BSD</td>
<td>Raven, ...</td>
<td>active</td>
</tr>
<tr>
<td>Sensinode</td>
<td>Own / EULA</td>
<td>Sensinode</td>
<td>active</td>
</tr>
<tr>
<td>Hitachi</td>
<td>Own / EULA</td>
<td>Renesas</td>
<td>unknown</td>
</tr>
</tbody>
</table>

Unfortunately...  

- The Jacobs implementation uses the TinyOS Active Message framing format and thus does not interoperate
### Feature Comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>Jacobs</th>
<th>Berkley</th>
<th>Contiki</th>
<th>Arch Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispatch Header</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Dispatch Type</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Mesh Header</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Mesh Routing</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>+</td>
</tr>
<tr>
<td>Multicasting Header</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Multicasting</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Fragmentation</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>HC1</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>HC2 for UDP</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>HC1g</td>
<td>-</td>
<td>-</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>ICMPv6 Echo</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

+ = supported and tested, o = supported but not tested, - = not supported, * = see [?] for details
Implementation via USB Serial Interfaces
K.D. Korte, I. Tumar, and J. Schönwälder.
Evaluation of 6lowpan Implementations.

IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals.

G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler.
Transmission of IPv6 Packets over IEEE 802.15.4 Networks.

M. Harvan and J. Schönwälder.
TinyOS Motes on the Internet: IPv6 over 802.15.4 (6lowpan).
RPL: IPv6 Routing Protocol for LLNs

1. IEEE 802.15.4
   - Radio Characteristics and Topologies
   - Frame Formats, Media Access Control, Security

2. IPv6 over IEEE 802.15.4 (6LoWPAN)
   - Header Compression
   - Fragmentation and Reassembly

3. RPL: IPv6 Routing Protocol for LLNs
   - Instances, DODAGs, Versions, Ranks
   - DODAG Construction and RPL ICMPv6 Messages

4. Constrained Application Protocol (CoAP)
   - Transactions and Methods
   - Message Formats
## Motivation and Requirements

### Routing Requirements

- Urban LLNs [RFC5548]
- Industrial LLNs [RFC5673]
- Home Automation LLNs [RFC5826]
- Building Automation LLNs [RFC5867]

### Common Characteristics

- Low power and Lossy Networks (LLNs) consisting largely of constrained nodes.
- Lossy and unstable links, typically supporting low data rates, relatively low packet delivery rates.
- Traffic patterns are not simply point-to-point, but in many cases point-to-multipoint or multipoint-to-point.
- Potentially comprising up to thousands of nodes.
Definition

An RPL Instance consists of multiple Destination Oriented Directed Acyclic Graphs (DODAGs). Traffic moves either up towards the DODAG root or down towards the DODAG leafs.
DODAG and RPL Instance Properties

**DODAG Properties**
- Many-to-one communication: upwards
- One-to-many communication: downwards
- Point-to-point communication: upwards-downwards

**RPL Instance Properties**
- DODAGS are disjoint (no shared nodes)
- Link properties: (reliability, latency, ...)
- Node properties: (powered or not, ...)
- RPL Instance has an optimization objective
- Multiple RPL Instances with different optimization objectives can coexist
Definition

A node’s Rank defines the node’s individual position relative to other nodes with respect to a DODAG root. The scope of Rank is a DODAG Version.
Route Construction and Forwarding Rules

**Route Construction**
- Up routes towards nodes of decreasing rank (parents)
- Down routes towards nodes of increasing rank
  - Nodes inform parents of their presence and reachability to descendants
  - Source route for nodes that cannot maintain down routes

**Forwarding Rules**
- All routes go upwards and/or downwards along a DODAG
- When going up, always forward to lower rank when possible, may forward to sibling if no lower rank exists
- When going down, forward based on down routes
RPL Control Messages

**DAG Information Object (DIO)**
- A DIO carries information that allows a node to discover an RPL Instance, learn its configuration parameters and select DODAG parents

**DAG Information Solicitation (DIS)**
- A DIS solicits a DODAG Information Object from an RPL node

**Destination Advertisement Object (DAO)**
- A DAO propagates destination information upwards along the DODAG
DODAG Construction

Construction

- Nodes periodically send link-local multicast DIO messages.
- Stability or detection of routing inconsistencies influence the rate of DIO messages.
- Nodes listen for DIOs and use their information to join a new DODAG, or to maintain an existing DODAG.
- Nodes may use a DIS message to solicit a DIO.
- Based on information in the DIOs the node chooses parents that minimize path cost to the DODAG root.

Comment

- Essentially a distance vector routing protocol with ranks to prevent count-to-infinity problems.
RPL: IPv6 Routing Protocol for Low power and Lossy Networks.
Internet-Draft (work in progress) <draft-ietf-roll-rpl-12>, Cisco Systems, Sigma Designs, LIX, Arch Rock Corporation, Ember Corporation, Stanford University, Dust Networks, October 2010.

Routing Requirements for Urban Low-Power and Lossy Networks.

Industrial Routing Requirements in Low-Power and Lossy Networks.
RFC 5673, Dust Networks, Cisco Systems, Shell, October 2009.

A. Brandt, J. Buron, and G. Porcu.
Home Automation Routing Requirements in Low-Power and Lossy Networks.
RFC 5826, Sigma Designs, Telecom Italia, April 2010.

J. Martocci, P. De Mil, N. Riou, and W. Vermeylen.
Building Automation Routing Requirements in Low-Power and Lossy Networks.
RFC 5867, Johnson Controls Inc, Ghent University, Schneider Electric, Arts Centre Vooruit, June 2010.
Constrained Application Protocol (CoAP)

1. IEEE 802.15.4
   - Radio Characteristics and Topologies
   - Frame Formats, Media Access Control, Security

2. IPv6 over IEEE 802.15.4 (6LoWPAN)
   - Header Compression
   - Fragmentation and Reassembly

3. RPL: IPv6 Routing Protocol for LLNs
   - Instances, DODAGs, Versions, Ranks
   - DODAG Construction and RPL ICMPv6 Messages

4. Constrained Application Protocol (CoAP)
   - Transactions and Methods
   - Message Formats
CoAP Overview

Characteristics

- Constrained machine-to-machine web protocol
- Representational State Transfer (REST) architecture
- Simple proxy and caching capabilities
- Asynchronous transaction support
- Low header overhead and parsing complexity
- URI and content-type support
- UDP binding (may use IPsec or DTLS)
- Reliable unicast and best-effort multicast support
- Built-in resource discovery
CoAP transactions provide reliable UDP messaging

CoAP methods resemble HTTP method requests and responses

CoAP method calls may involve multiple CoAP transactions

Roles at the transaction layer may change during a method request / response execution
CoAP Transactions

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON</td>
<td>Confirmable requests that the receiving peer sends an acknowledgement or a reset</td>
</tr>
<tr>
<td>NON</td>
<td>Non-confirmable messages do not request any message being sent by the receiving peer</td>
</tr>
<tr>
<td>ACK</td>
<td>Acknowledges that a CON has been received, may carry payload</td>
</tr>
<tr>
<td>RST</td>
<td>Indicates that a CON has been received but some context is missing to process it</td>
</tr>
</tbody>
</table>

- Transactions are invoked peer to peer (not client/server)
- Transactions are identified by a Transaction ID (TID)
CoAP Methods

### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>Retrieves information of an identified resource</td>
</tr>
<tr>
<td>POST</td>
<td>Creates a new resource under the requested URI</td>
</tr>
<tr>
<td>PUT</td>
<td>Updates the resource identified by an URI</td>
</tr>
<tr>
<td>DELETE</td>
<td>Deletes the resource identified by an URI</td>
</tr>
</tbody>
</table>

- Resources are identified by URIs
- Methods are very similar to HTTP methods
- Response codes are a subset of HTTP response codes
- Options carry additional information (similar to HTTP header lines, but using a more compact encoding)
CoAP Message Exchanges

Examples

- Synchronous transaction (left)
- Asynchronous transaction (middle)
- Orphaned transaction (right)
CoAP Message Format

The Ver field contains the version number, the T field the message type, and the OC field the number of options.

The Code field carries the method code / response code (methods are numbers not strings).

The unique Transaction ID is changed for every new request but not during retransmissions.
The option delta identifies the option type, encoded as the delta (difference) to the previous option code.

The option code implies the type of the encoded data.

URI parameters are carried in options.

The content-type defaults to text/plain.
Z. Shelby, B. Frank, and D. Sturek.  
Constrained Application Protocol (CoAP).  
Description

- ATmega1284PV: 8 bit, 20 MHz, 16K RAM, 128K Flash
- Contiki 2.4 (6LoWPAN, UDP, TCP, HTTP, ...)

Demo: ATMEL Raven / Contiki