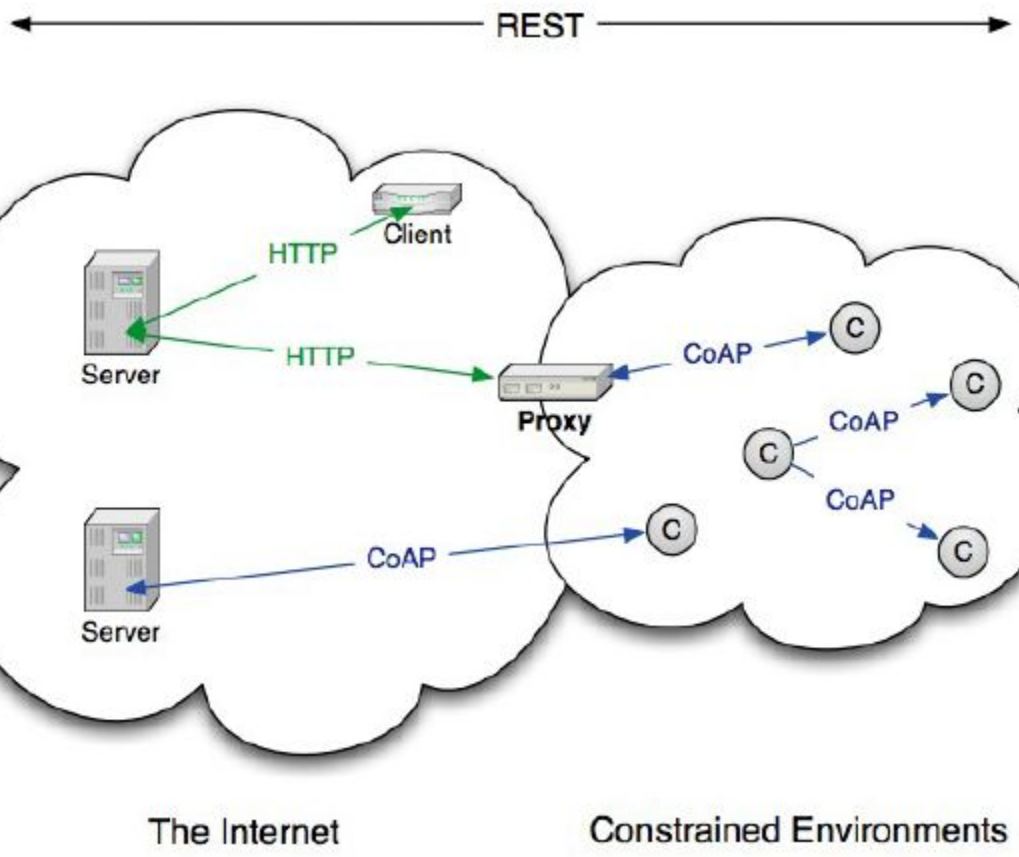
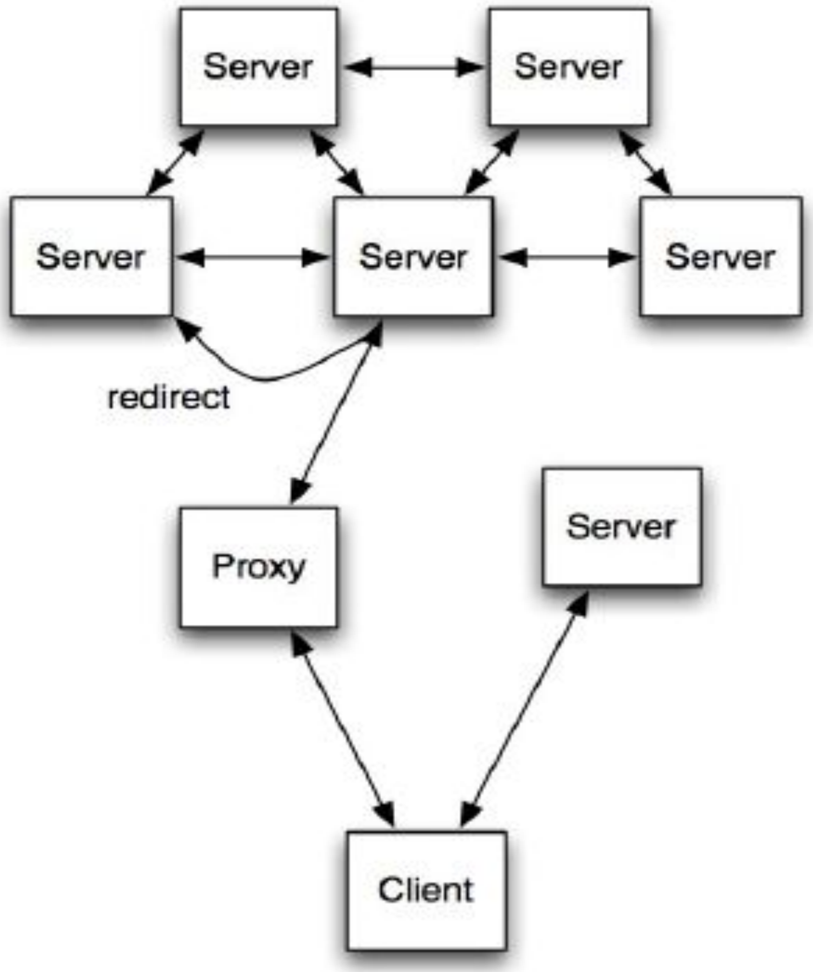
The Constrained Application Protocol (CoAP)

- The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks.
- The protocol is designed for machine- to-machine (M2M) applications such as smart energy and building automation.

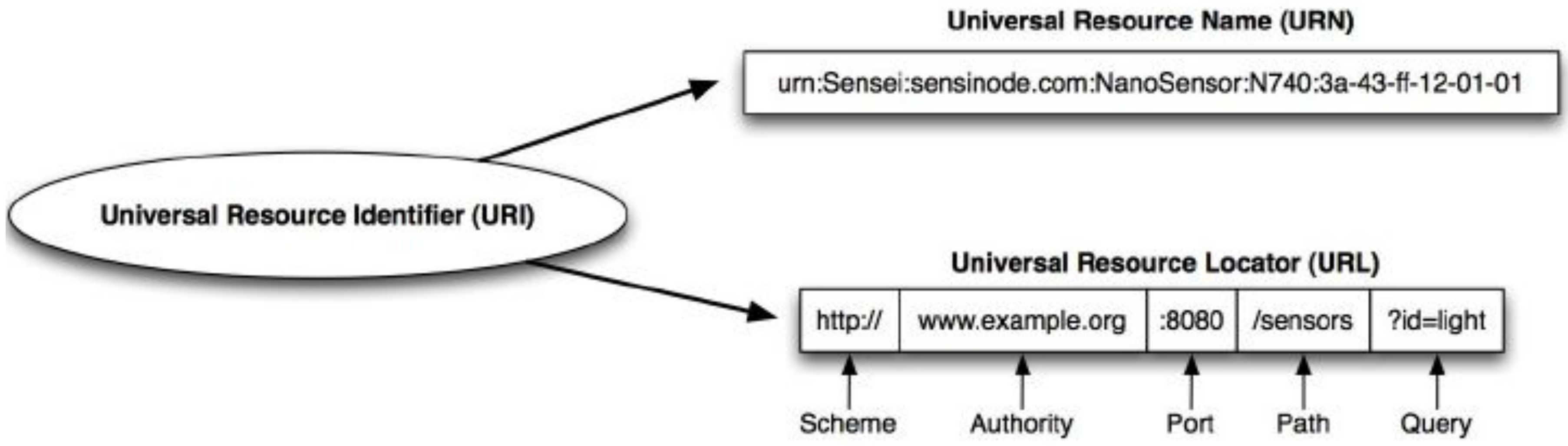
CoAP Architecture



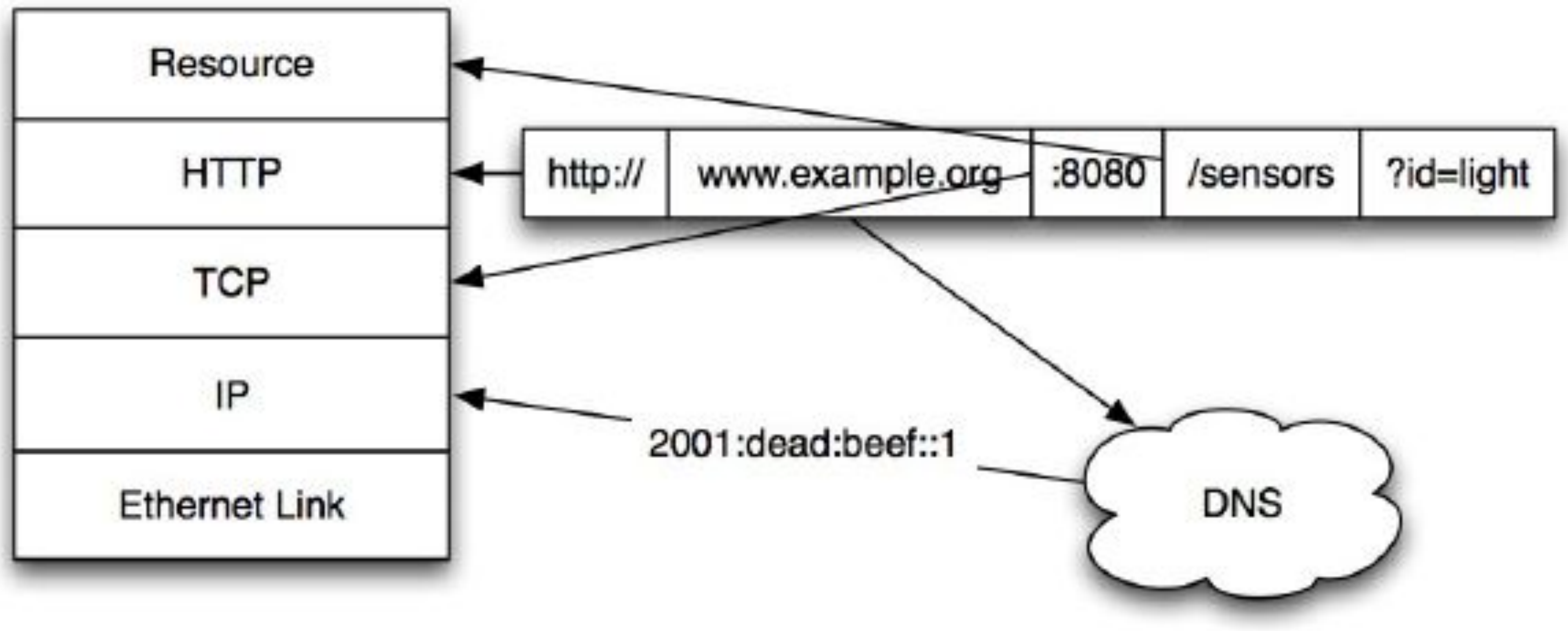
The Web Architecture



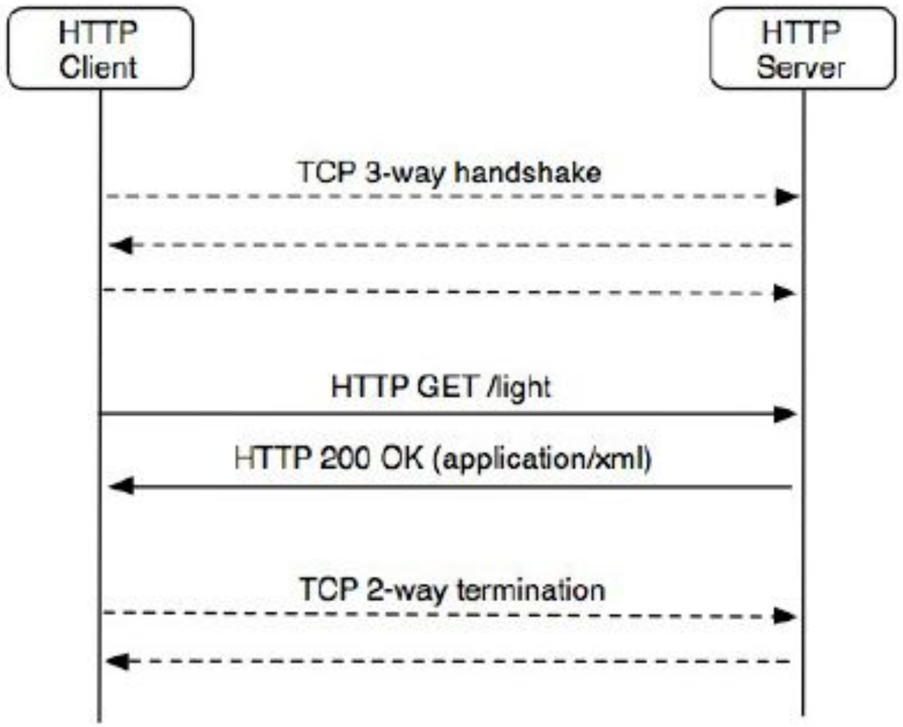
Web Naming


Universal Resource Name (URN)
urn:Sensei:sensinode.com:NanoSensor:N740:3a-43-ff-12-01-01
Universal Resource Identifier (URI)
Universal Resource Locator (URL)
http://
www.example.org
:8080
/sensors
?id=light
Scheme
Authority
Port
Path
Query

URL Resolution



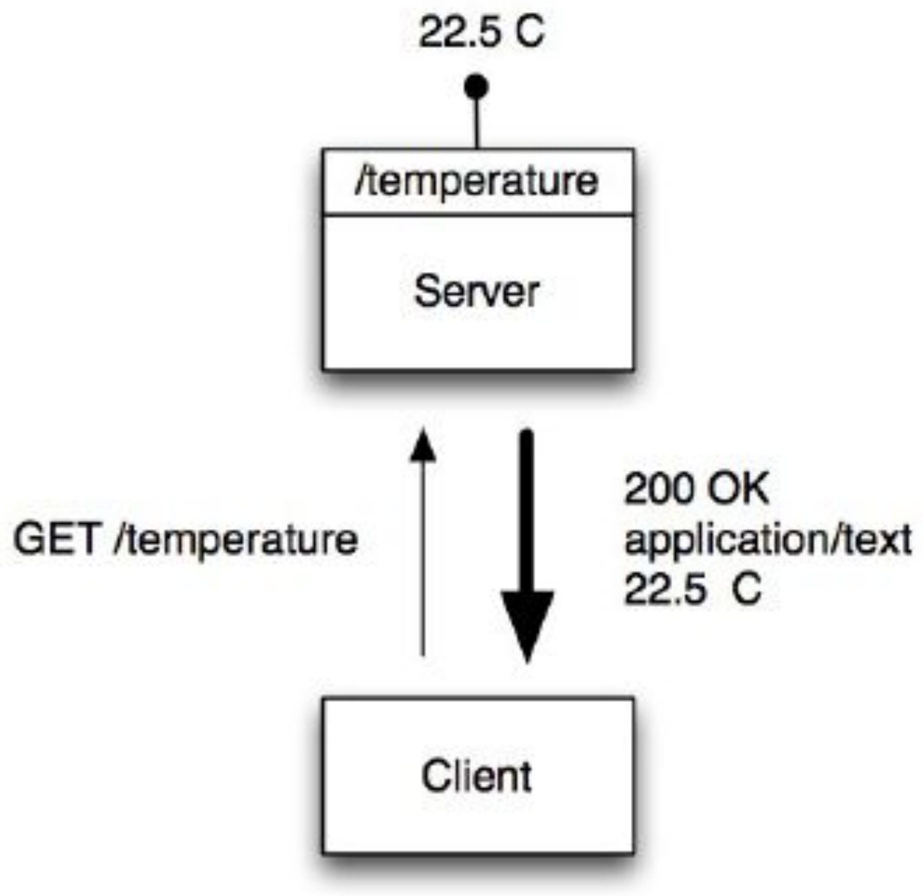
HTTP Request



HTTP

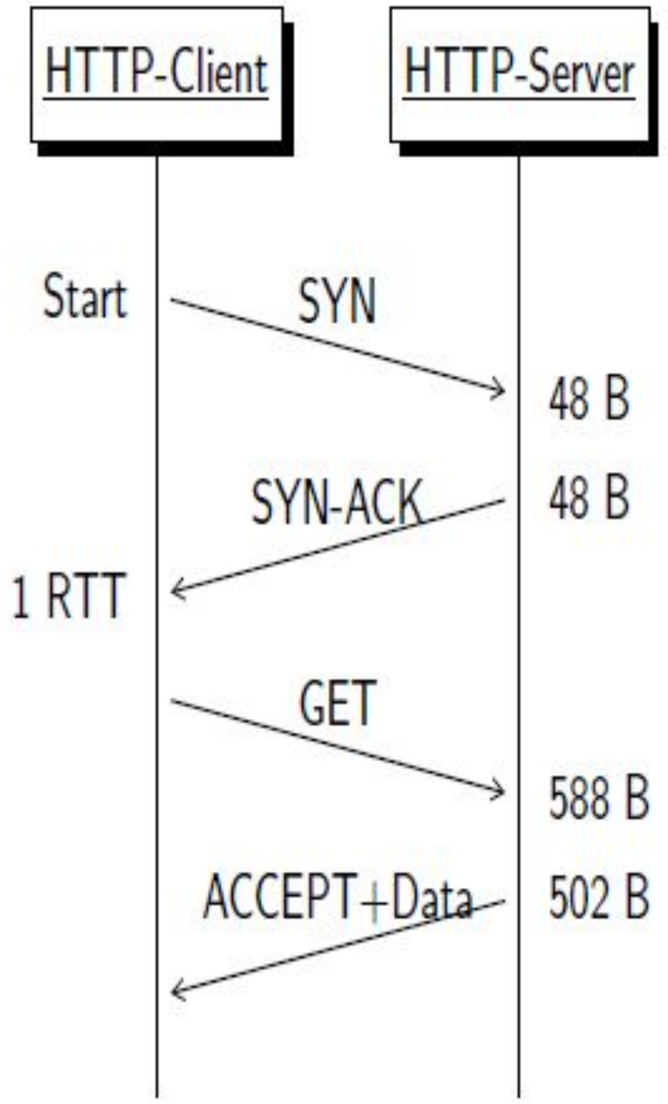
- De-facto standard of information transfer in the Internet
- TCP is the transport protocol for HTTP
- Problems of HTTP in IoT environment
  - **Lengthy HTTP headers**, For smaller packets, the protocol header overhead is more
  - The need to establish TCP sessions for each request-reply data transfer. For short data transfers, TCP connection establishment time will be up to 50% of the total transfer time

REST REQUEST



HTTP: Downloading a 10-byte object - phases of data transfer

- Legacy browser is used
- TCP connection establishment
- GET request
  - Get ./10 HTTP/1.1
  - Details of the browser, OS
  - Accepted data types, encoding and encryption schemes
- ACCEPT + Data
  - HTTP 1.1/ 200 OK
  - Date and time
  - Server details, name, last updated
- Total bytes transferred 1546 bytes including TCP ACKs



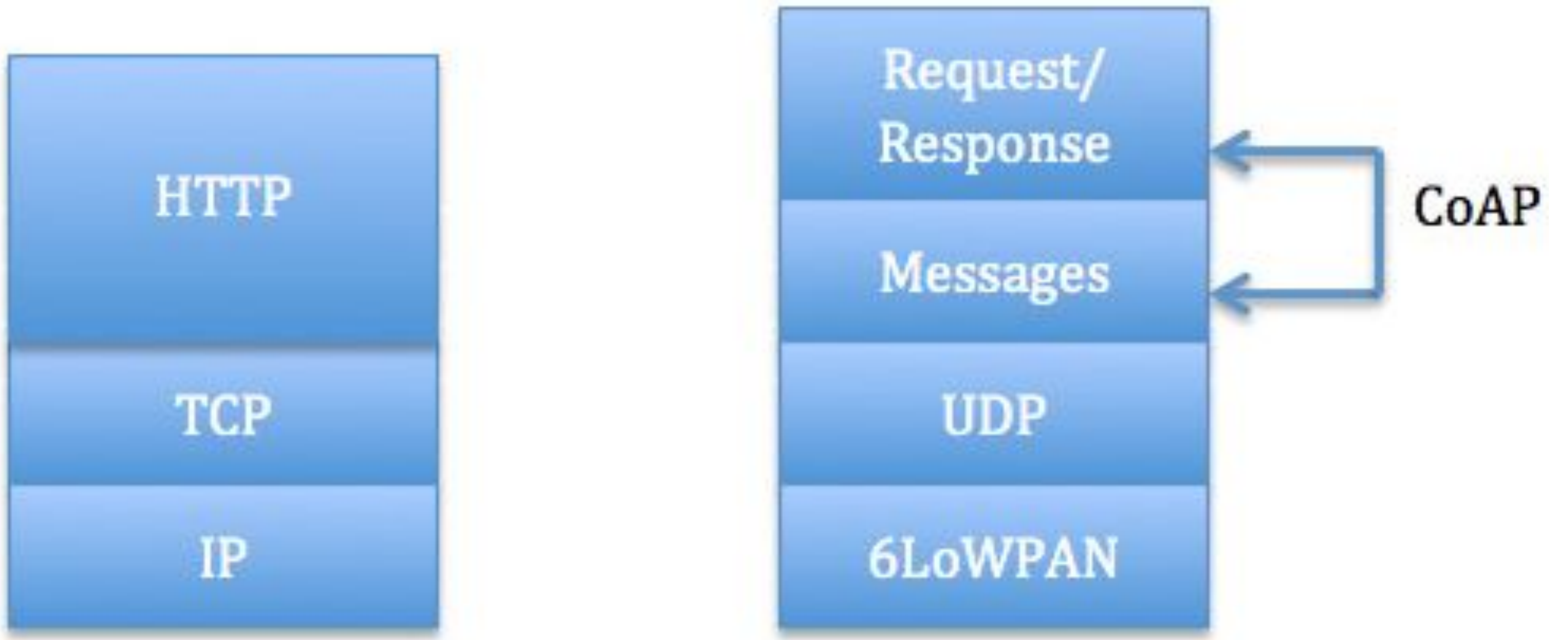
CoAP – Constrained Application Protocol

- It is a specialized web transfer protocol
- CoAP would become the standard protocol to enable interaction between device and to support IoT applications
- **CoAP operates over UDP**
- CoAP needs t**o consider optimized length of datagram and provide reliable communication**
- It supports REST protocol to support URI (Uniform resource identifier)
  - REST architecture – which is a general design for accessing Internet resources

CoAP – Constrained Application Protocol

- CoAP provides URI, REST methods such as GET,POST,PUT and DELETE
- CoAP redesigned some features of HTTP to accommodate these limitations

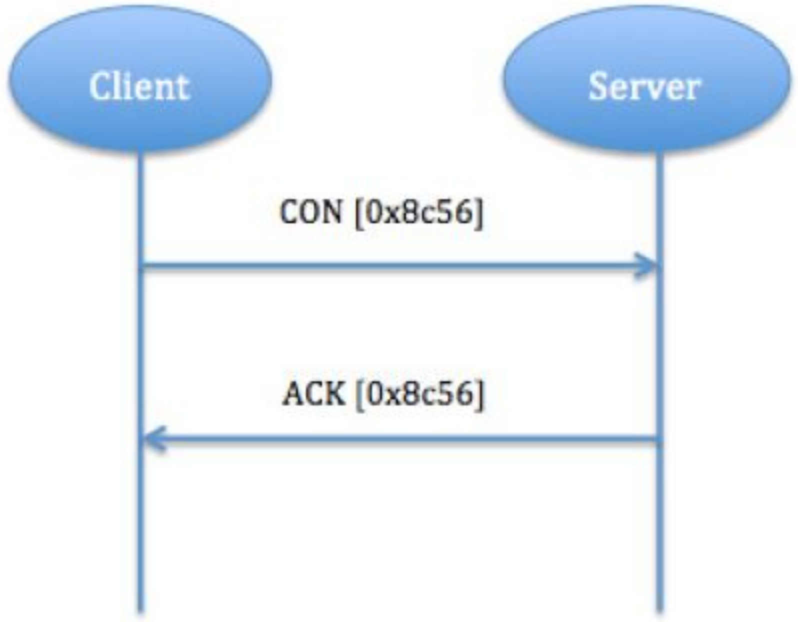
HTTP and CoAP Protocol Stack



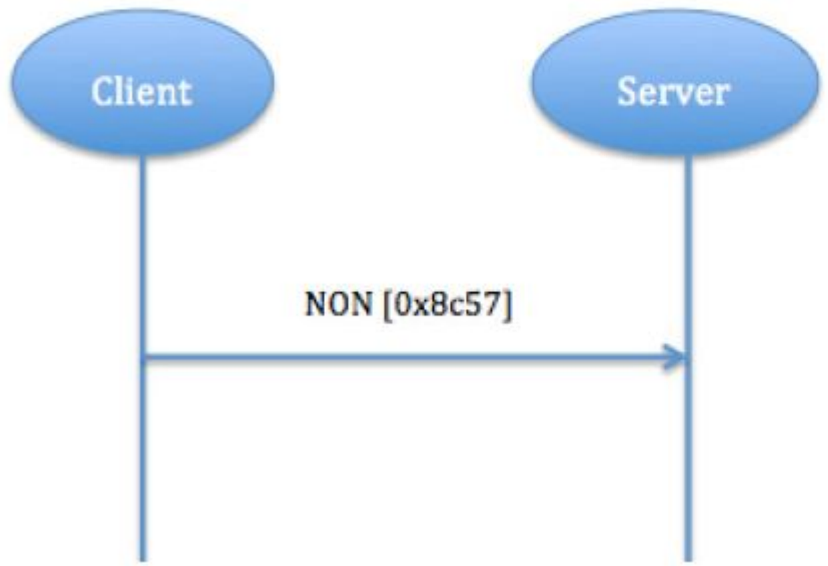
Message Layer Model

- It supports 4 types message
  - CON (Confirmable)
  - NON (non-confirmable)
  - ACK (Acknowledgement)
  - RST (Reset)
- It designed to deal with UDP and asynchronous switching

Reliable Message Transport



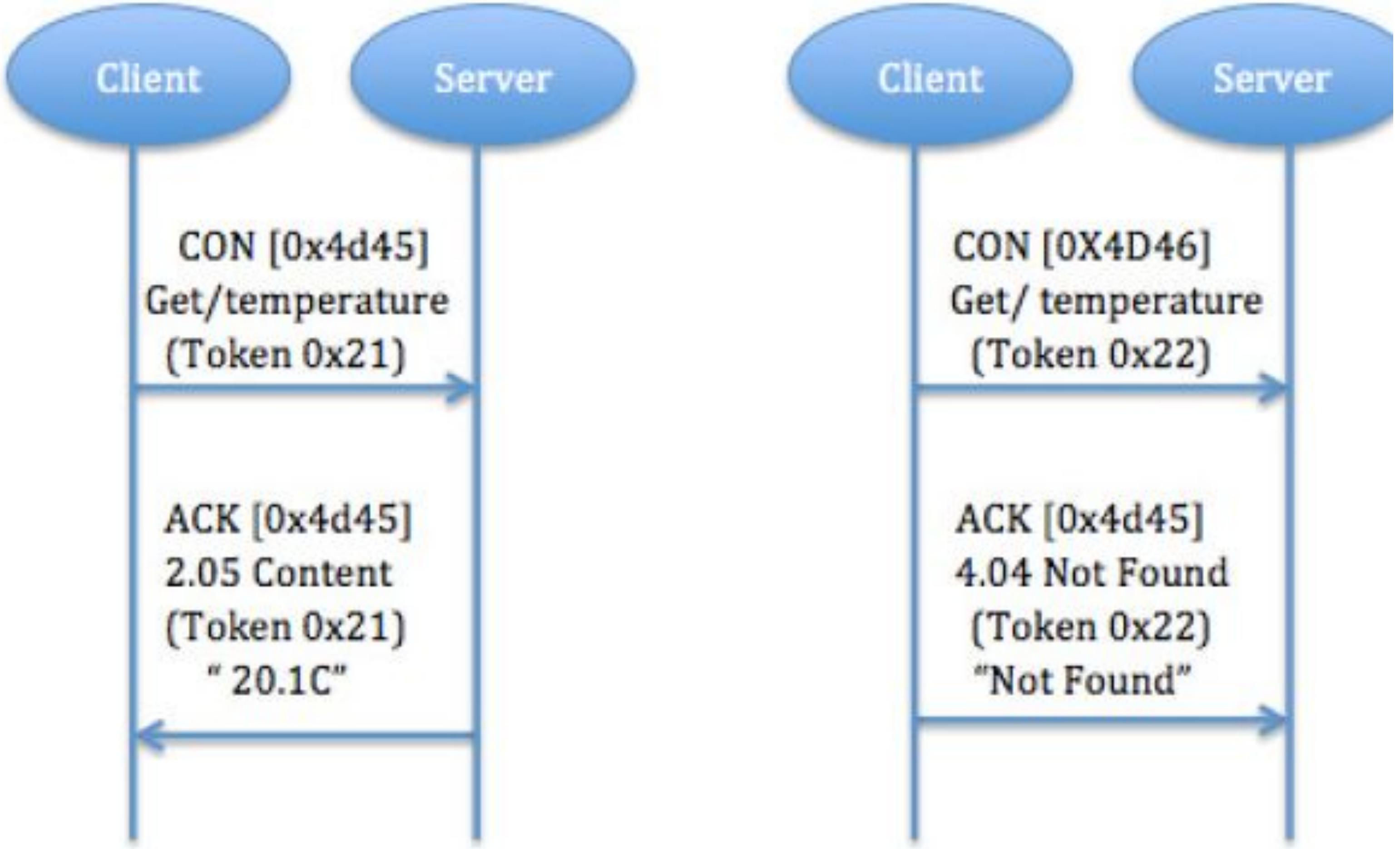
Unreliable message transport



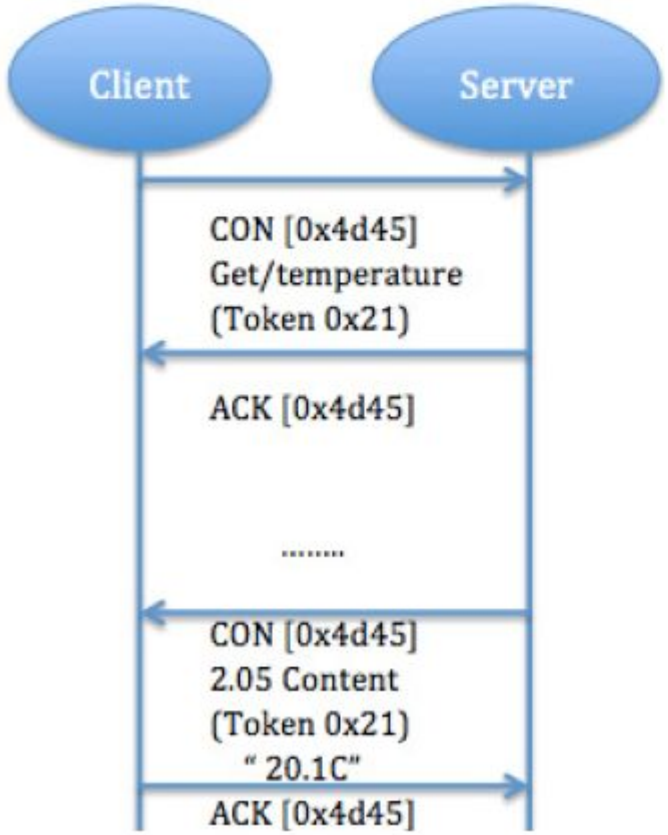
Request/Response Layer

- It concerns communication method and deal with request/response message

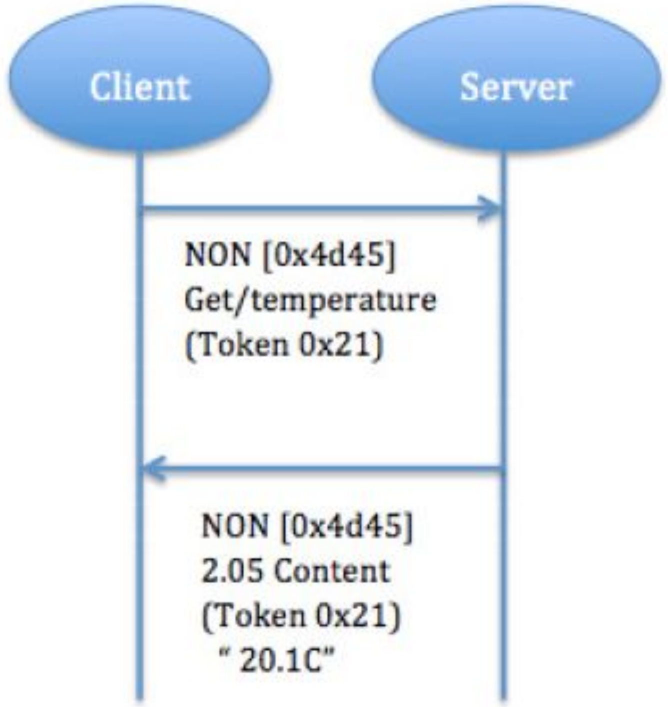
piggy backed message



Separate Response



Non confirmable Request and Response



The packet format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver| T |  TKL  |      Code     |          Message ID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Token (if any, TKL bytes) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 1 1 1 1 1|    Payload (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

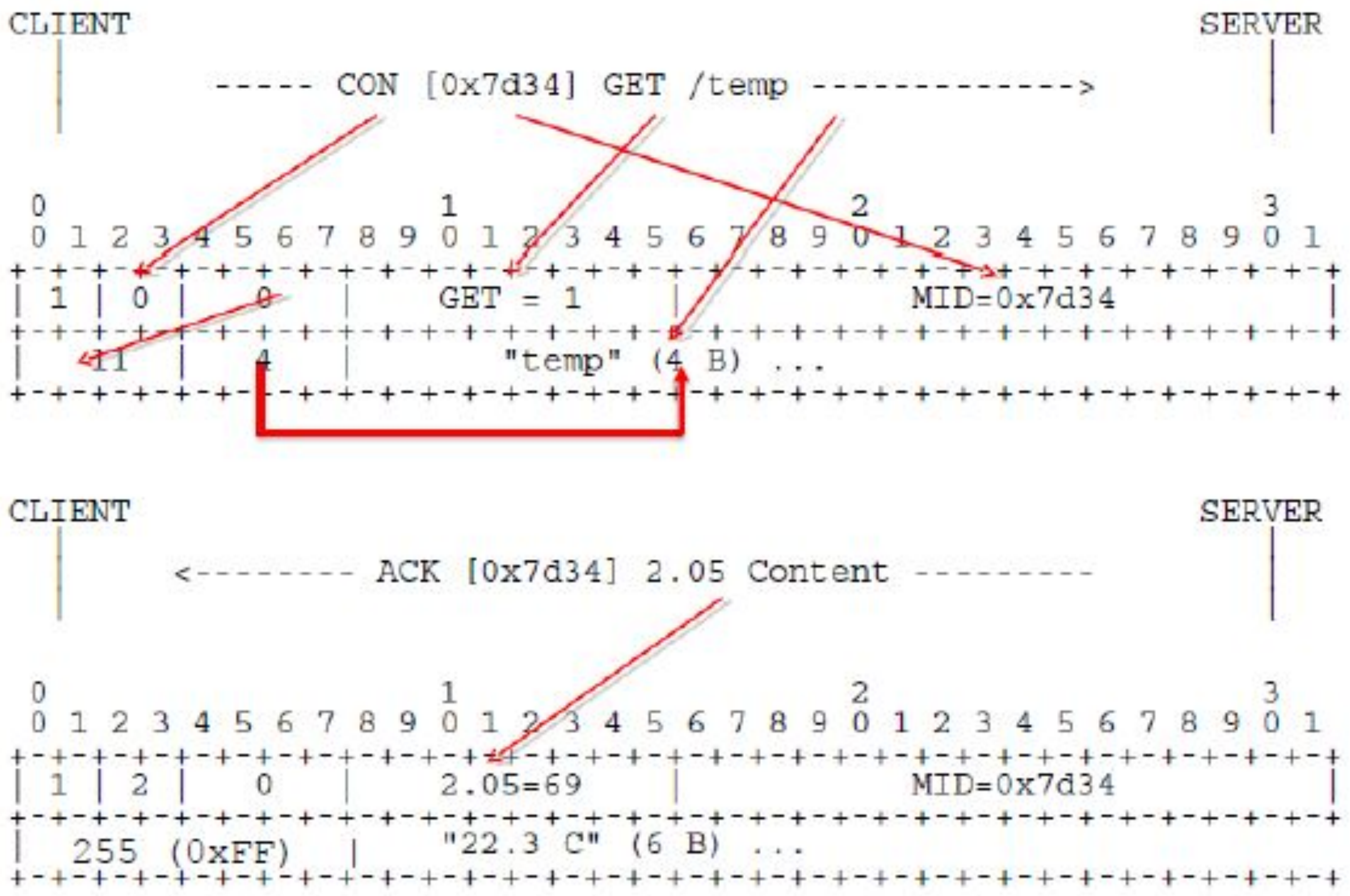
- Ver(sion) – 1
- T(ype) – CON / ACK / NON / RST
- T(o)K(en)L(ength)
- (Request or Response) Code
- 0xFF: payload marker

Packet Format

- Type (T): 2-bit unsigned integer. Indicates if this message is of type Confirmable (0), Non-confirmable (1), Acknowledgement (2), or Reset (3).
- Token Length (TKL): 4-bit unsigned integer. Indicates the length of the variable-length Token field (0-8 bytes)
- Code -class can indicate a request (0), a success response (2), a client error response (4), or a server error response (5)
- Message ID: 16-bit unsigned integer in network byte order. Used to detect message duplication and to match messages of type Acknowledgement/Reset to messages of type Confirmable/Non- confirmable.

- Token - The Token value is used to correlate requests and responses

Mapping



```
CLIENT                                                        SERVER
  |                                                             |
  |       ----- CON [0x7d34] GET /temp ------------->           |
  |                                                             |

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 1 | 0 |   0   |     GET = 1     |          MID=0x7d34           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  11   |   4   |     "temp" (4 B) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

CLIENT                                                        SERVER
  |                                                             |
  |     <-------- ACK [0x7d34] 2.05 Content ---------           |
  |                                                             |

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 1 | 2 |   0   |    2.05=69      |          MID=0x7d34           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  255 (0xFF)   |   "22.3 C" (6 B) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

(9 bytes)

CoAP - Messages

- Confirmable Message - Some messages require an acknowledgement. These messages are called "Confirmable". When no packets are lost, each Confirmable message elicits exactly one return message of type Acknowledgement or type Reset.
- Non-confirmable Message - Some other messages do not require an acknowledgement. This is particularly true for messages that are repeated regularly for application requirements, such as repeated readings from a sensor.
- Acknowledgement Message - An Acknowledgement message acknowledges that a specific Confirmable message arrived. By itself, an Acknowledgement message does not indicate success or failure of any request encapsulated in the Confirmable message, but the Acknowledgement message may also carry a Piggybacked Response

- Reset Message - A Reset message indicates that a specific message (Confirmable or Non-confirmable) was received, but some context is missing to properly process it. This condition is usually caused when the receiving node has rebooted and has forgotten some state that would be required to interpret the message. Provoking a Reset message (e.g., by sending an Empty Confirmable message) is also useful as an inexpensive check of the liveness of an endpoint ("CoAP ping").
- Piggybacked Response - A piggybacked Response is included right in a CoAP Acknowledgement (ACK) message that is sent to acknowledge receipt of the Request for this Response
- Separate Response - When a Confirmable message carrying a request is acknowledged with an Empty message (e.g., because the server doesn't have the answer right away), a Separate Response is sent in a separate message exchange
- Empty Message - A message with a Code of 0.00; neither a request nor a response. An Empty message only contains the 4-byte header.

CoAP Features

- Embedded web transfer protocol (coap://)
- Asynchronous transaction model
- UDP binding with reliability and multicast support
- GET, POST, PUT, DELETE methods
- URI support
- Small, simple 4 byte header
- DTLS based PSK, RPK and Certificate security
- Subset of MIME types and HTTP response codes
- Built-in discovery
- Optional observation and block transfer

What CoAP is (and is not)

- Sure, CoAP is
  - ✓ A very efficient RESTful protocol
  - ✓ Ideal for constrained devices and networks
  - ✓ Specialized for M2M applications
  - ✓ Easy to proxy to/from HTTP
- But hey, CoAP is not
  - ✓ A general replacement for HTTP
  - ✓ HTTP compression
  - ✓ Restricted to isolated “automation” networks