CONSTRAINED APPLICATION PROTOCOL (COAP)

Bhupendra Pratap Singh
R&D Engineer, IoT
ACTS, CDAC, Pune

OVERVEW

- CoAP is the product of the IETF RFC (7228) .
- The IETF Constrained Restful Environments (CoRE) working group created the first draft of the protocol in 2014 but had worked several years on its creation.
- It is specifically intended as a communication protocol for **constrained devices**

CONT. ….

- The core protocol is now based on **RFC 7252.**
- It also supports mapping to HTTP through the use of proxies.
- The HTTP mapping is on-board facility to get data across the Internet.
- It is excellent at providing a similar and easy structure of resource addressing familiar to anyone with experience of using Web but reduced resources and bandwidth demands.
- The protocol was designed for M2M needs initially, but was adapted in IoT as well, with support on gateways, high-end servers and enterprise integration.

HTTP VS COAP (SOME STATS)

- A study performed by Colitti et. Al demonstrated the efficiency of CoAP over standard HTTP.
- Outcome – CoAP Provides a similar functionality with significantly less overhead and power requirements.

	Bytes per-transaction	Power	Lifetime
CoAP	154	0.744 mW	151 days
HTTP	1451	1.333 mW	84 days

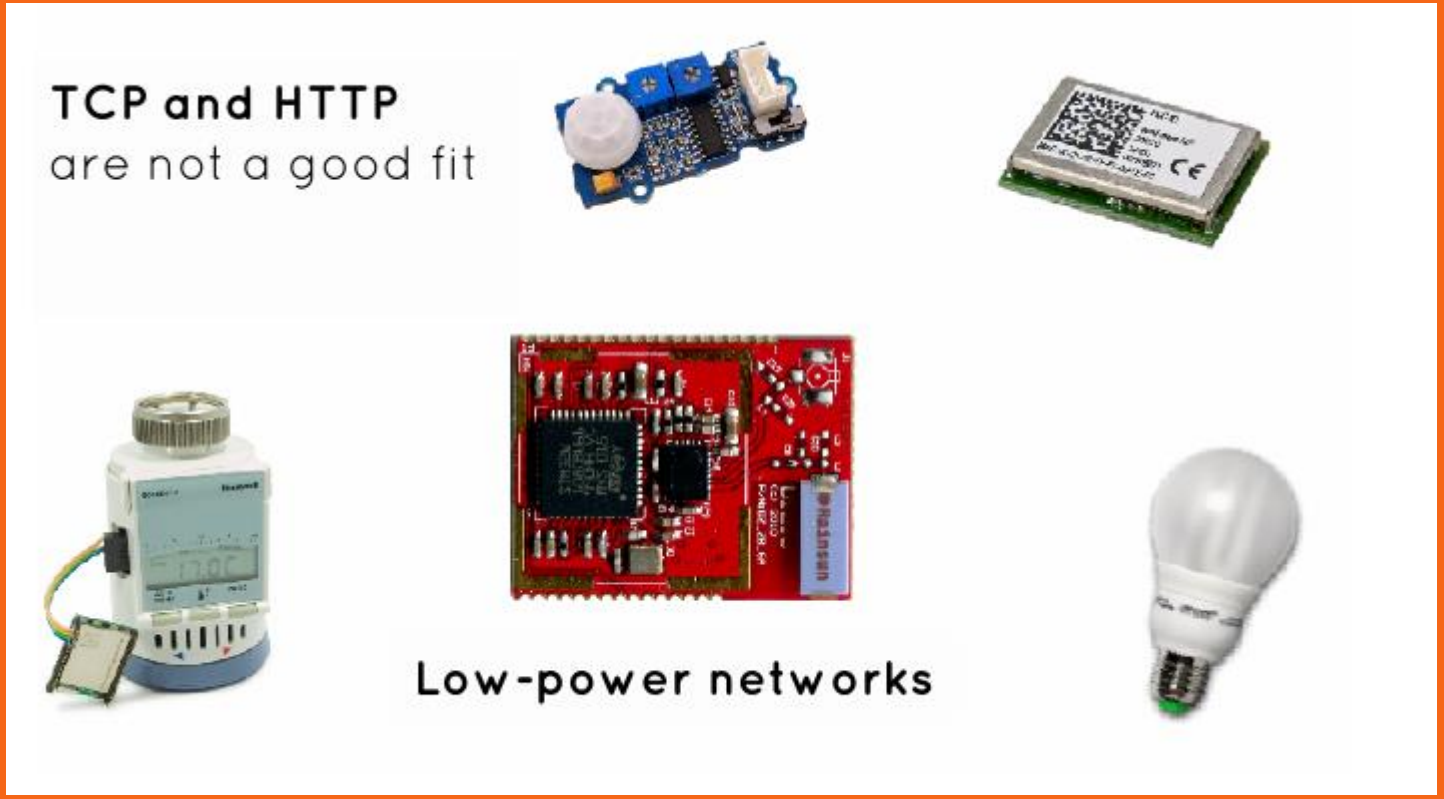
Full Research paper is available at : http://web.cs.wpi.edu/~rek/IoT/Papers/Colitti_CoAP_paper.pdf

TINY RESOURCE CONSTRAINED DEVICES

Class 1 devices
~100KiB Flash
~10KiB RAM

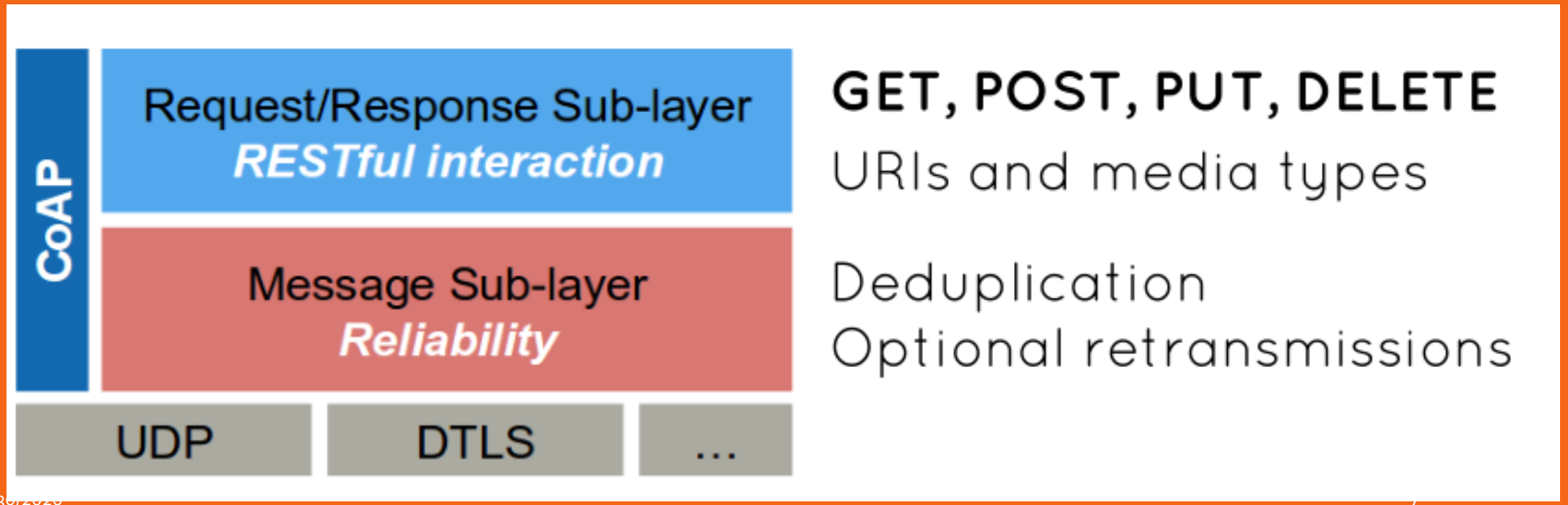
Target of less than 1$

CONT....



INTRODUCTION

- RESTful protocol designed from scratch Transparent mapping to HTTP Additional features of M2M scenarios.



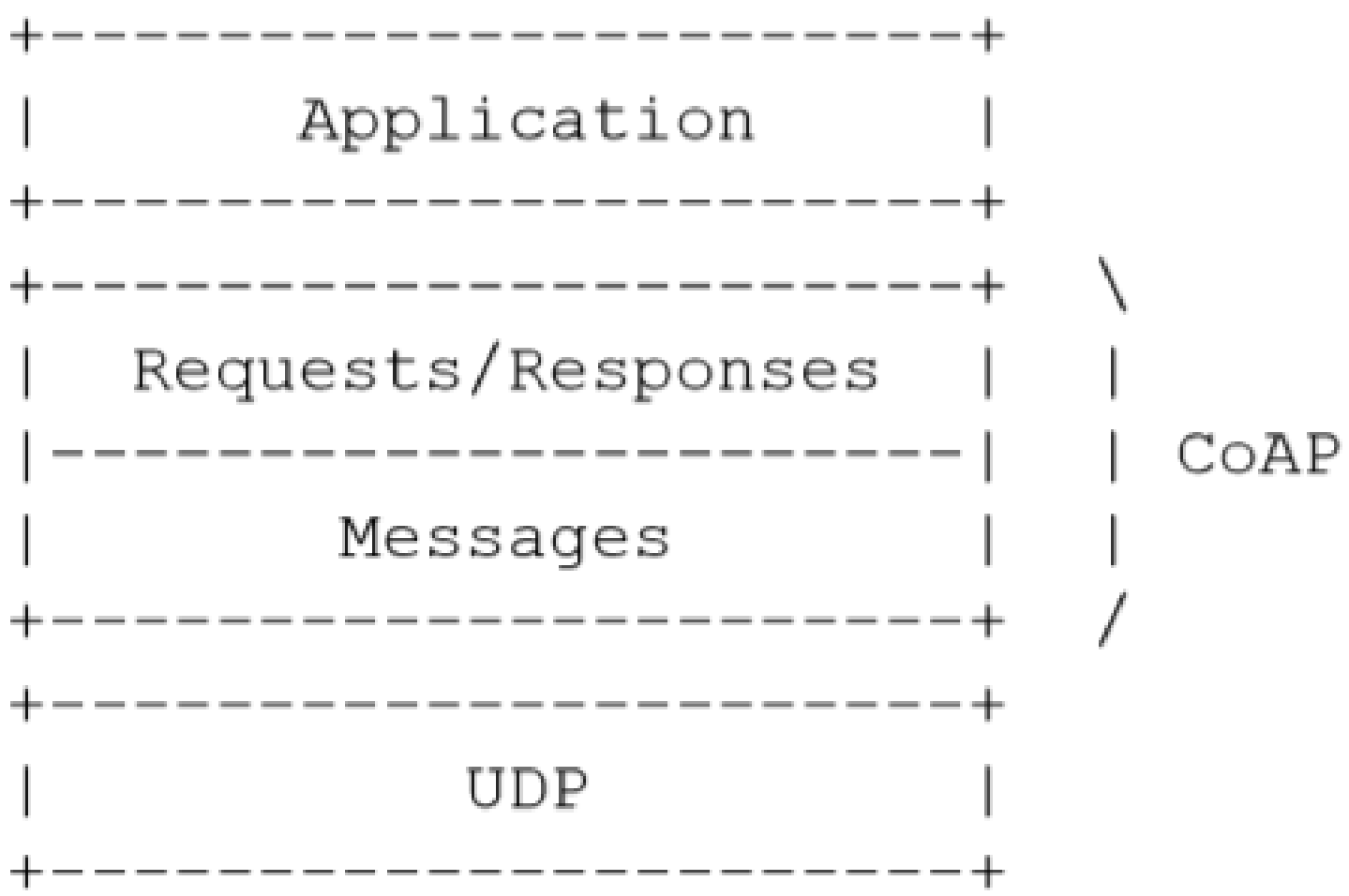
CONT….

- HTTP Like
- Runs over UDP by default but is not limited to it, as it can be implemented over other channels like TCP, DTLS or SMS
- Connection less protocol
- By default port number 5683 (in-secure mode)
- By default port number 5684 (secure mode) over DTLS

CONT…

- Binary based protocol
- 4 byte header size
- Security over DTLS rather than TLS in a normal TCP Transmission
- Concept of Tokens (match request & response)
- Support for URI and content-types
- built-in discovery, multicast support, and asynchronous message exchanges

ABSTRACT LAYERING OF COAP



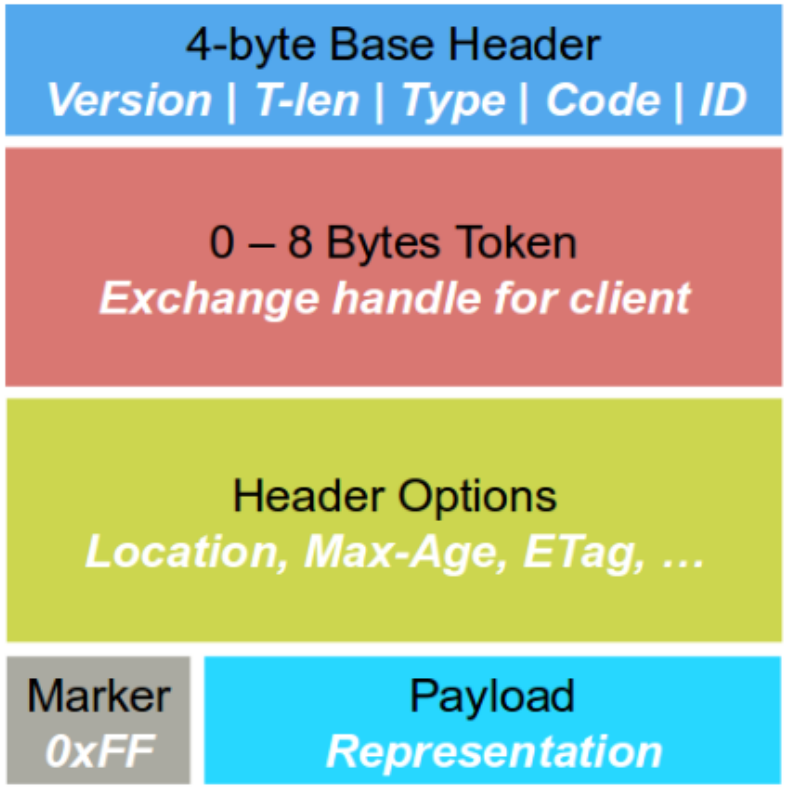
CONT....

Binary protocol

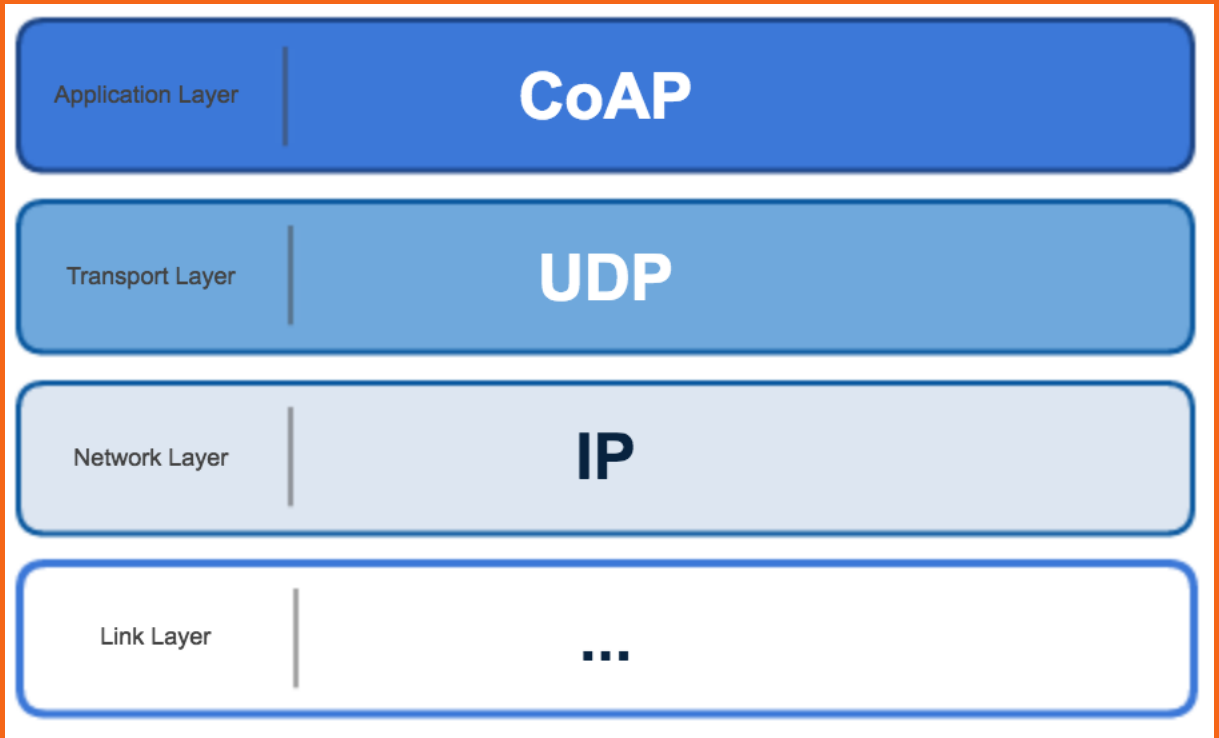
- Low parsing complexity
- Small message size

Options

- Numbers with IANA registry
- Type-Length-Value
- Special option header marks payload if present

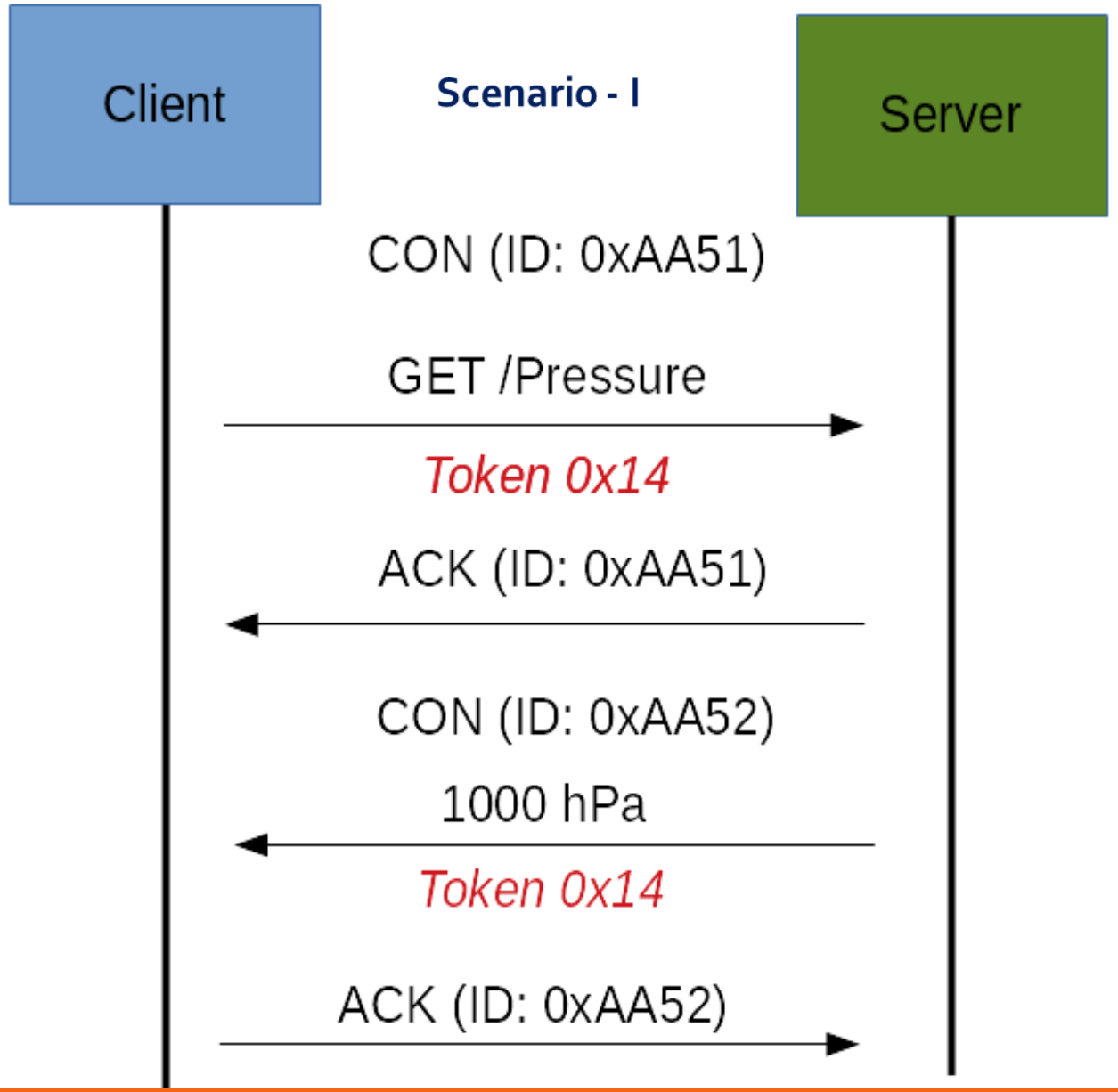


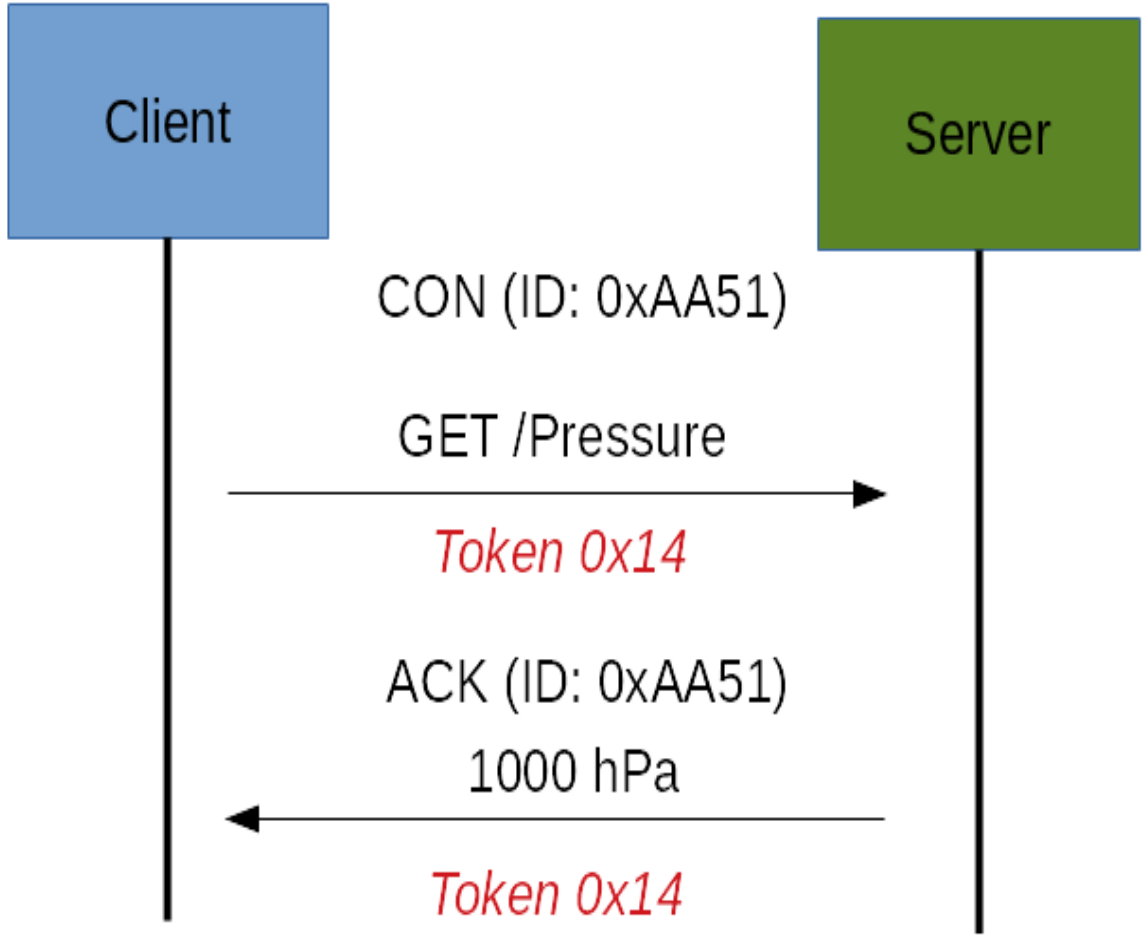
WHERE COAP FITS IN LAYERED MODEL



CONCEPT OF TOKEN -

- To match request and responses.
- Tokens are chosen by the client and help to identify request/response pairs that span several messages (e.g., a separate response, which has a new MID).
- **Servers do not generate Tokens** and only mirror what they receive from the clients.
- Tokens must be unique within the namespace of a client throughout their lifetime. This begins when being assigned to a request and ends when the open request is closed by receiving and matching the final response. Neither empty ACKs nor notifications (i.e., responses carrying the Observe option) terminate the lifetime of a Token.


Client
Scenario - I
Server
CON (ID: 0xAA51)
GET /Pressure
Token 0x14
ACK (ID: 0xAA51)
CON (ID: 0xAA52)
1000 hPa
Token 0x14
ACK (ID: 0xAA52)


Scenario - II
Client
Server
CON (ID: 0xAA51)
GET /Pressure
Token 0x14
ACK (ID: 0xAA51)
1000 hPa
Token 0x14

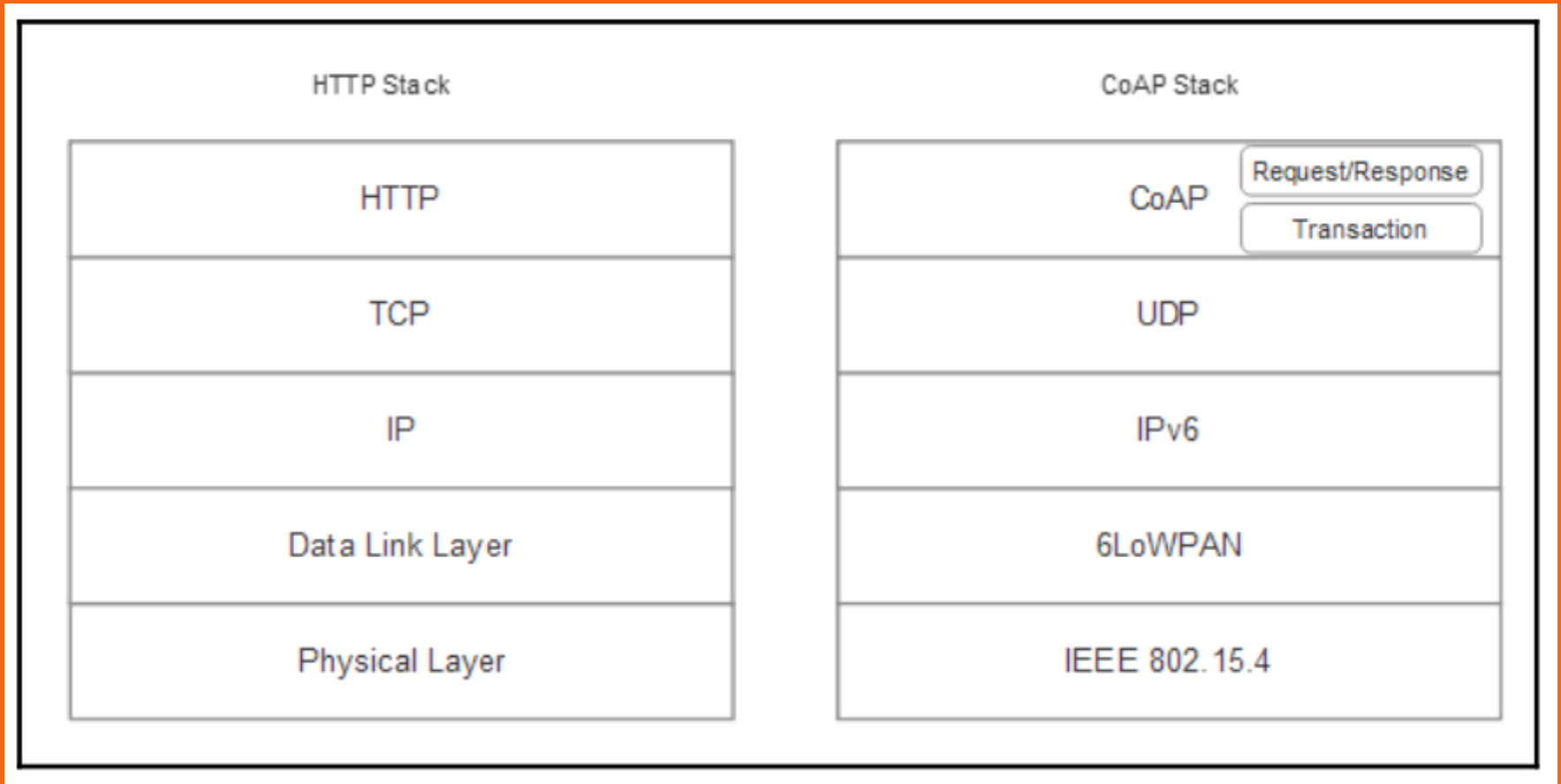
HOW TOKEN DIFFERS FROM MESSAGE ID

- Hope you have noticed in the last slide that the CoAP message, there is a Token. The Token is different from the Message-ID and it is used to match the request and the response.
- If the server can't answer to the request coming from the client immediately, then it sends an Acknowledge message with an empty response. As soon as the response is available, then the server sends a new Confirmable message to the client containing the response. At this point, the client sends back an Acknowledge message:

COAP LAYERS

- It has two basic layers – Request/Response & Transactional Layer
- **Request Response –**
  - Responsible for sending and receiving Restful-based queries. REST queries are piggybacked on CON or NON messages. A REST response is piggybacked on the corresponding ACK message.
- **Transactional Layer –**
  - Handles single message exchanges between endpoints using one of the four basic types (GET, PUT, POST and DELETE). The transactional layer also supports multicasting and congestion control.

HTTP STACK COMPARED TO COAP



CONT….

- At top most level, CoAP uses requests such as GET, PUT, POST AND DELETE as in HTTP.
- Similarly response codes mimic http such as
  - 2.01 – Created
  - 2.02 – Deleted
  - 2.04 – Changed
  - 2.05 – Content
  - 4.04 – Not found
  - 4.05 – Method not allowed

ADDRESSING/URIS

- URIs consist of the hostname, port number, path and query string, which are specified by option fields Uri-Host, Uri-Port, Uri-Path and Uri-Query, of which Uri-Host and Uri-Port are implicit as they are part of underlying layers. Uri-Path and Uri-Query are significant and part of the CoAP message.
- For example, if we request a resource with URI *coap://hostname:port/leds/green?q=state&on*, the following options are generated:

COAP SYSTEM – 7 MAIN FACTORS

1. End Points

- A host or node participating in CoAP communication is known as an endpoint.
- The endpoint on which resources are defined and is a destination for requests is known as a server or, more precisely, the origin server and the endpoint from which requests are made for target resources is known as a client.
- Similarly, a server is the source and a client is the destination for responses.
- Certain endpoints like proxies act as the intermediate client and server.

CONT….

2. PROXIES –

- A CoAP end point that is tasked by CoAP clients to perform requests on its behalf.
- Reducing network load, access sleeping nodes and providing a layer of security are some of the roles of a proxy.

Forward Proxy -

It is explicitly known by the client

CONT..

- **REVERSE PROXY:**
  - Acts as if it was the origin server
  - It knows explicitly the servers that is proxying

In Nutshell – A proxy can map from one CoAP request to another CoAP request or even translate to a different protocol (cross-proxying). A common situation is an edge router from a CoAP network to HTTP services for cloud based internet connections.

COAP SYSTEM – 7 MAIN FACTORS CONT..

3. Client –

- The originator of a request. The destination endpoint of a response.

4. Server –

- The destination endpoint of a request. The originator of a response.

5. Intermediary –

- A client acting as both a server and client towards an origin server. A proxy is an intermediary.

CONT..

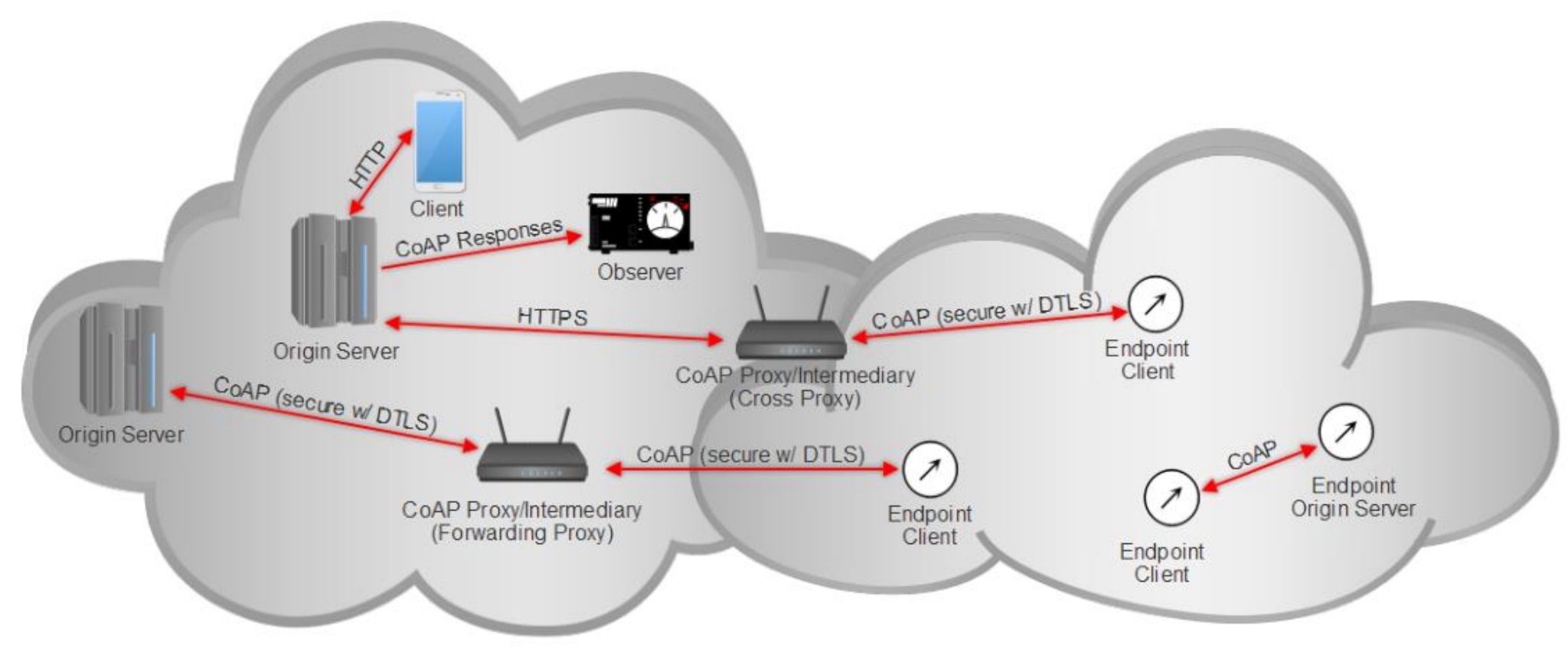
6. Origin Servers –

- The server on which a given resource resides.

7. Observers-

- A observer client can register itself using a modified GET message. The observer is then connected to a resource and if the state of the resource changes, the server will send a notification back to server.

COAP ARCHITECTURE



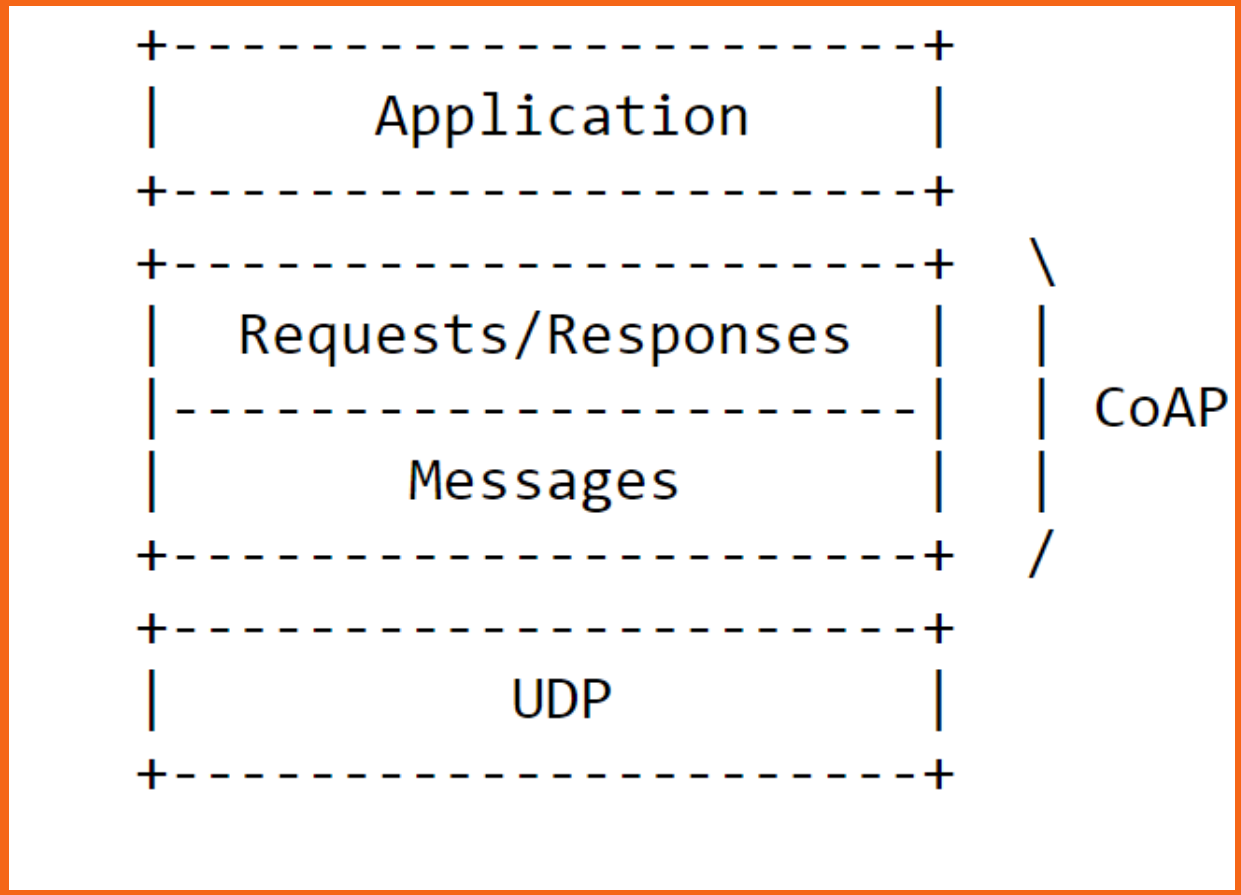
COAP MESSAGE FORMATS

- Protocols based on UDP transport imply that the connection may not be inherently reliable.
- To compensate for reliability issues, CoAP introduces two message types differ by either requiring acknowledgement or not. An additional feature of this approach is that message can be asynchronous.

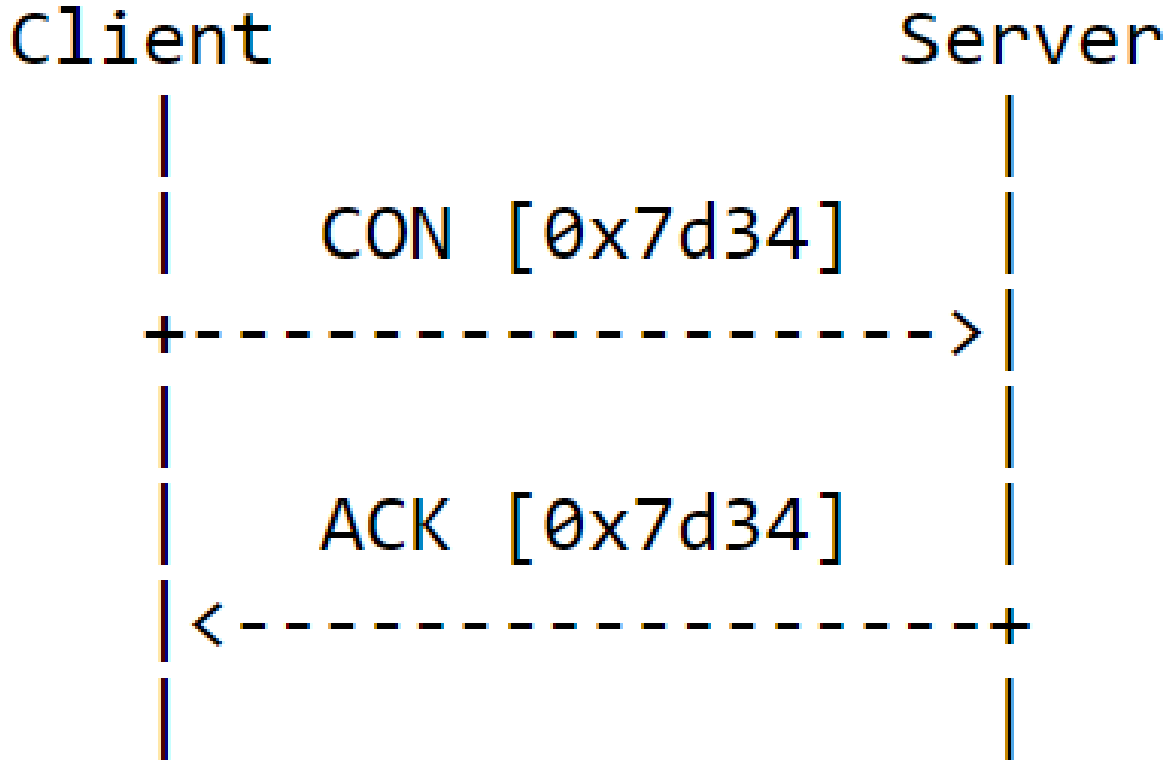
CONT.....

- CoAP defines four types of messages:
- Confirmable, Non-confirmable, Acknowledgement, Reset.
- Method Codes and Response Codes included in some of these messages make them carry requests or responses. The basic exchanges of the four types of messages are somewhat orthogonal to the request/response interactions.
- Requests can be carried in Confirmable and Non- confirmable messages, and responses can be carried in these as well as piggybacked in Acknowledgement messages.

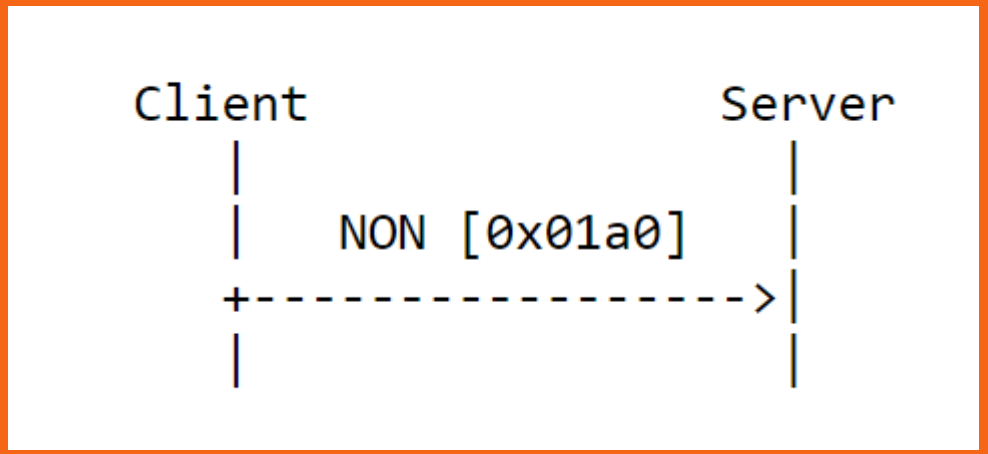
ABSTRACT LAYERING OF COAP



RELIABLE MESSAGE TRANSMISSION



NON CONFIRMABLE MESSAGE TRANSMISSION



MESSAGES (CONT.…)

COFIRMABLE (CON)

- Requires an ACK. If a CON message is sent an ACK must be received within a random time interval between ACK_TIMEOUT and (ACK_TIMEOUT * ACK_RANDOM_FACTOR).
- If the ACK is not received, the sender transmits the CON message over and over at exponentially until it receives the ACK or RST.
- Above mechanism in CoAP is form of congestion control. There is a maximum number of attempts set by MAX_RETRANSMIT. This is the resiliency mechanism to compensate for the lack of resiliency in UDP.

CONT. …

- **NON-CONFORMABLE (NON) –**
  - Requires no ACK. Essentially a fire and forget message or broadcast.
- **ACKNOWLEDGEMENT (ACK) –**
  - Acknowledges a CON message. ACK message can **piggyback** along with other data,

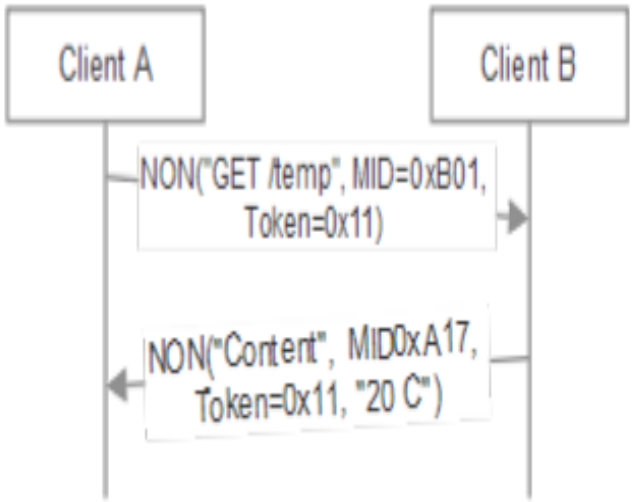
CONT…

RESET (RST) –

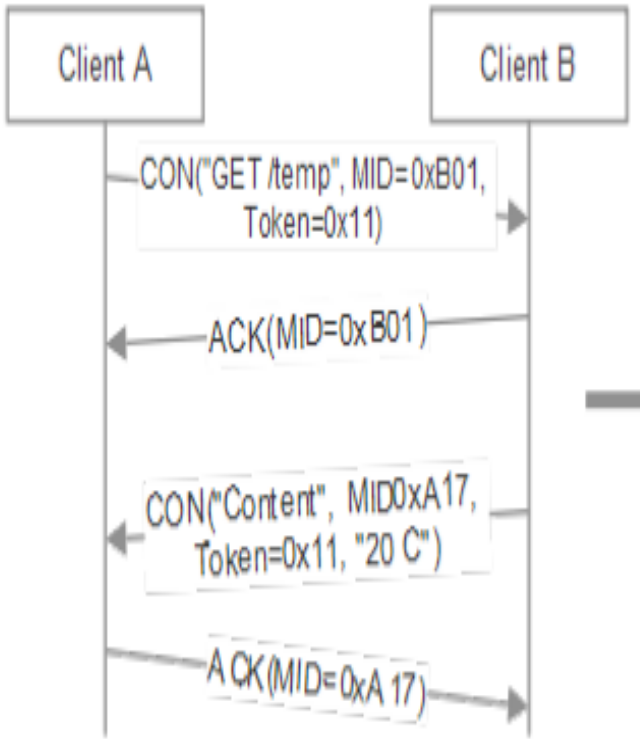
- Indicates a CON message has been received but the context is missing. The RST message can be piggybacked along with other data.

SOME EXAMPLES

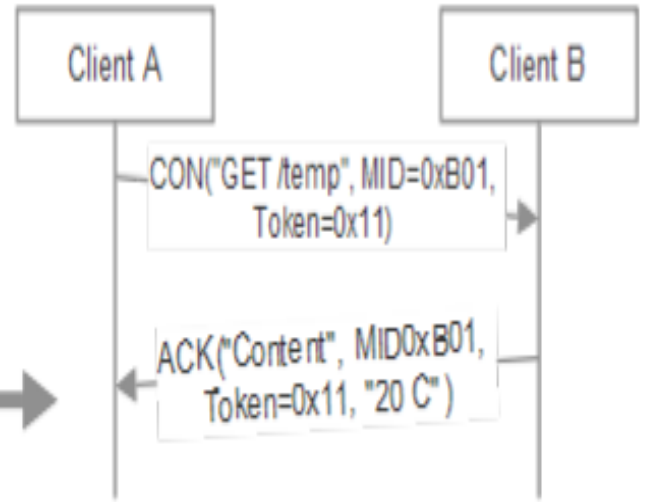
CoAP Example: Non-confirmable request and response



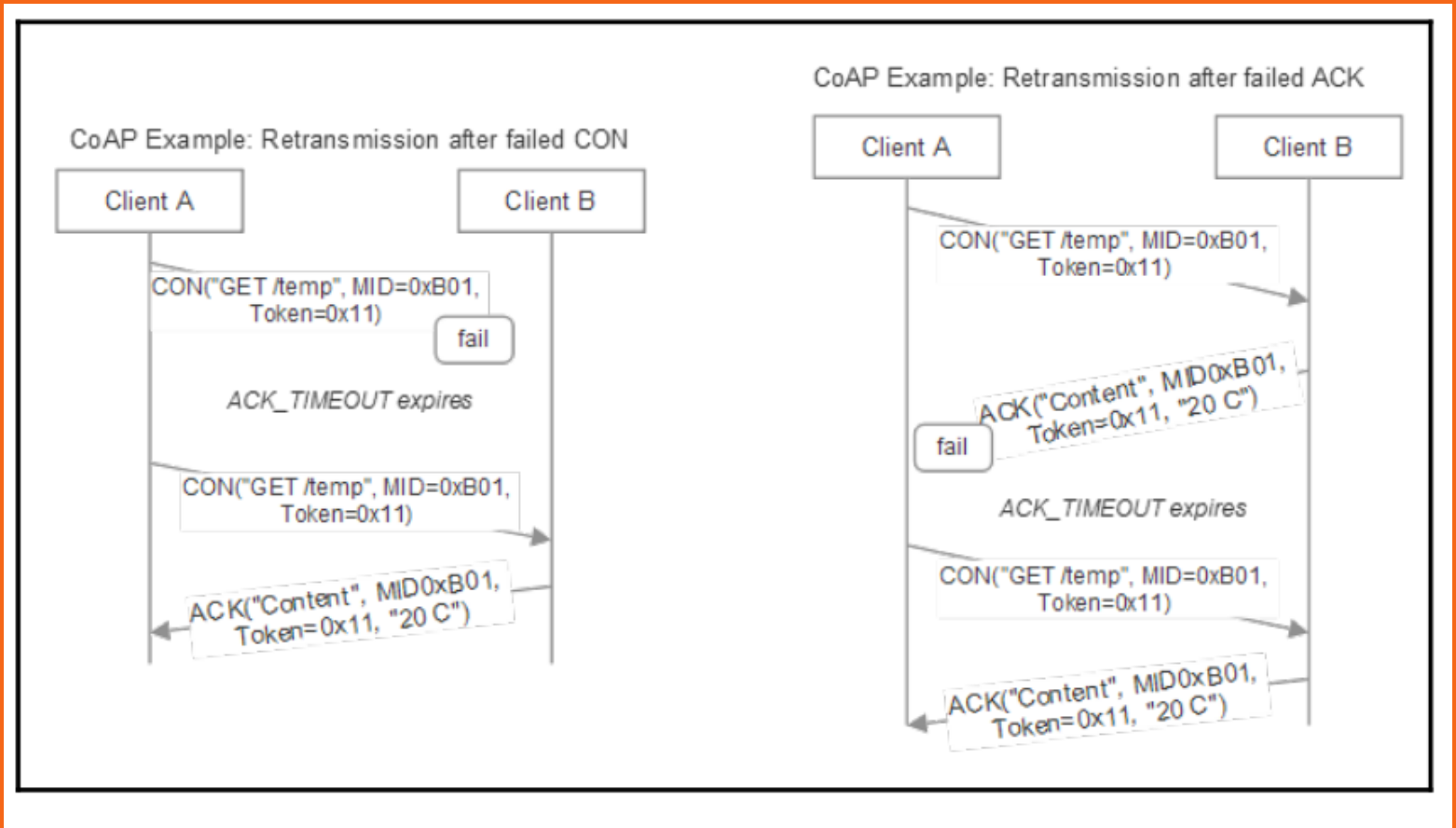
CoAP Example: Confirmable request and response



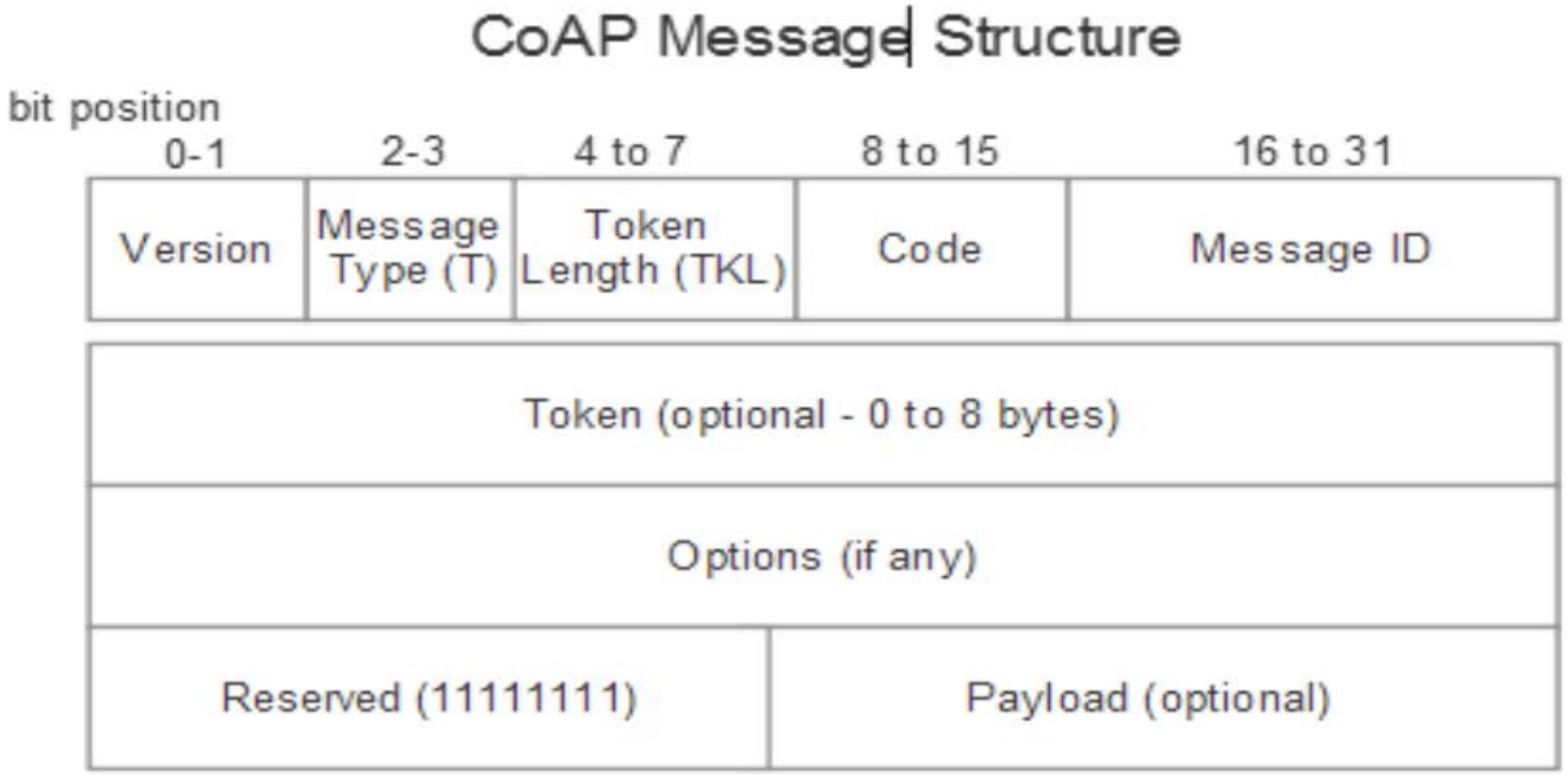
CoAP Example: Confirmable with Piggyback (alternate)



RETRANSMISSION FORMAT



MESSAGE STRUCTURE



CONT....

- One of the key design goals of CoAP is to avoid fragmentation at underlying layers, especially at the link layer, i.e., the whole CoAP packet should fit into a single datagram compatible with a single frame at the Ethernet or IEEE 802.15.4 layer. This is possible with a compact 4-byte binary header, optional fields and payload, as shown in previous slide.

CONT…

- **Version:** 2-bit version number, currently fixed at 0x01.
- **Type of messages:** CoAP supports four types of messages with the following 2 bit transaction codes.

CON 00(0)	ACK 10(2)
NON 01(1)	RST 11(3)

REQUEST/RESPONSE CODES

- A 3-bit class ID and 5-bit detail in the c.dd format forms this field. The class values used for requests, success responses, client error responses and server error responses are 0, 2, 4 or 5, respectively. The detail carries the request code or response code depending on the class value. Code 0.00 indicates an empty message.

MESSAGE ID

A 16-bit unsigned number in network byte order is used to match acknowledgement or reset messages and eliminate duplicate messages.

TOKENS

An optional token field, which is limited to 0 to 8 bytes currently to match request responses, may be kept after the header, which is of TKL number of bytes specified in the header.

OPTIONS

Zero or more option fields may follow a token. A few options are Content Format, Accept, Max-Age, Etag, Uri-Path, Uri-Query, etc.

PAYLOAD – ACTUAL DATA

- The minimum payload length is 0, and the maximum payload length is 2^**16-1**= 65535.

DISCOVERY (RFC6690)

- In M2M applications, for example, home or building automation, there is a need for local clients and servers to find and interact with each other without human intervention. The Core Link Format can be used by servers in such environments to enable Resource Discovery of the resources hosted by the server.

EXAMPLE

- CoAP messages are used to discover CoAP Discovery Gateways in a hierarchical fashion. Having completed the topology discovery phase, a CoAP client may initiate discovery of particular CoAP server resources, e.g. light dimmers, or a more general wildcard discovery may be done by the client; building a complete database. The same request MUST be sent to each discovered CoAP Discovery Gateway interface in a sequential fashion.

OBSERVE METHOD (RFC 7641)

- Mostly used for resource status poling.
- For instance:
  - State of a resource can be change over time.
  - coap://hostname:port/temperature/felt

THANK YOU !!