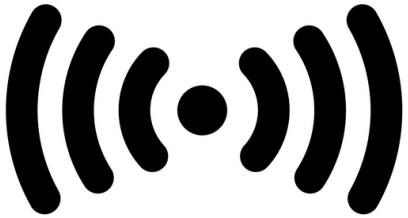


CoAP
Constrained **A**pplication
Protocol

CONSTRAINED APPLICATION PROTOCOL (COAP)

Bhupendra Pratap Singh
ACTS, CDAC

Bhupendra Pratap Singh

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$

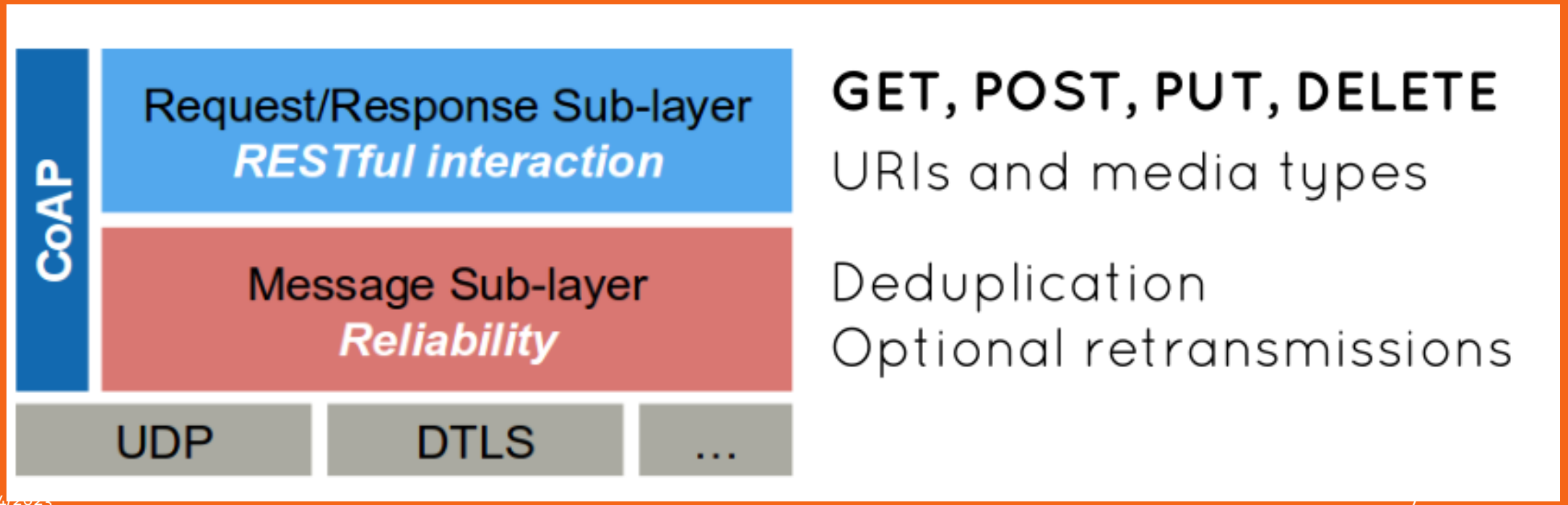
CONT....

TCP and HTTP
are not a good fit

Low-power networks

INTRODUCTION

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



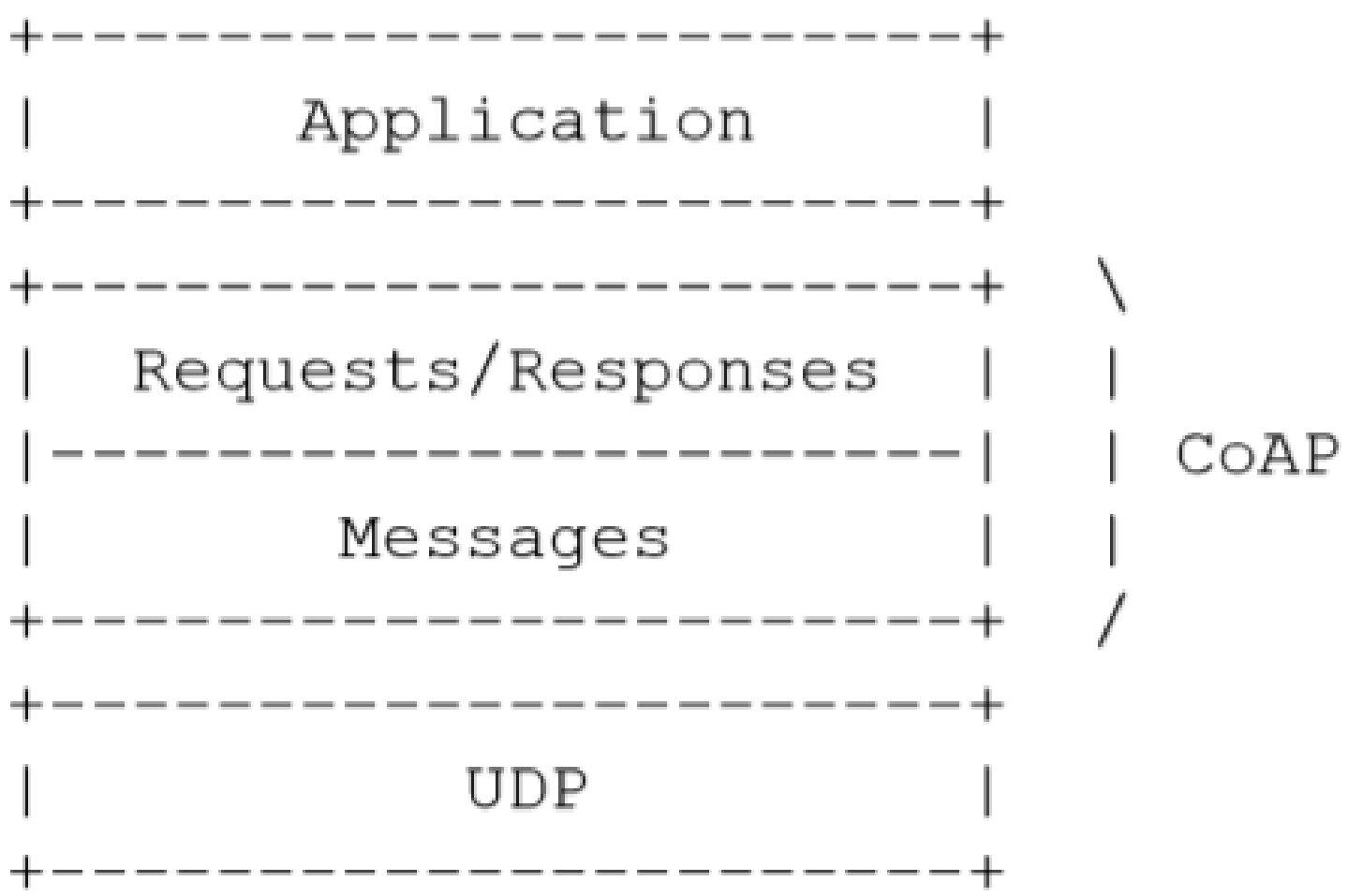
CONT….

- 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



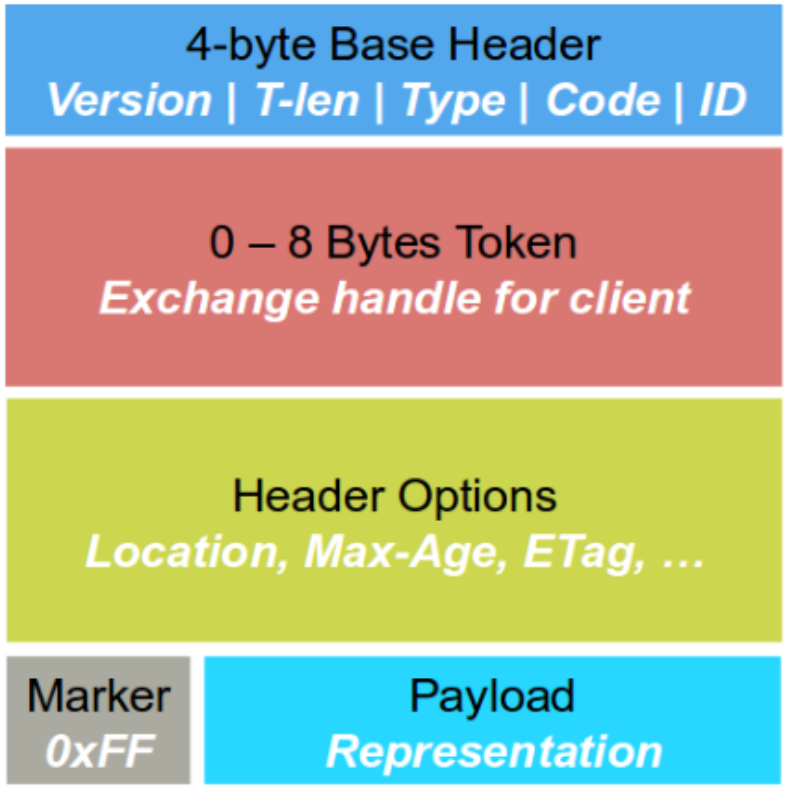
CONT....

Binary protocol

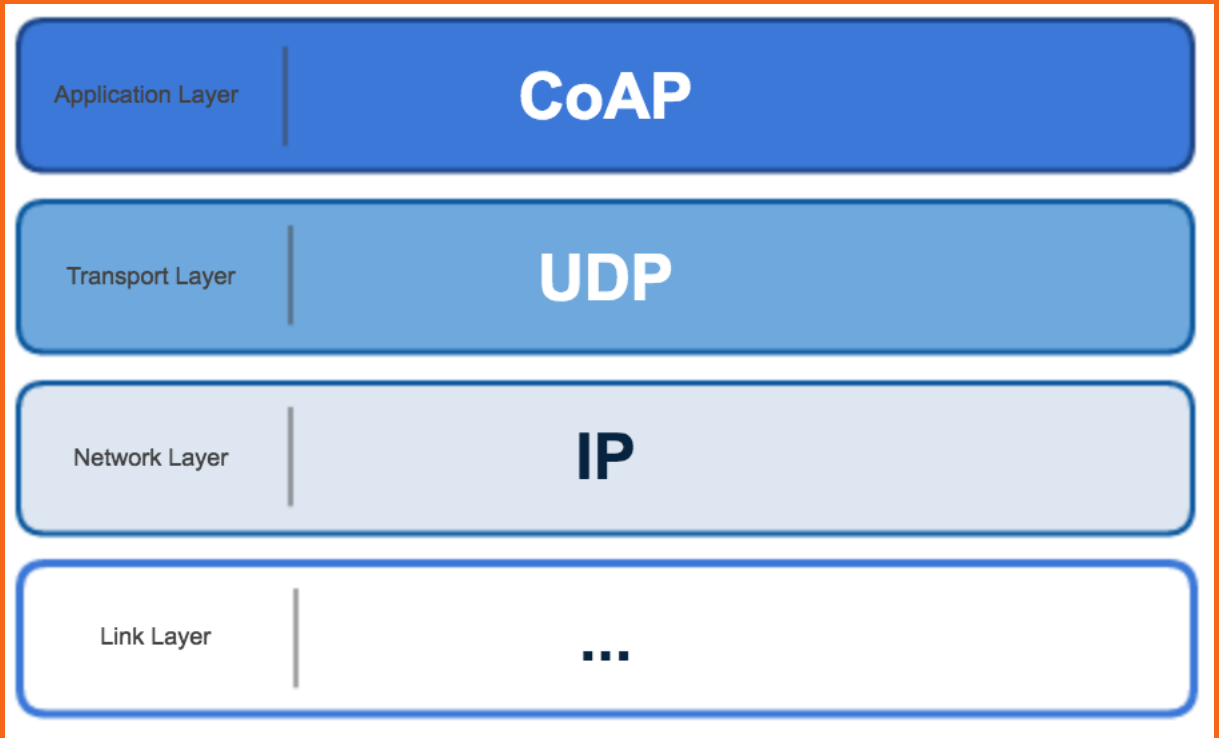
- Low parsing complexity
- Small message size

Options

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

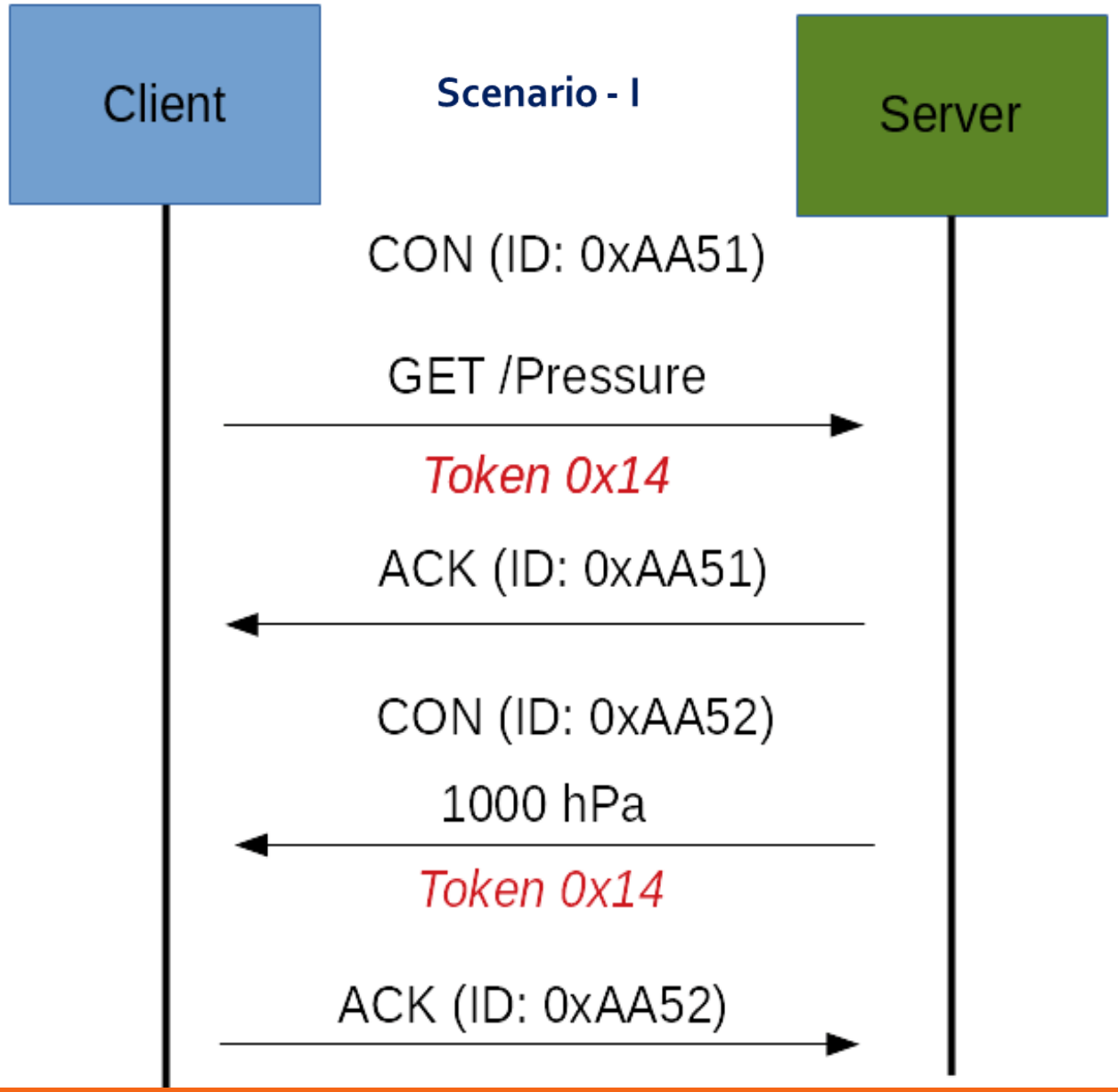


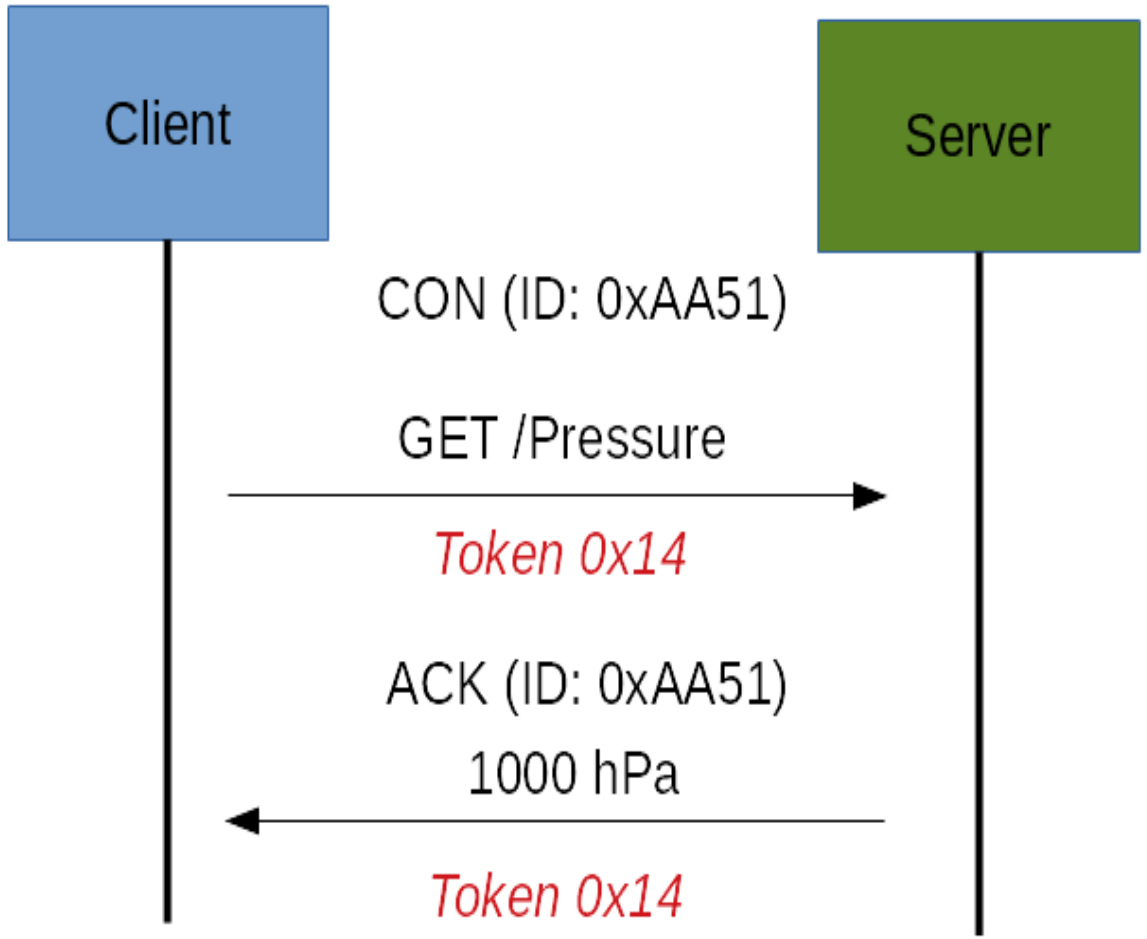
WHERE COAP FITS IN LAYERED MODEL



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

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:

CONT….

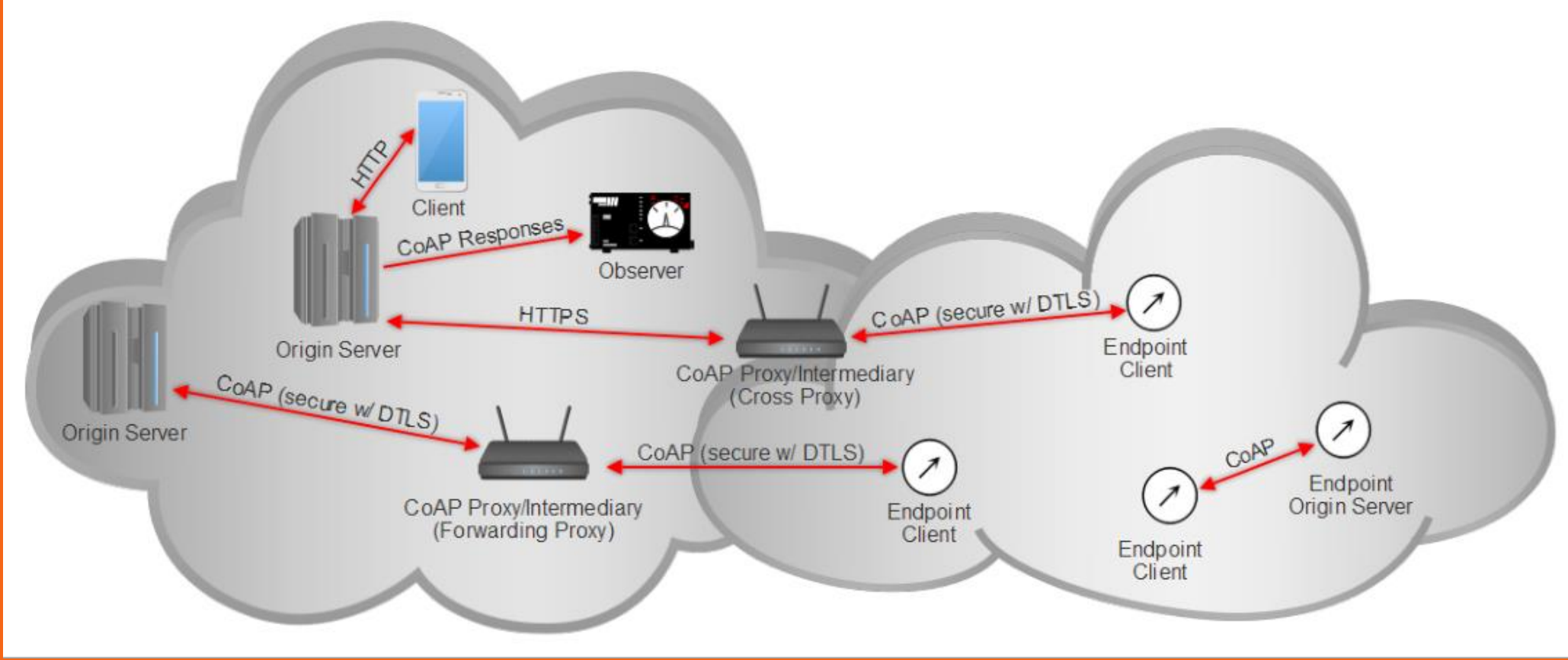
2. PROXIES –

- 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

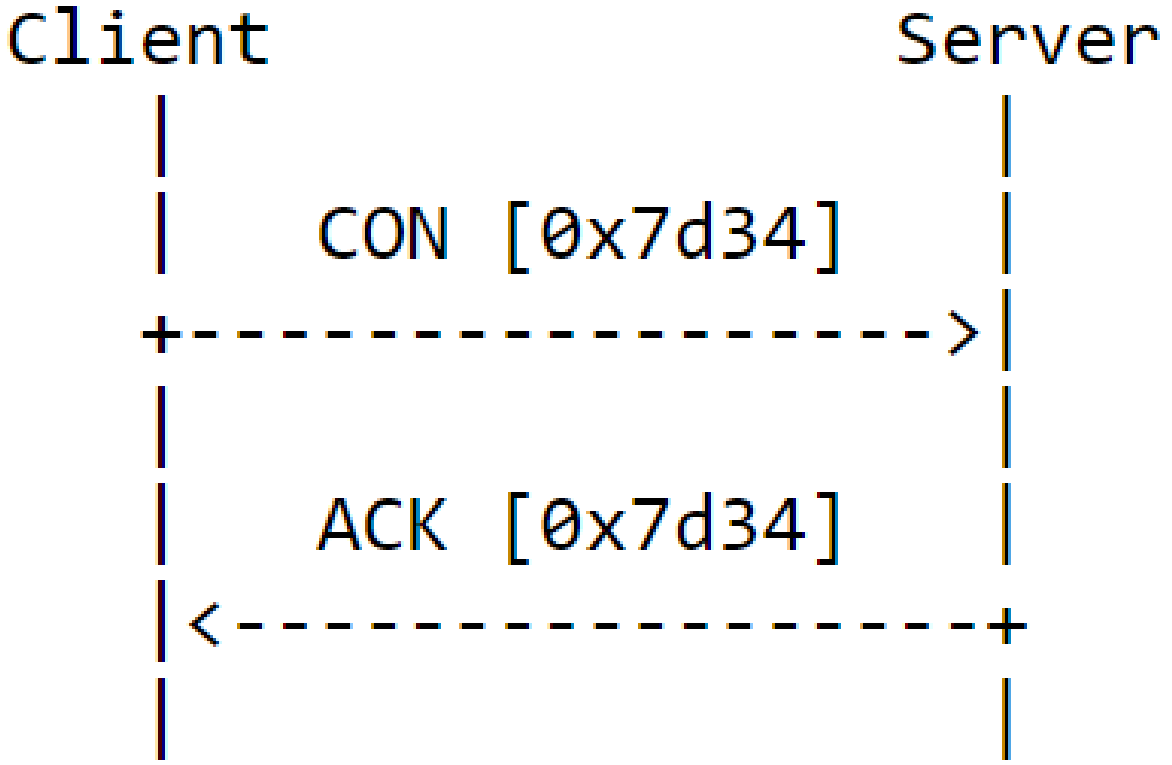
COAP ARCHITECTURE



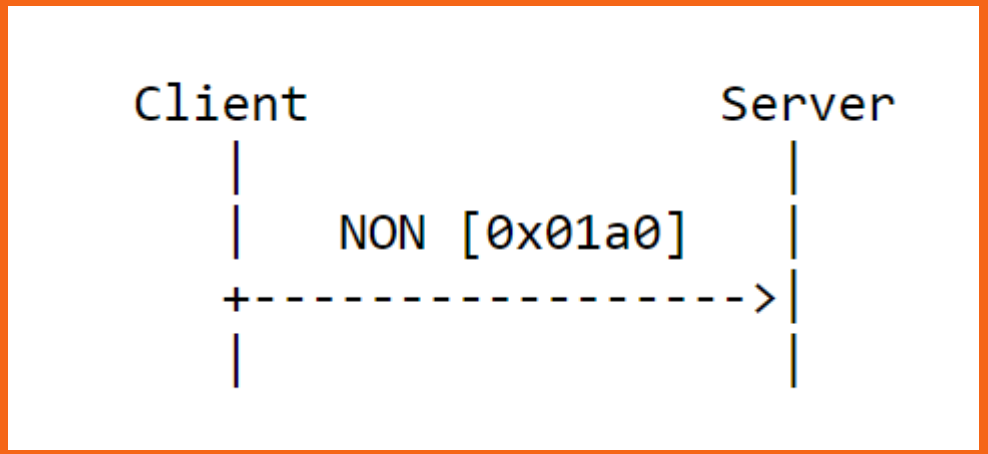
CONT.....

- CoAP defines four types of messages:
- Confirmable, Non-confirmable, Acknowledgement, Reset.

RELIABLE MESSAGE TRANSMISSION

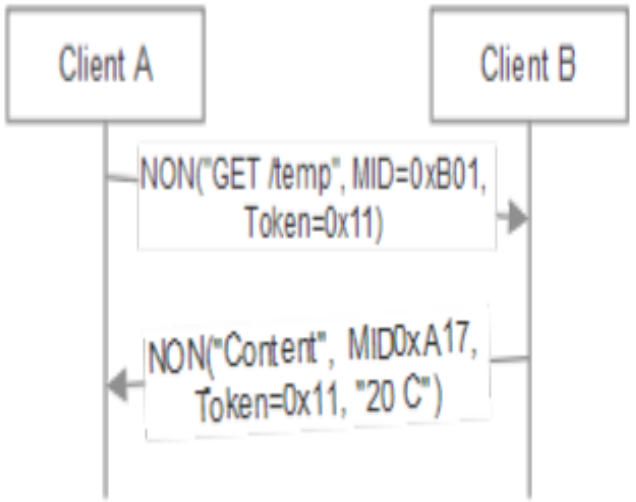


NON CONFIRMABLE MESSAGE TRANSMISSION

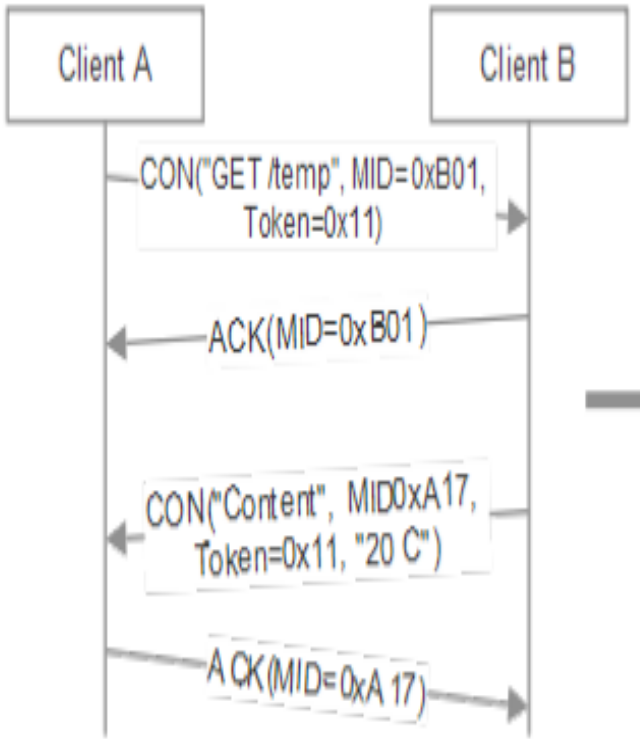


SOME EXAMPLES

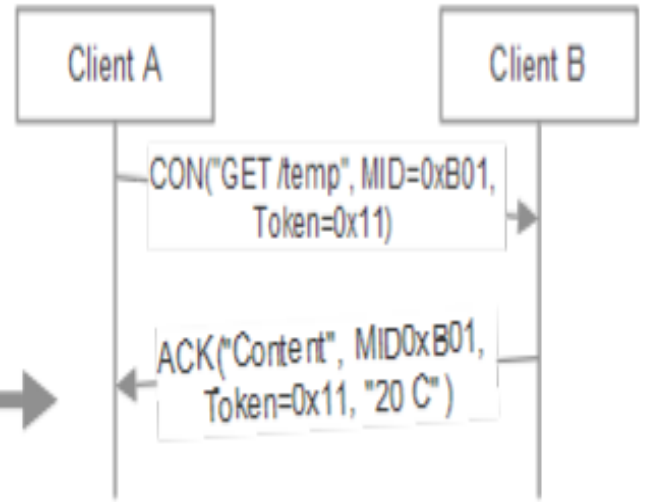
CoAP Example: Non-confirmable request and response



CoAP Example: Confirmable request and response



CoAP Example: Confirmable with Piggyback (alternate)



THANK YOU !!