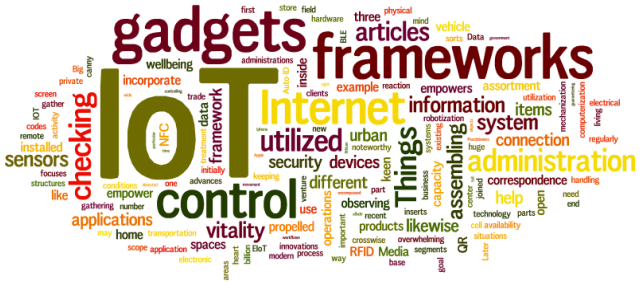
CS578: Internet of Things

CoAP: Constrained Application Protocol



Dr. Manas Khatua

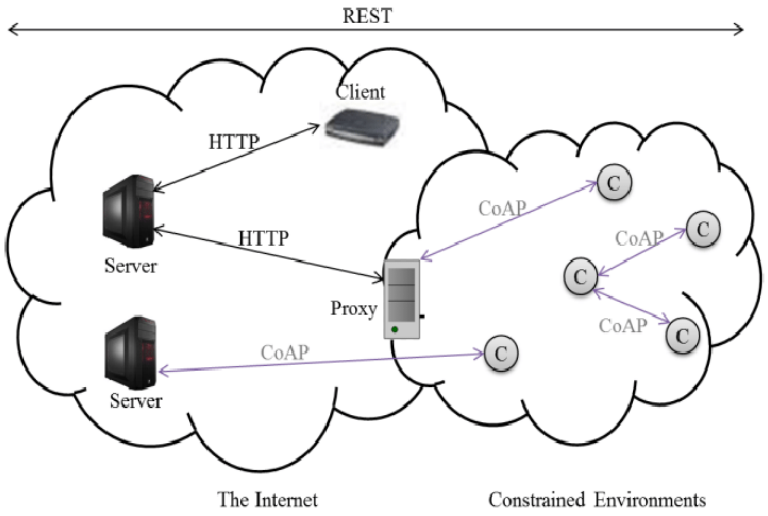
Assistant Professor

Dept. of CSE, IIT Guwahati

E-mail: manaskhatua@iitg.ac.in

What is CoAP

- CoAP - Constrained Application Protocol.
- **Specialized** web transfer protocol.
- Devised for **constrained and low power** networks.
- CoAP is an **application layer protocol** (similar as HTTP).
- Follows the **request-response** pattern used by HTTP.
- CoAP has a **transparent mapping** to HTTP.
- CoAP protocol spec is specified in **RFC 7252**.

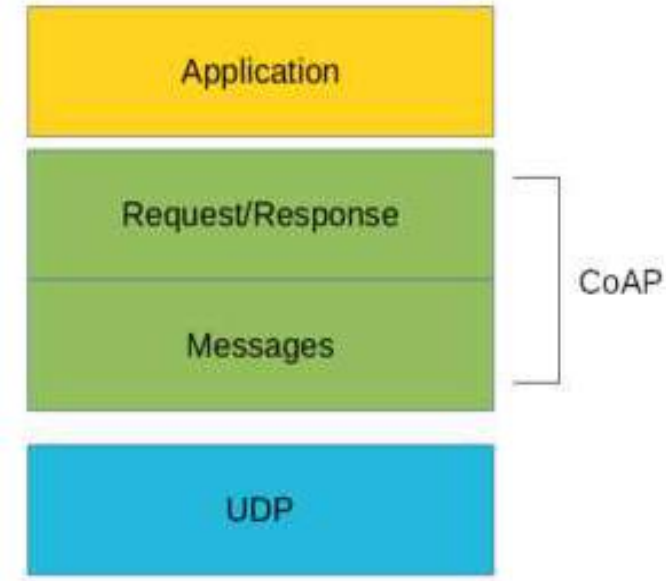


Characterstics of CoAP

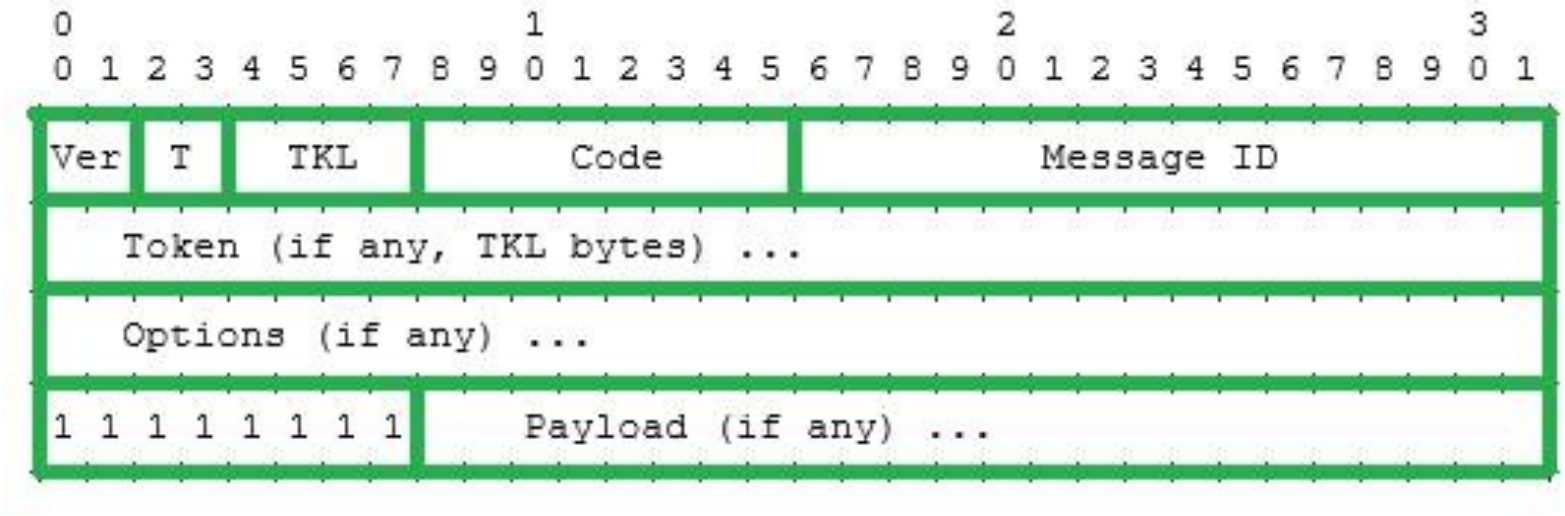
- It is very efficient **RESTful** protocol.
- It is Embedded web transfer protocol
- Low header overhead and parsing complexity.
- Proxied to/from HTTP.
  - GET, POST, PUT and DELETE methods are used.
  - Uses subset HTTP response codes.
  - URI is supported.
- It uses small and simple **4 byte header.**
- Supports binding to UDP, SMS and TCP.
- DTLS based certificate security is used.

CoAP Structure

- There are two different layers that make CoAP Protocol:
  - Messages
  - Request/Response.
- The **Messages layer** deals with UDP and with asynchronous messages.
- **Message layer** is meant for Re-transmitting lost packets.
- The **Request/Response layer** manages request/response interaction based on request/response messages.
- Request/Response layer contains methods like **GET,PUT**, **POST** and **DELETE**.
- Message layer supports **four** types of messages:
  - Confirmable(CON)
  - Non-confirmable(NON)
  - Acknowledgment(ACK)
  - Reset(RST)
- **Request Methods**: GET, POST, PUT, DELETE.
- **Response Methods**: 2.xx (success), 4.xx (client error), 5.xx(server error).



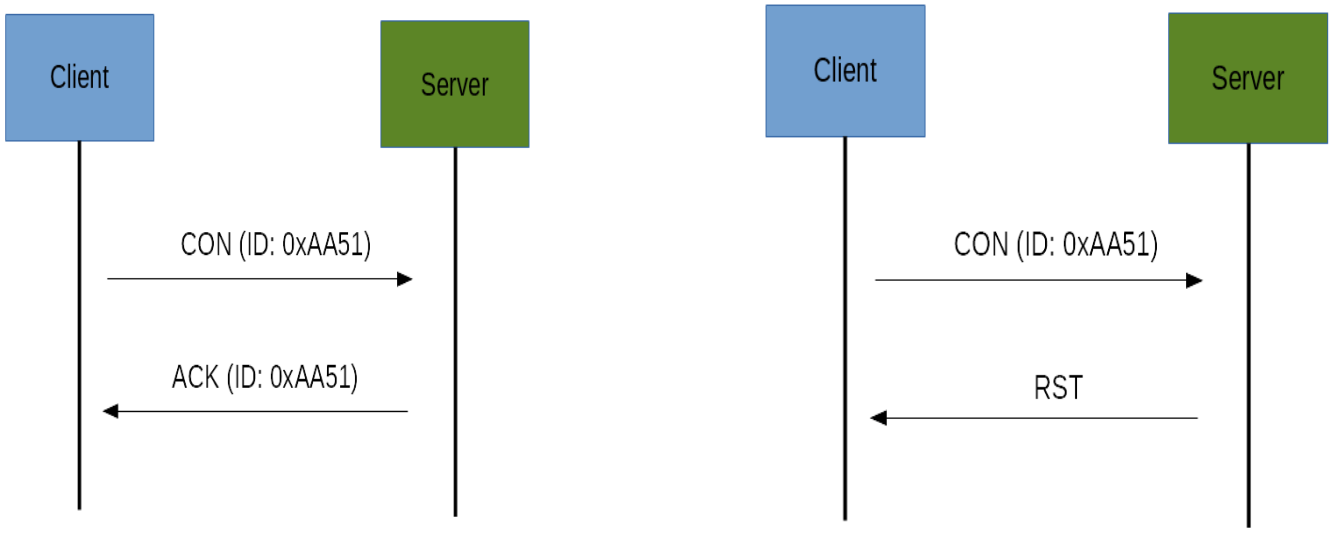
CoAP Message Format



- **Ver (2 bit):** indicating the CoAP **version.**
- **T (2 bit):** indicating the **message type**
  - Confirmable (CON), Non-confirmable (NON), Acknowledgement (ACK), Reset (RST)
- **TKL (4 bit)**: Specifies the size (0-8 bytes) of the Token field
- **Code (8 bit)**: indicates the request method for a request message, and a response code for a response message
  - Example: GET is the request method, 2.05 is the response code
- **Message ID (16 bit)**:
  - Detects message duplication
  - Used to match ACK and RST message types to CON and NON message types.
- **Token (0-8 byte):** correlates requests and responses

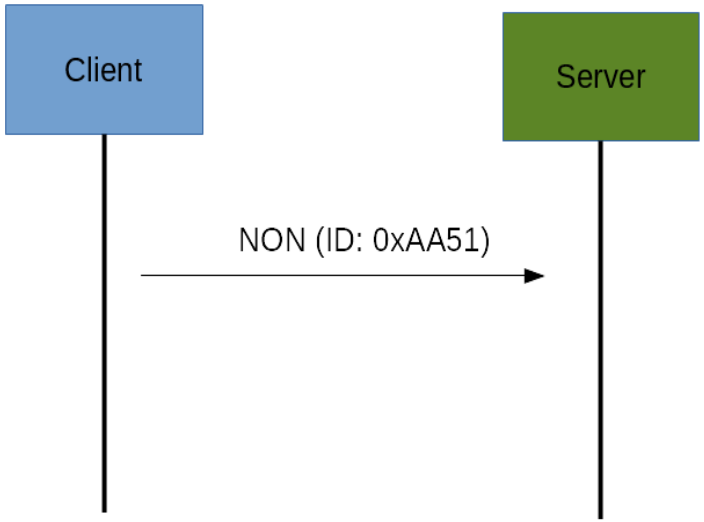
CoAP Messages Model

- Lowest layer of CoAP.
- **Deals with UDP** exchanging messages between endpoints.
- Each CoAP message has a **unique ID**
- **Unique ID** is useful to detect message duplicates.
- In CoAP, a **reliable** message is obtained using a **Confirmable message (CON).**
  - A Confirmable message is sent again and again until the other party sends an **acknowledge message (ACK).**
  - The ACK message contains the same ID of the **confirmable message (CON).**
  - If the server has troubles managing the incoming request.
  - Send back a **Reset message (RST)** instead of the Acknowledge message (**ACK**).



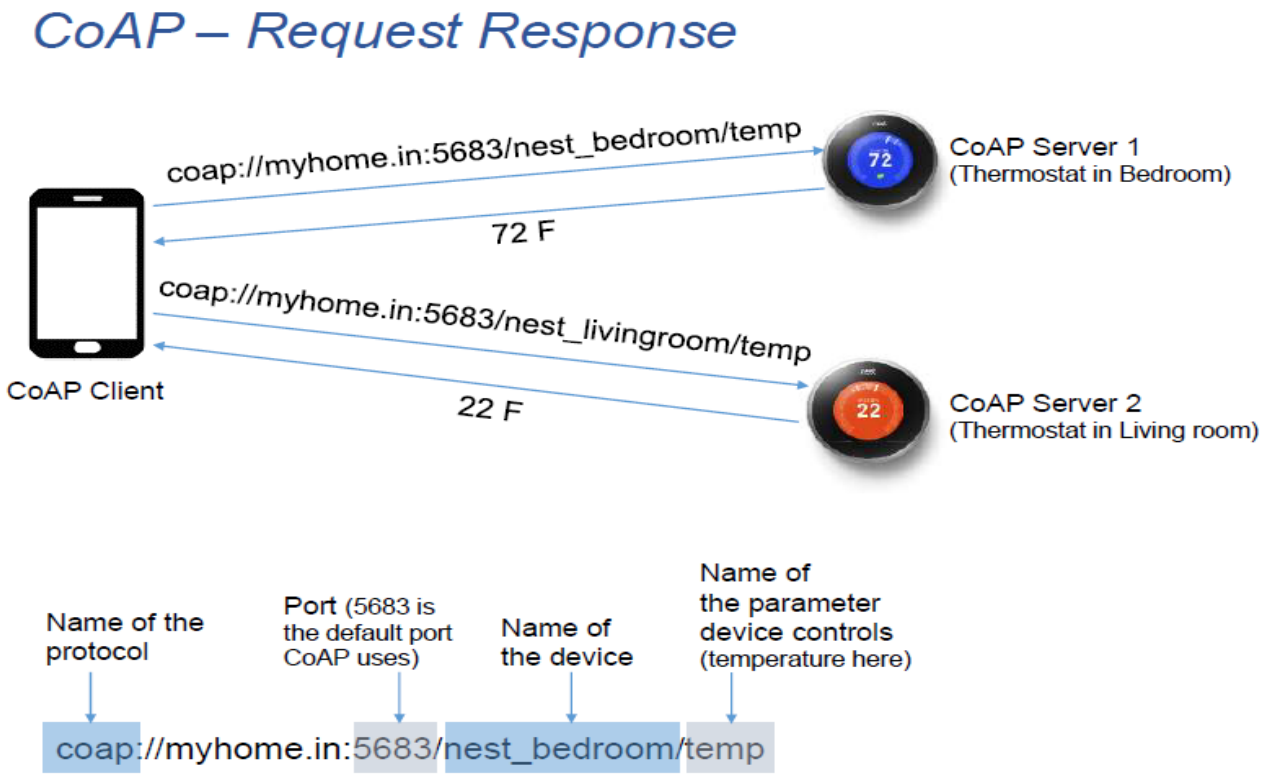
Cont…

- The other message category is the **Non-confirmable (NON)** messages.
- These messages **don't** require an **Acknowledge** by the server.
- They are unreliable messages or in other words messages that **do not** contain critical information that must be delivered to the server.
  - **Example :** Messages that contain **read** values from **sensors**.
- Even if these messages are unreliable, they have a **unique ID**.



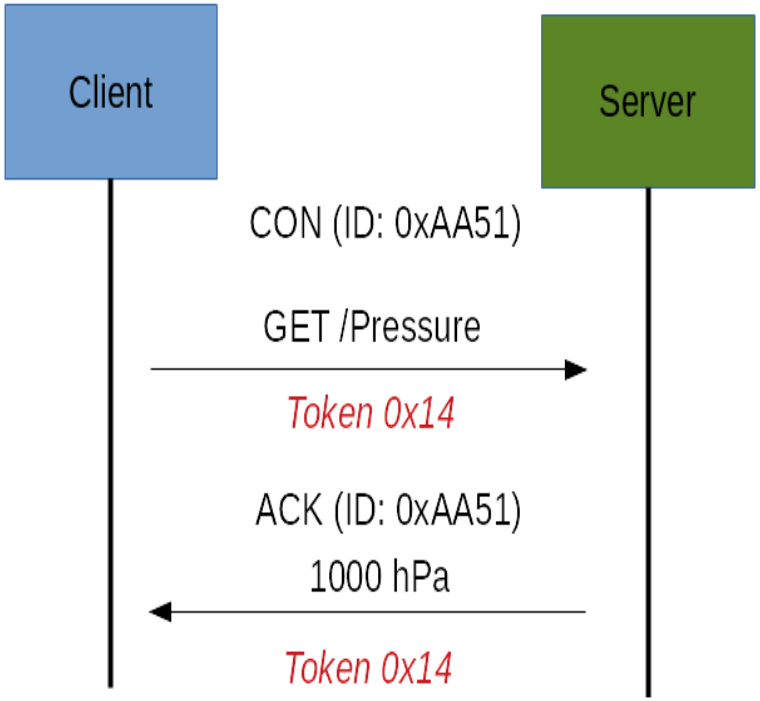
CoAP Request/Response Model

- The CoAP **Request/Response** is the second layer in the CoAP Abstraction layer.
- The request is sent using a Confirmable (CON) or Non-Confirmable (NON) message.
- There are several scenarios(Piggy-Backed,Separate Response) depending on if the server can answer immediately to the client request or answer if not available.



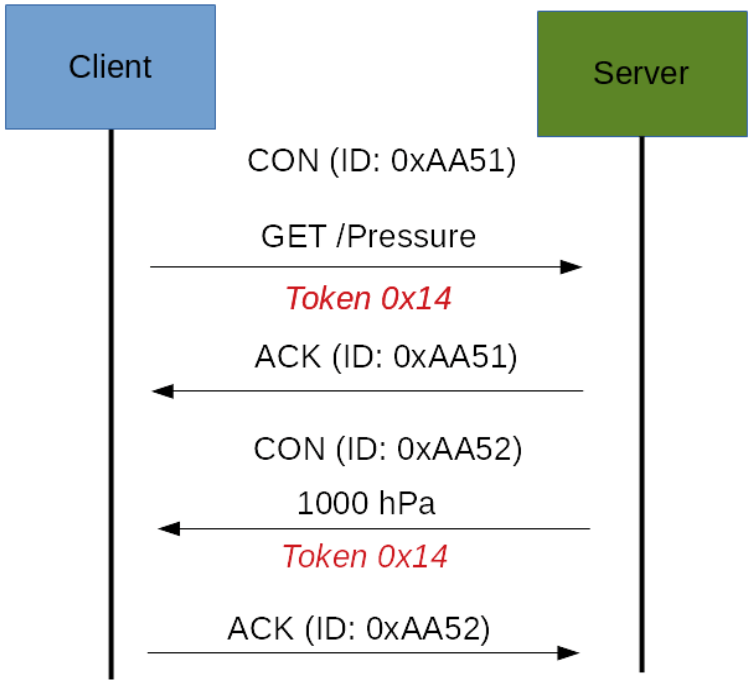
Cont…

- If the server can answer immediately to the client request(**Piggy-Backed response**)
- If the **request** is carried using a Confirmable message (CON).
  - the server sends back an Acknowledge message to client containing the **response or the error code**.
- **The Token** is different from the Message-ID and it is used to match the **request** and the **response**.



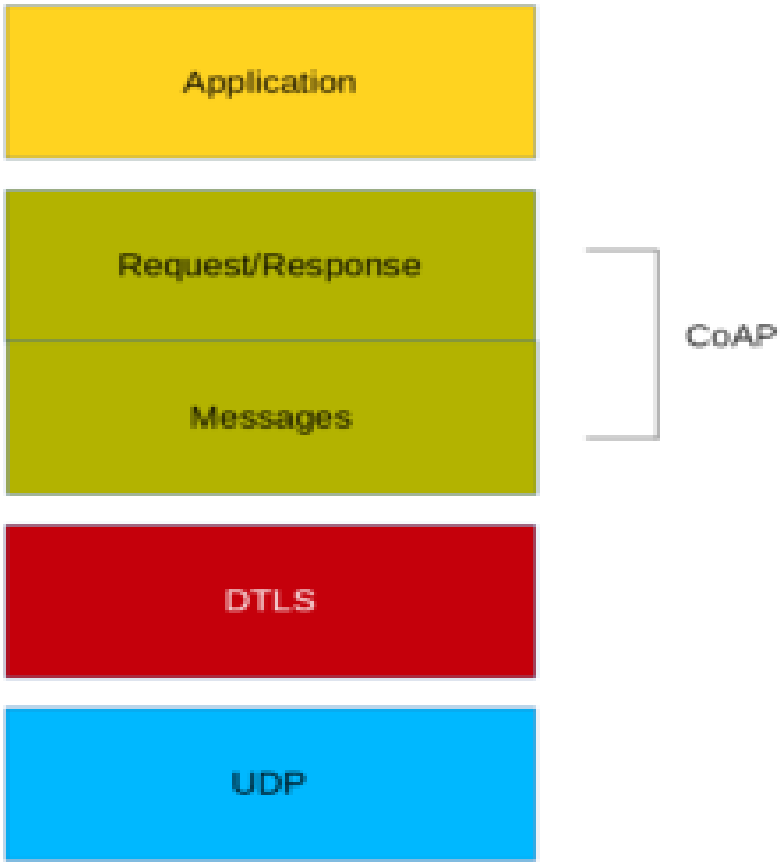
Cont…

- If the server **can't** answer to the request coming from the client **immediately(Separate Response)**.
- It sends an Acknowledge message with an **empty response**.
- When 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 Security Aspects

- CoAP uses **UDP** to transport information. CoAP relies on UDP security aspects to protect the information.
- As HTTP uses TLS over TCP, **CoAP uses Datagram TLS over UDP.** DTLS supports RSA, AES.
- In some constrained devices some of DTLS cipher suits may not be available.
- Some cipher suites introduces some **complexity** and constrained devices may not have resources enough to manage it.



CoAP Vs. MQTT

- MQTT uses a publisher -subscriber.
- MQTT uses a central broker to dispatch messages coming from the publisher to the clients.
- MQTT is an event-oriented protocol.
- MQTT uses Asynchronous messaging.

- CoAP uses a request-response paradigm
- CoAP is essentially a one-to one protocol very similar to the HTTP protocol.
- While CoAP is more suitable for state transfer.
- CoAP uses Asynchronous & Synchronous messaging

Thanks!