
# Sunbeam Institute of Information Technology
# Pune and Karad

## Module – Internet of Things (IoT)

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

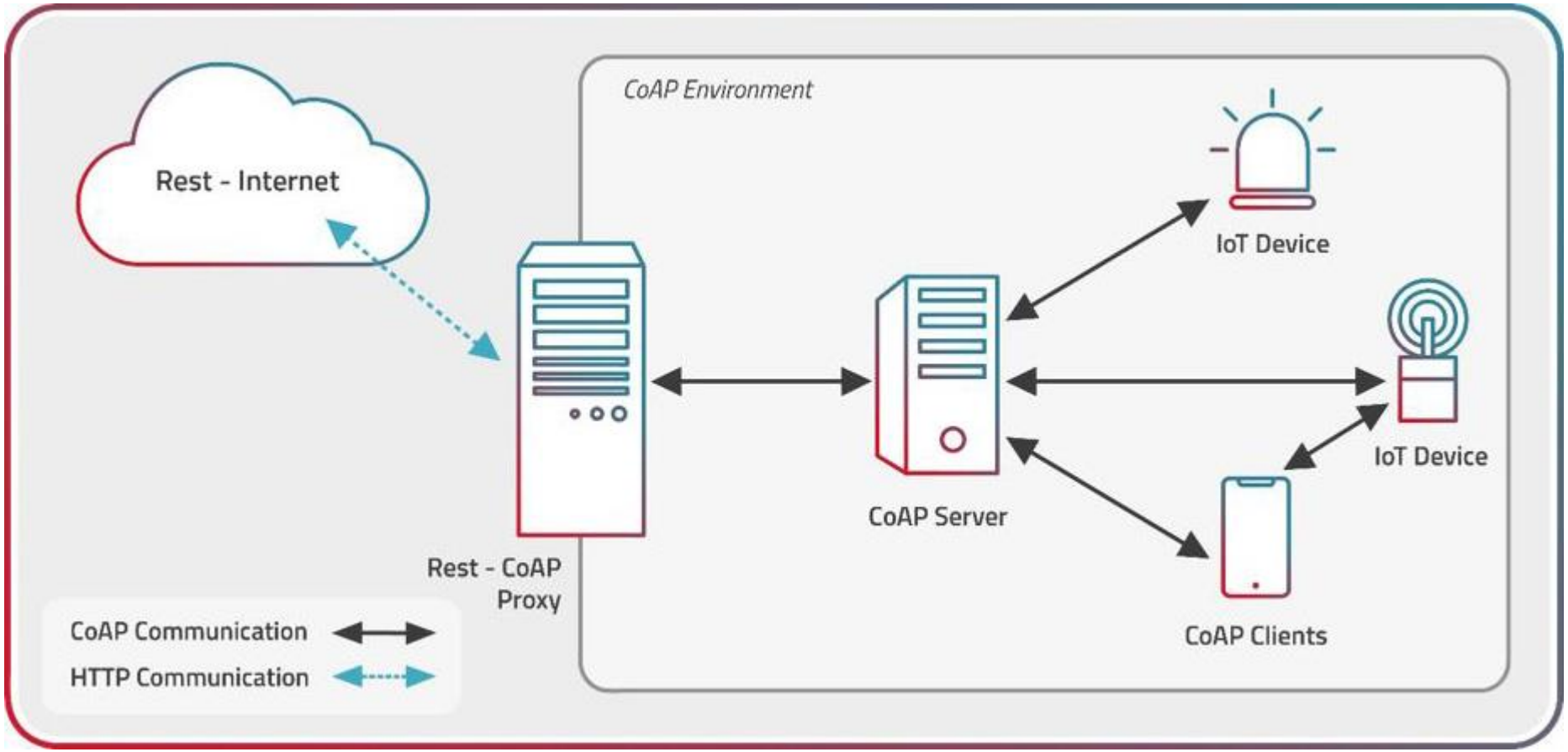
# CoAP

- **Constrained Application Protocol**
- specialized internet application protocol for **constrained devices**.
- designed to allow **small, low-power devices** to join the Internet of Things (IoT)
- operates over **UDP** and provides a **request/response**
- CoAP is also **highly reliable** for message delivery, even in cases of limited network connectivity or device power.
- web transfer protocol for use with **constrained nodes and constrained networks** in the Internet of Things.
- designed to enable simple, constrained devices to join the IoT even through constrained networks with **low bandwidth, low availability, high congestion and low power consumption**.
- used for machine-to-machine (M2M) applications
- The interaction model of CoAP is similar to the **client/server model of HTTP.**
- A CoAP request is equivalent to that of HTTP and is sent by a client to request an action (using a Method Code) on a resource (identified by a URI) on a server.
- The server then sends a response with a Response Code; this response may include a resource representation.
- CoAP deals with these interchanges **asynchronously over a datagram-oriented transport such as UDP**.

# CoAP Terminologies

- Endpoint - An entity participating in the CoAP protocol.
- Sender - The originating endpoint of a message. (source endpoint)
- Recipient - The destination endpoint of a message. (destination endpoint)
- Client - The originating endpoint of a request; the destination endpoint of a response.
- Server - The destination endpoint of a request; the originating endpoint of a response.
- Origin Server - The server on which a given resource resides or is to be created.
- Intermediary - A CoAP endpoint that acts both as a server and as a client.
- Proxy - An intermediary that mainly is concerned with
  - forwarding requests and relaying back responses
  - possibly performing caching
  - namespace translation
  - protocol translation in the process
- CoAP-to-CoAP Proxy
  - A proxy that maps from a CoAP request to a CoAP request
- Cross-Proxy
  - is a proxy that translates between different protocols, such as a CoAP-to-HTTP proxy or an HTTP-to-CoAP proxy.

# CoAP Network



# Request/Response Model

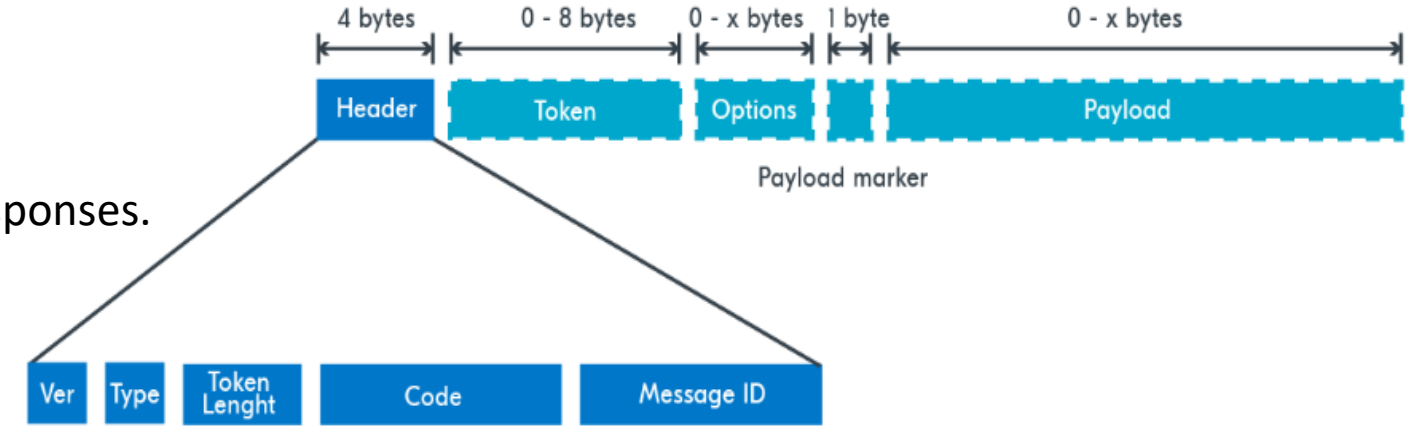
- CoAP request and response semantics are carried in CoAP messages, which include either a Method Code or Response Code, respectively.
- Optional (or default) request and response information, such as the URI and payload media type are carried as CoAP options.
- requests can be carried in Confirmable and Non-confirmable messages
- responses can be carried in these as well as in Acknowledgement messages
- If the server is not able to respond immediately to a request carried in a Confirmable message, it simply responds with an Empty Acknowledgement message so that the client can stop retransmitting the request.
- CoAP makes use of GET, PUT, POST, and DELETE methods in a similar manner to HTTP

# CoAP Message

- It starts with a fixed 4-byte header, which contains
- CoAP version – This is set to 1, other values are reserved for future versions.
- Type – Indicates if the message is confirmable, non-confirmable, an acknowledgment or reset.
- Token length – Indicates the length of the variable-length token field.
- Code – Split into two parts, class (0-7) and detail (0-31), where class indicates request, success response or error response and detail gives additional information to the class.
- Message ID – Unique ID in network byte order, used to detect duplicates and optionally for reliability.

# CoAP Message Format

- CoAP is based on the exchange of compact messages
- each CoAP message occupies the data section of one UDP datagram
- CoAP messages are encoded in a simple binary format.
- CoAP messages uses a short fixed-length binary header (4 bytes) that may be followed by compact binary options and a payload.
- Fields in header:
  - Version
  - Type
  - Token length
  - Code
  - Message ID
- This message format is shared by requests and responses.




*CoAP message format*

# CoAP Messaging Model

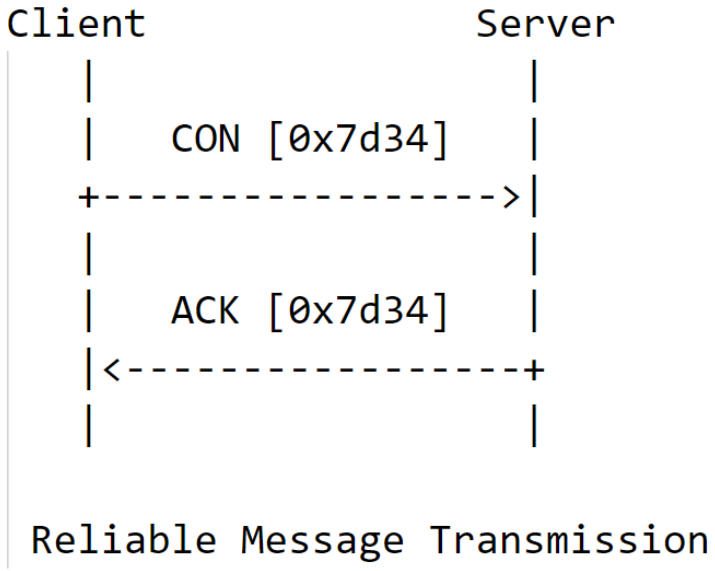
- CoAP employs structured message formats to facilitate efficient communication between IoT devices and applications.
- These formats encapsulate the necessary information for transmitting requests, responses, and control messages.
- There are four primary types of CoAP messages:

- **ACK (Acknowledgment) Message:**
  - ACK messages are sent in response to CON messages to acknowledge their receipt.
  - They indicate that the recipient has successfully received the message and is processing it.

- **RST (Reset) Message:**
  - RST messages are sent to cancel a pending CON message that hasn't yet been acknowledged.
  - They are typically used when a receiver cannot or chooses not to process a pending message.

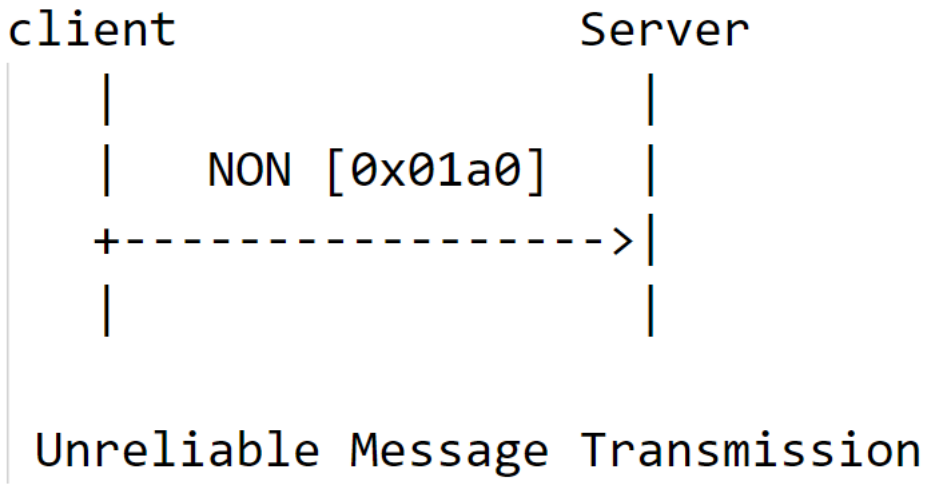
# CoAP Messaging Model

- **CON (Confirmable) Message:**
  - CON messages are used for reliable communication, ensuring that the recipient sends an acknowledgment.
  - They contain a CoAP request or response and are sent by a client or server, respectively.
  - The sender expects an acknowledgment (ACK) from the recipient and retransmits the message until the ACK is received.



# CoAP Messaging Model

- **NON (Non-Confirmable) Message:**
  - NON messages are used for faster communication without requiring acknowledgment.
  - They are similar to CON messages but don't demand an ACK.
  - NON messages are suitable for scenarios where real-time communication is prioritized over reliability.




# Thank you!!!

Devendra Dhande

devendra.dhande@sunbeaminfo.com