



# SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai



## Department of AI & DS

### Course Name – Internet of Things & AI III Year / V Semester

#### CONNECTIVITY TECHNOLOGIES AND COMMUNICATION PROTOCOLS



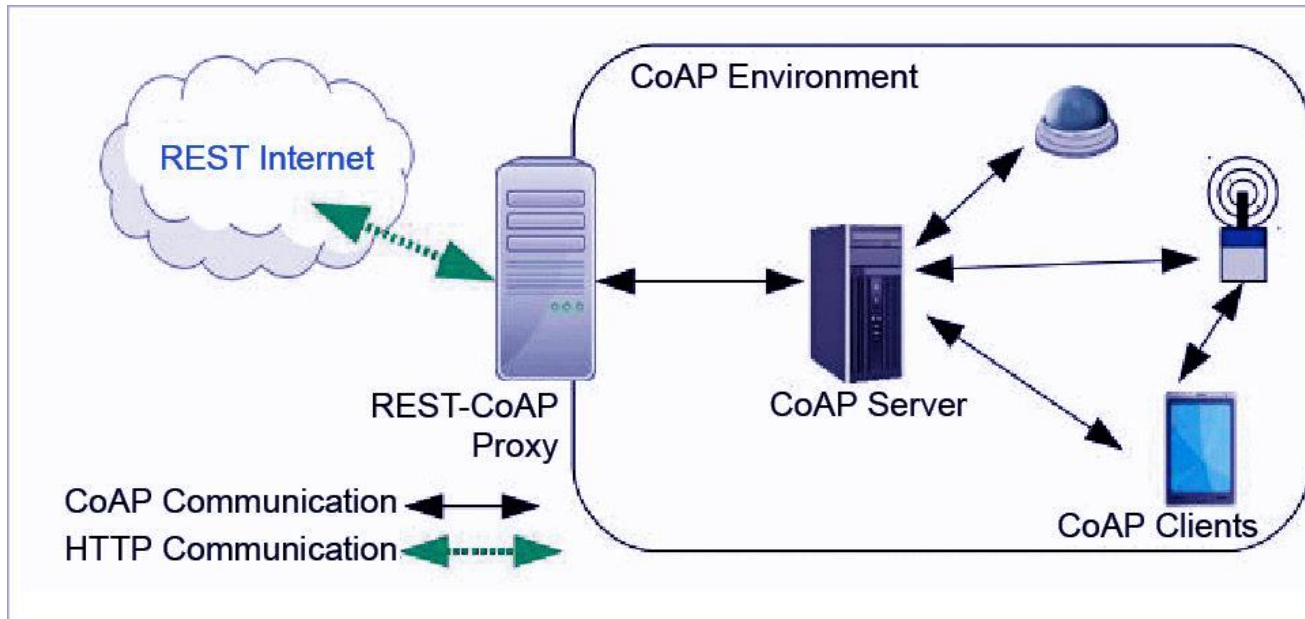
# CoAP



- Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things. .
- CoAP (Constrained Application Protocol) is a session layer protocol that provides the RESTful (HTTP) interface between HTTP client and server.
- It is designed to use devices on the same constrained network between devices and general nodes on the Internet.
- CoAP is designed to enable simple, constrained devices to join the IoT even through constrained networks with low bandwidth and low availability.
- This protocol is specially built for IoT systems primarily based on HTTP protocols.
- The protocol is designed for M2M and iot applications like smart energy and

This network is used within the limited network or in a constrained environment.

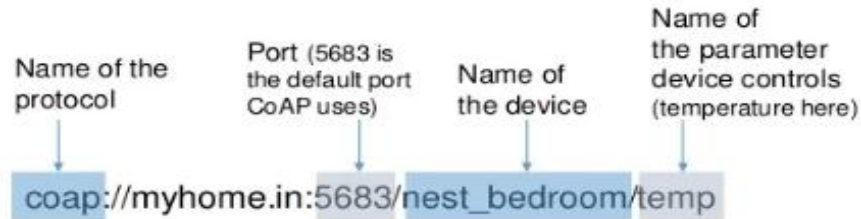
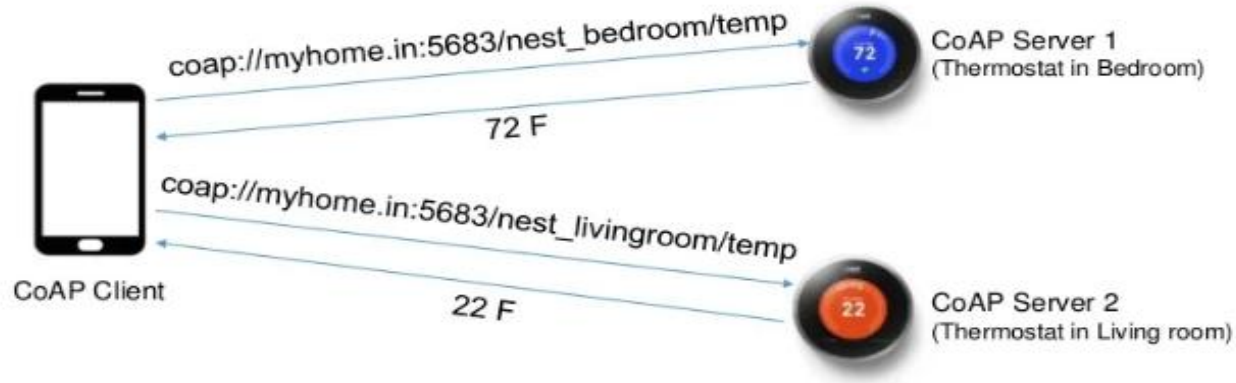
- The whole architecture of CoAP consists of CoAP client, CoAP server, REST CoAP proxy, and REST internet.





- The data is sent from CoAP clients (such as smartphones, RFID sensors, etc.) to the CoAP server and the same message is routed to REST CoAP proxy
- The REST CoAP proxy interacts outside the CoAP environment and uploads the data over REST internet.

## CoAP – Request Response





# CoAP Methods

- GET
- POST
- PUT
- DELETE
- OBSERVE (Not present in Http, New in CoAP)



## CoAP Message Types

### CON / Confirmable message

A confirmable message requires a response, either a positive acknowledgement or a negative acknowledgement. In case acknowledgement is not received, retransmissions are made until all attempts are exhausted.

### NON / Non-confirmable message

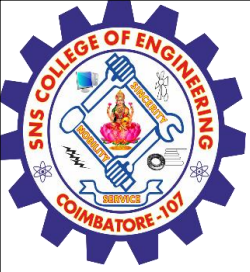
A non-confirmable request is used for unreliable transmission (like a request for a sensor measurement made in periodic basis. Even if one value is missed, there is not too much impact). Such a message is not generally acknowledged.

### ACK / Acknowledgement

Sent to acknowledge a confirmable (CON) message.

### RST / Reset

This represents a negative acknowledgement and means "Reset". It generally indicates, some kind of failure (like unable to parse received data)



<https://www.wallarm.com/what/coap-protocol-definition>





# XMPP



- XMPP is the Extensible Messaging and Presence Protocol, a set of open technologies for instant messaging, presence, multi-party chat, voice and video calls
- It's protocol for streaming XML elements over a network in order to exchange messages and presence information in close to real time.
- This protocol is mostly used by instant messaging applications like WhatsApp.



# XMPP

Let's dive into each character of word XMPP:

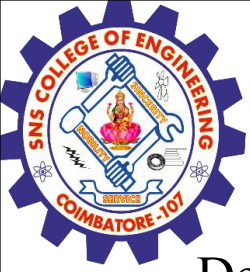
- X : It means eXtensible. XMPP is a open source project which can be changed or extended according to the need.
- M : XMPP is designed for sending messages in real time. It has very efficient push mechanism compared to other protocols.
- P : It determines whether you are online/offline/busy. It indicates the state.
- P : XMPP is a protocol, that is, a set of standards that allow systems to communicate with each other.



# XMPP

These are the basic requirements of any Instant Messenger which are fulfilled by XMPP:

- Send and receive messages with other users.
- Check and share presence status
- Manage subscriptions to and from other users.
- Manage contact list
- Block communications(receive message, sharing presence status, etc) to specific users.



# XMPP

Decentralised –

- XMPP is based on client-server architecture, i.e. clients don't communicate directly, they do it with the help of server as intermediary
- It is decentralised means there is no centralised XMPP server just like email, anyone can run their own XMPP server.



# XMPP

Each XMPP client is identified by JID (Jabber ID).

```
#JID
{
  user,
  server,
  resource
}
```

For example, I'm a whatsapp user and I'm identified by my mobile number, so

```
user = "8767898790"
server = "whatsapp.com"
resource = "mobile"

JID : "8767898790@whatsapp.com/mobile"
```



# XMPP

## CoAP Message Types

### CON / Confirmable message

A confirmable message requires a response, either a positive acknowledgement or a negative acknowledgement. In case acknowledgement is not received, retransmissions are made until all attempts are exhausted.

### NON / Non-confirmable message

A non-confirmable request is used for unreliable transmission (like a request for a sensor measurement made in periodic basis. Even if one value is missed, there is not too much impact). Such a message is not generally acknowledged.

### ACK / Acknowledgement

Sent to acknowledge a confirmable (CON) message.

### RST / Reset

This represents a negative acknowledgement and means "Reset". It generally indicates, some kind of failure (like unable to parse received data)





# AMQP

Advanced Message Queuing Protocol (AMQP) is an open source published standard for asynchronous messaging by wire.

- The protocol is used in client/server messaging and in IoT device management.
- It is a protocol that is used for communication between applications. It is a lightweight, protocol which supports the applications for transfer of data.

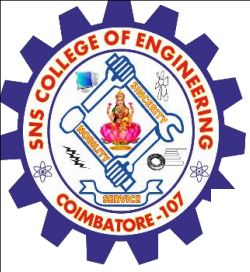


# AMQP

## Why Messaging ?

- Transfer data from point A to point B.
- Asynchronous.
- Decouple publishers and consumers.
- Queuing for later delivery.





# AMQP

## How does AMQP solve the problem?

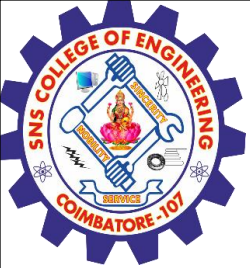
- AMQP is a wire level protocol and not an API.
  - JMS is an API.
  - Just like HTTP is for Internet, AMQP is for messaging.
- When a protocol is specified at the wire-level and published, most technologies can use it.
- Compare this to an API, where the actual implementation is specific to the platform.



# AMQP

AMQP allows for various guaranteed messaging modes specifying a message be sent:

- At-most-once (sent one time with the possibility of being missed).
- At-least-once (guaranteeing delivery with the possibility of duplicated messages).
- Exactly-once (guaranteeing a one-time only delivery).

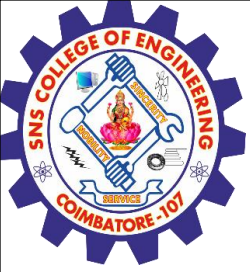


# HOW AMQP WORKS

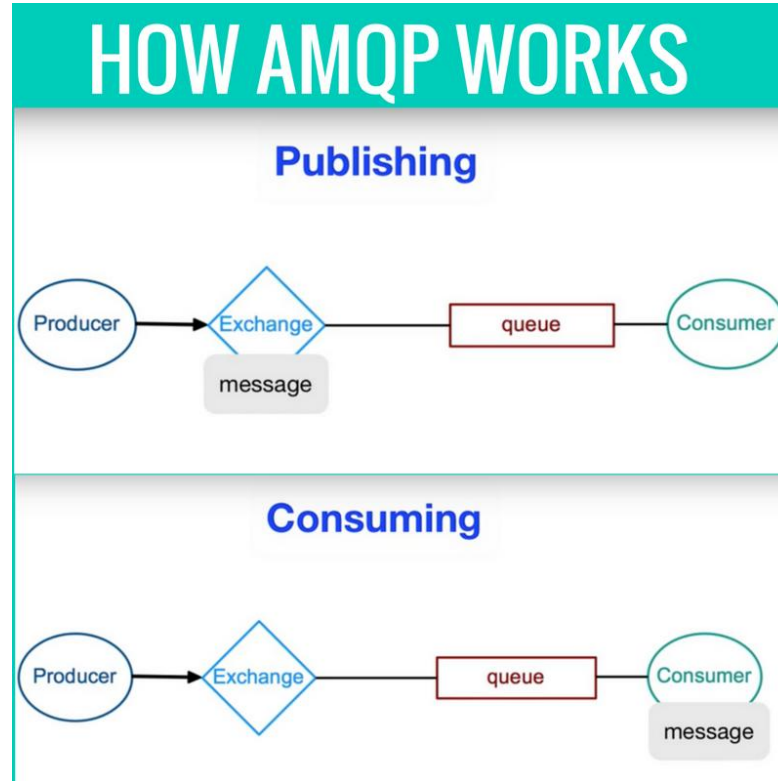


AMQP deals with publishers and consumers.

- The publishers produce the messages, the consumers pick them up and process them.
- It's the job of the message broker (such as RabbitMQ) to ensure that the messages from a publisher go to the right consumers.
- In order to do that, the broker uses two key components :-
  - 1.Exchanges
  - 2.Queues.



# HOW AMQP WORKS





# HOW AMQP WORKS

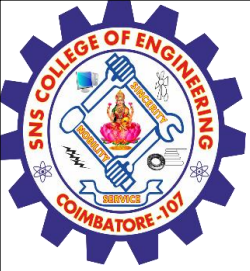


- A publisher sends messages to a named exchange and a consumer pulls messages from a queue or the queue pushes them to the consumer depending on the configuration.



## Message Queuing Telemetry Transport (MQTT)

- MQTT stands for **Message Queuing Telemetry Transport**.
- MQTT is a machine to machine internet of things connectivity protocol.
- It is an extremely lightweight and publish-subscribe messaging transport protocol.
- This protocol is useful for the connection with the remote location where the bandwidth is a premium.
- These characteristics make it useful in various situations, including constant environment such as for communication machine to machine and internet of things contexts.



## Message Queuing Telemetry Transport (MQTT)



- It is a publish and subscribe system where we can publish and receive the messages as a client.
- It makes it easy for communication between multiple devices.
- It is a simple messaging protocol designed for the constrained devices and with low bandwidth, so it's a perfect solution for the internet of things applications.



# Characteristics of MQTT

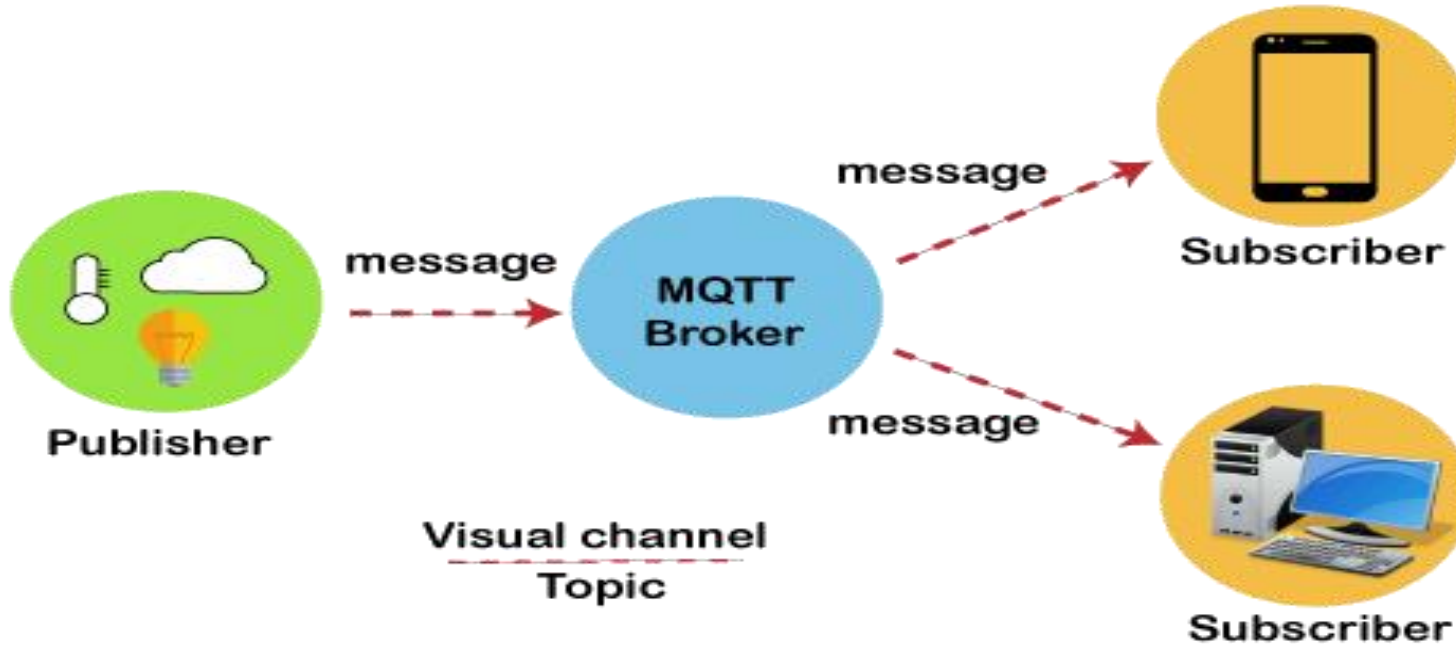
- The MQTT has some unique features which are hardly found in other protocols. Some of the features of an MQTT are given below:
- It is a machine to machine protocol, i.e., it provides communication between the devices.
- It is designed as a simple and lightweight messaging protocol that uses a publish/subscribe system to exchange the information between the client and the server.
- It does not require that both the client and the server establish a connection at the same time.





- It provides faster data transmission, like how WhatsApp/messenger provides a faster delivery. It's a real-time messaging protocol.
- It allows the clients to subscribe to the narrow selection of topics so that they can receive the information they are looking for.

## MQTT Architecture





- **Client**

- In MQTT, the subscriber and publisher are the two roles of a client. The clients subscribe to the topics to publish and receive messages. In simple words, we can say that if any program or device uses an MQTT, then that device is referred to as a client.
- **Message**
- The message is the data that is carried out by the protocol across the network for the application.
- **Publish:** When the client sends the data to the server, then we call this operation as a publish.
- **Subscribe:** When the client receives the data from the server, then we call this operation a subscription.