SNS COLLEGE OF TECHNOLOGY

Coimbatore-35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECT213- IoT SYSTEM ARCHITECTURE

II ECE / IV SEMESTER

UNIT 4 – CLOUD PLATFORMS FOR IoT

TOPIC 4 –Study of IoT Cloud Platforms-MQTT

MQTT

- MQTT (Message Queuing Telemetry Transport) is a lightweight, publish-subscribe based messaging protocol designed for resource-constrained devices and low-bandwidth, high-latency, or unreliable networks.

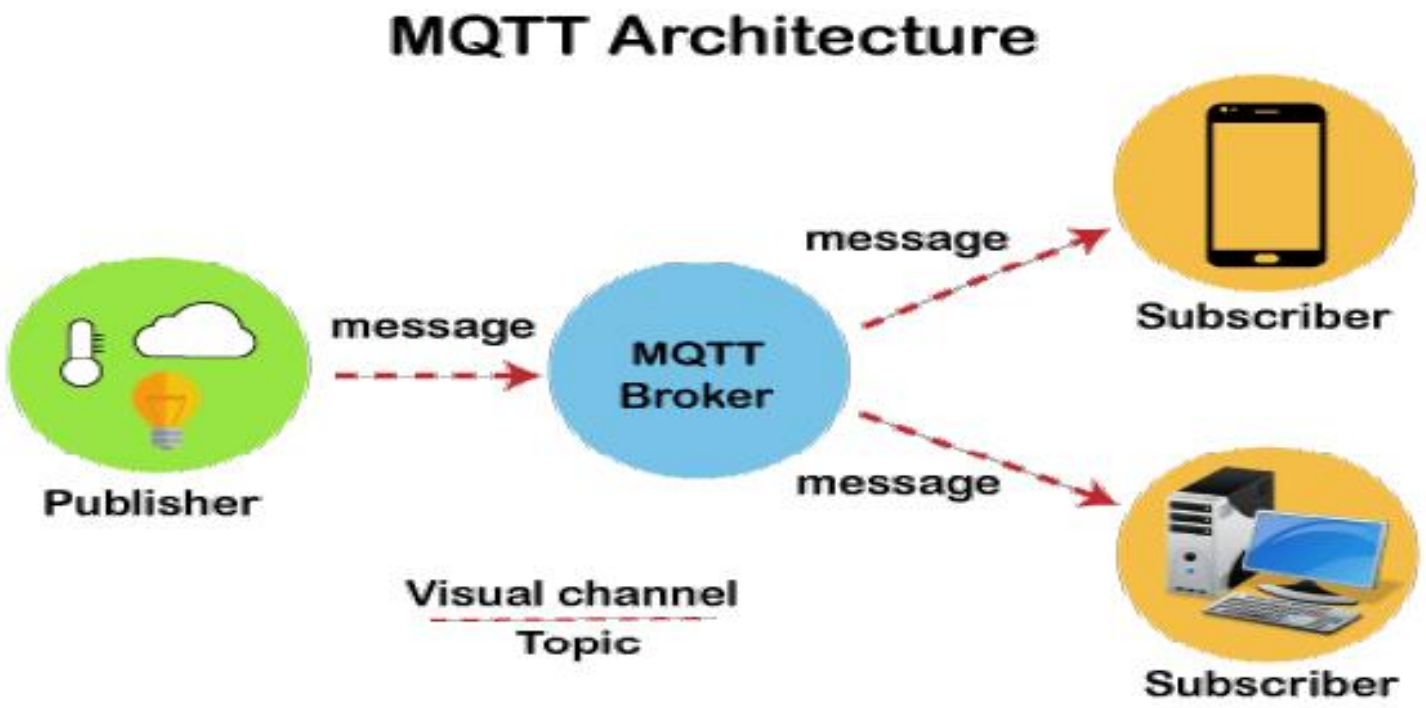
- It is widely used in Internet of Things (IoT) applications, providing efficient communication between sensors, actuators, and other devices.

- 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.
- 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.

CHARACTERISTICS OF MQTT

- It is a machine to machine protocol, i.e., it provides communication Between the devices.
- 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'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.

MQTT ARCHITECTURE



WHY IS MQTT THE BEST PROTOCOL FOR IOT?

MQTT has emerged as one of the best **IoT protocols** due to its unique features and capabilities tailored to the specific needs of IoT systems. Some of the key reasons include:

- **Lightweight:** IoT devices are often constrained in terms of processing power, memory, and energy consumption.
  - MQTT's minimal overhead and small packet size make it ideal for these devices, as it consumes fewer resources, enabling efficient communication even with limited capabilities.
- **Reliable:** IoT networks can experience high latency or unstable connections.
  - MQTT's support for different QoS levels, session awareness, and persistent connections ensures reliable message delivery even in challenging conditions, making it well-suited for IoT applications.

WHY IS MQTT THE BEST PROTOCOL FOR IOT?

- **Secure communications:** Security is crucial in IoT networks as they often transmit sensitive data. MQTT supports Transport Layer Security (TLS) and Secure Sockets Layer (SSL) encryption, ensuring data confidentiality during transmission.
- Additionally,provides **authentication** and **authorization** mechanisms through username/password credentials or client certificates, safeguarding access to the network and its resources.
- **Bi-directionality: MQTT's publish-subscribe model** allows for seamless bi-directional communication between devices.
- Clients can both publish messages to topics and subscribe to receive messages on specific topics, enabling effective data exchange in diverse IoT ecosystems without direct coupling between devices.
- This model also simplifies the integration of new devices, ensuring easy scalability.
- **Continuous, stateful sessions:** MQTT allows clients to maintain stateful sessions with the broker, enabling the system to remember subscriptions and undelivered messages even after disconnection.
- Clients can also specify a keep-alive interval during connection, which prompts the broker to periodically check the connection status.
- If the connection is lost, the broker stores undelivered messages (depending on the QoS level) and attempts to deliver them when the client reconnects.
- This feature ensures reliable communication and reduces the risk of data loss due to intermittent connectivity.

WHY IS MQTT THE BEST PROTOCOL FOR IOT?

- **Large-scale IoT device support:** IoT systems often involve a large number of devices, requiring a protocol that can handle massive-scale deployments.
- MQTT's lightweight nature, low bandwidth consumption, and efficient use of resources make it well-suited for large-scale IoT applications.
- The publish-subscribe pattern allows MQTT to scale effectively, as it decouples sender and receiver, reducing network traffic and resource usage.
- Furthermore, the protocol's support for different QoS levels allows customization of message delivery based on the application's requirements, ensuring optimal performance in various scenarios.
- **Language support:** IoT systems often include devices and applications developed using various programming languages.
- MQTT's broad language support enables easy integration with multiple platforms and technologies, fostering seamless communication and interoperability in diverse IoT ecosystems.

HOW DOES MQTT WORK?

To understand how MQTT works, listen the concepts of MQTT Client, MQTT Broker, Publish-Subscribe mode, Topic, and QoS:

MQTT Client

Any application or device running the **MQTT client library** is an MQTT client.

For example, an instant messaging app that uses MQTT is a client, various sensors that use MQTT to report data are a client, and various **MQTT testing tools** are also a client.

In MQTT, the client performs two operations:

1.Publish:

When the client sends the data to the server, then we call this

Operation as a publish.

2.Subscribe:

When the client receives the data from the server, then we call

This operation as subscription.

HOW DOES MQTT WORK?

MQTT Broker

The MQTT Broker handles client connection, disconnection, subscription, and unsubscription requests, and routing messages.

A powerful MQTT broker can support massive connections and million-level message throughput, helping IoT service providers focus on business and quickly create a reliable MQTT application.

Publish–subscribe pattern

The publish-subscribe pattern differs from the client-server pattern in that it separates the client that sends messages (publisher) from the client that receives messages (subscriber).

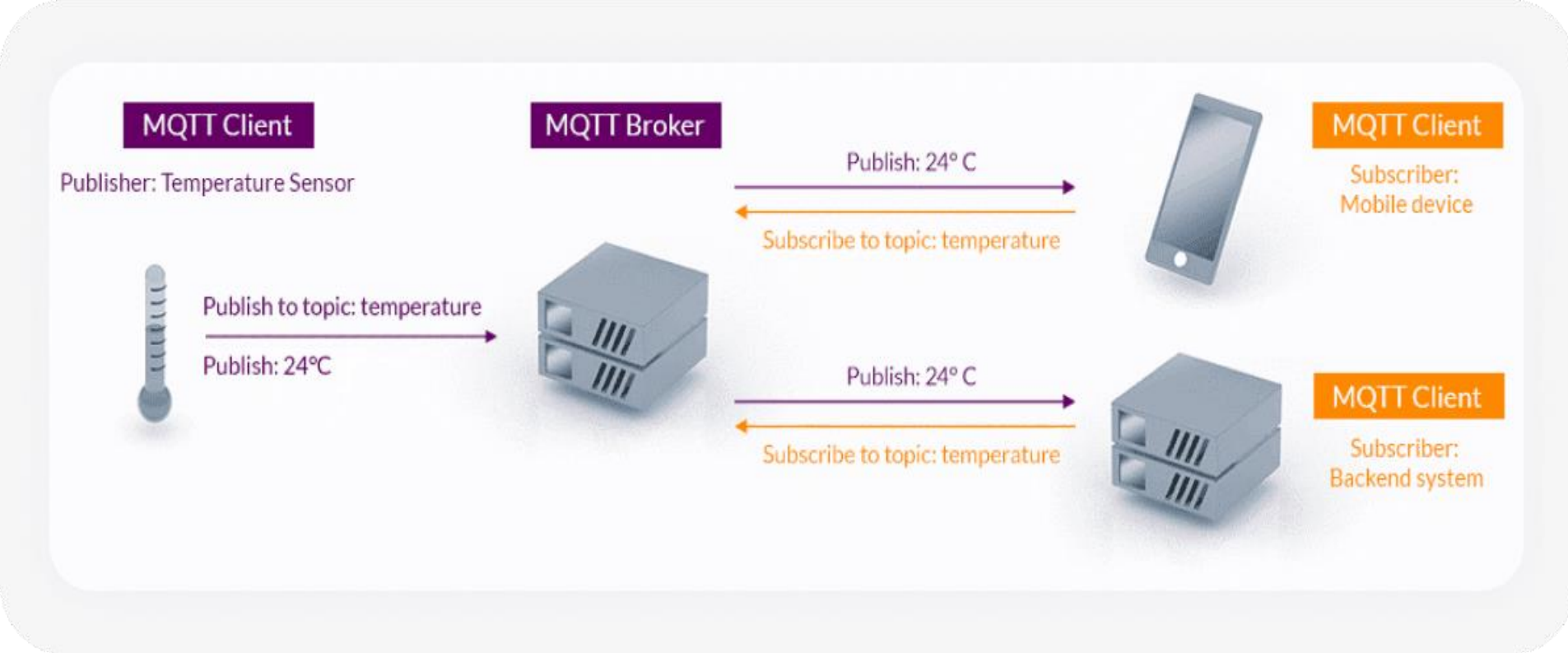
•Publishers and subscribers do not need to establish a direct connection, and the MQTT Broker is responsible for routing and distributing all messages.

•The following diagram shows the MQTT publish/subscribe process.

•The temperature sensor connects to the MQTT server as a client and publishes temperature data to a topic (e.g., Temperature)

•The server receives the message and forwards it to the client subscribed to the Temperature topic.

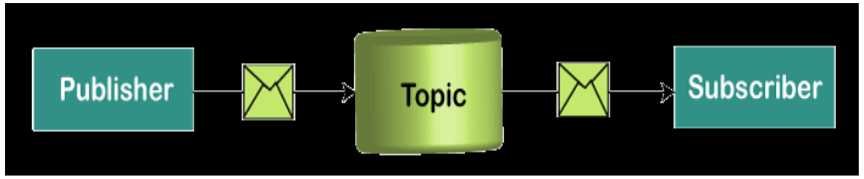
HOW DOES MQTT WORK?



HOW DOES MQTT WORK?

Topic

In MQTT, Topic refers to a UTF-8 string that filters messages for a connected client.
A topic consists of one or more levels separated by a forward slash (topic level separator).
In comparison to a message queue, MQTT topics are very lightweight.
The client does not need to create the desired topic before they publish or subscribe to it.
The broker accepts each valid topic without any prior initialization.



The MQTT protocol routes messages based on topic. The topic distinguishes the hierarchy by slash /, which is similar to URL paths, for example:
chat/room/1sensor/10/temperaturesensor/+/temperature
MQTT topic support the following wildcards: + and #.
•+: indicates a single level of wildcards, such as a/+ matching a/x or a/y.
•#: indicates multiple levels of wildcards, such as a/# matching a/x, a/b/c/d.

HOW DOES MQTT WORK?

Quality of Service (QoS)

MQTT provides three kinds of Quality of Service and guarantees messaging reliability in different network environments.

- QoS 0: The message is delivered at most once. If the client is not available currently, it will lose this message. (fire and forget)
- QoS 1: The message is delivered at least once. (acknowledged delivery)
- QoS 2: The message is delivered only once. (assured delivery)