IoT Protocols – MQTT

1. Introduction to MQTT

- MQTT stands for **Message Queuing Telemetry Transport**.
- A lightweight messaging protocol for clients that need a small code footprint and are connected to unreliable networks or networks with limited bandwidth.
- Primarily used for machine-to-machine (M2M) communication and Internet of Things connections.
- Implements the **publish/subscribe** model using clients and a broker.

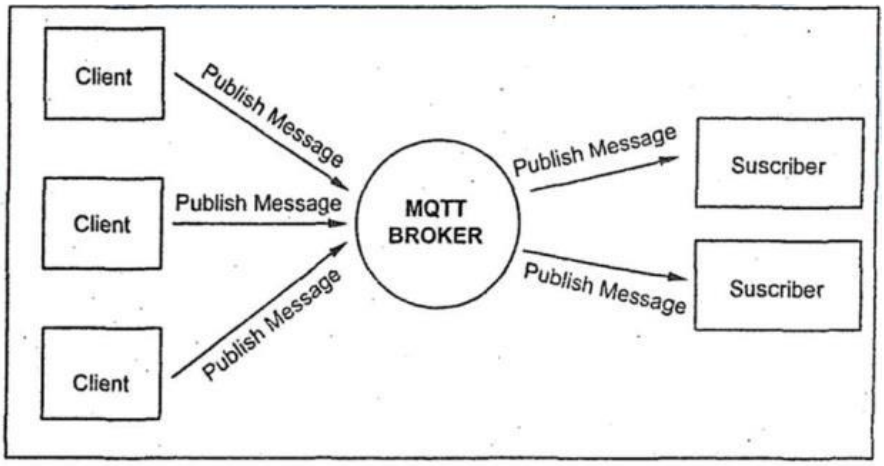

Fig 11.19 MQTT architecture

Figure 3.6 MQTT architecture

2. MQTT Components

2.1 MQTT Client

- Any device, from a server to a microcontroller, that runs an MQTT library and communicates using MQTT over a network.
- A client that sends messages acts as a **publisher**; one that receives messages acts as a **subscriber/receiver**.

2.2 MQTT Broker

The backend system that coordinates messages between clients. Responsibilities:

- Receiving and filtering messages
- Identifying clients subscribed to each message and sending them the messages
- Authorising and authenticating MQTT clients
- Passing messages to other systems for further analysis
- Handling missed messages and client sessions

2.3 MQTT Connection

- A client starts communication by sending a **CONNECT** message to the broker; the broker confirms with a **CONNACK** message.

- Clients never connect to each other – only to the broker.

3. Advantages of MQTT

Advantage	Explanation
Lightweight and efficient	Needs minimal resources on the IoT device, so it can run on small microcontrollers; message headers are small, optimising network bandwidth
Scalable	Needs a minimal amount of code that consumes very little power in operation
Secure	Makes it easy to encrypt messages and authenticate devices and users using modern authentication protocols
Well-supported	Several languages such as Python have extensive MQTT support, so developers can implement it quickly with minimal coding on any application