IOT PROTOCOLS

MQTT

MQTT stands, for Message Queuing Telemetry Transport. It is a lightweight messaging protocol for use in cases where clients need a small code footprint and are connected to unreliable networks or networks with limited bandwidth resources.

- A MQTT is primarily used for machine-to-machine (M2M) communication or Internet of Things types of connections.
- MQTT implements the publish/subscribe model by defining the clients and brokers as below:

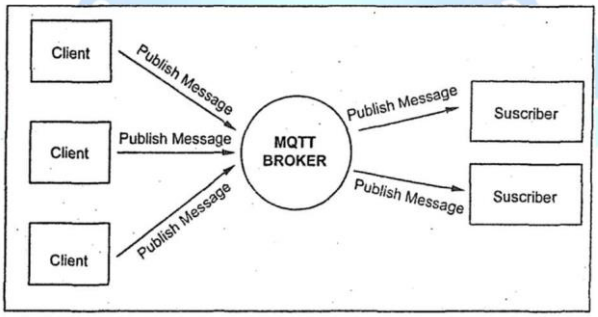

Fig 11.19 MQTT architecture

(i) MQTT client

- An MQTT client is any device from a server to a microcontroller that runs on an MQTT library. If the client is sending messages, it acts as a publisher, and if it is receiving messages, it acts as a receiver.
- Basically, any device that communicates using MQTT over a network can be called an MQTT client device.

(ii) MQTT broker

- The MQTT broker is the backend system which coordinates messages between the different clients. Responsibilities of the broker include receiving and filtering messages, identifying clients subscribed to each message, and sending them the messages. It is also responsible for other tasks such as:
    - ❖ Authorizing and authenticating MQTT clients.
    - ❖ Passing messages to other systems for further analysis.

❖ Handling missed messages and client sessions.

(iii) MQTT connection

- Clients and brokers begin communicating by using an MQTT connection. Clients initiate the connection by sending a CONNECT message to the MQTT broker. The broker confirms that a connection has been established by responding with a CONNACK message. Clients never connect with each other, they connects only with the broker.

Advantages

The advantages of MQTT are,

(i) Lightweight and efficient

MQTT implementation on the loT device requires minimal resources, so that it can even be used on small microcontrollers. MQTT message headers are also small so that network bandwidth can be optimized.

(ii) Scalable

MQTT implementation requires a minimal amount of code that consumes very little power in operations.

(iii) Secure

MQTT makes it easy for developers to encrypt messages and authenticate devices and users using modem authentication, protocols.

(iv) Well-supported

Several languages like Python have extensive support for MQTT protocol implementation. Hence, developers can quickly implement it with a minimal coding on any type of application.