MQTT Protocol (Publish/Subscribe, Topics and QoS)

1. Principle: Publish/Subscribe

In traditional networking a client requests data directly from a server. MQTT instead decouples the sender (publisher) from the receiver (subscriber) using a message broker, which filters incoming messages and distributes them to the right subscribers.

1.1 Three Kinds of Decoupling

Decoupling	Meaning
Space	Publisher and subscriber do not know each other's network location (no IP addresses or ports exchanged)
Time	They need not be connected at the same time
Synchronisation	Publishing and subscribing do not block each other; a subscriber need not wait for the publisher

2. MQTT Components

- **MQTT client:** any device, from a server to a microcontroller, running an MQTT library; a publisher when sending and a subscriber when receiving.
- **MQTT broker (server):** the backend that coordinates messages. Responsibilities:
  - receive and filter messages
  - identify subscribed clients and deliver to them
  - authenticate and authorise clients
  - optionally pass messages on for analysis
  - handle missed messages and client sessions
  - reject messages that break the protocol specification
- **MQTT connection:** the client sends CONNECT; the broker confirms with CONNACK. Both need a TCP/IP stack (WebSockets are also supported). Clients never connect directly to each other, only to the broker.

2.1 How MQTT Works

1. A client connects to the broker.
2. Once connected, it can publish, subscribe to topics, or both.
3. When the broker receives a message, it forwards it to all subscribers interested in that topic.

3. Topics

A topic is the keyword (or hierarchy) the broker uses to filter and route messages. Topics are hierarchical, like folders and files. Smart-home examples:

- ourhome/groundfloor/livingroom/light
- ourhome/firstfloor/kitchen/temperature

4. Publish and Subscribe

- **Publish:** a message contains a topic and a payload in bytes; the client chooses the format (text, binary, XML, JSON). A smart lamp may publish on livingroom/light.
- **Subscribe:** a client sends a SUBSCRIBE message with a unique packet identifier and a list of subscriptions. A smart-home phone app might subscribe to the wildcard +/light and count every "on" message.

5. Quality of Service (QoS)

QoS	Guarantee	Typical use
0	At most once: fire and forget, no acknowledgement	Frequent, non-critical sensor readings
1	At least once: acknowledged, duplicates possible	Commands where delivery matters and duplicates are tolerable
2	Exactly once: no duplicates, highest overhead	Billing, critical control commands

6. Other Useful Concepts

- **Retained message:** the broker stores the last message on a topic and delivers it to any new subscriber immediately.
- **Last Will and Testament (LWT):** a message the broker publishes automatically if a client disconnects unexpectedly, signalling device failure or offline status.
- **Keep-alive:** a periodic PING lets the broker detect a silent client.
- MQTT is lightweight (small header), works over low-bandwidth/high-latency links and uses little power, making it the most widely used IoT telemetry protocol compared with heavier ones such as HTTP.