Sunbeam Institute of Information Technology
Pune and Karad

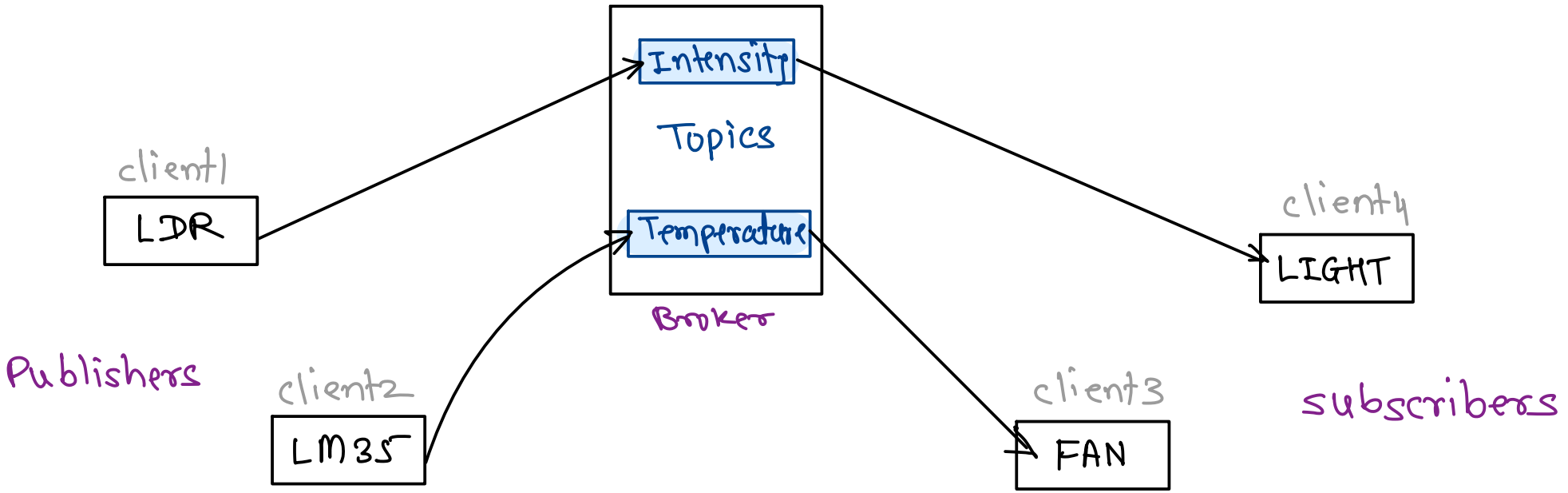
Module – Internet of Things (IoT)

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

MQTT – MQ Telementry Transport

- **MQTT** is a lightweight publish/subscribe messaging protocol.
- It is useful for use with low power sensors.
- MQTT protocol is based on the principle of publishing messages and subscribing to topics, or "pub/sub".
- Multiple clients connect to a broker and subscribe to topics that they are interested in.
- Clients also connect to the broker and publish messages to topics.
- The broker and MQTT act as a simple, common interface for everything to connect to.



MQTT – Terminologies

- **Client**
  - client is endpoint (devices which are going to send and receive data)
  - clients are of two types
    - **Publisher**
      - going to generate data or to send data
    - **Subscriber**
      - going to consume data or to receive data

home/room1/+ { home/room1/sensor1, home/room1/sensor2 }

home/#

- **Server**
  - intermediate between multiple clients
  - server is also known as broker

- **Topic**
  - string or path which will be used to differentiate your messages
  - to create multilevel topics strings are separated by '/'

+ – any single level

– all remaining levels

Home Automation

home/room1/sensor1

home/room1/sensor2

home/room2/sensor1

home/room2/sensor2

Office Automation

office/room1/sensor1

office/room1/sensor2

office/room2/sensor1

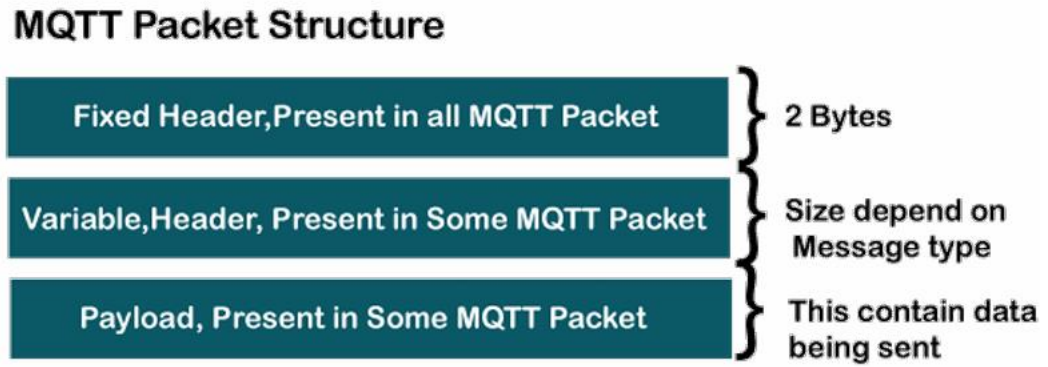
office/room2/sensor2

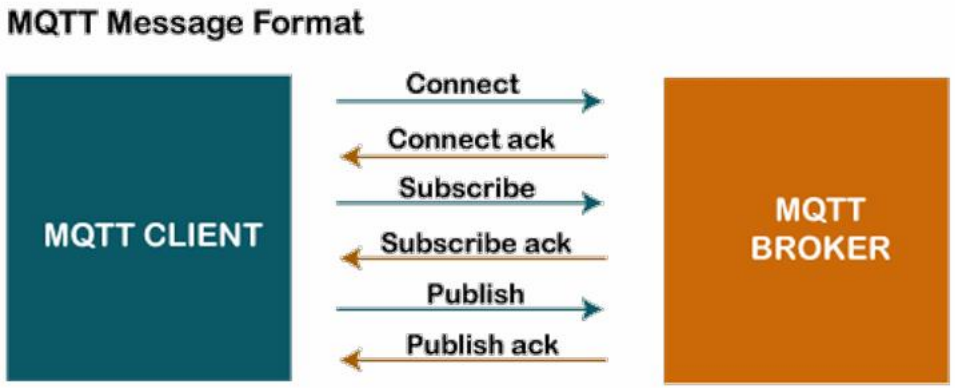
MQTT – Message

- message structure is lightweight
- **Message Structure**
  - fixed header (2 bytes)
  - variable header (optional)
  - variable payload (optional)

- depending on message type, message structure is different
- **Different Message structures**
  - fixed header
  - fixed header + variable header
  - fixed header + variable header + payload

- **Message types**
  - CONNECT
  - DISCONNECT
  - PUBLISH
  - SUBSCRIBE
  - CONNACK, DISCONNECTACK, PUBACK, SUBACK





MQTT – Methods

- **connect()**
  - this method is used to connect with broker
- **disconnect()**
  - this method is used to disconnect from broker
- **publish()**
  - this method is used to publish data on some topic
- **subscribe()**
  - this method is used to subscribe the topic

MQTT – Quality of Service (QoS)

- MQTT defines three levels of Quality of Service (QoS).
- The QoS defines how hard the broker/client will try to ensure that a message is received.
- Messages may be sent at any QoS level, and clients may attempt to subscribe to topics at any QoS level.

- Higher levels of QoS are more reliable, but involve higher latency and have higher bandwidth requirements.

- 0: The broker/client will deliver the message once, with no confirmation.
- 1: The broker/client will deliver the message at least once, with confirmation required.
- 2: The broker/client will deliver the message exactly once by using a four step handshake.

Server / broker : mosquitto

clients : publisher : mosquitto_pub
subscriber : mosquitto_sub

Location : C:\Program Files/mosquitto
↓
mosquitto.conf
(can do configurations of broker manually with this file)

mosquitto -c → to do the configurations
-d → to print debug messages

mosquitto_pub mosquitto_sub

-m : to specify message

-h : to specify host/ip
-p : to specify port = 1883 - plain MQTT
= 8883 - secure MQTT
-t : to specify topic
-d : to see the debug messages
-q : to specify qos level
-r : to specify whether to retain msg or not
-u : to specify user name
-P : to specify password

Thank you!!!

Devendra Dhande

devendra.dhande@sunbeaminfo.com