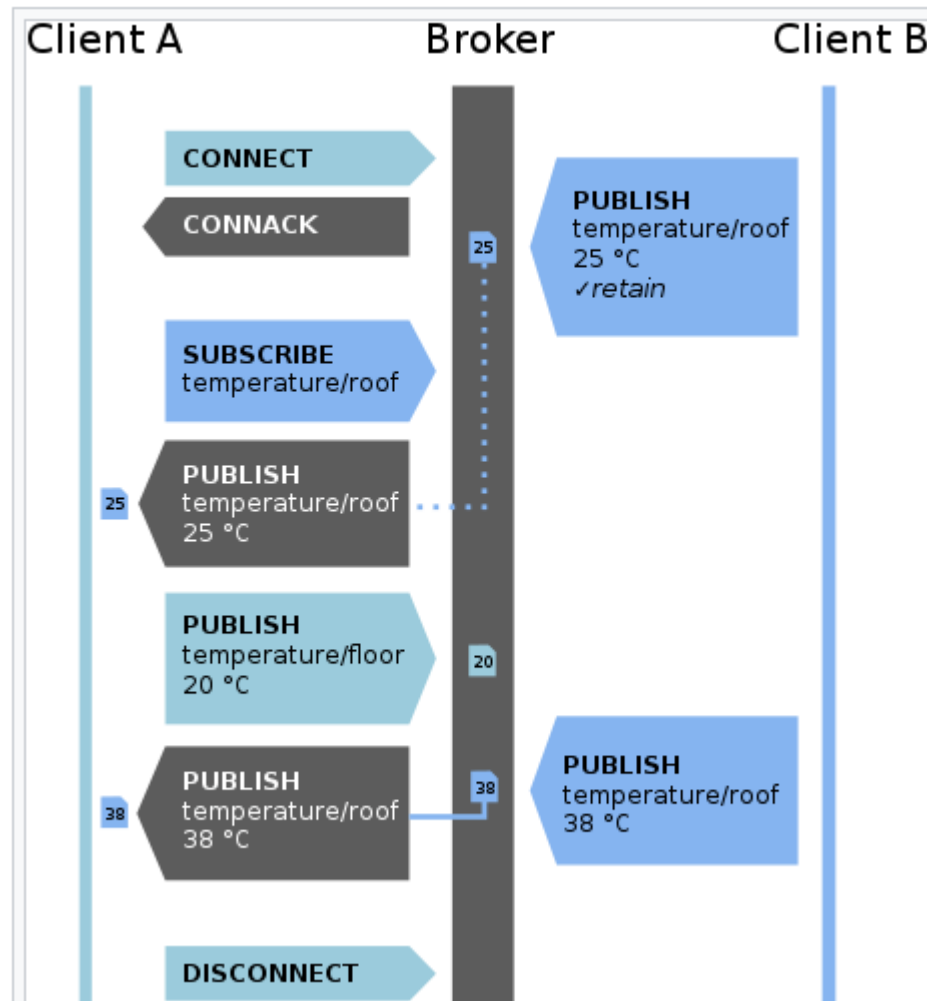


MQTT Application

MQTT

- Message Queuing Telemetry Transport (MQTT) is a lightweight, published-subscribe network protocol
- The protocol is in the application layer and usually run on TCP/IP
- It is designed for connections with remote location where a “small code footprint” is required

Publish-Subscribe protocol



Install Mosquitto

- Mosquitto software is MQTT broker
- To run the server, use the command:
`mosquitto -v`
- Client can publish or subscribe the message

Using Mosquitto

- To subscribe message:

```
mosquitto_sub -d -t pi_myqtt
```

- To publish message:

```
mosquitto_pub -d -t pi_myqtt -m  
“publishing hello”
```

Output

```
Client mosqsub/5170-raspberryp received PUBLISH (d0, q0, r0, m0,  
'pi_mqtt',  
    ... (16 bytes))  
publishing hello
```

Python code

- Application: Sending temperature every 100ms
- Install MQTT library

```
pip install paho-mqtt
```

- Publish the message
- Subscribe the message

Publish the message

```
import paho.mqtt.client as mqtt
from random import randrange, uniform
import time

# mqttBroker = "mqtt.eclipseprojects.io"
mqttBroker = "127.0.0.1"

client = mqtt.Client("Temperature_Inside")
client.connect(mqttBroker)

while True:
    randNumber = uniform(20.0, 21.0)
    client.publish("TEMPERATURE", randNumber)
    print("Just published " + str(randNumber) + " to topic TEMPERATURE")
    time.sleep(1)
```


Subscribe the message

```
import paho.mqtt.client as mqtt #import the client1
broker_address="127.0.0.1"
import time
def on_message(client, userdata, message):
    print("message received " ,str(message.payload.decode("utf-8")))
    print("message topic=",message.topic)
    print("message qos=",message.qos)
    print("message retain flag=",message.retain)

print("creating new instance")
client = mqtt.Client("P1") #create new instance
client.on_message=on_message
print("connecting to broker")
```

Subscribe the message

```
client.connect(broker_address) #connect to broker
client.loop_start()
print("Subscribing to topic","TEMPERATURE")
client.subscribe("TEMPERATURE")
print("Publishing message to topic","TEMPERATURE")
client.publish("TEMPERATURE","OFF")

time.sleep(10) # wait
client.loop_stop() #stop the loop
```

Output

```
creating new instance
connecting to broker
Subscribing to topic TEMPERATURE
Publishing message to topic TEMPERATURE
message received  OFF
message topic= TEMPERATURE
message qos= 0
message retain flag= 0
message received  hello
message topic= TEMPERATURE
message qos= 0
message retain flag= 0
message received  hello
message topic= TEMPERATURE
message qos= 0
message retain flag= 0
```

Questions?