PB_05_Kushagra Suryawanshi Batch B1.

E IOT LAB 6

Aim: To demonstrate MQTT/COAP/XMPP protocols using message broker to subscribe and sensor data publishing.

~~Obj~~ Theory:

Q.1. What is MQTT?
Message ~~Quing~~ Queuing Telemetry Transport. It is an ISO standard publish - subscribe - based messaging protocol. It works on top of TCP/IP protocol. It is designed for connections with remote locations where a 'small code footprint' is required or the network bandwidth is limited.

Q2. Principles of MQT:
A 1. simplicity
2. publish / subscribe messaging
3. zero administration
4. data agnostics.

Q3. MQTT libraries.

LIBRARY	DEVELOPER	TYPE
• Adafruit IO	Adafruit	Client
• Joram MQ	Scal Agent D.T	Broker
• M2Mqtt	Eclipse	Client
• moquette	Adrea Selva	Broker
• Mosquitto	Eclipse	Client & Broker.

4. Ways of using MQTT.
- plain MQTT
- MQTT over TLS
- MQTT over Websockets
- MQTT over Websockets with TLS.

FAQs.

1. Who invented MQTT?
→ Andy Stanford-Clark (1999) and Arlen Nipper.

2. Use of MQTT.
- Facebook messenger
- AWS
- Microsoft Azure
- Power Monitoring
- Lighting control
- Gardening.

3. Is MQTT a standard?

A. It is a standard protocol used for messaging and data exchange for IOT.

4. Is MQTT specification available to us?

A. MQTT is an OASIS standard and the specifications are managed and made available to us by the OASIS MQTT Technical Committee.

5. Does MQTT support security?

A. A username and password can be passed with an MQTT packet in V 3.1 of the protocol.

6. Standard ports for MQTT.

A TCP/IP port 1883 is reserved with IANA for use with MQTT. Port 8883 is also registered, for using MQTT over SSL.

7. What is a 'bridge'?

A. A bridge is a connection between 2 MQTT brokers.

PB_05_Kushagra Suryawanshi

EIOT Lab Assignment 06

Aim: To demonstrate MQTT/COAP/XMPP protocols using message broker to subscribe and publish sensor data.

Objectives:
1.To understand how sensor data will get published and subscribed
2.To learn various clients and brokers available for implementation

Dummy in Python for MQTT:
Code for publisher:

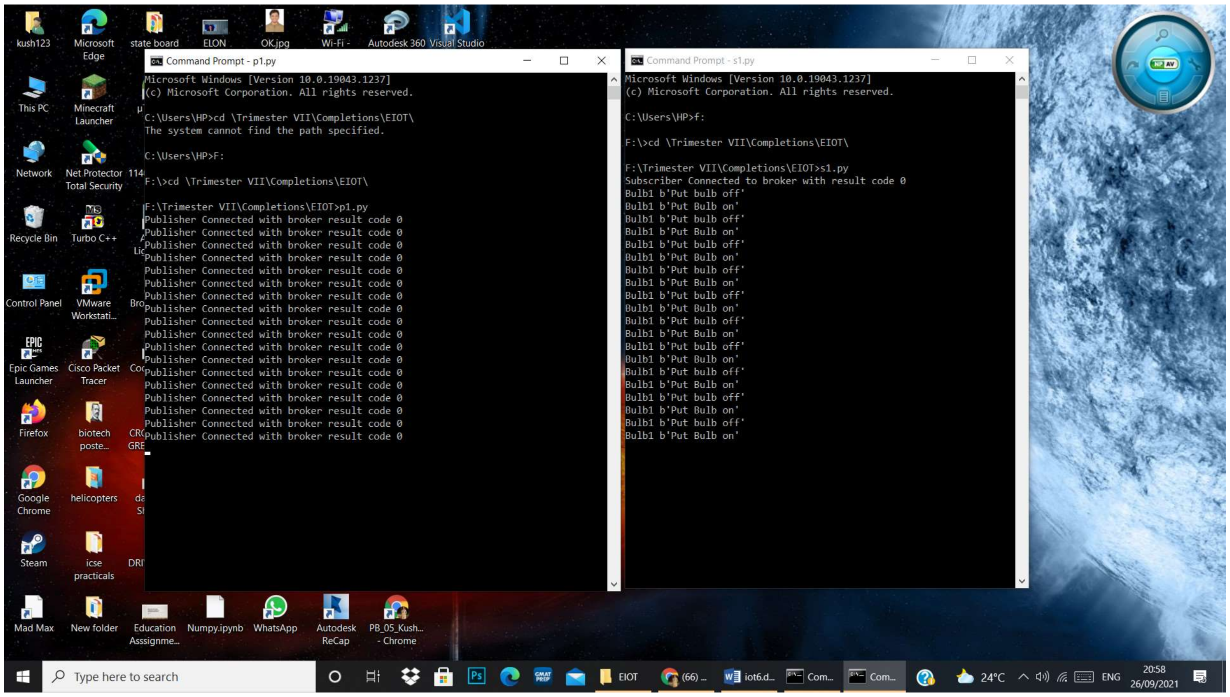
```
#IOT Publisher

import time

import paho.mqtt.client as mymqtt


#MQTT_SERVER = "test.mosquitto.org"

#MQTT_SERVER = "172.16.182.64"

MQTT_SERVER="broker.hivemq.com"

#MQTT_SERVER="iot.eclipse.org"

MQTT_TOPIC = "Bulb1"


def on_connect(client, userdata, flags, rc):

    print("Publisher Connected with broker result code "+str(rc))


client = mymqtt.Client()

client.on_connect = on_connect

i=1

while (i<=20):

  i+=1

  client.connect(MQTT_SERVER, 1883, 60)

  client.loop_start()
  client.publish(MQTT_TOPIC, "Put Bulb on")
  time.sleep(5)
  client.publish(MQTT_TOPIC, "Put bulb off")
  client.loop_stop()
```

Code for subscriber:

```
#IOT Subscriber
import paho.mqtt.client as mymqtt
import time

#MQTT_SERVER = "test.mosquitto.org"
#MQTT_SERVER = "172.16.182.64"
MQTT_SERVER="broker.hivemq.com"
#MQTT_SERVER="iot.eclipse.org"

MQTT_TOPIC = "Bulb1"

def on_connect(client, userdata, flags, rc):
    print("Subscriber Connected to broker with result code "+str(rc))
    client.subscribe(MQTT_TOPIC)


def on_message(client, userdata, msg):
    print(msg.topic+" "+ str(msg.payload))

client = mymqtt.Client()
client.on_connect = on_connect
client.on_message = on_message


client.connect(MQTT_SERVER, 1883, 60)
client.loop_start()
time.sleep(100)
client.loop_stop()
```

Output:

Publisher terminal and Subscriber terminal:



Command Prompt - p1.py

```
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>cd \Trimester VII\Completions\EIOT\
The system cannot find the path specified.

C:\Users\HP>F:

F:\>cd \Trimester VII\Completions\EIOT\

F:\Trimester VII\Completions\EIOT>p1.py
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
Publisher Connected with broker result code 0
```

Command Prompt - s1.py

```
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>f:

F:\>cd \Trimester VII\Completions\EIOT\

F:\Trimester VII\Completions\EIOT>s1.py
Subscriber Connected to broker with result code 0
Bulb1 b'Put bulb off'
Bulb1 b'Put Bulb on'
Bulb1 b'Put bulb off'
Bulb1 b'Put Bulb on'
Bulb1 b'Put bulb off'
Bulb1 b'Put Bulb on'
Bulb1 b'Put bulb off'
Bulb1 b'Put Bulb on'
Bulb1 b'Put bulb off'
Bulb1 b'Put Bulb on'
Bulb1 b'Put bulb off'
Bulb1 b'Put Bulb on'
Bulb1 b'Put bulb off'
Bulb1 b'Put Bulb on'
Bulb1 b'Put bulb off'
Bulb1 b'Put Bulb on'
Bulb1 b'Put bulb off'
Bulb1 b'Put Bulb on'
Bulb1 b'Put bulb off'
Bulb1 b'Put Bulb on'
```

Conclusion: Thus, we have studied how to use light weight messaging protocol so that sensors can publish and subscribe in over internet system and can communicate like one to one, one to many or many to many in order to disseminate the information for further processing.