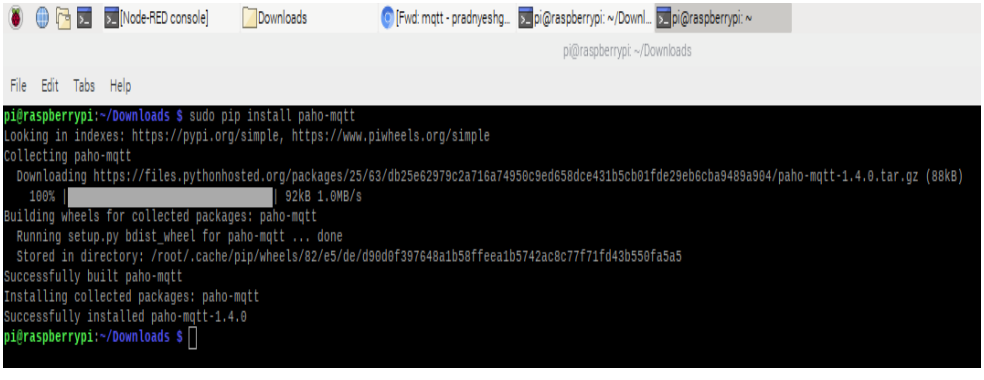
# Practical No. 9

**Aim:** Setup a Mosquitto MQTT server and client and write a Python script to communicate data between Pi's.

**Step 1:** Install MQTT server and clients packages

```
pi@raspberrypi:~ $ sudo apt-get install -y mosquitto mosquitto-clients
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libev4 libmosquitto1 libwebsockets8
The following NEW packages will be installed:
  libev4 libmosquitto1 libwebsockets8 mosquitto mosquitto-clients
0 upgraded, 5 newly installed, 0 to remove and 205 not upgraded.
Need to get 387 kB of archives.
After this operation, 843 kB of additional disk space will be used.
Get:1 http://mirror.ossplanet.net/raspbian/raspbian buster/main armhf libev4 armhf 1:4.25-1 [34.5 kB]
Get:2 http://mirror.ossplanet.net/raspbian/raspbian buster/main armhf libmosquitto1 armhf 1.5.7-1 [57.4 kB]
Get:3 http://mirror.ossplanet.net/raspbian/raspbian buster/main armhf libwebsockets8 armhf 2.0.3-3 [85.6 kB]
Get:4 http://mirror.ossplanet.net/raspbian/raspbian buster/main armhf mosquitto armhf 1.5.7-1 [143 kB]
Get:5 http://mirror.ossplanet.net/raspbian/raspbian buster/main armhf mosquitto-clients armhf 1.5.7-1 [66.7 kB]
Fetched 387 kB in 4s (89.3 kB/s)
Selecting previously unselected package libev4:armhf.
(Reading database ... 152993 files and directories currently installed.)
Preparing to unpack .../libev4_1%3a4.25-1_armhf.deb ...
Unpacking libev4:armhf (1:4.25-1) ...
Selecting previously unselected package libmosquitto1:armhf.
Preparing to unpack .../libmosquitto1_1.5.7-1_armhf.deb ...
Unpacking libmosquitto1:armhf (1.5.7-1) ...
Selecting previously unselected package libwebsockets8:armhf.
Preparing to unpack .../libwebsockets8_2.0.3-3_armhf.deb ...
Unpacking libwebsockets8:armhf (2.0.3-3) ...
Selecting previously unselected package mosquitto.
Preparing to unpack .../mosquitto_1.5.7-1_armhf.deb ...
Unpacking mosquitto (1.5.7-1) ...
Selecting previously unselected package mosquitto-clients.
Preparing to unpack .../mosquitto-clients_1.5.7-1_armhf.deb ...
Unpacking mosquitto-clients (1.5.7-1) ...
Setting up libmosquitto1:armhf (1.5.7-1) ...
Setting up libev4:armhf (1:4.25-1) ...
Setting up mosquitto-clients (1.5.7-1) ...
Setting up libwebsockets8:armhf (2.0.3-3) ...
Setting up mosquitto (1.5.7-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mosquitto.service → /lib/systemd/system/mosquitto.service.
Processing triggers for systemd (241-5+rpi1) ...
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for libc-bin (2.28-10+rpi1) ...
```

**Step 2:** Install paho package of MQTT



**Step 3**: set the server IP address in the python program of server and client

mqtt_publisher.py - I:\pract9\mqtt_publisher.py (3.7.3)

File Edit Format Run Options Window Help

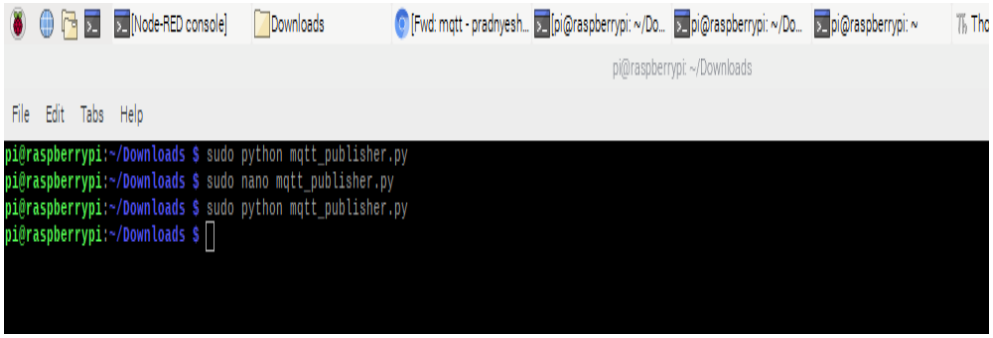
```
import paho.mqtt.publish as publish
MQTT_SERVER = "192.168.0.73"
MQTT_PATH = "test_channel"
publish.single(MQTT_PATH, "pradnyesh", hostname=MQTT_SERVER)
```

mqtt_subsciber.py - I:\pract9\mqtt_subsciber.py (3.7.3)

File Edit Format Run Options Window Help

```
import paho.mqtt.client as mqtt

MQTT_SERVER = "192.168.0.73"
MQTT_PATH = "test_channel"

# The callback for when the client receives a CONNACK response from the server.
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))

    # Subscribing in on_connect() means that if we lose the connection and
    # reconnect then subscriptions will be renewed.
    client.subscribe(MQTT_PATH)

# The callback for when a PUBLISH message is received from the server.
def on_message(client, userdata, msg):
    print(msg.topic+" "+str(msg.payload))
    # more callbacks, etc

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

client.connect(MQTT_SERVER, 1883, 60)

# Blocking call that processes network traffic, dispatches callbacks and
# handles reconnecting.
# Other loop*() functions are available that give a threaded interface and a
# manual interface.
client.loop_forever()
```

**Step 4:** Run the server program.




**Step 5:** Run client script and send message to server