University College

IT Technology
Assignment 55
Basic MQTT devices on VMWW bridged network.

Author
Dainty Lyka Bihay Olsen
dlbo28887@edu.ucl.dk

Date: 21/03-2021

Contents

1. Introduction

In this assignment the goal is to create a MQTT-broker, MQTT-publisher, and MQTT-subscribers. And attempt to transfer data from the publisher to the Broker which will be directed to any subscriber that are asking for it.

2. Audience

This report is designated for students of the IT Technology course.
This document will show how Group B3 set up the networks to fulfil the requirements of Assignment 55.
Everything following will show how the group went about setting up the system.

In this assignment there is also conclude a guide to set up the network in VMware Workstation and how to configure the virtual machines to make this assignment work.

3. Inventory

Installed on the computer or laptop.

- VMware Workstation
- At least tree virtual machines (preferably Xubuntu, and one RaspberryPi Buster)

Installed software on the Raspios-buster.

- Mosquitto
- Mosquitto-tools
- Paho-mqqt (not necessary but nice to have)
- Python3-pip

Installed software on the Linux Xubuntu

- Paho-mqqt
- Python3-pip

4. Learning objectives

Learning objectives. After having worked with this assignment:

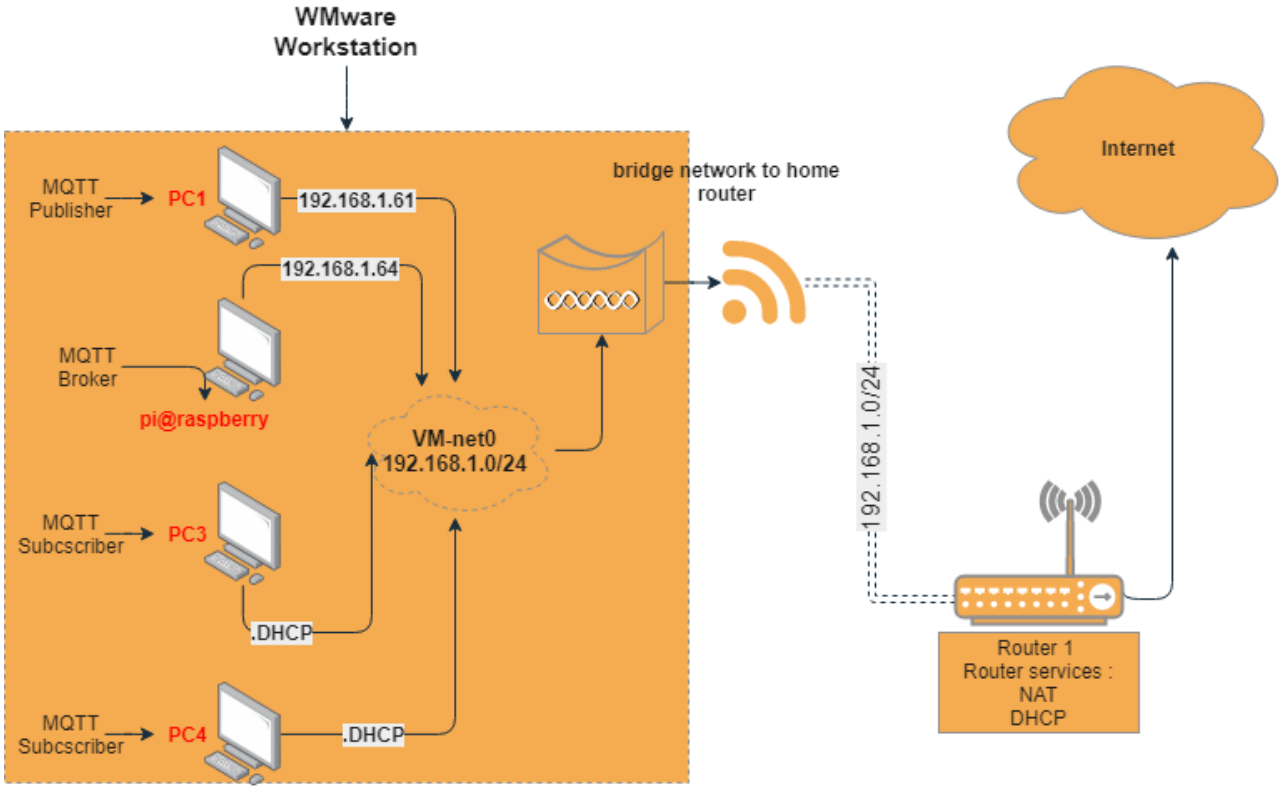
The student can at a basic level explain:

- How to set up MQTT broker, publisher, and subscriber on Linux.
- What Bridge does in VMware Workstation.

The student can at a basic level:

- Set up a Bridged network in VMware Workstation.
- Set up a MQTT Broker on Linux, using Mosquitto.
- Set up a MQTT Python Publisher on Linux.
- Set up a MQTT Python Subscriber on Linux.
- Verify that the Broker, Publisher, and Subscriber can communicate via MQTT.

5. Network diagram.

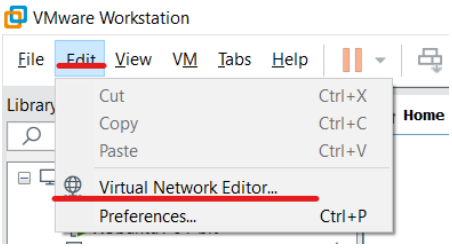


In this assignment one VM-net will be used, that is VM-net0 which runs on 192.168.1.0/24 all devices are connected to that network, VM-net0 is bridged to the real computers network that is connected to a home router. So, all computers can interconnect with each other and still have access to the internet.

6. How to configure the network

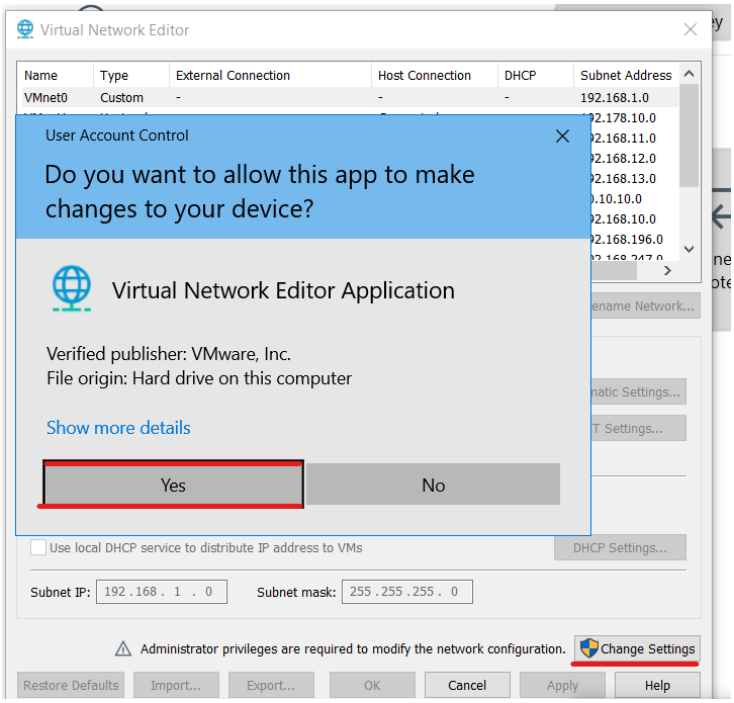
Configure VM-ware Workstation.

Step 1



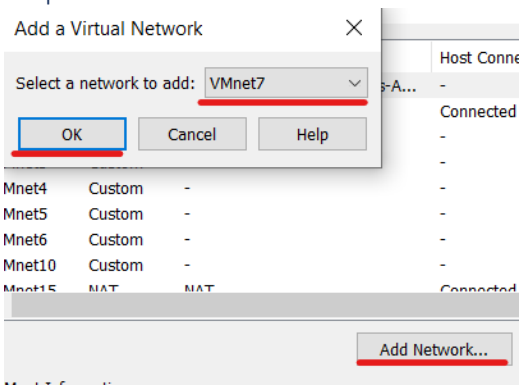
- Click on edit in the top left corner of VMware.
- Click on virtual network editor.

Step 2



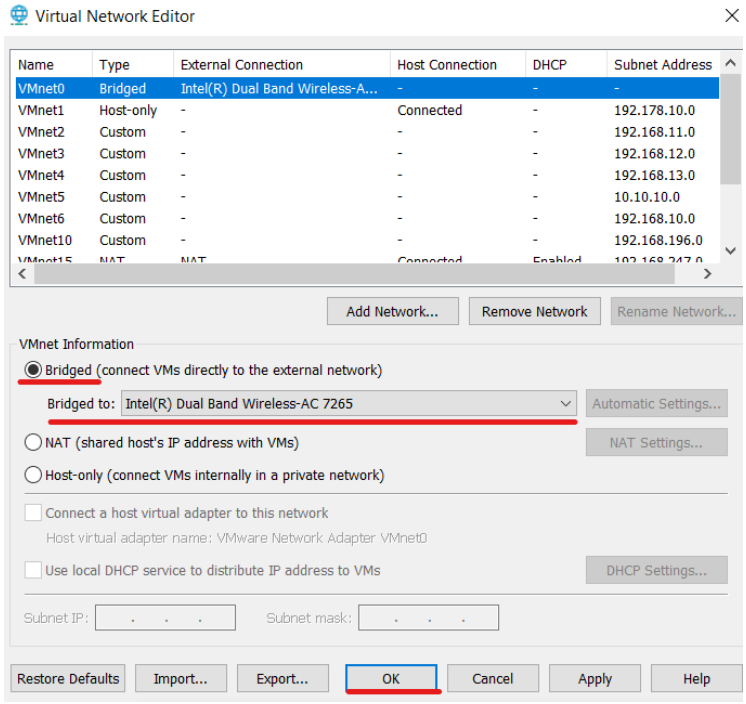
- Click on chance settings.
- In the popup window press Yes.

Step 3



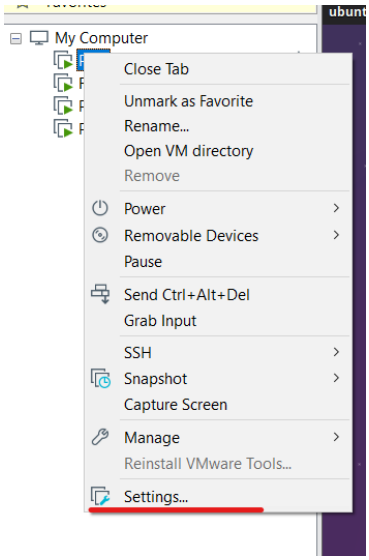
- Click Add Network.
- Select a network (in this example VMnet0 was chosen).
- Click OK.

Step 4



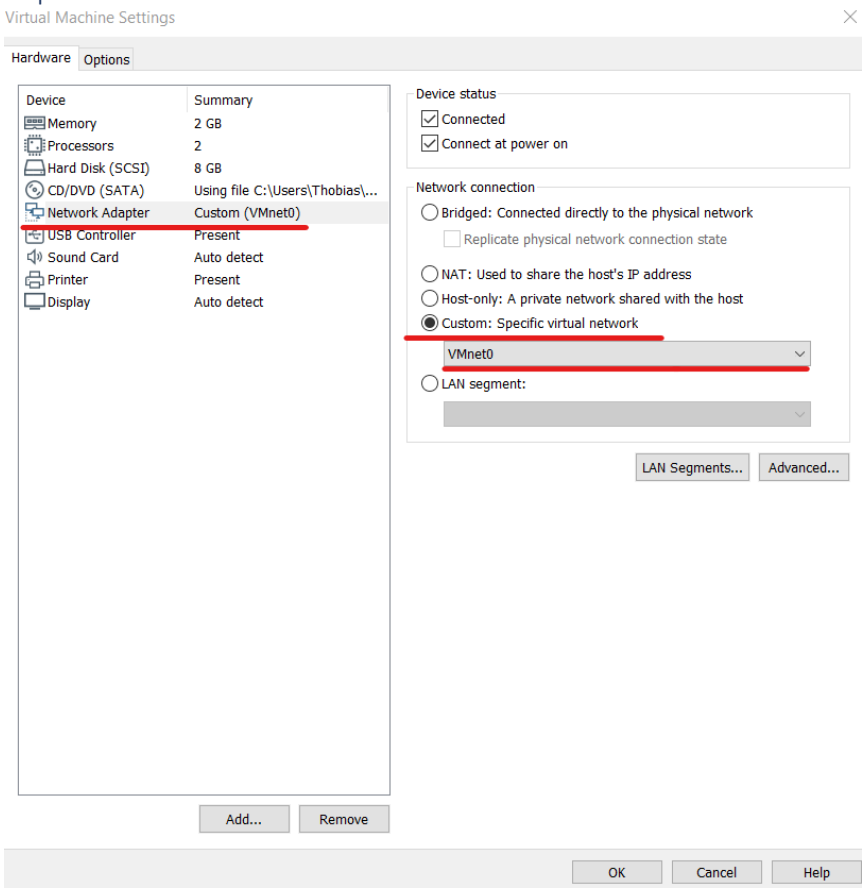
- Dobble click VMnet0.
- Check the box Bridge.
- Select the network your computer is runing on.
- Click OK.

Step 5



- Right click on one of the virtual computers that will be used doing this assignment.
- Click settings.

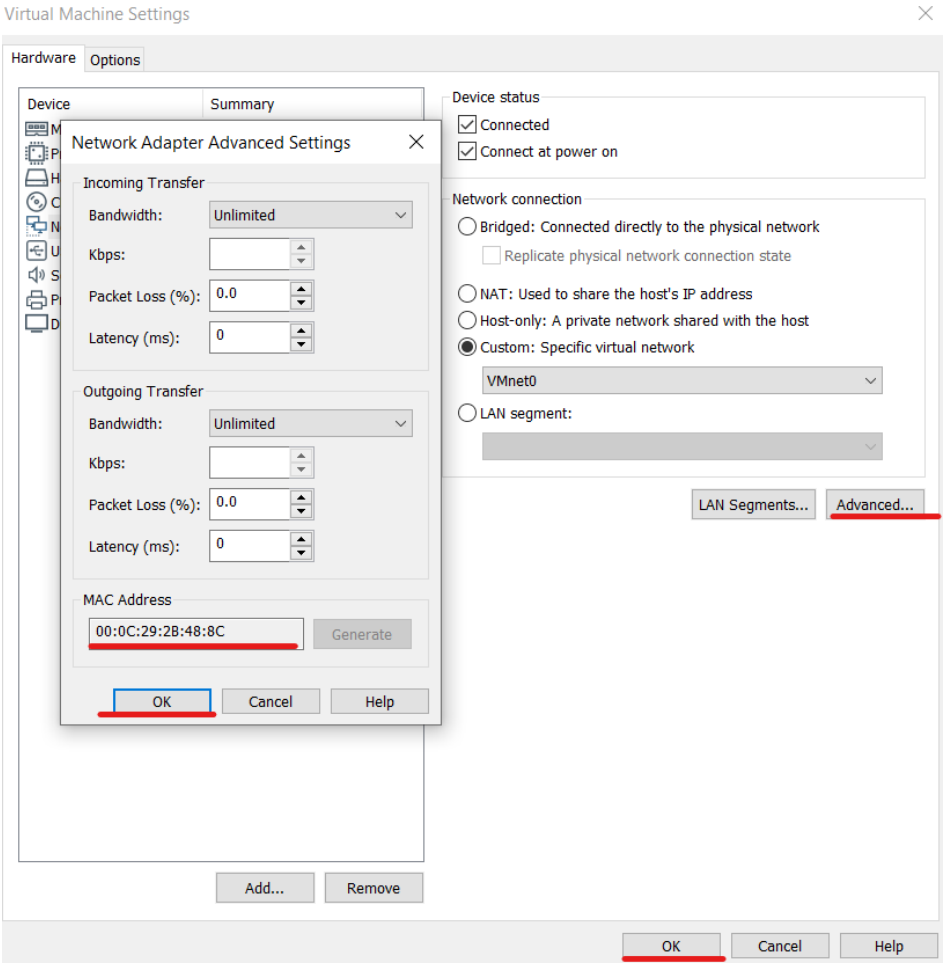
Step 6



- Click on Network apdapter.

- Check the box Costom.
- Select VMnet0.

Step 7



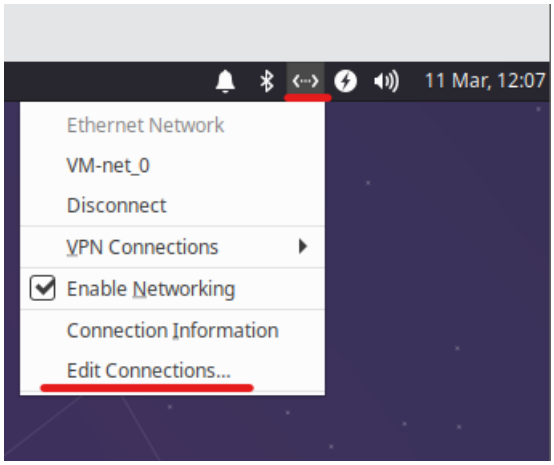
- Click Advanced.
- Write down the MAC address (it will be needed in next chapter).
- Click OK in advanced settings.
- Click OK.

Step 8

- Repeated these steps for all virtual machines.

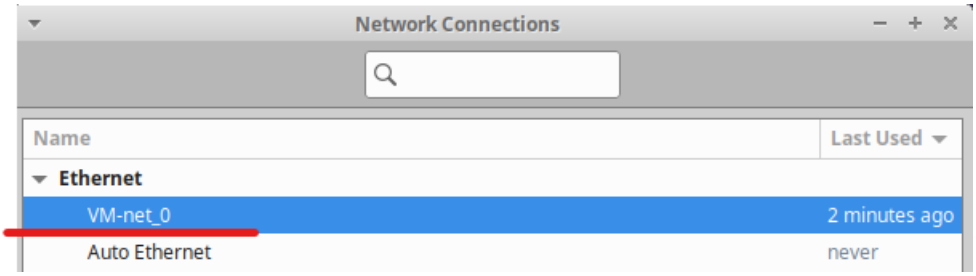
Configure Linux PC 1

Step 1



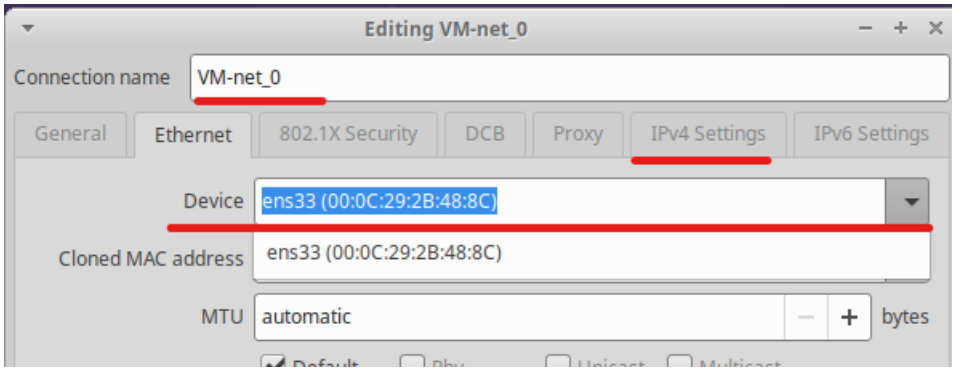
- Click on the network symbol in the top right corner.
- Click edit connection…

Step 2



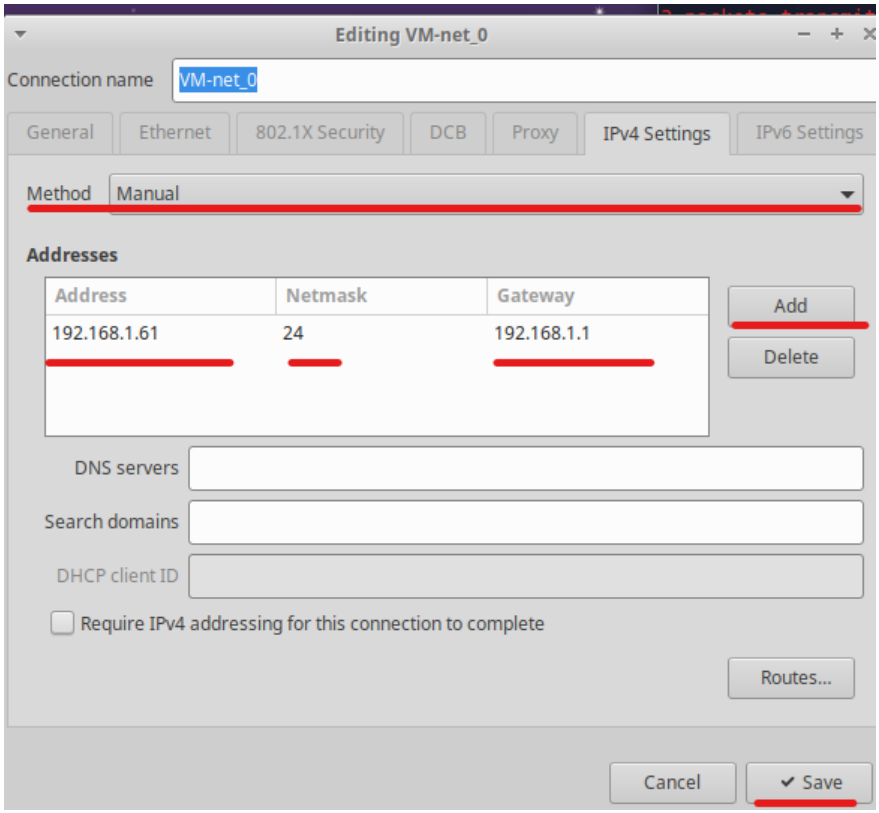
- Choose an Ethernet and double click it.

Step 3



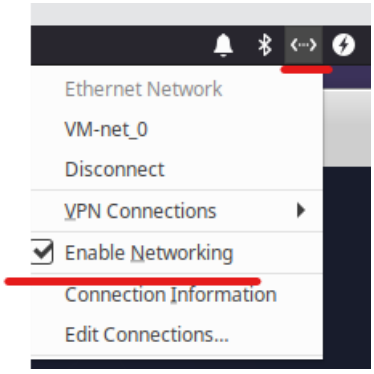
- Name the Connection.
- Make sure that the MAC address matches the ones that was written down in the previous chapter.
- Click on IPv4 Settings.

Step 4



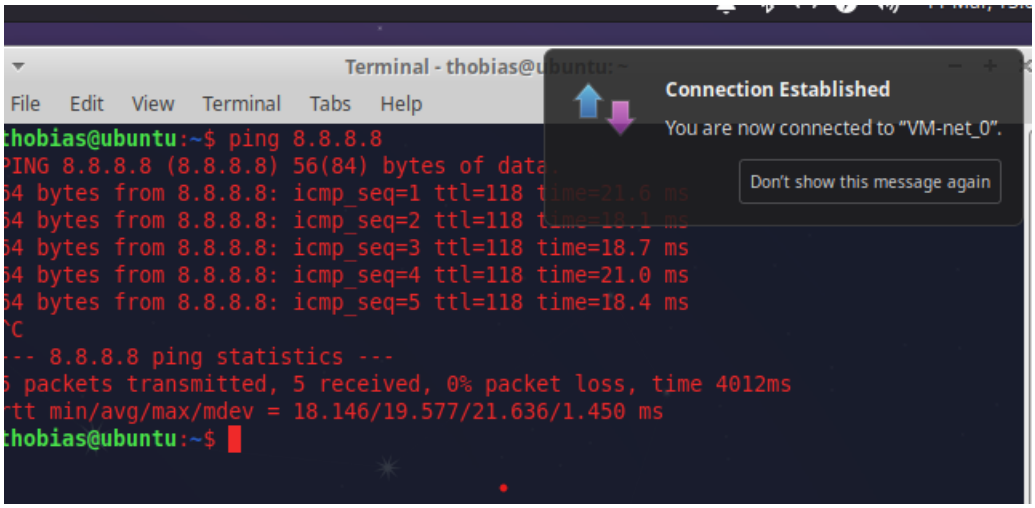
- Choose MANUAL in the method dropdown menu.
- Add a new address.
- Write in the static IP address, netmask, and gateway.
- Click Save.

Step 5



- Click on the Network symbol.
- Uncheck and then check the box in enable networking.

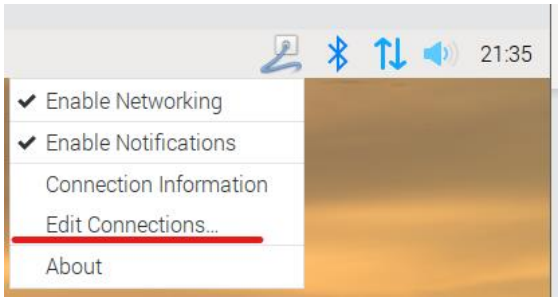
Step 6



- To be sure that there is a real connection ping google on 8.8.8.8

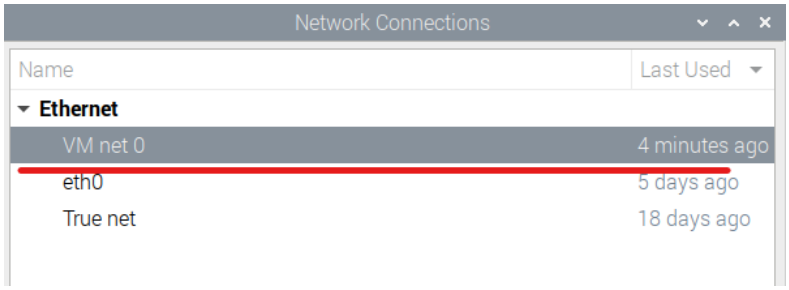
Configure Raspberry

Step 1



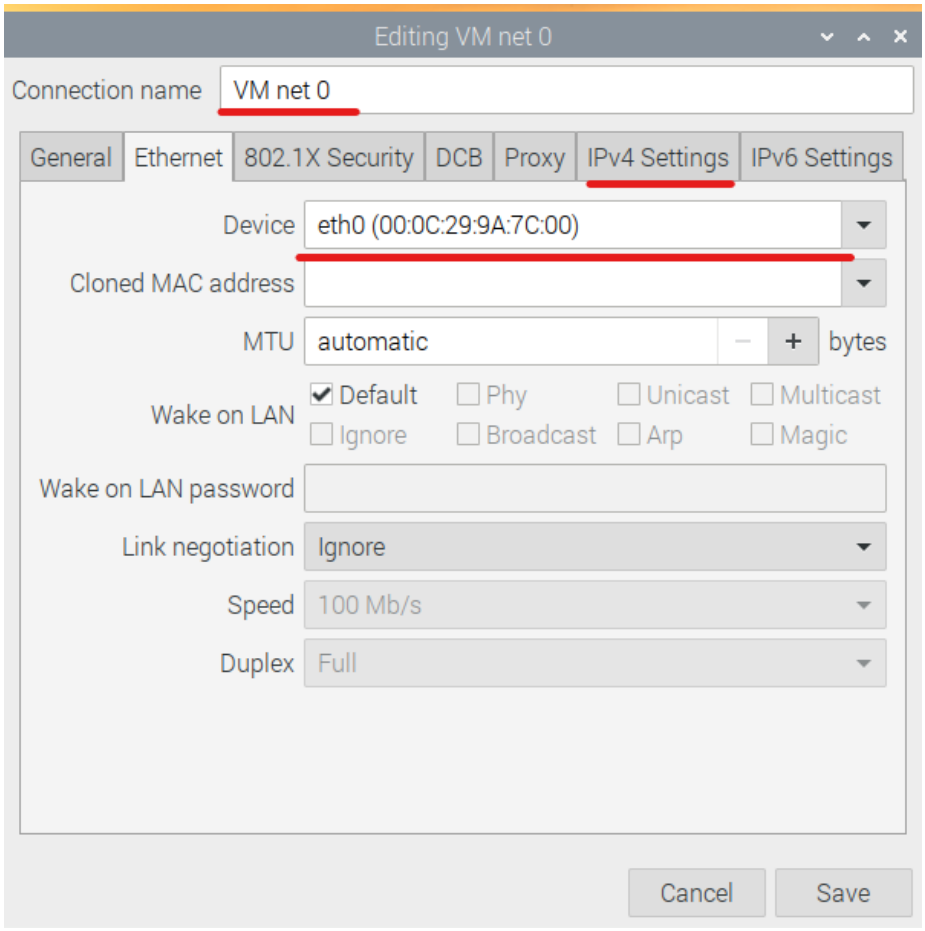
- Click on the network symbol in the top right corner.
- Click edit connection…

Step 2



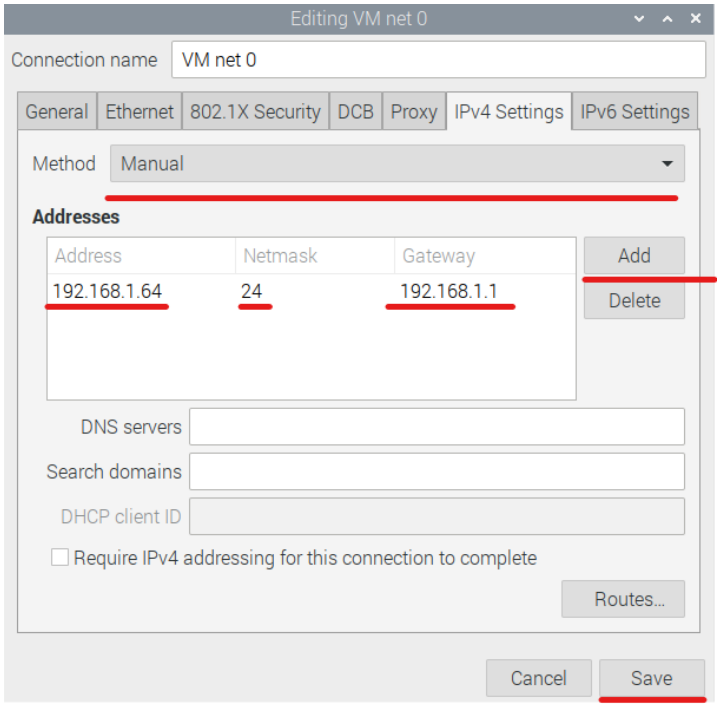
- Choose an Ethernet and double click it.

Step 3



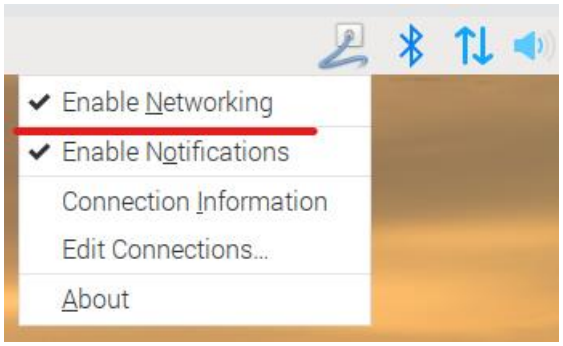
- Name the Connection.
- Make sure that the MAC address matches the ones that was written down in the previous chapter.
- Click on IPv4 Settings.

Step 4



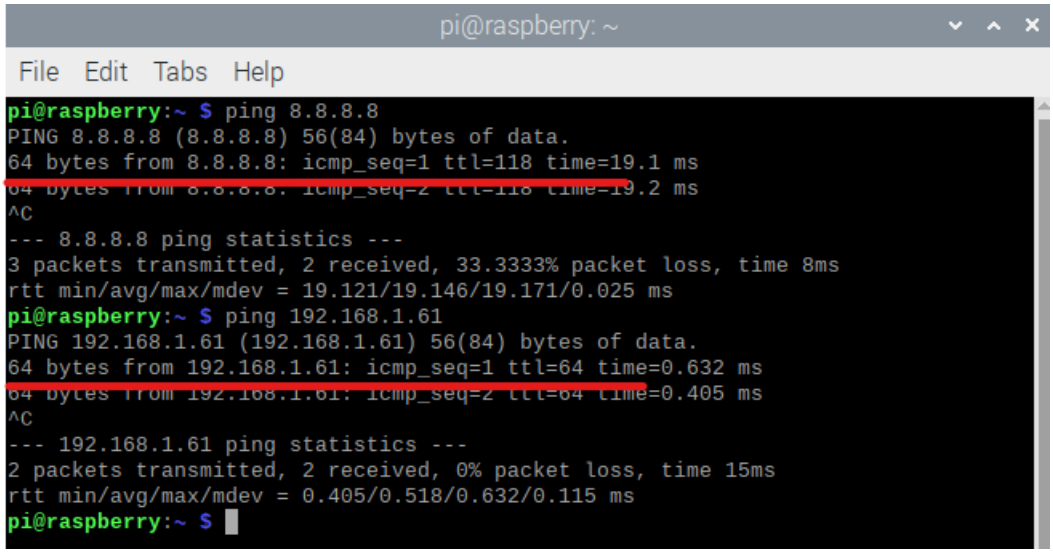
- Choose MANUAL in the method dropdown menu.
- Add a new address.
- Write in the static IP address, netmask, and gateway.
- Click Save.

Step 5



- Click on the Network symbol.
- Uncheck and then check the box in enable networking.

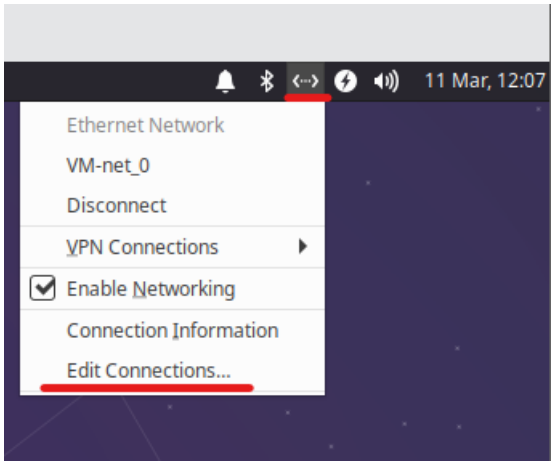
Step 6



- **To be sure that there is a real connection.**
  - ping google on 8.8.8.8
  - open a web browser and load a webpage.
- **To be sure that the is a connection between the virtual machines.**
  - ping pc 1 on 196.168.1.61

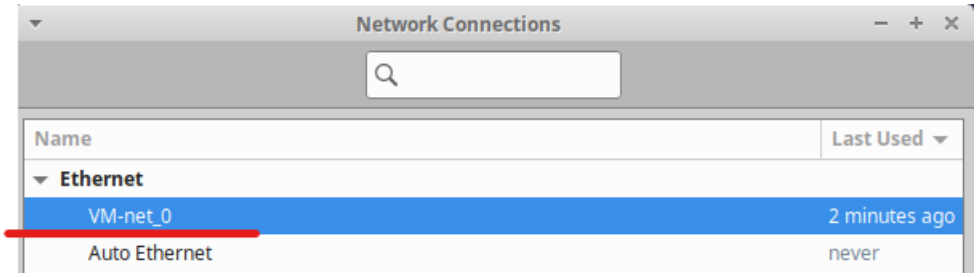
Configure Linux PC 3 and 4

Step 1



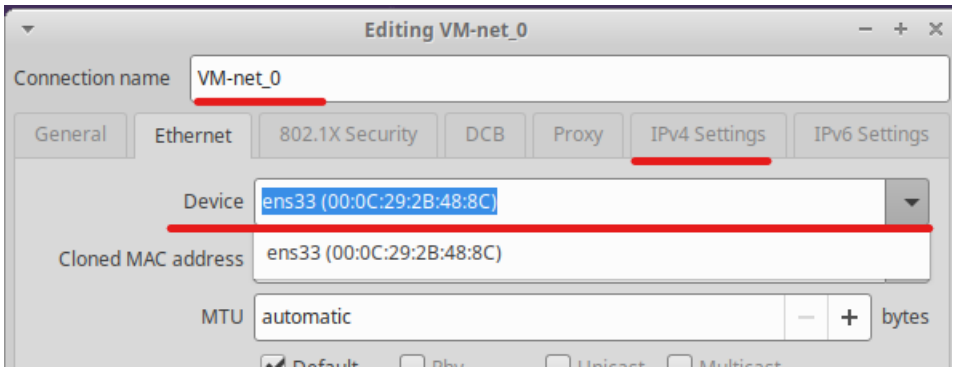
- Click on the network symbol in the top right corner.
- Click edit connection…

Step 2



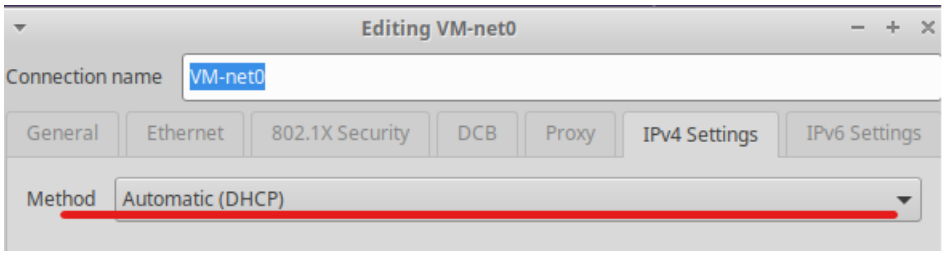
- Choose an Ethernet and double click it.

Step 3



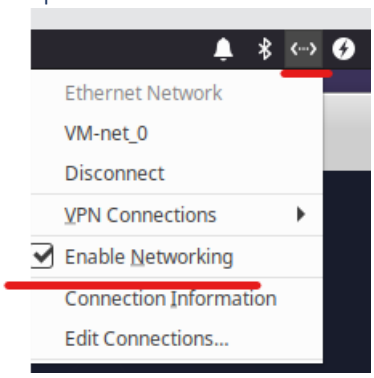
- Name the Connection.
- Make sure that the MAC address matches the ones that was written down in the previous chapter.
- Click on IPv4 Settings.

Step 4



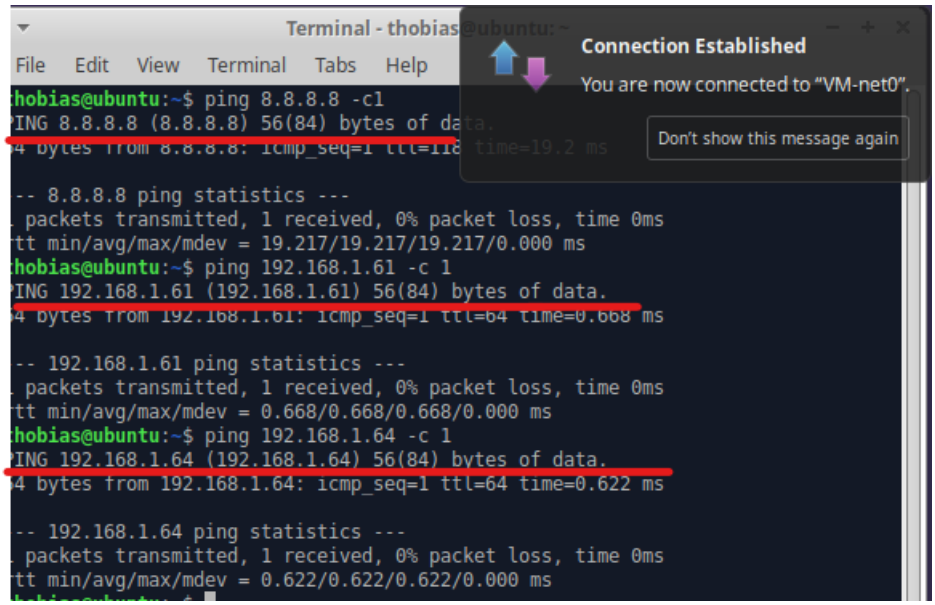
- Choose automatic (DHCP) in the method dropdown menu.
- Click Save.

Step 5



- Click on the Network symbol.
- Uncheck and then check the box in enable networking.

Step 6



- **To be sure that there is a real connection.**
  - ping google on 8.8.8.8
  - open a web browser and load a webpage.
- **To be sure that the is a connection between the virtual machines**.
  - ping pc 1 on 196.168.1.61
  - ping pc 2 on 196.168.1.64

7. Installation on the virtual machines

MQTT-publisher PC 1

On PC 1 open the terminal and run the commands shown below

These lines contain:

- Update of your system
- Upgrade of your system
- Python 3 pip installer
- Paho-MQTT

```
sudo apt-get update

.....

sudo apt-get upgrade

.....

sudo apt install python3-pip

.....

pip3 install paho-mqtt
```

MQTT-subscribers PC 3 and 4

On PC 1 open the terminal and run the commands shown below

These lines contain:

- Update of your system
- Upgrade of your system
- Python 3 pip installer
- Paho-MQTT

```
sudo apt-get update
.....
sudo apt-get upgrade
.....
sudo apt install python3-pip
.....
pip3 install paho-mqtt
.....
pip install paho-mqtt
```

MQTT-broker Raspberry

On PC 1 open the terminal and run the commands shown below

These lines contain:

- Update of your system
- Upgrade of your system
- Python 3 pip installer
- Paho-MQTT

```
sudo apt-get update
.....
sudo apt-get upgrade
.....
sudo apt install python3-pip
.....
pip3 install paho-mqtt
```

8. Hand in

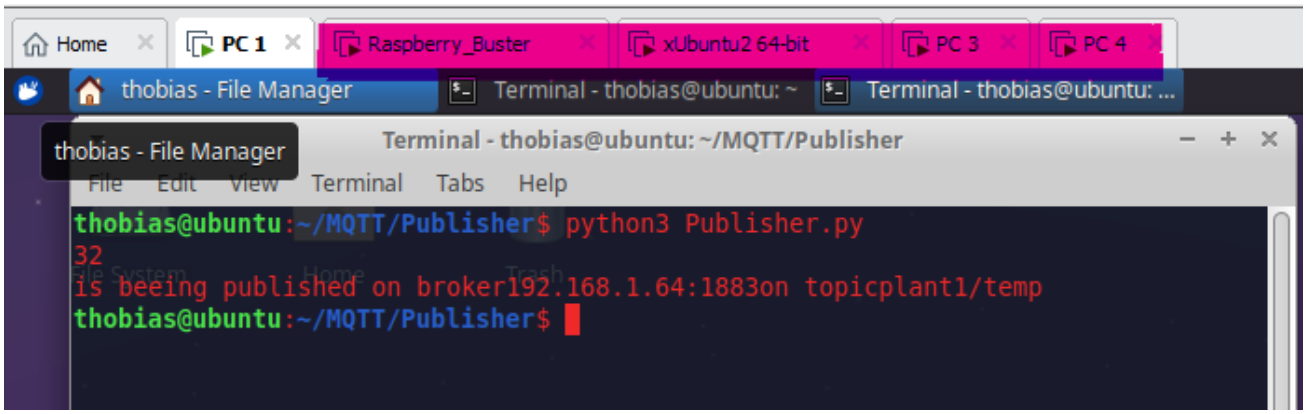
From here starts the requirements for passing the assignment.

MQTT Python Publisher program

Code and explanation

```
#! /usr/bin/env python
import paho.mqtt.client as mqtt
import random

brokerIP = "192.168.1.64" ### here is the IP for the broker
this it also IP address of the Raspberry

brokerPort = 1883 ### here is the broker port

brokerKeepAlive = 60 ### If no data flows over an open
connection for a time period of 60 sec.

myTopic = "plant1/temp" ### name of a "topic you search for"

myPayload = random.randint(1, 99) ### here is a random number
between 1 and 99 generated

myQoS = 1 ### this makes the broker reply to the publisher
that it has received the data.

myRetain = True ### help newly-subscribed clients get a status
update immediately after they subscribe to a topic.

client = mqtt.Client() ### here we make a client

### and as this client we are connecting on the broker via the
information of its IP address, port, and the KeepAlive
statement.
client.connect(brokerIP, brokerPort, brokerKeepAlive)

### here is the data-points that is being to send to the
broker
client.publish(topic=myTopic, qos=myQoS, payload=myPayload,
retain=myRetain)

### here the data-points being presented in a print statements
are using the parameter from earlier
print(myPayload)
print("is beeing published on broker " + brokerIP + " : " +
str(brokerPort) + " On topic " + myTopic)
client.disconnect()
```

Screenshot showing a sample Publisher run.



Here it can be observed from a terminal on MQTT-Publisher PC 1 that we are sending the random generated number 32 over to the broker on the IP-address of 192.168.1.64 through port 1883 with the topic message plant1/temp.

It can also be observed that there were no spaces before and after all sentence in the print statements.

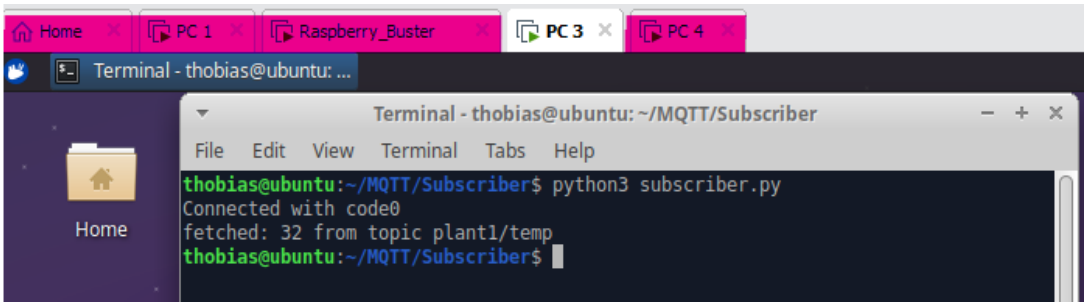
MQTT Python Subscriber program

Code and Explanation

```
#!/usr/bin/env/ python 3
import paho.mqtt.client as mqtt

brokerIP = "192.168.1.64" ### here is the IP for the broker
this it alsor IP address of the Raspberry

brokerPort = 1883 ### here is the broker port

brokerKeepAlive = 60 ### If no data flows over an open
connection for a time period of 60 sec.

myTopic = "plant1/temp"### name of a "topic you search for"

### here a function to connect the MQTT-broker is made.
def on_connect(client, userdata, flags, rc):
print("Connected with code" + str(rc))
client.subscribe(myTopic)

### here a fungtion to get the randon number and the topic
name is made
def message(client, userdata, msg):
print("fetched: " + str(msg.payload.decode())+" from topic "
+ myTopic)
client.disconnect()


client = mqtt.Client()### here we make a client

### and as this client we are connecting on the broker via
the information of its IP address, port, and the KeepAlive
statement.
client.connect(brokerIP, brokerPort, brokerKeepAlive)

###here we are calling the on_connect function
client.on_connect = on_connect

###here we are calling the messeage function
client.on_message = message

client.loop_forever()
```

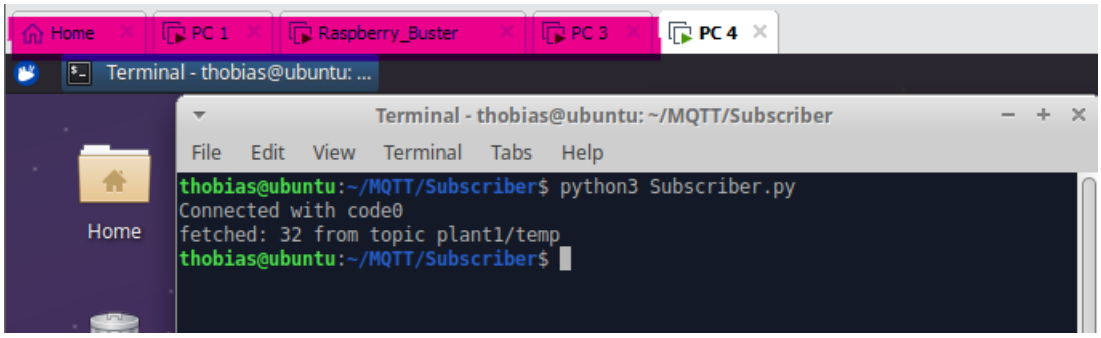
Screenshot showing a sample Subscriber run, proving that Subscriber can retrieve data from Publisher.

PC 3



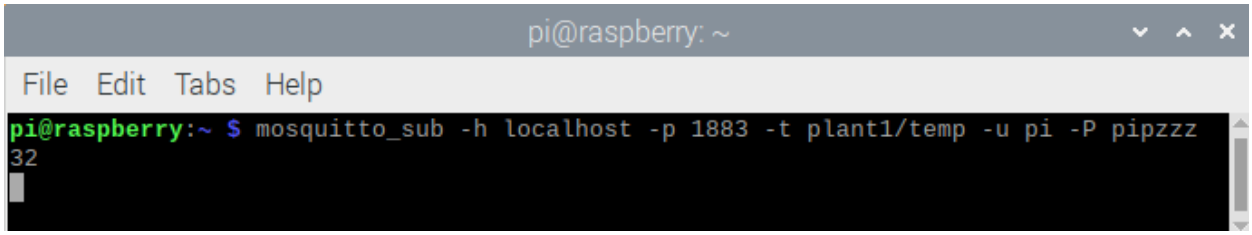
It can be observed from Subscriber-PC3 that when the script runs, it receives two messages, one about the connection and the other one is the random number that was generated by the publisher.

PC 4



The exact same thing can be observed on Subscriber-PC4.

Screenshot showing the MQTT broker working.



From the MQTT-Broker raspberry you will have to launch the broker by using

This command:

```
mosquitto_sub -h localhost -p 1883 -t plant1/temp -u pi -P pipzzz
```

After that whenever the Publisher is publishing something it will be shown in the terminal as can be observed in the picture above.
The number 32 was generated by the publisher and send to the Broker for storage and given to any subscriber that is asking for it.

9. Conclusion

In this paper the students have shown their capability regarding setting up VMnet0 to bridge to the host computer. A demonstration on the setup of multiple network adapters namely PC1,2,3 and 4 to use the same network as the host computer.

Furthermore, after successfully setting up MQTT Broker using Mosquitto, publisher and subscriber on Linux, a display of communication between network devices via MQTT was completed.