7COM1076/7COM1030/7COM1031 MULTIMEDIA AND MULTICASTING NETWORKING/ WIRELESS MOBILE AND ADHOC NETWORKING/ WIRELESS MOBILE AND MULTIMEDIA NETWORKING

LAB WORKSHEET 1 – INTRODUCTION TO MININET, MININET API AND PYTHON

Objectives

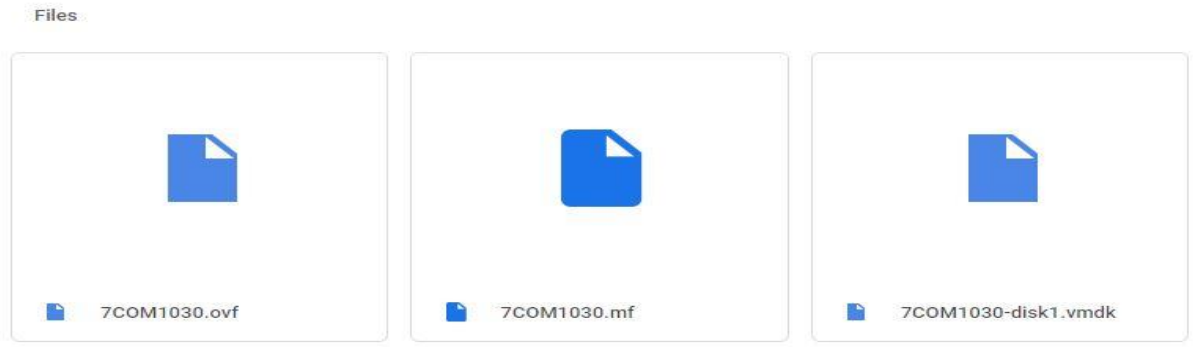
- Introduction to VmWare Player
- Launch a Virtualized Operating System in VmWare Player
    - With the evolution of virtualization in both academia and industry VmWare and Virtual Box have played a key role in life cycle of many applications and services. We are using VmWare Player a better tool for virtualization compared to Virtual Box.
- Introduction to Ubuntu 18.04 LTS
- Introduction to Mininet
    - To understand Mininet and it's functionality towards network emulation. Since Mininet has the ability to emulate a network and it's function escaping the simulated environments, more experiments under different scenarios and use cases can be launched.

TASK 1

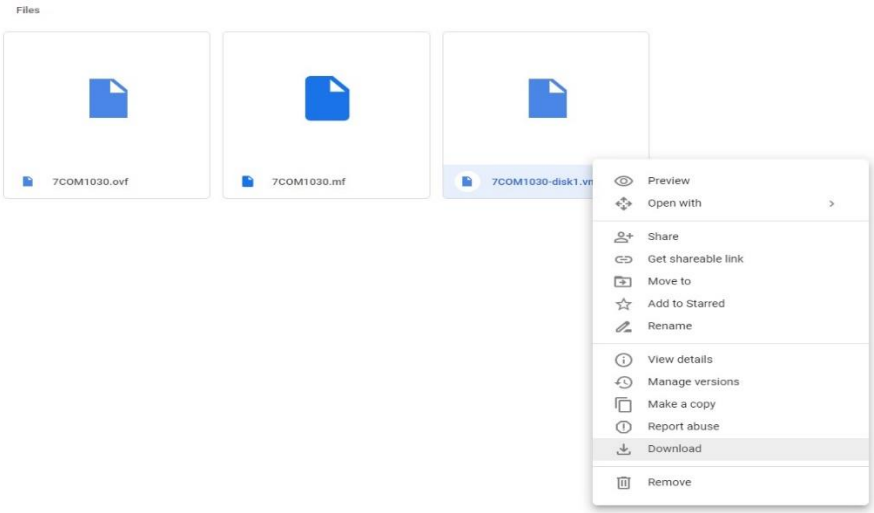
Download or obtain the OVA file which will run the VmWare Player for lab sessions.

https://drive.google.com/open?id=1_qfqkRoqS2EUEvmEztbo9KrtSUEVGPqs

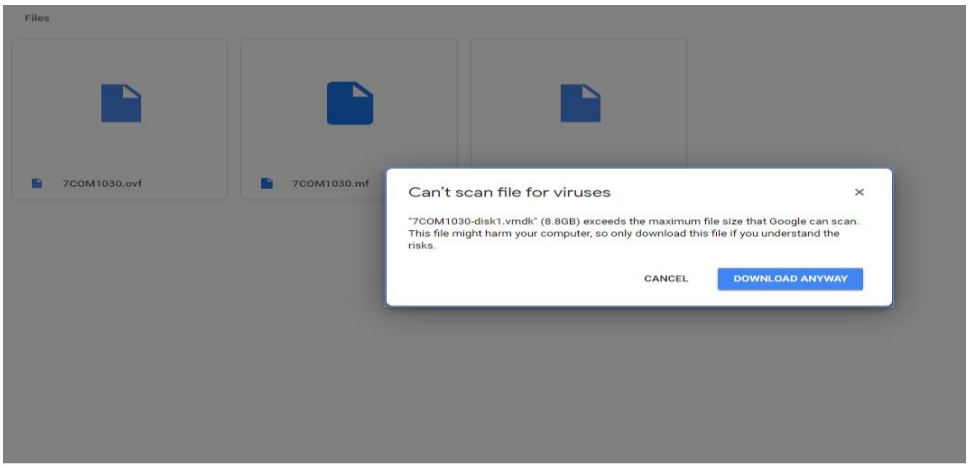
Copy the link above paste it on the browser. Follow the following instructions. There are three files in total. You need all three of them to work with.



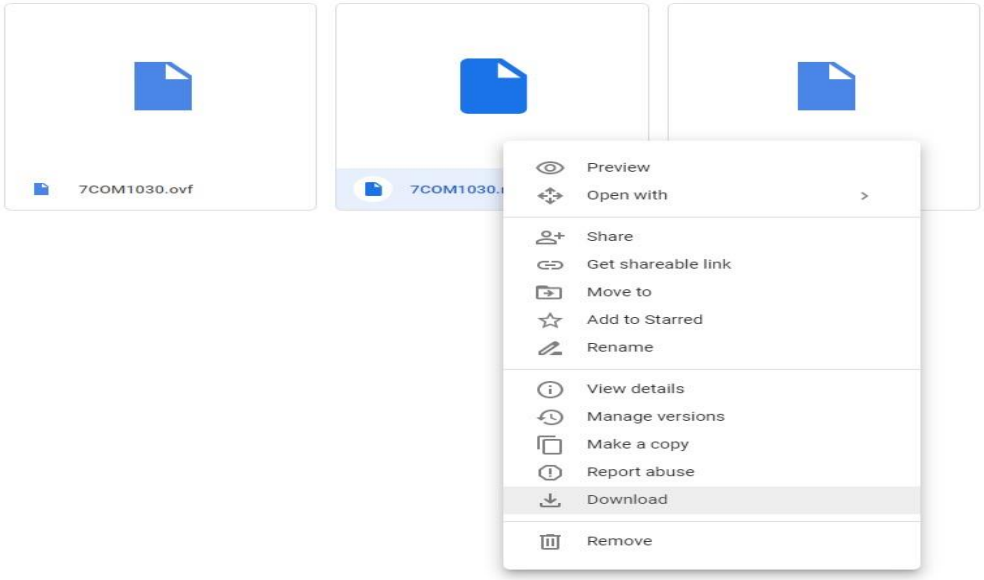
Right click on the file you want to download and select Download.



On the prompt select **Download anyway.**



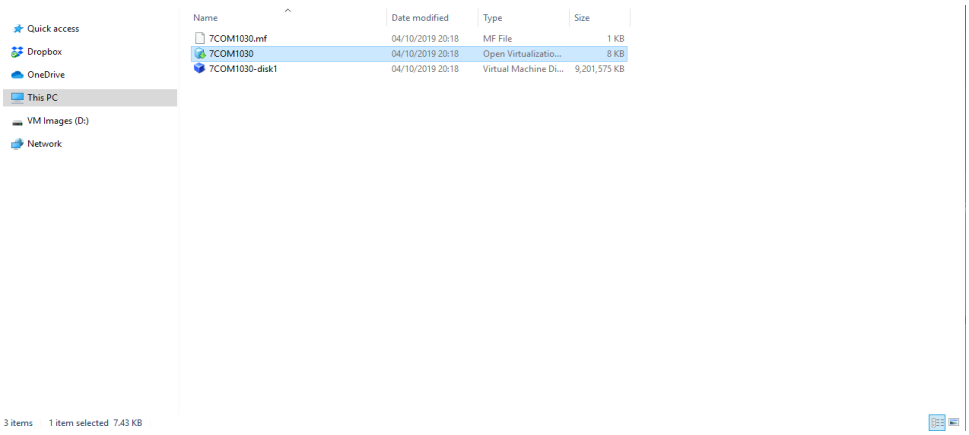
Do the above for the other two files as well.



You should see as the next image on the bottom of your browser.

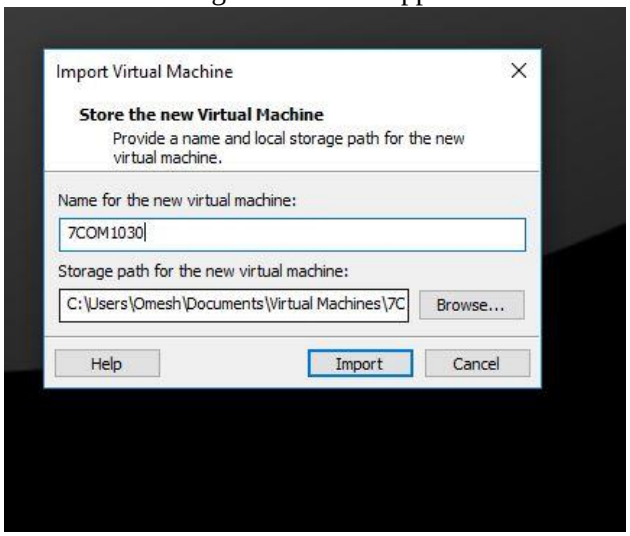
Click on the arrow button on one of the downloaded files and select **Show in Folder** option.

Once you have the relevant files as below diagram, follow the guide below.

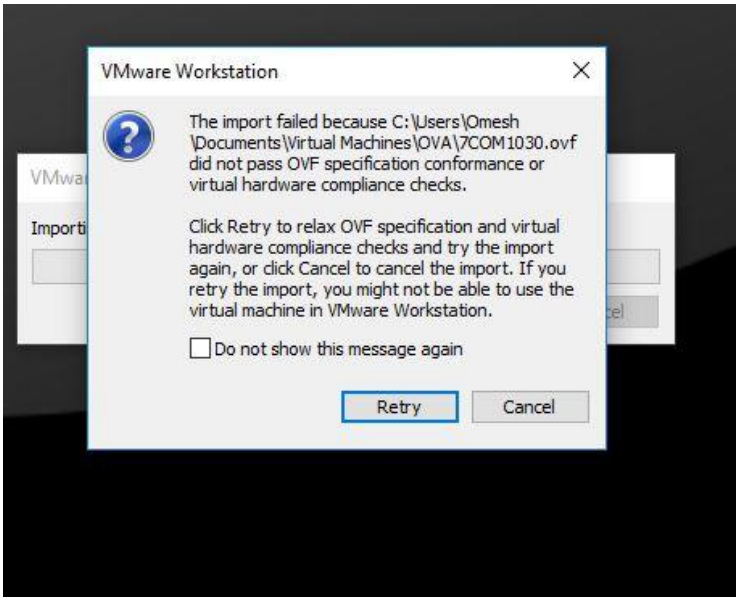


Double Click on the 7COM1030 ova file. The following pop up window will appear.

Next the following window will appear.

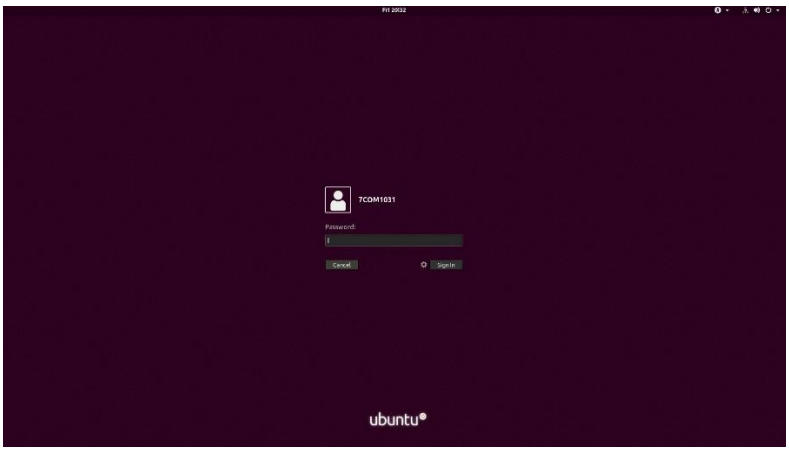


Change the storage location as to your desired folder and click Import. Do not change the name of the VM for the time being. The following window will appear. Click **RETRY.**



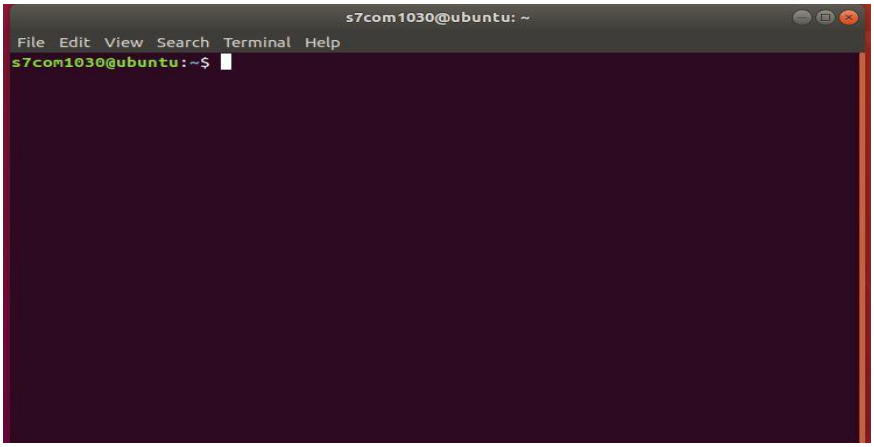
After that your VmWare Player will open up in the following way and click on Power On this Virtual Machine. If you have not reached this place, please ask for help. If you have reached this place, Ubuntu 18.04LTS will boot up now.

Now upon booting up, you will get the following window. If not click on the user 7COM1031. If you are in the courses of 7COM1030 and 7COM1076 this is still the valid OS to use.



Congratulations, your VM works perfectly. Password to enter is **123.** If you face any issues please feel free to ask for help. Now Let's start experimenting with Mininet. First we need to open up a terminal. On the Top left corner, you will see Activities. Click on it. Now type **terminal** as follows. Tap Enter.

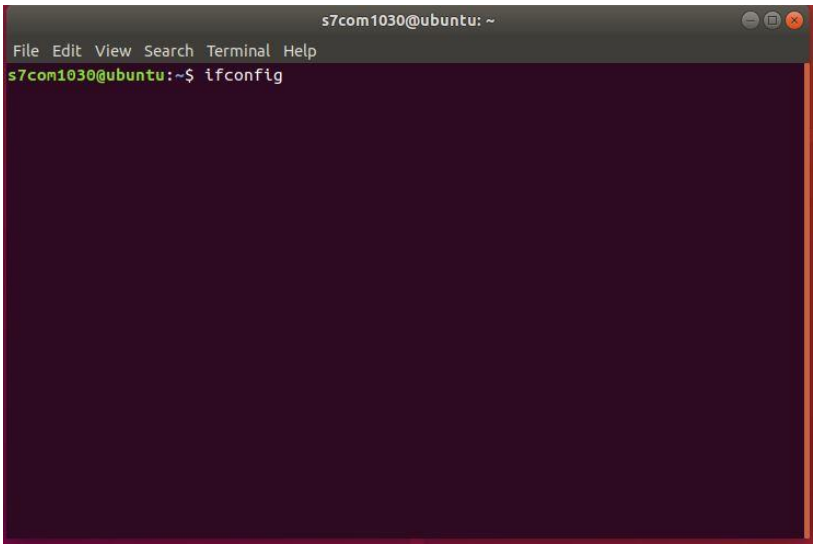
Following window must pop-up. Now we are working with Linux shell environment.



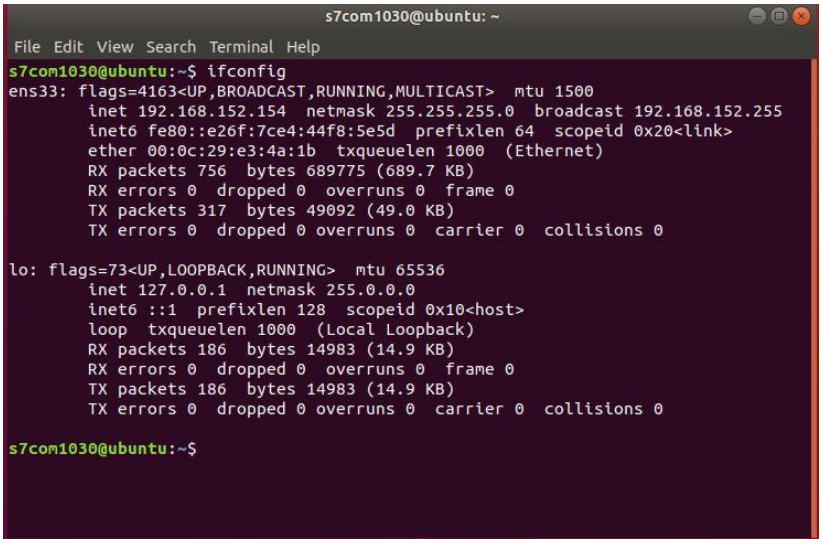
If you have difficulties in reaching this stage, please ask for help. If you are here now, we can continue to the next task at hand.

TASK 02 - Introduction to Mininet

First let's understand our VMs network configuration. On the terminal you have opened type the following. **ifconfig** and hit Enter.



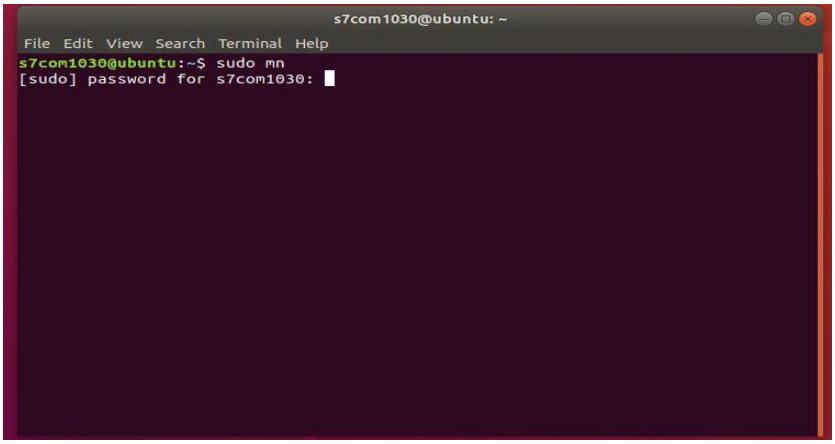
You should have something like the following.



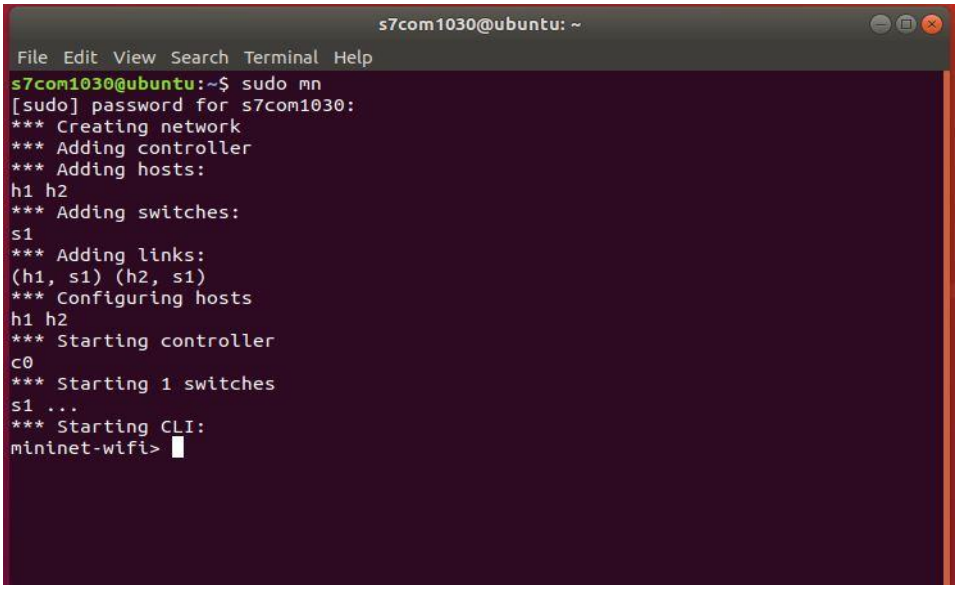
We have an Ethernet connection with the IP address 192.168.152.154. Yours may differ given the PC you are running. On your lab workbook, write down the following information from your output.

- What is the IP address you use?
- Write down the netmask
- Broadcast IP address

Upon completion now let's launch an emulated environment. On the same terminal window type **sudo mn** and shell will require a password from you. Type 123 and hit Enter (**mn** stands for mininet)



And you will see the following output.



Congratulations, you are in the Mininet environment now. On the Mininet shell now let's run some commands to understand the topology that it has created for us and later we will discuss how Mininet escapes a simulated environment. From now on the commands will be typed on the mininet shell till this lab sheet tells otherwise. First command to run is,

2.1)mininet-wifi> net

This command will print out the interfaces connected to each-other.

You should get the following output.

```
mininet-wifi> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
```

Now run the following command.

2.2)mininet-wifi> links

This command will print out the link status and the relevant ethernet interface that they are connected to.

You should get the following output.

```
mininet-wifi> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s1-eth2 (OK OK)
```

2.3)mininet-wifi> dump

This command is used to dump information of the hosts, controllers and switches connected to the network including their processor ID, IP address and active interfaces.

You should get the following output.

```
mininet-wifi> dump
<Host h1: h1-eth0:10.0.0.1 pid=2546>
<Host h2: h2-eth0:10.0.0.2 pid=2548>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=2553>
<Controller c0: 127.0.0.1:6653 pid=2539>
```

2.4)mininet-wifi> h1 ifconfig

```
mininet-wifi> h1 ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.1  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fe80::5805:38ff:fe9a:2203  prefixlen 64  scopeid 0x20<link>
        ether 5a:05:38:9a:22:03  txqueuelen 1000  (Ethernet)
        RX packets 40  bytes 4181 (4.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 936 (936.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

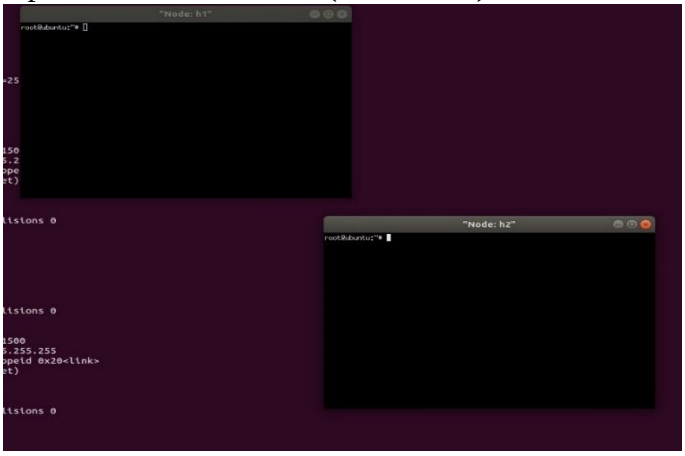
2.5)mininet-wifi> h2 ifconfig

```
mininet-wifi> h2 ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.2  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fe80::f80b:ecff:fe8b:201d  prefixlen 64  scopeid 0x20<link>
        ether fa:0b:ec:8b:20:1d  txqueuelen 1000  (Ethernet)
        RX packets 40  bytes 4181 (4.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 936 (936.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

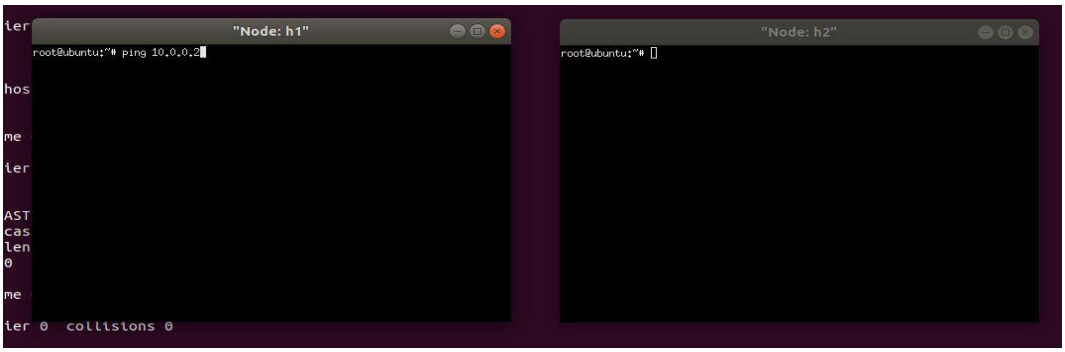
-> Based on the results you have gathered from the above commands (2.1-2.5), draw out the network topology, marking the ethernet interfaces, ip addresses, netmask and names of devices.

2.6) mininet-wifi> xterm h1 h2 this will open two additional shell commands which each represents the two nodes (PC/ HOST) that have been created by Mininet.

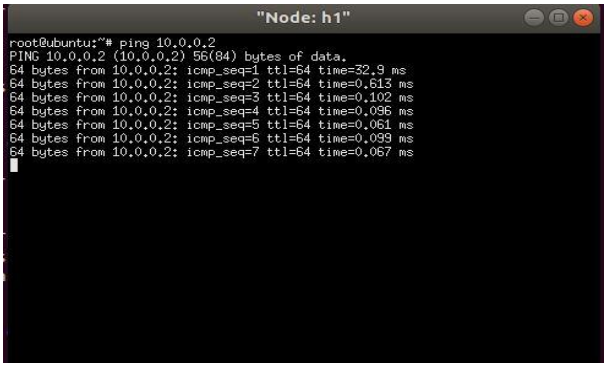


Now let's see if we can ping the host at the other end. On the terminal of H1 type the following as represented and hit Enter.

2.7) # ping 10.0.0.2



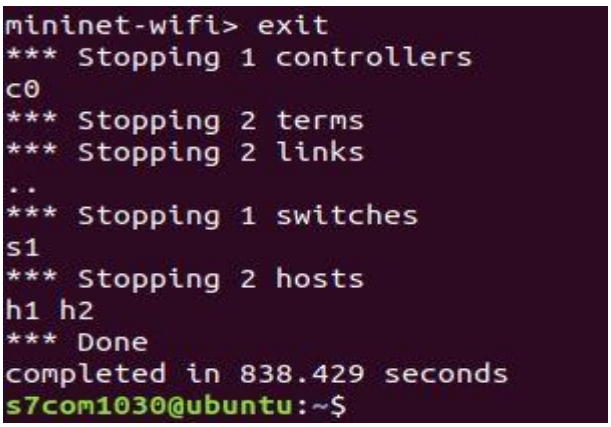
Confirm the below output



Congratulations your Ping is successful and you can ping the other node at 10.0.0.2 being 10.0.0.1. Press CTRL + C to cancel the ping stream. Type **Exit** on both terminal to exit h1 and h2.

Let's exit Mininet Type

2.8) Mininet-wifi> exit and hit enter.



Now that we have created a simple linear topology, let's create few other topologies like tree and linear with more hosts and switches.

TASK 3 – HOME WORK (EXTRA COMMANDS AND TOPOLOGIES TO TRY ON)

On the terminal type the following.

$ sudo mn --topo=single,4 hit enter. The following should be your output.

```
s7com1030@ubuntu:~$ sudo mn --topo=single,4
[sudo] password for s7com1030:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet-wifi>
```

Now repeat exercises 2.1 to 2.8 of above. If you have any questions, please feel free to ask. Let's run another topology.

$ sudo mn --topo=linear,4 hit enter. The following should be your output

```
s7com1030@ubuntu:~$ sudo mn --topo=linear,4
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (s2, s1) (s3, s2) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet-wifi>
```

Now repeat exercises 2.1 to 2.8 of above. If you have any questions, please feel free to ask. **Draw the topology in your Lab book marking the relevant information as earlier**. Let's run another topology.

Let's deploy a tree topology.

$ sudo mn --topo=tree,2,2 hit enter. The following should be your output

```
s7com1030@ubuntu:~$ sudo mn --topo=tree,2,2
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, s2) (s1, s3) (s2, h1) (s2, h2) (s3, h3) (s3, h4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet-wifi>
```

Now repeat exercises 2.1 to 2.8 of above. If you have any questions, please feel free to ask. **Draw the topology in your Lab book marking relevant information as earlier.** Let's run another topology

TASK 04 – Introduction to Python API.

Click on the icon Files in the left top corner. Go inside the folder, 7COM1030/Lab1.

You will find python script named **Lab01Ex01.py** Open that. Double click would suffice.

locate the section # **ADD HOSTS AND SWITCHES TODO** and type in the following. Python API is indentation specific. Where the indentation break will cause an error. So, type directly under the # sign. Now type in the following.

Host1 = self.addHost('h1')

Host2 = self.addHost('h2')

Switch1 = self.addSwitch('s1')

Next locate the section that reads, **# Add links TODO** and type in the following.

self.addLink(Host1, Switch1)

self.addLink(Host2, Switch1)

Pay attention to indentation and type under the # sign. Your final script should look like this.

```
from mininet.topo import Topo
class MyTopo( Topo ):
    "Simple topology example."
    def __init__( self ):
        "Create custom topo."

        # Initialize topology
        Topo.__init__( self )

        # Add hosts and switches TODO
        Host1 = self.addHost( 'h1' )
        Host2 = self.addHost( 'h2' )
        Switch1 = self.addSwitch('s1')

        # Add links TODO
        self.addLink( Host1, Switch1 )
        self.addLink( Host2, Switch1 )

topos = { 'mytopo': ( lambda: MyTopo() ) }
```

Save the python file by CTRL + S or use the save button on the right top corner. Now that we have filled our first python script let's feed it to Mininet. On the terminal that we opened previously type in the following

$ cd 7COM1030/Lab1/

Now your current working directory have changed to the above. You should see the following shell terminal now.

```
s7com1030@ubuntu:~$ cd 7COM1030/Lab1/
s7com1030@ubuntu:~/7COM1030/Lab1$
```

Now let's launch Mininet from here, type in the following.

$ sudo mn --custom Lab01Ex01.py --topo mytopo your output should be as follows. Enter the password if it prompted, (password: 123) and you should see the following output.

```
s7com1030@ubuntu:~/7COM1030/Lab1$ sudo mn --custom Lab01Ex01.py --topo mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet-wifi>
```

We have successfully created an emulated environment using a custom script that represents our first topology.

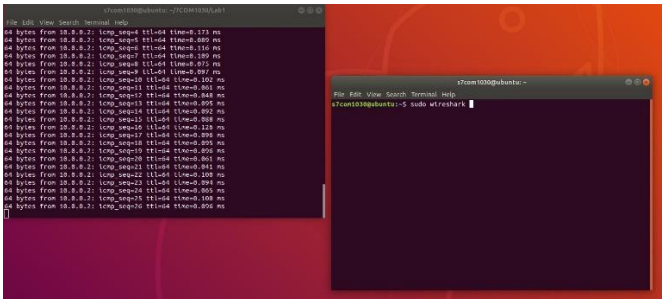
On the Mininet terminal type the following.

Mininet-wifi> h1 ping h2

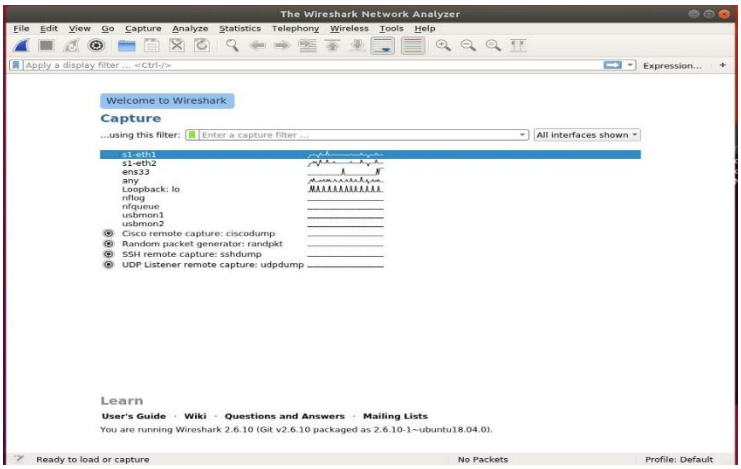
```
s7com1030@ubuntu:~/7COM1030/Lab1$ sudo mn --custom Lab01Ex01.py --topo mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet-wifi> h1 ping h2
```

You should see ICMP traffic passing the network. Now let it run like that. Without killing the stream, let's open another terminal. Go to **Activities** and type **Terminal** and hit enter. On the second terminal type in the following.

$ sudo wireshark hit enter, type in the password 123



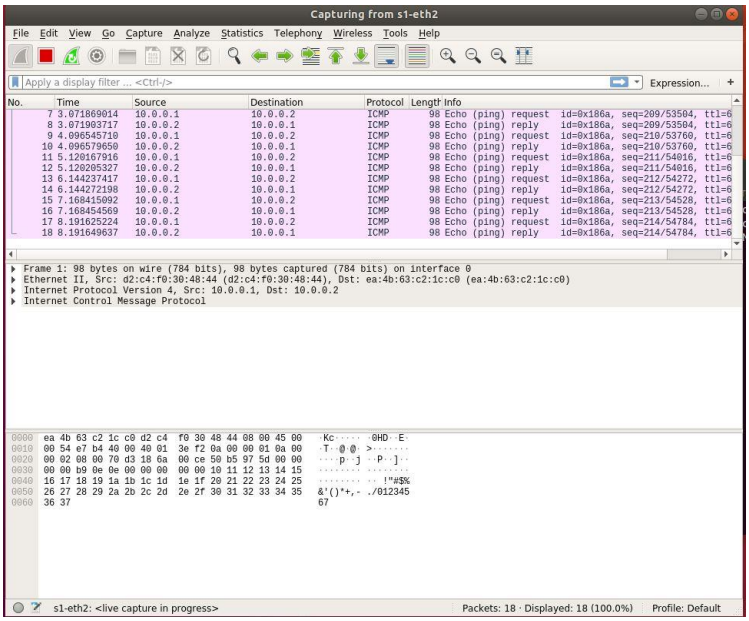
Now following environment will pop up.



Double click on the interface that reads

s1-eth2

You should see the below output.



We have successfully captured traffic from h1 – h2. Only an emulated environment can do this experiment.

Click on stop. Close wireshark on the prompt click **Quit Without saving.**

On the ICMP stream type **CTRL + C** and type

Mininet > h2 ping h1

Observe the changes in the output of the Wireshark, see how the host and the destination changed with the change of commands. Observe the relevance of the protocol. Since ICMP traffic is being sent out in the environment, Protocol prints out as ICMP. Observe the sequence of Request and Reply from the Wireshark output.

Write down any subsequent two lines in your lab work book. You may amend the time field to three decimals. We will have more protocols tested in future, you will be able to see the difference on the network capture file.

Mininet-wifi> exit

To power off your VM top most right corner you will see a power button, click on it and select **Poweroff.**

We have successfully completed the lab session. WELL DONE!!

TASK 05 – HOME WORK (MORE EXERCISES ON PYTHON API)

Write down python scripts to emulate the topologies at TASK3. Feed them to the Mininet API using the commands as above.

Bring the answers of all homework and extra exercises to next week's lab session.

TASK 06- Extra Exercises

Upon launching the Mininet environment run the command **ifconfig** on a new terminal. Can you observe the new interfaces differentiating from the original ethernet and loopback interface at the beginning of Task 2? Mininet creates new interfaces and assign them resources to emulate an environment escaping a simulated software based environment.

On the mininet type in the following.

Mininet-wifi> **dump**

Notice the **pid** values, write them down. Now let's open another shell terminal. Go to activities and type terminal hit enter.

$ ifconfig

You should see the created topology have assigned IP address inside your OS. Now type in the following

ps -ef | grep 5814 change 5814 to the values you wrote down hit enter, you will see how Linux have created real switches and hosts inside the OS. Therefore, Mininet escapes simulations and thereby known as an emulated environment.

Please work on Task3, Task 5 and Task 6 on your own. We will upload the answers to the exercises by next week including the topology diagrams and python code.

Reference

1. http://mininet.org/
2. http://mininet.org/api/annotated.html
3. http://mininet.org/walkthrough/
4. https://www.wireshark.org/
5. https://www.wireshark.org/#learnWS