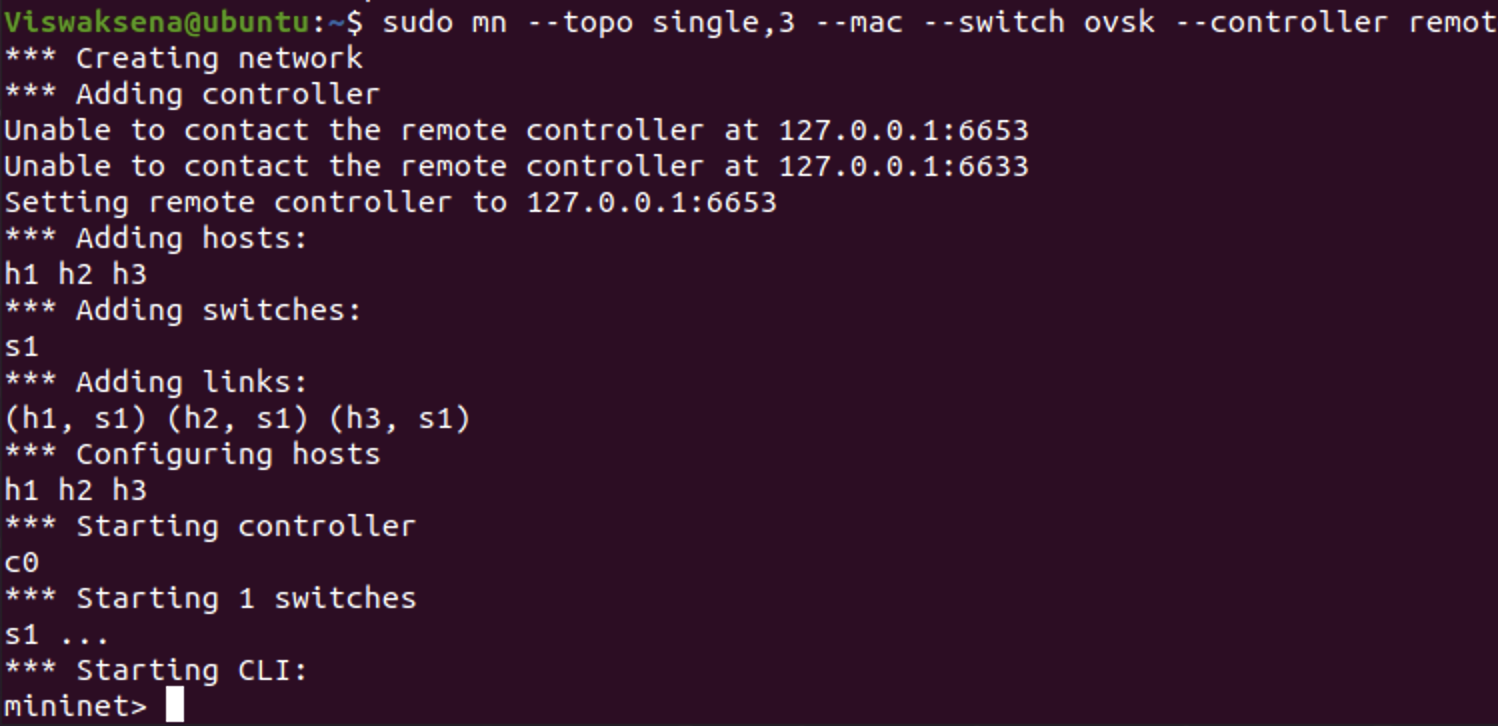
AUM Amriteswaryai Namah

Assignment 2 on Mininet

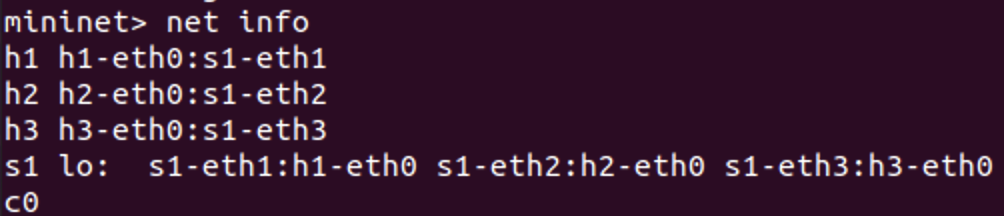
Name: J Viswaksena Roll.No: AM.EN.UAIE21035

Deliverable

1. Screenshot for Pingall before and after adding flow table entries.
2. Screenshot for dump-flow commands
3. Run this command to create a simple host with 3 hosts and 1 Open vswitch and a remote controller sudo mn --topo single,3 --mac --switch ovsk --controller remote

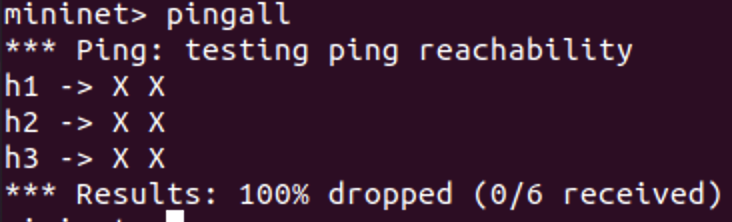


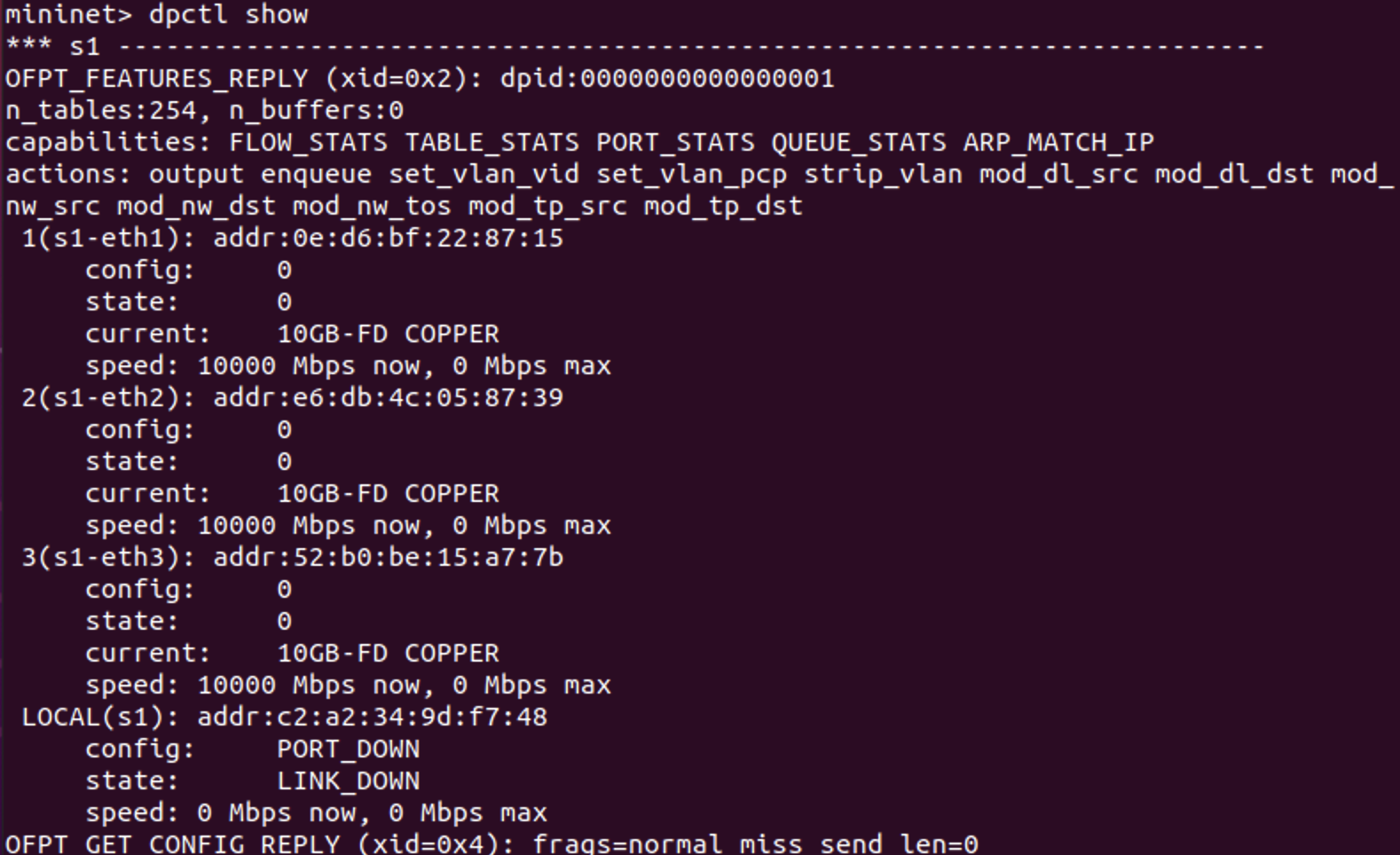
1. Run a net command and see all the interfaces and find out how the switches are connected to each other.

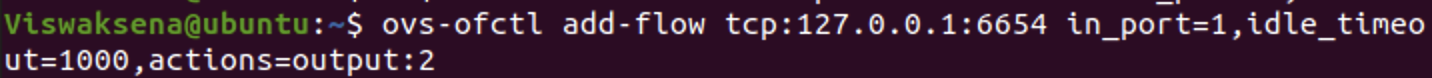


1. Run a pingall command on mininet prompt and write the output. Explain the reason why you got the output that way.

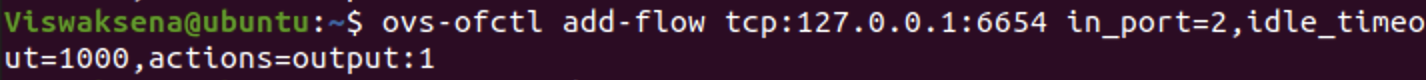
Due to absence of entries in flow table, packets are struct which host it need to reach. So for this reason 100% packet loss get occurs. To receive packets we need to give inputs to the flow table then they can do conversation.



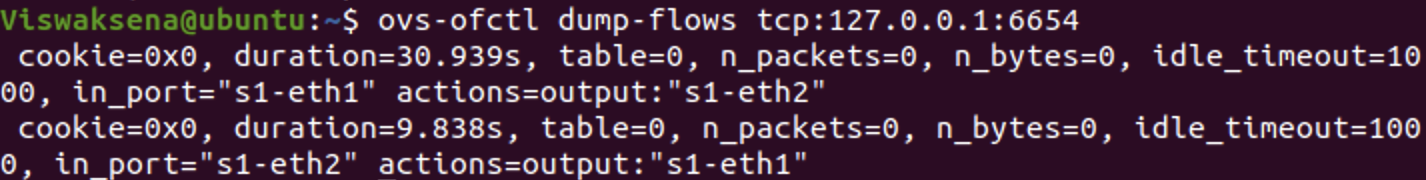
1. Next, you see the flow table at the switch s1
2. $ovs-ofctl add-flow tcp:127.0.0.1:6654 in\_port=1,idle\_timeout=1000,actions=output:2



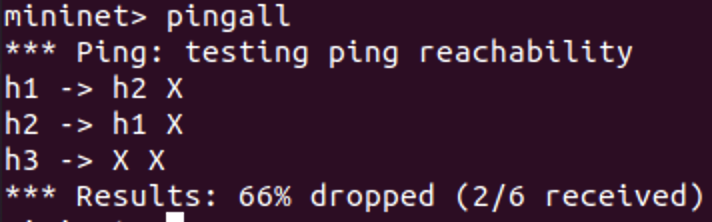
1. $ovs-ofctl add-flow tcp:127.0.0.1:6654 in\_port=2,idle\_timeout=1000,actions=output:1



1. $ovs-ofctl dump-flows tcp:127.0.0.1:6654

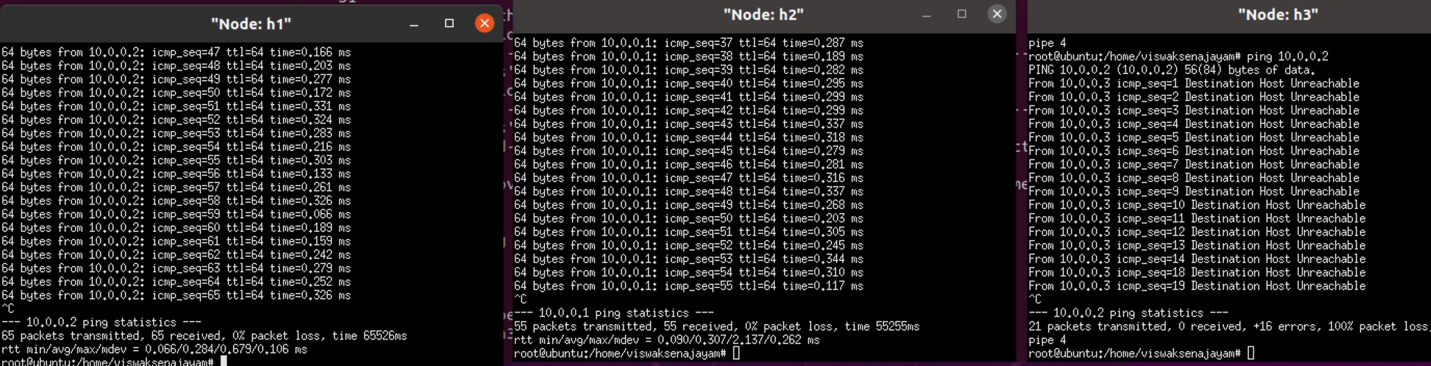


1. Mininet> pingall

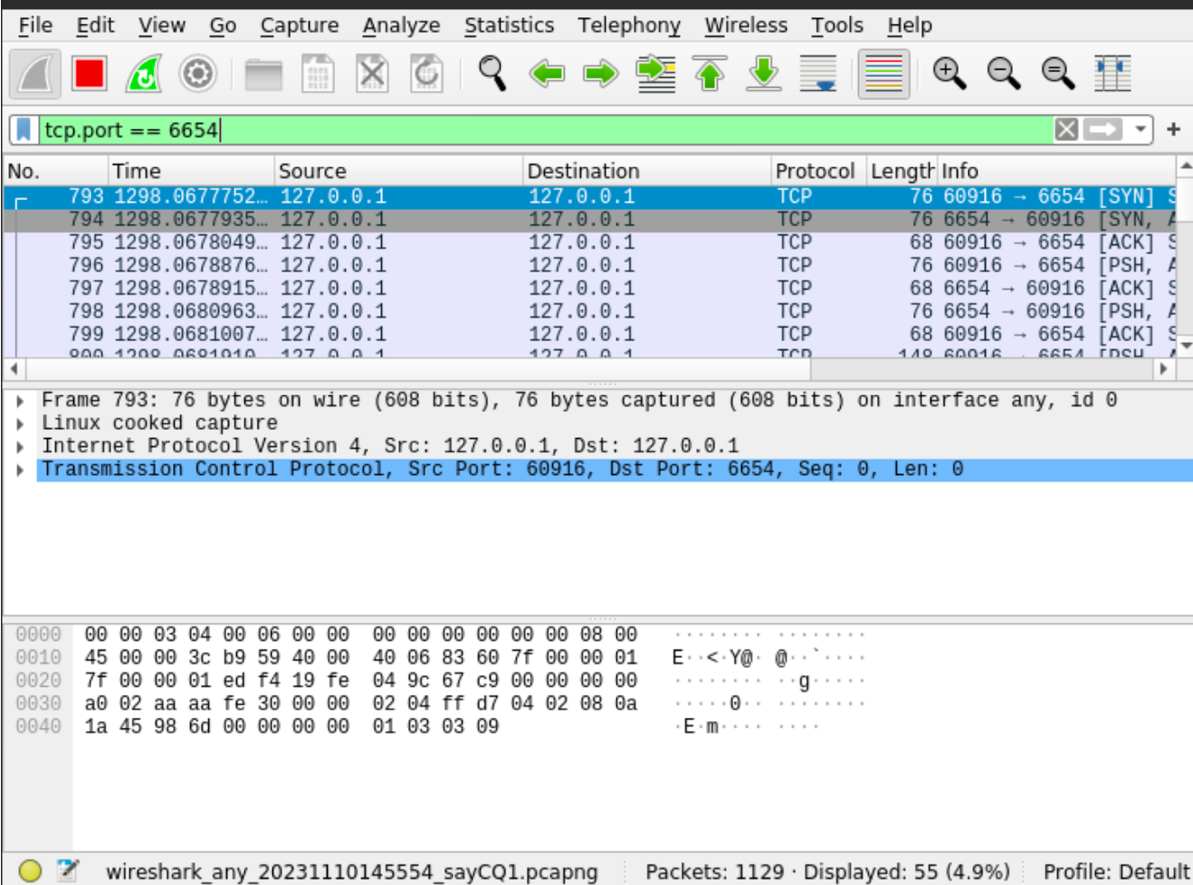


1. **h1 and h2 are now connected. [ If there are switches, Port 6655 is used for Switch S2]**

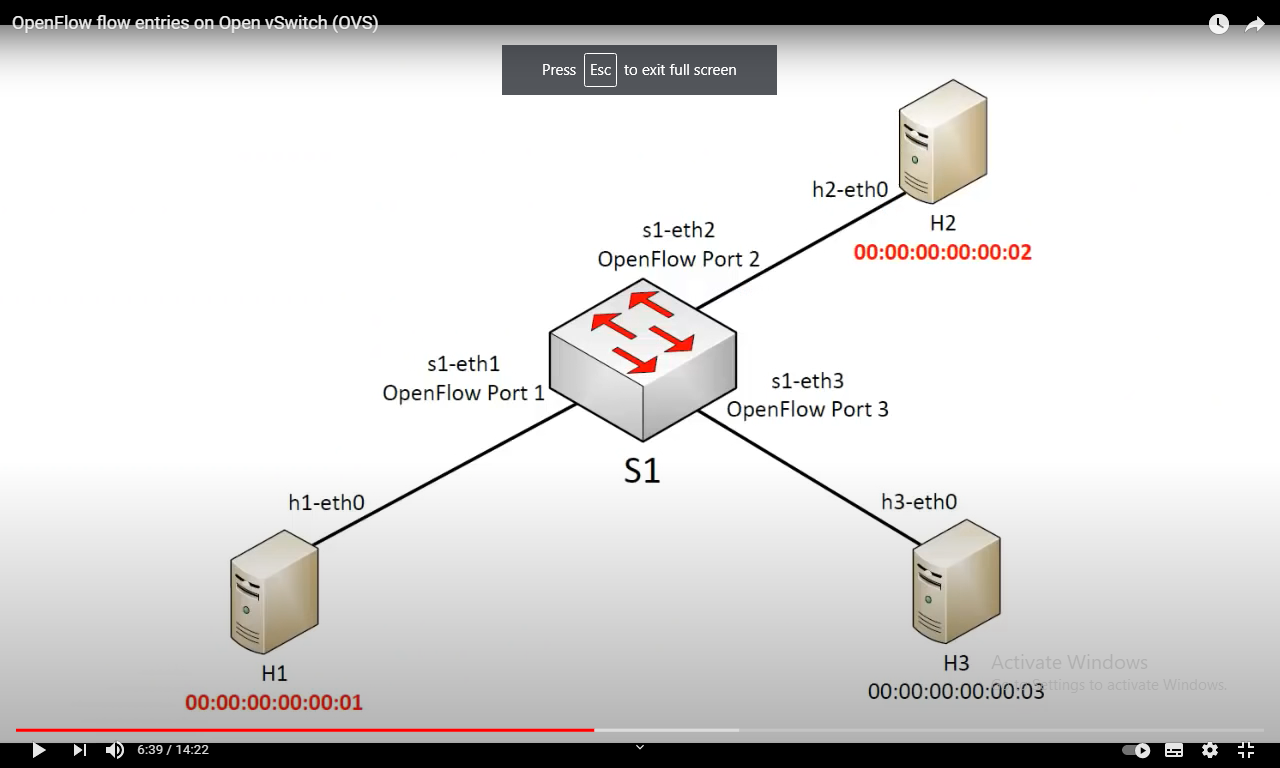
From the below diagram we can say that the host 1 and host 2 are connected transmitted packets from h1 to h2, we get 0% packet loss, same from h2 to h1. But from h3 to h2,h1 transmitted 21 packets then happens 100% packet loss. h3 to h1 either h2 are not connected. Because we only given entries to flow table only for h1 and h2 only.



1. Start Wireshark on a window and before doing the next dump-flows command, start capturing packets.



1. $ovs-ofctl dump-flows tcp:127.0.0.1:6654 and check the statistics on packet sent

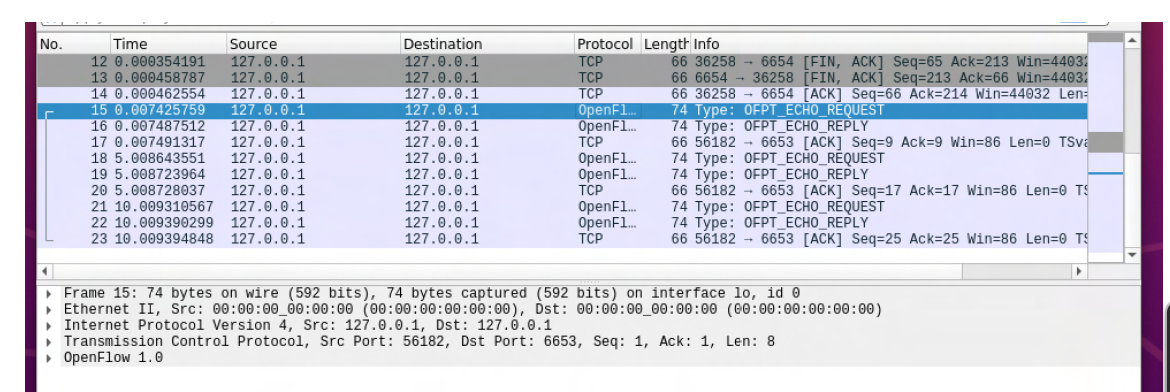


After adding links

1. Run Wireshark on another window and Identify all the messages corresponding to **OpenFlow protocol**. Take a screenshot and mark two messages – one from switch to Controller and another one Controller to switch

Sample is shown here



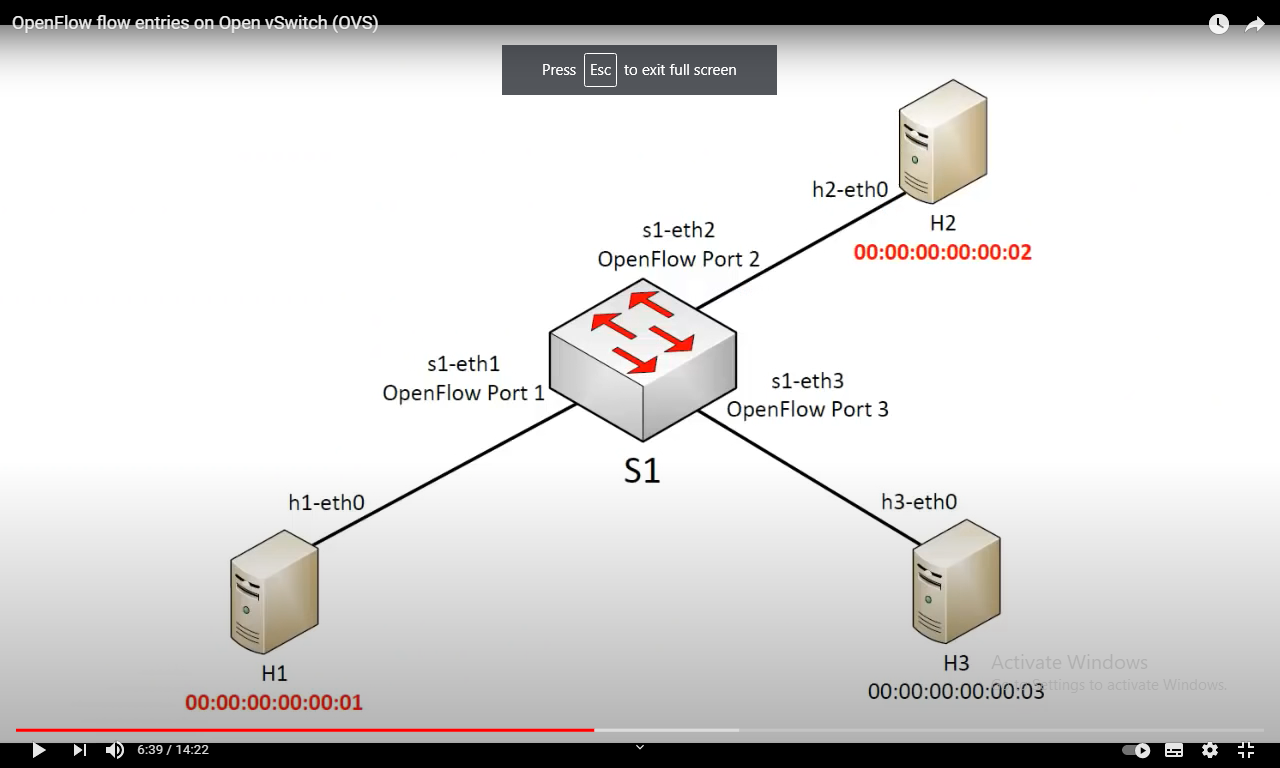
1. Continue the exercise to completely install flow table for all hosts.
2. Ping every pair of hosts and make sure they work. You may have to add the following for the switch which directs packets destined for a host to a particular port. It does it based on MAC address and it cannot be done using ip\_src or ip\_dst only. In the below commands,

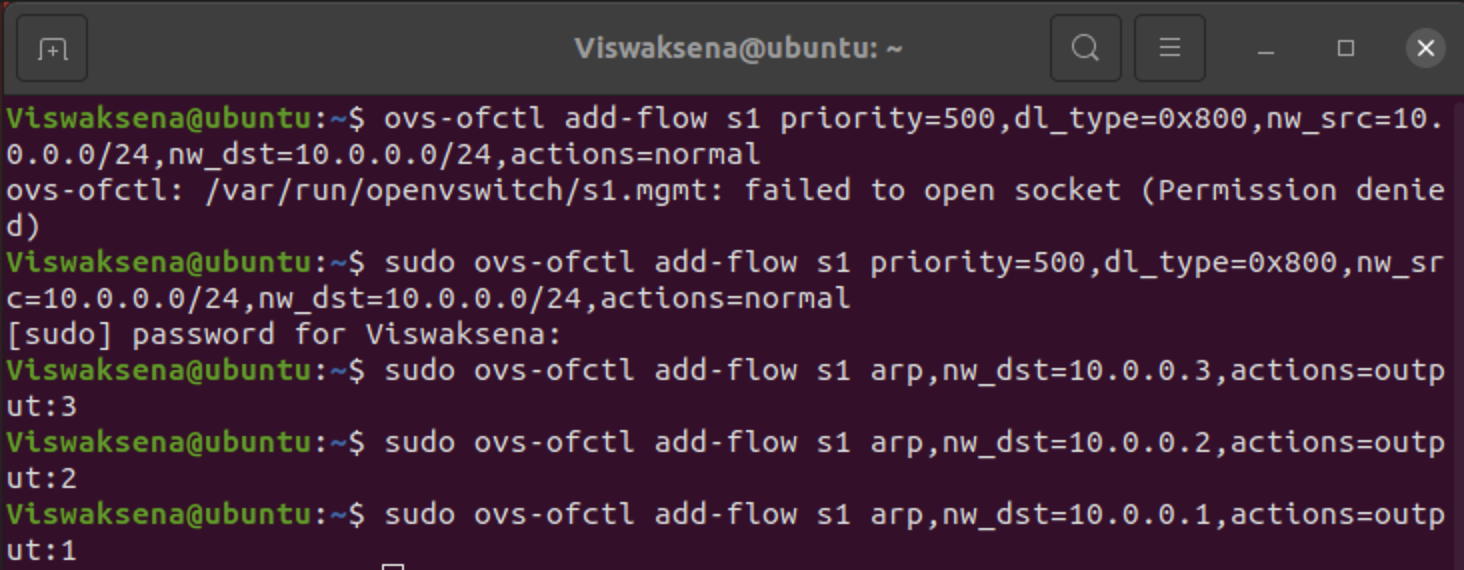
dl\_src and dl\_dst give the source MAC address and destination MAC address

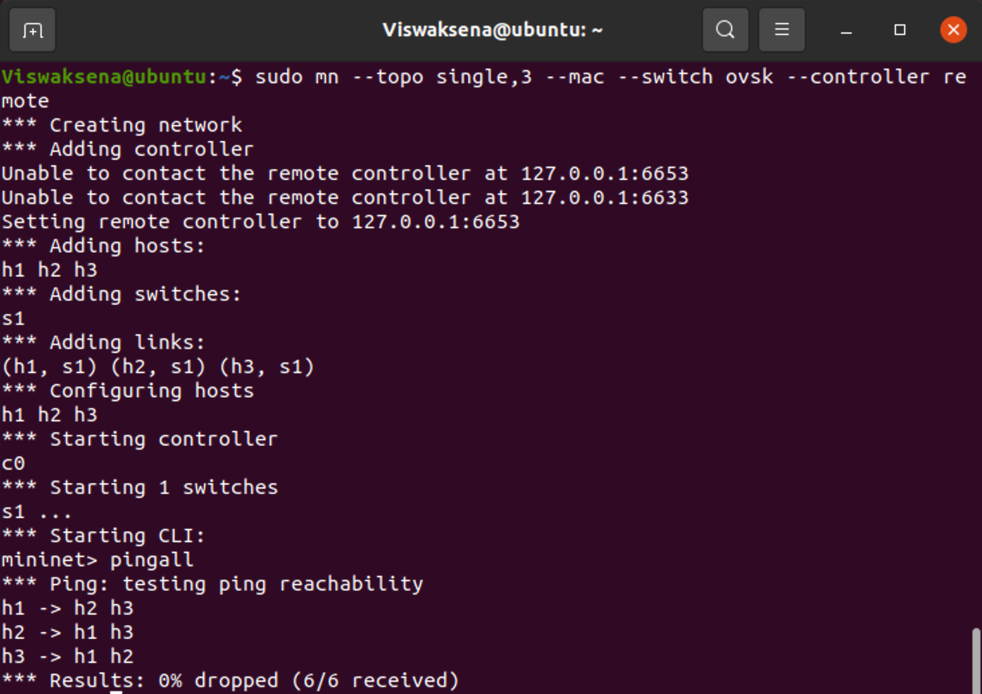
$dpctl add-flow tcp:127.0.0.1:6654 dl\_dst=0:0:0:0:0:1,idle\_timeout=1000,actions=output:1

$dpctl add-flow tcp:127.0.0.1:6654 dl\_dst=0:0:0:0:0:2,idle\_timeout=1000,actions=output:2

$dpctl add-flow tcp:127.0.0.1:6654 dl\_dst=0:0:0:0:0:3,idle\_timeout=1000,actions=output:3







**Additional Exercise for Assignment 2 –** for those who have already completed.

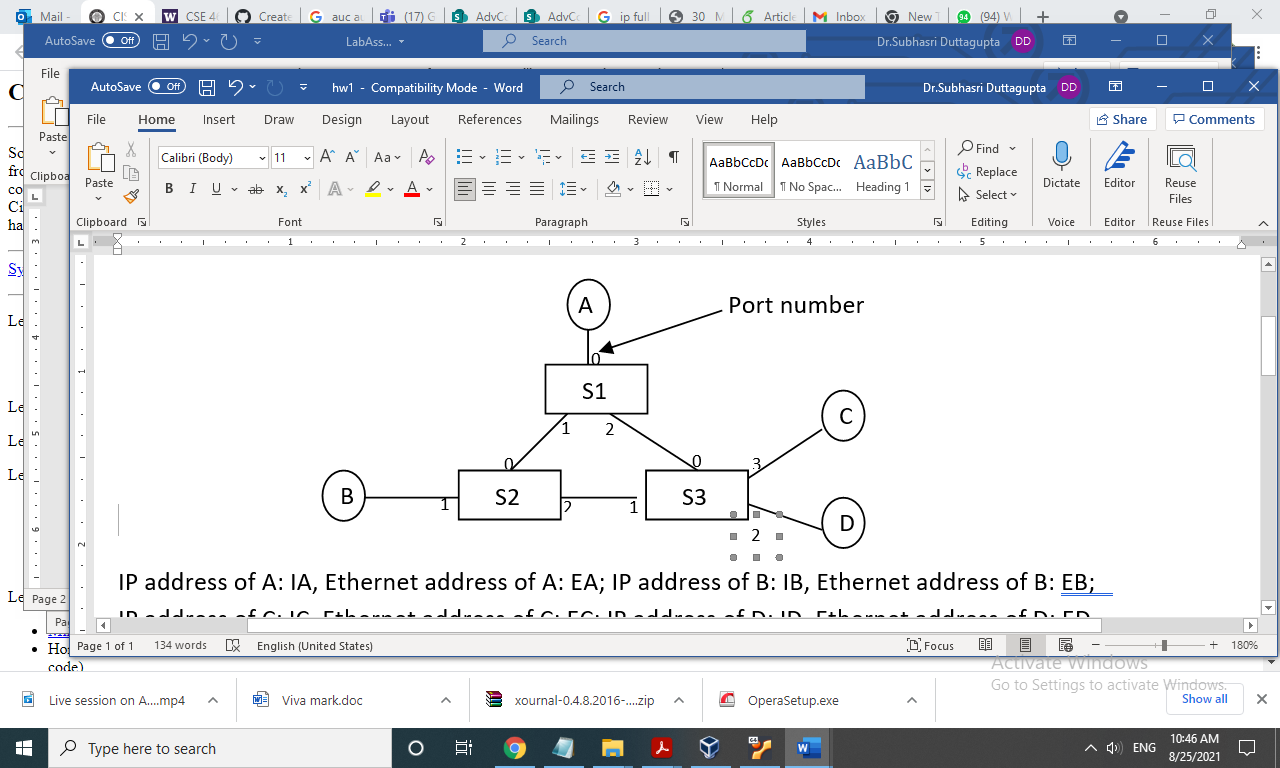
Assuming that you are managing a network with the following topology where A can be the Controller. The Interface numbers can be different. By default, eth0 of S1 is connected to the controller.

Give -- mac option for invoking the mininet. You can assign specific IP address to a host through Python API.

Please see the examples in the examples folder under your mininet installation directory. For me it is

/usr/lib/python2.7/dist-packages/mininet

Port number

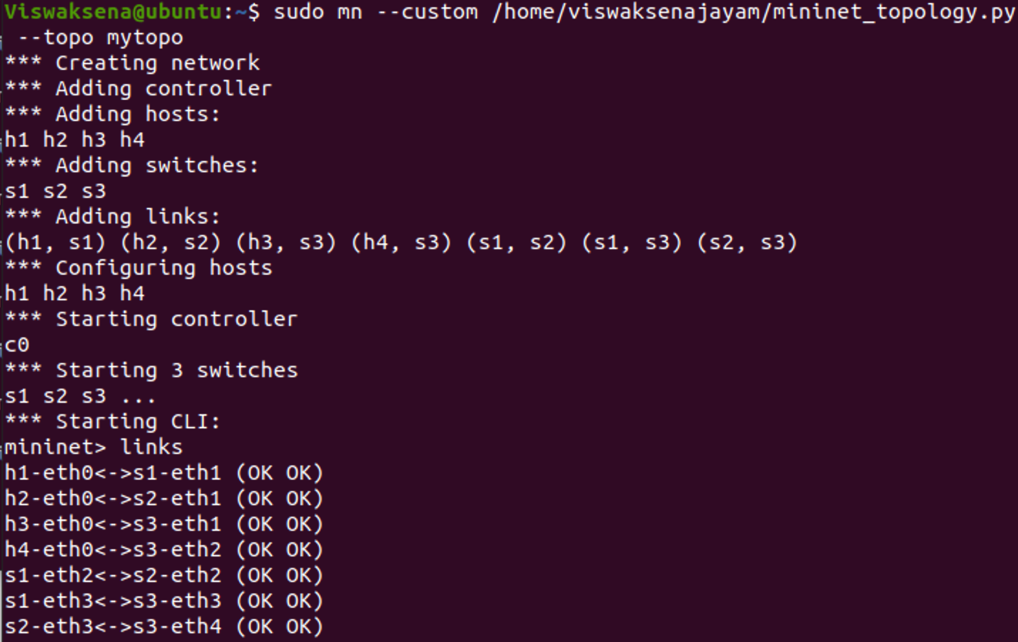


Assume IP address of A, B, C, D: 10.0.0.1, 10.0.0.2, 10.0.0.3,10.0.0.4

Specify the Openflow rules on switches S1, S2, S3 that enforce the following policies. Make sure that all needed communication including acknowledgement can happen.

* Nodes A, B, and C can talk to one another freely.
* Node D has access to ports 22 and 80 of Nodes A and B, but nothing else.

Node D and Node C cannot talk to each other.



Code:-

