MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

Santosh,Tangail-1902

LAB REPORT

Lab Report No : 04

Lab Report name : SDN controllers and mininet

Course Title : Computer Networks

Course Code : ICT- 3207

Date of Performance : 05/02/2021

Date of Submission :

Submitted by,

Name: Tanvir Ahmed and Farhana Afrin Shikha

ID: IT-18043 and IT-18038

Session: 2017-18

3rd year 2nd semester

Dept. of ICT

Submitted to,

Nazrul Islam
Assistant Professor
Dept. of ICT
MBSTU.

Theory:

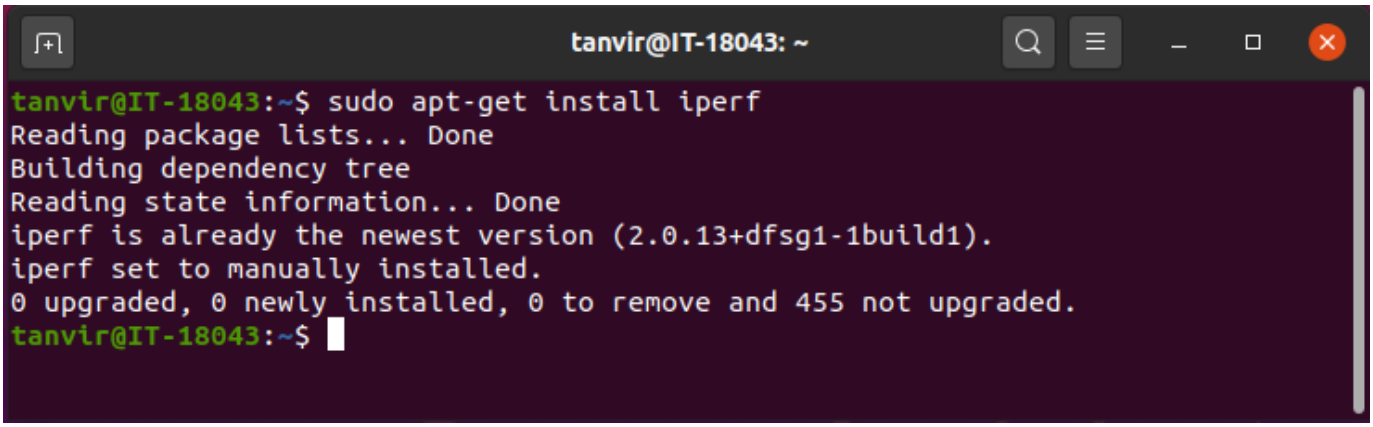
Traffic Generator:

iPerf : iPerf is a tool for active measurements of the maximum achievable bandwidth on IP networks. It supports tuning of various parameters related to timing, buffers and protocols (TCP, UDP, SCTP with IPv4 and IPv6). For each test it reports the bandwidth, loss, and other parameters.

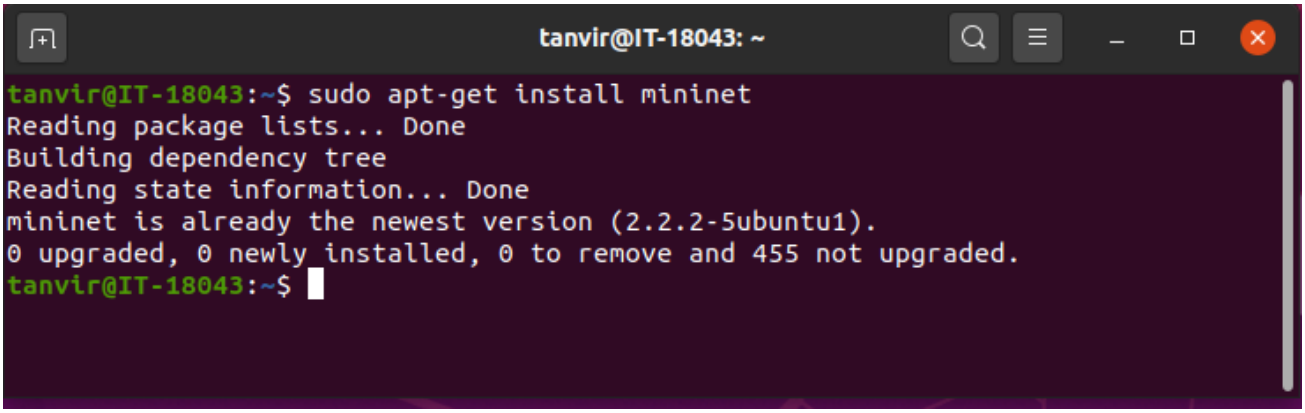
Mininet:

Mininet creates a realistic virtual network, running real kernel, switch and application code, on a single machine (VM, cloud or native) Because you can easily interact with your network using the Mininet CLI (and API), customize it, share it with others, or deploy it on real hardware, Mininet is useful for development, teaching, and research. Mininet is also a great way to develop, share, and experiment with OpenFlow and Software-Defined Networking systems.

Install iperf:

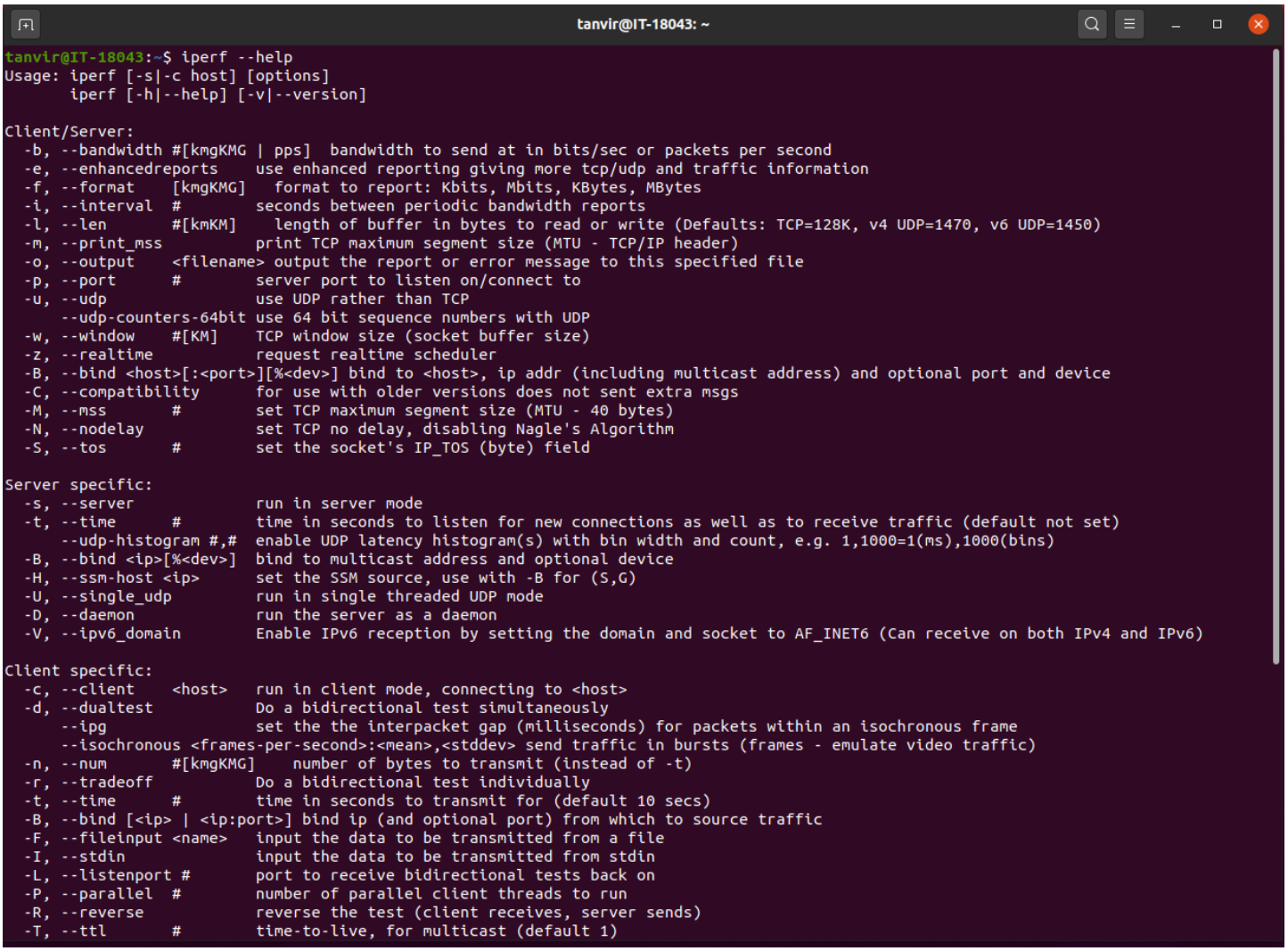


Install Mininet:



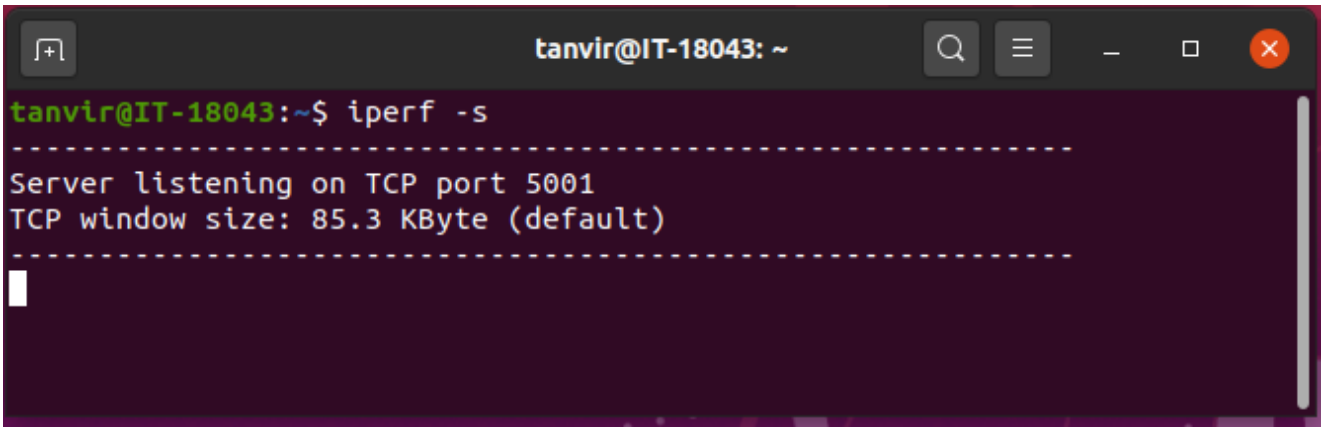
4. Exercises Exercise

4.1.1: Open a Linux terminal, and execute the command line iperf --help. Provide four configuration options of iperf.

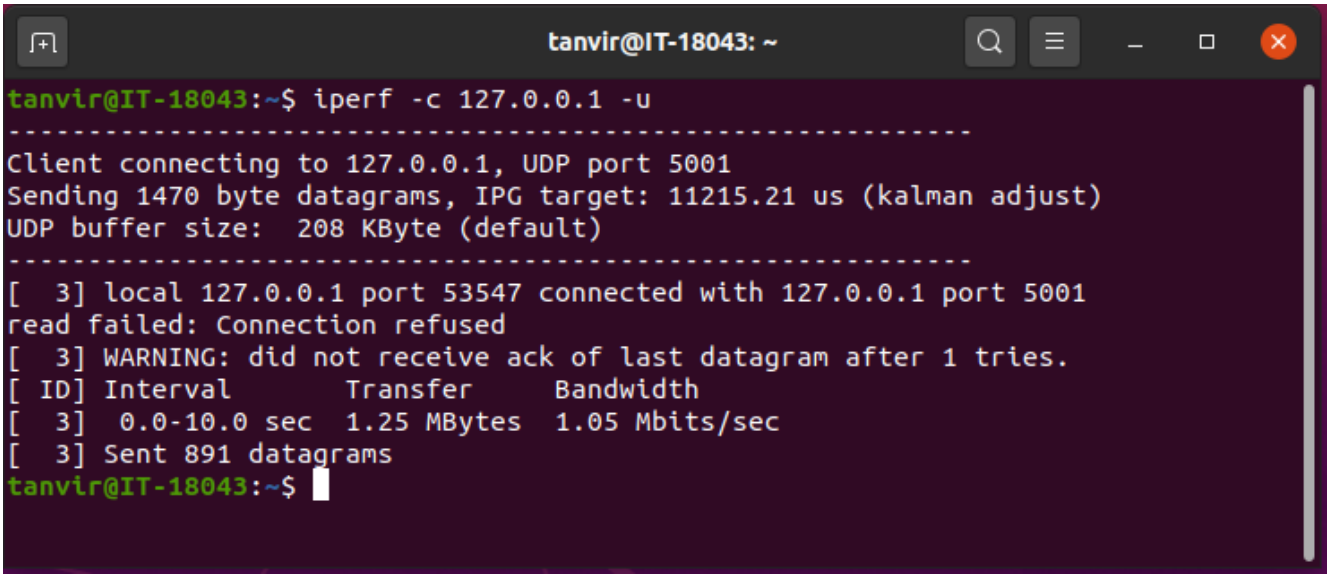


Exercise 4.1.2: Open two Linux terminals, and configure terminal-1 as client (iperf –c IPv4_server_address) and terminal-2 as server (iperf -s).

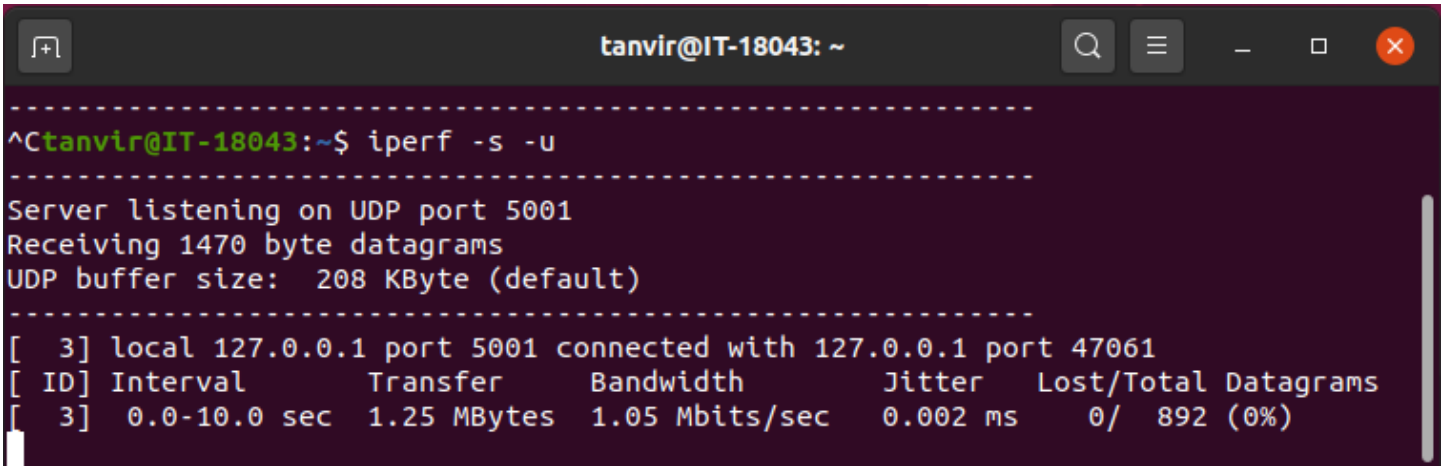
For terminal -1:

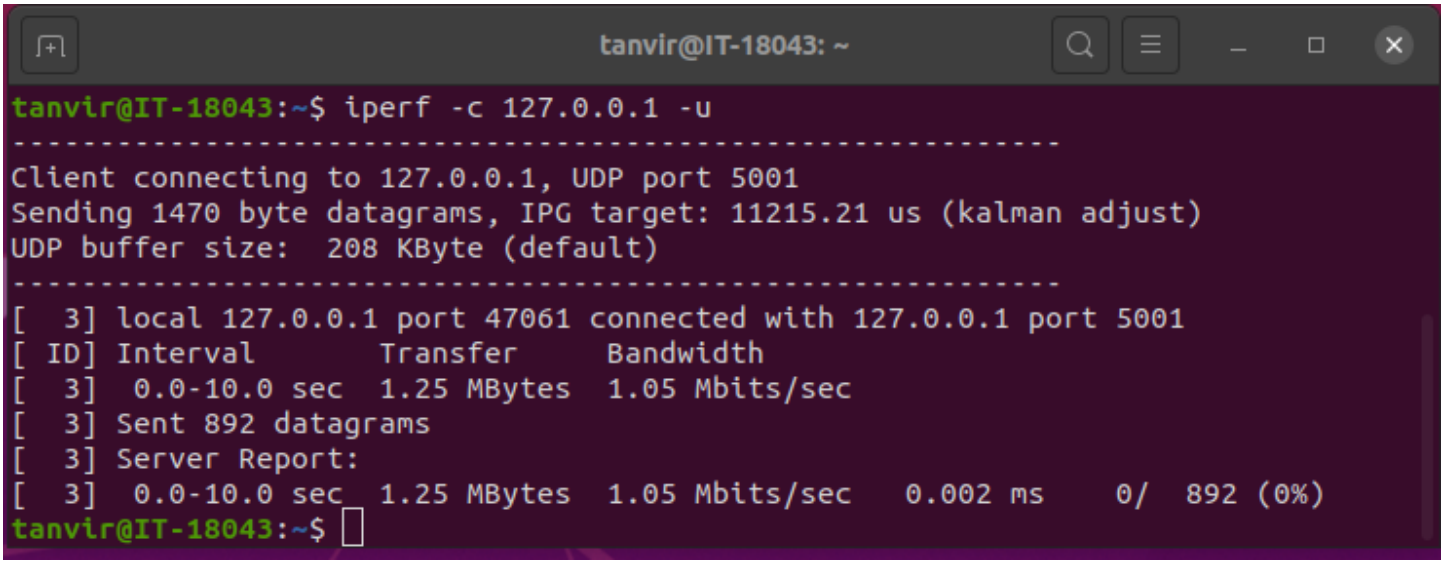


For terminal -2:



Exercise 4.1.3: Open two Linux terminals, and configure terminal-1 as client and terminal-2 as server for exchanging UDP traffic, which are the command lines? Which are the statistics are provided at the end of transmission?





Exercise 4.1.4: Open two Linux terminals, and configure terminal-1 as client and terminal-2 as server for exchanging UDP traffic, with:

o Packet length = 1000bytes

o Time = 20 seconds

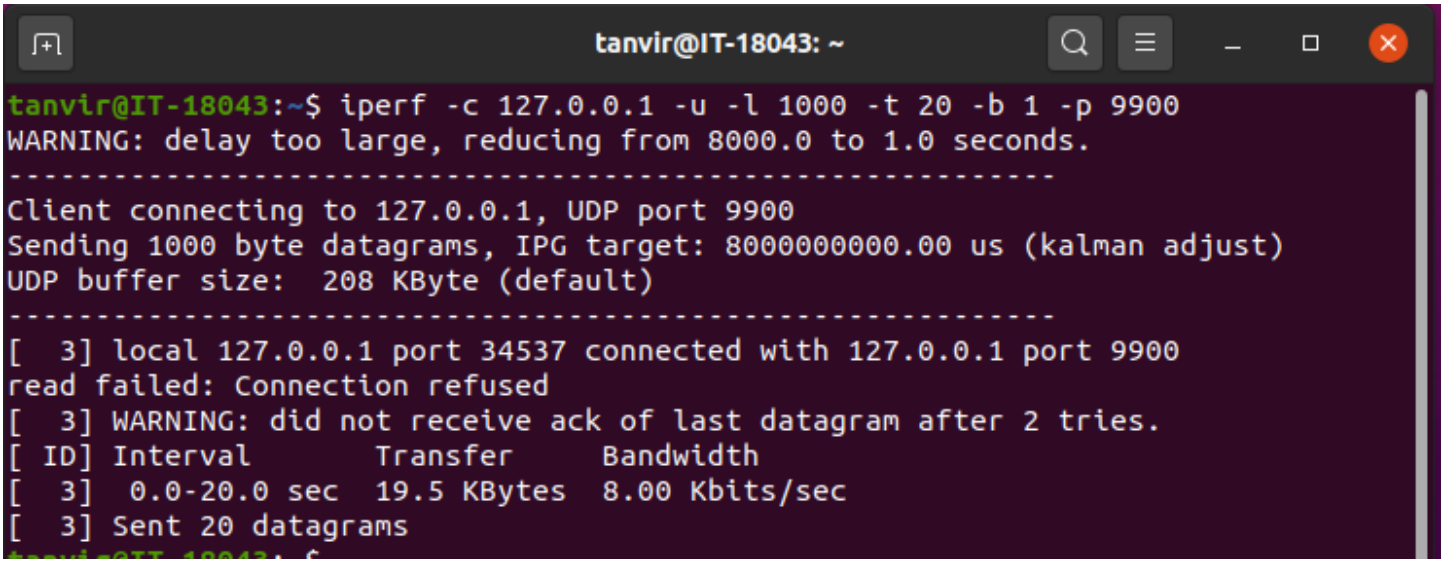
o Bandwidth = 1Mbps

o Port = 9900

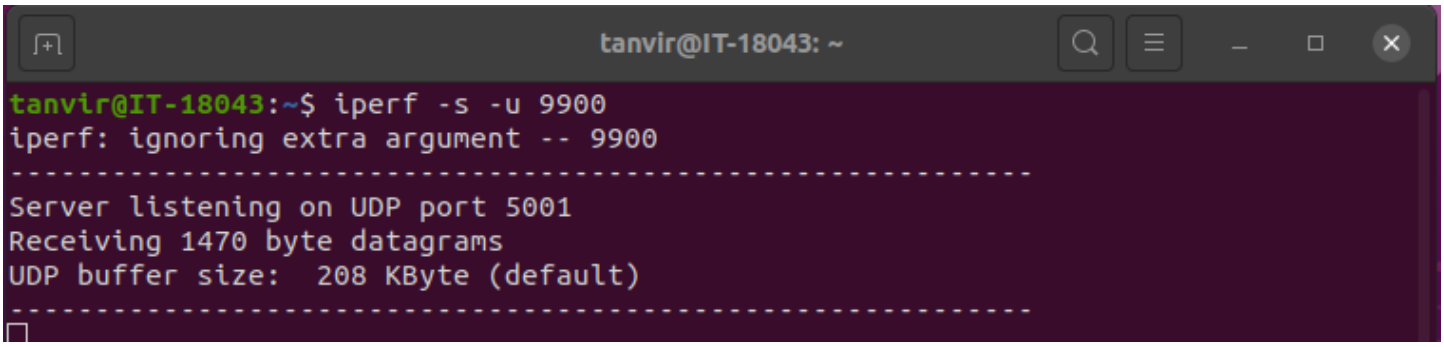
Which are the command lines?

The command lines are:

For terminal 1:



For terminal 2:

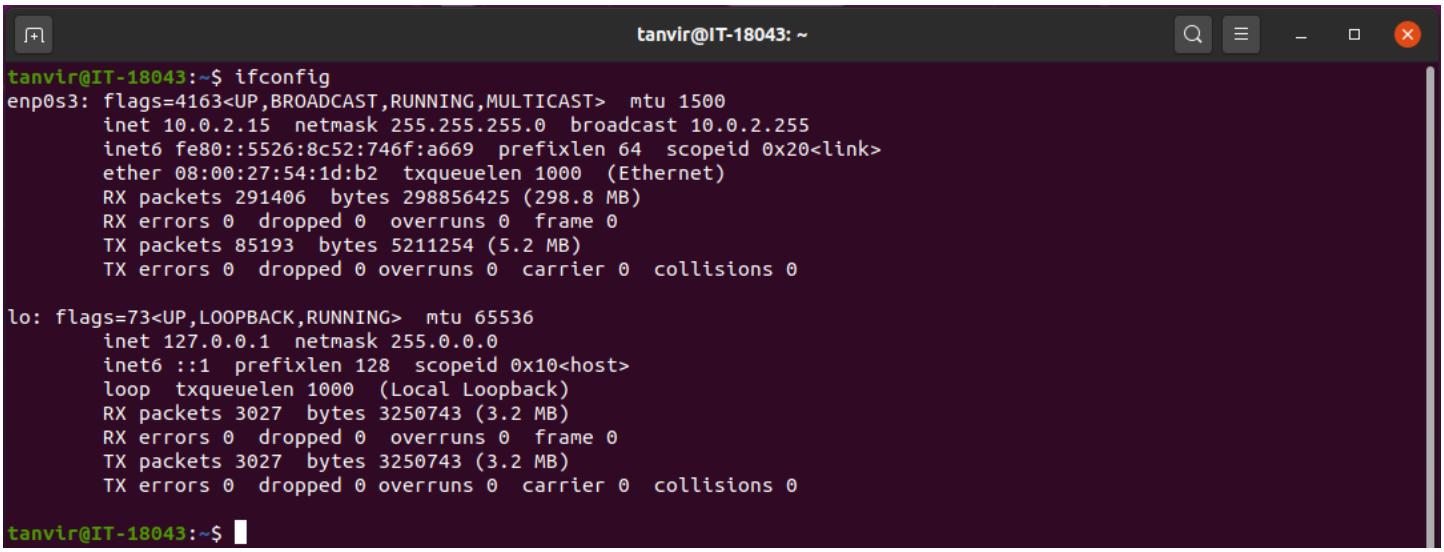


Using Mininet

Exercise 4.2.1: Open two Linux terminals, and execute the command line ifconfig in terminal1. How many interfaces are present?

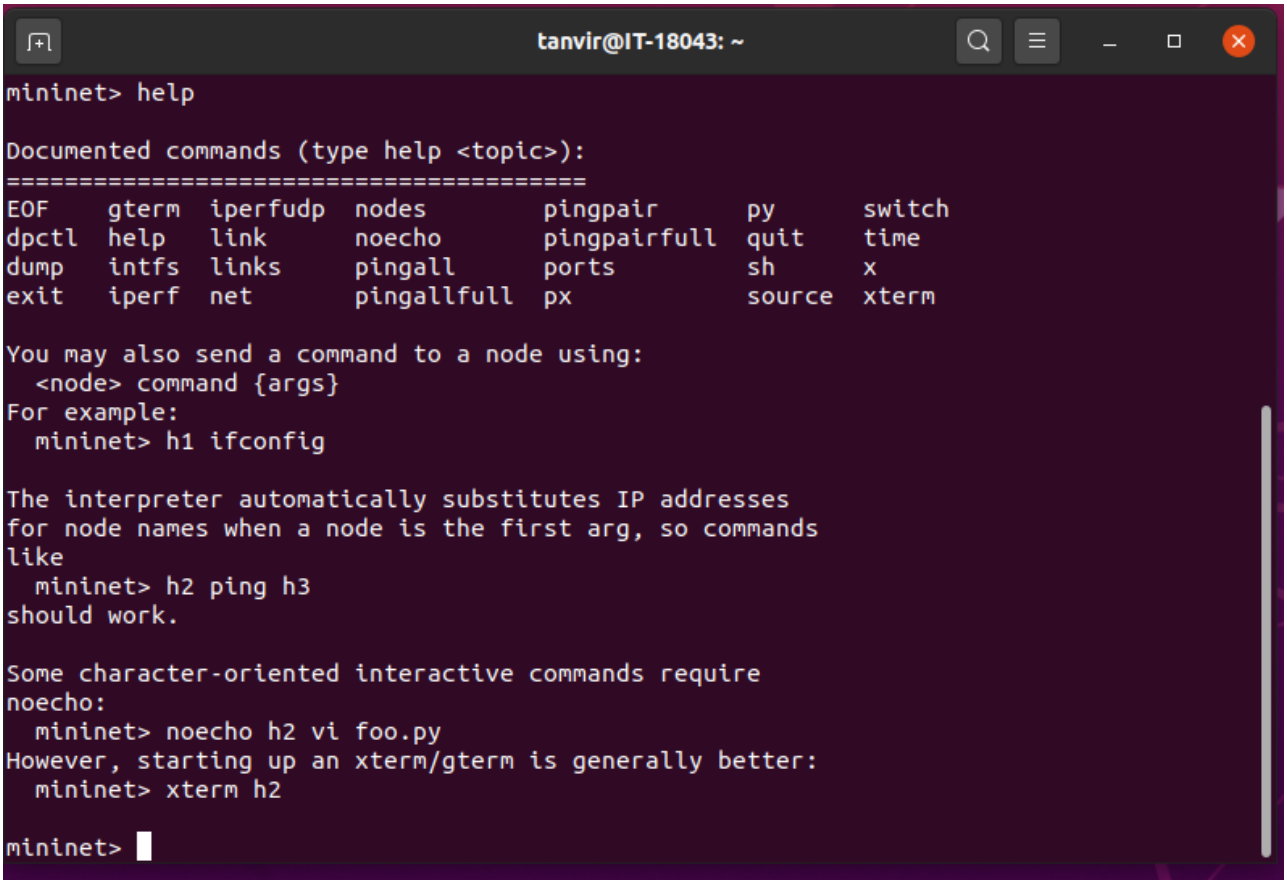
In terminal-2, execute the command line sudo mn, which is the output?

In terminal-1 execute the command line ifconfig. How many real and virtual interfaces are present now?



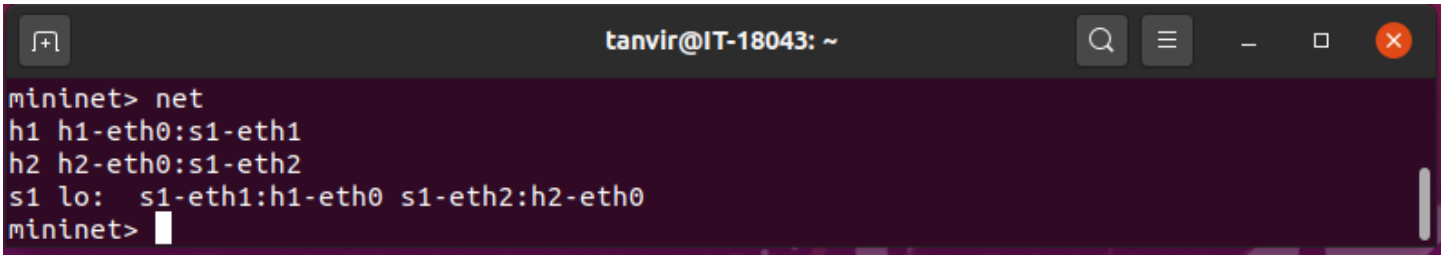
Exercise 4.2.2: Interacting with mininet; in terminal-2, display the following command lines and explain what it does:

mininet> help

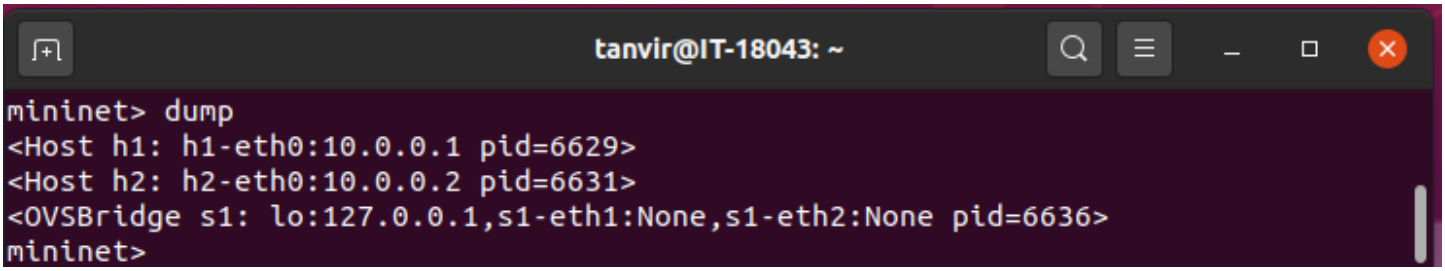


mininet> nodes

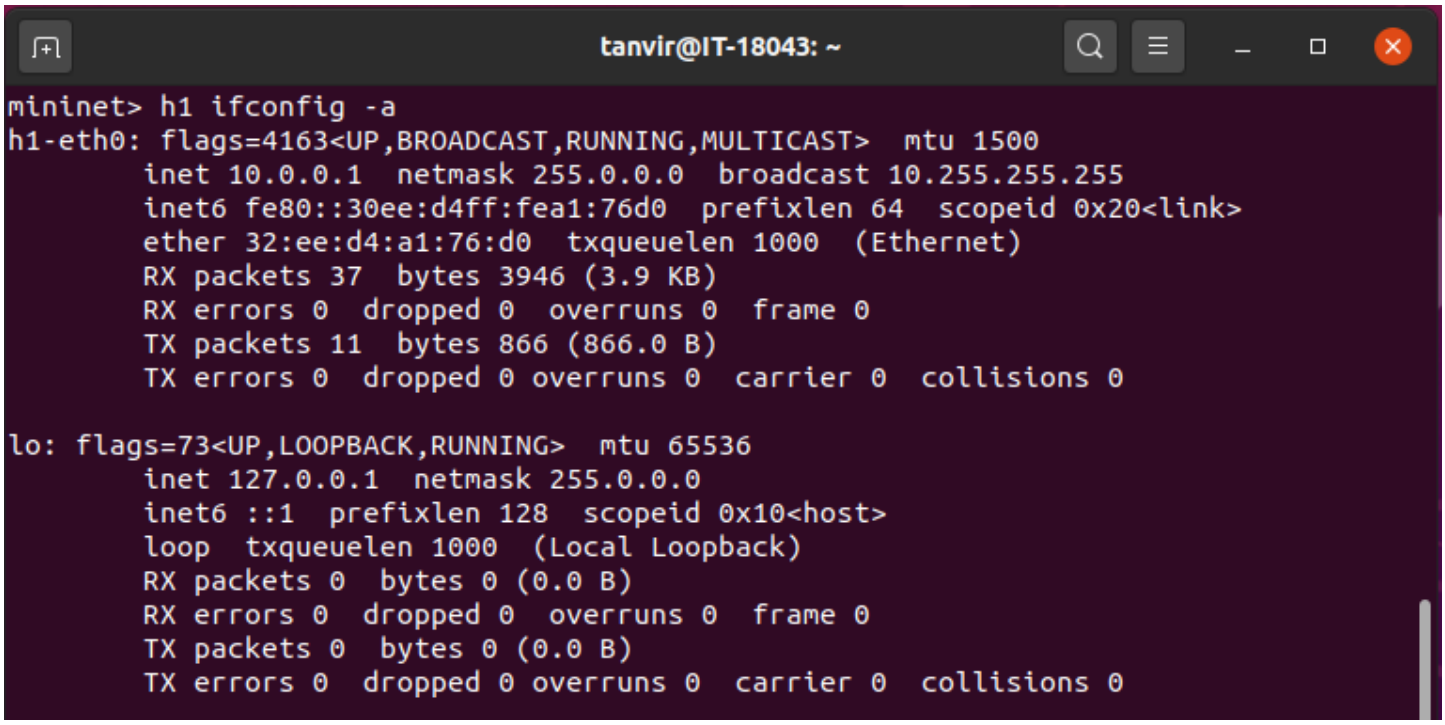
mininet> net



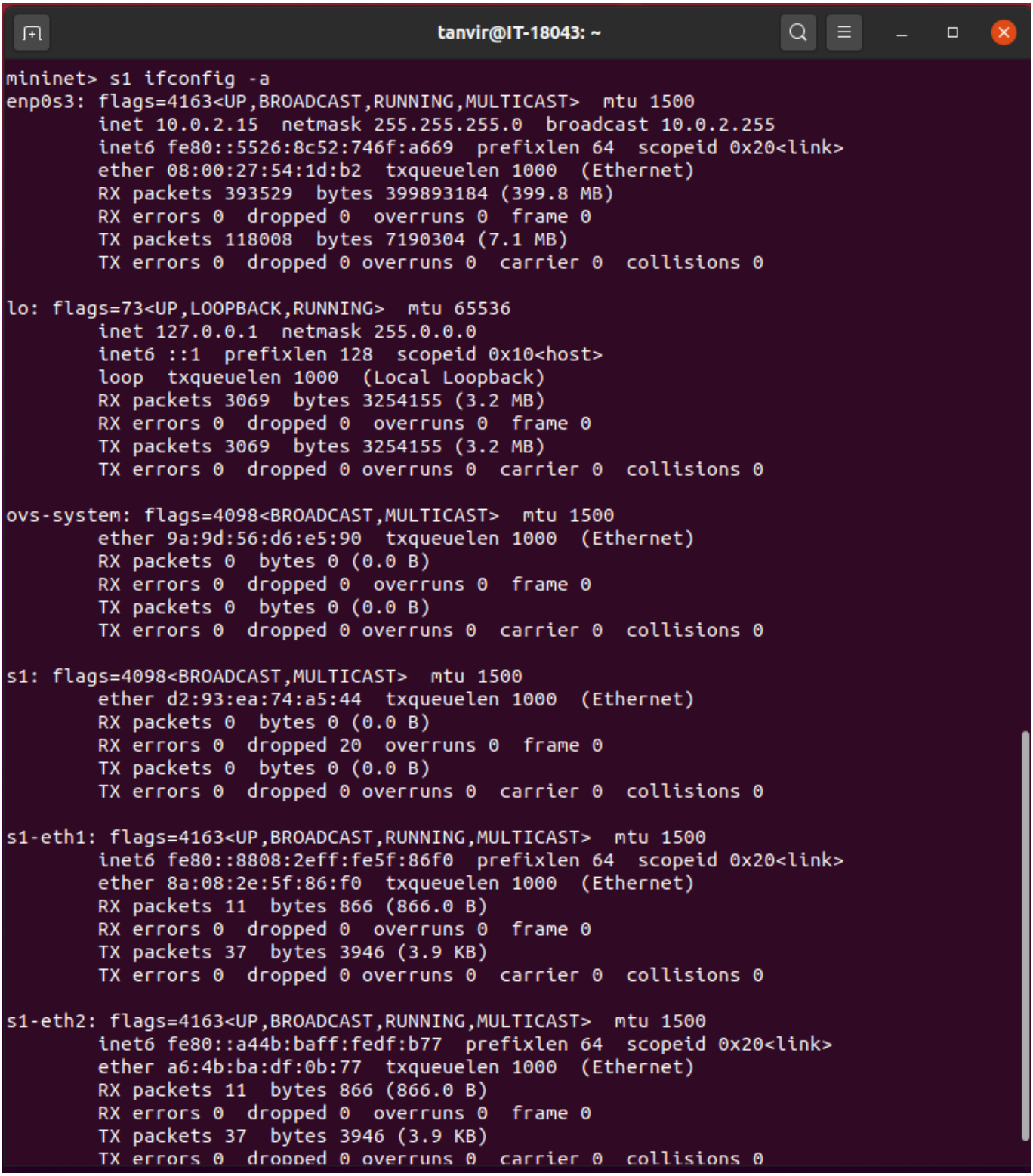
mininet> dump



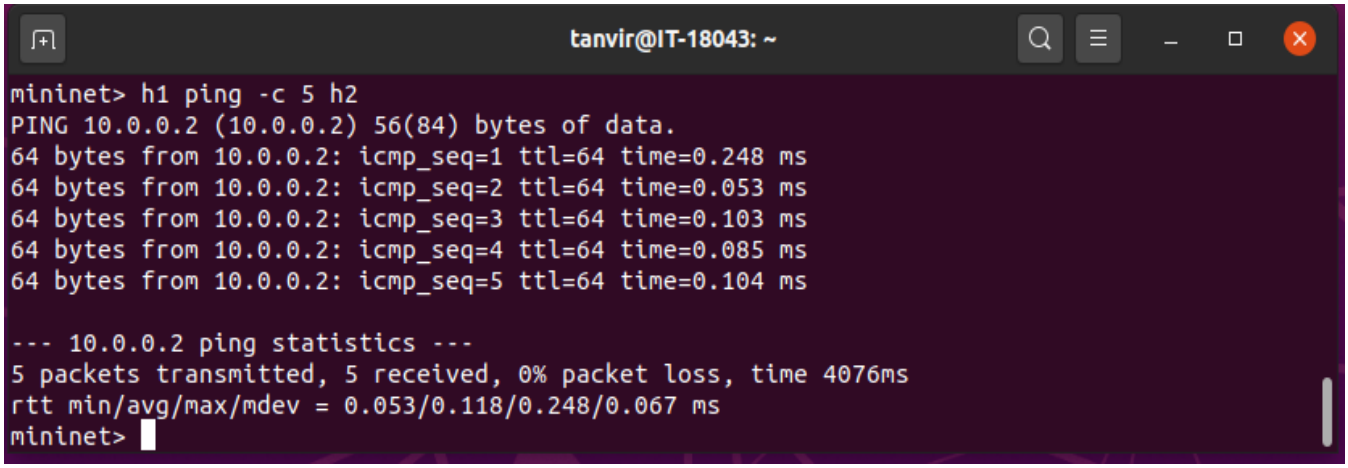
mininet> h1 ifconfig –a



s1 ifconfig -a

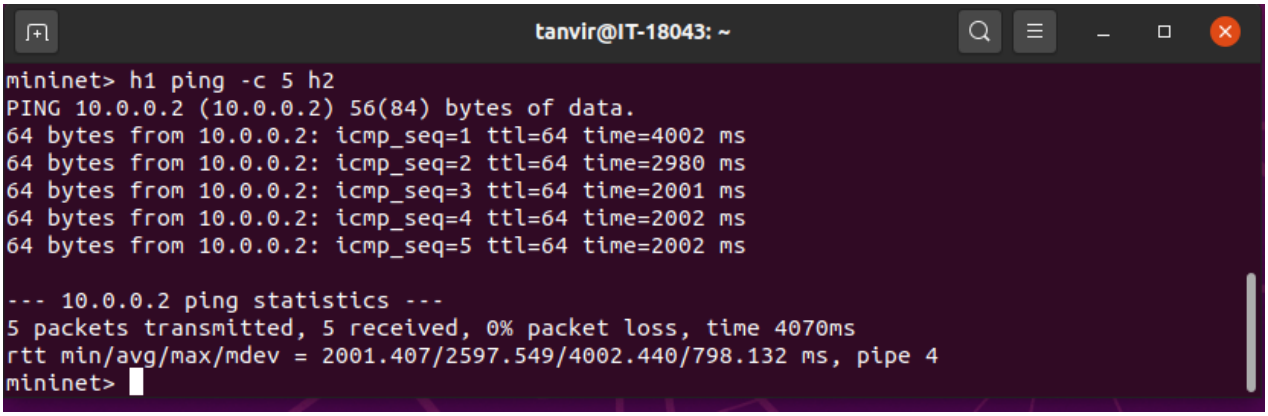

tanvir@IT-18043: ~
mininet> s1 ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fe80::5526:8c52:746f:a669 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:54:1d:b2 txqueuelen 1000 (Ethernet)
RX packets 393529 bytes 399893184 (399.8 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 118008 bytes 7190304 (7.1 MB)
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 3069 bytes 3254155 (3.2 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 3069 bytes 3254155 (3.2 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
ovs-system: flags=4098<BROADCAST,MULTICAST> mtu 1500
ether 9a:9d:56:d6:e5:90 txqueuelen 1000 (Ethernet)
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
s1: flags=4098<BROADCAST,MULTICAST> mtu 1500
ether d2:93:ea:74:a5:44 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 20 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
s1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet6 fe80::8808:2eff:fe5f:86f0 prefixlen 64 scopeid 0x20<link>
ether 8a:08:2e:5f:86:f0 txqueuelen 1000 (Ethernet)
RX packets 11 bytes 866 (866.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 37 bytes 3946 (3.9 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
s1-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet6 fe80::a44b:baff:fedf:b77 prefixlen 64 scopeid 0x20<link>
ether a6:4b:ba:df:0b:77 txqueuelen 1000 (Ethernet)
RX packets 11 bytes 866 (866.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 37 bytes 3946 (3.9 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet> h1 ping -c 5 h2



Exercise 4.2.3: In terminal-2, display the following command line: sudo mn --link tc,bw=10,delay=500ms

- **mininet> h1 ping -c 5 h2, What happen with the link?**
- **mininet> h1 iperf –s –u &**
- **mininet> h2 iperf –c IPv4_h1 -u, Is there any packet loss?**



Discussion :

Mininet is a network emulator which creates a network of virtual hosts, switches, controllers, and links. Mininet hosts run standard Linux network software, and its switches support OpenFlow for highly flexible custom routing and Software-Defined Networking.

Mininet supports research, development, learning, prototyping, testing, debugging, and any other tasks that could benefit from having a complete experimental network on a laptop or other PC.

Mininet:

- Provides a simple and inexpensive network testbed for developing OpenFlow applications
- Enables multiple concurrent developers to work independently on the same topology

- Supports system-level regression tests, which are repeatable and easily packaged
- Enables complex topology testing, without the need to wire up a physical network
- Includes a CLI that is topology-aware and OpenFlow-aware, for debugging or running network-wide tests
- Supports arbitrary custom topologies, and includes a basic set of parametrized topologies
- is usable out of the box without programming, but
- also Provides a straightforward and extensible Python API for network creation and experimentation

Mininet provides an easy way to get correct system *behavior* (and, to the extent supported by your hardware, performance) and to experiment with topologies.