Name: Nadira Islam
ID   : IT-17051

Lab report no: 07

Name of the report: SDN Controllers and Mininet

Theory:

Explain how the traffic generators work?
Ans:
Traffic Generators

A traffic generators is used to put traffic onto a network for other machines to consume. Logically, a traffic generator has a physical layer address (and usually higher-level address), because it is supposed to look like a machine on the network to the target machines receiving the traffic

Question 5.3: Which is the main difference between configuring UDP and TCP traffic?
Ans:
UDP traffic if fully customized in terms of packet lengh and batwidth. Instead TCP only allow customize the amount of traffic.

Question 5.4: In your opinions which are the main advantages and disadvantages of working with mininet? Provide at least 3.
Ans:
Advantage:

1. free
2. easy to install
3. allow test scalability

Disadvantage:

1. It is not real devices
2. No real traffic can be injected
3. No INTERNET output.

Question 5.5: Explain the architecture of Software defined Networks?

Ans:
Controller role: Brain of the network
Switch role: Hands of the network
Host role: Clients

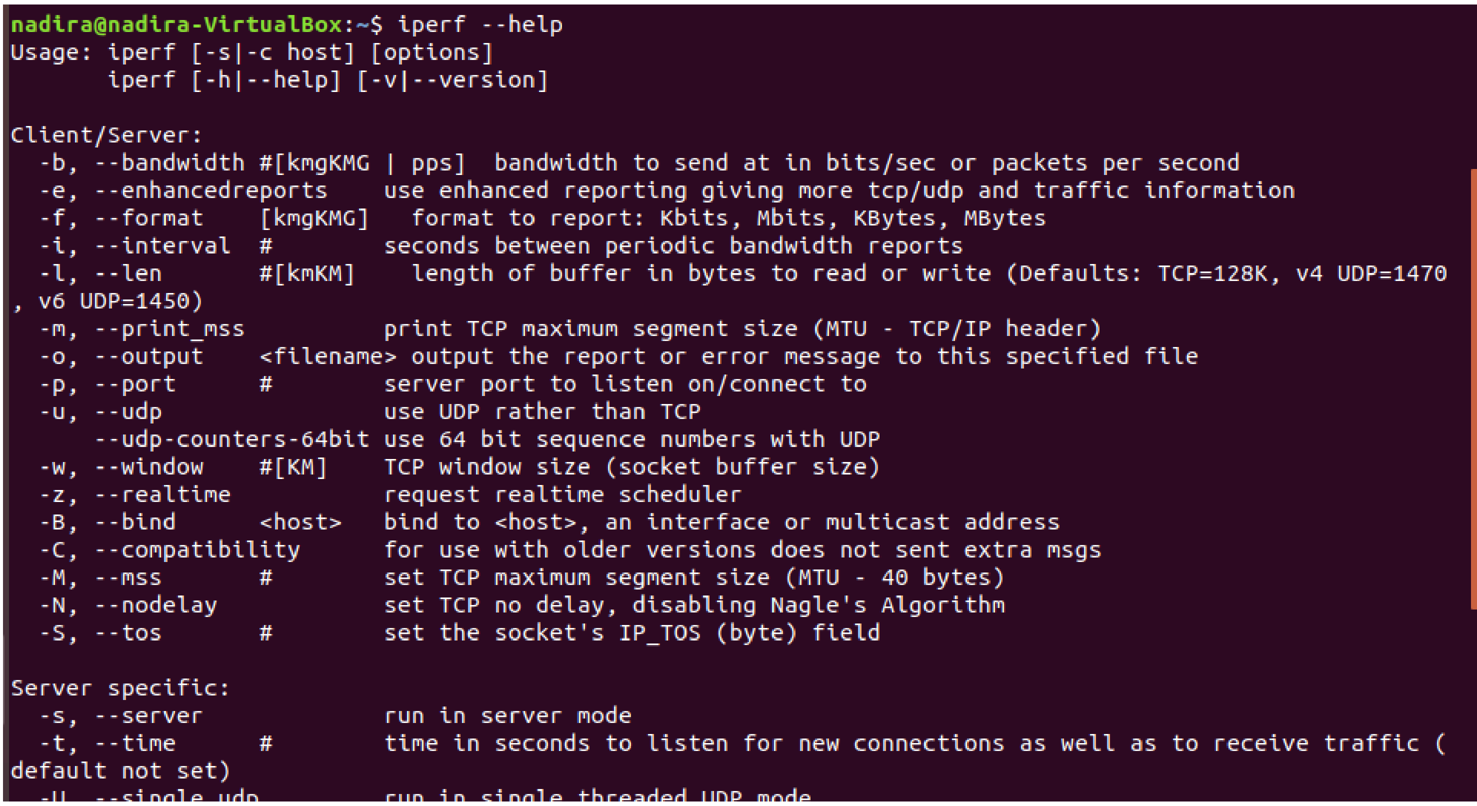
Question 5.6: What is the advantage of having a programmable Controller?
Ans:
Fully customize, program and controll your network.

Section 4.1: Using traffic Generators

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

Ans:

Client/Server:
-f, --format   [kmKM] format to report: Kbits, Mbits, KBytes, MBytes
-i, --interval   seconds between periodic bandwidth reports
-l, --len   [KM] length of buffer to read or write (default 8 KB)
-m, --print_mss print TCP maximum segment size (MTU - TCP/IP header)



```
nadira@nadira-VirtualBox:~$ iperf --help
Usage: iperf [-s|-c host] [options]
       iperf [-h|--help] [-v|--version]

Client/Server:
  -b, --bandwidth #[kmgKMG | pps]  bandwidth to send at in bits/sec or packets per second
  -e, --enhancedreports    use enhanced reporting giving more tcp/udp and traffic information
  -f, --format    [kmgKMG]   format to report: Kbits, Mbits, KBytes, MBytes
  -i, --interval  #        seconds between periodic bandwidth reports
  -l, --len       #[kmKM]    length of buffer in bytes to read or write (Defaults: TCP=128K, v4 UDP=1470
, v6 UDP=1450)
  -m, --print_mss          print TCP maximum segment size (MTU - TCP/IP header)
  -o, --output    <filename> output the report or error message to this specified file
  -p, --port      #        server port to listen on/connect to
  -u, --udp                use UDP rather than TCP
      --udp-counters-64bit use 64 bit sequence numbers with UDP
  -w, --window    #[KM]    TCP window size (socket buffer size)
  -z, --realtime           request realtime scheduler
  -B, --bind      <host>   bind to <host>, an interface or multicast address
  -C, --compatibility      for use with older versions does not sent extra msgs
  -M, --mss       #        set TCP maximum segment size (MTU - 40 bytes)
  -N, --nodelay            set TCP no delay, disabling Nagle's Algorithm
  -S, --tos       #        set the socket's IP_TOS (byte) field

Server specific:
  -s, --server             run in server mode
  -t, --time      #        time in seconds to listen for new connections as well as to receive traffic (
default not set)
  -U, --single_udp         run in single threaded UDP mode
```

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).Which are the statistics provided at the end of transmission?

Ans:

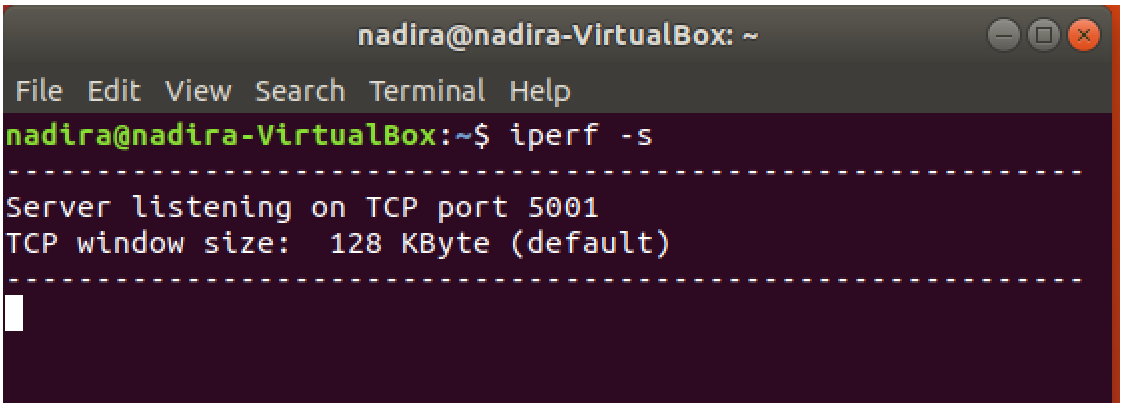
First we need to create a server by the following command:

$ iperf -s

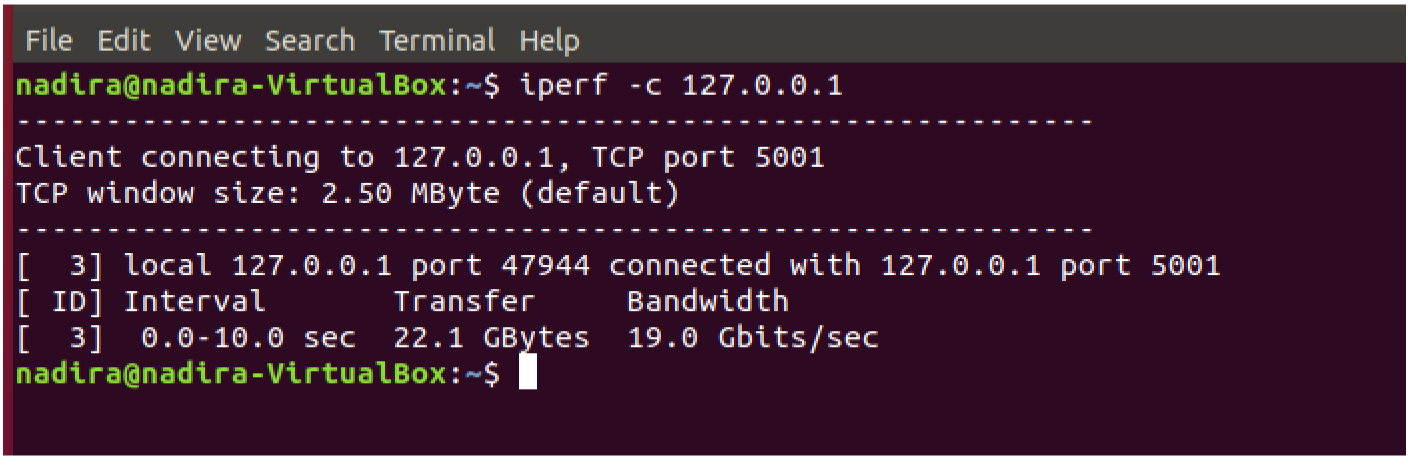
Then we need to create a client by the following command:

$ iperf -c 127.0.0.1

Here is the output:


nadira@nadira-VirtualBox: ~
File Edit View Search Terminal Help

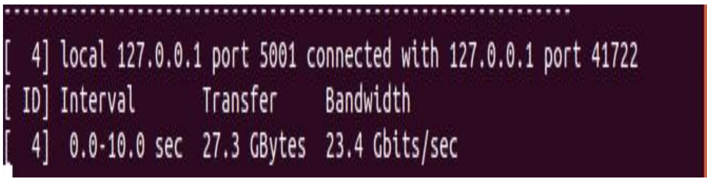
```
nadira@nadira-VirtualBox:~$ iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
```



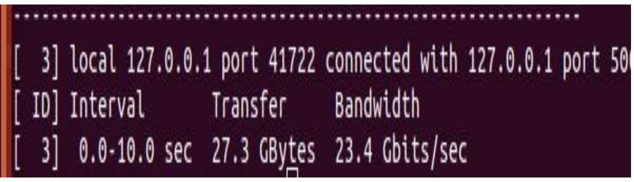
```
nadira@nadira-VirtualBox:~$ iperf -c 127.0.0.1
------------------------------------------------------------
Client connecting to 127.0.0.1, TCP port 5001
TCP window size: 2.50 MByte (default)
------------------------------------------------------------
[  3] local 127.0.0.1 port 47944 connected with 127.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  22.1 GBytes  19.0 Gbits/sec
nadira@nadira-VirtualBox:~$
```

The following statistics are provided at the end of the transmission:

The time interval, the amount of data transfer and the bandwidth.



```
------------------------------------------------------------
[  4] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 41722
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-10.0 sec  27.3 GBytes  23.4 Gbits/sec
```



```
------------------------------------------------------------
[  3] local 127.0.0.1 port 41722 connected with 127.0.0.1 port 50
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  27.3 GBytes  23.4 Gbits/sec
```

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?

Ans:
First we need to create a server by the following command:

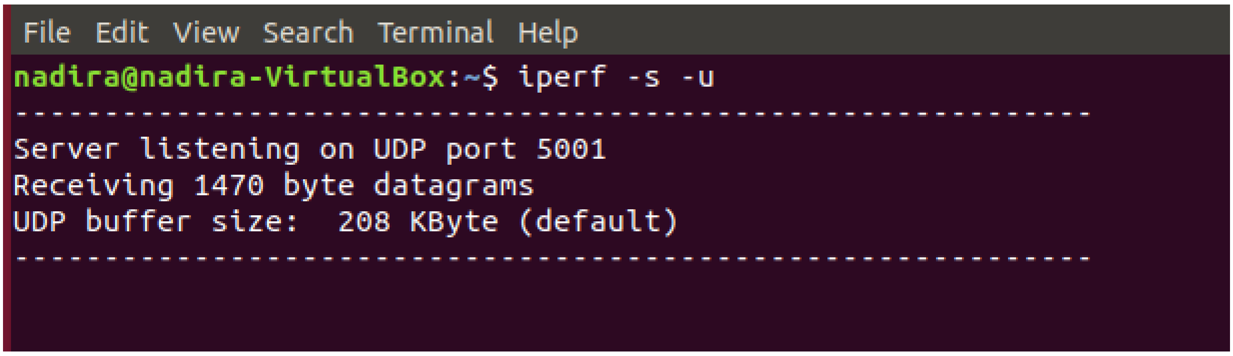
$ iperf -s -u

Here **u** for UDP trafic

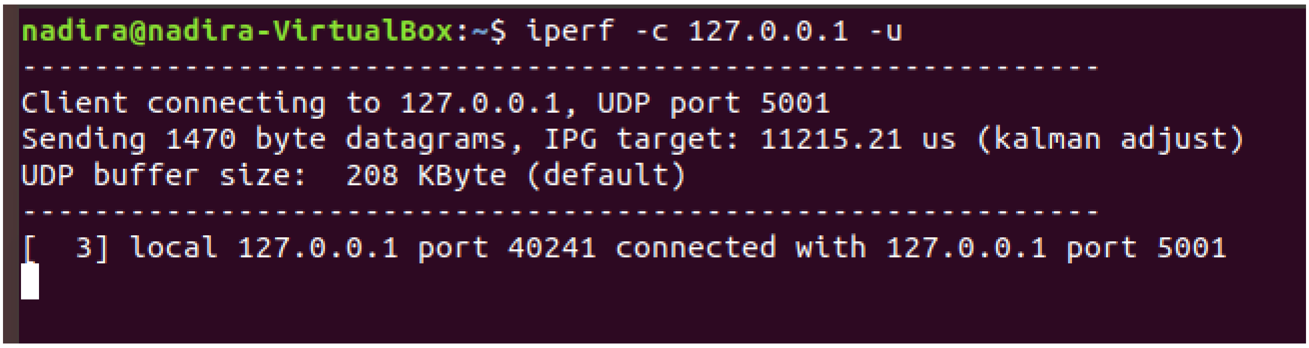
Then we need to create a client by the following command:

$ iperf -c 127.0.0.1 -u

Here is the output of server and client:



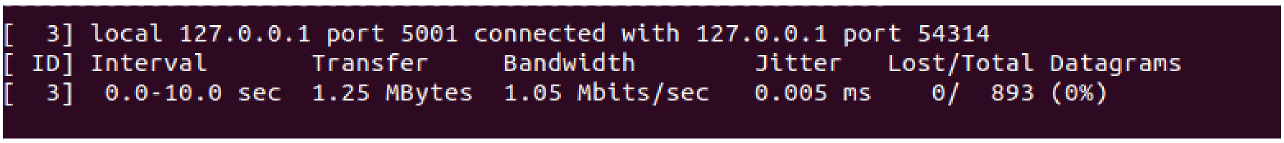
```
nadira@nadira-VirtualBox:~$ iperf -s -u
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
```



```
nadira@nadira-VirtualBox:~$ iperf -c 127.0.0.1 -u
------------------------------------------------------------
Client connecting to 127.0.0.1, UDP port 5001
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 127.0.0.1 port 40241 connected with 127.0.0.1 port 5001
```

Which are the statistics are provided at the end of transmission?

Ans: The time interval, the amount of data transfer, the bandwidth, the jitter and the datagram loss percentage.



```
[  3] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 54314
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec   0.005 ms    0/  893 (0%)
```

What is different from the statistics provided in exercise 4.1.1.?
Ans:
The jitter and the datagram loss percentage, because TCP guaranties 0% packet loss.

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

- ✔ Packet length = 1000bytes
- ✔ Time = 20 seconds
- ✔ Bandwidth = 1Mbps
- ✔ Port = 9900

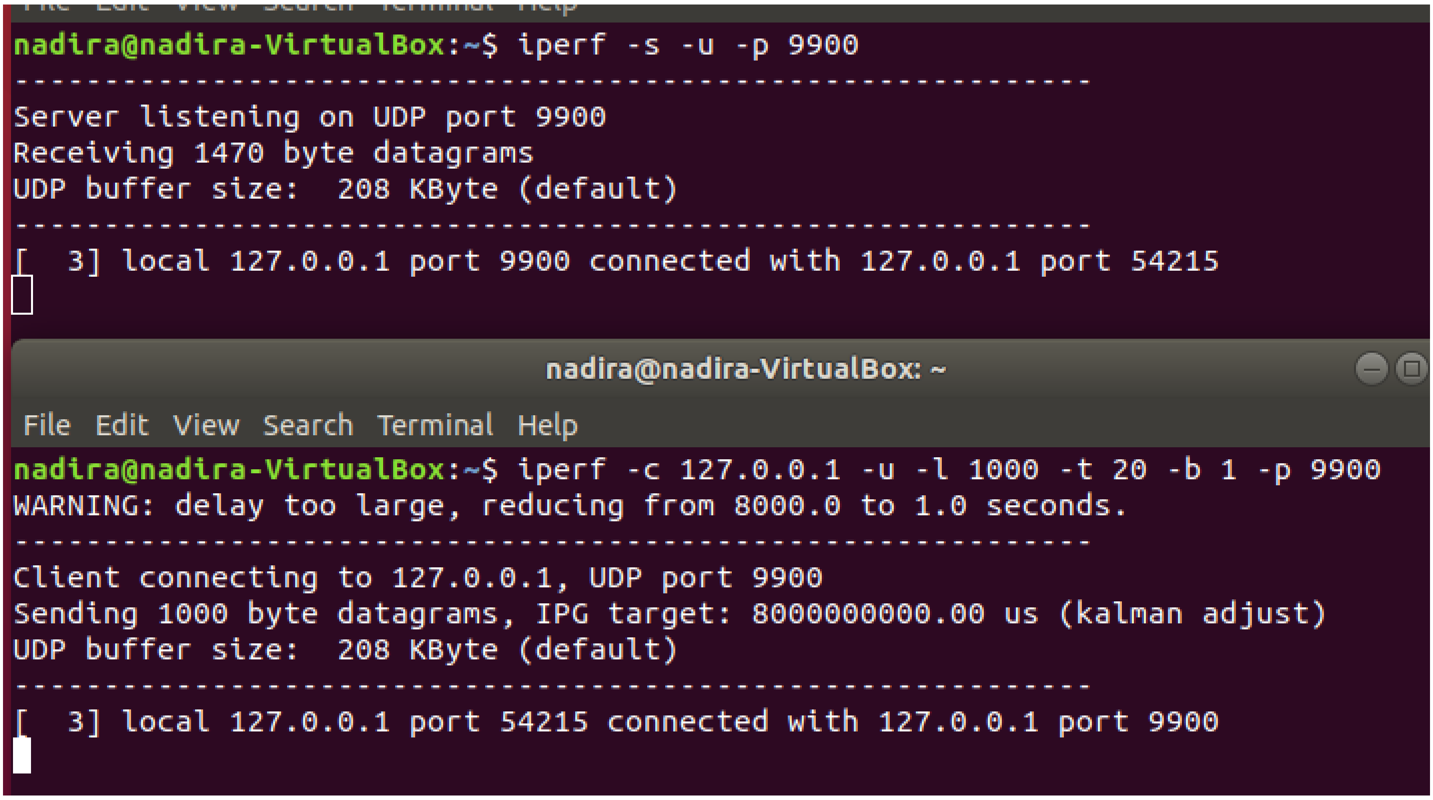
Which are the command lines?

Ans:

The client command is:
iperf -c 127.0.0.1 -u -l 1000 -t 20 -b 1 -p 9900

The server command is:
iperf -s -u -p 9900

Output:



```
nadira@nadira-VirtualBox:~$ iperf -s -u -p 9900
------------------------------------------------------------
Server listening on UDP port 9900
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 127.0.0.1 port 9900 connected with 127.0.0.1 port 54215
```

nadira@nadira-VirtualBox: ~
File Edit View Search Terminal Help

```
nadira@nadira-VirtualBox:~$ iperf -c 127.0.0.1 -u -l 1000 -t 20 -b 1 -p 9900
WARNING: delay too large, reducing from 8000.0 to 1.0 seconds.
------------------------------------------------------------
Client connecting to 127.0.0.1, UDP port 9900
Sending 1000 byte datagrams, IPG target: 8000000000.00 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 127.0.0.1 port 54215 connected with 127.0.0.1 port 9900
```