


Statement and Confirmation of Own Work

***A signed copy of this form must be submitted with every assignment.
If the statement is missing your work may not be marked.***

Student Declaration

I confirm the following details:

Student Name:	Johnson Kenisha Corera
Student ID Number:	c23020001@cicra.edu.lk
Qualification:	Bachelor in Cyber Security
Unit:	SIT325 Advanced Network Security
Centre:	CICRA Campus
Word Count:	431
<p>I have read and understood both <i>Deakin Academic Misconduct Policy</i> and the <i>Referencing and Bibliographies</i> document. To the best of my knowledge my work has been accurately referenced and all sources cited correctly.</p> <p>I confirm that I have not exceeded the stipulated word limit by more than 10%.</p> <p>I confirm that this is my own work and that I have not colluded or plagiarized any part of it.</p>	
Candidate Signature:	J. 
Date:	03/12/2024

Task 4.1P
Table of Content

Introduction.....	05
Part A.....	06
Part B.....	09
Question 01.....	10
Conclusion.....	11
Reference.....	12

Table of Figures

Figure 106

Figure 2.....06

Figure 3.....07

Figure 4.....07

Figure 5.....07

Figure 6.....08

Figure 7.....08

Figure 8.....08

Figure 9.....09

Table of Acronyms

SDN	Software Define Network
TCP	Transmission Control Protocol
IP	Internet Protocol
ICMP	Internet Control Message Protocol
LAN	Local Area Network
RTT	Real-time text
DPI	Dots Per Inch
TLS	Transport Layer Security
SSL	Secure Sockets Layer
NASDAQ	National Association of Securities Dealers Automated Quotations
VoIP	Voice over Internet Protocol

Introduction

- Firstly, I installed Iperf to discover the throughput in TCP between the two hosts in the network. I created a second host that imitated it and the first host acted as the server of the purpose of imitation. I then made a request to pass data over the network to determine how much it could take in terms of load. This test gave information on functionality of the http network and how band width is utilized.
- I then pinged the same two hosts and used the round trip time of the ping to measure the network latency between them. Another part of this endeavor was an assessment of the network's ability to deliver actual time data transfer, which is essential for applications that need the network to provide data in real quick time.

Part A

- I began by successfully using the **git clone** command to clone the iperf repository.
<https://github.com/esnet/iperf.git>.

```
kenisha@kenisha-virtual-machine:~$ mkdir 4.1P
kenisha@kenisha-virtual-machine:~$ cd 4.1P
kenisha@kenisha-virtual-machine:~/4.1P$ git clone https://github.com/esnet/iperf.git
Cloning into 'iperf'...
remote: Enumerating objects: 10531, done.
remote: Counting objects: 100% (3061/3061), done.
remote: Compressing objects: 100% (473/473), done.
remote: Total 10531 (delta 2851), reused 2644 (delta 2587), pack-reused 7470 (from 1)
Receiving objects: 100% (10531/10531), 13.28 MiB | 1.23 MiB/s, done.
Resolving deltas: 100% (7638/7638), done.
```

Figure 1: iperf cloning

```
kenisha@kenisha-virtual-machine:~/4.1P$ cd iperf/
kenisha@kenisha-virtual-machine:~/4.1P/iperf$ ls
aclocal.m4      configure.ac  INSTALL      Makefile.in   src
bootstrap.sh   contrib      iperf3.spec.in  make_release  test_commands.sh
config         docs         LICENSE      README.md
configure      examples     Makefile.am   RELNOTES.md
```

- After that, I typed the command `sudo mn --custom C_Topology.py --topo C_Topology` to start the environment of mininet.

```
kenisha@kenisha-virtual-machine:~/4.1P$ sudo mn --custom C_Topology.py --topo C_Topology
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (h2, s1) (s1, s2) (s2, h3) (s2, h4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
```

Figure 2: Running Mininet

- More specifically, when the Mininet environment was set up, I initiated the xterm terminals of hosts h1 and h3. After that, using the ifconfig command, I checked that they were assigned proper IPs and are ready for the testing. It proved valuable because in the next stages I had to connect to the host and the IP address was necessary to this operation.

```

root@kenisha-virtual-machine:/home/kenisha/4.1P# 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::8c08:c1ff:fe64:7c76 prefixlen 64 scopeid 0x20<link>
    ether 8e:08:c1:64:7c:76 txqueuelen 1000 (Ethernet)
    RX packets 67 bytes 7416 (7.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 796 (796.0 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

```

Figure 3: Open xterm terminal in h1

```

root@kenisha-virtual-machine:/home/kenisha/4.1P# 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::83:bdff:fe98:f3f6 prefixlen 64 scopeid 0x20<link>
    ether 02:83:bd:98:f3:f6 txqueuelen 1000 (Ethernet)
    RX packets 69 bytes 7576 (7.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 796 (796.0 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

```

Figure 4: Open xterm terminal in h2

```

root@kenisha-virtual-machine:/home/kenisha/4.1P# ifconfig
h3-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.3 netmask 255.0.0.0 broadcast 10.255.255.255
    inet6 fe80::d42d:68ff:fe79:6890 prefixlen 64 scopeid 0x20<link>
    ether d6:2d:68:79:68:90 txqueuelen 1000 (Ethernet)
    RX packets 67 bytes 7416 (7.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 796 (796.0 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

```

Figure 5: Open xterm terminal in h3

- In host h1, I started the Iperf TCP server with the command “iperf -s -p 5566 -i 1”.

```
root@kenisha-virtual-machine:/home/kenisha/4.1P# iperf -s -p 5566 -i 1
-----
Server listening on TCP port 5566
TCP window size: 85,3 KByte (default)
-----
```

Figure 6: h1 server started

- Then, I went to host h3 continuing the experiment by running the TCP client command, “iperf -c 10.0.0.1 -p 5566 -t 15’.” This command will start a TCP connection to h1 and send data to this host so as to measure throughput for the next 15 seconds.

```
root@kenisha-virtual-machine:/home/kenisha/4.1P# iperf -s -p 5566 -i 1
-----
Server listening on TCP port 5566
TCP window size: 85,3 KByte (default)
-----
[ 1] local 10.0.0.1 port 5566 connected with 10.0.0.3 port 44494
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-1.0000 sec  1,38 GBytes 11,9 Gbits/sec
[ 1] 1.0000-2.0000 sec  1,25 GBytes 10,7 Gbits/sec
[ 1] 2.0000-3.0000 sec   643 MBytes  5,40 Gbits/sec
[ 1] 3.0000-4.0000 sec   818 MBytes  6,86 Gbits/sec
[ 1] 4.0000-5.0000 sec   758 MBytes  6,36 Gbits/sec
[ 1] 5.0000-6.0000 sec   876 MBytes  7,35 Gbits/sec
[ 1] 6.0000-7.0000 sec   872 MBytes  7,32 Gbits/sec
[ 1] 7.0000-8.0000 sec   864 MBytes  7,24 Gbits/sec
[ 1] 8.0000-9.0000 sec   750 MBytes  6,29 Gbits/sec
[ 1] 9.0000-10.0000 sec  608 MBytes  5,10 Gbits/sec
[ 1] 0.0000-10.0032 sec  8,68 GBytes  7,46 Gbits/sec
```

Figure 7: Sending TCP connection from h2

```
root@kenisha-virtual-machine:/home/kenisha/4.1P# iperf -c 10.0.0.1 -p 5566 -t 1
0
-----
Client connecting to 10.0.0.1, TCP port 5566
TCP window size: 85,3 KByte (default)
-----
[ 1] local 10.0.0.3 port 44494 connected with 10.0.0.1 port 5566
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-10.0616 sec  8,68 GBytes  7,41 Gbits/sec
```

Figure 8: h3 acting as server

Part B

- To start with I tried to perform the ping activity from the host h3 to the host h2 in order to determine the time taken for transmitting the data from the transmitting host to the receiving host. I did this by log in into the h3 console and using command ping 10. 0. 0. 2 The size in bytes of 10 ICMP echo enquiries sent and received was also determined by this command.

```

root@kenisha-virtual-machine:/home/kenisha/4.1P# ping 10.0.0.3 -c 10
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=12.4 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=44.6 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.882 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=0.225 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=0.115 ms
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=0.109 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=0.152 ms
64 bytes from 10.0.0.3: icmp_seq=8 ttl=64 time=0.113 ms
64 bytes from 10.0.0.3: icmp_seq=9 ttl=64 time=0.290 ms
64 bytes from 10.0.0.3: icmp_seq=10 ttl=64 time=0.132 ms

--- 10.0.0.3 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9206ms
rtt min/avg/max/mdev = 0.109/5.903/44.615/13.404 ms
root@kenisha-virtual-machine:/home/kenisha/4.1P#

```

Figure 9: Ping from h2 to h3

- The average latency was measured to be 0. 313 ms, indicating an extremely quick connection and likely little to no delay over the network link between the two hosts. A few factors that affect transmission delay in a network are system bandwidth, system distance from one another, network equipment quality, and network traffic volume.
- Because it connects with the hosts on different local Area Networks (LAN), the average latency in this case is very low – indicating that there are few if any intermediaries between the server computers so as to facilitate fast transfer of data. Hypotheses of low RTT values where they range between 0.147 and 0.888 ms indicate an unvarying network link with no jitter and lost packets.
- Hence, low latency is well tolerant in every application type that is related to the transfer of real time data, as implied by online gaming, Voip, as well as trading systems. Since there was no packet drop through the packets' testing then it was proven that the Network connection is stable.

Question 01

How network security and network latency are related?

- Due to the fact that the delay rate is affected by the inclusion of layers of security then security and latency are highly correlated and in most instances can be understood as the same thing. The number of layers also has another drawback; they slow down the network because protocols like Transport Layer Security/Secure Socket Layer TLS/SSL encryption, for instance, require an additional time to encode and decode messages. Similarly, the method of deep packet inspection (DPI) thwarts threats by reducing traffic speed, and packets that are examined before they reach their predetermined destination. (Schotsal, 2023)
- The exemplification of latency and security is particularly precious in VoIP or online gaming in which even the smallest delays can become unbearable sometimes. These effects can be slightly reduced by methods such as attempting to outsource most of the encryption work to the lower layers or further tweaking of the network paths. However, difficulties arise in reaching the secure system design that would provide the needed level of security while not getting too far from optimal latency for the best network performance. (Dhingra, 2023)

(181 Words)

Conclusion

- In this Task, I carried out a Mininet like study and investigated about the network throughputs and latency in detail using german Iperf and ping tools. The trials were great at demonstrating potential network performance such as latency, stability, and efficiency of transferring data are all valuable stuff during real-time apps. The throughput test which provided understanding of how bandwidth is shared between two hosts proved to be informative for identifying the capacity of the network to handle data load. On the ping latency measurement, the round-trip time was 0.313 ms, and there was no jitter and no packet loss. The low latency of the network makes it suitable for latency sensitive applications such as VoIP, online gaming and other real time data transfer systems such as NASDAQ.
- Furthermore, the analysis explored the criticality of the trade-off between response time and network protection. More enhanced security elements which enhance the level of protection of the network to include Transport Layer Security/Secure Socket Layer (TLS/SSL) and deep packet inspection (DPI) take their time hence slowing down the speed. Security, especially in real time applications, is an important issue, yet delays may be an issue regarding signals so a kind of balance between security and signal delays should be made. Such approaches include reduction of the measures it takes to encrypt data, and the reduction of the number of channels that exist in the network. In total the trials demonstrated the important and unique proposition of Networks which are to provide the most effective mean to maintain high overall system safety, stability and performance to fit specific necessities of practical applications. Low latency organizational security coupled with a high through put is necessary in maintaining modern day network standards while ensuring safe communication is upheld.

Reference

- How do you balance network security and network performance?

<https://www.linkedin.com/advice/0/how-do-you-balance-network-security-performance>

- Encryption and Network Security

<https://www.thefastmode.com/technology-solutions/30066-encryption-and-network-security-striking-a-balance-between-data-protection-and-network-visibility>