Assignment 3: Susmita Mondal (20310071)

Mininet-based

* Tool Setup:

1. Virtual Box
2. Mininet-VM
3. Iperf3
4. Xming server :0.0
5. Putty x-11 enable in “:0.0”
6. Putty Session IP of required virtual machine

* Migrating to Mininet & Analysis

1. A. Need to define a program where server and client fetch the previous program
   * + The program is in the same directory as the previous server and client program
     + the topology is to be defined in the program for mininet
2. B. (a) Single Topology
   * + swich 4 host. 1 host act as server, rest as client.
     + Sudo mn –topo single,4
     1. Server-client model on port 12345
     2. The program is written as single.py
     3. Sudo python3 single.py to run

Text

Description automatically generated

Text, table

Description automatically generated

Concurrent mode:

A picture containing text

Description automatically generated

All data can be requested in a concurrent fashion.

Transfer speed is much higher than earlier (non-mininet)

(b) Linear Topology Bandwidth

-2 node 1 switch

- link bandwidth 10Mb, 100 Mb and 1Gbps respectively

- sudo mn –test pingall –topo liner,1

* + - Sudo mn –link tc,bw=10
    - Sudo mn –link tc,bw=100
    - Sudo mn –link tc,bw=1Gb
    - Server-client model on port 12345
    - The program is written as bw.py
    - Sudo python3 bw.py to run

Text

Description automatically generated

* + - * + 10Mb result 
        + 100Mb result 
        + 1 Gb result 

Basically, the 10Mb link has the best throughput among these three links.

( c ) Linear Topology Delay

-2 node 1 switch

- link bandwidth 10 Mb, 100 Mb and 1Gbps respectively

- sudo mn –test pingall –topo liner,1

* + - Sudo mn –link tc,bw=10, delay=1ms / 2ms / 5ms
    - Sudo mn –link tc,bw=100, delay=1ms / 2ms / 5ms
    - Sudo mn –link tc,bw=1Gb, delay=1ms / 2ms / 5ms
    - Server-client model on port 12345
    - The program is written as delay.py
    - Sudo python3 delay.py to run

|  |  |  |  |
| --- | --- | --- | --- |
| Link | 10Mb | 100Mb | 1Gb |
| Delay = 1ms |  |  |  |
| Delay = 2ms |  |  |  |
| Delay = 5ms |  |  |  |

* + - 10Mb link has better throughput in 5ms delay,
    - 100Mb link has better throughput in 2ms delay
    - 1Gb link has better throughput in 1ms delay

Throughput vary depending upon Link speed as well as delay

( d ) Linear Topology Loss

-2 node 1 switch

link bandwidth 10 Mb, 100 Mb and 1Gbps respectively

sudo mn –test pingall –topo liner,1

* + - Sudo mn –link tc,bw=10, loss =1% / 2% / 5%
    - Sudo mn –link tc,bw=100, loss =1% / 2% / 5%
    - Sudo mn –link tc,bw=1Gb,, loss =1% / 2% / 5%
    - Server-client model on port 12345
    - The program is written as loss.py
    - Sudo python3 loss.py to run

|  |  |  |  |
| --- | --- | --- | --- |
| Link | 10Mb | 100Mb | 1Gb |
| Loss 1% |  |  |  |
| Loss 2% |  |  |  |
| Loss 5% |  |  |  |

2% loss has given a better performance overall

( e ) Linear Topology Hops

-2 host (n) nodes 2 switches (k)

* + - Switches goes to n or n-1 depending upon even and odd number
    - 2 hops are taken starting from 2 up to 10 switches (eg. 2,4,6,8,10)
    - Server-client model on port 12345
    - The program is written as hops.py
    - Sudo python3 hops.py to run
      * + 2 hops  - 0.006sec
        + 4 hops  - 0.017 sec
        + 6 hops  - 0.046 sec
        + 8 hops - 0.091 sec
        + 10 hops - 0.163sec
    - The more hops, the more time taken and less bandwidth.
* Custom Topology

{i} Varying Link Bandwidth

-Aggregated 6.9Mbps-12.0Mbps

Though the highest capacity of the link is 40Mbps the throughput is very less than that. Consider bottle neck link which is 10Mbps

* + - The program is written as Bandwidth.py
    - Sudo python3 Bandwidth.py to run
    - Tree like topology with depth 3

{ii} Horizontal scaling and Load Balancing

* + - The program is written as Scale.py & Load.py
    - Sudo python3 Scale.py & Load.py to run
    - Tree like topology with depth 3
    - Load Balancing

Graphical user interface

Description automatically generated

* + - The codes are submitted separately.