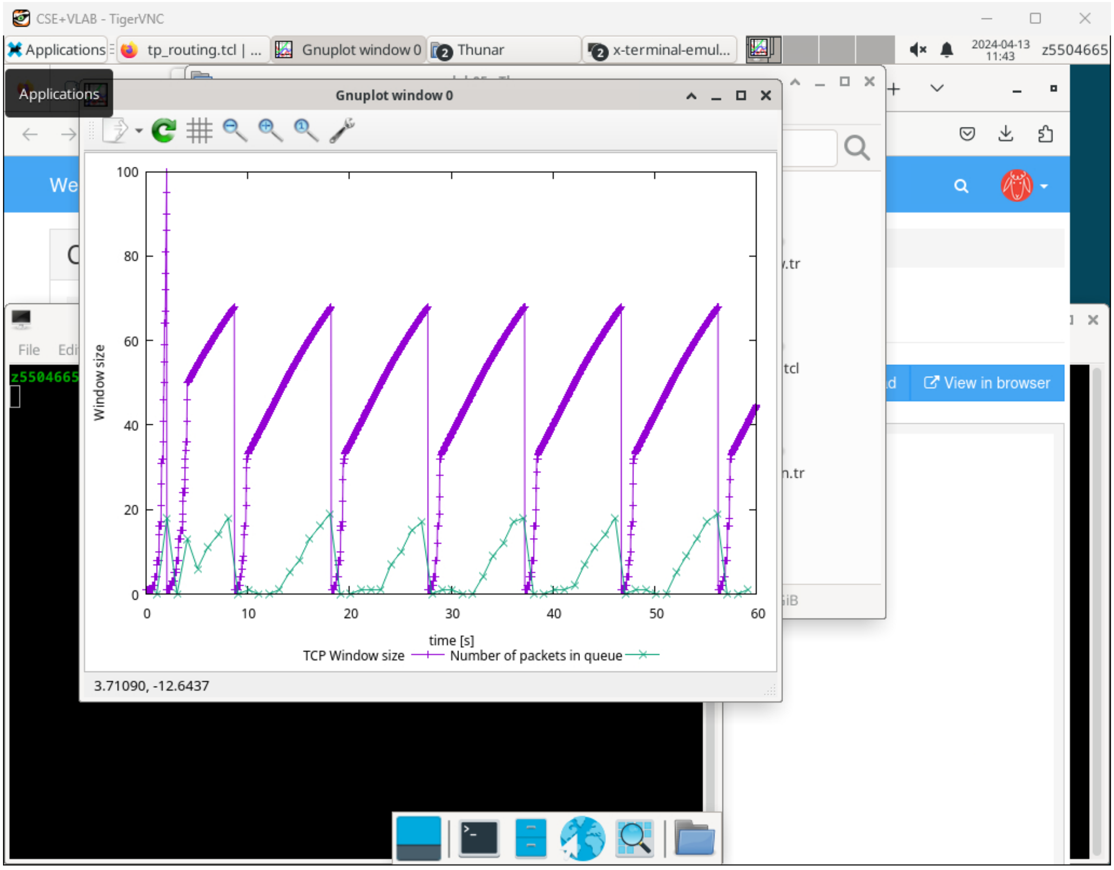
Lab5

Exercise 1

Q1

a)



The maximum size of the congestion window is 100

b)

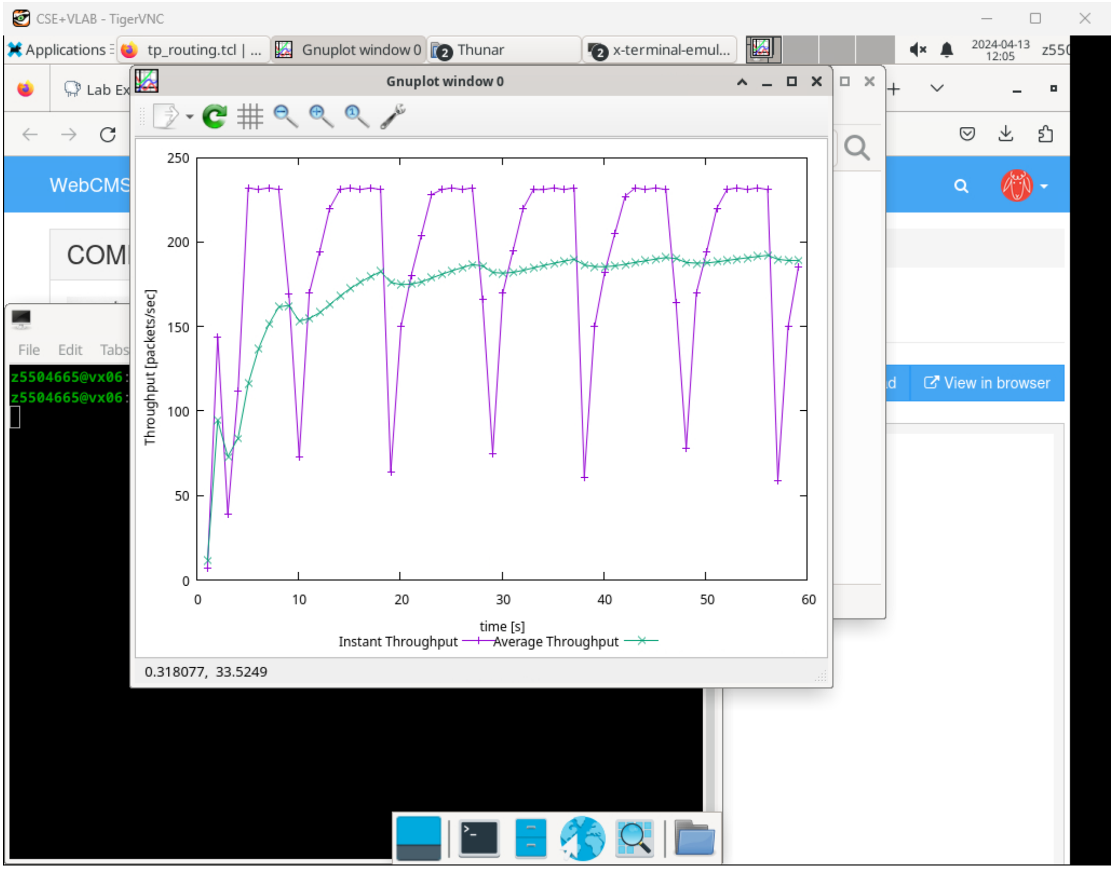
1) At this point, network congestion occurred, to avoid it, the window size was adjusted to 1, and the program re-entered the slow-start state.

2) Adjusting the window size to 1 and re-entering the slow-start state is done in response to network congestion. When congestion occurs, reducing the window size helps decrease the sending rate, thereby reducing network load and the likelihood of further congestion. Re-entering the slow-start state allows the sender to gradually increase its sending window at a slower rate, testing the available bandwidth cautiously to avoid further congestion. This reactive mechanism helps maintain network stability and ensure reliable data transmission.

c)

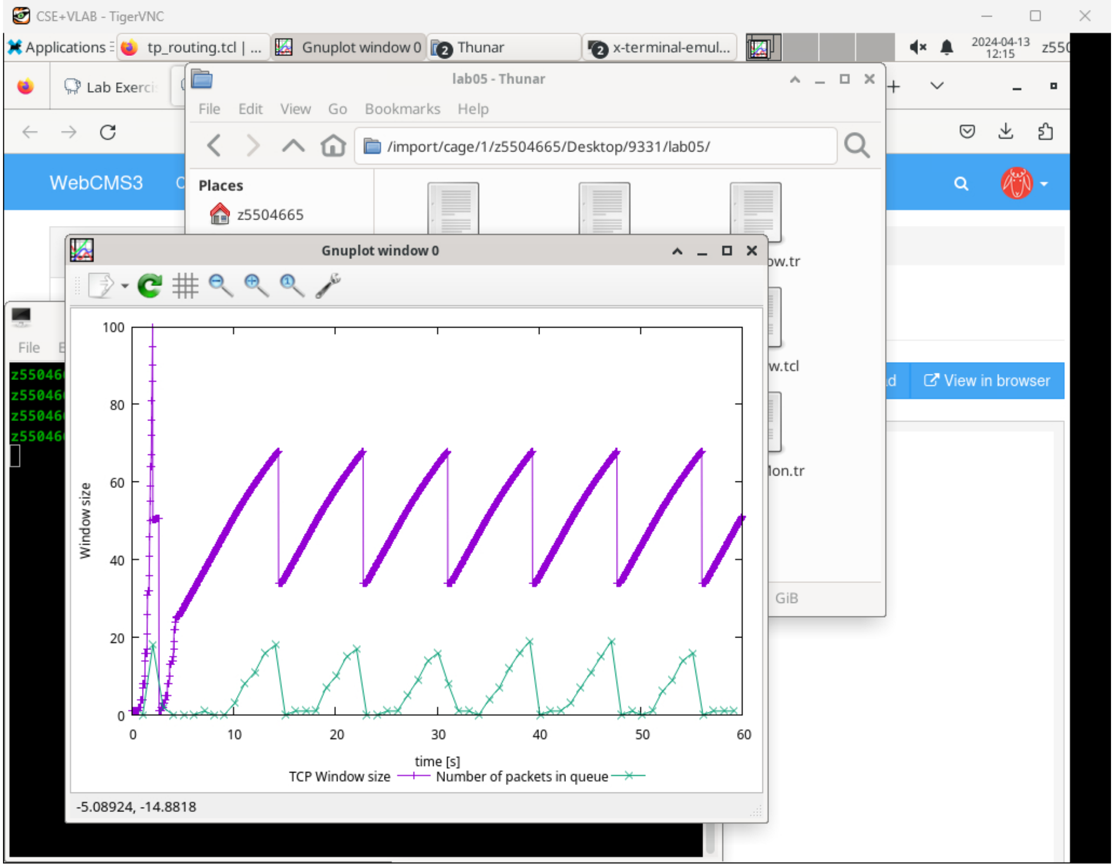
The program re-entered the slow-start state

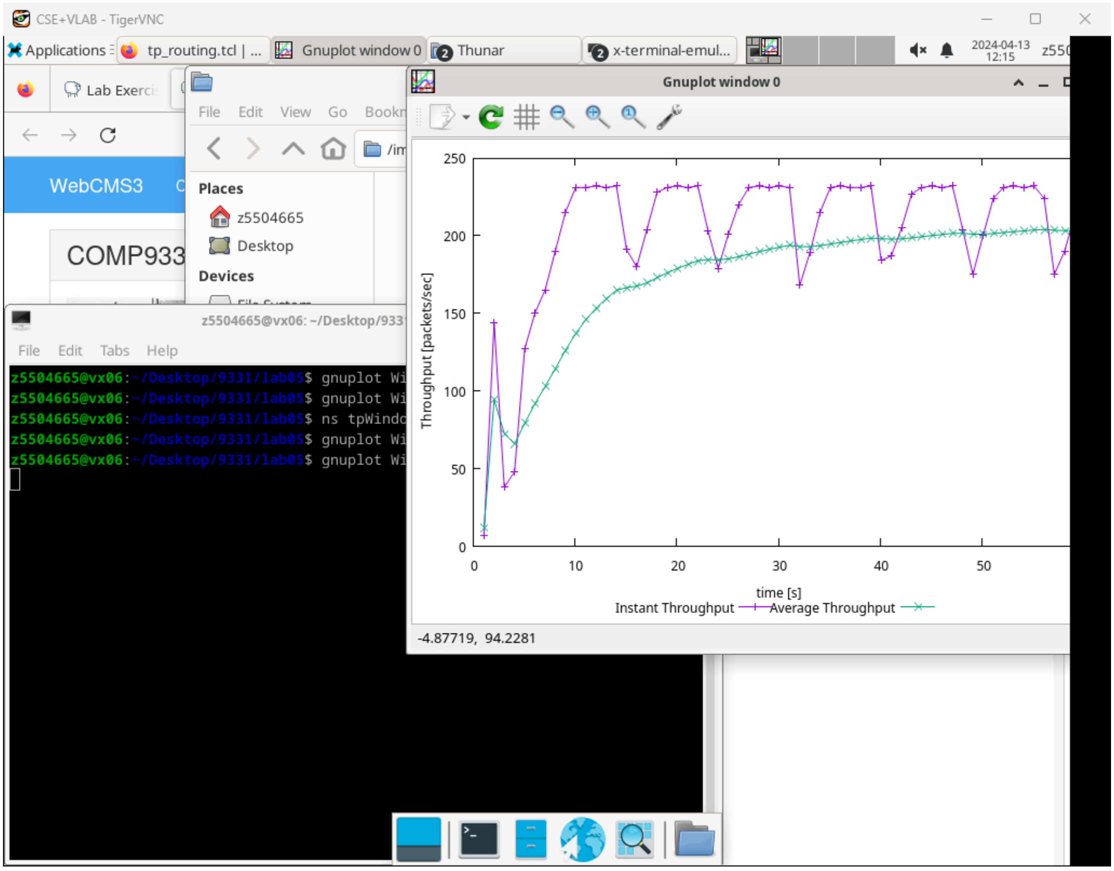
Q2



By the picture, the average throughput of TCP in this case like 190 packets per second which is 190 x 500 x 8 = 760 kbps

Q3


CSE+VLAB - TigerVNC
Applications
tp_routing.tcl | ...
Gnuplot window 0
Thunar
x-terminal-emul...
2024-04-13
12:15
lab05 - Thunar
File Edit View Go Bookmarks Help
/import/cage/1/z5504665/Desktop/9331/lab05/
Places
z5504665
WebCMS3
View in browser
Gnuplot window 0
Window size
time [s]
TCP Window size
Number of packets in queue
-5.08924, -14.8818

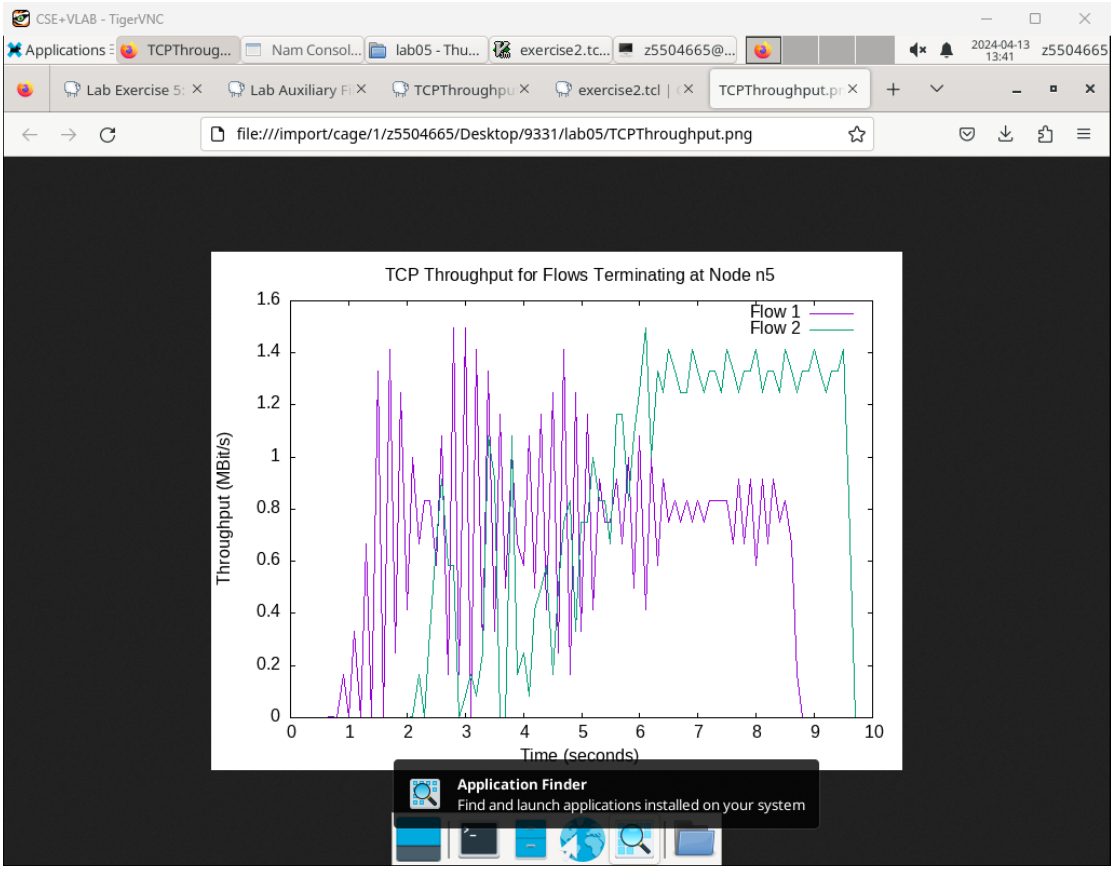

CSE+VLAB - TigerVNC
Applications
tp_routing.tcl | ...
Gnuplot window 0
Thunar
x-terminal-emul...
2024-04-13
12:15
WebCMS3
COMP933
Places
z5504665
Desktop
Devices
z5504665@vx06: ~/Desktop/9331
File Edit Tabs Help
z5504665@vx06:~/Desktop/9331/lab05$ gnuplot Wi
z5504665@vx06:~/Desktop/9331/lab05$ gnuplot Wi
z5504665@vx06:~/Desktop/9331/lab05$ ns tpWindo
z5504665@vx06:~/Desktop/9331/lab05$ gnuplot Wi
z5504665@vx06:~/Desktop/9331/lab05$ gnuplot Wi
Gnuplot window 0
Throughput [packets/sec]
time [s]
Instant Throughput
Average Throughput
-4.87719, 94.2281

Reno: Implements Fast Retransmit and Fast Recovery mechanisms. When multiple duplicate acknowledgments (dupacks) are received, Reno will fast retransmit the missing segment and enter fast recovery, reducing the congestion window by half and slowly increasing it after a new acknowledgment is received.

Tahoe: Does not implement Fast Retransmit and Fast Recovery. Instead, it relies solely on the slow-start and congestion avoidance mechanisms.

In summary, Reno leads to potentially higher average throughput compared to Tahoe.

Exercise 2



Q1

Because tcp1 experienced congestion first and consequently reduced its window size, this allowed tcp2 to achieve a larger window. As a result, tcp2 attained a higher throughput than tcp1 during this time, forming a relative steady state.

Q2

Because initially the correct congestion control window couldn't be accurately determined, the connection experienced multiple timeouts or triple duplicate ACK events. This led to the sending window suddenly reducing to either 1 or half of its size, causing an instantaneous change in throughput and resulting in fluctuations between 0.5s to 2s.

Exercise 3

Q1

n0 with n5
route: n0-n1-n4-n5

n2 with n5
route: n2-n3-n5

No, the route does not change overtime.

Q2

The link n1-n4 is down.
No, it does not.

Q3

Yes, there is an additional traffic n1-n2
The n1 reroute the traffic to n2 when the link n1-n4 was down

Q4

The route from n1 to n4 was affected when the cost increased. And n0 choses another route n0-n1-n2-n3-n5 to communicate with n5.

Q5

Node set multiPath_ 1 allows n2 to use multiple paths to communicate with n5. By setting the costs using $ns cost $n1 $n4 2 and $ns cost $n3 $n5 3, you're modifying the costs between n1→n4 and n3→n5. This means n2 will utilize both paths.