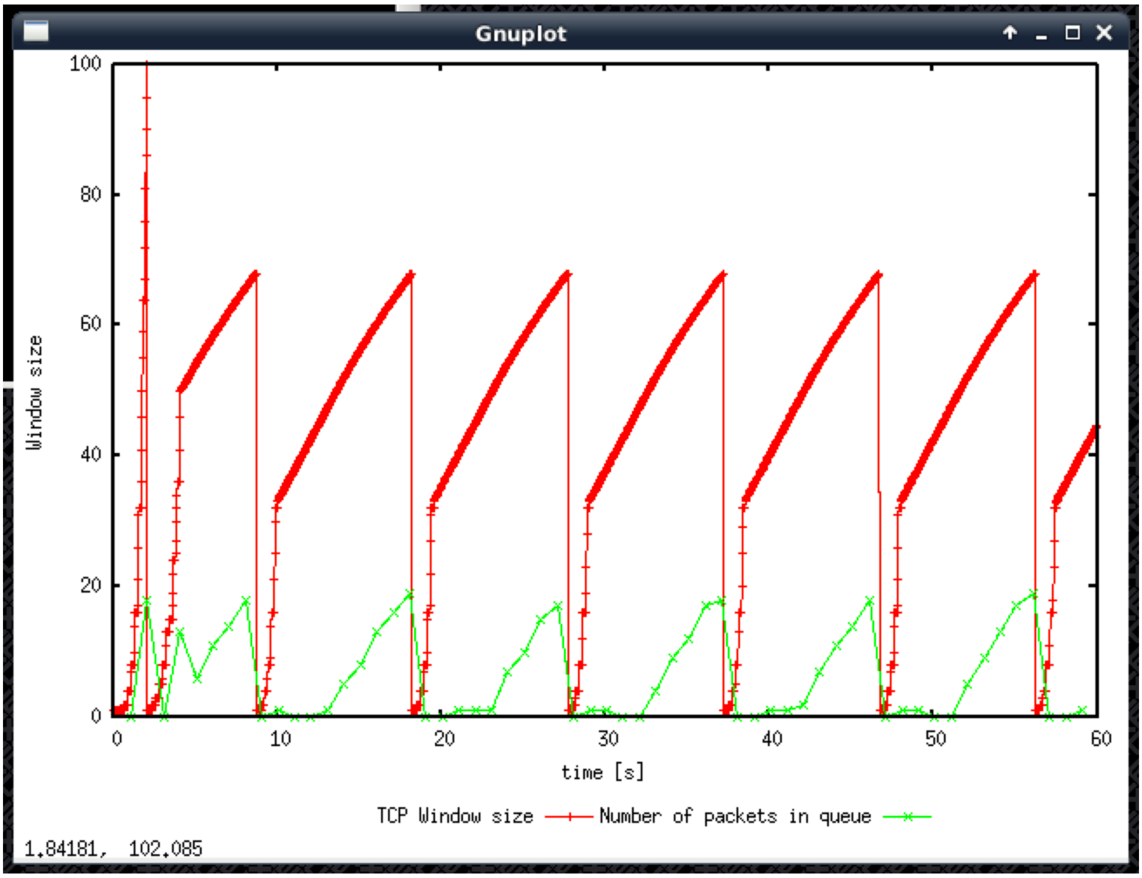
COMP3331 Computer Networks and Applications
Lab Exercise 05: TCP Congestion Control and Fairness

HyoJoo Kwon
z5222646
W12A

Exercise 1: Understanding TCP Congestion Control using ns-2

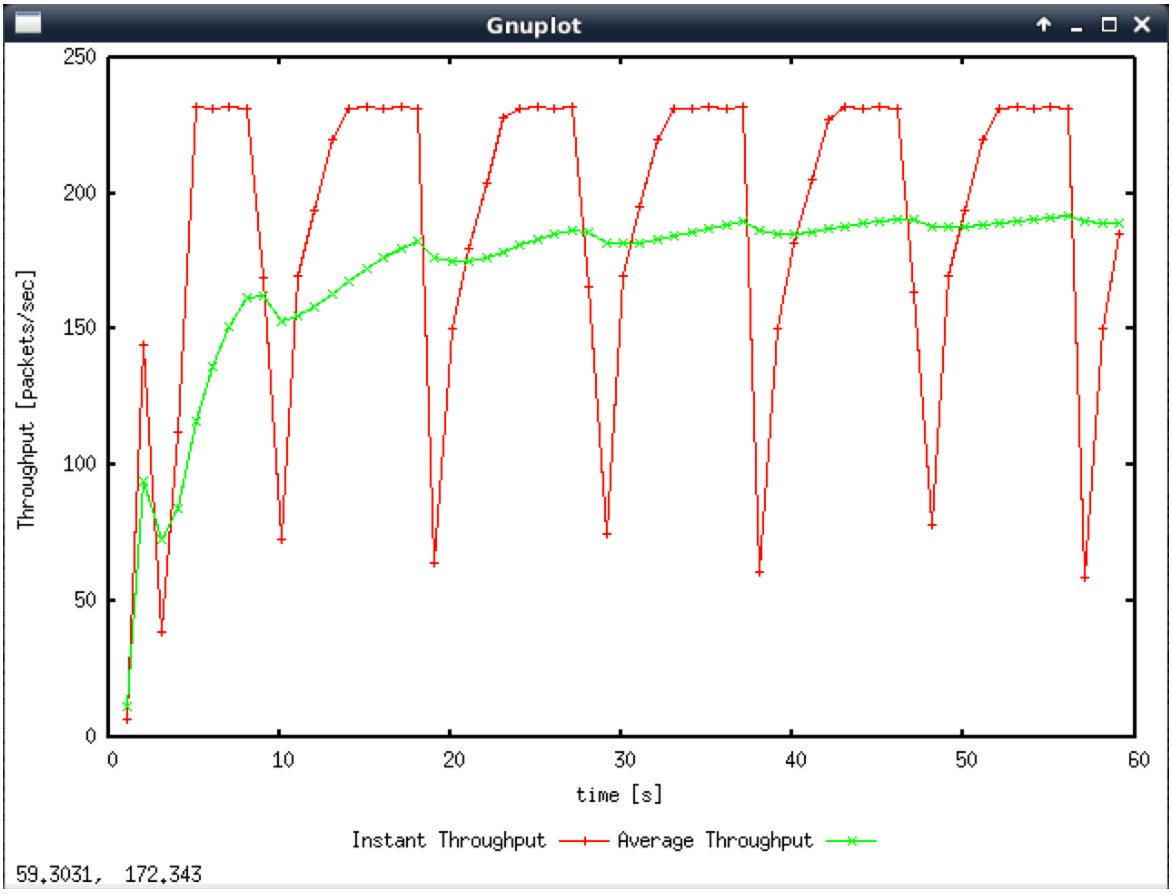
Question1: What is the maximum size of the congestion window that the TCP flow reaches in this case? What does the TCP flow do when the congestion window reaches this value? Why? What happens next?

- The maximum size of the congestion window: 100 (according to slow-start mechanism).
- The congestion window is then dropped to 1 after it detects a packet loss.
- It then sets the ssthresh to 50 and begin another slow-start.



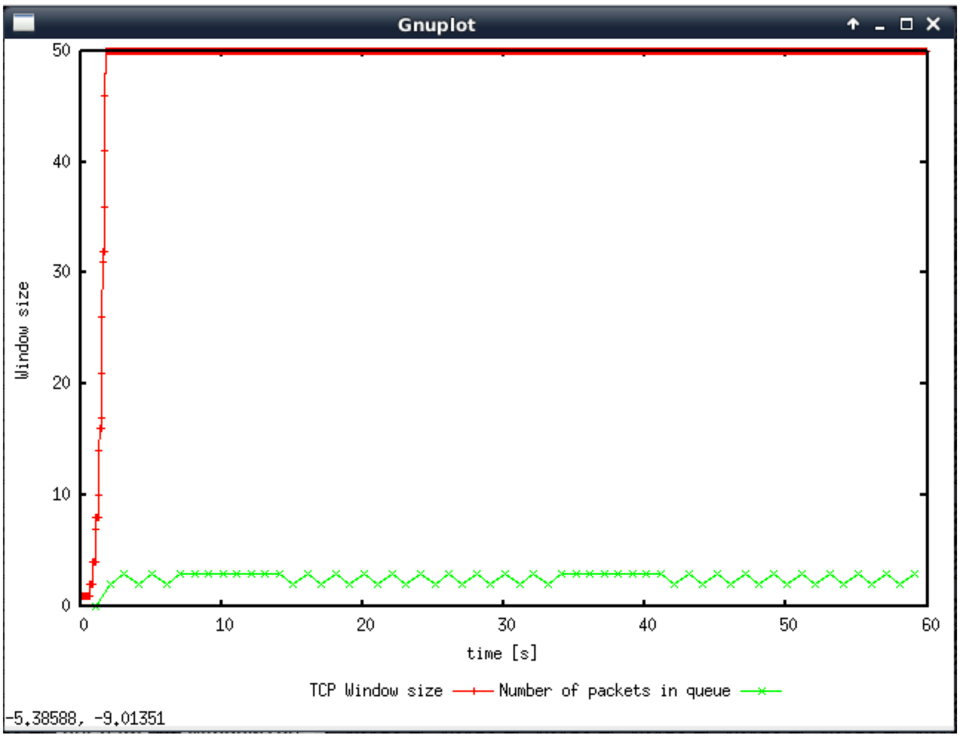
Question 2: From the simulation script we used, we know that the payload of the packet is 500 Bytes. Keep in mind that the size of the IP and TCP headers is 20 Bytes, each. Neglect any other headers. What is the average throughput of TCP in this case? (both in number of packets per second and bps)

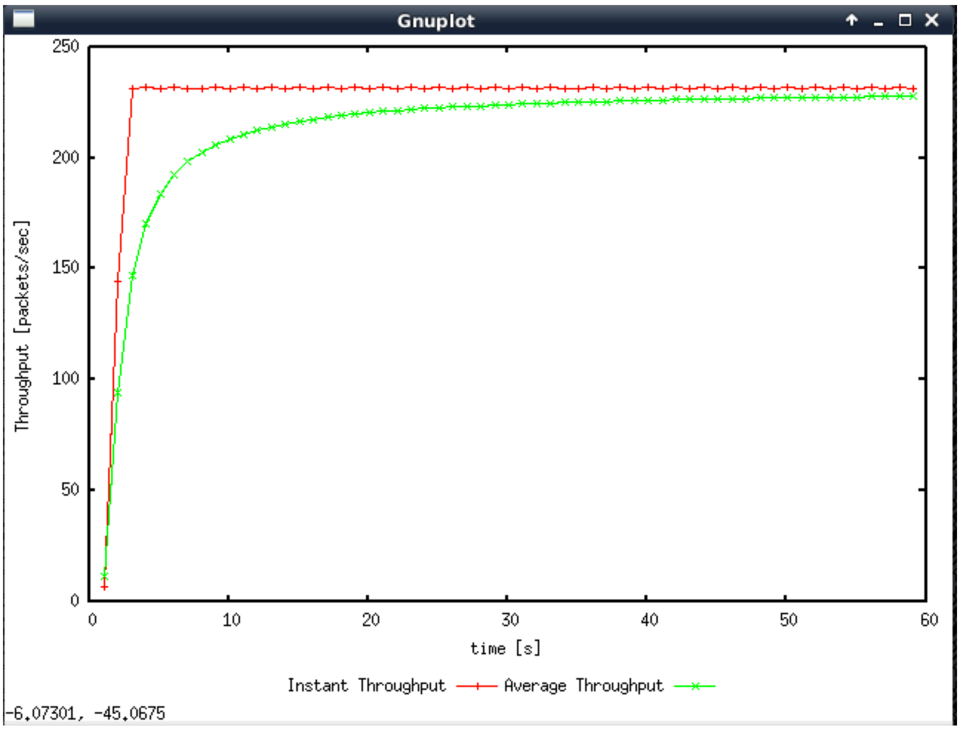
- Average throughput (in number of packets per sec): ~190 packets/sec
- Average throughput (bps : (500-20-20)*190 = 87,400 byte/sec



Question 3: Rerun the above script, each time with different values for the max congestion window size but the same RTT (i.e. 100ms). How does TCP respond to the variation of this parameter? Find the value of the maximum congestion window at which TCP stops oscillating (i.e., does not move up and down again) to reach a stable behaviour. What is the average throughput (in packets and bps) at this point? How does the actual average throughput compare to the link capacity (1Mbps)?

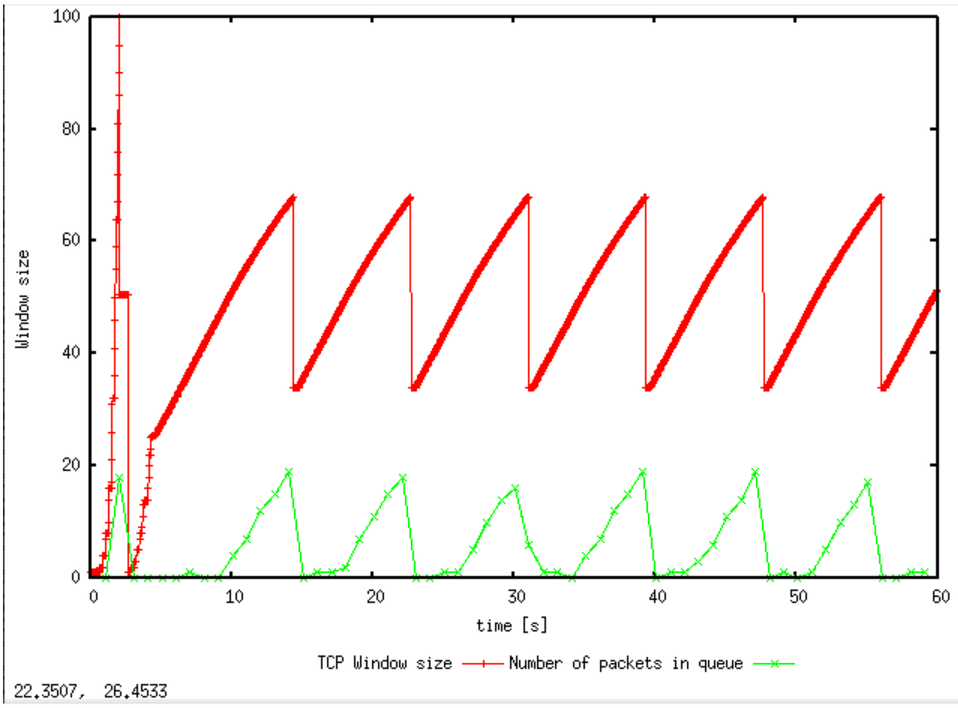
- There is only one oscillating from slow-start. If the size is less than 50, there is no oscillation. If the size is greater than 60, there will be packet loss and more than one oscillation occurs.
- Maximum congestion window at which TCP stops oscillating: 50
- Average throughput: ~230packets/sec = 105800 byte/sec
- Link capacity: 1Mbps = 125000 bytes = uses ~85% of link capacity

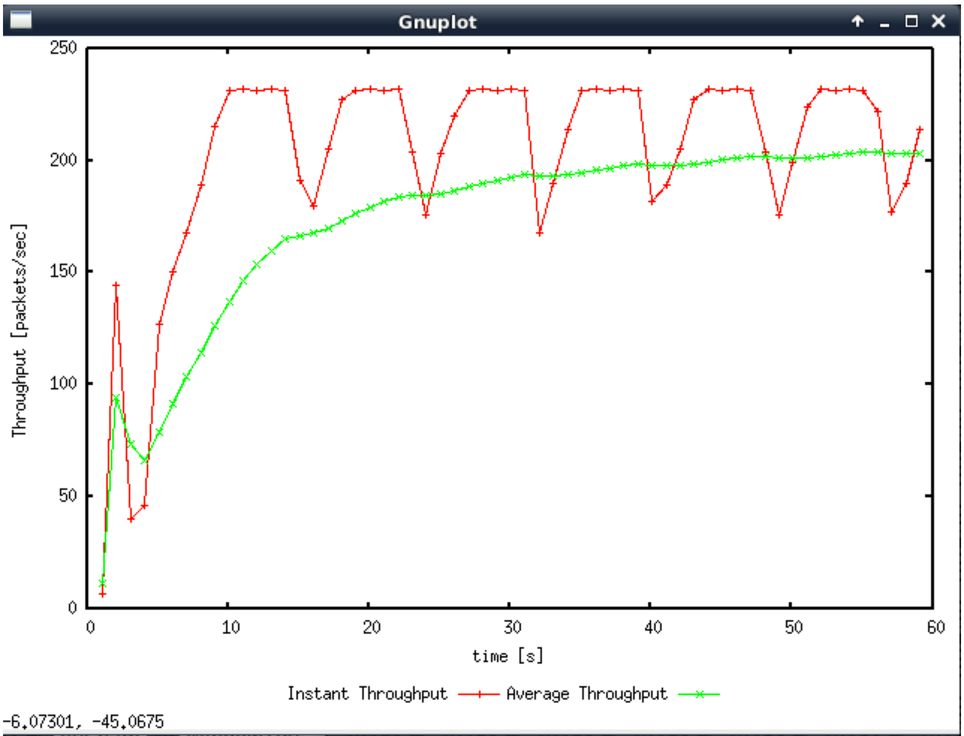




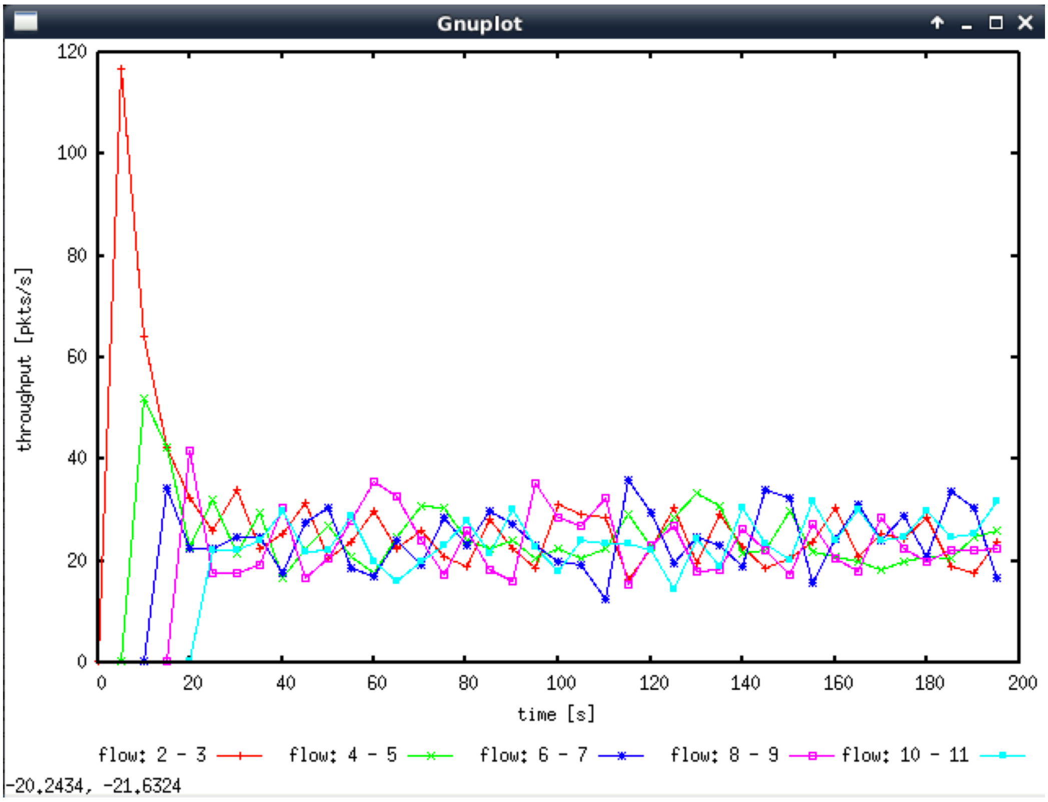
Question 4: Repeat the steps outlined in Question 1 and 2 (NOT Question 3) but for TCP Reno. Compare the graphs for the two implementations and explain the differences. (Hint: compare the number of times the congestion window goes back to zero in each case). How does the average throughput differ in both implementations?

- The window size only reaches zero only once after the slow-start, which is different from Tahoe. The average throughput for Reno is ~200 packet/sec, which is higher than Tahoe. Therefore TCP Reno performs better.





Exercise 2: Flow Fairness with TCP



Question 1: Does each flow get an equal share of the capacity of the common link (i.e., is TCP fair) ? Explain which observations lead you to this conclusion.

- Yes, each flow gets an equal share of the capacity of the common link.
- From the graph, we can see that although throughput for the first few connections starts at much higher point than the others, but all of them soon converges to an average point.

Question 2. What happens to the throughput of the pre-existing TCP flows when a new flow is created? Explain the mechanisms of TCP which contribute to this behaviour. Argue about whether you consider this behaviour to be fair or unfair.

- As mentioned in the above question, throughput for the first few connections starts at higher point and then decreases for each new flow that is created, then they all converge to an average point.
- Congestion control mechanism contributes to this behaviour.
- I argue this to be a fair behaviour, as each connection adjusting the size of the connection window when a new connection comes in allows sharing of the capacity of the common link.

Exercise 3: TCP competing with UDP

Question 1: How do you expect the TCP flow and the UDP flow to behave if the capacity of the link is 5 Mbps? Can you guess which colour represents the UDP flow and the TCP flow respectively ?

- TCP have flow control, so it will oscillate due to congestion control mechanism. In contrast, UDP uses all the capacity from the beginning of the transmit. Thus throughput for UDP will be higher than the TCP.
- Red: UDP, blue: TCP

Question 2: Why does one flow achieve higher throughput than the other? Try to explain what mechanisms force the two flows to stabilize to the observed throughput.

- UDP doesn't have congestion control mechanism.
- UDP does not wait for ACK.
- UDP doesn't care about packet loss.

Question 3: List the advantages and the disadvantages of using UDP instead of TCP for a file transfer, when our connection has to compete with other flows for the same link. What would happen if everybody started using UDP instead of TCP for that same reason?

- Advantage:
  - High speed
  - Connection setup not required
- Disadvantage:
  - No congestion control mechanism
  - Doesn't care about packet loss.

- If everyone started using UDP instead of TCP, the network will collapse.