CSC 458/2209: Computer Networking Systems, Fall 2025

Department of Computer Science, University of Toronto

Problem Set 2	**Date:** Tuesday, November 4, 2025
Name:	**Student ID:**

Instructions:

(i) This assignment has 5 problems and a total of 30 points. Some problems have multiple parts.

(ii) This assignment covers 0% of your final grade in this course. There will be an in-class quiz based on this assignment.

(iii) For the in-class quiz, please be clear and concise in your answers. Explain in your reasoning if needed. If the final answer is wrong, but your reasoning is correct you are going to receive partial credit for the problem.

(iv) For deadline and submission instructions, please see class website.

1) Nagle's Algorithm. In class we learned about Nagle's Algorithm, and how it tries to minimize TCP header's overhead while ensuring performance for both interactive and bulk transfers.

1a) (2 points) Consider a TCP connection between two nodes **X** and **Y** where an application running on node **X** sends the letters **ABCDEFGHIJLKMNOP** to **Y**. The application on **X** sends two letters back-to-back, then pauses for 1 second before sending two more letters. Assume the RTT of the connection is 4.1 seconds. Assuming the first packet is sent at time 0, show the departure time of all packets from **X** to **Y**, as well as the content of each packet (which letter or letters it carries). You can assume MSS is very large (say 1000 bytes) here, and Nagle's algorithm is enabled.

1b) (2 points) Assume mouse position changes are also sent over the connection every 0.5 second and that each mouse position contains 4 bytes of data. How would the user perceive the mouse motion with and without Nagle's algorithm?

2) Random Early Detection. Consider a router implementing Random Early Detection (RED) for congestion avoidance. The router's buffer has a maximum capacity of 100 packets. The RED parameters are set as follows:

- Minimum threshold **minTh**: 30 packets
- Maximum threshold **maxTh**: 70 packets
- Maximum drop probability **maxP**: 0.03

2a) (2 points) Can we ever have 90 packets queued in the router buffer? Explain your answer in 2-3 sentences.

2b) (2 points) Is it possible for RED to drop any packets when we have 25 packets in the buffer? Explain your answer in 2-3 sentences.

3. Punching Holes. Network Address Translation (NAT) is a networking technique that translates private IP addresses within a local network to a single public IP address when accessing the Internet. Let an address be represented by an (IP:Port) pair.

In the class, we learned that a NAT device maps internal addresses (*i_ipaddr*: *i_port*) to external addresses (*e_ipaddr*: *e_port*), where the external port, *e_port*, is selected at random. Packets from any outside host that is directed to (*e_ipaddr*: *e_port*) can traverse the mapping and be delivered to the internal host at (*i_ipaddr*: *i_port*). This is typically referred to as a **Full Cone NAT**.

A **Port Restricted NAT** is similar to full cone NAT with the exception that a packet from outside host (*h_ipaddr: h_port*) can traverse the NAT only if (*i_ipaddr*: *i_port*) has previously sent a message to (*h_ipaddr: h_port*).

Two computers, Machine **A** and Machine **B**, are both located behind separate NAT devices. Neither machine has a public IP address, but they need to establish a direct peer-to-peer connection to exchange data (such as in a VoIP or multiplayer gaming scenario).

3a) (2 points) Explain why using NAT devices prevents direct communication between Machine **A** and Machine **B.**

3b) (2 points) Let us assume Machines **A** and **B** are sitting behind full cone NAT devices. We have an external server **S** with a public IP address. How can we use this server to help Machines **A** and **B**

establish a direct connection even thought they are both sitting behind NAT devices? Explain the details of how this works. List all the packets sent by **A**, **B**, and **S** for the connection to be established. For each packet show the internal and external address (inside the local network, and after the NAT).

3c) (2 points) Does the the technique you described above work if **A** and **B** were sitting behind Port Restricted NAT devices? If not, what changes we need to make for the solution to work even with port restricted NAT devices?

3d) (2 points) The solution you described in parts 3b and 3c is called "hole punching" and is usually used for UDP connections. Can you think of why hole punching is more challenging for TCP connections compared to UDP? Explain your answer in 2-3 brief sentences.

4) Estimating RTT. In class we learned how TCP estimates the round-trip-time using a moving average:

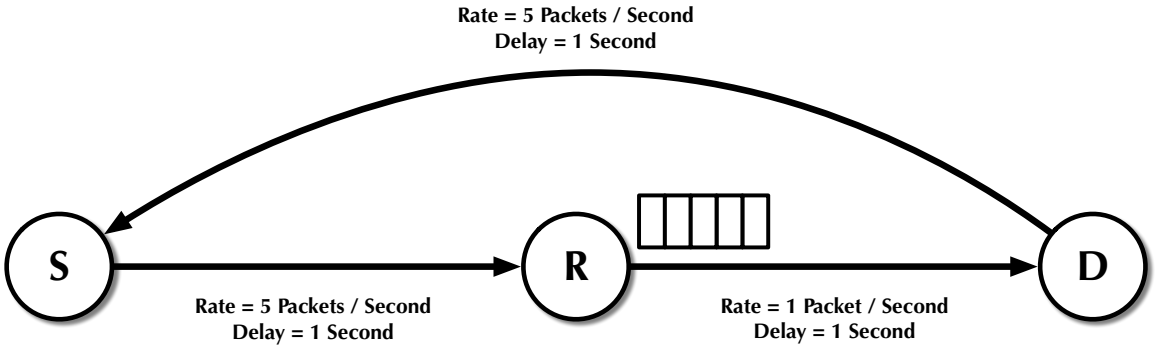
$$\text{EstimatedRTT} = \alpha \times \text{EstimatedRTT} + (1 - \alpha) \times \text{SampleRTT}.$$

Here, α is a constant, and SampleRTT is the RTT measured based on the most recent acknowledgement.

4a) (2 points) Let us assume $\alpha = 0.8$, and the initial EstimatedRTT is 200 ms. If every measured RTT is equal to 100 ms, how long does it take for the EstimatedRTT to reach a value less than 140 ms? How long would it take the EstimatedRTT to reach a value less than 140 ms if α is equal to 0.7?

4b) (2 points) If we want to ensure EstimatedRTT converges to a value of less than 140 ms in at most three RTTs, what is the maximum value that α can be set to in the equation above?

5. TCP Dynamics and Buffer Sizing. Consider two hosts **S** and **D** connected through a router **R** as depicted in the figure below. The capacity and delay of the links connecting **S** to **R**, **R** to **D**, and **D** back to **S** are shown in the figure. The source node **S** starts a TCP connection with destination **D**.

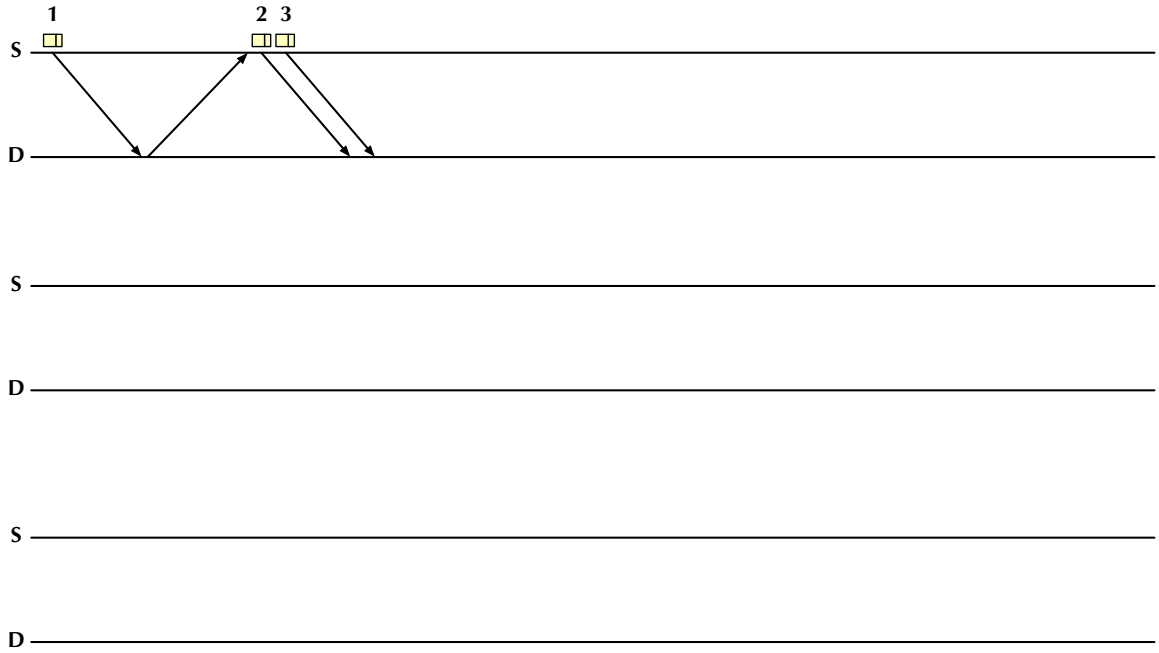


We make the following assumptions:

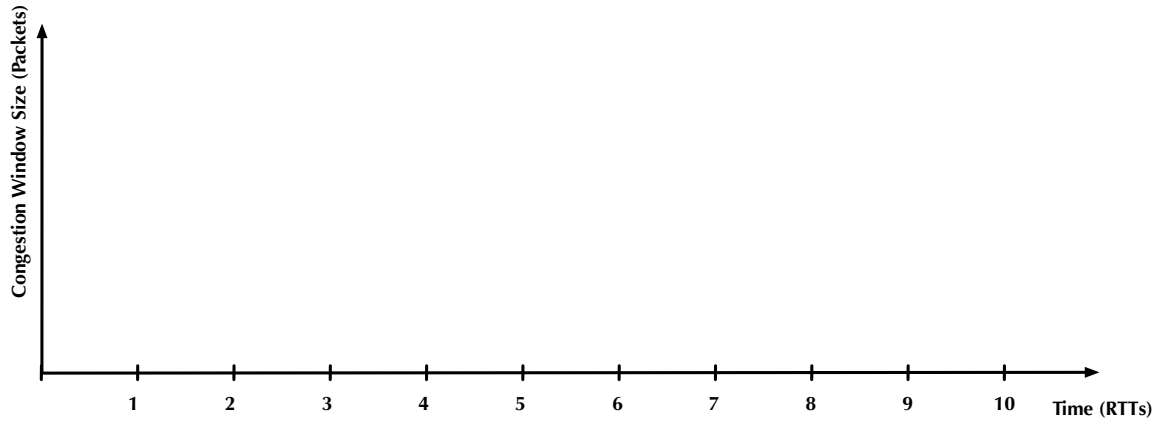
- The initial congestion window size is 1.
- We ignore the 3-way handshake.
- There is no slow-start, and the TCP connection is in additive-increase, multiplicative-decrease mode from the beginning.
- The link connecting the router **R** to **D** has a buffer of size 5 packets.
- The TCP connection from **S** to **D** is used to deliver a very large number of packets (i.e. flow will not end for the cases we consider here).

5a) (2 points) What is the maximum congestion window size for the TCP flow? What is the maximum rate that this TCP connection can achieve (in packets per second)?

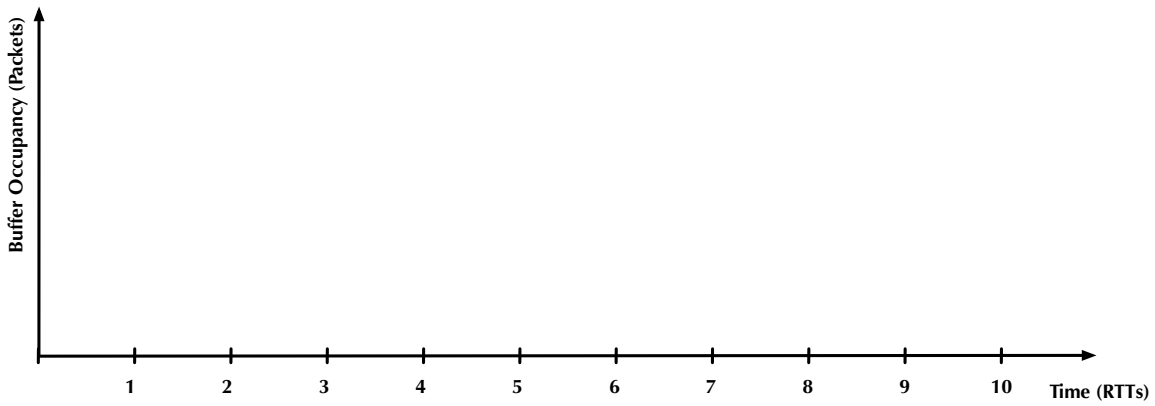
5b) (2 points) Show the sequence of packets sent from **S** to **D**, and the acknowledgements coming back from **D** to **S** in the following diagram. Do this for at least 12 RTTs.



5c) (2 points) Show the evolution of congestion window size (in packets) as a function of time (in RTTs) in the graph below. Clearly mark the Y-axis to show the size of congestion window in packets.



5d) (2 points) For the buffer on router **R** show the evolution of buffer occupancy (in packets) as a function of time in the graph below. Clearly mark the Y-axis to show the value of queue size in packets.



5e) (2 points) What is the average delay of packets in this TCP flow?