Homework 3: TCP Sliding Window

Due: Thursday, November 7 @ 11:59 pm EDT

Overview and instructions

This homework has 3 problems. Most problems involve writing a short response, or filling out some tables with your answers. You can write your responses in your own document or add annotations to this one.

Note on collaboration

You are welcome (and encouraged!) to collaborate with your peers, but the solutions you write down must be **your own work** (ie, written by you). You are responsible for independently understanding all work that you submit—after discussing a problem as a group, you should ensure that you are able to produce your own answers independently to ensure that you understand the problem. For more information, please see the course Collaboration Policy.

In your submission, we ask that you include a brief *collaboration statement* describing how you collaborated with others on each problem—see the next section for details.

How to submit

You will submit your work in PDF form on Gradesope. Your PDF should conform to the following requirements:

- Please **do not** include any identifying information (name, CS username, Banner ID, etc.) in your PDF, since all homeworks are graded anonymously
- Each problem (where "problem" is one of the Problems 1–4) should start on a separate page. When you submit on Gradescope, you will be asked to mark which pages correspond to which problem
- At the start of each problem, write a brief *collaboration statement* that lists the names and CS usernames of anyone you collaborated with and what ideas you discussed together
- If you consulted any outside resources while answering any question, you should cite them with your answer

1 TCP sequence numbers

Relevant lectures: Lecture 13–15

Suppose A and B create a TCP connection with initial sequence numbers 5000 and and 80000, respectively, and an initial window of 4000 bytes. The table below depicts the flow of the connection, which has 3 main events:

1. A sends three 200-byte segments, (Which we will name DataA1, DataA2, and DataA3), and B sends ACKs for each.
2. Between segments DataA2 and DataA3, the application on B calls `read()` on the socket associated with this connection, which returns 400 bytes.
3. B sends a 200-byte segment DataB1 to A and begins the connection termination process with a FIN.

In the table, fill in the SEQ, ACK, and WIN fields for each packet shown, given the initial sequence numbers and window sizes.

> **Hint**: Try to create a similar connection flow using the TCP reference, while looking at the packets sent in Wireshark—this should allow you to view the changes in sequence numbers, and window sizes. Another reference that may be useful is Section 17.3 of the Dordal textbook[1].

t	Packets sent by A	Packets sent by B
0	SYN, seq=5000, win=4000	
1		SYN,ACK, seq=80000, ack=**5001**, win=4000
2	ACK, seq=**5001**, ack=**80001**, win=4000	
3	ACK, seq=**5001**, ack=**80001**, win=**4000**, data=DataA1	
4		ACK, seq=**80001**, ack=**5201**, win=**3800**
5	ACK, seq=**5201**, ack=**80001**, win=**4000**, data=DataA2	
6		ACK, seq=**80001**, ack=**5401**, win=**3600**
7		[*B calls* `read()`*, which returns 400 bytes*]
8	ACK, seq=**5401**, ack=**80001**, win=**4000**, data=DataA3	
9		ACK, seq=**80001**, ack=**5601**, win=**3800**
10		ACK, seq=**80001**, ack=**5601**, win=**3800**, data=DataB1
11	ACK, seq=**5601**, ack=**80201**, win=**3800**	
12		FIN,ACK, seq=**80201**, ack=**5601**, win=**3800**
13	. . .	

Grading rubric:

- +3pts: Handshake sequence numbers increment correctly
- +3pts: Sequence numbers correct during data transmission
- +3pts: ACK numbers correct during data transmission
- +3pts: Window size decrease before `read()`

- +3pts: Window size updates appropriately after `read()`

For each item: 3/3 if values are correct, 2/3 for a minor issue, 1/3 for multiple issues, 0/3 if major conceptual problem or answer missing.

Note: Since the fields for each packet depend on each other, if a student makes a mistake on one value, continue grading as if their answer were correct—ie, students should not be penalized multiple times for the same mistake.

2 Flow control

Relevant lectures: Lecture 14–15

Suppose A and B establish a TCP connection and complete the handshake, with packets (and their SEQ, ACK, WIN fields) as shown in the table.

After the handshake, A begins sending data. If A has an infinite amount of data to send, **how much data can A send before receiving any ACKs from B**?

Express your answer by filling in the table with the segments that A would send until it needs to stop sending, based on the following conditions/assumptions:

- Both A and B use a maximum segment size (MSS) of 200 bytes
- B sends no packets after the handshake (imagine that the latency is *very* high, so A stops sending long before any ACKs from B arrive).
- You can ignore retransmissions, zero-window probing, and any TCP features we haven't discussed in class (keep-alive packets, window updates, etc.). In other words, don't consider any segments other than normal data segments.
- You can ignore congestion control. (Perhaps it's 1981 and congestion control doesn't exist yet.)

The table includes a template for one example segment. Add in any other segments you may need (you may have extra rows at the end of your table). For the data field, just write in the *amount* of data in the segment (eg. "data=(10 bytes)").

t	Packets sent by A	Packets sent by B
0	SYN, seq=0, win=1000	
1		SYN,ACK, seq=0, ack=1, win=512
2	ACK, seq=1, ack=1, win=1000	
3	ACK, seq=**1**, ack=**1**, win=**1000**, data=**(200 bytes)**	
4	ACK, seq=**201**, ack=**1**, win=**1000**, data=**(200 bytes)**	
5	ACK, seq=**401**, ack=**1**, win=**1000**, data=**(112 bytes)**	
6		
7		
8		
9	. . .	

Grading rubric:

1. +3pts: Sender respect receiver's window: data sent must be finite (+1.5), must send at most 512 bytes total (+1.5)
2. +3pts: Packet sizes are correct (all should be len=MSS, until last packet)
3. +3pts: Sequence numbers are correct
4. +3pts: ACK and window values are correct

For items 2-4: 3/3 if values are correct, 2/3 for a minor issue, 1/3 for multiple issues, 0/3 if major conceptual problem or answer missing.

Note: Since the fields for each packet depend on each other, if a student makes a mistake on one value, continue grading as if their answer were correct—ie, students should not be penalized multiple times for the same mistake.

3 Super quick tutorial: NAT

Relevant lectures: Lecture 8 (A throwback, because Nick forgot to add this to HW2.)

Now that we've learned about NAT in class, you can observe it in real life:

1. Open up the network settings on our computer and find *your* IP address for the interface you use to connect to the Internet (eg. your Wifi interface):
   - On a Mac: Open up **System Settings** → Type **"TCP/IP"** in the search bar → Click on the result. Your IP address should be displayed next to "IP address" in the box that pops up.
   - On Windows: In the settings app, go to **Network & Internet** → **Wifi** → Click on <**your wifi name**> **properties**. Your IP should be displayed next to "IPv4 address".
2. In your browser, search for "my IP address" in Google.

In your submission, write down both IPs (you can change the last byte if you don't want to reveal it[2]).

Then, consider: **are you connected to the Internet via a NAT? Explain your reasoning (≤ 1 sentence).**

6pts total: +3 for including IP addresses, +3 for correct reasoning about NAT based on the IPs listed.

[2] Bonus discussion questions (just for fun, not part of this assignment): a) Search both IP addresses in Google–what can someone learn about you from each IP? b. It's likely both IPs will be different by the time someone reads your HW submission, why?