Networks Homework 5 (Due Friday 9/18/2020)

1. Suppose a sender is sending segments of size 250 bytes. If a receiver sends and acknowledgement with number 20000, that means

   (a) the first 19999 segments all arrived.

   (b) segment number 20000 arrived, though earlier segments might not have.

   (c) all the bytes 0 to 19999 have arrived.

   (d) bytes 19750 to 19999 have arrived, though some earlier bytes might not have.

2. To slow down a sender, a receiver should

   (a) drop back to slow start

   (b) only ACK every 2nd packet

   (c) stop sending ACKs entirely

   (d) reduce their window size

3. TCP provides each the following. Which ones are also provided by UDP? Choose all that apply.

   (a) ability to detect lost packets

   (b) ability to put packets in right order

   (c) congestion control

   (d) checksum

4. What is the official name of the solution to silly window syndrome?

5. A sender sends a TCP segment that contains byte numbers 4500 to 5720 of data stream. This happens to be the 7th segment sent.

   (a) What is the sequence number of this segment?

   (b) What is the acknowledgement number the receiver would use to acknowledge this segment?

6. Suppose a sender sends segments 1, 2, 3, 4, 5, and 6. Suppose segment 4 was lost.

   (a) We could ACK segments 1, 2, and 3 separately. To accomplish the same thing with a single ACK what can we do?

   (b) Is it possible to use ACKs to acknowledge 1, 2, 3, and 5, but not 4?

7. This is about TCP flow control. Suppose a sender has sent a bunch of full-sized segments, with the MSS at 500 bytes, and with the last segment sent having sequence number 9500. An ACK comes in with ACK number 6000 and a window size of 8000 bytes. What is the maximum number of new segments the sender is allowed to send?

8. Shown below is a Wireshark trace of an HTTP request on my computer. Answer the questions that follow.

No.	Time	Source	Destination	Protocol	Length	Info
868	38.456874	192.168.1.112	128.118.28.29	TCP	66	9066 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
870	38.473345	128.118.28.29	192.168.1.112	TCP	66	80 → 9066 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=128
871	38.473442	192.168.1.112	128.118.28.29	TCP	54	9066 → 80 [ACK] Seq=1 Ack=1 Win=16384 Len=0
872	38.473649	192.168.1.112	128.118.28.29	HTTP	524	GET / HTTP/1.1
875	38.495036	128.118.28.29	192.168.1.112	TCP	54	80 → 9066 [ACK] Seq=1 Ack=471 Win=15744 Len=0
878	38.508825	128.118.28.29	192.168.1.112	TCP	1514	80 → 9066 [ACK] Seq=1 Ack=471 Win=15744 Len=1460 [TCP segment of a reassemble
879	38.509755	128.118.28.29	192.168.1.112	TCP	1514	80 → 9066 [ACK] Seq=1461 Ack=471 Win=15744 Len=1460 [TCP segment of a reassem
880	38.509857	192.168.1.112	128.118.28.29	TCP	54	9066 → 80 [ACK] Seq=471 Ack=2921 Win=16384 Len=0
881	38.510286	128.118.28.29	192.168.1.112	TCP	1514	80 → 9066 [ACK] Seq=2921 Ack=471 Win=15744 Len=1460 [TCP segment of a reassem
882	38.511464	128.118.28.29	192.168.1.112	TCP	1514	80 → 9066 [ACK] Seq=4381 Ack=471 Win=15744 Len=1460 [TCP segment of a reassem
883	38.511544	192.168.1.112	128.118.28.29	TCP	54	9066 → 80 [ACK] Seq=471 Ack=5841 Win=16384 Len=0
884	38.512052	128.118.28.29	192.168.1.112	TCP	1514	80 → 9066 [ACK] Seq=5841 Ack=471 Win=15744 Len=1460 [TCP segment of a reassem
885	38.514977	128.118.28.29	192.168.1.112	TCP	1450	80 → 9066 [PSH, ACK] Seq=7301 Ack=471 Win=15744 Len=1396 [TCP segment of a re
886	38.514980	128.118.28.29	192.168.1.112	HTTP	59	HTTP/1.1 200 OK (text/html)
887	38.514981	128.118.28.29	192.168.1.112	TCP	54	80 → 9066 [FIN, ACK] Seq=8702 Ack=471 Win=15744 Len=0
888	38.515130	192.168.1.112	128.118.28.29	TCP	54	9066 → 80 [ACK] Seq=471 Ack=8703 Win=16384 Len=0
889	38.515961	192.168.1.112	128.118.28.29	TCP	54	9066 → 80 [FIN, ACK] Seq=471 Ack=8703 Win=16384 Len=0
891	38.532852	128.118.28.29	192.168.1.112	TCP	54	80 → 9066 [ACK] Seq=8703 Ack=472 Win=15744 Len=0

(a) The first thing under the "Info" section is usually either 9066 → 80 or 80 → 9066. What are those two numbers?

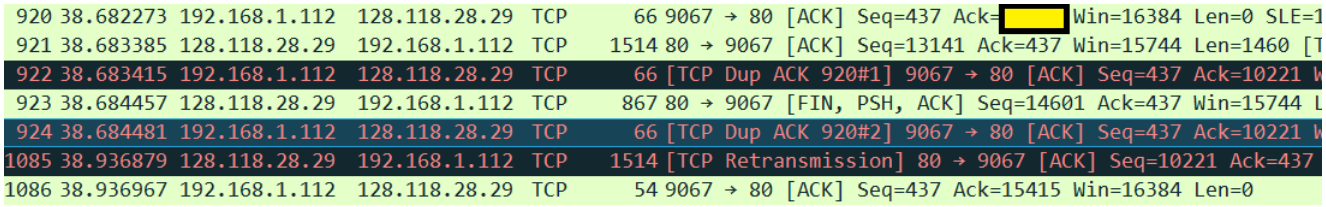
(b) Which segment numbers (indicated in the first column) are involved in the 3-way handshake?

(c) Both sides have agreed on a limit to the size of the segments they will send. What is that size?

(d) Some TCP implementations don't send ACKs for every segment, but rather send them only for every couple of segments. Is my machine doing that?

(e) Which segment numbers are involved in closing the connection?

9. Shown below is a Wireshark trace. A particular ACK number has been blocked out. What is that number?



920	38.682273	192.168.1.112	128.118.28.29	TCP	66	9067 → 80 [ACK] Seq=437 Ack=[redacted] Win=16384 Len=0 SLE=1
921	38.683385	128.118.28.29	192.168.1.112	TCP	1514	80 → 9067 [ACK] Seq=13141 Ack=437 Win=15744 Len=1460 [T
922	38.683415	192.168.1.112	128.118.28.29	TCP	66	[TCP Dup ACK 920#1] 9067 → 80 [ACK] Seq=437 Ack=10221 W
923	38.684457	128.118.28.29	192.168.1.112	TCP	867	80 → 9067 [FIN, PSH, ACK] Seq=14601 Ack=437 Win=15744 L
924	38.684481	192.168.1.112	128.118.28.29	TCP	66	[TCP Dup ACK 920#2] 9067 → 80 [ACK] Seq=437 Ack=10221 W
1085	38.936879	128.118.28.29	192.168.1.112	TCP	1514	[TCP Retransmission] 80 → 9067 [ACK] Seq=10221 Ack=437
1086	38.936967	192.168.1.112	128.118.28.29	TCP	54	9067 → 80 [ACK] Seq=437 Ack=15415 Win=16384 Len=0

10. In class we talked about common window sizes. You can actually compute what the window size can max out at. The formula is to multiply the round-trip-time by the bandwidth. A typical round trip time is 100 ms and a typical bandwidth is 10 mbps. What is the window size in *bytes* according to the formula? [Note: be careful to make sure all the units match.]