To create my socket interface, I decided I would implement a write, read, listen, connect, and close command for the socket, as well as the ability to see the remaining receive and write buffer size. In order to call LISTEN, the caller simply says which port they want to listen on, and then the socket interface stores a list of listening ports. In order to connect, the socket interface sends a SYN packet with the destination port and source port, and then the server checks if it is listening on that port, and if it does then it sends back a SYN ACK. Once the handshake process is finished, each node initializes sequence numbers at 0, and both can write or read. Packets are sent according to the congestion window, which is initialized at 1. The amount of packets currently in flight is stored, so if the congestion window is 5 but 5 packets are in flight, no more data will be sent until an ack is received. The RTTs are initialized as 1000ms, and timeouts are calculated with RTT + 4 * RTTvar, in order to account for shakiness in the network. Initially, all sockets start in slowstart, and so each packet received adds 1 to the congestion window, but once the first packet has been timed out, the congestion window will be divided by 2 and packets received will instead increment congestion window by 1/congestion_window. One thing I noticed is that even if no packets are lost on the network, they are still extremely likely to timeout for high congestion window values, because, for instance, if 4 packets are sent out at once, they will all have the same timeout time, but each packet after the 1st packet gets delayed behind the next, so that the 4th packet ACK arrives without any loss, but can still be timed out. I could possibly fix this without delaying the time at which packets are sent, or changing the timeout to be calculated from the last ACK received, rather than the time the packet was sent out. Anyways, I tested my implementation and I was able to send 10000 bytes of data over the network even using the noise simulations, and handshaking for establishing / closing connections works, so it appears to be fully functional.