Andrew Reyes
Dongjun Cho
CSE 160

Project 3: Reliable Transport and Congestion Control

We begin by implementing the Transport interface and setting up a connection using a three-way handshake. First we begin by setting up a server at a node address and bind it to a port. This socket then listens for connections. Next we initiate a client and bind it to a source port. Here we make a call to Transport.connect() where it makes the connection with the address and sends out a SYN pack with PROTOCOL_TCP protocol. We made a TCPProtocol function to handle all the TCP packets.

Inside the TCPProtocol function we receive the first SYN packet update the socket state and make a new SYN_ACK pack that we send out to the next hop. In our flag = 2 block we receive the SYN_ACK and reply with a ACK and set the connection to ESTABLISHED. Here we also have a call to Transport.write() which writes to the socket from a buffer and sends out a DATA pack. Our next flag, 3, receives the ACK and completes the 3-way handshake. Our next flag is the DATA flag where the data is received and then sends out a DATA_ACK packet. The handling of the data and calculation of effective windows are done in the Transport.read() and Transport.write() functions. The fifth flag is a where the DATA_ACK is received and the data has reached its destination and is written onto the socket. Our closeClient command creates and sends out the FIN packet where it then goes back to our final sixth flag in TCPProtocol. Here it makes a call to Transport.close() which handles the final closing of the socket.