Darrell Percey && Andrew Boven

3/30/17

Sys Net II Project 2

Protocol

Packet-

For the packet, we had 10Bytes for the segment. This allowed for us to use 1byte for a sequence number(int8_t), 2bytes for a checksum(uint16_t), and the remaining 7bytes to fit our piece of the message. The Sender, Receiver, and Network use this same packet style. Sequence numbers were either 0 or 1 depending on which sequence it was on. If a 2 was sent that was considered a “FIN” request.

Sender-

The sender was the main driver of the three programs. Starting off the sender would create and packet and send it out. It would then wait for 5 seconds for an ACK to be received. If no ACK is received it will timeout and resend the last packet just sent out. If a ACK is received it will check to see if it is corrupt and it’s the correct sequence. If it is corrupt or the wrong sequence it will timeout and resend the packet again. If the packet was correct however it will change to the next sequence number (0 or 1), make the next packet and send it. Repeating from the top. Once the sender segment length (variable in program) gets bigger than the length of the message being sent it will try to send a FIN request and wait for a FIN ACK.

Receiver-

The receiver will wait for a packet to arrive and decide based on the sequence number if it is the next piece to the message. If it is the next piece and it is not corrupt it will add it to the completed string. If it is corrupt then it will drop the packet allowing the sender to time out. If the packet is the wrong sequence number, it will send it back to the sender with an ACK letting the sender know that they received that packet. Once a 2 for the sequence number is received then it knows that it is finished making the string as this is a FIN request.

Network-

The network starts by receiving a packet and immediately forwarding it after deciding if it should be a lost/delayed/corrupted packet. Once it receives a packet it will read the destination ip and port from the packet. Once it has the destination set it will forward the packet to the sender or receiver whichever the destination is.

Problems/Extra –

As for problems the only one that occurred was the SSH server freezing on the first call of Select() in the sender if the sender didn’t receive at least one packet. This was fixed by sending the sender a fake packet from itself. Once it pushes one packet through it will timeout appropriately. Also in the packet structure we used a sequence number as well as a true checksum. The program will use 1byte for a sequence number, 2 bytes for the checksum as it can do overflow so no additional bytes were needed. The remaining bytes were used for the string to be transferred. If the sender and receiver are ran on the same server with the same IP the statistics will display that Sender is the only one sending as it compares the first IP from the first packet as the “sender” and the receiver is packets received from a different IP. So if both are run with the same the network statistics will believe it’s the sender each time from similar IP.