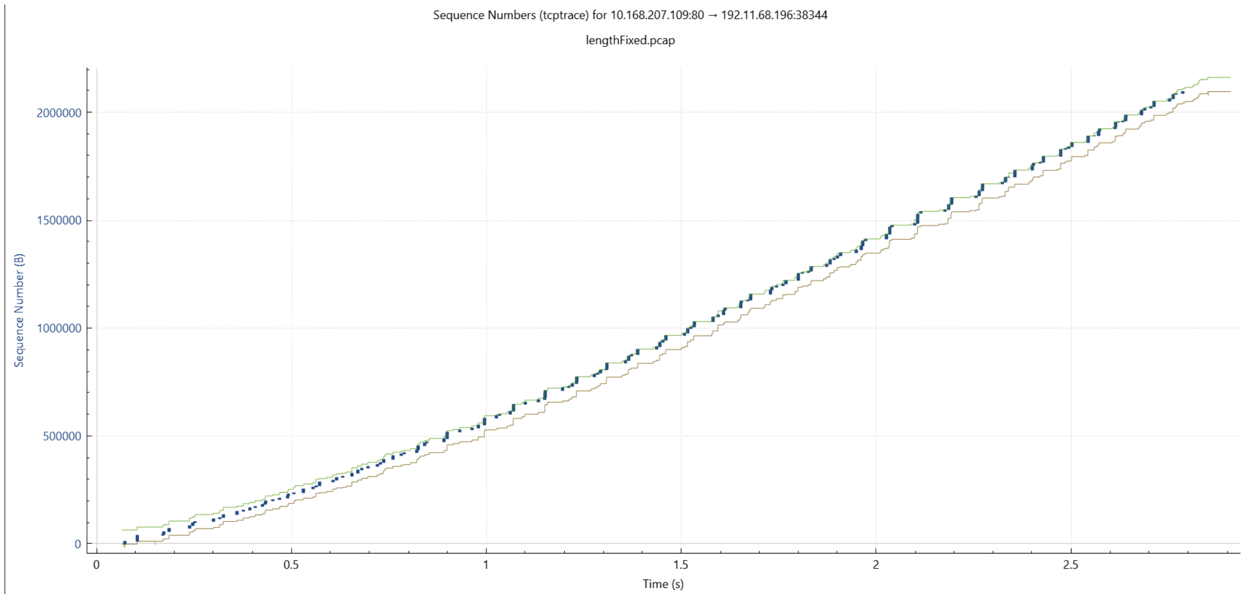
Prof. Kim **Jeffrey Jacobson**

Zonghua Ouyang

Project #1 Problem 2

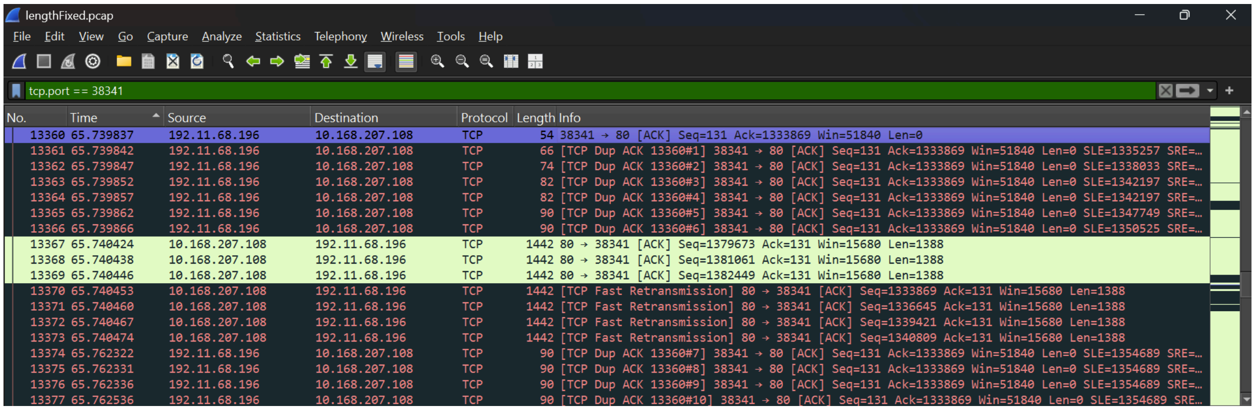
Figure below is the TCP Sequence Graph for the connection between Server #2 (10.168.207.109:80) and Client #2 (192.11.68.196:38344). This connection is an example of a stable TCP connection. The discrete blue lines represent the individual TCP segments being sent - the line length scales with the size of the segment. The orange line at the bottom shows the progress of the ACK number from the client receiver, which increments with every packet received by the client. From the analysis, it can be seen that Client #2 advertises a constant 65535 Byte receive window to Server #2 (Scaling Factor of 1), which fits with the linear growth of the green receive-window line (advertised RWIND + Receiver ACK Number). Since the blue packet lines remain very close to the RWIND slope throughout, a pattern can be seen of Server #2 transmitting to the full capability of Client #2's RWIND, then waiting for an ACK from the client to continue transmission. The linear behavior of the graph demonstrates the consistent throughput from server to client, though its throughput is also noticeably lower than Server #1's due to the stricter window scale (719KBps vs Server #1's 961KBps).



Sequence graph for Server #2 → Client #2

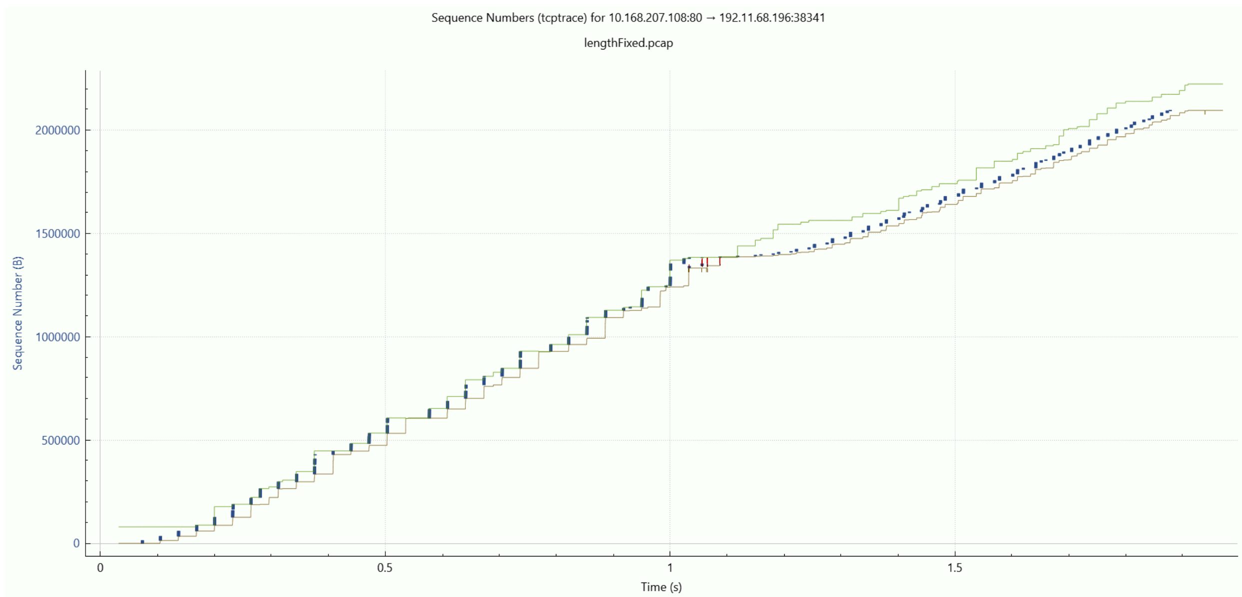
Below is the TCP sequence graph for the connection between Server #1 (10.168.207.108:80) and Client #1 (192.11.68.196:38341). In this case, the client advertises a window scaling factor of 128 to the server using its initial SYN packet as seen in the session analysis. Compared to the consistent throughput as seen with Client/Server #2, this connection shows a faster throughput in the beginning, given the sharper slope. This can be attributed to the larger window size provided to the server. However, this connection also shows greater instability as the server attempts to keep up with the client's window size. Numerous fast retransmissions can be seen after 1 second into the connection life due to duplicate ACKs sent from the client. The server then transitions to a slow-start state, creating a large gap between the receive window and the ACK sequence number.

Both connections demonstrate the impact window scaling can have on the efficiency of a TCP connection. Using default window scaling can keep transmissions stable and lines uncongested, but at the cost of a smaller throughput and underutilization of the bandwidth available. Increased window scaling creates potential for greater throughput in a connection, but is best reserved for when both ends of the connection can take full advantage of the available buffer.

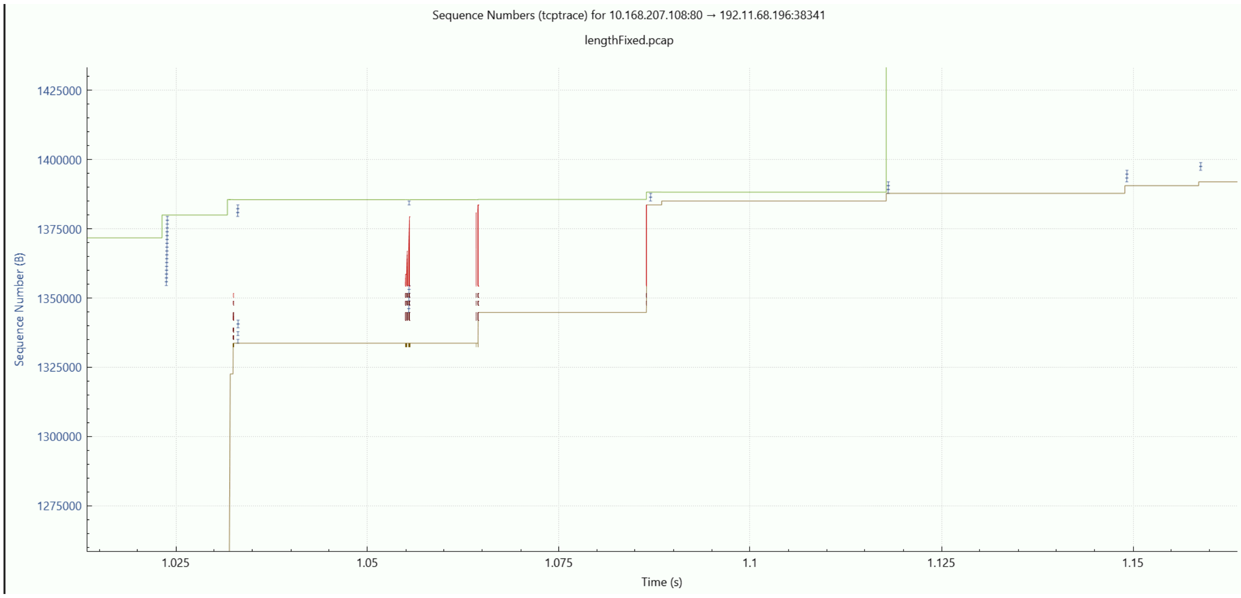


No.	Time	Source	Destination	Protocol	Length	Info
13360	65.739837	192.11.68.196	10.168.207.108	TCP	54	38341 → 80 [ACK] Seq=131 Ack=1333869 Win=51840 Len=0
13361	65.739842	192.11.68.196	10.168.207.108	TCP	66	[TCP Dup ACK 13360#1] 38341 → 80 [ACK] Seq=131 Ack=1333869 Win=51840 Len=0 SLE=1335257 SRE=…
13362	65.739847	192.11.68.196	10.168.207.108	TCP	74	[TCP Dup ACK 13360#2] 38341 → 80 [ACK] Seq=131 Ack=1333869 Win=51840 Len=0 SLE=1338033 SRE=…
13363	65.739852	192.11.68.196	10.168.207.108	TCP	82	[TCP Dup ACK 13360#3] 38341 → 80 [ACK] Seq=131 Ack=1333869 Win=51840 Len=0 SLE=1342197 SRE=…
13364	65.739857	192.11.68.196	10.168.207.108	TCP	82	[TCP Dup ACK 13360#4] 38341 → 80 [ACK] Seq=131 Ack=1333869 Win=51840 Len=0 SLE=1342197 SRE=…
13365	65.739862	192.11.68.196	10.168.207.108	TCP	90	[TCP Dup ACK 13360#5] 38341 → 80 [ACK] Seq=131 Ack=1333869 Win=51840 Len=0 SLE=1347749 SRE=…
13366	65.739866	192.11.68.196	10.168.207.108	TCP	90	[TCP Dup ACK 13360#6] 38341 → 80 [ACK] Seq=131 Ack=1333869 Win=51840 Len=0 SLE=1350525 SRE=…
13367	65.740424	10.168.207.108	192.11.68.196	TCP	1442	80 → 38341 [ACK] Seq=1379673 Ack=131 Win=15680 Len=1388
13368	65.740438	10.168.207.108	192.11.68.196	TCP	1442	80 → 38341 [ACK] Seq=1381061 Ack=131 Win=15680 Len=1388
13369	65.740446	10.168.207.108	192.11.68.196	TCP	1442	80 → 38341 [ACK] Seq=1382449 Ack=131 Win=15680 Len=1388
13370	65.740453	10.168.207.108	192.11.68.196	TCP	1442	[TCP Fast Retransmission] 80 → 38341 [ACK] Seq=1333869 Ack=131 Win=15680 Len=1388
13371	65.740460	10.168.207.108	192.11.68.196	TCP	1442	[TCP Fast Retransmission] 80 → 38341 [ACK] Seq=1336645 Ack=131 Win=15680 Len=1388
13372	65.740467	10.168.207.108	192.11.68.196	TCP	1442	[TCP Fast Retransmission] 80 → 38341 [ACK] Seq=1339421 Ack=131 Win=15680 Len=1388
13373	65.740474	10.168.207.108	192.11.68.196	TCP	1442	[TCP Fast Retransmission] 80 → 38341 [ACK] Seq=1340809 Ack=131 Win=15680 Len=1388
13374	65.762322	192.11.68.196	10.168.207.108	TCP	90	[TCP Dup ACK 13360#7] 38341 → 80 [ACK] Seq=131 Ack=1333869 Win=51840 Len=0 SLE=1354689 SRE=…
13375	65.762331	192.11.68.196	10.168.207.108	TCP	90	[TCP Dup ACK 13360#8] 38341 → 80 [ACK] Seq=131 Ack=1333869 Win=51840 Len=0 SLE=1354689 SRE=…
13376	65.762336	192.11.68.196	10.168.207.108	TCP	90	[TCP Dup ACK 13360#9] 38341 → 80 [ACK] Seq=131 Ack=1333869 Win=51840 Len=0 SLE=1354689 SRE=…
13377	65.762536	192.11.68.196	10.168.207.108	TCP	90	[TCP Dup ACK 13360#10] 38341 → 80 [ACK] Seq=131 Ack=1333869 Win=51840 Len=0 SLE=1354689 SRE…

Fast retransmissions from Server #1 after detecting duplicate ACKs from Client #1



Sequence Graph of Server #1 → Client #1



Duplicate ACKs can be seen by the hitches under the yellow ACK line. Retransmissions can be seen by the long red lines.