Haotian Yan 1855119

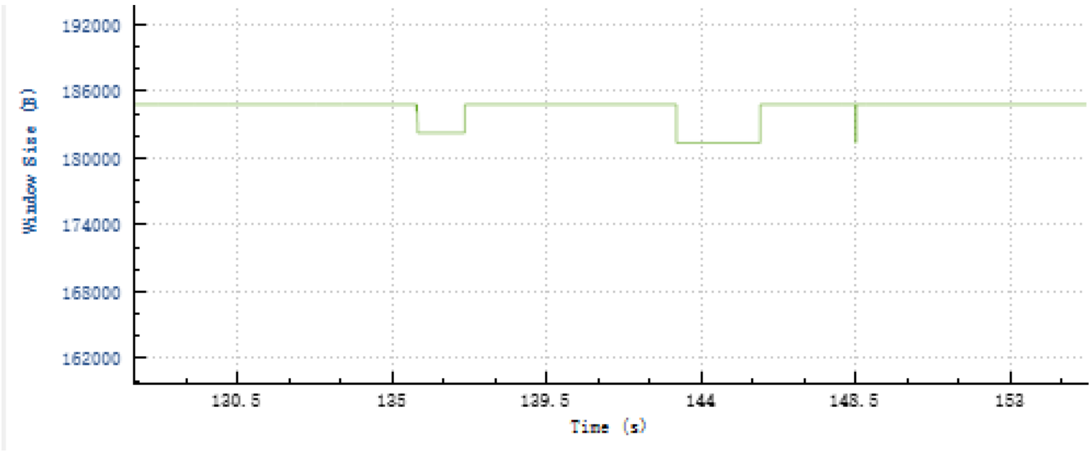
Part 1:

The retransmission timer is used to determine whether the retransmission is necessary or not, and it is responsible for maintaining the retransmission timeout (RTO) value. When a packet is sent, but the recipient has not sent a TCP ACK packet, the host assumes that the original packet was lost and retransmits the original packet. When the retransmission is sent, the RTO value is doubled; if no ACK packet is received before that value is reached, another retransmission will occur. If an ACK is not received, the RTO value will be doubled for the next retransmission. This process will continue until the ACK packet is received or until the sender reaches the maximum number of transmission attempts.

If the receiver receives an out-of-order packet, a repeated TCP ACK packet will be sent. For example, as the picture shown below, the TCP Dup ACK appeared for six times, 328#1, 328#2, 328#3, 328#4, 328#5, 328#6. Under normal circumstances, a TCP ACK packet should be seen soon after the first packet is sent. In this case, however, the TCP retransmission and TCP Dup ACK appeared. After the host receives multiple duplicate ACKs from the recipient, it assumes that the packet was indeed lost in transit, and send a Fast Retransmission immediately. Once a fast retransmission is triggered, all other packets being transmitted are queued until the fast retransmission packet is sent.

338	88.962128	172.22.254.123	23.214.162.1…	TCP	78	[TCP Dup ACK 328#2] 62082 → 80 [ACK] Seq=383 Ack=8245 Win=131072 Len=0 TS
339	88.962128	172.22.254.123	23.214.162.1…	TCP	78	[TCP Dup ACK 328#3] 62082 → 80 [ACK] Seq=383 Ack=8245 Win=131072 Len=0 TS
340	88.962128	172.22.254.123	23.214.162.1…	TCP	78	[TCP Dup ACK 328#4] 62082 → 80 [ACK] Seq=383 Ack=8245 Win=131072 Len=0 TS
341	88.962129	172.22.254.123	23.214.162.1…	TCP	78	[TCP Dup ACK 328#5] 62082 → 80 [ACK] Seq=383 Ack=8245 Win=131072 Len=0 TS
342	88.962129	172.22.254.123	23.214.162.1…	TCP	78	[TCP Dup ACK 328#6] 62082 → 80 [ACK] Seq=383 Ack=8245 Win=131072 Len=0 TS
343	88.962378	23.214.162.177	172.22.254.1…	TCP	1440	80 → 62082 [ACK] Seq=17863 Ack=383 Win=30080 Len=1374 TSval=1200014275 TSe
344	88.962383	23.214.162.177	172.22.254.1…	TCP	1440	[TCP Fast Retransmission] 80 → 62082 [ACK] Seq=8245 Ack=383 Win=30080 Len=
345	88.962384	23.214.162.177	172.22.254.1…	TCP	1440	80 → 62082 [ACK] Seq=19237 Ack=383 Win=30080 Len=1374 TSval=1200014275 TSe

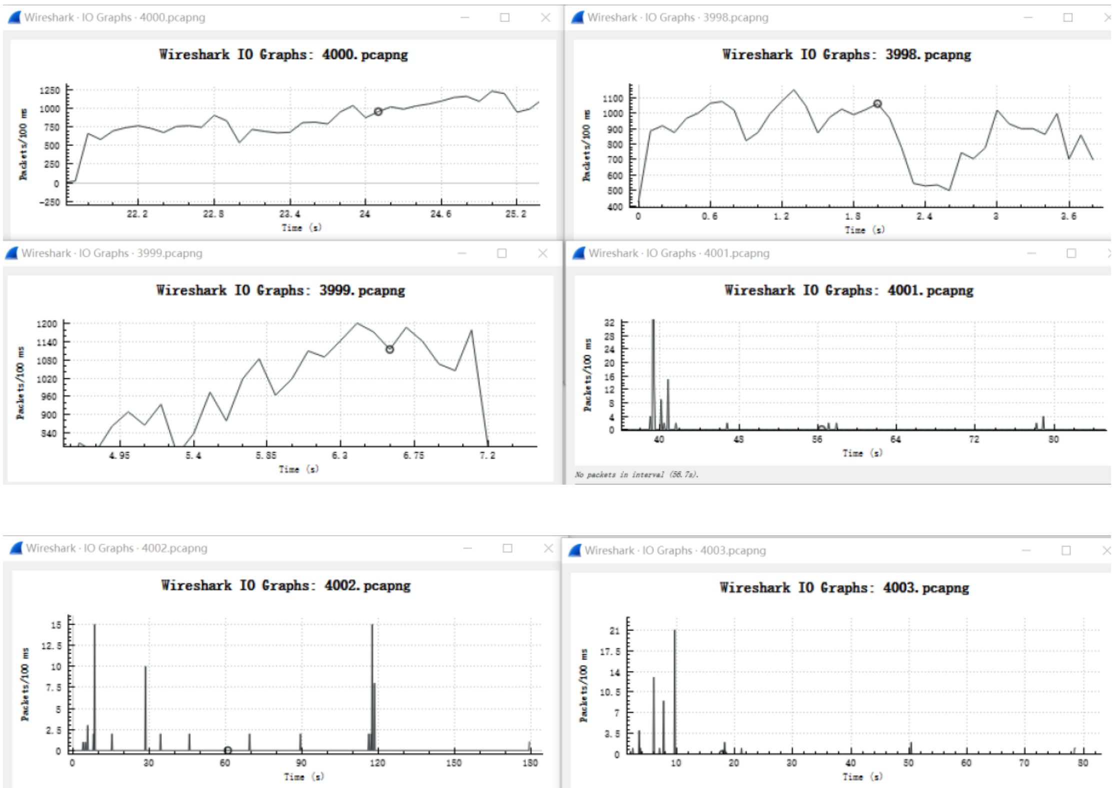
TCP implements a window-adjusting mechanism to deal with this situation. When a server becomes too busy to process data at the rate of receive window, it decreases the size of the window and less data is sent by the client, and the server can process its buffer contents at an acceptable rate that allows data to flow. And after this, the window size will be adjusted to the normal value, which could be seen below.



Part 2: compare the behavior of ports 4000, 3998(5% loss), 3999(10% loss), 4001(20% loss), 4002(40% loss) and 4003(60% loss).

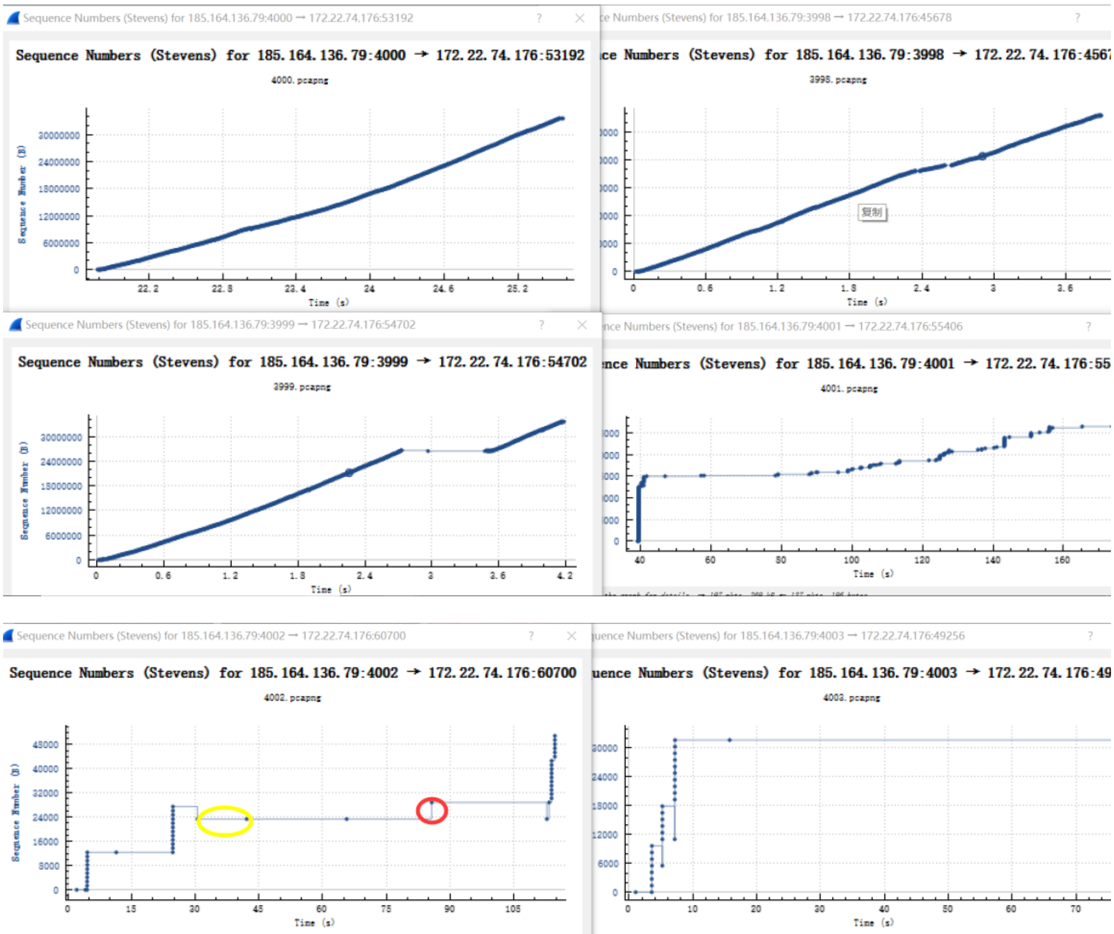
IO Graph

The IO Graph window shows a graphical view of the throughput of data, and IO Graph for each port has been show below. We can see that, when the transfer is relatively good (port 4000), the transfer rate curve is relatively smooth and the transfer rates have similar values. However, when the data loss increases, the consistency of IO Graph decreases, and the fluctuation of curve increases.



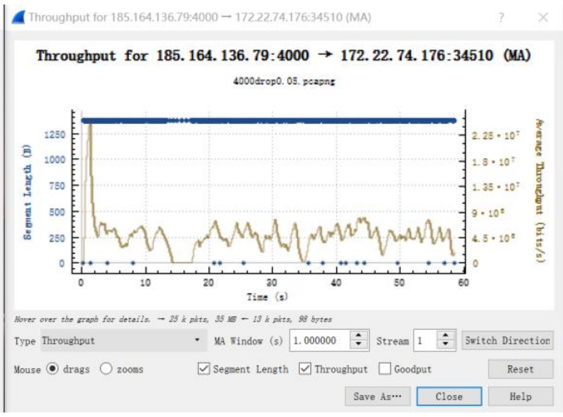
TCP Stream Graph (Time Sequence Stevens):

Time Sequence (Stevens) is a graph of TCP sequence number over time, which helps us see if traffic is moving along without interruption, packet loss or long delays. As we can see from the graphs below, the segment of the curve which is marked in red ellipse represents the TCP segment appeared; and the segment which is marked in yellow ellipse represents the duplicate ACK occurred and indicates time when there is no throughput. Under the same scale, when we observed Port 4000, it is almost a straight line with little or less TCP segment, and in Port 3998 which has 5% data loss, we could see a short segment through this process. And in Port 3999(10% loss), we could see the TCP segment becomes larger. And in Port 4001, 4002, and 4003, we could see the frequency of TCP segment and duplicate ACK increased with the increasing of data loss.

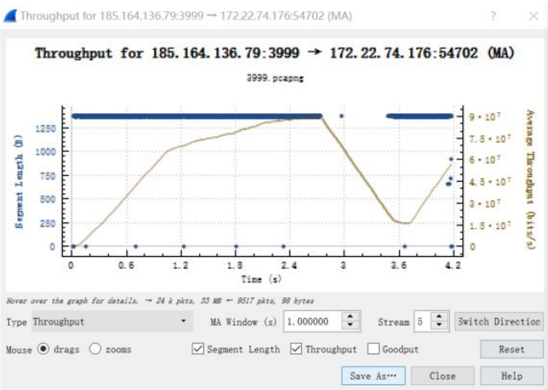


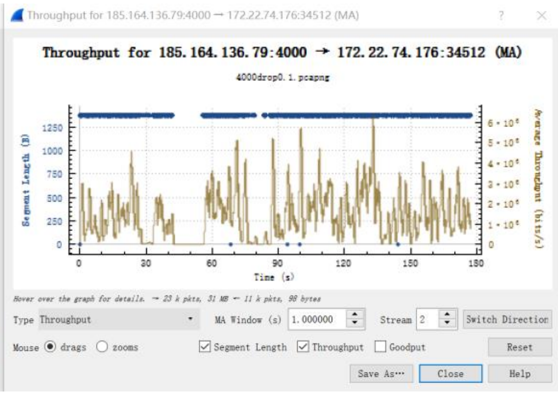
Part 3:

Port 3998 **VS** Drop 5% packets



Port 3999 **VS** Drop 10% packets





As shown in the above figures, when the server loses packets (Port 3998 and 3999) the data throughput curve is smoother, because when the packet lose occurs, the server always waits for the ACK from the client side to transmit the following data. In this case, the server will send packet with larger sequence number to the client, and during a particular period of time, the throughput should be changed relative continually.
In another case, when we drop packets arriving from port (drop 5% packets and 10% packets), the client will send more duplicate ACK to the server and the server will retransmit data more frequently. Therefore, in a particular period of time, the data throughput curve will fluctuate relative fiercely.
And comparing to the graph of drop 5% packets and 10% packets, we could see the “drop 10%” graph fluctuates more frequently than the “drop 5%” graph.