NAMES

Kaushik Nambi Vivekanandan (916935609) Wednesday Discussion

Prabhdeep Kainth (916823863) Monday Discussion

Sreeram Sandrapati (916822738) Monday Discussion

Tom Abraham (917096660) Wednesday Discussion

Question 1:

NO TC

1.

```
osboxes@osboxes:~/Desktop/ECS152A$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.013 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.023 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.026 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.022 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.024 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.022 ms
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.025 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.021 ms
64 bytes from localhost (127.0.0.1): icmp_seq=9 ttl=64 time=0.028 ms
^C
--- localhost ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8184ms
rtt min/avg/max/mdev = 0.013/0.022/0.028/0.004 ms
```

WITH TC

```
osboxes@osboxes:~/Desktop/ECS152A$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=198 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=196 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=213 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=216 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=203 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=208 ms
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=200 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=198 ms
64 bytes from localhost (127.0.0.1): icmp_seq=9 ttl=64 time=201 ms
64 bytes from localhost (127.0.0.1): icmp_seq=10 ttl=64 time=200 ms
64 bytes from localhost (127.0.0.1): icmp_seq=11 ttl=64 time=192 ms
64 bytes from localhost (127.0.0.1): icmp_seq=12 ttl=64 time=195 ms
64 bytes from localhost (127.0.0.1): icmp_seq=13 ttl=64 time=202 ms
64 bytes from localhost (127.0.0.1): icmp_seq=14 ttl=64 time=190 ms
64 bytes from localhost (127.0.0.1): icmp_seq=15 ttl=64 time=189 ms
64 bytes from localhost (127.0.0.1): icmp_seq=16 ttl=64 time=204 ms
64 bytes from localhost (127.0.0.1): icmp_seq=17 ttl=64 time=207 ms
^C
--- localhost ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16024ms
rtt min/avg/max/mdev = 188.878/200.691/215.990/7.207 ms
```

```
osboxes@osboxes:~/Desktop/ECS152A/ECS152A_P2$ ./train.sh
[sudo] password for osboxes:
net.ipv4.tcp_congestion_control = cubic
entering loop
500kbps for 5 seconds
1000kbps for 10 seconds
700kbps for 5 seconds
500kbps for 5 seconds
1000kbps for 10 seconds
```

Since the network bandwidth is changing as specified in train.sh, the speed at which packets are transmitted is also changing. When the bandwidth is low, it takes longer to transmit the packet as

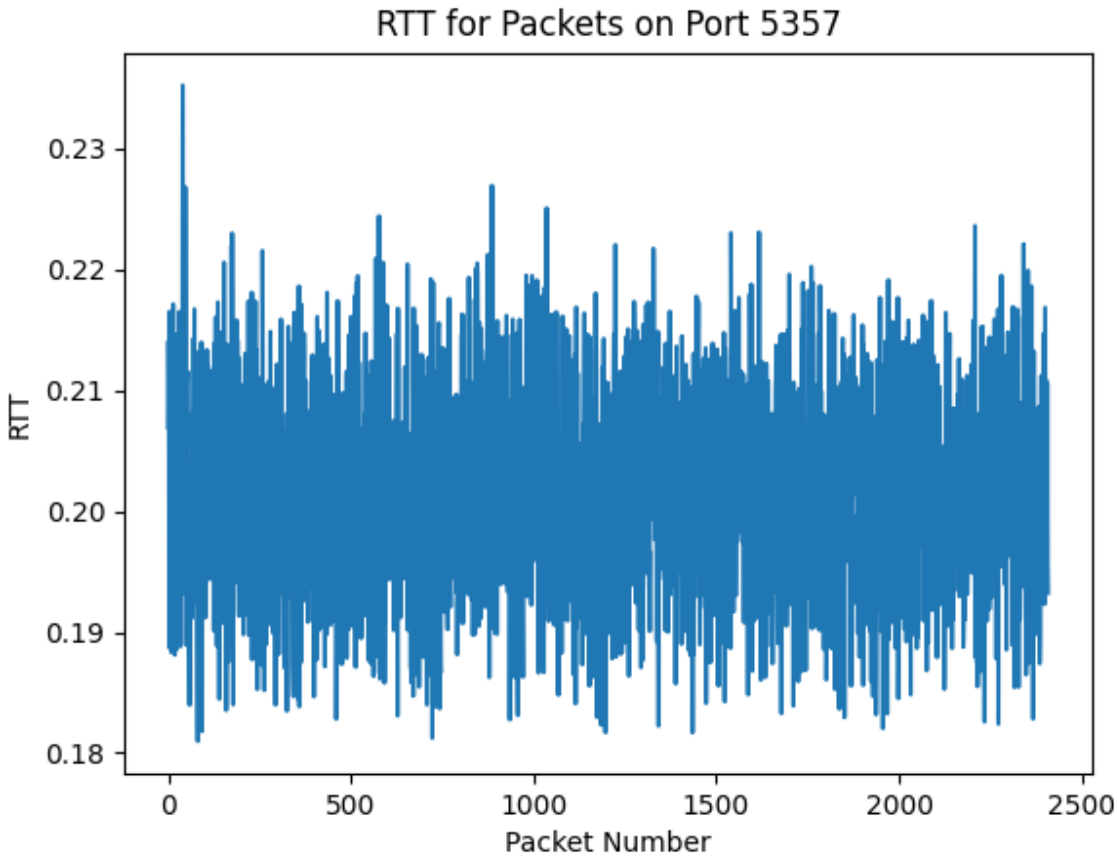
when the bandwidth is high. With no tc, packets are sent and received using the full available bandwidth, hence the much faster transmission.

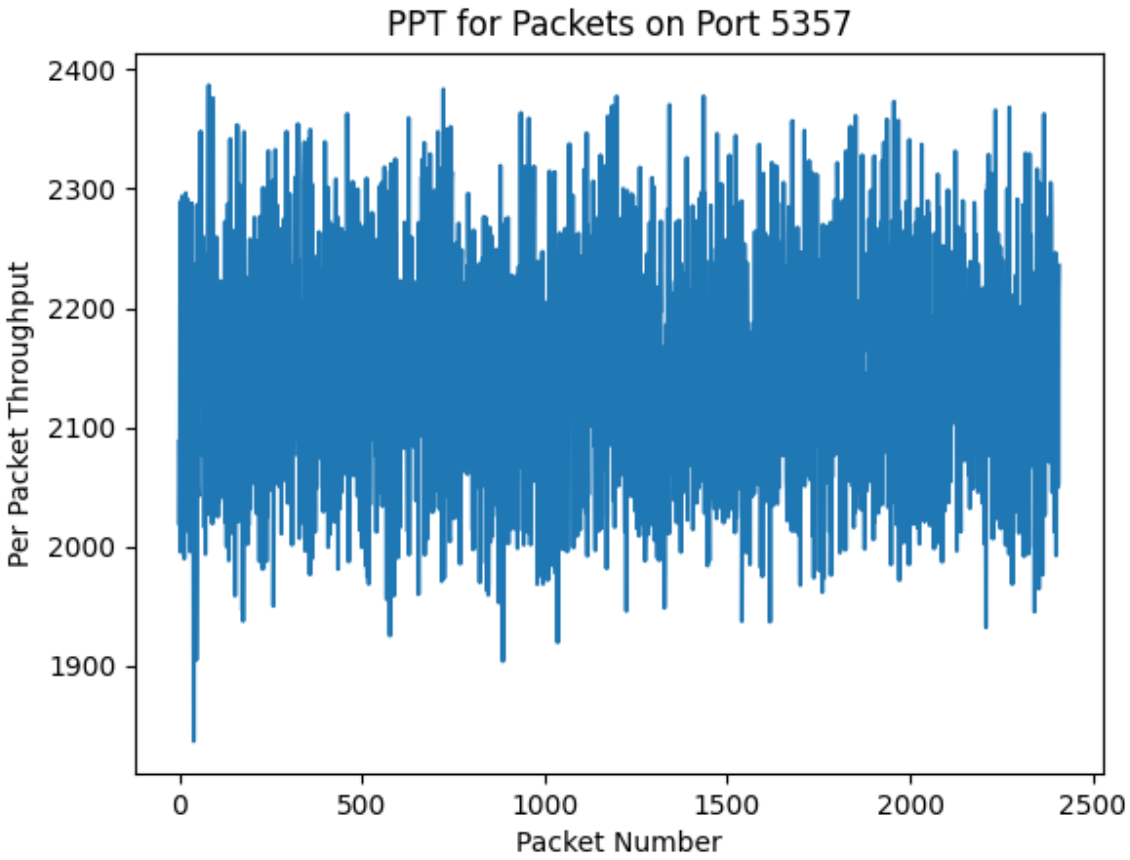
2. For part 1, we lost about 24 packets. In part 2, we lost about 124 packets. And in part3, we lost about 24 packets. These values are varied due to the bandwidth changes that train.sh provides. When the bandwidth is low, more packets are lost. Since the implementation for part 3 accounts for lost packets and dynamically send fewer packets during lower bandwidth times, less packets are lost as when compared to part 2, which sends a constant 5 packets regardless of the bandwidth available. Part 1 also has less packets lost since it only sends one packet at a time; however it takes much longer to transfer the file than the other two parts.

Question 2 (Part 1):

1.
```
Average Delay = <201.81179808699375>
Average Throughput = <95494.91250106468>
Performance = <2.6750336832647834>
Packets lost = 27
```

2.

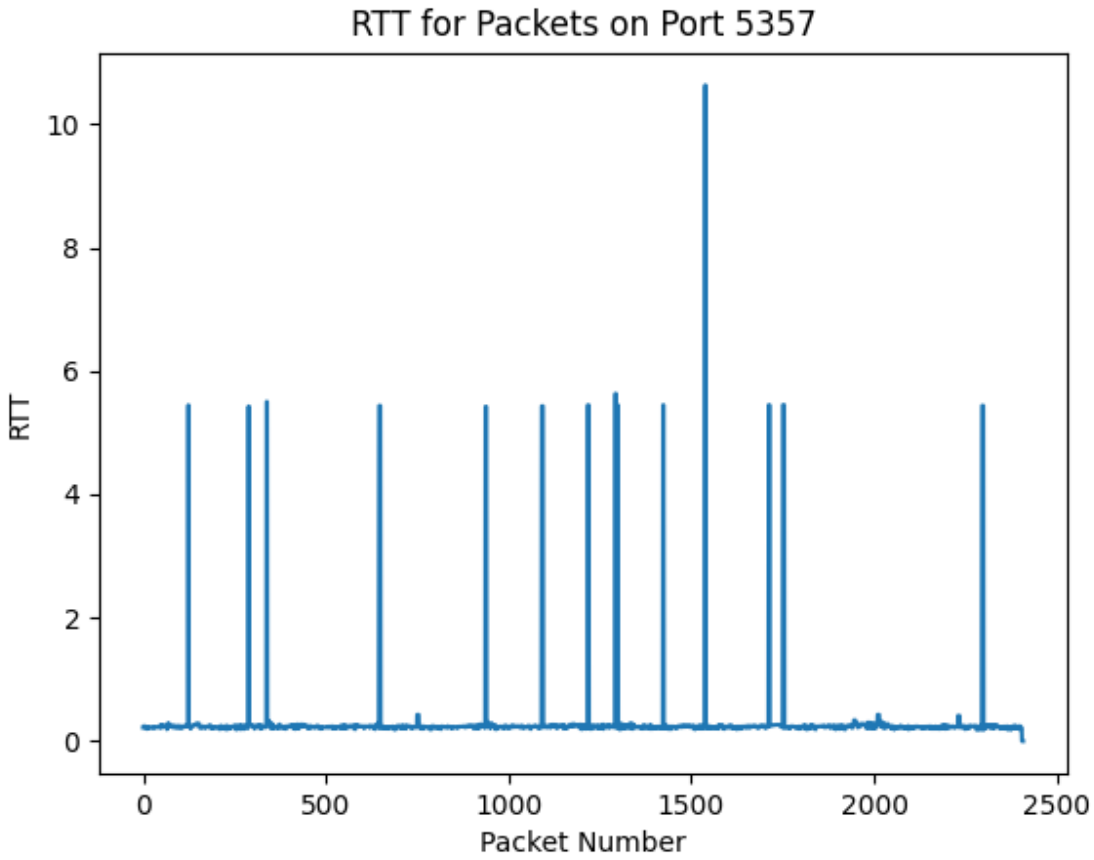

RTT for Packets on Port 5357
RTT
0.23
0.22
0.21
0.20
0.19
0.18
0
500
1000
1500
2000
2500
Packet Number

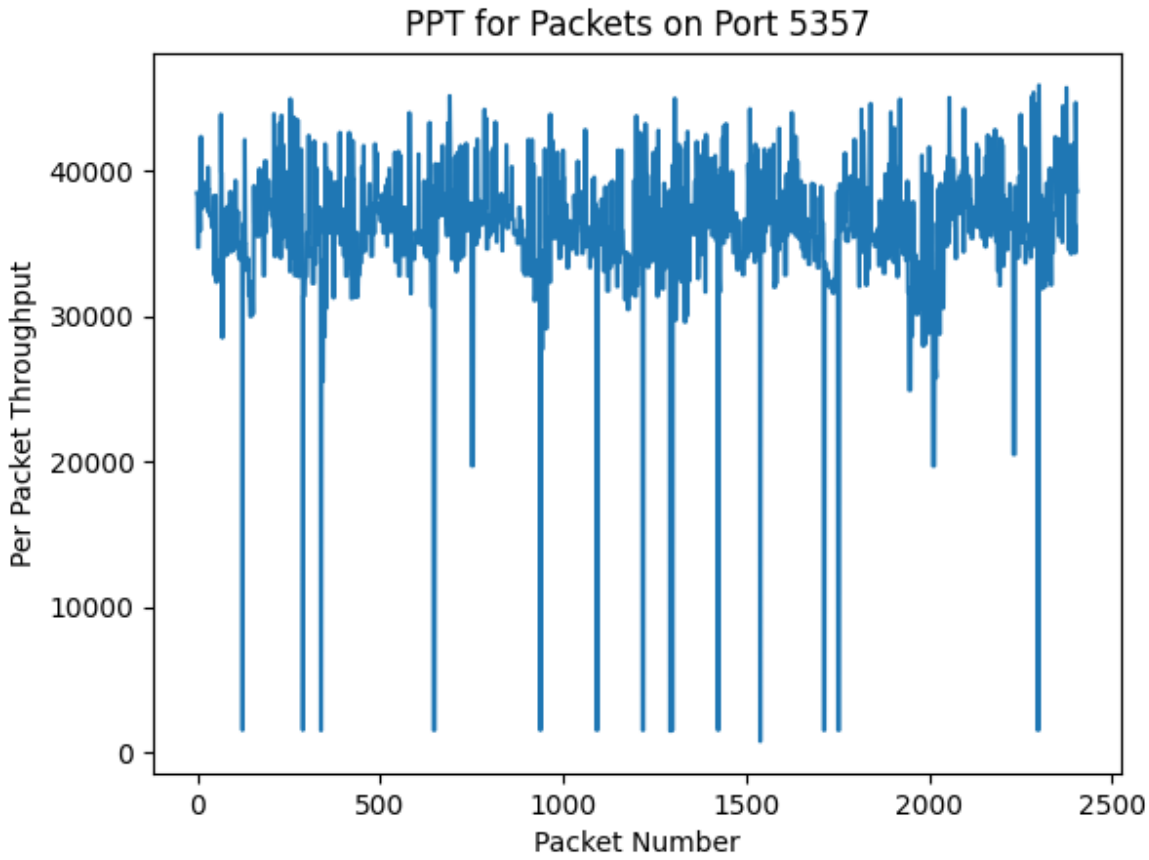
3. 

Question 2 (Part 2):

1.
```
Average Delay = 78.88833206686651
Average Througput= 244221.66745355638
Performance = 3.490771418230292
Packets lost = 121
```

2.

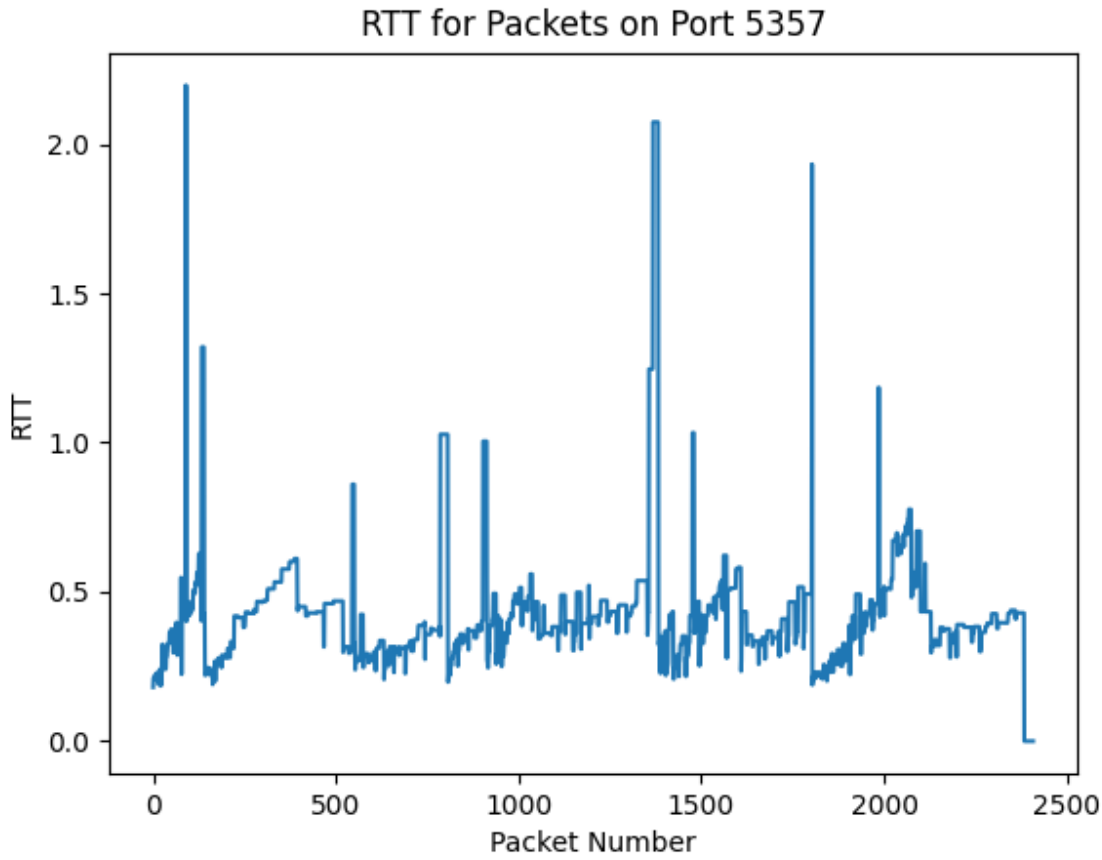

RTT for Packets on Port 5357
RTT
Packet Number
0
2
4
6
8
10
0
500
1000
1500
2000
2500

3. 

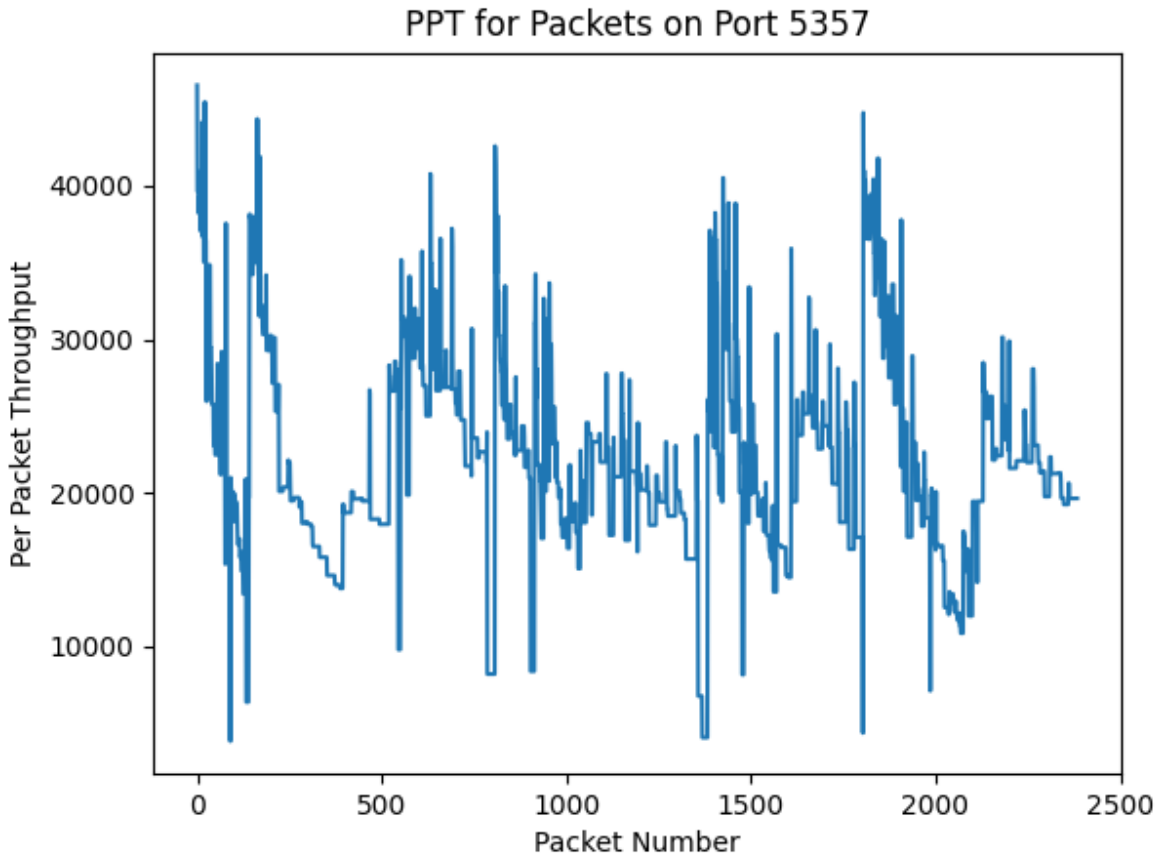
Question 2 (Part 3):

1.
```
Average Delay = 35.40173443880948
Average Througput= 544217.4036218747
Performance = 4.1867478860870575
Packets lost = 24
```

2.


RTT for Packets on Port 5357
RTT
2.0
1.5
1.0
0.5
0.0
0
500
1000
1500
2000
2500
Packet Number

3.

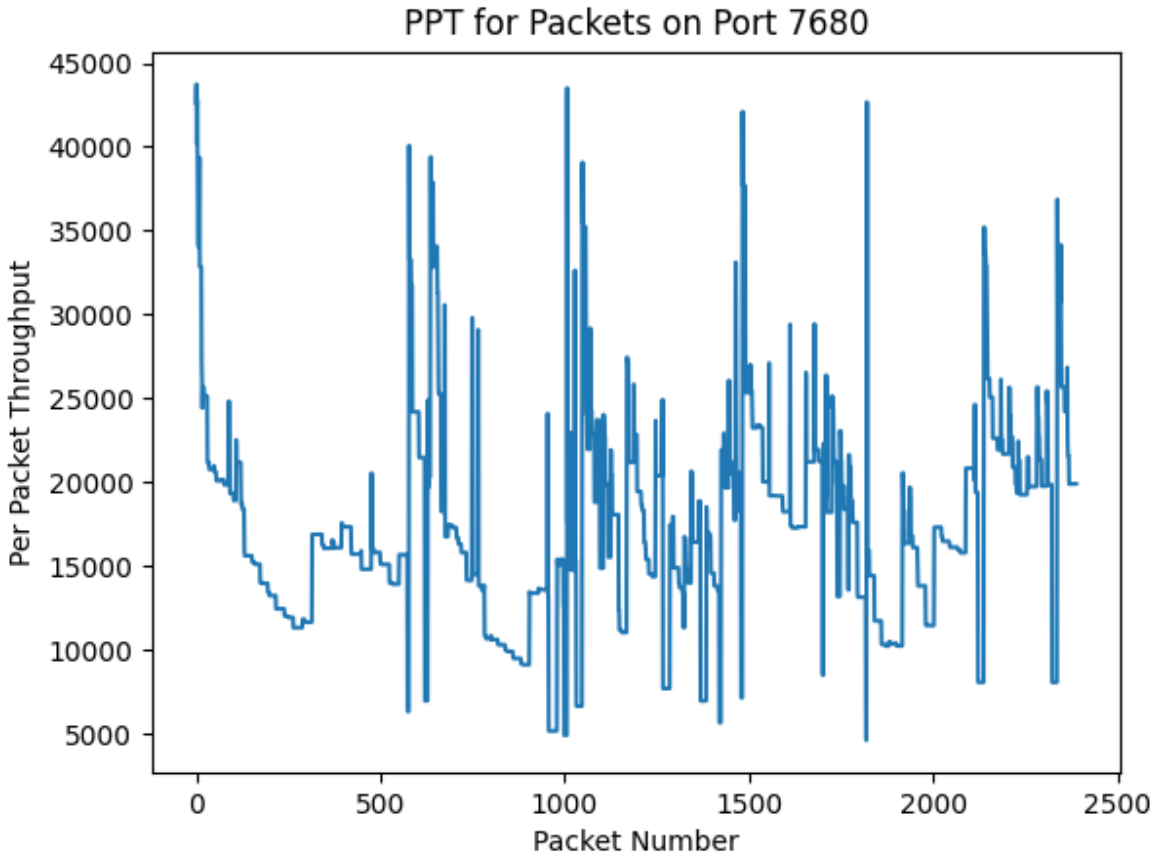


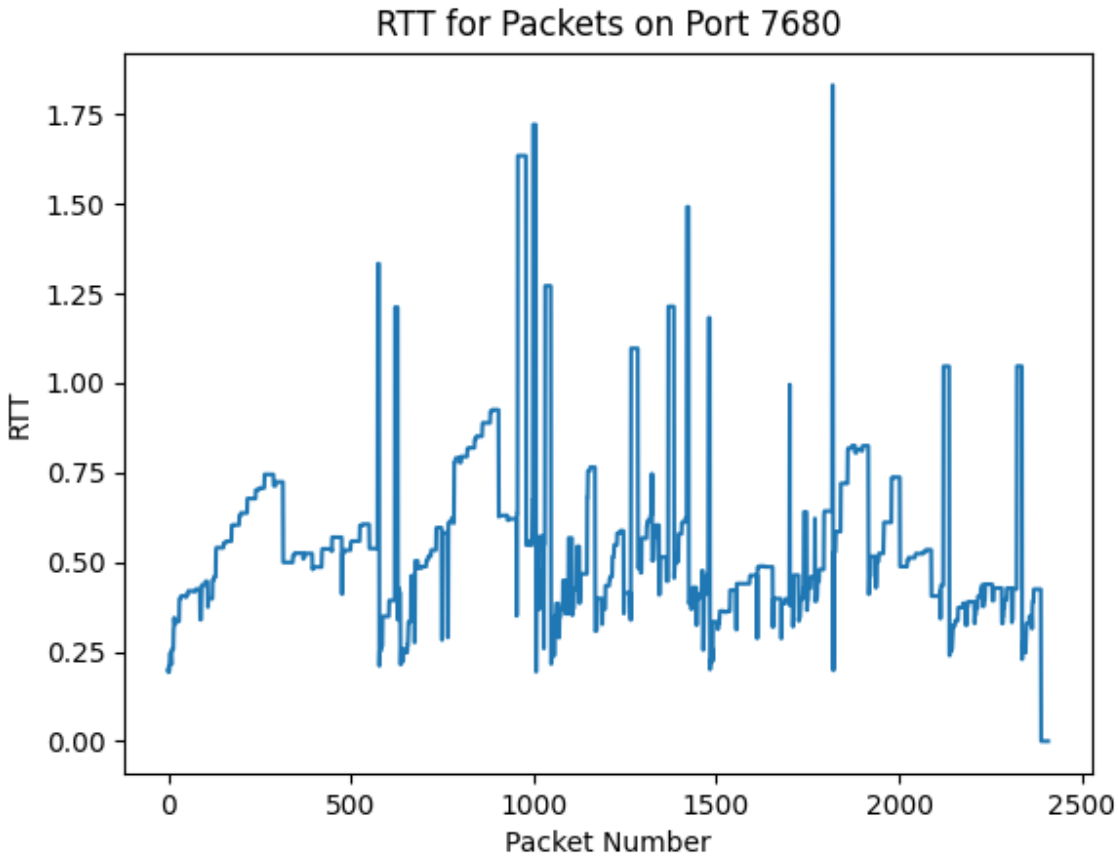
Question 2 (Part 4):

The approach for part 4 was to improve upon the existing implementation of part 3. There were two key areas that were modified to improve the performance: ssthresh and cwnd. We decided to be more aggressive in how much we lower both of these in response to a duplicate acknowledgement. Instead of going all the way down to 1 for the cwnd, we decided to lower it to 30% of what it was so that the bandwidth can be used efficiently faster. The same logic was applied for lowering ssthresh to 80% of the previous value instead of half. Both these changes ensure that the available bandwidth is used up much faster, which in turn speeds up the rate of

packet transfer. The changes are effective, as average delay is lowered, average throughput is higher, and overall performance is higher for our part 4 implementation

```
Average Delay = 29.461728135992153
Average Througput= 653941.2729310758
Performance = 4.346280530687471
Packets lost = 23
```




RTT for Packets on Port 7680
RTT
Packet Number
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
0
500
1000
1500
2000
2500