CSC 573 – INTERNET PROTOCOLS
PROJECT #2 FALL 2021

Aayush Indrapratap Singh, asingh48 **Nitish Pravin Talekar, ntaleka**

1. IMPLEMENTING GO BACK N

Files:

Sender.py	-	Uploads the File to FTP Server
Receiver.py	-	Receives File

Steps to Run:

1. **Run Receiver.py in Terminal 1**

```
python Reciever.py 7735 data/output.txt <PROB_LOSS>
# Gives <SERVER_HOST>
# Waits 5Sec for a connection to establish.
```

2. **Run Sender.py in Terminal 2**

```
python Sender.py <SERVER_HOST> 7735 data/input.txt <N> <MSS>
# Starts sending Data from file to Reciever
```

Example:

Terminal 1: python Receiver.py 7735 data/input.txt 0.01
Terminal 2: python Sender.py Aayushs-MBP.lan 7735 data/input.txt 8 512

Output Expected:

Packets being shown lost in Terminal 1 (Receiver.py)
Timeout being printed in Terminal 2 (Server.py)

2. IMPLEMENTING TASKS 1, 2, 3

Files:

Task_Sender.py	-	Acts as Sender for each Task
Task_Receiver.py	-	Acts as Receiver for each Task

Steps to Run:

1. **Run Task_Receiver.py in Terminal 1**

```
python Task_Reciever.py 7735
# Waits 5Sec for a connection to establish.
```

2. **Run Task_Sender.py in Terminal 2**

```
python Task_Sender.py
# Starts sending Data from file to Receiver
```

These codes implement Task 1 2 3 sequentially in coordination with each other. Output of graphs is obtained in outputs folder.

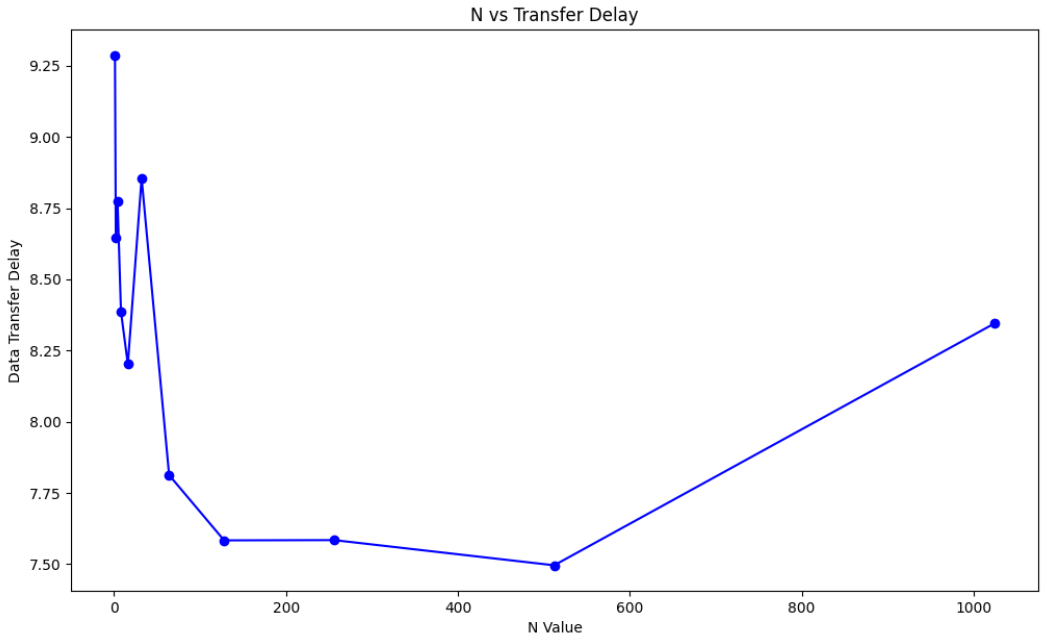
Example:

Terminal 1: python Task_Receiver.py
Terminal 2: python Task_Sender.py

Task 1: Effect of Window Size N

* Each value is in Seconds

N	Transmit 1	Transmit 2	Transmit 3	Transmit 4	Transmit 5	Average
1	9.774863	9.91823506	8.60867	8.78685808	9.34976506	9.28767824
2	8.42079306	9.85144663	9.02333593	7.72389817	8.20120716	8.64413619
4	9.00532818	8.6229949	9.42322731	8.56465983	8.26101422	8.77544489
8	8.38827634	8.44750071	9.12760711	7.92363787	8.04503298	8.386411
16	7.72719002	8.15718818	8.58365297	7.88149095	8.65931177	8.20176678
32	8.20436478	8.59465194	8.38811707	9.25314879	9.83768702	8.85559392
64	7.30492902	8.48442888	7.20666218	8.6877408	7.37965202	7.81268258
128	7.51196218	6.33308721	9.07572818	7.50896573	7.48800802	7.58355026
256	8.00184917	7.71371317	7.98575306	6.6490922	7.5714519	7.5843719
512	7.25316811	6.32377529	7.02749324	8.284132	8.59128499	7.49597073
1024	9.87658882	8.04588079	8.21311522	7.72742105	7.86043906	8.34468899



Observation:
The graph of N vs Average Transmission Delay is graph similar to parabolic curve where we see higher values of delay on extreme values of N.

Reason:
Smaller value of N helps in faster retransmission but small amount of packet transfer at a given time. Thus, this results in high overall Transmission Delay.

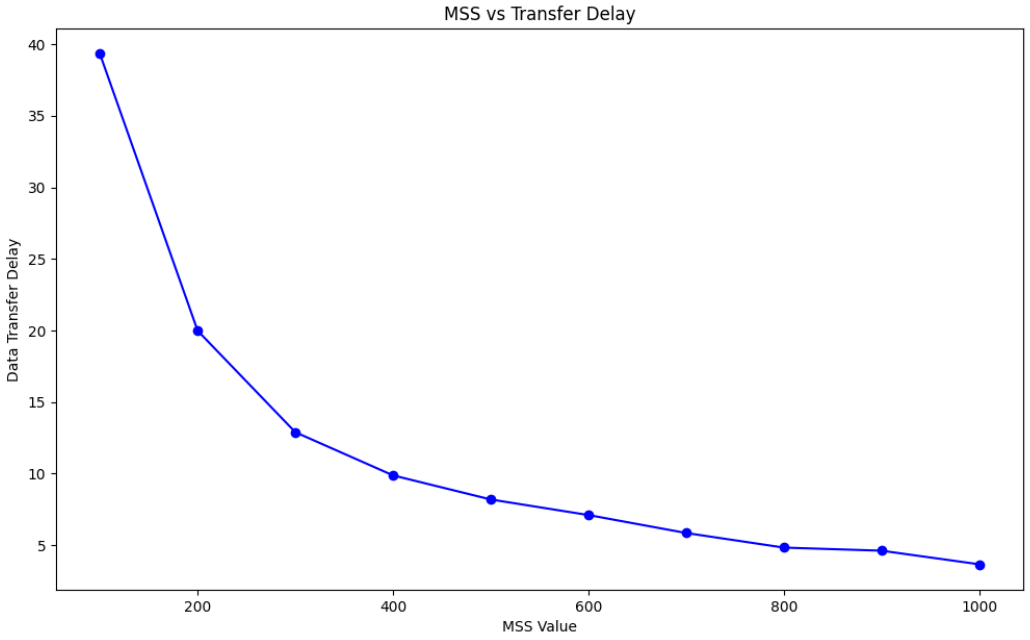
Higher value of N implies higher number of packet transfer at a given time, but it leads to slower retransmission if a packet is dropped. This again leads to higher Transmission Delay.

Values of N which are not too small or big help us keep the transmission delay and packet drops well-balanced and thus we get smaller transmission delays.

Task 2: Effect of MSS

* Each value is in Seconds

MSS	Transmit 1	Transmit 2	Transmit 3	Transmit 4	Transmit 5	Average
100	39.83781099	39.24486613	37.94376802	39.50162196	40.11788392	39.32919021
200	18.20690703	20.47037792	20.80181599	21.11461401	19.33801317	19.98634562
300	13.72897816	11.79656792	12.81402278	13.00163412	13.12758708	12.89375801
400	10.46794605	9.473315239	9.144143105	10.80568099	9.491830111	9.876583099
500	9.191089869	8.276691914	7.428396225	7.828085899	8.241970062	8.193246794
600	6.375527143	6.99002099	8.526383877	6.525587082	7.10576725	7.104657269
700	6.407234907	6.07425189	5.223448038	5.373620987	6.16852808	5.84941678
800	4.938368082	5.048585892	5.146148682	4.640244961	4.372987032	4.82926693
900	4.500380039	4.604979277	5.044509888	4.342978001	4.572220802	4.613013601
1000	3.701091051	3.568786144	4.163603067	3.276177883	3.602416039	3.662414837



Observation:

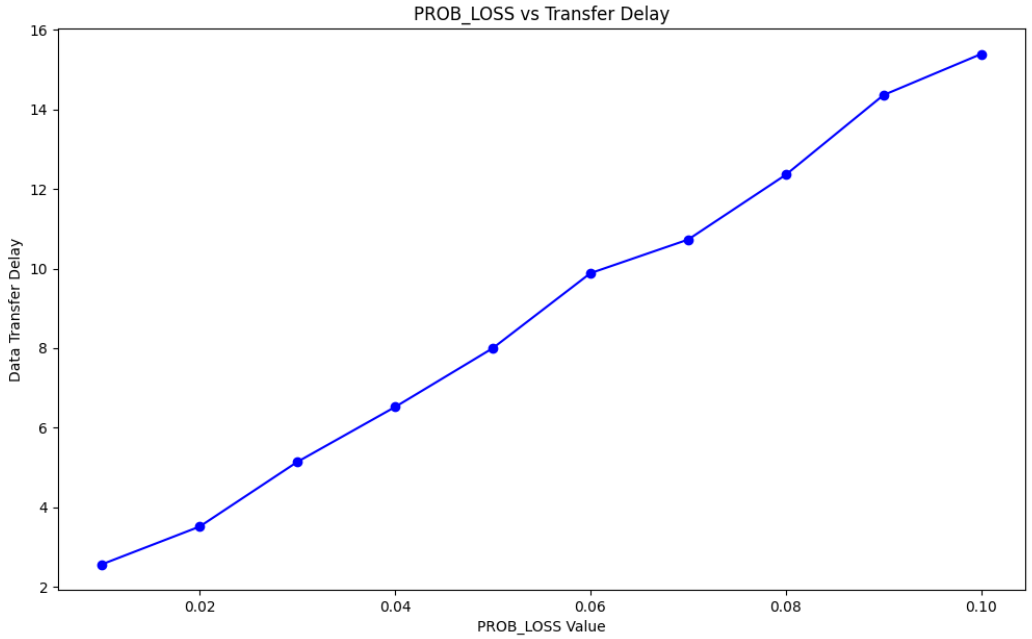
The graph of MSS vs Average Transmission Delay is a decreasing curve where the transmission delay decreases as the MSS value increases

Reason:
As MSS increases, we are able to transfer more data and reduce total number of packets which are to-be sent. Thus, resulting in faster delivery of File.

Task 3: Effect of Loss Probability p

* Each value is in Seconds

PROB_LOSS	Transmit 1	Transmit 2	Transmit 3	Transmit 4	Transmit 5	Average
0.01	2.85155296	2.44306684	2.92854214	2.5183537	2.10535812	2.56937475
0.02	3.52420807	3.7870388	3.25171304	3.63815188	3.38462806	3.51714797
0.03	4.30210614	5.89367914	5.23707795	4.77297783	5.51045489	5.14325919
0.04	6.68921399	5.67826104	5.70769382	7.320297	7.2155931	6.52221179
0.05	7.69587398	7.03787589	7.7772429	8.75358582	8.73395991	7.9997077
0.06	9.56681514	10.5013261	9.61319208	10.0273883	9.71605706	9.88495574
0.07	9.50333691	11.8575311	10.496294	10.6174841	11.173836	10.7296964
0.08	12.1278281	11.7120628	12.024421	12.6416583	13.2873631	12.3586667
0.09	14.700702	15.513921	14.0530691	14.5191629	13.0189979	14.3611706
0.1	15.9669991	14.9324441	15.1792681	14.9334829	15.9520867	15.3928562



Observation:
The graph of Prob_Loss vs Average Transmission Delay is similar to an increasing line where the transmission delay increases as the Probability of packet drop increases.

Reason:
We observe more packets drops when the Prob_Loss value is high and this results in a higher amount of retransmission. Thus, we transmit higher number of packets than required, resulting in higher delay.