Project2: Simple FTP with Selective Repeat ARQ

Qiufeng Yu (qyu4

Mingxuan Shi (mshi4)

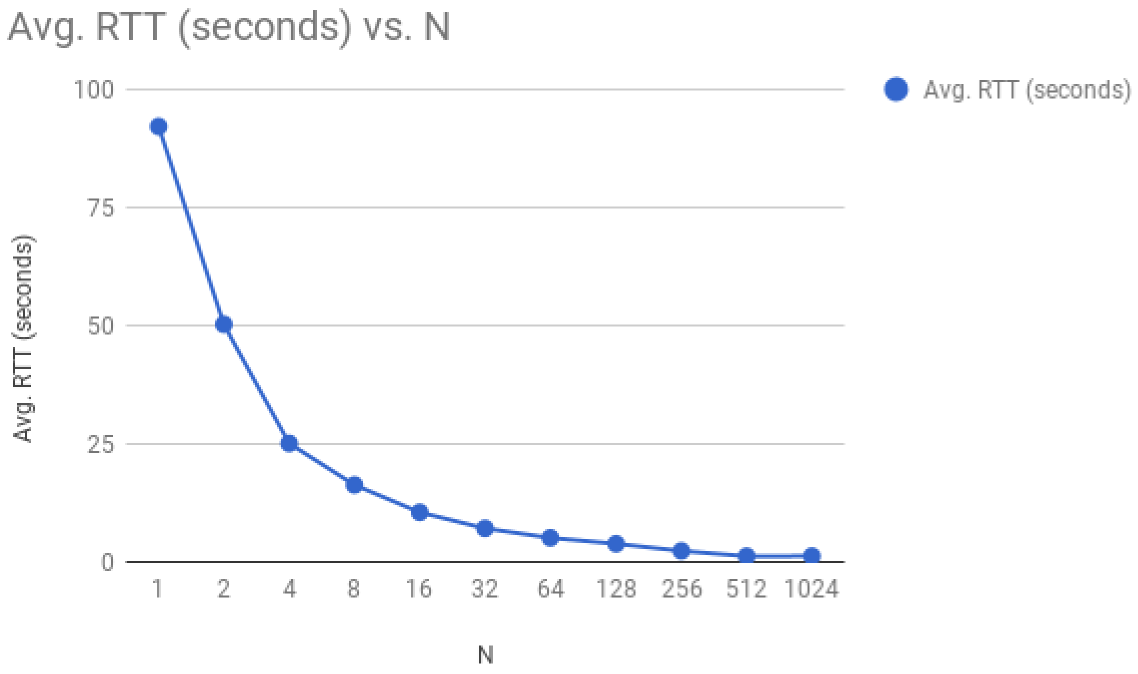
For the extra credit part, we have implemented a simple FTP protocol using UDP to transfer a 1MB file from the client to the server using the selective repeat ARQ scheme. We performed three experiments just like for the Go-Back-N protocol and evaluated the effect of the window size N, maximum segment size MSS, and the probability for packet loss p, and the relationships between those and the average total round trip delay (RTT). For this extra credit project, we performed the experiment on two machines, one local laptop, and the other one is an Ubuntu 16.04 EC2 instance on Amazon AWS. These two machines were separated by 9 hops and the RTT between the two machines was 3.6ms. We measured the RTT five times for each task and put the average RTT into the result.

TASK1: Effect of Window Size N

Fixed variables: MSS = 500, P = 0.05

Results:

N	**Avg. RTT (seconds)**
1	92.154
2	50.254
4	25.125
8	16.326
16	10.526
32	7.125
64	5.123
128	3.866
256	2.352
512	1.236
1024	1.256



Explanation:

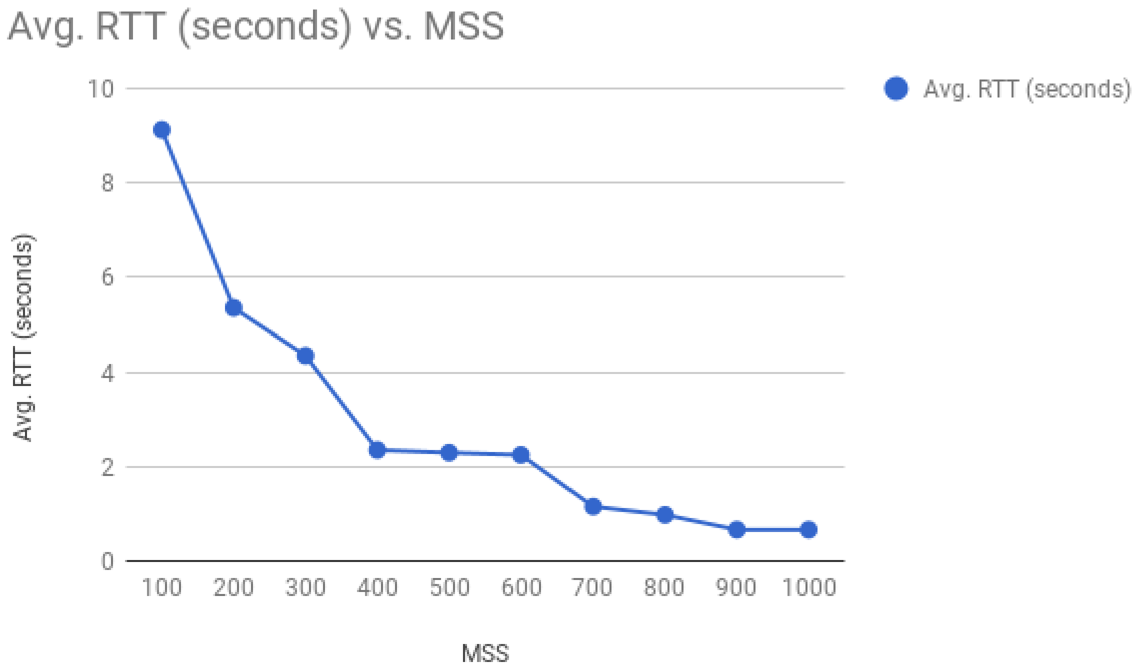
Unlike GBN, in selective ARQ, we observe that when window size N increases, RTT decreases. This is because unlike GBN, with selective repeat ARQ, the receiver (server) only reject a single or some frame, which may be retransmitted alone. On the contrary, with GNB, the sender has to retransmit all packets which haven’t been ACK’d. Therefore, with selective repeat ARQ, as long as most of the packets have been ACK’d, a small amount of retransmissions won’t take much time.

TAKS2: Effect of MSS

Fixed variables: N = 64, p = 0.05

Results:

MSS	Avg. RTT (seconds)
100	9.124
200	5.356
300	4.3464
400	2.352
500	2.295
600	2.2 48
700	1.154
800	0.978
900	0.665
1000	0.664



Explanation:

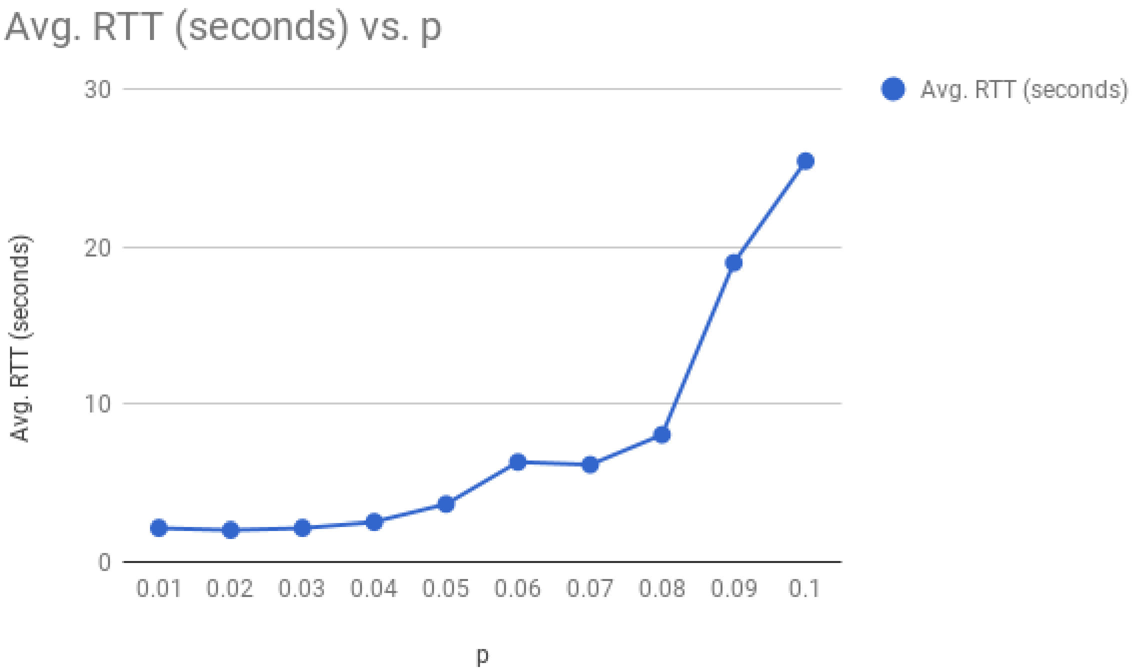
We observe that as max segment size MSS increases, RTT decreases. This behavior is totally expected, as performing this task in selective repeat ARQ is pretty much similar to the task 2 in the GBN report. As we have discussed in the GBN report, when MSS is very small, there are more number of packets need to be sent than situations when MSS is large. Due to timeout or loss of ACK, more number of packets have to be sent, hence leads to large RTT. However, as MSS gets larger, less packets need to be transmitted, which leads to less RTT time.

TAKS3: Effect of Loss Probability p

Fixed variables: N = 64, MSS = 500

Results:

p	**Avg. RTT (seconds)**
0.01	2.150
0.02	2.025
0.03	2.150
0.04	2.526
0.05	3.648
0.06	6.321
0.07	6.152
0.08	8.055
0.09	18.985
0.10	25.450



Explanation:

The graph above shows the trend when p increases from 0.01 to 0.1, RTT increases. This result was expected because even though selective repeat ARQ protocol only retransmits specific packets instead of all packets which haven’t been ACK’d like what GBN does. However, as p increases, more and more packets are lost by the server, it is no different than retransmitting the entire window size, which pretty much is like the Go-Back-N protocol. The only difference is that even with p=0.1 the performance of selective repeat ARQ is still better than GBN, because even though we retransmitting a lot of packets, it is still less the number of packets that we retransmitted when using GBN.