Project 2: Simple FTP with selective repeat ARQ

Aparna Patil - akpatil
Nikunj Shah - nmshah5

In this project we have implemented simple FTP protocol using UDP to send packets and for reliable delivery we have used selective repeat ARQ protocol. We have carried out 3 experiments to evaluate the effect of the window size N, MSS, and packet loss probability p on the total delay for transferring a large file. For the experiments we were separated by 7 hops on our own machine and machine on university network and RTT between the two hosts is approximately 0.1. For transmission we are using file size of 1.1 MB.

Task 1: Effect of window size N

For Task 1 we have considered MSS as 500 bytes, loss probability as p=0.05 and file size greater than 1.1 MB for transfer. We have run the Selective Repeat ARQ protocol to transfer the file selected by varying value of the window size as N = 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024. We have transmitted the file 5 times for each value of N.

Following table represents data obtained for various iterations for value of N:

Task 1					
Server	**Client**	**N**	**MSS**	**Probability**	**Avg. RTT**
EOS Machine	Laptop	1	500	0.05	18.82
		2	500	0.05	7.44
		4	500	0.05	4.01
		8	500	0.05	3.27
		16	500	0.05	3.73
		32	500	0.05	4.19
		64	500	0.05	4.57
		128	500	0.05	5.55
		256	500	0.05	5.00
		512	500	0.05	6.38
		1024	500	0.05	10.58

The graph in figure 1 below plots the average round trip transfer time against N. In this graph, values of window size are plotted on the x-axis and values obtained for average round trip time are plotted on y-axis.

Project 2: Simple FTP with selective repeat ARQ

Aparna Patil - akpatil
Nikunj Shah - nmshah5

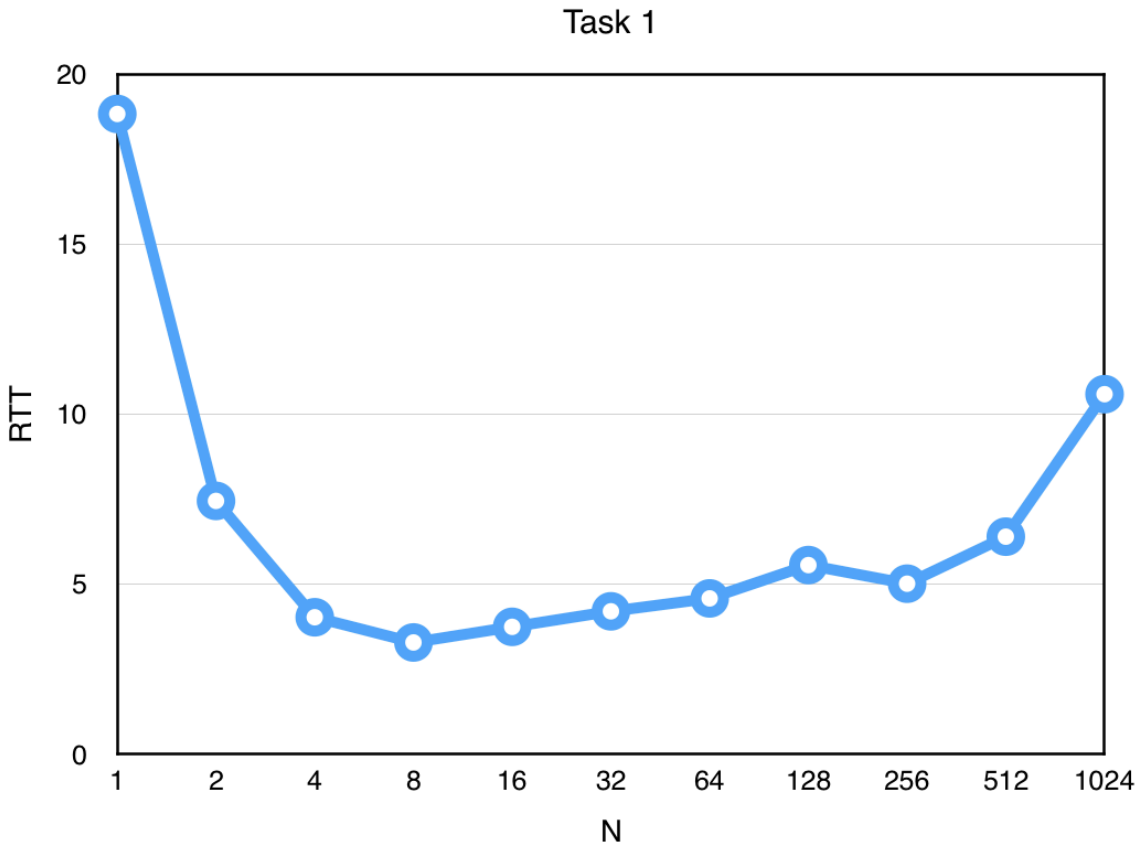

Figure 1: Transfer Time (seconds) vs. Window Size (N)

In this, unnecessary retransmissions of the entire sliding window from sender end are avoided. Hence, it gives better performance as compared to only Go-back-N protocol. As we compare values for both the protocols, it is evident that selective repeat gives better performance. Since we only ask the sender to resend the negatively acknowledged packets, this protocol will perform better. For N=1 it behaves like what GoBackN did but for bigger values significant reduction in transfer time was seen due to less unwanted retransmissions. Hence, ideally best window size should be for values of N in middle range as that will ensure the window at receiver end will slide more often and lead to better performance

Task 2: Effect of MSS

In this experiment, we have considered window size N = 64 and the loss probability p = 0.05. to transfer the same file was run with varying the MSS values from 100 bytes to 1000 bytes in

increments of 100 bytes. For each value of MSS, we have transmitted the file 5 times, and computed the average delay over the five transmissions.

Following table represents values obtained for varying MSS values:

Task 2					
Server	**Client**	**N**	**Probability**	**MSS**	**Avg. RTT**
EOS Machine	Laptop	64	0.05	100	19.7
		64	0.05	200	10.4
		64	0.05	300	6.6
		64	0.05	400	5.9
		64	0.05	500	4.16
		64	0.05	600	3.66
		64	0.05	700	3.19
		64	0.05	800	2.85
		64	0.05	900	2.43
		64	0.05	1000	2.20

The graph in figure 2 below plots the average round trip transfer time against MSS. In this graph, values of MSS are plotted on the x-axis and values obtained for average round trip time are plotted on y-axis.

Project 2: Simple FTP with selective repeat ARQ

Aparna Patil - akpatil
Nikunj Shah - nmshah5

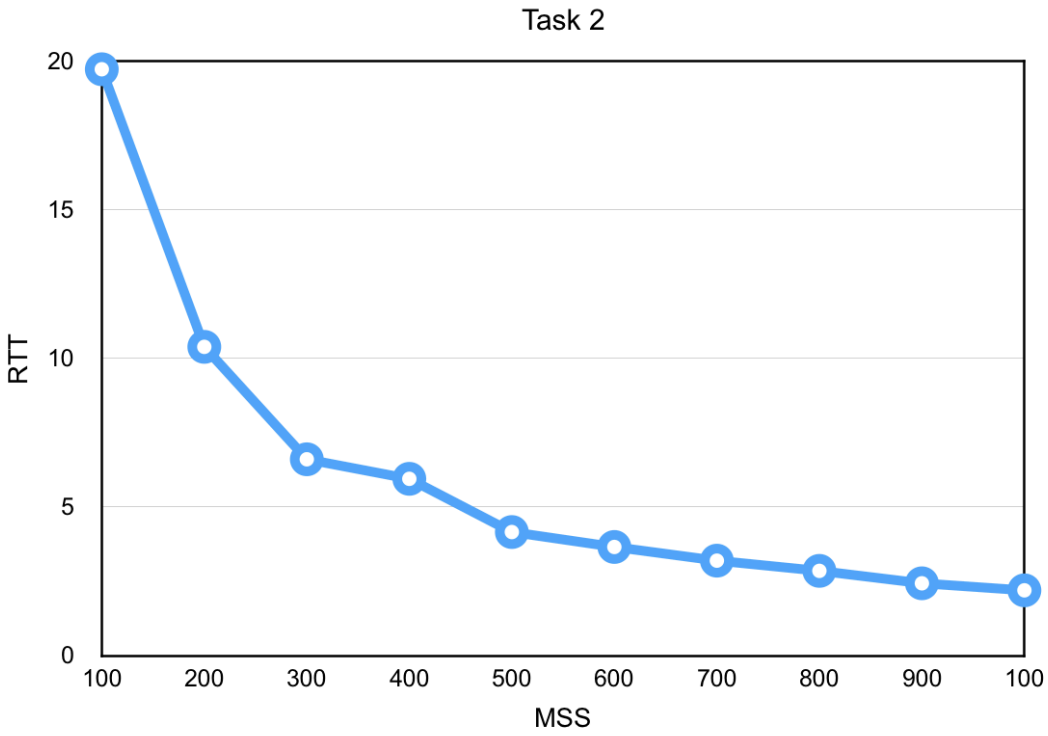

Figure 2: Transfer Time (seconds) vs. Maximum Segment Size (MSS)

In graph plotted above is exponentially decreasing as MSS value increases. This behavior is expected. We know in Go-back-N, as MSS is increased, the average delay decreases exponentially and it is dependent on the bandwidth of the network. Comparing it with Go-Back-N protocol, it is evident that selective repeat ARQ gives better performance than GO-Back-N for the similar experiment even though the graph line loos similar as Selective repeat ensure retransmission only NACK'ed packets which leads to less unwanted retranmissions. As we increase MSS value (considering that the MSS value is supported by network bandwidth), average file transfer time decreases exponentially.

Task 3: Effect of loss probability p

For this experiment, we have considered the MSS as 500 bytes and the window size N = 64. We have executed the program to transfer the same file by varying the loss probability from p = 0.01 to p = 0.10 in increments of 0.01. For each value of p we have transmitted the file 5 times, and computed the average delay over these five transfers.

Following table represents values obtained for varying probability values:

Project 2: Simple FTP with selective repeat ARQ

Aparna Patil - akpatil
Nikunj Shah - nmshah5

Task 3					
Server	**Client**	**N**	**MSS**	**Probability**	**Avg. RTT**
EOS Machine	Laptop	64	500	0.01	2.9
		64	500	0.02	3.5
		64	500	0.03	4.1
		64	500	0.04	4.8
		64	500	0.05	4.8
		64	500	0.06	4.90
		64	500	0.07	5.00
		64	500	0.08	5.1
		64	500	0.09	5.4
		64	500	0.1	5.98

The graph in figure 3 below plots the average round trip transfer time against loss probability. In this graph, values of loss probability are plotted on the x-axis and values obtained for average round trip time are plotted on y-axis.

Project 2: Simple FTP with selective repeat ARQ

Aparna Patil - akpatil
Nikunj Shah - nmshah5

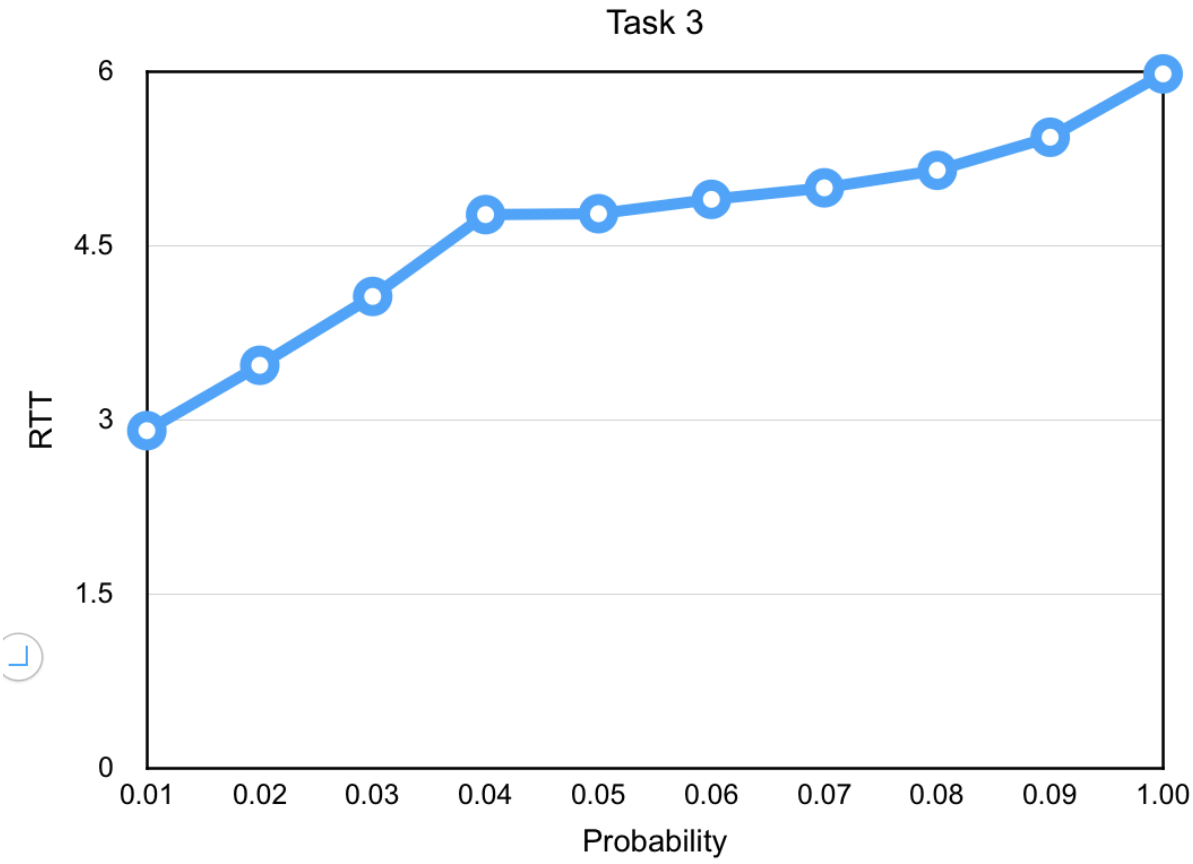

Figure 3: Transfer Time (seconds) vs Probability

Transfer time increases with the increase in loss probability. In Go-back-N protocol as packets are lost at receiver, sender has to resend the packets of the entire window again until an ACK is received. If the packet error rate increases, then sender has to send entire window again hence the time needed to retransmit the lost packets also increases. As compared to that here since we are not sending the entire window but only requested NACK'ed packets, the transfer time even increases linearly but with selective repeat ARQ, it has a lower value as compared to only GoBackN. Here, as seen in the plot, it increases linearly for first few values of probability after than for each increase, transfer time increases slowly.