Lab 9

Summary

In this lab we learned the implementation of the Go-Back-N ARQ protocol. Instead of using the traditional ‘timeout’ function to see when to re-transmit packets, we used ACKs and NAKs to figure out when to re-transmit. Another thing I notcies was that the string length must be at least 6, that is 3 packets with 2 letters in each. The reason is that we set our N to be 3.

Exercise

After running the program 6 times with BER values:

0.001, 0.002, 0.005, 0.01, 0.02 and 0.05

Graphing Average **number of transmissions** per data packet vs **BER values**

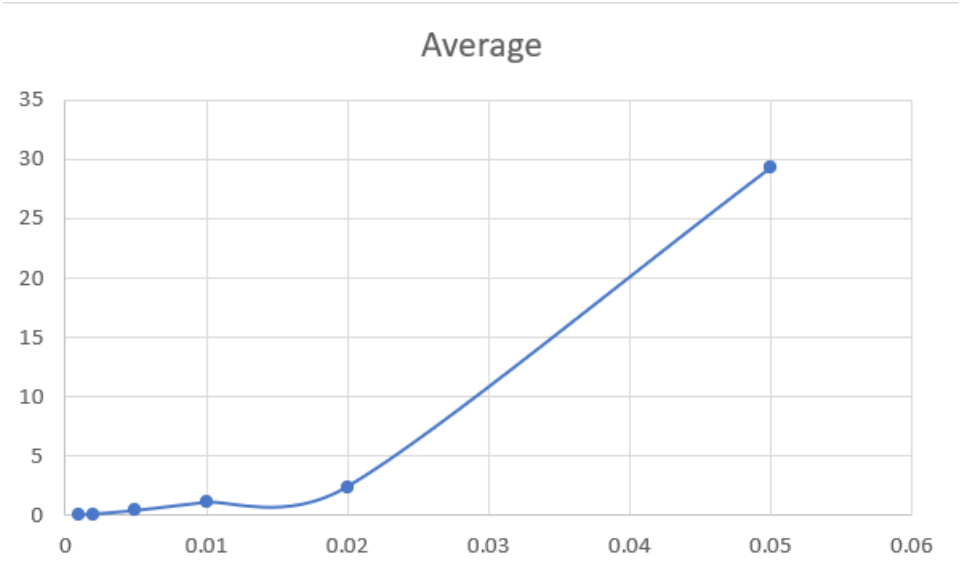
```
bash-4.4$ ./sender 127.0.0.1 50404 0.05
```

```
bash-4.4$ ./receiver 50404 | grep "^Received corrupted packet." | wc -l
381
```

Above shows screenshot of sender and receiver. For BER of 0.05, the number of corrupted packages were too many to count manually, so I used regex.

We see that total number of corrupted packages are 381. Therefore, the average will be: 381/13 = 29.31

BER	Average
0.001	0
0.002	0.077
0.005	0.384
0.01	1.07
0.02	2.38
0.05	29.31



From this graph we observe that the graph is exponential. As the BER increases the average re-transmission also increases. We see that when BER was 0.001, there were 0 re-transmissions totally. When BER was 0.05, there were 381 re-transmissions totally.