Go-back-N Automatic Repeat Request

Using UDP Sockets

Name 1: Harish Pullagurla   Student ID1:   200178872

Name 2: Venkata Surya Subrahmanyam Nukala   Student ID2:   200158956

Go-Back-N ARQ

For the experiments, the Server and Client were configured on two different PCs via the VCL services and the following tasks were carried out

Task 1 :

As the window size increases within a constrained set of values, more number of packets will be in the pipeline. This would eventually result in a reduced RTT as more number of packets are pipelined. The following experiments were carried out in this regard :
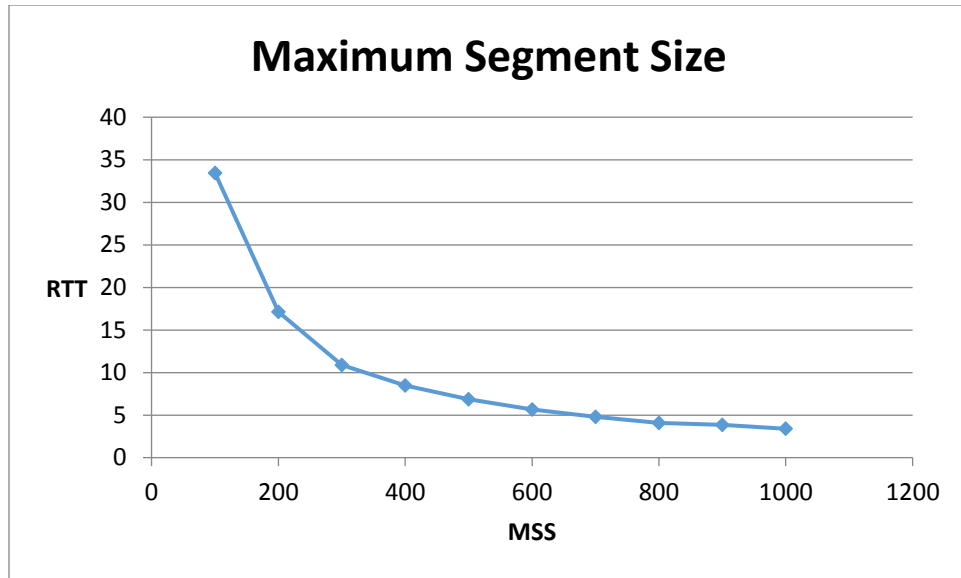
The window size was gradually increased from 1 to 1024 bytes and the time delay corresponding to each of the transition has been observed and documented in the graph below : As it can be observed, as the window size increases, the time delay decreases and gradually reaches a constant.
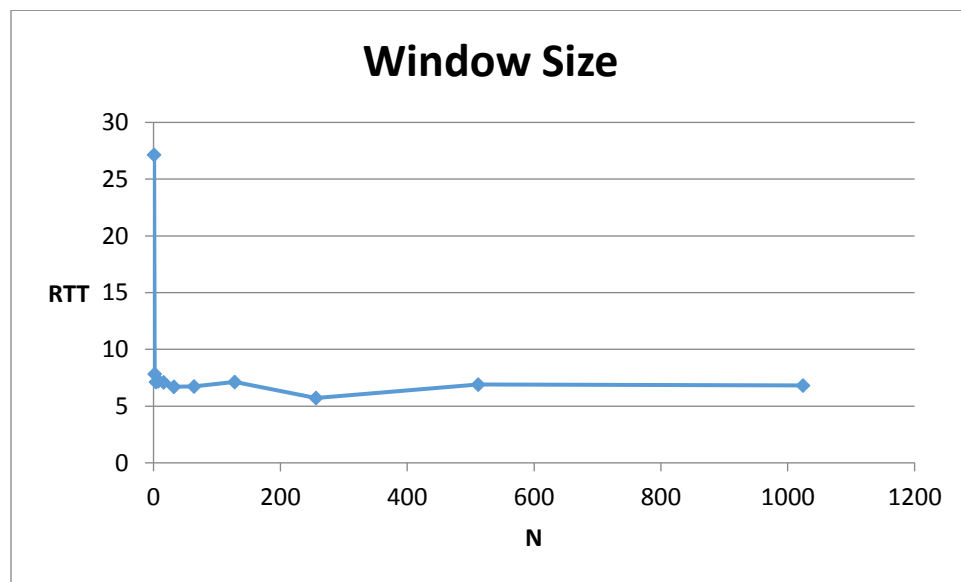
Task 2:

As the MSS increases, the size of the packet to be transmitted also increases and for a fixed amount of data to be transmitted, it results in a reduced total number of packets to be transmitted. The following experiments were carried out in this regard :

The MSS of the packets was gradually increased from 100 bytes to 1000 bytes and the time delay for each transition has been observed and documented in the graph below : As it can be observed, as the maximum segment size increases, the time delay gradually decreases.

Maximum Segment Size

Task 3:

In order to simulate the packet drop scenario in this environment , a random variable has been introduced to determine if a packet is being dropped. A high value of packet loss probability indicates that more number of packets might be dropped and as a result the average time delay increases. The following experiments were carried out in this regard :

The packet loss probability was gradually increased from 0.01 to 0.1 and the time delay for each transition has been observed and documented in the graph below : As it can be observed, as the probability increases, the time delay gradually increases.

Probabilistic loss service