ISTANBUL TECHNICAL UNIVERSITY

Computer Communications

Project- 1

Due Date: 04.01.2021

Hüseyin Alperen YUCAL- 150160035

Esin Ece AYDIN - 150160151

2020-2021 Fall

In this project, it was required to implement a server code that provides reliable data transfer in an unreliable channel by using asymmetrical encryption scheme RSA-1024 and symmetrical scheme AES-246. Go-Back-N protocol is used to meet pipelining in segments transmission.

For this project, we implemented the server.py in accordance with the given client.py source code. Then, we evaluated the performance of the system according to below aspects by assigning different values to the N variable which is the size of the window in Go-back-N and to the errRate variable which is the error rate that an control parameter in packet loss:

- Packet loss,
- Number of packets,
- Throughput,

During the implementation of the project, we gained a fundamental understanding of Transport Layer services such as encapsulating application data into the segments, and managing data, managing the window size at the server side, and using the python UDP socket interface. Besides we gained using multiple protocols(encryption protocols, triple handshake, and go-back-n) for satisfying the reliable transport. We used a timeout of 0.0001 seconds. Therefore the server is implemented to be able to detect timed out packets according to the timeout value.

Task 1: Effect of error rate (errRate)

For this task, we tried to analyze the effect the error rate (errRate) on the performance. Packet loss will take these values: 1%, 5%, 10%, 20%, 40%. (When N is taken as %10)

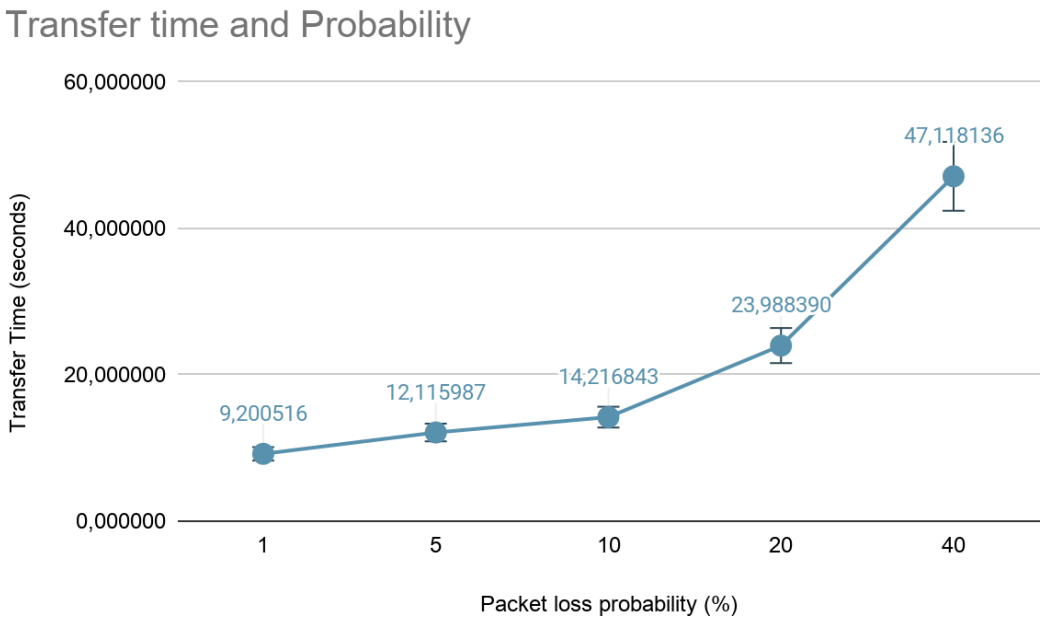

Figure1: Transfer Time (seconds) vs Window Size (N)

The linear relation between likelihood and transfer time is seen in Figure 1. As likelihood increases, so does time for transition. For a protocol like Go-back-N, this is what we expect. If the rate at which an error occurs increases, it is therefore important to increase the time taken to retransmit missing packets.

Task 2: Effect of Window Size N

For this task, we tried to analyze the effect the window size (N) on the performance. We then varied the value of the window size N using these values: 1, 5, 20, 50, 100.
We completed 3 transmissions for each value of and take the average transfer time of the file: (When error rate is taken as %10)

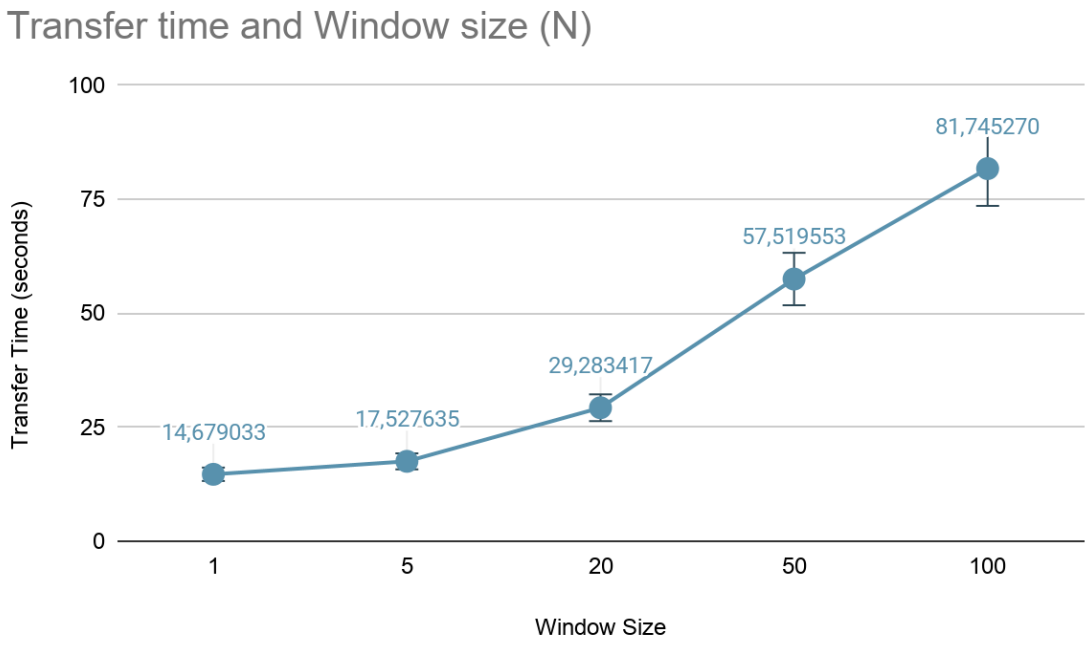

Figure 2: Transfer Time (seconds) vs Window Size (N)

Due to timeouts, a limited window size in the Go-back-N protocol contributes to lost time in retransmitting packets. On the other hand, a wide window size contributes to time-intensive recovery concerns for errors. For instance, if a segment is lost, all of the following segments are also regarded as lost. A window size somewhere in the middle is also optimal. In Figure 2, this can be seen.