Lebanese American University

Department of Computer Science and Mathematics

CSC 450 Computer Networks

PROJECT REPORT

Group Members:

|  |  |  |
| --- | --- | --- |
| Name | ID Number | Email |
| 1. Ali Sheib | 201508275 | [Ali.sheib@lau.edu](mailto:Ali.sheib@lau.edu) |
| 1. Christopher Chalhoub | 201804127 | [Christopher.chalhoub@lau.edu](mailto:Christopher.chalhoub@lau.edu) |
| 1. Farah AL Amine | 201702465 | [Farah.alamine@lau.edu](mailto:Farah.alamine@lau.edu) |
| 1. Ihab El Arab | 201803760 | [Ihab.alarab@lau.edu](mailto:Ihab.alarab@lau.edu) |
| 1. Daniel Hayek | 201804669 | [Daniel.hayek@lau.edu](mailto:Daniel.hayek@lau.edu) |
| 1. Mira Azhari | 201702961 | [Mira.azhari@lau.edu](mailto:Mira.azhari@lau.edu) |

## Describe Your Approach to Reliable File Transfer Protocol Design and Detail the Steps along the Way.

This project is required to establish a transport protocol that provide reliable datagram service for file transfer. After our course knowledge and some research, the first step was to create the skeleton of the code by building simple transfer method without focusing on the other features. A transfer protocol was set in place including two hosts (a sender and a receiver) no handshake was required, however with a FIN request to signal the end of a transmission.

We assume that the receiver is working beforehand and waiting for a message from the sender. The data exchanged is 256-byte packets that are constructed from the standard 20-byte header and 236 bytes of data. The receiver receives the data on the listening port, writes back to the standard output or to a defined file (stdout, or file) and sends ACKs. As for the sender, firstly it converts the file into bytes, defines the required parameters (socket, buffer writer...) and holds packets in case they need to be retransmitted.

Afterwards, the team focused on the other details such as in-order delivery, error detection, duplicates...For these purposes, we search for the needed concepts and method in order to build them and insert them in the code. The protocol is based on Go-Back-N concept with dynamically changing timeout timers in which the sending cycle keeps on sending a number of frames determined by a window size even without accepting an acknowledgment packet from receiver. Moreover, out-of-order packets are dismissed and include cumulative ACKs, re-transmits when corrupted packets are received... Additionally, the system of a checksum was implemented to ensure no corrupted data was transfer and to eliminate errors.

***Challenges Faced – How Did You Solve Them?***

Problem:

I had an error while receiving data from the listening port where i was not transforming the received data into bytes, crashe occurred and 1 stream of data was being sent without organization or flow control.

Solution:

I did some research and ended up using the java byte class to transform the received data into variable of type byte could be arranged as coded first 20 bytes for the packet's header and the rest are for the data load.

Problem:

I didn’t really know how to implement checksum comparison in java and through coding in general .However, I reached a documentation

on the internet where I could use the java MessageDigest class to do.

The JAVA MessageDigest provides cryptographic hash function which can calculate a message digest from binary data.

Problem:

At first, I face a problem with the environment used, the virtual matching that I had didn’t have netem installed in it.

Solution:

This was solved by installing Ubuntu in order to be able to test our code.

Another main problem we faced is that the project was done purely virtually, which may make things harder, however our teammates are hardworking and helpful and we were able to our work and test using zoom.

## List All Features in Your Design

• The sender accepts data from standard-input (stdin), sending data until an end-of-file (eof) is reached.

• The sender and receiver work together to transfer the data reliably, which can be proved by the checksum.

• The receiver outputs the received data to standard-output (stdout), in order, using a queue, and without errors.

• The sender and receiver print out specified debugging messages to standard-error (stderr).

• The sender and receiver gracefully exit.

• Or code is able to transfer a file regardless of file size or type, with any number of packets dropped, damaged, duplicated, and delayed, and under a variety of different available bitrates and link latencies.

## Provide an Overview of Testing Methods

The method that we in order to test our code is by using an unreliable network which is Linus netem. Netem allows the testing of applications and protocols by imitating the real properties of wide area networks, which are packet delay, loss, duplication, and re- ordering. We tested all of the mentioned properties and our code was able to handle them

***Show Your Test Results***

## Show the Contributions of Each Member of the Team.

Once the group was formed, we divided ourselves into two groups: Ali, Christopher, and Farah were responsible for coding and implementing the receiver class while Mira, Ihab, and Daniel worked on implementing the sender class.

Everyone managed to implement a set of methods according to the part he was working on. Zoom meetings when needed were set up between available group members from both teams to ensure the synchrony between the sender and the receiver and testing was done simultaneously. We shared everything on our whatsapp group, and whenever we figured a method or found one on google, we shared it with other members to use it and build up based on it.