

BLG 337E Project 3

Naci Toygun Görmüş

150210719

1. Introduction

The objective of this project is to design a graphics simulator that emulates the functioning of reliable transport protocols, specifically from rdt1.0 to TCP, based on the definitions provided in the textbook. The simulation will cover various versions of the Reliable Data Transfer (RDT) protocol, namely rdt1.0, rdt2.0, and rdt3.0, along with the TCP protocol.

The simulation will be presented on the screen, divided into four subsections, each showing different aspects of the protocol operation.

2. Implementation

The protocol simulations implemented in this project is based on the definitions from the textbooks. This graphics simulator references state diagram actions and operation sequence diagrams given in the textbook. For each protocol mentioned, different cases like normal transaction, premature timeout, ack loss and packet loss are explained step by step by visualizing sender and receiver states.

- For rdt 1.0, the single case of a normal transaction is simulated and visualized.
- For rdt 2.0, the case of a normal transaction and the case of packet loss (corrupted packet) is simulated and visualized.
- For rdt 3.0, the cases of a normal transaction, the case of a packet loss, the case of a premature timeout and the case of an ACK loss is simulated and visualized.
- For TCP, the cases of a normal transaction, the case of a packet loss, the case of a premature timeout and the case of an ACK loss together with TCP connection establishing and closing is simulated and visualized.

3. Guideline for the Simulation

The usage of this simulation program is explained as follows:

The upper left part of the user interface shows Reliable Data Transfer Service Model. The right upper corner shows corresponding state diagram actions both for sender and receiver. The left bottom corner of the user interface shows operation sequence diagram step by step. The right bottom corner has options to choose the protocol to simulate, a command prompt, a simulate button, a user command input and a button for running the command.

The user first picks the protocol to simulate from the right bottom corner. Then hits the run simulation button. The command prompt displays the command options where each of these options are associated with a different case. By entering one of the options in the user command input and hit the run command button the simulation runs and starts showing each step of that case for the selected protocol.

The commands being run in the command prompt can be seen by scrolling down.

The user stops the simulation to simulate another case. At any step of the simulation, the user can enter stop in the user command input to stop and reset the simulation.

At any step of the TCP simulation, the user can enter close in the user command input to close the TCP connection.

4. Outcomes and Observations

This project aims to create a comprehensive graphics simulator for simulating TCP protocols, covering various versions of the Reliable Data Transfer protocol. The visual representation and command prompt outputs will contribute to a better understanding of the protocol's behavior in different scenarios. As I implemented the protocols for different cases, I have expanded my knowledge on how protocols behave under different circumstances to ensure that a reliable connection has been established for reliably transferring data.

5. Conclusion

In conclusion, I tried to implement the simulation as a python program at the best of my understanding of the project description. I believe that this graphics simulator project can serve as a valuable educational tool and testing ground for networking and protocol design concepts. The analysis of simulation scenarios positions this project as a valuable resource for both educational and practical insights into reliable data transfer protocols.