

BLG337E Project III

Ozan Çetin - 150190021

ISTANBUL TECHNICAL UNIVERSITY

December 4, 2023

Introduction

This report presents the design and implementation of a graphics simulator for simulating the TCP protocol, starting from rdt1.0 to TCP, according to the definitions provided in the textbook. The simulation is divided into four subsections, each representing a different aspect of the protocol.

Tools and Libraries Used

Python

The core programming language for implementing the simulation logic and scripts.

SimPy

A discrete-event simulation library for Python. SimPy is employed to model and simulate the various events and interactions in the TCP protocol. It

facilitates the creation of processes, events, and resources, allowing for an event-driven simulation approach.

Simulation Components

Reliable Data Transfer Model

The simulation begins with the implementation of a Reliable Data Transfer (RDT) service model. Three versions of RDT are simulated: rdt1.0, rdt2.0, and rdt3.0. The corresponding functions for sending and receiving data are implemented for each version, considering potential errors, acknowledgments, and timeouts.

State Diagrams

State diagrams for the sender and receiver are simulated. The diagrams depict the various states of each entity and the events that trigger state transitions. The states include CLOSED, SYN_SENT, and ESTABLISHED for both sender and receiver.

Operation Sequence Diagram

The operation sequence diagram illustrates the sequence of actions between the sender and receiver during a simulated transaction. This includes sending and receiving SYN, SYN-ACK, DATA, ACK, and handling timeouts.

Command Prompt

A command prompt is simulated to illustrate the commands and outputs of each command during the simulation. This provides a textual representation of the actions taken by the sender and receiver.

Simulation Approach

Event-Driven Modelling

SimPy is utilized for discrete-event simulation, allowing the modeling of events that occur at distinct points in time. This aligns with the nature of network protocols, where events such as sending/receiving messages and timeouts are critical.

Processes and Events

Simulation components, such as the sender and receiver processes, are modeled as SimPy processes. Events, such as message arrivals and timeouts, are explicitly triggered within the simulation environment.

Resource Handling

SimPy's resource handling capabilities are leveraged to model the message buffer, where messages are stored before being processed by the sender or receiver.

Simulation Scenarios

Normal Transaction

The simulation begins with a normal transaction, where the sender sends a SYN, the receiver receives the SYN, responds with SYN-ACK, and the sender transitions to the ESTABLISHED state.

Premature ACK

A scenario simulating a premature ACK is presented, where the sender sends data, and the receiver acknowledges the data prematurely. This demonstrates

the robustness of the protocol in handling unexpected events.

Timeout Event

The final scenario involves a timeout event. The sender sends data, but due to a timeout, it resends the data. This scenario tests the timeout and re-transmission mechanism of the protocol.

Conclusion

The implemented simulation provides a comprehensive visualization of the TCP protocol, including its various versions and the handling of different scenarios. The state diagrams, operation sequence diagrams, and command prompt outputs offer insights into the inner workings of the protocol, demonstrating its reliability and error recovery mechanisms.

This simulation serves as an educational tool for understanding the complexities of reliable data transfer in network protocols, offering a hands-on experience in a controlled environment.