Reliable Protocol Report

Ashkan Zahedanaraki
Gagan Bahugun
December 12th, 2023

Purpose

This project aims to develop and implement a reliable communication protocol over UDP. We use a proxy to simulate real-world network issues like packet loss and delays, adding a layer of complexity and realism and preparing the protocol for robust performance in unpredictable network conditions. We also made a GUI for real-time protocol monitoring and recorded the statistics in text files at the end.

Requirements

Task	Status
Read input from the keyboard	Fully implemented
Randomly drop data from the sender(with probability)	Fully implemented
Randomly drop ACK from the receiver(with probability)	Fully implemented
Randomly delay data from the sender(with probability)	Fully implemented
Randomly delay ACK from the receiver(with probability)	Fully implemented
Message is sent from sender to receiver without errors	Fully implemented
Proxy acts like a host between the sender and receiver	Fully implemented
Read and write with UDP socket	Fully implemented
Support IPv4 and IPv6	Fully implemented
The receiver writes to console the data it receives	Fully implemented
The receiver sends ACK back to the sender via proxy	Fully implemented
The sender will resend the packet if no ACK is received within a reasonable time.	Fully implemented
A GUI that graphs the data on the sender, receiver, and proxy	Fully implemented
(bonus) Dynamically changing probability of drop or delay	Fully implemented
(bonus) window-based protocol	Not implemented

Platforms

Reliable Protocol has been tested on:

- macOS 14.1
- Ubuntu 22.04.3
- Fedora 39

Language

- Python

Documents

- Design attached
- Testing attached
- User Guide attached