TCP-Based File Transfer System - Short Report

1. Protocol Design

A simple application-layer protocol is designed on top of TCP to support reliable one-to-one file transfer. TCP guarantees ordered and reliable delivery, but the application must define how metadata and file data are exchanged.

Protocol Steps:

1. The client establishes a TCP connection with the server.

2. The client sends metadata:

- Filename (string)

- Filesize (integer)

3. The server sends an acknowledgment (ACK) after each metadata item.

4. The client sends the file content in fixed-size chunks (4096 bytes).

5. The server writes the received bytes to a file until reaching the advertised filesize.

6. Both sides close the connection.

Protocol Diagram:

CLIENT SERVER

connect ----------> accept

send(filename) ----------> recv → ACK

<---------- send(ACK)

send(filesize) ----------> recv → ACK

<---------- send(ACK)

send(file data in chunks) ----------> write to file

close ----------> close

2. System Organization

The system uses a classic TCP client/server architecture.

Server:

- socket()
- bind()
- listen()
- accept()
- receive metadata
- receive file
- close()

Client:

- socket()
- connect()
- send(metadata)
- send(file chunks)
- close()

System Architecture Diagram:

SERVER

socket()

bind()

listen()

accept() <---------- connect()

receive metadata

receive file

close()

TCP

CLIENT

socket()

connect()

send(filename, size)

send(file chunks)

close()

3. Implementation Summary

Server Implementation:

- Creates TCP socket on port 5001

- Accepts connection

- Receives filename and filesize with ACK sync

- Receives file in chunks

- Saves file as recv_<filename>

Client Implementation:

- Creates socket

- Connects to server

- Sends metadata and waits for ACKs
- Sends file in 4096-byte chunks
- Closes connection