Application Layer Protocol Design

Application Layer Protocol Design

1. Overview

This design document outlines a simple FTP-like protocol for file transfer between a client and a server over a TCP/IP network. The protocol will handle file listing, file upload, file download, and session management using a control channel and a separate data channel for file transfers.

2. Protocol Overview

- **Control Channel:** Maintained for the duration of the session for command and response exchanges.
- **Data Channel:** Established for each file transfer operation and closed after the transfer completes.

3. Message Types

3.1. Command Messages from Client to Server:

- **LS:** Request a list of files in the server directory.
- **GET <filename>:** Request to download a file from the server.
- **PUT <filename>:** Request to upload a file to the server.
- **QUIT:** Request to terminate the session.

3.2. Response Messages from Server to Client:

- **OK <message>:** Acknowledgment of successful operation.
- **ERR <error_message>:** Indication of an error with a descriptive message.

- **DATA <port> <size>:** Details for initiating a data channel, including the port and size of data for transfer.

4. Message Exchange Process

4.1. Session Initiation:

- Client establishes a TCP connection to the server's control port.
- Server acknowledges the connection.

4.2. Command Execution:

- Client sends a command through the control channel.
- Server processes the command:
  - If it requires data transfer (GET, PUT), the server opens a new data socket on an ephemeral port, listens, and sends the port number back to the client along with the expected file size for GET requests.
  - For LS, the server gathers a file list and sends it directly over the control channel.

4.3. Data Transfer Setup (for GET and PUT commands):

- Client connects to the provided data port.
- Server accepts the connection and either sends the file (GET) or receives the file (PUT).
- Data transfer is completed, and the data connection is closed.

4.4. Session Termination:

- Client sends a QUIT command.
- Server closes the control connection after sending an acknowledgment.

Application Layer Protocol Design

5. Data Formats

- All commands and textual data are sent as UTF-8 encoded strings.
- Numeric values (e.g., port numbers, file sizes) are sent as strings and converted by the receiver.
- Files are transferred in binary mode over the data channel.

6. Flow Control and TCP Buffer Management

- Implement a chunked data transfer mechanism to prevent TCP buffer overflow.
- Senders must wait for an acknowledgment for each chunk before sending the next, ensuring receiver buffer availability.

Application Layer Protocol Design

7. Protocol Diagram