Protocol: Client-Server Architecture

**1. Overview**

This document outlines the client-server architecture of the twmailer application, detailing the technologies used, the development strategy employed, and any necessary adaptations made during implementation.

**2. Client-Server Architecture**

The twmailer application adopts a classic client-server architecture. The client, responsible for user interaction, initiates communication with the server, which manages message storage, retrieval, and deletion. The communication is based on a custom protocol where the client sends commands to the server, and the server responds accordingly.

**2.1 Client Responsibilities**

* **User Interaction:** The client facilitates user interaction by providing a command-line interface for sending, listing, reading, and deleting messages.
* **Command Handling:** It interprets user commands, constructs messages based on user input, and sends them to the server.
* **Feedback Display:** The client displays feedback received from the server, acknowledging successful operations or reporting errors.

**2.2 Server Responsibilities**

* **Connection Handling:** The server accepts incoming client connections, creating a new thread for each connected client to handle communication.
* **Command Processing:** It interprets received messages, executes the corresponding commands, and sends feedback to the clients.
* **Message Handling:** The server manages user directories, stores messages in text files, and retrieves or deletes messages based on client requests.

**3. Used Technologies**

**3.1 Common Technologies**

* **Socket Programming:** Both the client and server utilize socket programming to establish communication over the TCP/IP protocol.
* **C++ Language:** The application is implemented in C++, providing a robust and efficient foundation for system-level programming.

**3.2 Client-Specific Technologies**

* **Standard Input/Output:** The client interacts with users through the command line, utilizing standard input and output operations.
* **Thread Handling:** The client handles user input and server communication concurrently using threads.

**3.3 Server-Specific Technologies**

* **File I/O:** Message data is stored in text files on the server's file system, and the server uses file I/O operations for message management.