Practical Work 2: RPC File Transfer System

Dương Đức Anh 23BI1401223BI14012

December 5, 2025

Abstract

This report describes the implementation of a simple Remote Procedure Call (RPC)–based file transfer system using raw C sockets. Instead of using an actual IDL-based RPC framework such as gRPC, we designed a custom application protocol using a fixed-size metadata header that simulates RPC semantics. The project includes a client, a server, service design, protocol structure, and group contributions.

1 RPC Service Design

The server exposes one RPC-style method:

UploadFile(filename, filesize, data)

Because no external serialization libraries are allowed, we built our own fixed-size RPC header:

```c
typedef struct {
    char method[32];
    char filename[256];
    long long filesize;
} Metadata;
```

Listing 1: RPC Metadata Structure

Protocol Steps

1. Client connects and sends fixed-size RPC Metadata.
2. Server validates the method ("UploadFile").
3. Server sends acknowledgment code (200).
4. Client streams file bytes in chunks.
5. Server writes chunks to disk until reaching 'filesize'.
6. Server sends final response code (201 = success).

2 System Organization

Below is the diagram required in the assignment.

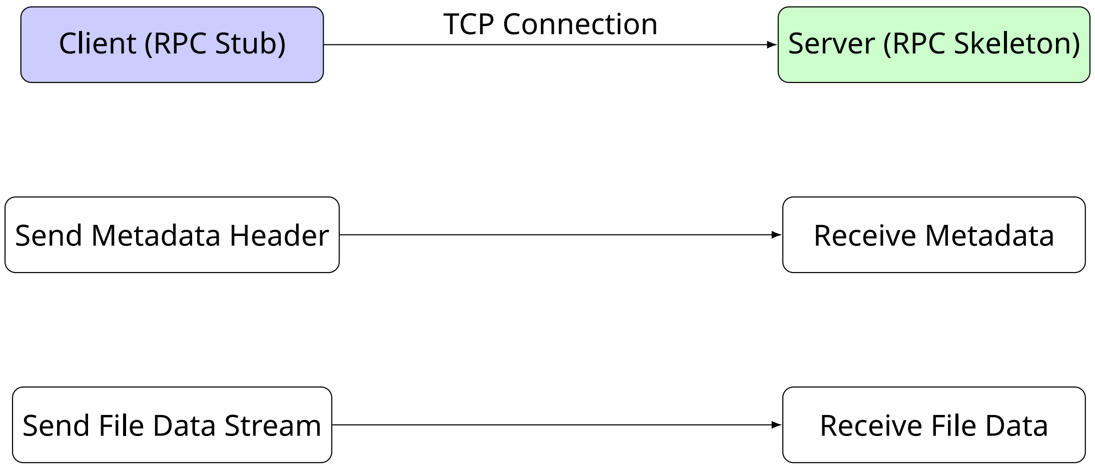

Figure 1: RPC-like File Transfer System Architecture

3 Implementation

3.1 Server Code Snippet

```c
ssize_t recv_all(int sockfd, void *buf, size_t len) {
    size_t total = 0;
    ssize_t n;
    while (total < len) {
        n = recv(sockfd, buf + total, len - total, 0);
        if (n <= 0) return n;
        total += n;
    }
    return total;
}
```

Listing 2: RPC File Transfer Server

```c
while ((bytes = recv(client_fd, buffer, sizeof(buffer), 0)) > 0) {
    write(out_fd, buffer, bytes);
    received += bytes;
}
```

Listing 3: Server Receiving File Data

3.2 Client Code Snippet

```c
Metadata meta;
strcpy(meta.method, "UploadFile");
strcpy(meta.filename, filename);
meta.filesize = filesize;

send(sock, &meta, sizeof(meta), 0);
```

Listing 4: Client Sending Metadata

```c
while ((bytes = read(fd, buffer, sizeof(buffer))) > 0) {
    send(sock, buffer, bytes, 0);
}
```

```
shutdown(sock, SHUT_WR);
```

Listing 5: Client File Upload Loop

4 Conclusion

This practical work demonstrates how RPC semantics can be implemented without a formal RPC framework by combining a structured fixed-size header with stream-based file transfer over TCP. The separation between metadata exchange and data streaming mimics real RPC behavior while remaining simple and lightweight.