UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

Information and Communication Technology Department

DISTRIBUTED SYSTEMS REPORT - FTP Proxy via RPC

Nguyen Ha Trung - 22BI13436
Nguyen Quang Bach - 22BI13053
Do Le Hoang Viet - 22BI13463
Le Quang Tung - 22BI13454
Nguyen Quang Minh - 22BI13299
Pham Le Vu - 22BI13480
Tran Duc Trung - BA12-179

HANOI, DEC 2024

1 Abstract

This system, called ”FTP over RPC,” shows how we can use FTP (File Transfer Protocol) with RPC (Remote Procedure Call). The main goal is to make an FTP proxy that uses RPC to handle all the file transfers. Doing this makes the process safer, easier to manage, and better at fixing errors.

2 Introduction

2.1 What Are FTP and RPC?

FTP is a way to send files between a client (you) and a server. It's old and works fine but has issues like being unsafe and not great for big systems. RPC lets one computer ask another to do something, almost like they're working together on the same machine. It makes networking stuff easier to deal with.

2.2 Why Use FTP with RPC?

Switching FTP to work with RPC helps because:

- It fixes FTP's security issues.
- It works better in complicated systems.
- It's easier to handle errors.

2.3 What's an FTP Proxy?

An FTP proxy is like a middleman. The client asks it to do something, and it talks to the server to make it happen. This makes the whole process smoother and easier to control.

3 Background

3.1 How FTP Works

FTP uses two connections:

- One for sending commands like upload or download.
- One for sending or receiving the actual files.

3.2 How RPC Works

RPC lets one computer call a function on another computer, like:

- The client asks the server to do something.
- The server does it and sends back the result.

3.3 Why FTP Needs Help

FTP isn't safe—it doesn't hide your data or password and struggles in big, modern systems.

4 System Architecture

4.1 How the System Is Set Up

The ”FTP over RPC” system has three parts:

- Client: This sends requests to upload, download, or list files.
- Server: This does the actual work, like saving or sending files.
- Proxy: This connects the client and server using RPC.

4.2 Why Use RPC?

RPC makes everything easier by hiding the complicated network details.

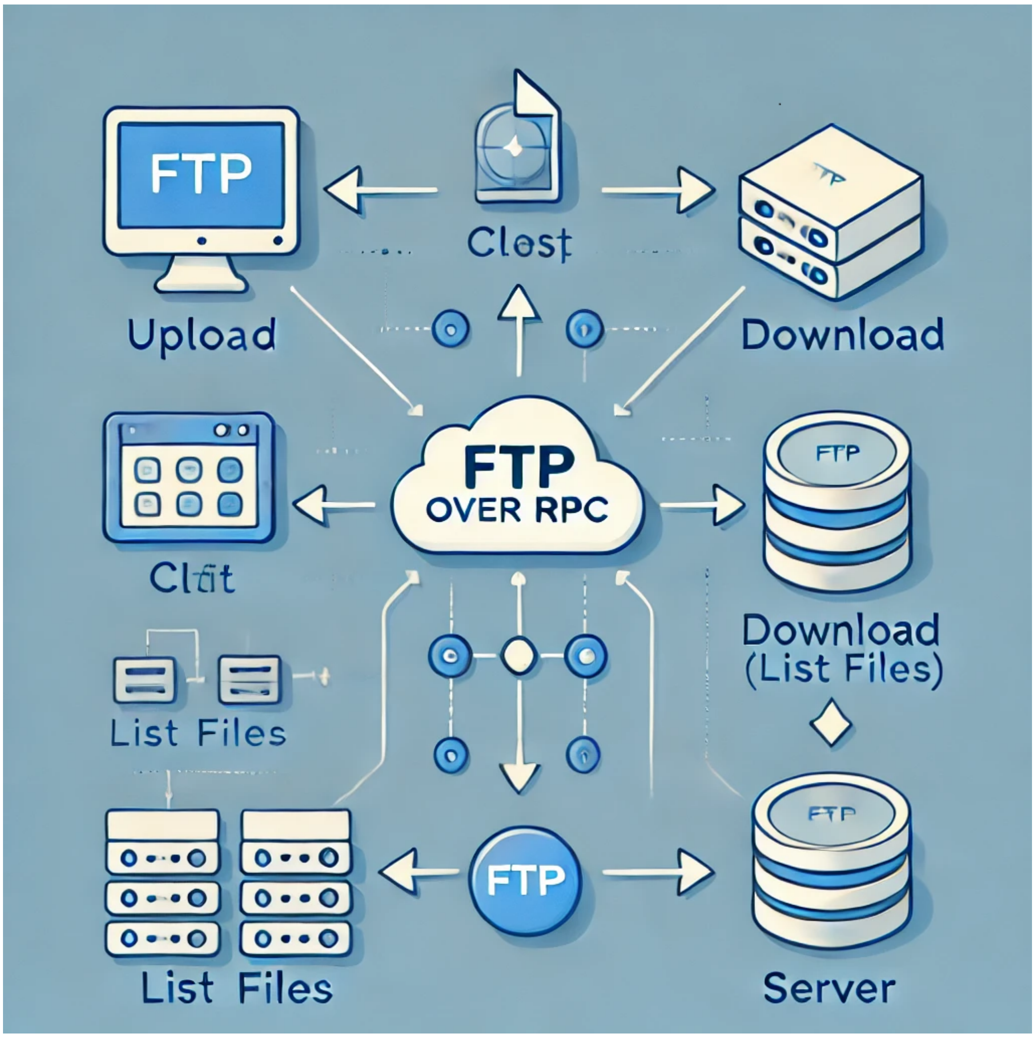


Figure 1: Architecture Diagram of FTP over RPC

5 Design Analysis

5.1 How It Works

The client sends a request, like ”upload this file.” The server does what it’s asked and sends back a reply.

5.2 What Happens When Something Goes Wrong?

If there's a problem (like a missing file), the server sends an error message. If the server doesn't answer, the system stops waiting after a while.

5.3 How It Stays Safe

Data is encrypted while it's being sent. Only verified clients are allowed to send requests.

6 Implementation

6.1 What Tools Are Used?

The code is written in C. A tool called rpcgen is used to create RPC code automatically.

6.2 What Can It Do?

- upload: Send a file to the server.
- download: Get a file from the server.
- list_files: See all the files on the server.

6.3 Example Code: Upload Function

```
int *upload_1_svc(char **file_name, struct svc_req *req) {
    static int result = 0;
    FILE *file = fopen(*file_name, "wb");
    if (!file) {
        result = -1;
        return &result;
    }
    fwrite((*file_name), sizeof(char), strlen(*file_name), file);
    fclose(file);
    result = 0;
    return &result;
}
```

7 Results and Testing

7.1 How Well Does It Work?

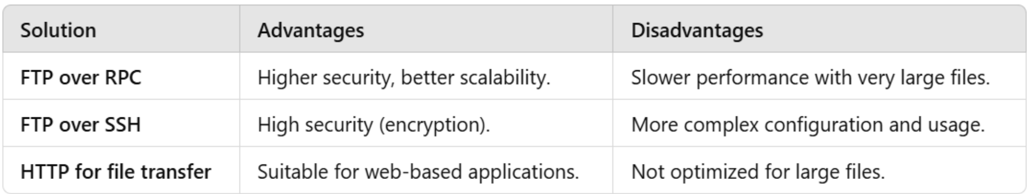
Small files (less than 10MB): Very fast.
Big files (more than 100MB): Slower because it depends on your internet speed.

7.2 What Did We Test?

- File transfers: Uploaded and downloaded files like .txt and .jpg.
- File listings: Showed the correct list of files on the server.
- Errors: Responded properly when files were missing or something went wrong.

8 Comparison with Other Solutions

This system is better than normal FTP because:



Solution	Advantages	Disadvantages
FTP over RPC	Higher security, better scalability.	Slower performance with very large files.
FTP over SSH	High security (encryption).	More complex configuration and usage.
HTTP for file transfer	Suitable for web-based applications.	Not optimized for large files.

Figure 2:

- It's safer.
- It works well in big, complex networks.
- It's better at handling errors.

9 Practical Applications

- File Management Companies: Storing and transferring files between branches.
- Education Sector: Sharing learning materials across remote locations.

10 Conclusion

The "FTP over RPC" system fixes many of FTP's problems by using RPC. It's safer, easier to use, and works better for today's networks. There's still room to improve, like adding stronger encryption and speeding up transfers for big files.

11 References

- Andrew S. Tanenbaum, *Distributed Systems: Principles and Paradigms*
- RFC 959 - File Transfer Protocol Specification
- Linux Documentation on RPC Programming