FTP Server Configuration and Simulation using Cisco Packet Tracer

Harinandanan A M AM.EN.U4ECE22016
Nandakumar Govind AM.EN.U4ECE22029
Pavithra K AM.EN.U4ECE22036
Samuel Sebastian AM.EN.U4ECE22041
Sreerag C R AM.EN.U4ECE22045

***Abstract*—This paper presents the configuration and simulation of a File Transfer Protocol (FTP) server using Cisco Packet Tracer. The project emulates a real-world scenario where file transfer between multiple networks is performed over TCP/IP with user authentication. The FTP setup involved client-server communication across routed networks using dynamic routing (RIP). Through controlled permission levels, different file operations were tested. This simulation provides hands-on experience in managing file transfers, understanding access control, and configuring routing protocols in a virtual environment.**

***Index Terms*—FTP, Cisco Packet Tracer, TCP/IP, RIP Routing, File Transfer, Access Control, Client-Server Model.**

I. Introduction

File Transfer Protocol (FTP) is a widely used application-layer protocol that operates over TCP to enable file transfers between client and server systems. This protocol is instrumental in computer networking education for teaching secure and structured file sharing. In this project, Cisco Packet Tracer was used to simulate a network in which an FTP server facilitates the uploading and downloading of files between users located in different subnetworks. The primary focus is on showcasing routing configurations, client authentication, and real-time file transfer mechanisms with user-specific permissions.

II. Objectives

The main objectives of this project are to set up a centralized FTP server accessible from multiple subnets, configure RIP-based dynamic routing between networks for seamless communication, enable user-based access control using unique FTP credentials and permission levels, demonstrate file upload and download operations through CLI using FTP commands, and analyze file access control through permissions such as Read, Write, Delete, Rename, and List.

III. Methodology

A. Network Topology

The simulation was carried out using two Class C networks. Network A (192.168.1.0/24) contains PC0, PC1, PC2, and Switch1. Network B (192.168.2.0/24) includes PC3, PC4, PC5, and Switch2.

A centralized FTP Server with IP address 100.50.20.3 was positioned behind Router2. Three routers were interconnected to support dynamic routing. Router1 interfaces were connected to 192.168.1.0, 10.0.0.0, and 11.0.0.0. Router2 interfaces were connected to 100.0.0.0, 11.0.0.0, and 12.0.0.0. Router3 interfaces were connected to 192.168.2.0, 11.0.0.0, and 12.0.0.0.

RIP (Routing Information Protocol) was configured on all routers to enable automatic route exchange.

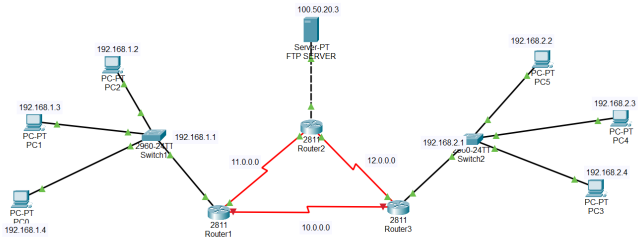

Fig. 1. Complete Network Topology Diagram

B. FTP Server Configuration

The FTP server was configured with a static IP address and multiple user accounts with varying permission levels, as shown below:

TABLE I
User Permissions Table

Username	Password	Permissions
Govind	gvd@123	RW
Hari	Hari@456	RWDNL
Pavithra	pav@123	RWD
Samuel	sam@123	RWD
Sreerag	Sreerag@469	RWDN

The permission key is as follows: R for Read, W for Write, D for Delete, N for Rename, and L for List.

The server was set up to authenticate users based on these credentials and apply corresponding permission constraints during FTP sessions.

IV. Results

A. FTP Command Usage and Demonstration

FTP functionality was tested using command-line interface commands from the client PCs. The most common commands used were: `ftp [IP_ADDRESS]` to connect to

the FTP server, `put [filename]` to upload a file, `get [filename]` to download a file, and `dir` to list directory contents.

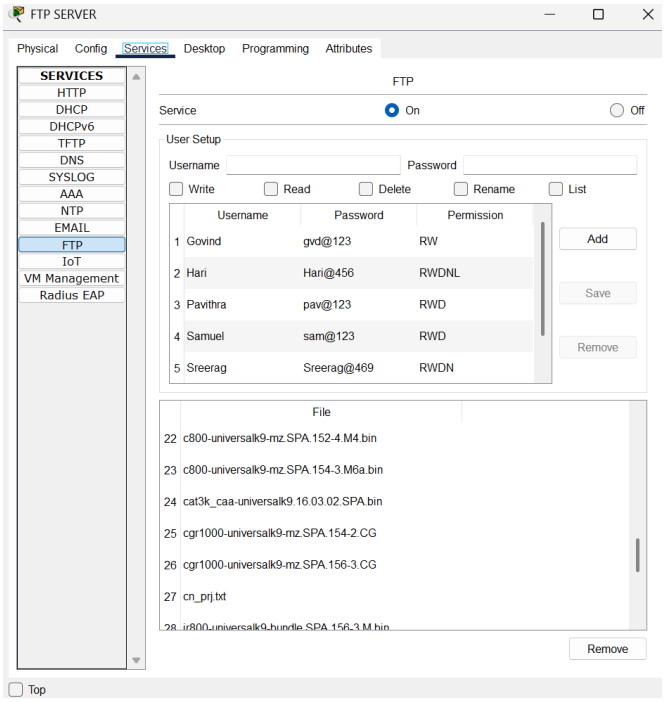

Fig. 2. FTP Server User Configuration

Additionally, after uploading, the file `cn_prj.txt` was observed in the server directory using the `dir` command, confirming successful upload and visibility to authorized users.

1) Upload Demonstration (PC2): User Hari on PC2 (192.168.1.2) established an FTP connection and uploaded a file using the command:

```
ftp 100.50.20.3
Username: Hari
Password: Hari@456
put cn_prj.txt
```

The upload was successful, and the file `cn_prj.txt` was visible on the server.

2) Download Demonstration (PC4): User Sreerag on PC4 (192.168.2.3) retrieved the uploaded file using:

```
ftp 100.50.20.3
Username: Sreerag
Password: Sreerag@469
get cn_prj.txt
```

The download succeeded, verifying the read permission of user Sreerag.

V. Conclusion

The project effectively demonstrated the configuration of an FTP server and simulation of file transfers in a multi-router environment. Routing and communication between sub-networks were established using RIP. User authentication and permission control were successfully applied to FTP operations. This simulation enhanced the understanding of client-server file transfer concepts. Future improvements may include implementing secure FTP variants, integrating firewalls, and expanding protocol use cases.

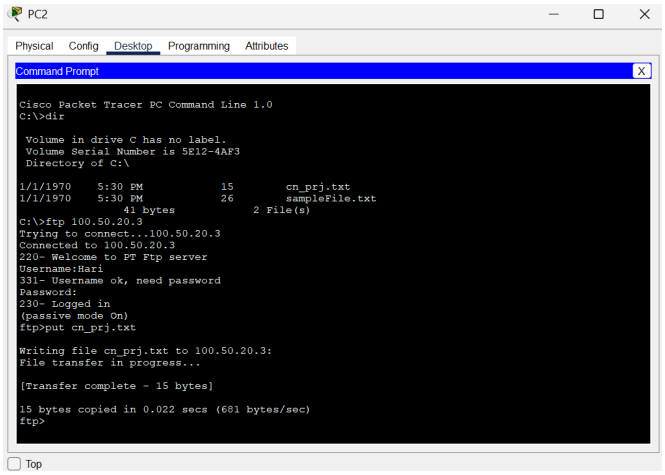

Fig. 3. Uploading File to Server

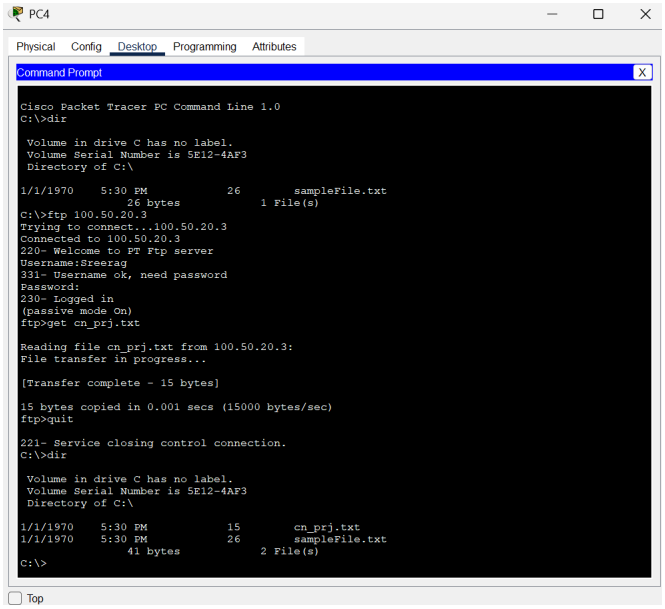

Fig. 4. Downloading File from Server

VI. References

References

[1] W. Richard Stevens, "TCP/IP Illustrated, Volume 1: The Protocols," Addison-Wesley, 1994.

[2] B. A. Forouzan, "Data Communications and Networking," 5th Edition, McGraw-Hill, 2012.

[3] Cisco Networking Academy, "Packet Tracer – Network Simulation Tool," [Online]. Available: https://www.netacad.com/courses/packet-tracer

[4] J. Postel and J. Reynolds, “File Transfer Protocol (FTP),” RFC 959, IETF, Oct. 1985. [Online]. Available: https://www.rfc-editor.org/rfc/rfc959.html
[5] A. S. Tanenbaum and D. Wetherall, “Computer Networks,” 5th Edition, Pearson Education, 2010.