

# File Transfer Protocol in Passive Mode

Mrs. Anto Ramya<sup>1</sup>, Poovizhi R<sup>2</sup> Roopa M<sup>3</sup>

Assistant Professor, <sup>1,2,3</sup>Department of computer Science St .Joseph's College Arts and Science for Women-Hosur,

**Abstract:** File Transfer Protocol (FTP) is a standard network protocol used to send a copy of the file from one host to another, over a TCP/IP-based network, such as the Internet. FTP is built on a client-server architecture. Active and passive are the two modes that FTP can run on. FTP uses two channels between client and server, the command/control channel and the data channel, which are actually separate TCP connections. The command channel is for commands and responses, the data channel is for transferring files. It's a nifty way of sending commands to the server without having to wait for the current data transfer to finish. In active mode, the client establishes the command channel but the server establishes the data channel. In passive mode, the client establishes both channels. In that case, the server tells the client which port should be used for the data channel. Passive mode is generally used in situations where the FTP server is not able to establish the data channel. One of the major reasons for not establishing the data channel is network firewalls. While we may have a firewall rule which allows to open up FTP channels to ftp.microsoft.com, Microsoft's servers may not have the power to open up the data channel back through the firewall. Passive mode solves this by opening up both types of channel from the client side. Hence, we prefer passive mode over active and have tried to use the concept of passive mode in the project.

## I. INTRODUCTION

FTP stands for File Transfer Protocol used to transfer files from one host to another host over a TCP-based network. TCP (Transmission Control Protocol) is a standard that defines how to establish and maintain a network conversation via which application programs can exchange data. TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other. Together, TCP and IP are the basic rules defining the Internet. FTP is built on a client-server architecture and uses separate control and data connections between the client and the server. FTP users may authenticate themselves using a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that hides (encrypts) the username and password, and encrypts the content, FTP is often secured with SSL/TLS ("FTPS"). SSH File Transfer Protocol ("SFTP") is sometimes used instead. The first FTP client applications were command-line applications developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems. Dozens of FTP clients and automation utilities have since been developed for desktops, servers, mobile devices and FTP has been incorporated into hundreds of productivity applications, such as web page editors. To initiate transfer of files with FTP, a program called the 'Client', is used to establish a connection to the remote computer which runs the FTP 'server' software. Once the remote connection is established, the client is free to choose any file to send or to receive a copy of file. To establish a successful connection to a FTP server, the client requires a username and password that has been set by the administrator of that server. One of the most commonly seen questions when dealing with firewalls and other Internet connectivity issues is the difference between active and passive FTP and the best way to support either or both of them.

They are three modules in this project:

### A. Issuing Command

In this module, the server is started first and then the client opens two random unprivileged ports locally ( $N > 1023$  and  $N+1$ ). Client initiates the communication by sending a command PASV and the random data port  $N+1 > 1023$  from port  $N$ . The server acknowledges by sending its random data port  $P > 1023$ .

### B. Sending File

In this module, after request and response to the command port by the client and server, the client initiates the connection from port  $N+1$  to port  $P$  (random port chosen by server for data) on the server to transfer data. The user will select a file to be sent to the server from any location on the client machine, which will be handed over to the TCP/IP protocol for transmission. If required, a preview of the selected file can be seen.

### C. Receiving File

In this module, the file sent from the client's data port (port N+1) is received by the server's random data port P. The user can select the location to store the received file. Finally, an acknowledgement is sent by the server to the client after receiving the file.

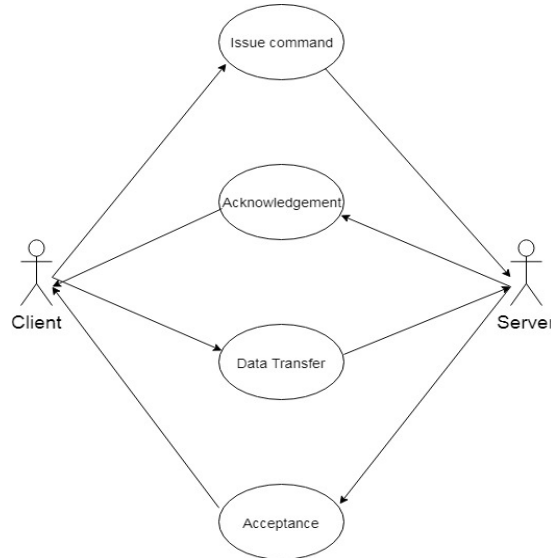


Fig :1FTP Process

### II. CONCLUSION

This project was developed to full fill the file transfer without any traffic and to protect your system from the virus. Since, Active mode in FTP has data port problem, we have used passive mode. In this project, a file is transferred from one directory to another directory using the same IP address as both client and server, it is the simulation for file transfer using FTP (File Transfer Protocol).

### III. FUTURE WORK

The future enhancement of this project is, we will try to implement the file transfer using different IP address for client and server. We can also try to transfer more than one file at a time without any traffic.

### REFERENCES

- [1] Java Computer Networks 2<sup>nd</sup> Edition by Andrew Tanenbaum.
- [2] Computer Networking by James F. Kurose.
- [3] Networking All –In –One for Dummies by Doug lowe
- [4] TCP/IP Unleashed by Karanjit S. Siyan