**BIOINFO Computer Engineering**
ISSN: 2249-3980 & E-ISSN: 2249-3999, Volume 2, Issue 1, 2012, pp.-31-35.
Available online at http://www.bioinfo.in/contents.php?id=322

# REMOTE FILE TRANSFER PROTOCOL BY USING MULTITHREDING

**GAMPAWAR A.D., BHAKTI JAIN, GABANE P. AND NARSWANI A.B.**

Department of Computer Science & Engineering, J.D.I.E.T., Yavatmal, India
*Corresponding Author: Email- adityagampawar10@gmail.com

Received: February 21, 2012; Accepted: March 15, 2012

**Abstract-** First module (FTP) of our framework would provide a user friendly FTP client application which would help even novice user to retrieve the file from remote destination server. Here users fall into two categories:
A: Anonymous Users and
B: Server Account Holder.
In the first category the user can be anyone with no right to update, delete or upload the files on the server thus having limited capabilities.
In case of the user having the account he/she has the privilege of writing, reading, update, delete or upload file on to the server. Following implementations would be involved in complete project –
Creating a FTP Client Application with all functionalities and capabilities for executing various FTP commands and related functions. Configuring DNS Server. This application provides a portal to the user through which the user has to specify the keywords **a**long with the frequency. The FTP Client will allow user to download, upload, update, delete and execute various FTP Commands.

**Keywords-** FTP client, FTP server, DNS, TCP/IP

**Citation:** Gampawar A.D., et al. (2012) Remote File Transfer Protocol by using Multithreding. BIOINFO Computer Engineering, ISSN: 2249-3980 & E-ISSN: 2249-3999, Volume 2, Issue 1, pp.-31-35.

## Introduction

The ftp is denoted as the file transfer protocol. The objectives of FTP are

- To promote sharing of files (computer programs and/or data),
- To encourage indirect or implicit (via programs) use of remote computers,
- To shield a user from variations in file storage systems among hosts, and
- To transfer data reliably and efficiently. FTP, though usable directly by a user at a terminal, is designed mainly for use by programs.

File Transfer Protocol (FTP) is a standard network protocol used to transfer files from one host to another host over a TCP-based network, such as the Internet. FTP is built on a client-server architecture and uses separate control and data connections between the client and server. FTP users may authenticate themselves using a clear-text sign-in protocol but can connect anonymously if the server is configured to allow it.

File Transfer Protocol (FTP) powers one of the most fundamental Internet functions: the transfer of files between computers. Prior to 1995, FTP generated more traffic on the Internet than any other service. Today, Web developers use FTP protocols to upload/ update their web sites and download other information.

A basic understanding about the FTP process and software programs is important for every Web developer. You'll use to post and modify your Web pages at your Web host's server. The FTP (File Transfer Protocol) utility program is commonly used for copying files to and from other computers. These computers may be at the same site or at different sites thousands of miles apart. FTP is a general protocol that works on UNIX systems as well as a variety of other (non-UNIX) systems.

FTP access is restricted, a FTPmail service can be used to circumvent the problem. This service is less flexible than an FTP client, as it is not possible to view directories interactively or to issue modify commands. There can also be problems with large file attachments in the response not getting through mail servers. The service was used when some users' only internet access was via e-mail through gateways such as a BBS or online service. As most internet users these days have ready access to FTP, this procedure is no longer in everyday use.

It allows you to send a file ‘as-is’ from your computer to a student’s UNIX account, an anonymous FTP site where they can access it from home, a computer lab or across the country; it is a relatively safe means of transferring data with little chance for error; and it is quick and easy to use.

### Architecture of FTP

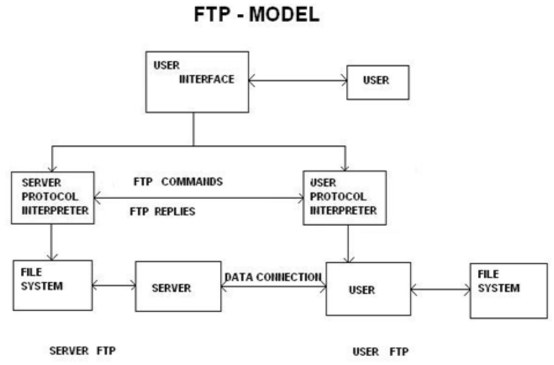


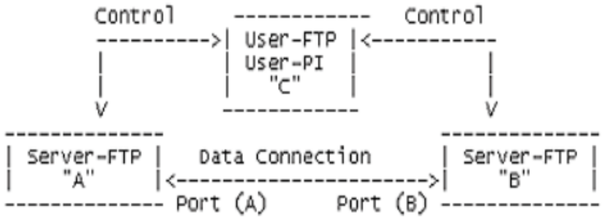
**Fig. 1-** Architecture of Model

NOTES: 1.The data connection may be used in either direction. 2. The data connection need not exist all of the time.

In the model described in Fig1, the user-protocol interpreter initiates the control connection. The control connection follows the Telnet protocol. At the initiation of the user, standard FTP commands are generated by the user-PI and transmitted to the server process via the control connection. (The user may establish a direct control connection to the server-FTP and generate standard FTP commands independently, bypassing the user-FTP process.) Standard replies are sent from the server-PI to the user-PI over the control connection in response to the commands.

The FTP commands specify the parameters for the data connection (data port, transfer mode, representation type, and structure) and the nature of file system operation (store, retrieve, append, delete, etc.). The user- FTP or its designate should "listen" on the specified data port, and the server initiate the data connection and data transfer in accordance with the specified parameters. It should be noted that the data port need not be in the same host that initiates the FTP commands via the control connection, but the user or the user-FTP process must ensure a "listen" on the specified data port. It ought to also be noted that the data connection may be used for simultaneous sending and receiving.

In another situation a user might wish to transfer files between two hosts, neither of which is a local host. The user sets up control connections to the two servers and then arranges for a data connection between them. In this manner, control information is passed to the user-PI but data is transferred between the server data transfer processes. Following is a model of this server-server interaction.



The protocol requires that the control connections be open while data transfer is in progress. It is the responsibility of the user to request the closing of the control connections when finished using the FTP service, while it is the server who takes the action. The server may abort data transfer if the control connections are closed without command.

1) **Control Connection:** The communication path between the USER-PI and SERVER-PI for the exchange of commands and replies. This connection follows the Telnet Protocol. FTP Server Port 21.Connection stays up during the whole session, in which many files may be transferred.
2) **Data connection:** A full duplex connection over which data is transferred, in a specified mode and type. The data transferred may be a part of a file, an entire file or a number of files. The path may be between a server-DTP and a user-DTP or between two server-DTPs. FTP Server Port 20 (for active FTP).
3) **Server-FTP process:** A process or set of processes which perform the function of file transfer in cooperation with a user-FTP process and, possibly, another server. The functions consist of a protocol interpreter (PI) and a data transfer process (DTP).
4) **User-FTP process:** A set of functions including a protocol interpreter, a data transfer process and a user interface which together perform the function of file transfer in cooperation with one or more server-FTP processes. The user interface allows a local language to be used in the command-reply dialogue with the user.
5) **Server-PI:** The server protocol interpreter "listens" on Port for a connection from a user-PI and establishes a control communication connection. It receives standard FTP commands from the user-PI, sends replies, and governs the server-DTP.
6) **User-PI:** The user protocol interpreter initiates the control connection from its port to the server-FTP process, initiates FTP commands, and governs the user-DTP if that process is part of the file transfer.
7) **PI:** The protocol interpreter. The user and server sides of the protocol have distinct roles implemented in a user-PI and a server-PI.

### The Relationship between FTP and Telnet

The FTP uses the Telnet protocol on the control connection. This can be achieved in two ways: first, the user-PI or the server-PI may implement the rules of the Telnet Protocol directly in their own procedures; or, second, the user-PI or the server-PI may make use of the existing Telnet module in the system. Ease of implementaion, sharing code, and modular programming argue for the second approach. Efficiency and independence argue for the first approach. In practice, FTP relies on very little of the Telnet Protocol, so the first approach does not necessarily involve a large amount of code.

### FTP Commands[3]

### Access Control Commands

**User Name (USER)**

The argument field is a Telnet string identifying the user. The user identification is that which is required by the server for access to its file system. This command will normally be the first command transmitted by the user after the control connections are made (some servers may require this). Additional identification information in the form of a password and/or an account command may also be required by some servers. Servers may allow a new USER command to be entered at any point in order to change the access control and/or accounting information. This has the effect of flushing any user, password, and account information already supplied and beginning the login sequence again. All transfer parameters are unchanged and any file transfer in progress is completed under the old access control parameters.

**Password (PASS)**

The argument field is a Telnet string specifying the user's password. This command must be immediately preceded by the user name command, and, for some sites, completes the user's identification for access control. Since password information is quite sensitive, it is desirable in general to "mask" it or suppress type out. It appears that the server has no foolproof way to achieve this. It is therefore the responsibility of the user-FTP process to hide the sensitive password information.

**Chang Working Directory (CWD)**

This command allows the user to work with a different directory or dataset for file storage or retrieval without altering his login or accounting information. Transfer parameters are similarly unchanged. The argument is a pathname specifying a directory or other system dependent file group designator.

**Chang To Parent Directory (CDUP)**

This command is a special case of CWD, and is included to simplify the implementation of programs for transferring directory trees between operating systems having different syntaxes for naming the parent directory.

**Logout (QUIT)**

This command terminates a USER and if file transfer is not in progress, the server closes the control connection.

### Transfer Parameter Commands

**Data port (PORT)**

The argument is a HOST-PORT specification for the data port to be used in data connection. There are defaults for both the user and server data ports, and under normal circumstances this command and its reply are not needed. If this command is used, the argument is the concatenation of a 32-bit internet host address and a 16-bit TCP port address. This address information is broken into 8-bit fields and the value of each field is transmitted as a decimal number (in character string representation). The fields are separated by commas. A port command would be:

PORT h1,h2,h3,h4,p1,p2

where h1 is the high order 8 bits of the internet host address.

**Passive (PASV)**

This command requests the server-DTP to "listen" on a data port (which is not its default data port) and to wait for a connection rather than initiate one upon receipt of a transfer command. The response to this command includes the host and port address this server is listening on.

**Transfer Mode (MODE)**

The argument is a single Telnet character code specifying the data transfer modes described in the Section on Transmission Modes. The following codes are assigned for transfer modes:

S - Stream  
B - Block  
C - Compressed  
The default transfer mode is Stream.

### Transfer Service Commands

**Retrieve (RETR)**

This command causes the server-DTP to transfer a copy of the file, specified in the pathname, to the server or user-DTP at the other end of the data connection. The status and contents of the file at the server site shall be unaffected.

**Store (STOR)**

This command causes the server-DTP to accept the data transferred via the data connection and to store the data as a file at the server site. If the file specified in the pathname exists at the server site, then its contents shall be replaced by the data being transferred. A new file is created at the server site if the file specified in the pathname does not already exist.

**Append (with create) (APPE)**

This command causes the server-DTP to accept the data transferred via the data connection and to store the data in a file at the server site. If the file specified in the pathname exists at the server site, then the data shall be appended to that file; otherwise the file specified in the pathname shall be created at the server site.

**Restart (REST)**

The argument field represents the server marker at which file transfer is to be restarted. This command does not cause file transfer but skips over the file to the specified data checkpoint. This command shall be immediately followed by the appropriate FTP service command which shall cause file transfer to resume.

**Delete (DELE)**

This command causes the file specified in the pathname to be deleted at the server site. If an extra level of protection is desired (such as the query, "Do you really wish to delete?"), it should be provided by the user-FTP process.

**Status (STAT)**

This command shall cause a status response to be sent over the control connection in the form of a reply. The command may be sent during a file transfer (along with the Telnet IP and Synch signals--see the Section on FTP Commands) in which case the server will respond with the status of the operation in progress, or it may be sent between file transfers. In the latter case, the command may have an argument field. If the argument is a pathname, the command is analogous to the "list" command except that data shall be transferred over the control connection. If a partial pathname is given, the server may respond with a list of file names or attributes associated with that specification. If no argument is given, the server should return general status information about the server FTP process. This should include current values of all transfer parameters and the status of connections.

### Help (HELP)

This command shall cause the server to send helpful information regarding its implementation status over the control connection to the user. The command may take an argument (e.g., any command name) and return more specific information as a response. The reply is type 211 or 214. It is suggested that HELP be allowed before entering a USER command. The server may use this reply to specify site-dependent parameters,
e.g., in response to HELP SITE.

### Noop (NOOP)

This command does not affect any parameters or previously entered commands. It specifies no action other than that the server send an OK reply.

### Multi-threading

Downloading many files simultaneously can often increase FTP speeds because each file requires its own connection to be established. For example, you can download a single file at 100 KB/s, but you should be able to achieve speeds of 1 MB/s. If you download 10 files simultaneously, each should be able to reach their 100 KB/s peak, resulting in a total download speed of 1000 KB/s. Note: This method only works with the simultaneous transfer of many files. If you are wanting to increase speed when downloading a single file, read the next section about segmented downloading.

### Segmented Downloading

If you are having poor speeds when downloading from your slot, you may want to try "segmented downloading."

Essentially, what it does is break up a single file into many smaller pieces. For example, let us say that you only get 100 KB/s when downloading a single file from your slot via FTP, but you should be able to achieve speeds of 1 MB/s. If you can break that single file into ten pieces, each requiring their own connection to be established, then you can theoretically achieve ten times the speed.

There are not many FTP clients that support segmented downloading, but a few of those that can are mentioned below.

Windows: CuteFTP Pro, SmartFTP Pro
Mac OS X: Captain
FTP, lftp (via Homebrew or MacPorts)
Linux: lftp

### Multi-threading vs Segmented

A common misconception is that multithreading = segmented downloading. This is not the case. GoFTP is a client that claims to support multi-threading, but in truth doesn’t offer segmented. Segmentation allows an FTP client to split a single large file into multiple parts and use multiple transfers to download those parts simultaneously (i.e - separate segmented parallel FTP connections). These parts are then recombined into a single file upon completion. During the active download you’ll notice multiple (temporary) parts of the same file:

### DNS

The DNS translates Internet domain and host names to IP addresses. DNS automatically converts the names we type in our Web browser address bar to the IP addresses of Web servers hosting those sites. When clients like Web browsers issue requests involving Internet host names, a piece of software called the DNS resolver.

DNS networking is based on the client / server architecture. Web browser functions as a DNS client (also called DNS resolver) and issues requests to Internet provider's DNS servers when navigating between Web sites.

Internet names are the names which is use to refer to hosts on the Internet, such as www.debianhelp.co.uk.

Internet addresses are the numbers which routers use to move traffic across the Internet, such as 211.1.13.115.When a DNS server receives a request not it temporarily transforms from a server to a DNS client. The server automatically passes that request to another DNS server or up to the next higher level in the DNS hierarchy as needed. Eventually the request arrives at a server that has the matching name and IP address in its database and the response flows back through the chain of DNS servers to your client.

The hierarchical Domain Name System, organized into zones, each served by a name server

Administrative responsibility over any zone may be divided by creating additional zones. Authority is said to be delegated for a portion of the old space, usually in the form of sub-domains, to another nameserver and administrative entity. The old zone ceases to be authoritative for the new zone.

### Domain name syntax

The definitive descriptions of the rules for forming domain names appear in RFC 1035, RFC 1123, and RFC 2181. A domain name consists of one or more parts, technically called labels, that are conventionally concatenated, and delimited by dots, such asexample.com.

The right-most label conveys the top-level domain; for example, the domain name www.example.com belongs to the top-level domain com.

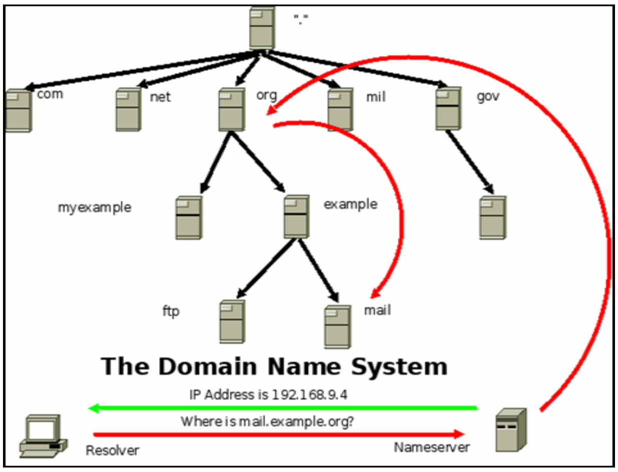
The hierarchy of domains descends from right to left; each label to the left specifies a subdivision, or subdomain of the domain to the right. For example: the label example specifies a subdomain of the com domain, and www is a sub domain ofexample.com. This tree of subdivisions may have up to 127 levels.

Each label may contain up to 63 characters. The full domain name may not exceed a total length of 253 characters in its external dotted-label specification In the internal binary representation of the DNS the maximum length requires 255 octets of storage. In practice, some domain registries may have shorter limits.

DNS names may technically consist of any character representable in an octet. However, the allowed formulation of domain names in the DNS root zone, and most other sub domains, uses a preferred format and character set. The characters allowed in a label are a subset of the ASCII character set, and includes the characters a through z, A through Z, digits 0 through 9, and the hyphen. This rule is known as the LDH rule (letters, digits, hyphen). Domain names are interpreted in case-independent manner. Labels may not start or end with a hyphen.

A hostname is a domain name that has at least one IP address associated. For example, the domain names www.example.com andexample.com are also hostnames, whereas the com domain is not.

### DNS Servers and Home Networking



### TCP/IP

The Internet protocol suite is the set of communications protocols used for the Internet and other similar networks. It is commonly known as TCP/IP from its most important protocols: Transmission Control Protocol (TCP) and Internet Protocol (IP), which were the first networking protocols Telnet is TCP/IP networking application for direct access and the File Transfer Protocol (FTP) for indirect network use. FTP designed to allow the efficient transfer of files between any two devices on a TCP/IP internetwork & automatically takes care of the details of how files are moved, provides a rich command syntax to allow various supporting file operations to be performed (such as navigating the directory structure and deleting files) and operates using the TCP transport service for reliability which contains many of the applications we consider central to TCP/IP networking, such as electronic mail, file transfer and the World Wide Web and also describes interactive and remote application protocols, which are used traditionally to allow a user of one computer to access another, or to permit the real-time exchange of information.

### Acknowledgments

Our thanks to the experts who have contributed towards development of the template.

### References

[1] Feinler, Elizabeth (1982) *Internet Protocol Transition Workbook*.

[2] Postel, Jon, RFC 793, DARPA (1981).

[3] Postel, Jon, and Joyce Reynolds (1983) *RFC* 854.

[4] Reynolds, Joyce, and Jon Postel (1985) *RFC* 943.