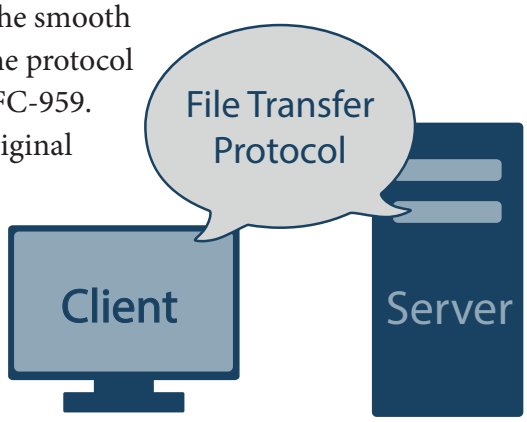
FTP – The File Transfer Protocol

The Internet is composed of file transfers. Every minute of every day, Internet users download files from websites and webmasters upload files, generating content for their websites. The most common method for transferring files is the File Transfer Protocol, or FTP.

Overview

FTP is a widely-accepted Internet Standard, created and made available through the Internet Engineering Task Force (IETF), an open international community of network designers, operators, vendors, and researchers concerned with the evolution of Internet architecture and the smooth operation of the Internet. After FTP's introduction in the early 70s, the protocol was officially approved by the IETF in the mid-80s and designated RFC-959. Although subsequent Internet Drafts have added extensions to the original specification, RFC-959 is still the rulebook that defines FTP.



FTP falls into the standard client/server model. To use FTP, there needs to exist both an FTP client program and an FTP server program. The FTP server will store or house the files accessed during file transfer, and the FTP client will connect to the FTP server and send files to, or retrieve files from, the server.

Client-Server Conversation Structure

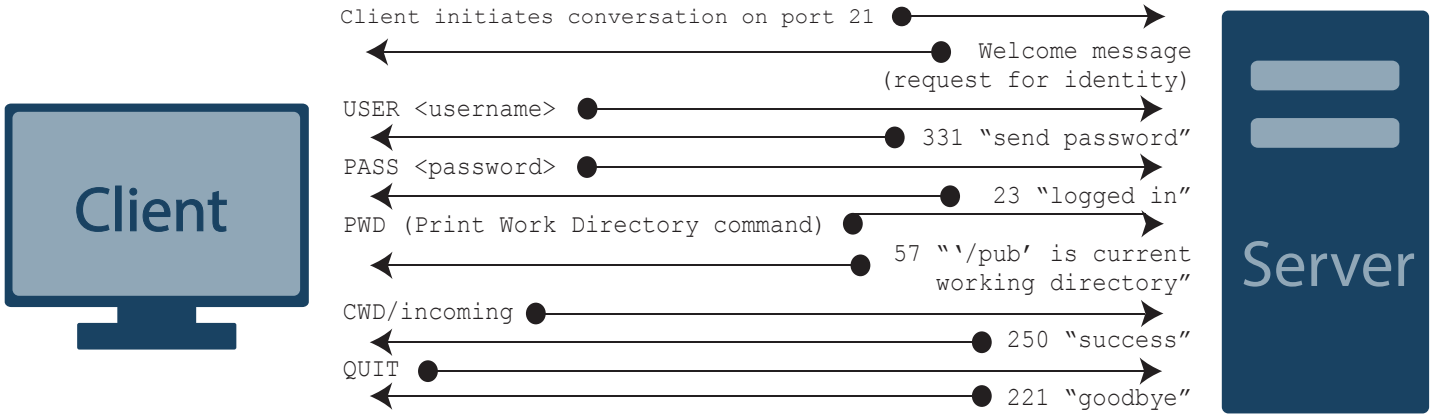
FTP uses a basic command/reply mechanism. The FTP client will connect to the FTP server, usually on port 21, the port traditionally reserved for FTP traffic. The client will begin a synchronized conversation by sending a command to the server, which the server will respond to, signaling readiness for the next command.

Responses from the server come in a standardized format. The first three characters of the response will be a 3-digit response code. The codes have the same general meanings, though the exact message that follows may vary. The first digit of the response code is the most important, as it is an indicator of the overall success or failure status of the command. Generally response codes follow these rules:

- **1, 2, or 3 are good**
- **4 or 5 are not good**

For example, if the client were to issue a Change Working Directory command to change the current directory to /incoming/ (using the CWD/incoming/ command) the server could reply with a 200 "Success" response, indicating that the command succeeded. The server might also reply with a 500-level command, such as 550 "Access Denied," indicating that the client does not have adequate rights to access the specified directory.

Client-Server Conversation to Establish FTP Transfer



Control Connections and Data Connections

Assuming the client successfully establishes a conversation with the server and passes authentication, the client will then attempt to download or retrieve a file (filename.txt) from the server.

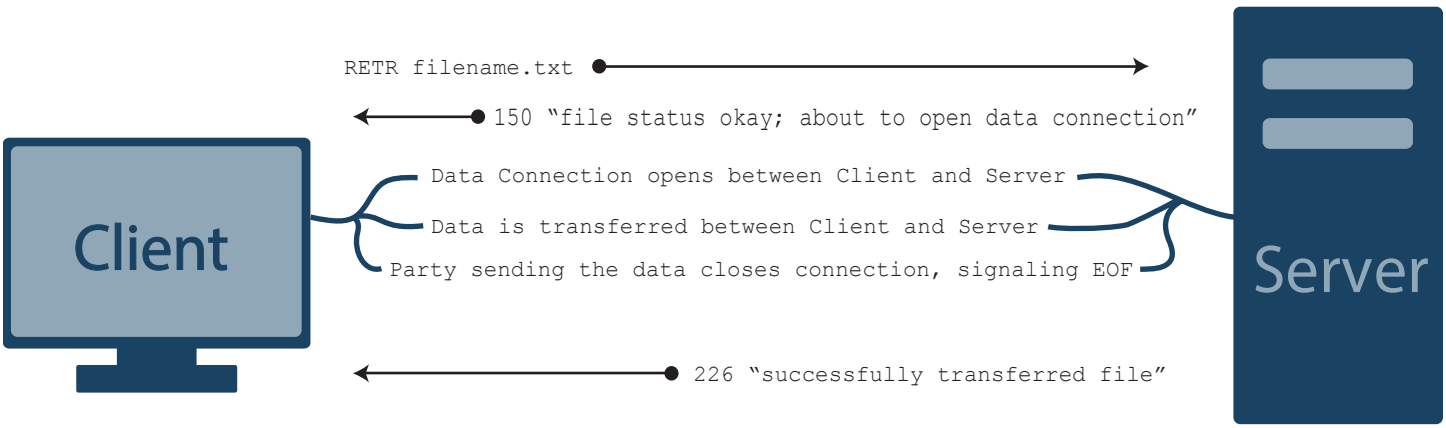
In a typical session where files are transferred, FTP will use two separate connections: the Control and Data Connections.

Control Connections

The **Control Connection**, generally established on port 21, is the primary connection and is used to send commands back and forth between the client and server.

After establishing a connection or "handshake," the client issues the retrieve command, RETR, to initiate the file transfer, followed by the name of the file to be retrieved. If the file exists and if the client has rights to access the file, the server will issue a reply indicating that everything is OK and that the file transfer will now begin.

Typical File Transfer after Handshake is Established



Data Connections

Using the established Control Connection, the client and server will create a separate **Data Connection**, used solely to transfer the requested data. The Data Connection stays open until the transfer is complete, after which the Data Connection is closed. Data Connections are opened on a port negotiated by the client and server prior to the RETR command (see Passive and Adctive Modes below). Data Connections are closed by either the client or the server, depending on which party is **sending** the information. When a client is retrieving data **from** a server, the server will close the connection once all data has been transferred. When the client is transferring data **to** the server, the client will terminate the connection when all information is transferred.

Passive and Active Modes

When a client and server intend to transfer data, they usually negotiate the details of the Data Connection prior to opening it. Though RFC-959 defines a mechanism for pre-negotiated details, FTP clients rarely rely on the default values; if the client fails to issue a PASV or PORT command, the Data Connection defaults to port 20. Nearly all FTP clients will explicitly specify the IP address and port for the Data Connection each time a file is transferred. The IP address used for the original Control Connection must be combined with an unused port—usually a port numbered higher than 1024 and and lower than 65535. Ports below 1024, other than port 20, are reserved for system services.

During the address/port negotiation phase, the client will issue either the PORT command (when in Active Mode) or the PASV command (when in Passive Mode).

- **Active Mode**—The client issues a PORT command to the server signaling that the client will "actively" provide an IP and port number to open the Data Connection back to the client.
- **Passive Mode**—The client issues a PASV command to indicate that the client will wait "passively" for the server to supply an IP and port number, after which the client will create a Data Connection to the server.

Once the IP address and port number have been selected, the party that chose the IP address and port will begin to listen on the address/port specified and wait for the other party to connect. When the other party connects to the listening party, the data transfer begins.

After the data has been transferred, the party that has sent the data will close the Data Connection, signaling end-of-file (EOF).

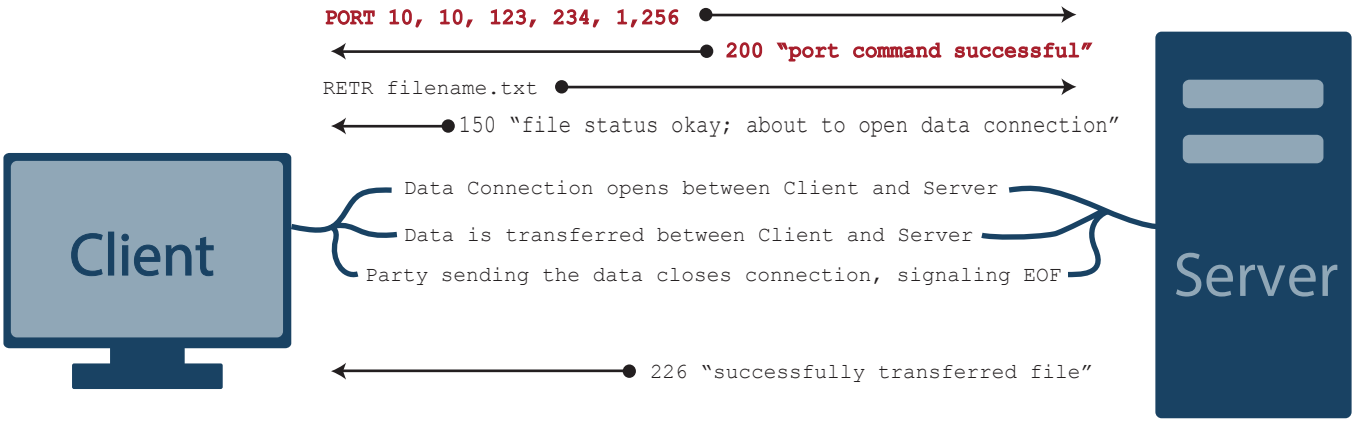
Why Active or Passive?

The ability to choose between Active and Passive Mode Data Connections primarily comes in handy when navigating firewalls.

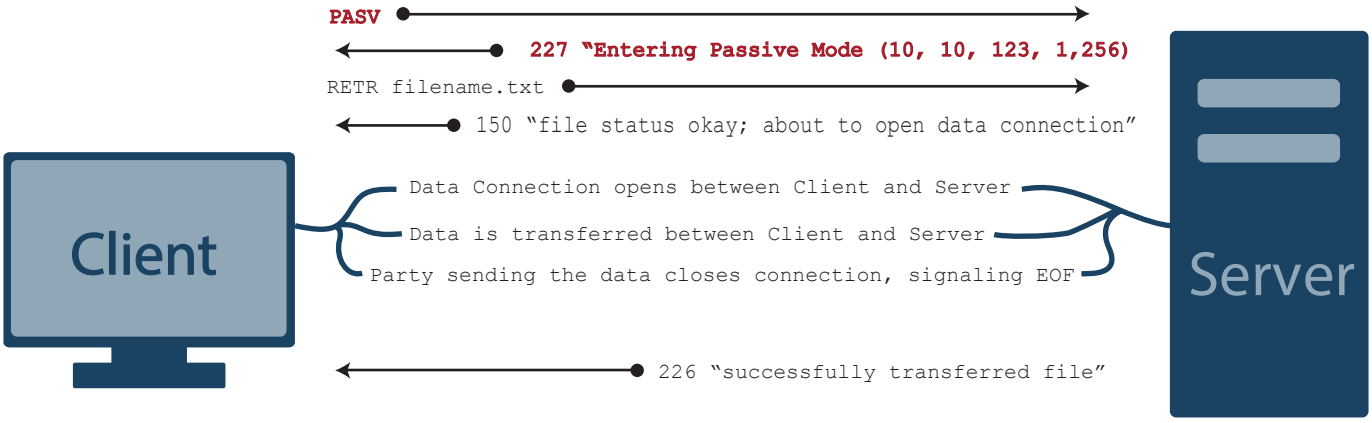
For instance, say you have an FTP server sitting behind a firewall. The firewall blocks all incoming traffic

except for traffic bound for port 21 (the default for FTP traffic), but the server is able to send information out through the firewall freely. If the FTP client were to issue the PASV command to the FTP server, the server would respond with an IP address and port that the client should use to connect back to the server. However, since the firewall is blocking access to all ports except 21, the FTP client will not be able to connect to server's chosen port. To correct this problem, the FTP client would need to issue the PORT command to the server. Since the client is now the active entity in establishing the connection, the server would be able to open an outbound connection of its choice through the firewall to the FTP client's designated port to make a connection only long enough to make a secure transaction.

Active Mode File Transfer



Passive Mode File Transfer



However, due to today's security-conscious environment, clients usually have a firewall as well, which would prevent the Server from opening a connection back to the client in response to the PORT command. Many FTP servers now specify a Passive Port Range/Block, which defines a pool of ports set aside for connections with the FTP server, which would be applied to the firewall. The firewall administrator would then set up routing rules on the corporate Firewall so that any connections from clients on these ports would be automatically forwarded to the FTP Server. The corporate FTP Server could then use PASV mode to instruct the client to use one of the ports within the allowed server port range.

Conclusion

The longevity of FTP has ensured its strong foothold and widespread acceptance in the Internet community. There are numerous FTP clients and servers on the market today, nearly all of which support the features of RFC-959. Nearly all internet service and broadband providers supply FTP features which allow customers to upload pages to their websites. An FTP server is the preferred repository for software patches and drivers for many hardware vendors. For instance, Linksys and Dell have FTP servers to house patches and drivers for their products, and customers are encouraged to download updates as needed.

Recent revisions to the original RFC-959 specification include added support for security extensions that allow FTP traffic to be secure. Many servers support FTPS, or FTP over SSL (Secure Sockets Layer). When preparing to replace company FTP servers, many enterprises are choosing to upgrade to an FTPS server instead of more expensive Virtual Private Network (VPN) or Secure Shell (SSH) servers.