FTP

File Transfer Protocol

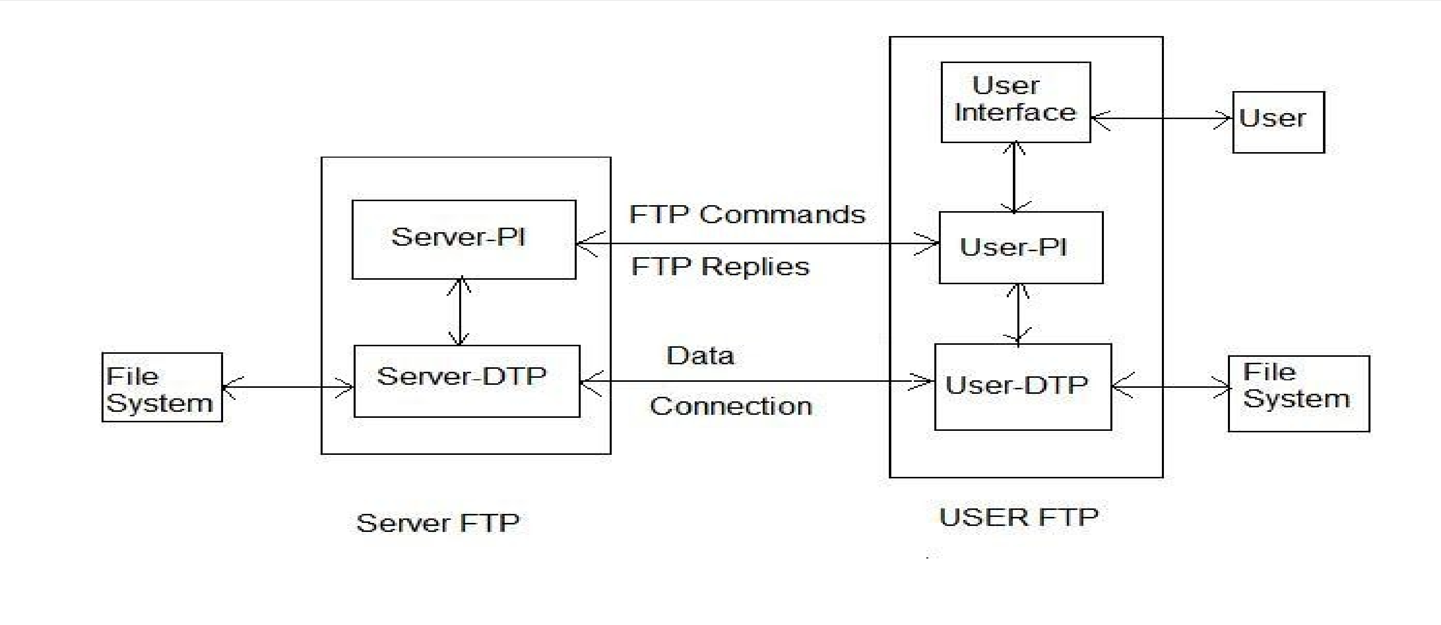
Introduction

- FTP is one of the oldest application protocols still used on the Internet, and is invoked by web browsers when a user requests a file download.
- FTP predates both IP and TCP. As TCP/IP was created, a new version of FTP was developed to work with the new Internet protocols. FTP is still heavily used -- only in 1995 did web traffic on the Internet surpass FTP traffic for the first time.
- File Transfer Protocol requires two programs, a server program, and a client program. Normally the server program offer files to the client program. But in some cases, the server will also allow the client to upload files.

Objectives

- 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
- to transfer data relaible and efficiently

FTP Model

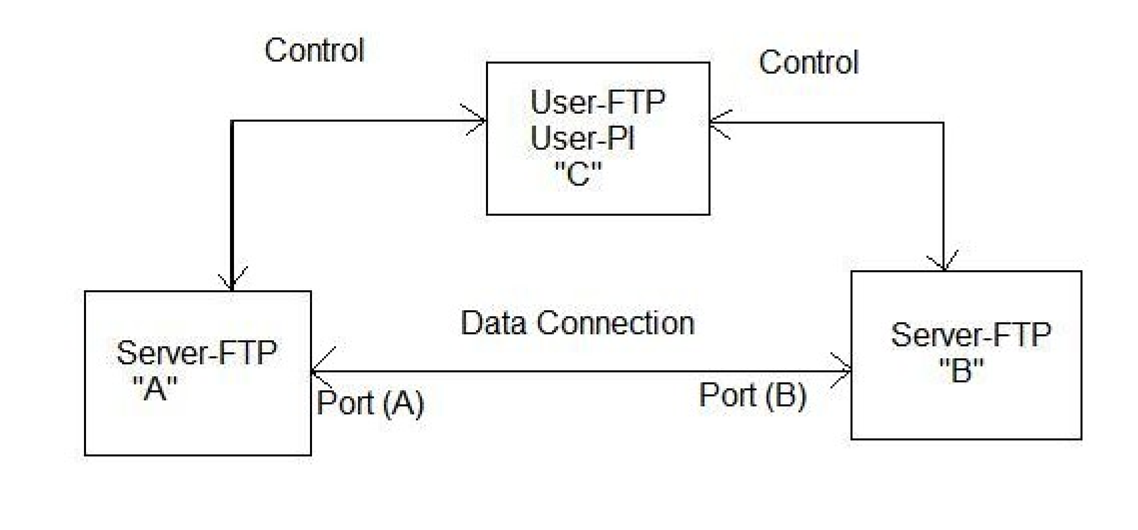


- The data connection may be used in either direction.
- The data connection need not exist all of the time.
- Figure 1 Model for FTP Use
- The Protocol Interpreter (PI) implements the FTP protocol itself, while the Data Transfer Process (DTP) actually performs data transfer. The FTP protocol and the data transfer use entirely separate TCP sessions.

Model for FTP Use

- The user-protocol interpreter(user-PI) initiates the control connection
- 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
- 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-DTP 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.
- the data connection may be used for simultaneous sending and receiving

File Transfer between two hosts



- The user sets up control connections to the two servers and then arranges for a data connection between them.
- The protocol requires that the control connections be open while data transfer is in progress.
- The FTP uses the Telnet protocol on the control connection.

Data Transfer Functions

- Files are transferred only via the data connection.
- The control connection is used for the transfer of commands, which describe the functions to be performed, and the replies to these commands
- Data transfer commands include the MODE command which specify how the bits of the data are to be transmitted, and the STRUcture and TYPE commands, which are used to define the way in which the data are to be represented.

Data representatin and storage

- Data is transferred from a storage device in the sending host to a storage device in the receiving host. Often it is necessary to perform certain transformations on the data because data storage representations in the two systems are different.
- A different problem in representation arises when transmitting binary data (not character codes) between host systems with different word lengths.
- FTP provides for very limited data type representations.

Data Type

- Data representations are handled in FTP by a user specifying a representation type.

  •ASCII TYPE This is the default type and must be accepted by all FTP

  • EBCDIC TYPE This type is intended for efficient transfer between hosts which use EBCDIC for their internal character representation.

  • IMAGE TYPE

  •LOCAL TYPE

  •FORMAT CONTROL

    - NON PRINT
    - TELNET FORMAT CONTROLS
    - CARRIAGE CONTROL (ASA)

Data Structures

- Three file structures are defined in FTP:
  - file-structure: where there is no internal structure and the file is considered to be a continuous sequence of data bytes
  - record-structure: where the file is made up of sequential records
  - page-structure: where the file is made up of independent   indexed pages.

Establishing data connection

- The mechanics of transferring data consists of setting up the data connection to the appropriate ports and choosing the parameters for transfer.
- The user-process default data port is the same as the control connection port (i.e., U). The server-process default data port is the port adjacent to the control connection port (i.e., L-1).
- The transfer byte size is 8-bit bytes.
- The passive data transfer process (this may be a user-DTP or a second server-DTP) shall "listen" on the data port prior to sending a transfer request command.
- The FTP request command determines the direction of the data transfer. The server, upon receiving the transfer request, will initiate the data connection to the port. When the connection is established, the data transfer begins between DTP's, and the server-PI sends a confirming reply to the user-PI.
- In general, it is the server's responsibility to maintain the data connection--to initiate it and to close it.

Data Connection Management

- Default Data Connection Ports:
  - Every FTP implementation must support the use of the default data ports, and only the USER-PI can initiate a change to non-default ports.
- Negotiating Non-Default Data Ports:
  - The User-PI may specify a on-default user side data port with the PORT command.
  - The User-PI may request the server side to identify a non-default server side data port with the PASV command.
- Reuse of the Data Connection:
  - When using the stream mode of data transfer the end of the file must be indicated by closing the connection. This causes a problem if multiple files are to be transfered in the session, due to need for TCP to hold the connection record for a time out period to guarantee the reliable communication. Thus the connection can not be reopened at once.
  - There are two solutions to this problem.
    - The first is to negotiate a non-default port.
    - The second is to use another transfer mode

Transmission Mode

- There are three modes:
  - STREAM MODE : mode which passes the data with little or no processing. the data is transmitted as a stream of bytes;
  - BLOCK MODE :. one which formats the data and allows for restart procedures; The file is transmitted as a series of data blocks preceded by one or more header bytes
  - COMPRESSED MODE one which also compresses the data for efficient transfer;
- All data transfers must be completed with an end-of-file (EOF) which may be explicitly stated or implied by the closing of the data connection.
- For the purpose of standardized transfer, the sending host will translate its internal end of line or end of record denotation into the representation prescribed by the transfer mode and file structure, and the receiving host will perform the inverse translation to its internal denotation.

Error Recovery and Restart

- There is no provision for detecting bits lost or scrambled in data transfer; this level of error control is handled by the TCP.
- However, a restart procedure is provided to protect users from gross system failures (including failures of a host, an FTP-process, or the underlying network).
- The restart procedure is defined only for the block and compressed modes of data transfer.
- It requires the sender of data to insert a special marker code in the data stream with some marker information. The marker could represent a bit-count, a record-count, or any other information by which a system may identify a data checkpoint.
- The receiver of data, if it implements the restart procedure, would then mark the corresponding position of this marker in the receiving system, and return this information to the user.

File Transfer Functions

- The communication channel from the user-PI to the server-PI is established as a TCP connection from the user to the standard server port.
- The user protocol interpreter is responsible for sending FTP commands and interpreting the replies received;
- The server-PI interprets commands, sends replies and directs its DTP to set up the data connection and transfer the data.

FTP Commands
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
- PASSWORD (PASS) The argument field is a Telnet string specifying the user's password.
- ACCOUNT (ACCT) The argument field is a Telnet string identifying the user's account.
- CHANGE WORKING DIRECTORY (CWD)allows the user to work with a different directory or dataset for file storage or retrieval without altering his login or accounting information.
- CHANGE 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.
- STRUCTURE MOUNT (SMNT) This command allows the user to mount a different file system data structure without altering his login or accounting information.
- REINITIALIZE (REIN) terminates a USER, flushing all I/O and account information, except to allow any transfer in progress to be completed.
- LOGOUT (QUIT) This command terminates a USER and if file transfer is not in progress, the server closes the control connection.

Transfer Parameter Commands

- All data transfer parameters have default values, and the commands specifying data transfer parameters are required only if the default parameter values are to be changed.
- The following commands specify data transfer parameters:
  - •DATA PORT (PORT)The argument is a HOST-PORT specification for the data port to be used in data connection.
  - 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.
  - •REPRESENTATION TYPE (TYPE)The argument specifies the representation type as described in the Section on Data Representation and Storage. Several types take a second parameter.
  - •FILE STRUCTURE (STRU)The argument is a single Telnet character code specifying file structure described in the Section on Data Representation and Storage.
  - •TRANSFER MODE (MODE)The argument is a single Telnet character code specifying the data transfer modes described in the Section on Transmission Modes

FTP Service Commands

- The FTP service commands define the file transfer or the file system function requested by the user. The argument of an FTP service command will normally be a pathname. The data, when transferred in response to FTP service commands, shall always be sent over the data connection, except for certain informative replies.
- The following commands specify FTP service requests:
  - RETRIEVE (RETR)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.
  - 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.
  - STORE UNIQUE (STOU)behaves like STOR except that the resultant file is to be created in the current directory under a name unique to that directory
  - APPEND (with create) (APPE)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.
  - ALLOCATE (ALLO) may be required by some servers to reserve sufficient storage to accommodate the new file to be transferred.
  - RESTART (REST)The argument field represents the server marker at which file transfer is to be restarted.
  - RENAME FROM (RNFR) specifies the old pathname of the file which is to be renamed.

FTP Service Commands

- RENAME TO (RNTO)specifies the new pathname of the file specified in the immediately preceding "rename from" command.
- ABORT (ABOR)tells the server to abort the previous FTP service command and any associated transfer of data
- DELETE (DELE)causes the file specified in the pathname to be deleted at the server site.
- REMOVE DIRECTORY (RMD)causes the directory specified in the pathname to be removed as a directory (if the pathname is absolute) or as a subdirectory of the current working directory (if the pathname is relative)
- MAKE DIRECTORY (MKD)causes the directory specified in the pathname to be created as a directory (if the pathname is absolute) or as a subdirectory of the current working directory (if the pathname is relative)
- PRINT WORKING DIRECTORY (PWD)causes the name of the current working directory to be returned in the reply
- LIST (LIST)causes a list to be sent from the server to the passive DTP.
- NAME LIST (NLST)causes a directory listing to be sent from server to user site
- SITE PARAMETERS (SITE)used by the server to provide services specific to his system that are essential to file transfer but not sufficiently universal to be included as commands in the protocol.
- SYSTEM (SYST)used to find out the type of operating system at the server.
- STATUS (STAT)shall cause a status response to be sent over the control connection in the form of a reply
- HELP (HELP)shall cause the server to send helpful information regarding its implementation status over the control connection to the user
- NOOP (NOOP) specifies no action other than that the server send an OK reply.

FTP Replies

- Replies to File Transfer Protocol commands are devised to ensure the synchronization of requests and actions in the process of file transfer, and to guarantee that the user process always knows the state of the Server.
- Every command must generate at least one reply.
- A FTP reply consists of a three digit number (transmitted as three alphanumeric characters) followed by some text.
- The number is intended for use by automata to determine what state to enter next; the text is intended for the human user.
- The first digit denotes whether the response is good, bad or incomplete.
- A user-process that wants to know approximately what kind of error occurred (e.g. file system error, command syntax error) may examine the second digit, reserving the third digit for the finest gradation of information (e.g., RNTO command without a preceding RNFR).