File Transfer Protocol

What is FTP?

- File transfer protocol: Protocol used to copy a file from one host to another.
- It is an application layer protocol that uses the TCP/IP.
- Why use FTP over HTTP?
  - FTP can transfer large files or transfer files using different formats.
  - Requires authentication which HTTP doesn't provide and hence is a secure form of communication.
  - However, there is no form of encryption involved so it provides easy access for hackers to obtain your username, password etc.
- Why the need for such a protocol?
  - Different file name conventions.
  - Data representation may be different.
  - Different kinds of directory structures.

How to access a FTP server?

- Using the browser:- Type ftp://address to ensure the browser uses FTP and not HTTP
- Using the File Explorer:- Type ftp://address
- Using an FTP Client:- Type ftp://address , username, password and port number (depends on the Client's configuration)

Uses TCP so a 3-way handshake is done initially to establish a connection between client and server.

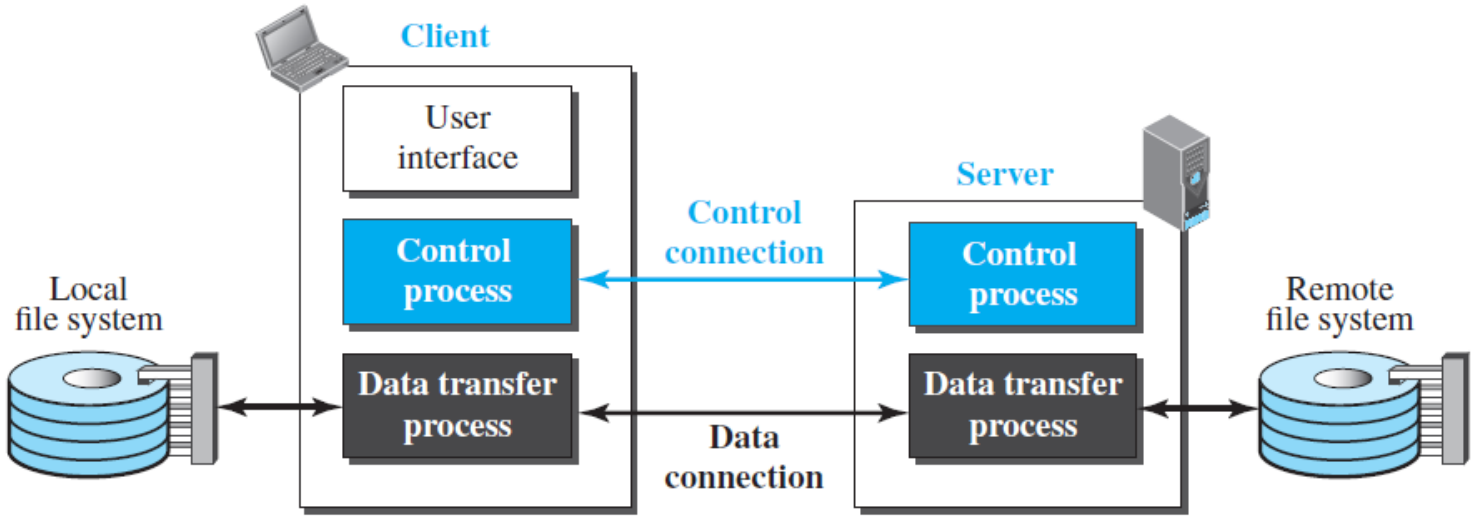
Control Connection:

- Established between the control processes of the Client and Server.
- Simple rules for communication:
- Only a line of command or response required.

Data Connection:

- More complex rules and commands due to the variety of data formats supported by FTP.

Figure 26.10 *FTP*



Control connection: Connected during the entire interactive FTP session.
Data connection: Opened and then closed for each file transfer activity. It opens each time commands that involve transferring files are used, and it closes when the file is transferred.

Control Connection

- Uses the NVT ASCII Character set as used by TELNET (used to virtually access a computer and provide a two-way, collaborative and text-based communication channel between two machines)
- Disadvantage: Plain text transmission, so hacker can eavesdrop and obtain the information.
- Each command is terminated with a 2-character end of line token. (\r-carriage return and \n-line feed)
- Client sends to server a COMMAND, Server sends to client a RESPONSE

Control Connection- Commands (Client to Server)

Commands sent from FTP client's "control process" are in the form of ASCII uppercase which may or may not be followed by arguments.

Table 26.4 *Some FTP commands*

Command	*Argument(s)*	*Description*
ABOR		Abort the previous command
CDUP		Change to parent directory
CWD	Directory name	Change to another directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
MKD	Directory name	Create a new directory
PASS	User password	Password
PASV		Server chooses a port
PORT	Port identifier	Client chooses a port
PWD		Display name of current directory
QUIT		Log out of the system
RETR	File name(s)	Retrieve files; files are transferred from server to client
RMD	Directory name	Delete a directory
RNFR	File name (old)	Identify a file to be renamed
RNTO	File name (new)	Rename the file
STOR	File name(s)	Store files; file(s) are transferred from client to server
STRU	**F**, **R**, or **P**	Define data organization (**F**: file, **R**: record, or **P**: page)
TYPE	**A**, **E**, **I**	Default file type (**A**: ASCII, **E**: EBCDIC, **I**: image)
USER	User ID	User information
MODE	**S**, **B**, or **C**	Define transmission mode (**S**: stream, **B**: block, or **C**: compressed

Control Connection-
Responses (Server to Client)

- Every FTP command generates at least ONE response.
- Response consists of two fields: One numeric field and one text field.

Numeric field:

- The first digit defines the status of the command.
- The second digit defines the area in which the status applies.
- The third digit provides additional information.

Text field:

- Define parameters or other explanations.

Table 26.5 *Some responses in FTP*

Code	*Description*	*Code*	*Description*
125	Data connection open	**250**	Request file action OK
150	File status OK	**331**	User name OK; password is needed
200	Command OK	**425**	Cannot open data connection
220	Service ready	**450**	File action not taken; file not available
221	Service closing	**452**	Action aborted; insufficient storage
225	Data connection open	**500**	Syntax error; unrecognized command
226	Closing data connection	**501**	Syntax error in parameters or arguments
230	User login OK	**530**	User not logged in

Data Connection

- Two kinds of data connections:

1. Active: Server initiates the session. This fails when there is a firewall present as it will not let the connection to be established.

2. Passive: Client initiates the session. This can work even in the presence of a firewall.

- Establishment of the connection:

1. The client issues a passive open using an ephemeral (lasting for a short period of time) port. This must be done by the client because it is the client that issues the commands for transferring files.
2. Using the PORT command, the client sends this port number to the server.
3. The server receives the port number and issues an active open using the well-known port 20 and the received ephemeral port number.

Communication over the data connection

- Preparation for transmission is done using the control connection. The following must be defined by the client:

1. File Type: ASCII, EBCDIC (Extended binary coded decimal interchange code: 8bit character encoding) , Image

2. Data Structure: File, Record, Page
    - File- No default structure, just a continuous stream of bytes
    - Record- File divided into records and can be used only with text files
    - Pages- File divided into pages, each comprising a page header and number (access: random or sequential)

3. Transmission Mode: Stream, Block, Compressed
    - Block: Each block preceded by 3byte header:
        - First byte- Block Descriptor
        - Second and third byte- Size of block in bytes

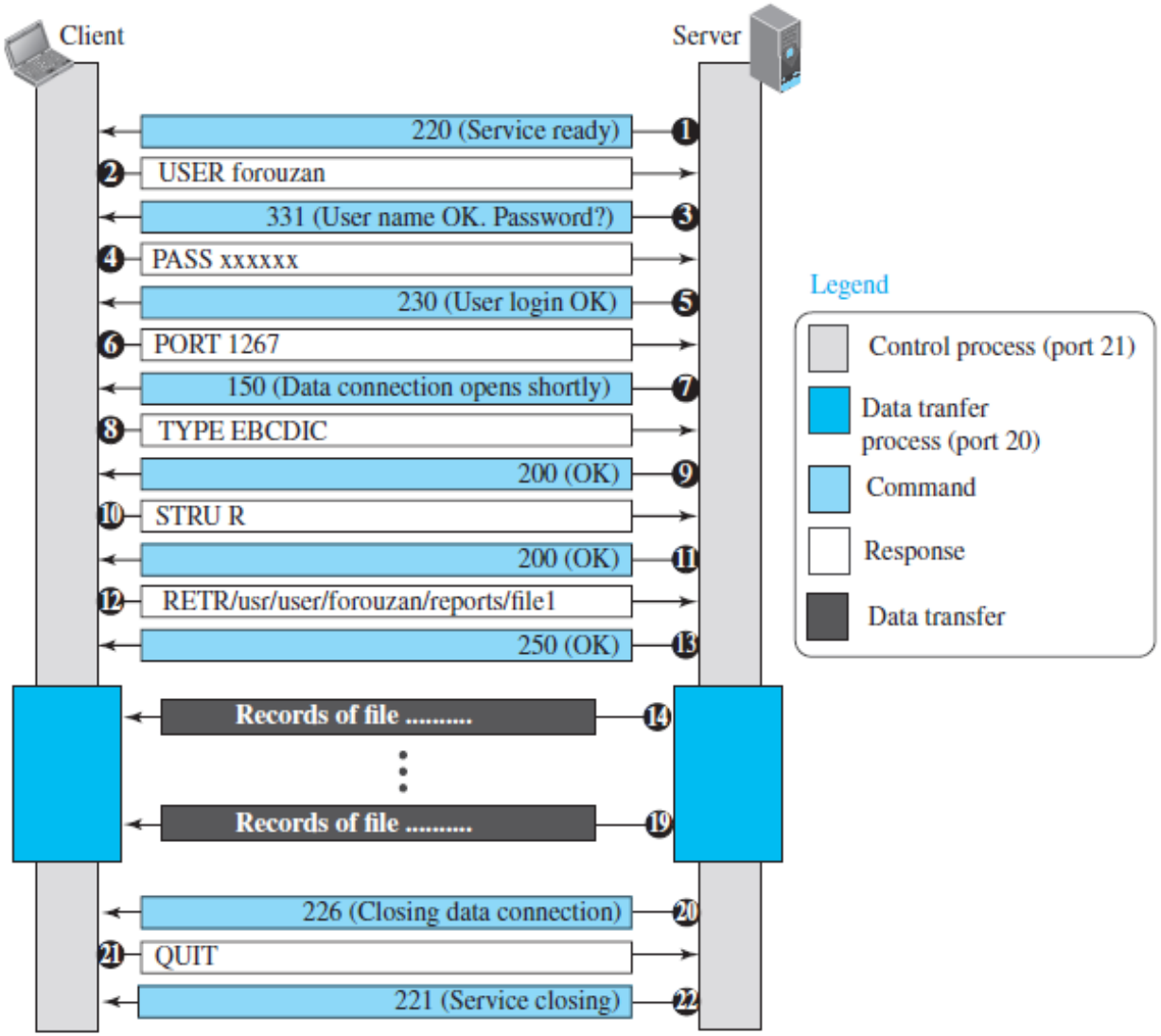
- File transfer- through data connection controlled by control section commands
    - Receiving a file (S to C), Storing a file (C to S), Directory listing (S to C)

1 – 13 : Commands and responses sent through the CONTROL connection.

14 – 19: Data transfer in the form of records sent through the DATA connection.

20 – 22: Termination commands and responses sent through the CONTROL connection.

Figure 26.11 *Example 26.10*



Black lines: Commands from the client

Blue lines: Responses from the server

Black background with White lines: Data transfer

```
$ ftp voyager.deanza.fhda.edu
Connected to voyager.deanza.fhda.edu.
220 (vsFTPd 1.2.1)
530 Please login with USER and PASS.
Name (voyager.deanza.fhda.edu:forouzan): forouzan
331 Please specify the password.
Password:*********
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
227 Entering Passive Mode (153,18,17,11,238,169)
150 Here comes the directory listing.
drwxr-xr-x    2    3027    411    4096    Sep 24    2002    business
drwxr-xr-x    2    3027    411    4096    Sep 24    2002    personal
drwxr-xr-x    2    3027    411    4096    Sep 24    2002    school
226 Directory send OK.
ftp> quit
221 Goodbye.
```

FTP Security

- Despite FTP requiring a password, the password is sent in plaintext (unencrypted), which means it can be intercepted and used by an attacker.
- The data transfer connection also transfers data in plaintext, which is insecure.
- To be secure, one can add a Secure Socket Layer between the FTP application layer and the TCP layer. In this case FTP is called SSL-FTP (FTPS). => Uses TCL and SSL Encryption

Thank You