

# COMPUTER NETWORKS AND INTERNET PROTOCOLS

## Application Layer – II [Cli-Srv, FTP]

**SOUMYA K GHOSH**

COMPUTER SCIENCE AND ENGINEERING  
IIT KHARAGPUR

**SANDIP CHAKRABORTY**

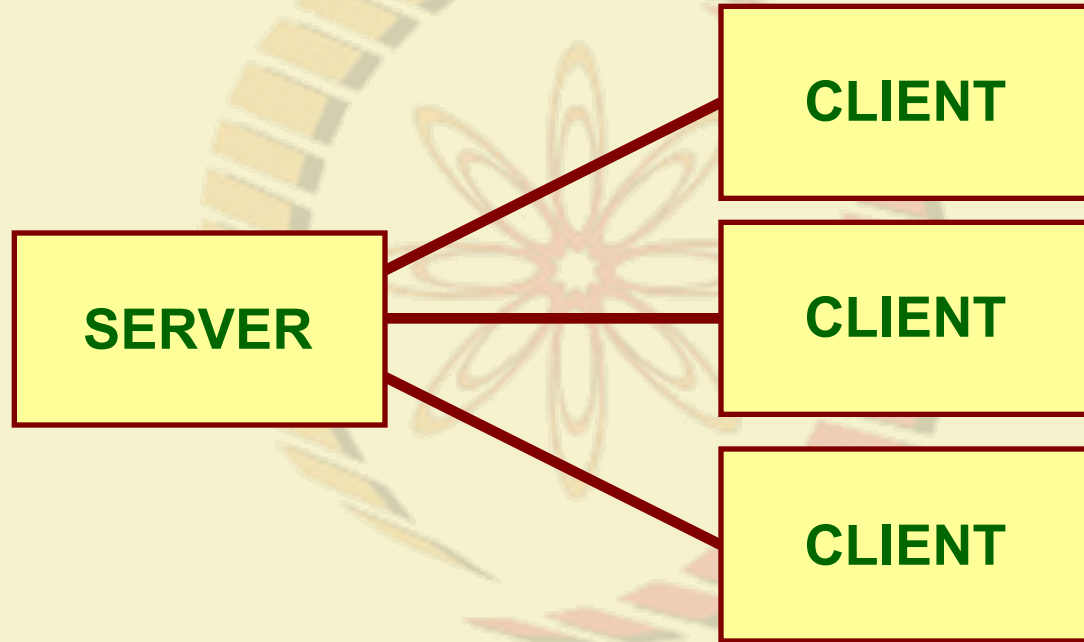
COMPUTER SCIENCE AND ENGINEERING  
IIT KHARAGPUR

# Client-Server Model

# Client-server Model

- Standard model for developing network applications.
- Notion of client and server.
  - A server is a process that is offering some service.
  - A client is a process that is requesting the service.
  - Server or client may be running on different machines.
  - Server waits for requests from client(s).

# Client-Server Model (contd.)



# Client-Server Model (contd.)

- Typical scenario:
  - The server process starts on some computer system.
    - Initializes itself, then goes to sleep waiting for a client request.
  - A client process starts, either on the same system or on some other system.
    - Sends a request to the server.

# Client-Server Model (contd.)

- When the server process has finished providing its service to the client, the server goes back to sleep, waiting for the next client request to arrive.
- The process repeats.

# Client-Server Model (contd.)

- Roles of the client and the server processes are asymmetric.
- Two types of servers:
  - *Iterative servers.*
  - *Concurrent servers.*

# Iterative Servers

- Used when the server process knows in advance how long it takes to handle each request and it handles each request itself.
  - Single copy of server runs at all times.
  - A client may have to wait if the server is busy.



# Concurrent Servers

- Used when the amount of work required to handle a request is unknown; the server starts another process to handle each request.
  - A copy of the server caters to a client's request in a dedicated fashion.
  - As many copies of server as there are client requests.

# Using TCP or UDP

- Before start of communication, a connection has to be established between the two hosts.
- Five components in a connection:
  - Protocol used
  - Source IP address
  - Source port number
  - Destination IP address
  - Destination port number

# Develop a Network Application

- The best way is to use some standard and well-accepted protocol.
  - At the data link layer level, use **Ethernet**.
  - At the network layer level, use **IP**.
  - At the transport layer level, use **TCP**.
  - At the application layer level, use a standard API like the **Berkeley Socket Interface**.

# What is a Socket?

- The *socket* is the method for achieving inter-process communication (IPC).
- It is used to allow one process to speak to another (on same or different machine).
  - *Analogy*: Like the telephone is used to allow one person to speak to another.

# Socket - Basic Idea

- When two processes located on two machines communicate, we define association and socket.
  - *Association*: basically a 5-tuple
    - Protocol
    - Local IP address
    - Local port number
    - Remote IP address
    - Remote port number

# Socket - Basic Idea

- *Socket*: also called half-association (a 3-tuple)
  - Protocol, local IP address, local port number
  - Protocol, remote IP address, remote port number

# File Transfer Protocol (FTP)



IIT KHARAGPUR



NPTEL ONLINE  
CERTIFICATION COURSES

# FTP

- Facilitates transfer of files over network
- Server/Client model
- FTP often works with
  - Transmission Control Protocol (TCP)
  - Telnet Protocol
- Defined as RFC959



# Overview of FTP

- FTP uses TCP as a transport protocol to provide reliable end-to-end connections and implements two types of connections in managing data transfers.
- The FTP client initiates the first connection, referred to as the control connection, to well-known port 21 (the client's port is typically ephemeral). It is on this port that an FTP server listens for and accepts new connections.
- The control connection is used for all of the control commands a client user uses to log on to the server, manipulate files, and terminate a session. This is also the connection across which the FTP server will send messages to the client in response to these control commands.

*Ref: IBM Redbooks*

# Overview of FTP (contd.)

- The second connection used by FTP is referred to as the data connection.
- Typically, the data connection is established on server port 20. However, depending on how the data connection is established, both the client and server might use ephemeral ports.
- FTP transfers the data over data connection. FTP only opens a data connection when a client issues a command requiring a data transfer, such as a request to retrieve a file, or to view a list of the files available. It is possible for an entire FTP session to open and close without a data connection ever having been opened.
- The data connection is unidirectional. FTP can transfer data only from the client to the server, or from the server to the client, but not both.
- The data connection can be initiated from either the client or the server. Data connections initiated by the server are active, while those initiated by the client are passive.

# FTP – Basic working

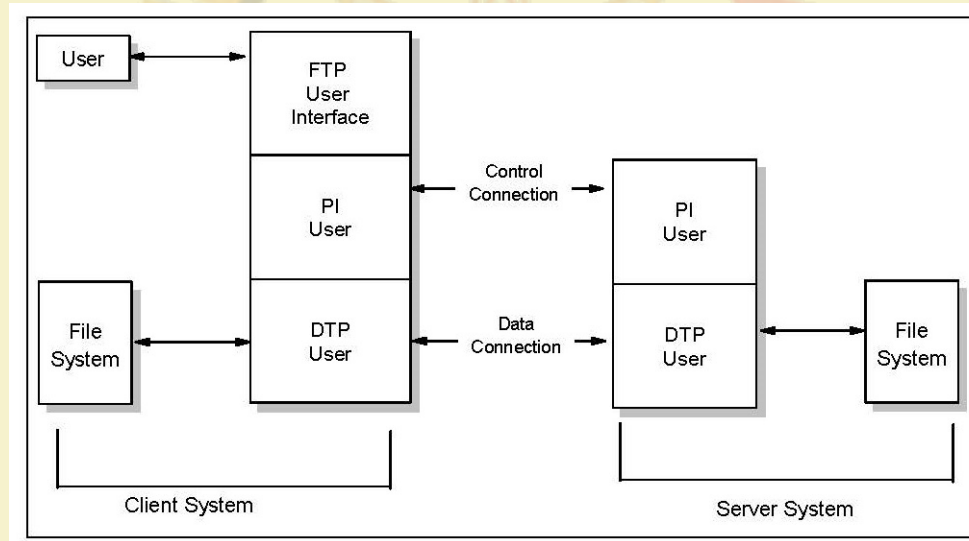
- FTP has to be on both server and client computers to work
- Connection
  - Control connection (port 21)
    - Used to send and receive FTP commands
  - Data connection (port 20)
    - Used to upload and download files
- Processes
  - Data Transfer Process (DTP)
    - Establishing the connection and managing the data channel
  - Protocol Interpreter (PI)
    - Interprets the protocol
    - let DTP be controlled using commands received over the control channel

# FTP – Basic working (contd.)

- Transferring mode between server and client
  - Active mode
    - Control connection port: Client: Large port ( $N > 1023$ ); Server: 21
    - Data connection port: Client:  $N+1$ ; Server: 20
  - Passive mode
    - Control connection port: Client: Large port ( $N > 1023$ ); Server: 21
    - Data connection port: Client:  $N+1$ ; Server: large port ( $P > 1023$ )
- File Transferring mode
  - ASCII mode
    - .txt, .html, .asp, .vbs, .js
  - Binary
    - .doc, .pdf, .mp3/mp4

# FTP – Basic working (contd.)

- The client FTP application is built with a protocol interpreter (PI), a data transfer process (DTP), and a user interface.
- The server FTP application typically only consists of a PI and DTP



Ref: IBM Redbooks

# FTP – Basic working (contd.)

- FTP client's user interface communicates with the protocol interpreter (PI), which manages the control connection.
- PI translates any application-specific commands to the RFC architected FTP commands, and then communicates these control commands to the FTP server.
- The FTP server's PI receives these commands, and then initiates the appropriate processes to service the client's requests. If the requests require the transfer of data, data management is performed by the DTPs on both the client and server applications.
- After the completion of the data transfer, the data connection is closed, and control is returned to the PIs of the client and server applications.
- Only one data transfer can occur for each data connection. If multiple data transfers are required for a single FTP session, one distinct control connection will be opened for each transfer.

*Ref: IBM Redbooks*

# FTP Operation – User's Perspective

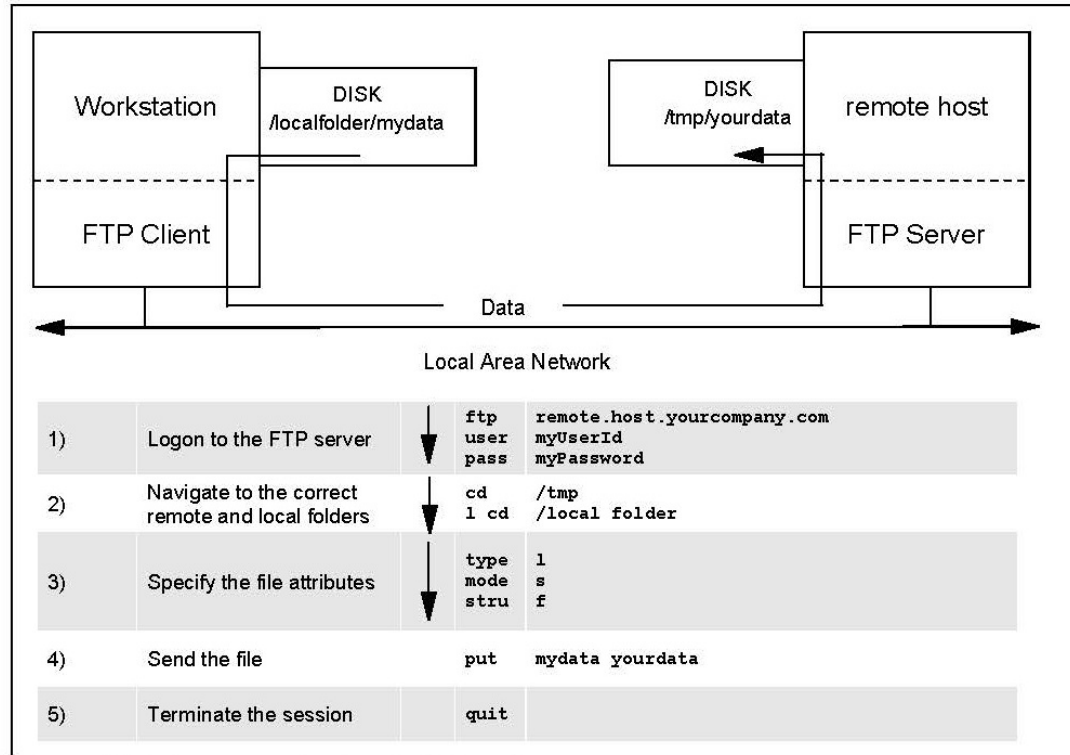
When using FTP, the user performs some or all of the following operations:

- Connect to a remote host.
- Navigate and manipulate the directory structure.
- List files available for transfer.
- Define the transfer mode, transfer type, and data structure.
- Transfer data to or from the remote host.
- Disconnect from the remote host.

*Ref: IBM Redbooks*



# A Typical FTP scenario



Ref: IBM Redbooks



# Trivial File Transfer Protocol (TFTP)

- TFTP file transfer is a disk-to-disk data transfer, and is a simple protocol used to transfer files. The simplicity of the architecture is deliberate in order to facilitate ease of implementation.
- This simplistic approach has many benefits over traditional FTP, including:
  - Use by diskless devices to download firmware at boot time
  - Use by any automated process for which the assignment of a user ID or password is not feasible
  - Small application size, allowing it to be implemented inexpensively and in environments where resources are constricted
- TFTP is implemented on top of the User Datagram Protocol.
- The TFTP client initially sends read/write request through well-known port 69. The server and the client then determine the port that they will use for the rest of the connection.
- TFTP lacks most of the features of FTP, and instead is limited to only reading a file from a server or writing a file to a server.
- TFTP has no provisions for user authentication; in that respect, it is an insecure protocol.

*Ref: IBM Redbooks*

# FTP – Access Commands

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
<b>USER</b>	User id	User information
<b>PASS</b>	User password	Password
<b>ACCT</b>	Account to be charged	Account information
<b>REIN</b>		Reinitialize
<b>QUIT</b>		Log out of the system
<b>ABOR</b>		Abort the previous command

Ref: TCP/IP Protocol Suite

# FTP – File Management Commands

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
<b>CWD</b>	Directory name	Change to another directory
<b>CDUP</b>		Change to the parent directory
<b>DELE</b>	File name	Delete a file
<b>LIST</b>	Directory name	List subdirectories or files
<b>NLIST</b>	Directory name	List the names of subdirectories or files without other attributes
<b>MKD</b>	Directory name	Create a new directory
<b>PWD</b>		Display name of current directory
<b>RMD</b>	Directory name	Delete a directory
<b>RNFR</b>	File name (old file name)	Identify a file to be renamed
<b>RNTO</b>	File name (new file name)	Rename the file
<b>SMNT</b>	File system name	Mount a file system

Ref: TCP/IP Protocol Suite

# FTP – Data Formatting Commands

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
<b>TYPE</b>	A (ASCII), E (EBCDIC), I (Image), N (Nonprint), or T (TELNET)	Define the file type and if necessary the print format
<b>STRU</b>	F (File), R (Record), or P (Page)	Define the organization of the data
<b>MODE</b>	S (Stream), B (Block), or C (Compressed)	Define the transmission mode

Ref: TCP/IP Protocol Suite

# FTP – File Transfer Commands

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
<b>RETR</b>	File name(s)	Retrieve files; file(s) are transferred from server to the client
<b>STOR</b>	File name(s)	Store files; file(s) are transferred from the client to the server
<b>APPE</b>	File name(s)	Similar to STOR except if the file exists, data must be appended to it
<b>STOU</b>	File name(s)	Same as STOR except that the file name will be unique in the directory; however, the existing file should not be overwritten

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
<b>ALLO</b>	File name(s)	Allocate storage space for the files at the server
<b>REST</b>	File name(s)	Position the file marker at a specified data point
<b>STAT</b>	File name(s)	Return the status of files

Ref: TCP/IP Protocol Suite



thank you!

