

FTP

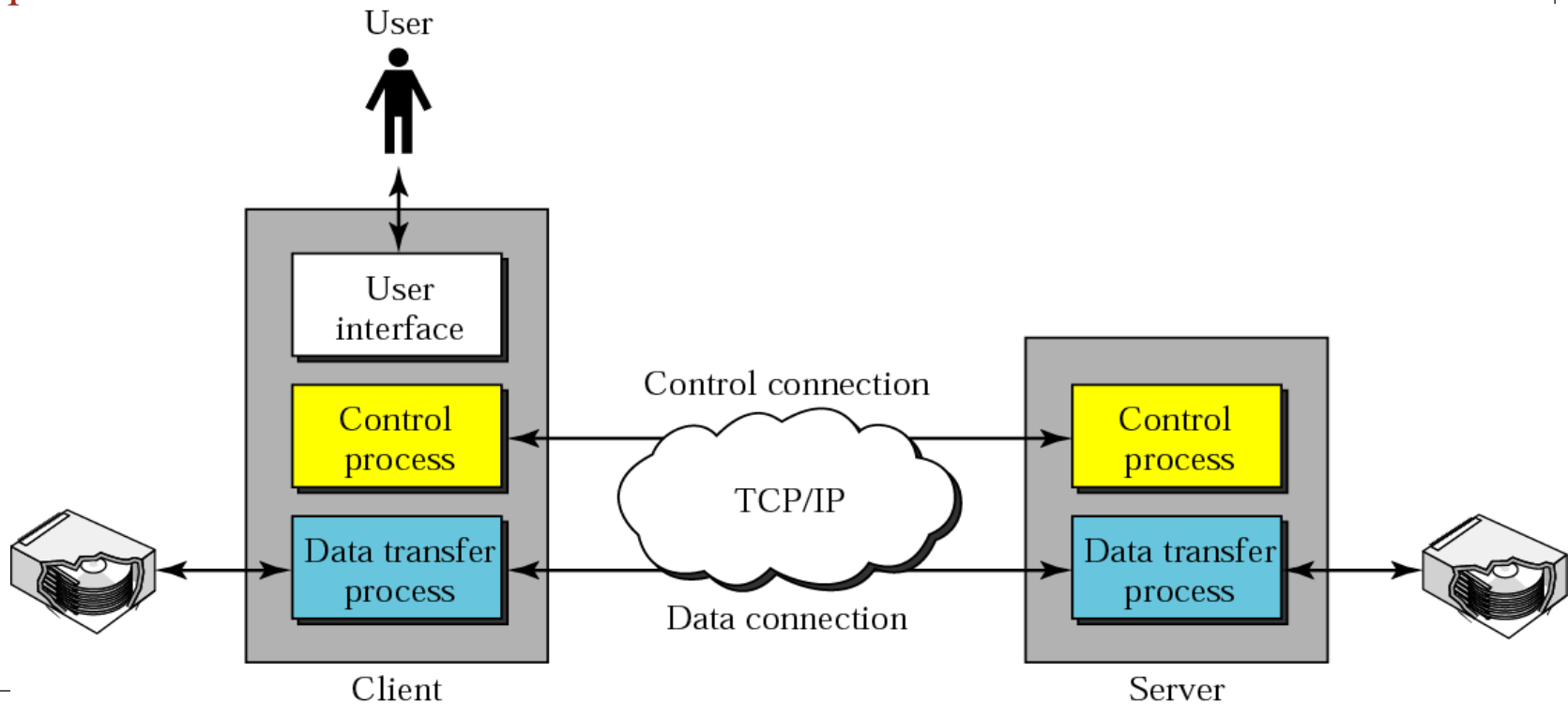
- It is a client-server application.
- FTP is a standard mechanism provided by TCP/IP for **copying a file from one host to another.**
- It establishes two connections between hosts
 - First for data transfer
 - Second for control information (commands and responses)
- Separation of commands and data transfer makes FTP more efficient
- Control connection uses very simple rules for communication, while data transfer uses complex rules due to the variety of data types transferred.

FTP uses the services of TCP. It needs two TCP connections.

The well-known port 21 is used for the control connection and the well-known port 20 is used for the data connection.

Basic Model of FTP

- Client has 3 components - user interface, client control process and client data transfer process and the Server has 2 components - server control process and server data transfer process.
- Control connection is made between the control processes and remains connected during the entire interactive FTP session.
- Data connection is made between the data transfer process and connection is opened and closed for each file transfer



Connections

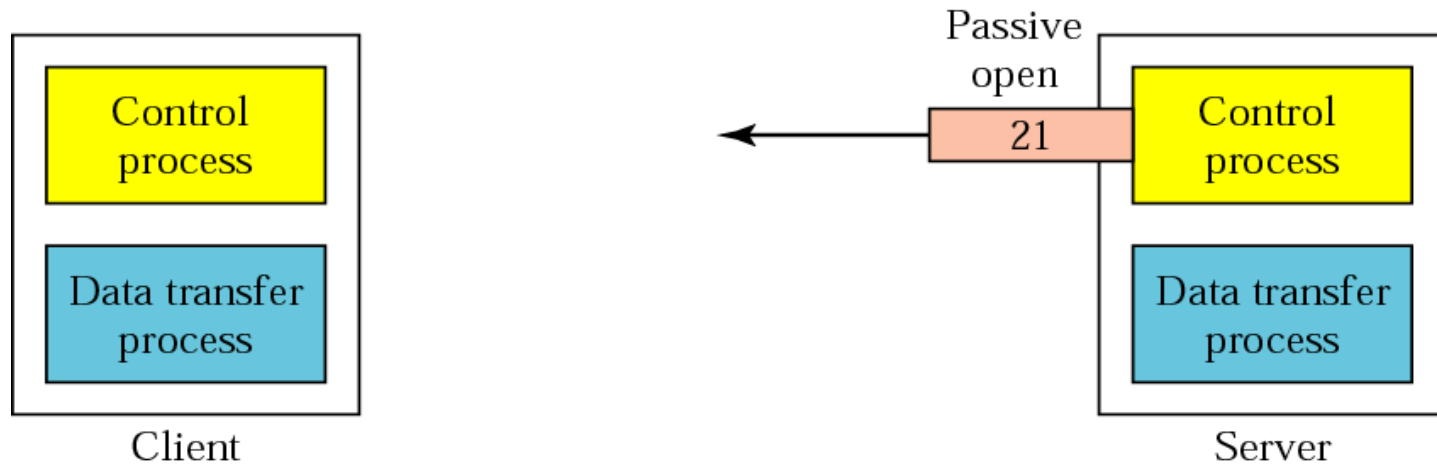
Two FTP connections, control and data, use different strategies and different port numbers.

1. Control Connection

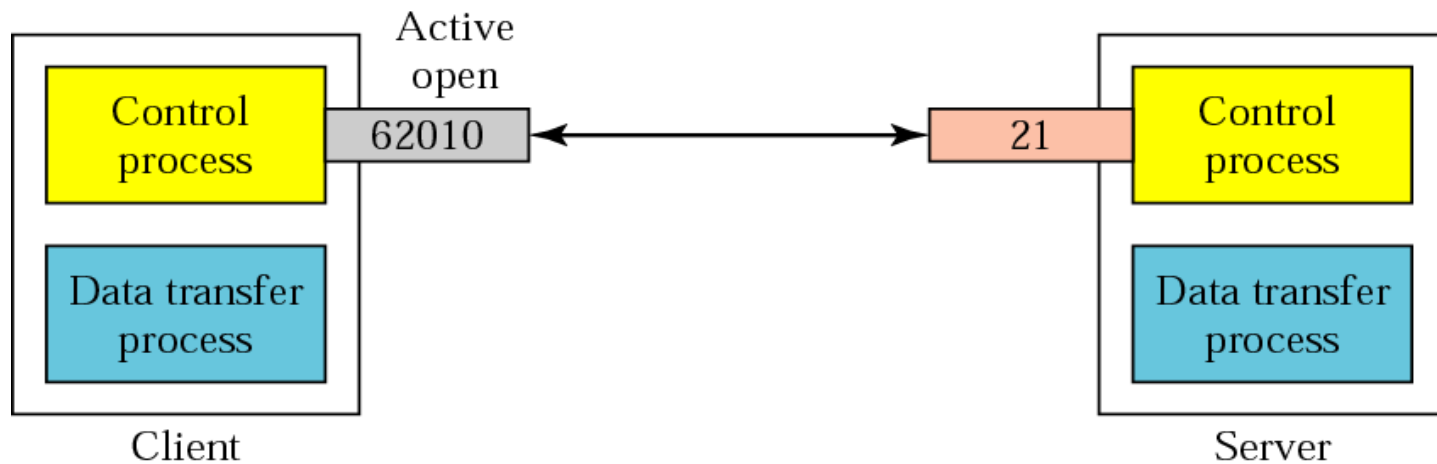
Two steps

1. Server issues a passive open on the well-known port 21 and wait for a client connection.
 2. Client uses an ephemeral port and issues an active open command
- Connection remains open during the entire process. The service type used by IP protocol is minimize delay because this is an interactive connection between a user and a server.

Opening the control connection



a. Passive open by server



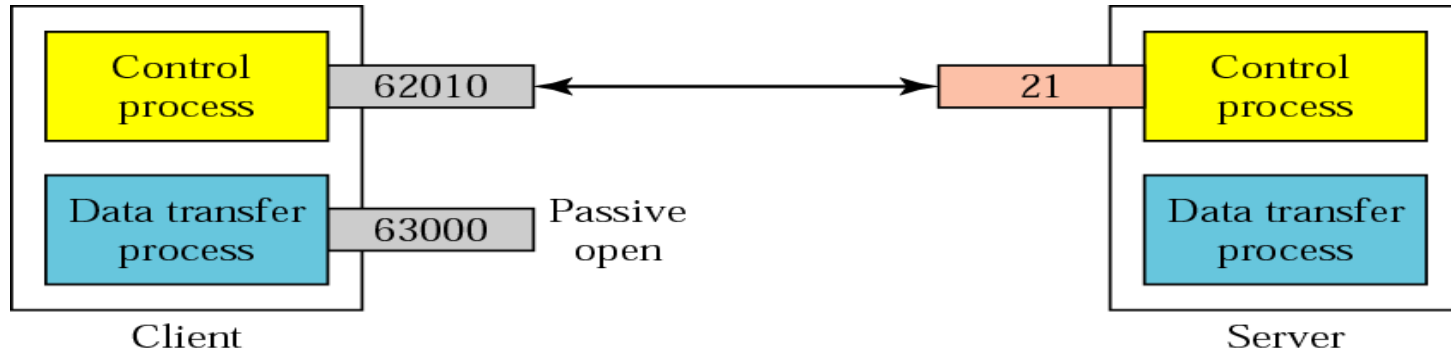
b. Active open by client

2. Data Connection

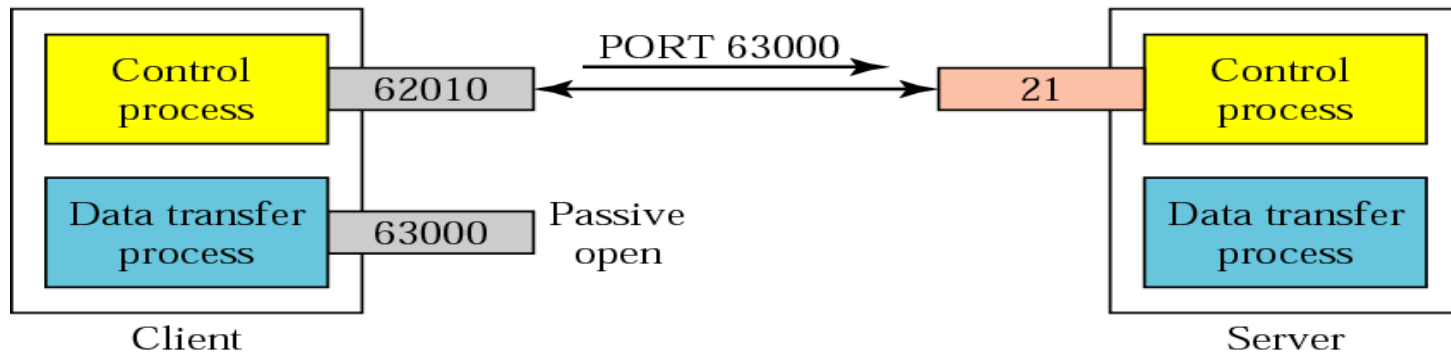
3 steps:-

1. Client issues a passive open using ephemeral port. Client initiates this connection, since client issues the command for file transfer
2. The client sends his port number to the server using the PORT command
3. The server receives the port number and issues an active open command using the well-known port number 20 and the received ephemeral port number

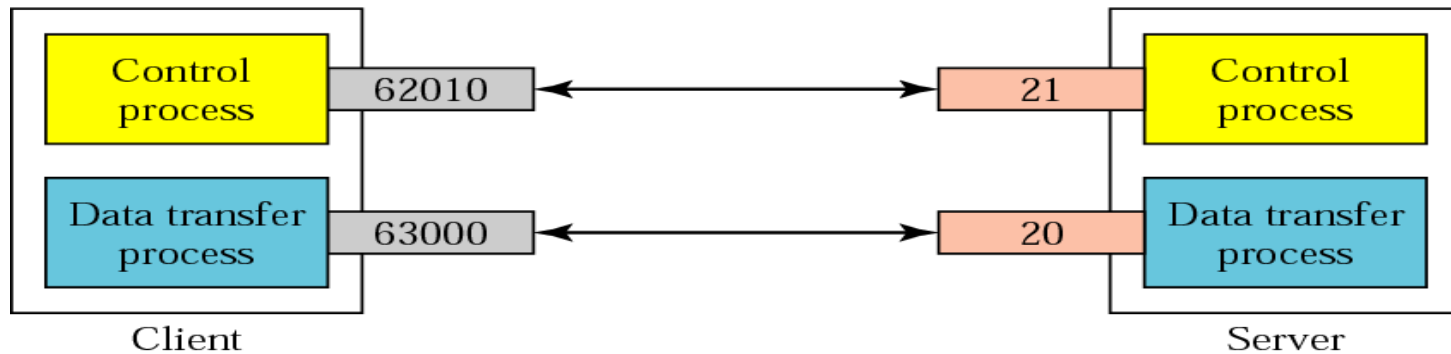
Creating the data connection



a. Passive open by client



b. Sending ephemeral port number to server



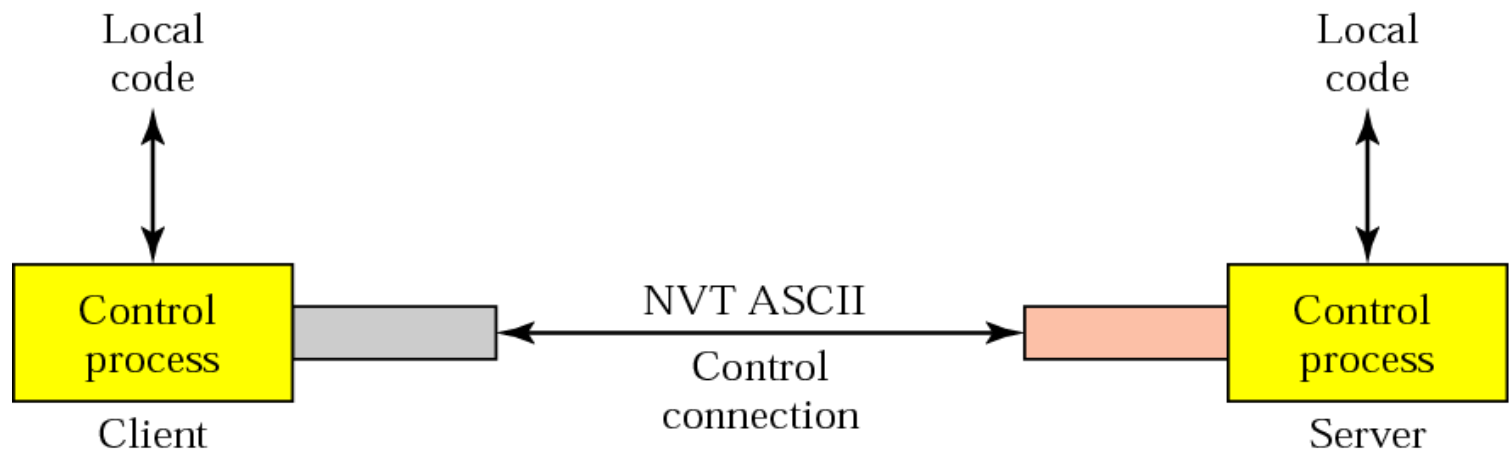
c. Active open by server

Communication

- FTP client and server runs on different type of computers an
- FTP uses two different approaches to resolve this problem
 - One for control connection
 - One for data connection

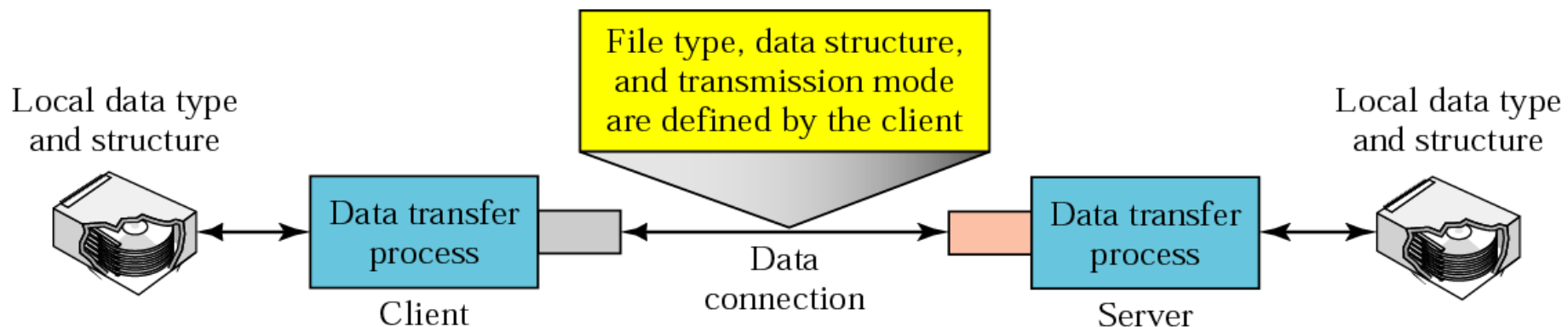
Communication over Control Connection

- It uses NVT ASCII (Network Virtual Terminal)character set
- Communication is achieved through commands and responses
- One command or response is issued at a time over this connection. Each command or response is only one short line and each line is terminated with two character (carriage return and line feed) end-of-line token.



Communication over Data Connection

- Files are transferred through data connection
- Client must define the transferring file's type, the structure of data, and transmission mode
- All these information will be transferred prior to transfer actual data



1. File Types

- **ASCII file:** Default format for transferring text files. Each character is encoded using NVT ASCII.
- **EBCDIC file:** if one or both ends of the connection use EBCDIC encoding, the file can be transferred using EBCDIC encoding.
- **Image file:** Default format for transferring binary files. File will sent as continuous stream of bits without any interpretation or encoding

- ASCII and EBCDIC file type uses another attribute to define the printability of file

- a. Nonprint

- Default format for transferring text file
 - Format is used for files that will be stored and processed later

- b. TELNET

- Format of the file contains NVT ASCII vertical characters such as CR, LF, NL, VT(Vertical Tab). The file is printable after transfer.

2. Data Structure

- **File Structure (Default)** – The file has no structure. It contains continuous stream of bytes
- **Record Structure** - File is divided into records. This can be used with text files.
- **Page Structure** - File is divided into pages. Each page will have a page number and page header. Pages can be stored and accessed randomly or sequentially

3. Transmission Mode

- **Stream Mode (Default mode)** - Data is delivered from FTP to TCP as continuous stream of bytes. TCP is responsible for chopping data into segments of appropriate size. No end of file information (file structure), hence closing of data connection is end of file. In record structure, each record will have a 1byte end-of-record character and 1byte end-of-file character at the end of file
- **Block Mode** - Data can be delivered from FTP to TCP in blocks. Each block will preceded by 3byte header. First byte is Block description and next two bytes define the size of block in bytes
- **Compressed Mode** - Compressed when file is big. Use run-length encoding for compression

File Transfer

- File transfer occurs over data connection under the control of the commands sent over control connection
- File transfer can be any of the three things
 - File is to be copied from server to client. It is called **retrieving a file**. It is done under the supervision of **RETR** command
 - File is to be copied from client to server. It is called **storing a file**. It is done by using **STORE** command
 - A list of directory or file names is to be sent from server to client. It is done using **LIST** command. (List of directory or filenames are treated as a file)

File Transfer

