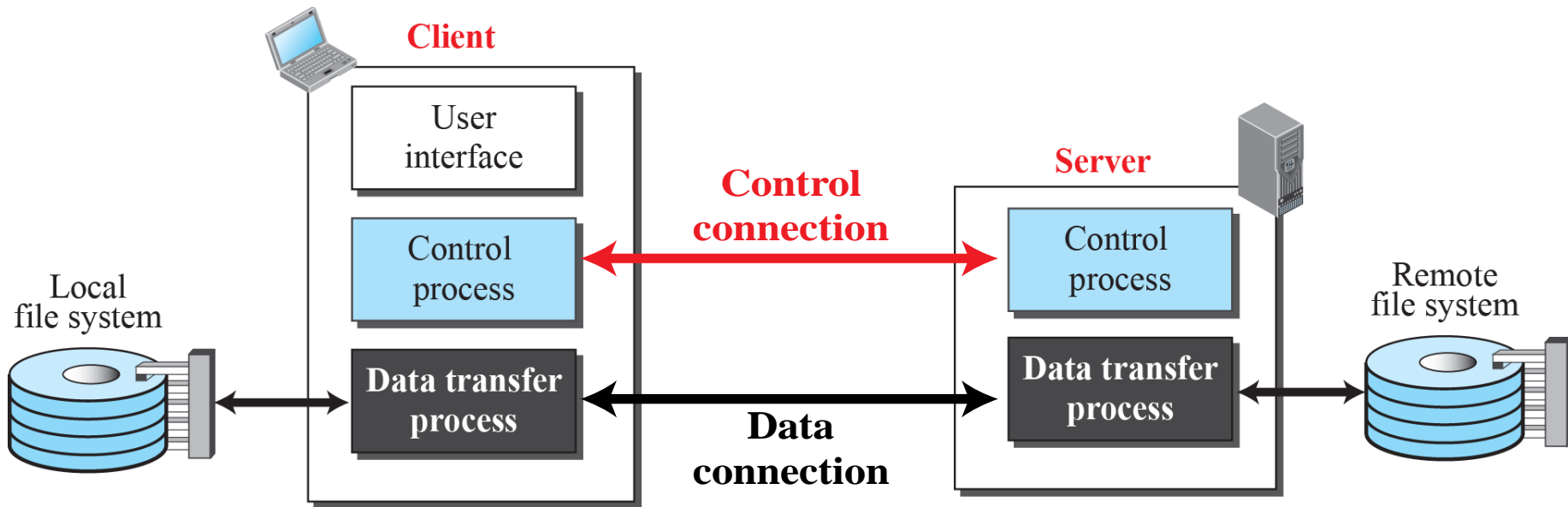


FTP

Purpose:

- File Transfer Protocol (FTP) is the standard protocol provided by TCP/IP for copying a file from one host to another.
- Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first.
- ✓ Two systems may use different file name conventions.
- ✓ Two systems may have different ways to represent data.
- ✓ Two systems may have different directory structures.
- we can transfer files using HTTP, FTP is a better choice to transfer large files or to transfer files using different formats.

FTP



basic model of FTP. The client has three components: the user interface, the client control process, and the client data transfer process.

The server has two components: the server control process and the server data transfer process.

The control connection is made between the control processes.

The data connection is made between the data transfer processes.



The two connections in FTP have different lifetimes.

- 1. The control connection remains **connected during the entire interactive FTP session.***
- 2. The data connection is **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

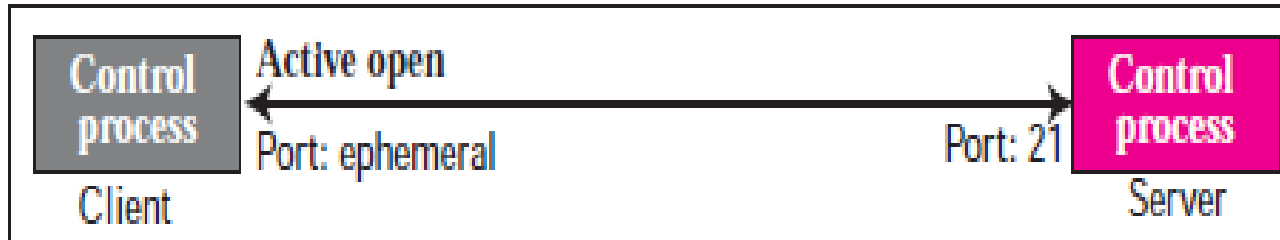
There are two steps:

1. The server issues a passive open on the well-known port 21 and waits for a client.
 2. The client uses an ephemeral port and issues an active open.
- The connection remains open during the entire process.
 - The service type, used by the IP protocol, is *minimize delay because this is an interactive connection between a user (human) and a server.*
 - The user types commands and expects to receive responses without significant delay.

Opening the control connection



a. First, passive open by server



b. Later, active open by client

Commands, which are sent from the FTP client control process, are in the form of ASCII uppercase, which may or may not be followed by an argument.

| <i>Command</i> | <i>Argument(s)</i> | <i>Description</i> |
|----------------|--------------------|---|
| 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) |

Some responses in FTP

Every FTP command generates at least one response. A response has two parts: three-digit number followed by text.

The numeric part defines the code; the text part defines needed parameters or further explanations.

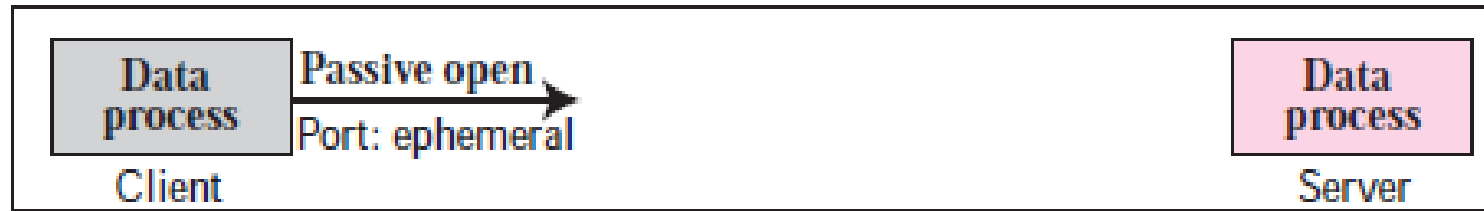
| <i>Code</i> | <i>Description</i> | <i>Code</i> | <i>Description</i> |
|-------------|-------------------------|-------------|---|
| 125 | Data connection open | 250 | Request file action OK |
| 150 | File status OK | 331 | User name OK; password is needed |
| 200 | Command OK | 425 | Cannot open data 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 |



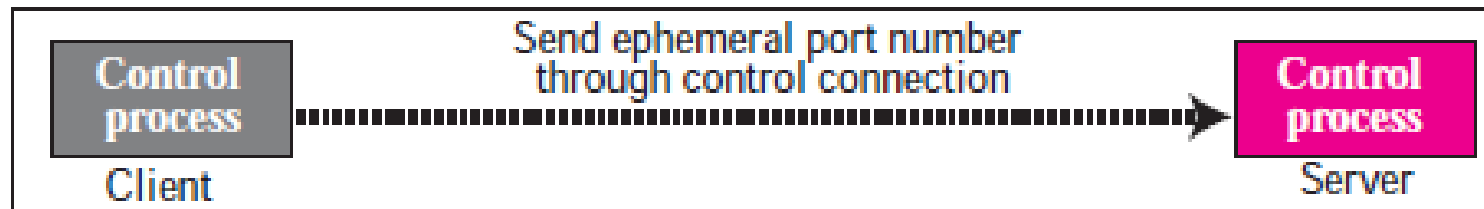
The data connection uses the well-known port 20 at the server site. However, the creation of a data connection is different from the control connection. The following shows the steps:

1. The client, not the server, issues a passive open using an ephemeral port.
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.

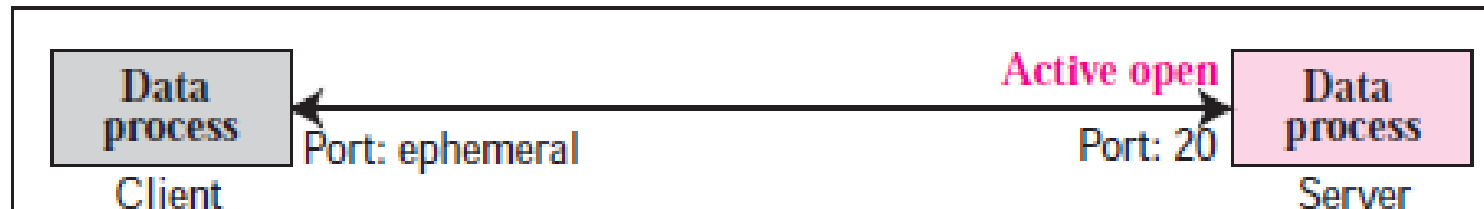
Creating the data connection



a. First, passive open by client



b. Second, sending of ephemeral port

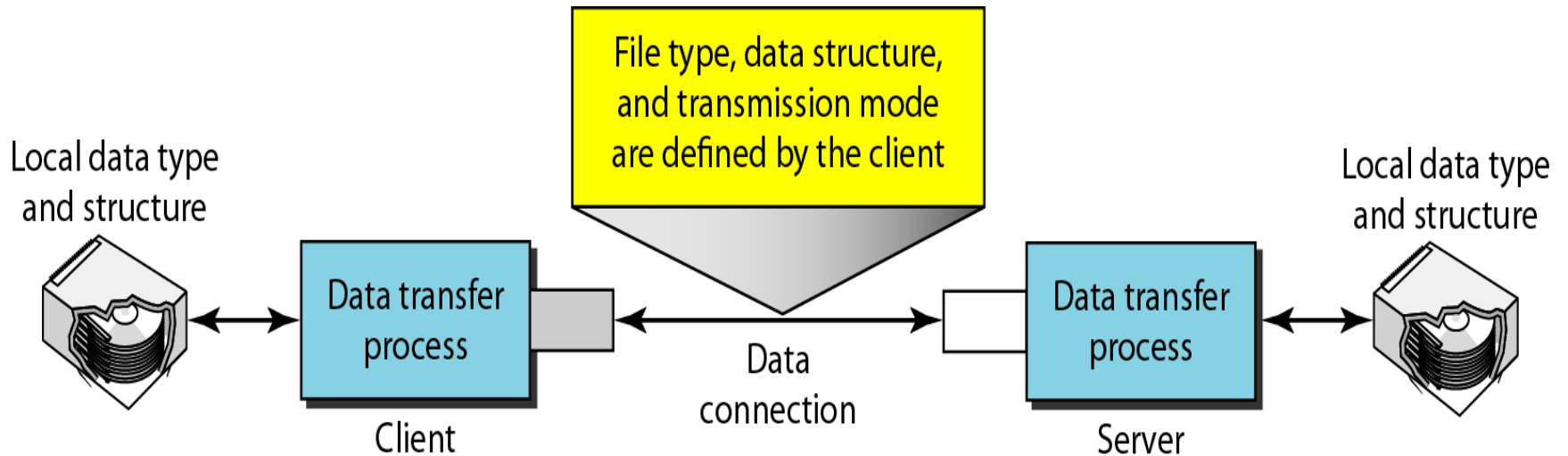


c. Third, active open by server

Communication over Data Connection

- File transfer occurs over the data connection under the control of the commands sent over the control connection.
- We prepare for transmission through the control connection.
- The heterogeneity problem is resolved by defining three attributes of communication before sending the file through the data connection :
 - file type
 - data structure
 - transmission mode

Using the data connection



File Type

- FTP can transfer one of the following file types across the data connection:
 - ASCII file.
 - EBCDIC file
 - image file.
- **The ASCII file** is the default format for transferring text files.
 - Each character is encoded using 7-bit ASCII.
 - The sender transforms the file from its own representation into ASCII characters, and the receiver transforms the ASCII characters to its own representation.
- **The EBCDIC file** used If one or both ends of the connection use EBCDIC encoding (the file format used by IBM).
- **The image file** is the default format for transferring binary files.
 - The file is sent as continuous streams of bits without any interpretation or encoding.
 - This is mostly used to transfer binary files such as compiled programs.

Data Structure

- FTP can transfer a file across the data connection by using one of the following interpretations about the structure of the data:
 - **file structure**: the file is a continuous stream of bytes.
 - **record structure**: the file is divided into records. This can be used only with text files.
 - **page structure**: the file is divided into pages, with each page having a page number and a page header.
 - The pages can be stored and accessed randomly or sequentially.

Transmission Mode

- FTP can transfer a file across the data connection using one of the following three transmission modes:
 - Stream mode.
 - Block mode.
 - Compressed mode.

Stream mode:

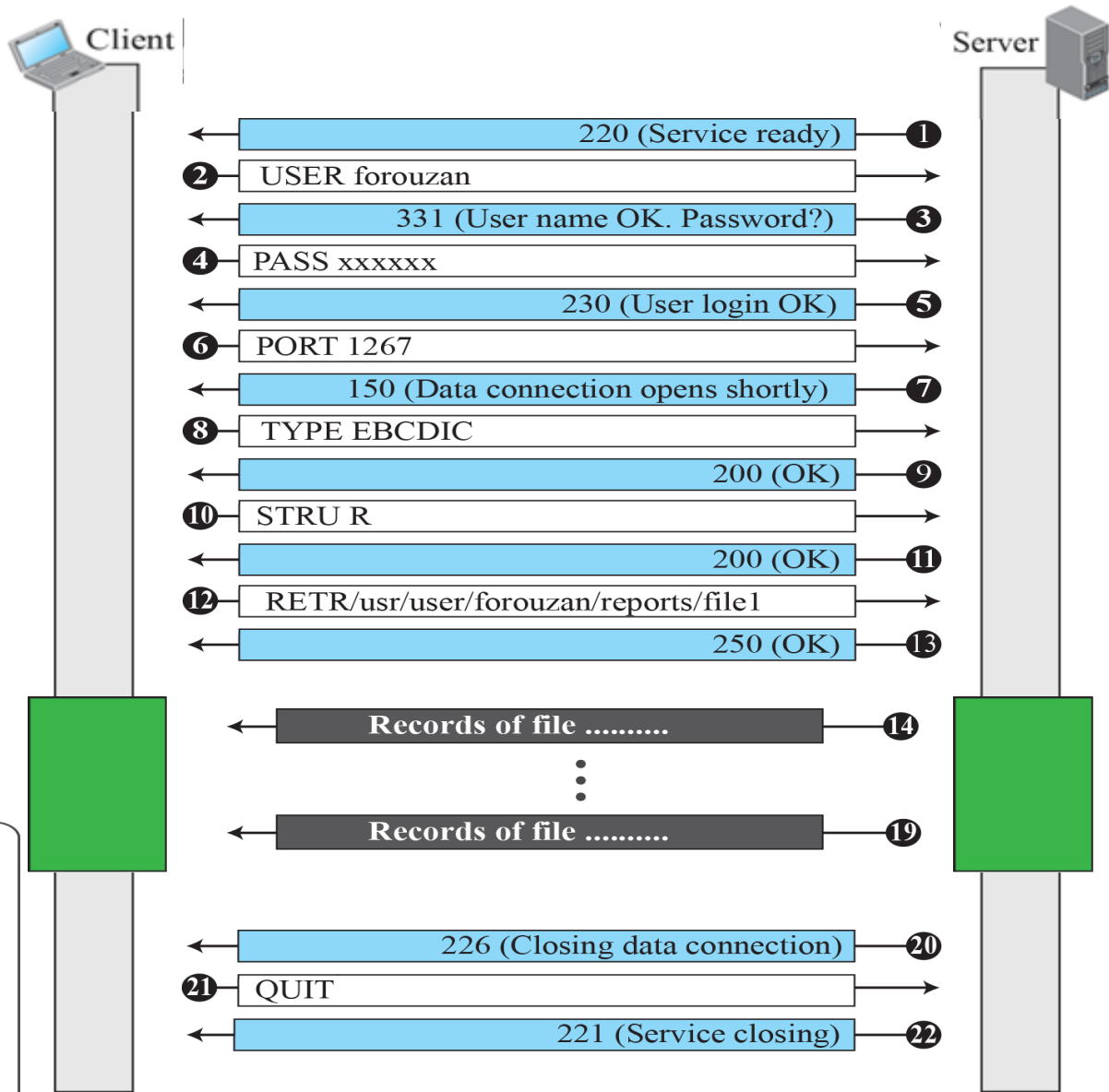
- This is the default mode. Data are delivered from FTP to TCP as a continuous stream of bytes.
- TCP is responsible for chopping data into segments of appropriate size.
- If the data is simply a stream of bytes (file structure), no end-of-file is needed.
- End-of-file in this case is the closing of the data connection by the sender.
- If the data are divided into records (record structure), each record will have a 1-byte end-of-record (EOR) character and the end of the file will have a 1-byte end-of-file (EOF) character.

Example of using FTP for retrieving a file.

The figure shows only one file to be transferred. The control connection remains open all the time, but the data connection is opened and closed repeatedly. We assume the file is transferred in six sections. After all records have been transferred, the server control process announces that the file transfer is done. Since the client control process has no file to retrieve, it issues the QUIT command, which causes the service connection to be closed.

Legend

- Control process (port 21)
- Data transfer process (port 20)
- Command
- Response
- Data transfer



File Transfer

- File transfer occurs over the data connection under the control of the commands sent over the control connection.

However, we should remember that file transfer in FTP means one of three things:

1. *retrieving a file (server to client),*
2. *storing a file (client to server), and*
3. *directory listing (server to client).*

The following shows an actual FTP session that lists the directories.

```

$ 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.

```



The FTP protocol was designed when security was not a big issue. Although FTP requires 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.