Chapter 5

Application Layer

- INTRODUCTION
- CLIENT-SERVER PARADIGM
- STANDARD APPLICATIONS
  - HTTP
  - FTP
  - SMTP
  - Telnet
  - SSH
  - DNS

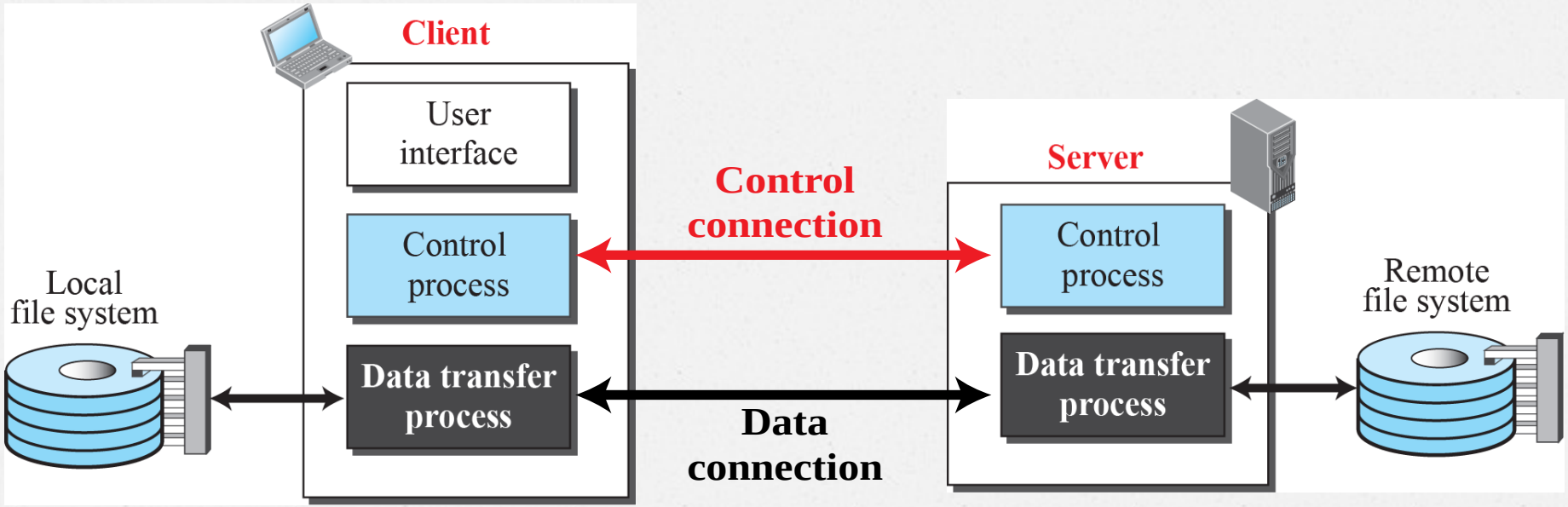
File Transfer Protocol (FTP)

- Components of FTP Client & Server
- Connections
  - Control connection
    - Commands
      - Data structure
      - File type
      - Transmission mode
      - File transfer
    - Responses
  - Data connection
- FTP Security

File Transfer Protocol(FTP)

- Standard protocol provided by TCP/IP for copying a file from one host to another.
- Connection Oriented Protocol
- Though HTTP can also transfer files FTP is ideal protocol and better choice for large files.
- Several issues have to be handled by FTP, for example, two systems may use different file name conventions, different ways to represent data, different directory structures.
- Components of FTP Client: user interface, client control process, and the client data transfer process.
- Components of FTP Server: the server control process and the server data transfer process

FTP



FTP uses two well-known TCP ports:

- port 21 is used for the control connection, and
- port 20 is used for the data connection.

Lifetimes of Two Connections

- FTP Uses Two Connections
  - The control connection is made between the control processes.
  - The data connection is made between the data transfer processes.
- Separation of commands and data transfer makes FTP more efficient
- Two connections in FTP have different lifetimes
  - The control connection remains connected during the entire interactive FTP session.
  - The data connection is opened and then closed for each file transfer activity.
- When a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.

Control Connection

- Port-21
- It uses the NVT ASCII character set
- **commands** are sent from the client to the server.(Only one command)
- **responses** are sent from the server to the client.(Only one command)
- 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
- Each line is terminated with a two-character (carriage return and line feed) end-of-line token
- Every FTP command generates at least one response
- We want to transfer files through the data connection. The **client must define the type of file to be transferred**, the structure of the data, and the transmission mode using commands in control connection.

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

Some FTP commands (continued)

Command	*Argument(s)*	*Description*
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

Responses in FTP

- A response has two parts: **a three-digit number** followed by **text.**
  - The numeric part defines the code
    - The first digit defines the status of the command. (i.e. good, bad, incomplete)
    - The second digit defines the area in which the status applies. (i.e. syntax, file system, authentication, connection..)
    - The third digit provides additional information.
  - The text part defines needed parameters or further explanations

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 Structure

Three kinds of interpretations

- The **file structure** format (used by default) has no structure. It is a continuous stream of bytes.
- In the **record structure**, the file is divided into records. This can be used only with text files.
- In the **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

File Type

FTP can transfer one of the following file types across the data connection: ASCII file, EBCDIC file, or image file.

Transmission Mode

Three kinds of modes

- The **stream mode** is the default mode; data are delivered from FTP to TCP as a continuous stream of bytes..
- The **block mode**, data can be delivered from FTP to TCP in blocks. In this case, each block is preceded by a 3-byte header. The first byte is called the block descriptor; the next two bytes define the size of the block in bytes.
- In the **compressed mode**, the file is compressed and the file is delivered from FTP to TCP as a continuous stream of bytes

File Transfer

File transfer in FTP means one of three things:

- retrieving a file (server to client),
- storing a file (client to server), and
- directory listing (server to client).

Data Connection

- Port 20
- Steps in data connection
  1. The client, not the server, issues a passive open using an ephemeral port. This must be done by the client because it is the client that issues the commands for transferring files.
  2. The client sends this port number to the server using the PORT command.
  3. The server receives the port number and issues an active open using the wellknown port 20 and the received ephemeral port number.

FTP Connection Modes

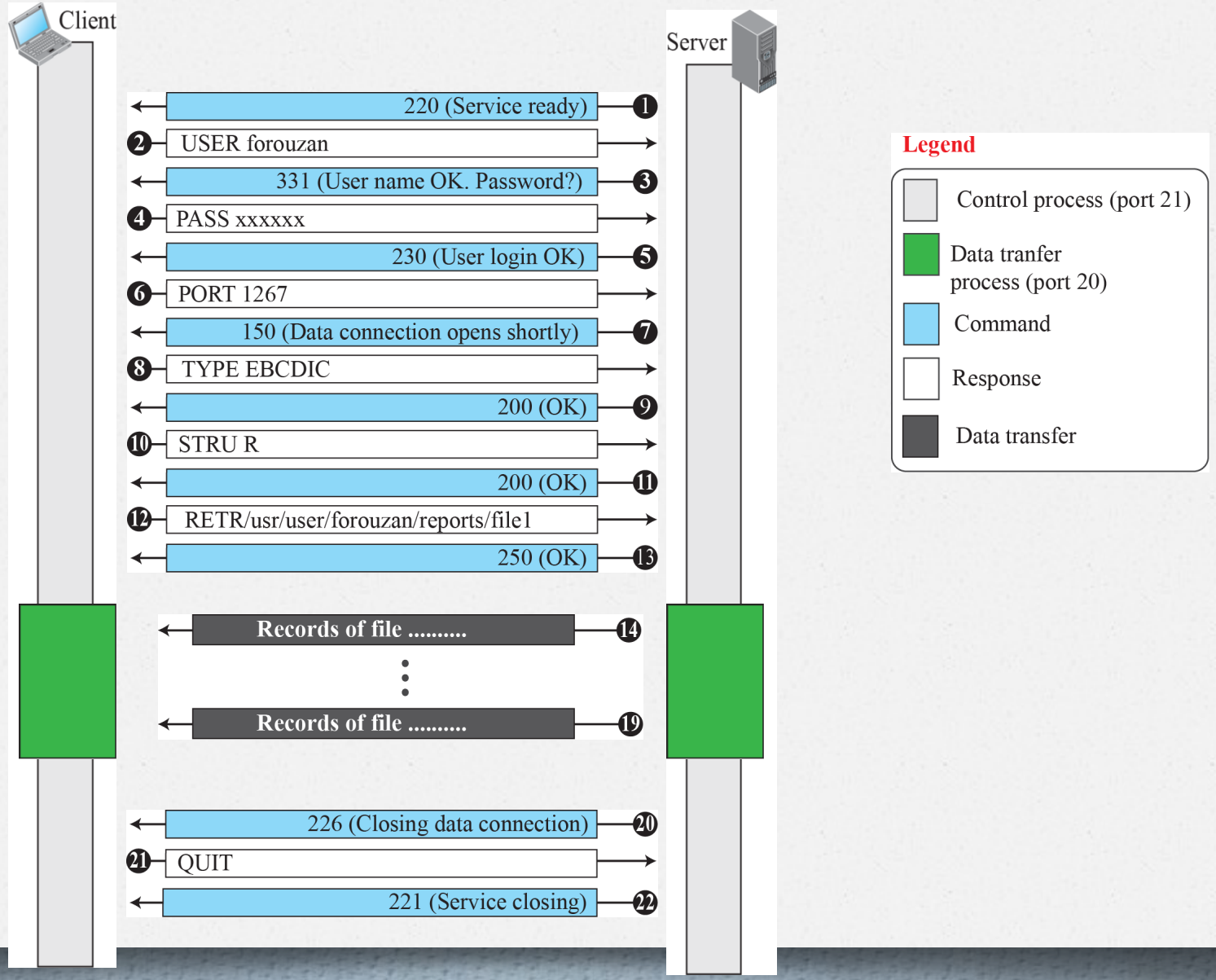
FTP may operate in an active or a passive mode, which determines how a data connection is established. In both cases, a client creates a TCP control connection to an FTP server command port 21.

- In the ***active* mode**, the client starts listening on a random port for incoming data connections from the server (the client sends the FTP command PORT to inform the server on which port it is listening).
- In the ***passive* mode**, the client uses the control connection to send a PASV command to the server and then receives a server IP address and server port number from the server, which the client then uses to open a data connection to the server IP address and server port number received.

Example

An 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.

Example


Client
Server
220 (Service ready)
USER forouzan
331 (User name OK. Password?)
PASS xxxxxx
230 (User login OK)
PORT 1267
150 (Data connection opens shortly)
TYPE EBCDIC
200 (OK)
STRU R
200 (OK)
RETR/usr/user/forouzan/reports/file1
250 (OK)
Records of file .....
Records of file .....
226 (Closing data connection)
QUIT
221 (Service closing)
Legend
Control process (port 21)
Data tranfer process (port 20)
Command
Response
Data transfer

Example

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.
```

Security for FTP

- 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.
- one can add a Secure Socket Layer between the FTP application layer and the TCP layer. In this case FTP is called SSL-FTP.