

DRONACHARYA
College of Engineering

Computer Science & Engineering

Data Communication and Computer
Networks

(MTCSE-101-A)

OBJECTIVES:

- ❑ **To discuss FTP and two connections used in this protocol: control connection and data connection.**
- ❑ **To discuss six classes of commands sent by the client to establish communication with the server.**
- ❑ **To explain three types of file transfer transferred by FTP.**
- ❑ **To show some user-friendly commands used by some FTP interfaces.**
- ❑ **To discuss anonymous FTP and its application.**
- ❑ **To discuss how file transfer can be done using a secure channel.**
- ❑ **To discuss TFTP as a simple file transfer protocol without the complexities and sophistication of FTP.**

OBJECTIVES (*continued*):

- ❑ To discuss five types of TFTP messages and their applications.
- ❑ To discuss the sorcerer's apprentice bug related to TFTP's flow- and error-control mechanisms.
- ❑ To show how TFTP can be used in conjunction with DHCP to initialize devices by downloading configuration files.

Chapter Outline

21.1 ***FTP***

21.2 ***TFTP***

21-1 FTP

File Transfer Protocol (FTP) is the standard mechanism 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. For example, two systems may use different file name conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures. All of these problems have been solved by FTP in a very simple and elegant approach.

Topics Discussed in the Section

- ✓ **Connections**
- ✓ **Communication**
- ✓ **Command Processing**
- ✓ **File Transfer**
- ✓ **Anonymous FTP**
- ✓ **Security for FTP**
- ✓ **The sftp Program**



Note

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 for the data connection.

Figure 21.1 *FTP*

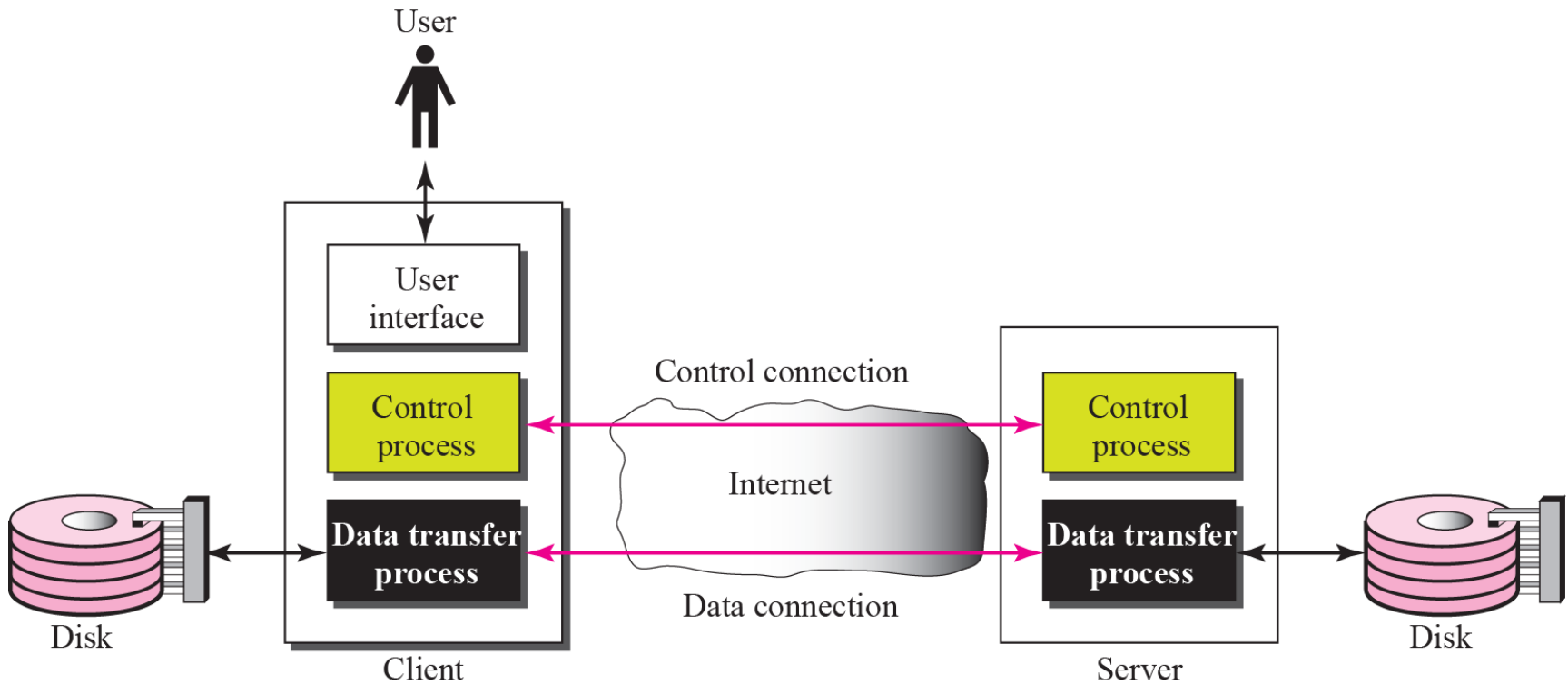
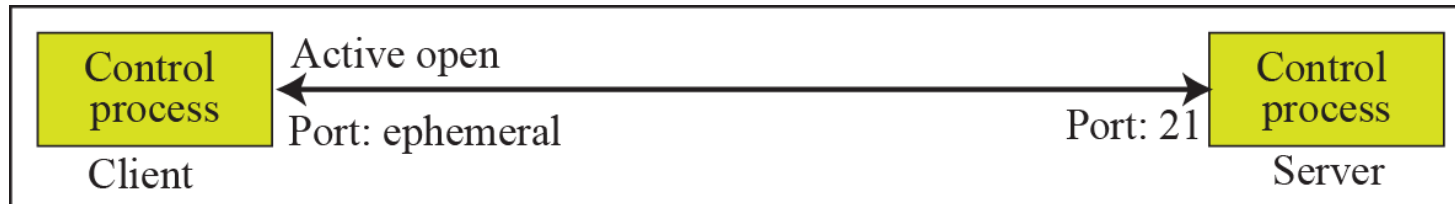


Figure 21.2 *Opening the control connection*



a. First, passive open by server

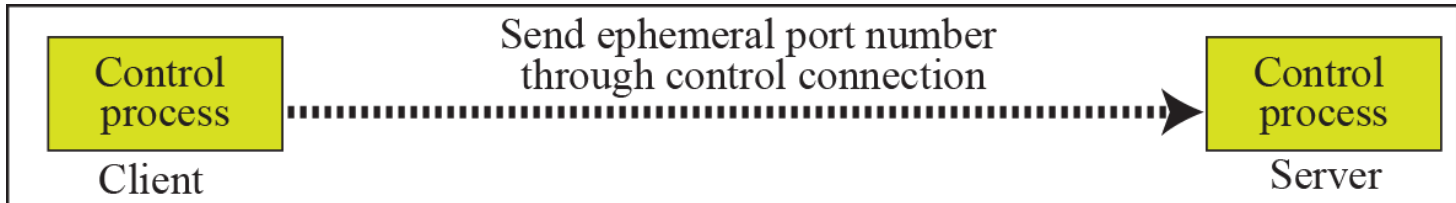


b. Later, active open by client

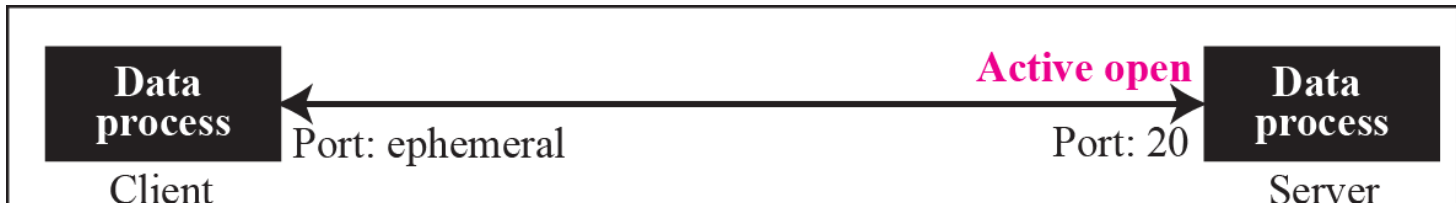
Figure 21.3 *Creating the data connection*



a. First, passive open by client



b. Second, sending of ephemeral port



c. Third, active open by server

Figure 21.4 *Using the control connection*

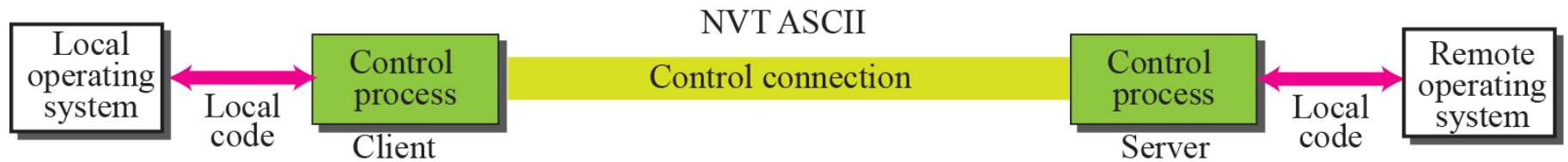


Figure 21.5 *Using the data connection*

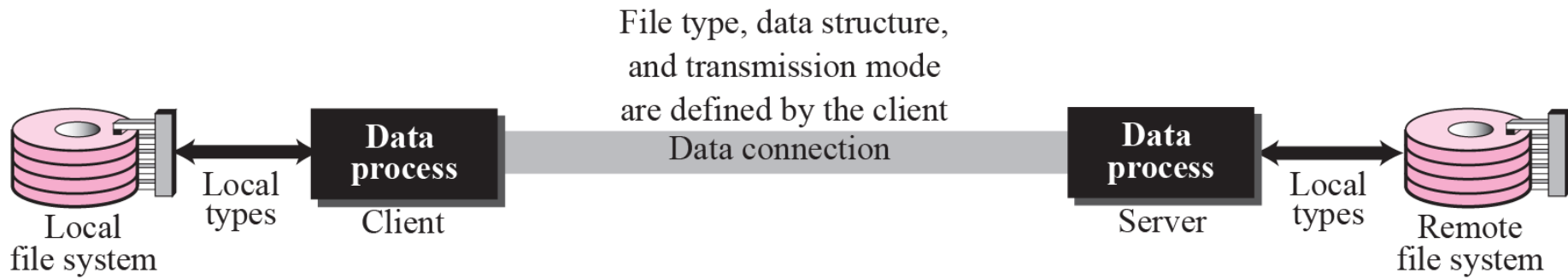
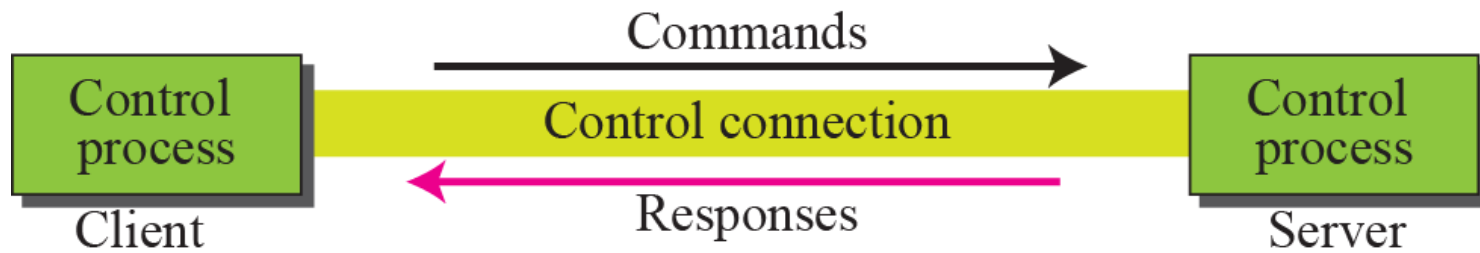


Figure 21.6 *Command processing*



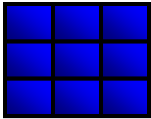


Table 21.1 *Access commands*

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

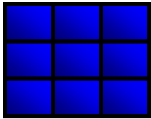


Table 21.2 *File management commands*

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

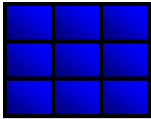


Table 21.3 *Data formatting commands*

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

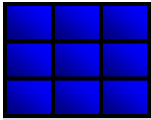


Table 21.4 *Port defining commands*

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
PORT	6-digit identifier	Client chooses a port
PASV		Server chooses a port

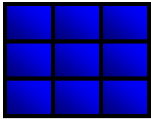


Table 21.5 *File transfer commands*

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
RETR	File name(s)	Retrieve files; file(s) are transferred from server to client
STOR	File name(s)	Store files; file(s) are transferred from client to server
APPE	File name(s)	Similar to STOR, but if file exists, data must be appended to it
STOU	File name(s)	Same as STOR, but file name will be unique in the directory
ALLO	File name(s)	Allocate storage space for files at the server
REST	File name(s)	Position file marker at a specified data point
STAT	File name(s)	Return status of files

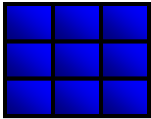


Table 21.6 *Miscellaneous commands*

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
HELP		Ask information about the server
NOOP		Check if server is alive
SITE	Commands	Specify the site-specific commands
SYST		Ask about operating system used by the server

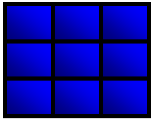


Table 21.7 *Responses*

<i>Code</i>	<i>Description</i>
Positive Preliminary Reply	
120	Service will be ready shortly
125	Data connection open; data transfer will start shortly
150	File status is OK; data connection will be open shortly
Positive Completion Reply	
200	Command OK
211	System status or help reply
212	Directory status
213	File status
214	Help message
215	Naming the system type (operating system)
220	Service ready
221	Service closing
225	Data connection open
226	Closing data connection
227	Entering passive mode; server sends its IP address and port number
230	User login OK
250	Request file action OK
Positive Intermediate Reply	
331	User name OK; password is needed
332	Need account for logging
350	The file action is pending; more information needed

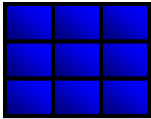
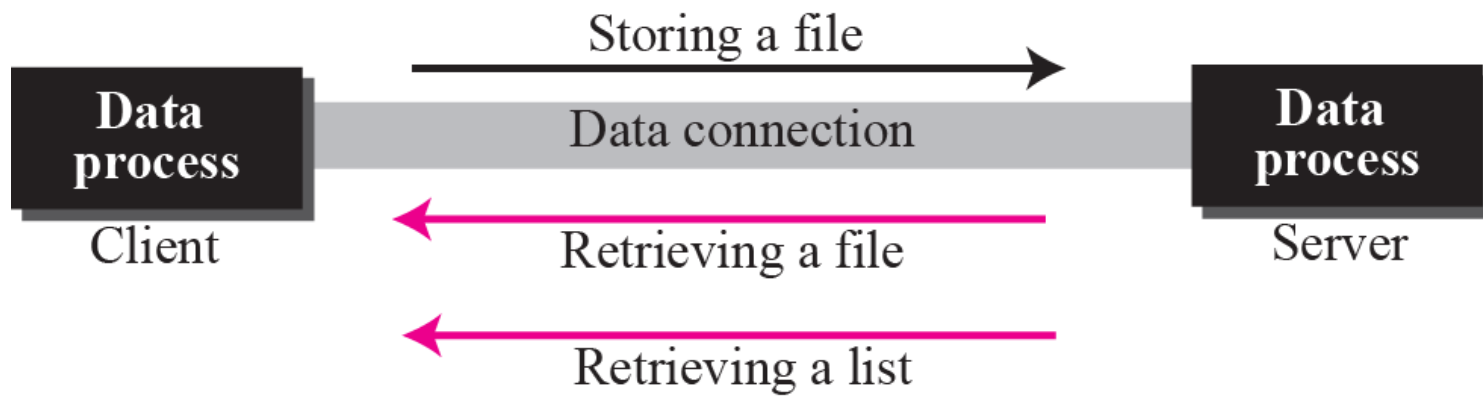


Table 21.7 *Responses (continued)*

<i>Code</i>	<i>Description</i>
Transient Negative Completion Reply	
425	Cannot open data connection
426	Connection closed; transfer aborted
450	File action not taken; file not available
451	Action aborted; local error
452	Action aborted; insufficient storage
Permanent Negative Completion Reply	
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command parameter not implemented
530	User not logged in
532	Need account for storing file
550	Action is not done; file unavailable
552	Requested action aborted; exceeded storage allocation
553	Requested action not taken; file name not allowed

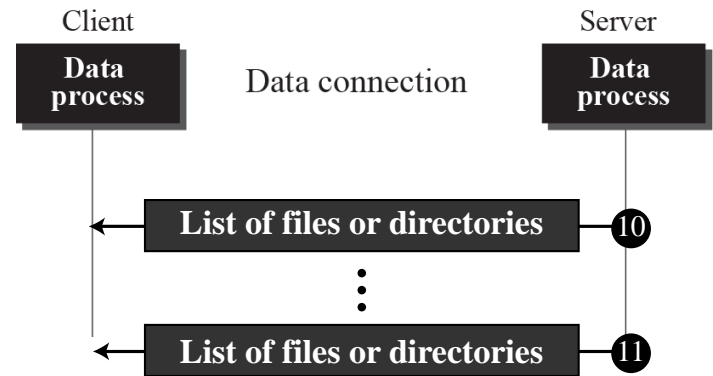
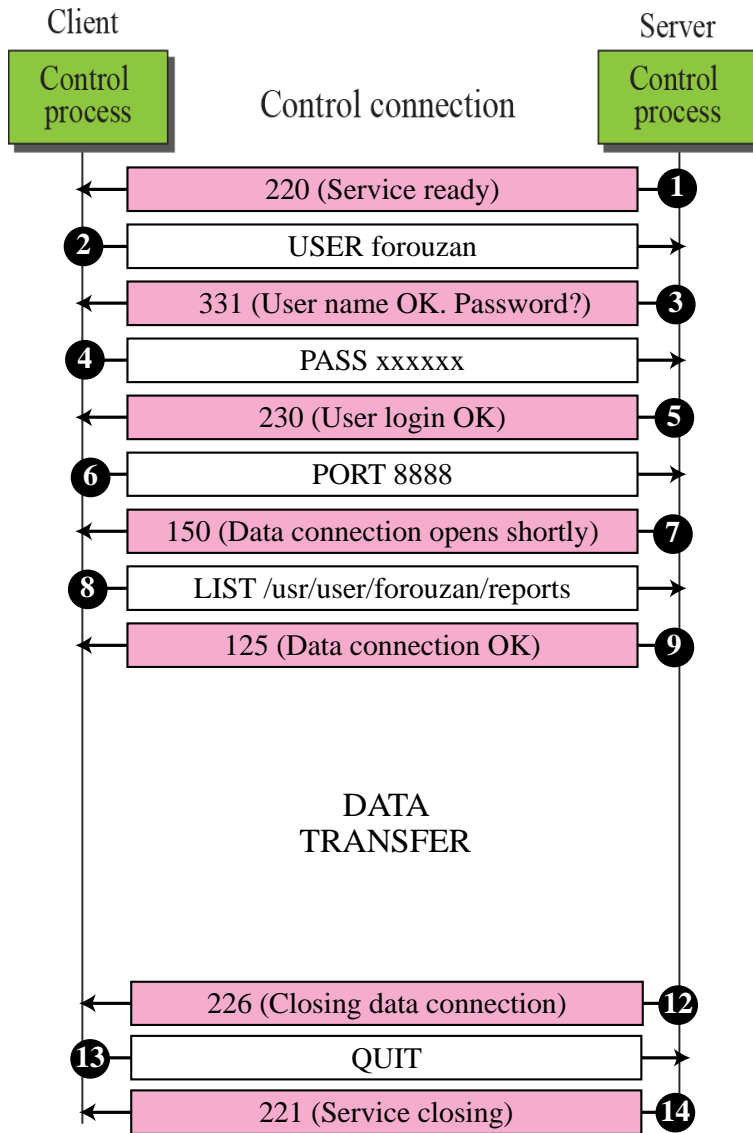
Figure 21.7 *File transfer*



Example 21.1

Figure 21.8 shows an example of using FTP for retrieving a list of items in a directory.

Figure 21.8 *Example 21.1*



Example 21.2

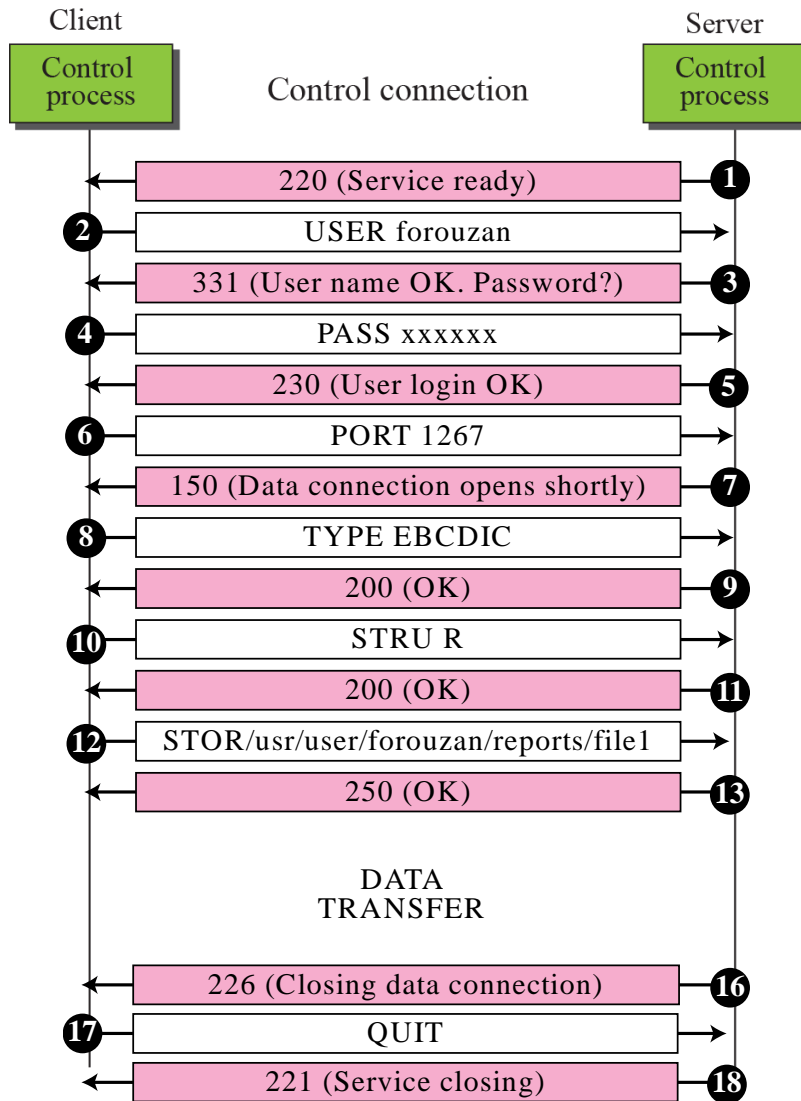
The following shows an actual FTP session that parallels Example 21.1. The colored lines show the responses from the server control connection; the black lines show the commands sent by the client. The lines in white with black background show data transfer.

```
$ 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.
ftp> ls reports
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.
```

Example 21.3

Figure 21.9 shows an example of how an image (binary) file is stored.

Figure 21.9 *Example 21.3*



Example 21.4

We show an example of anonymous FTP. We assume that some public data are available at internic.net.

```
$ ftp internic.net
Connected to internic.net
220 Server ready
Name: anonymous
331 Guest login OK, send "guest" as password
Password: guest
ftp > pwd
257 '/' is current directory
ftp > ls
200 OK
150 Opening ASCII mode
```

```
bin
```

```
. . .
. . .
. . .
```

```
ftp > close
221 Goodbye
ftp > quit
```

21-2 TFTP

There are occasions when we need to simply copy a file without the need for all of the features of the FTP protocol. For example, when a diskless workstation or a router is booted, we need to download the bootstrap and configuration files. Here we do not need all of the sophistication provided in FTP. We just need a protocol that quickly copies the files.

Topics Discussed in the Section

- ✓ **Messages**
- ✓ **Connection**
- ✓ **Data Transfer**
- ✓ **UDP Ports**
- ✓ **TFTP Example**
- ✓ **TFTP Options**
- ✓ **Security**
- ✓ **Applications**



Note

TFTP uses the services of UDP on the well-known port 69.

Figure 21.10 *Message categories*

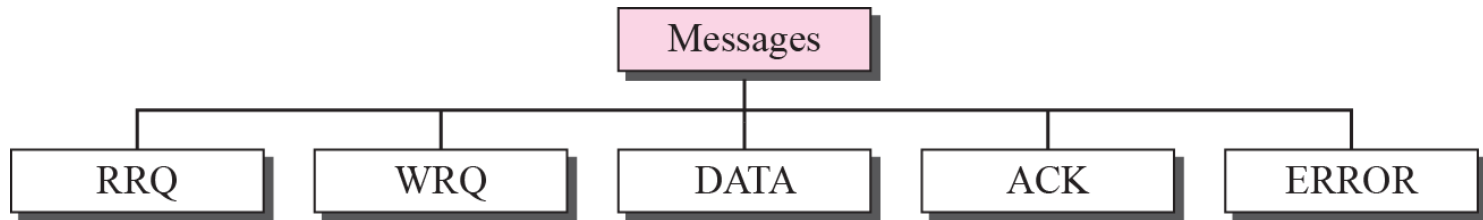


Figure 21.11 *RRQ format*



Figure 21.12 *WRQ format*



Figure 21.13 *Data Format*

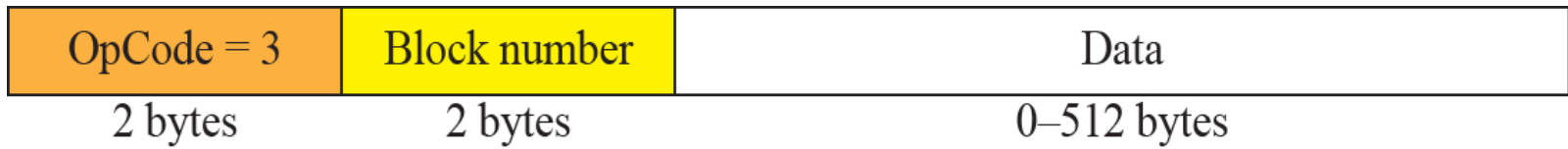


Figure 21.14 *ACK format*

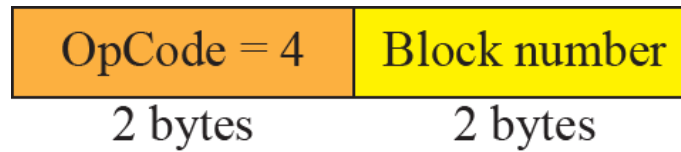
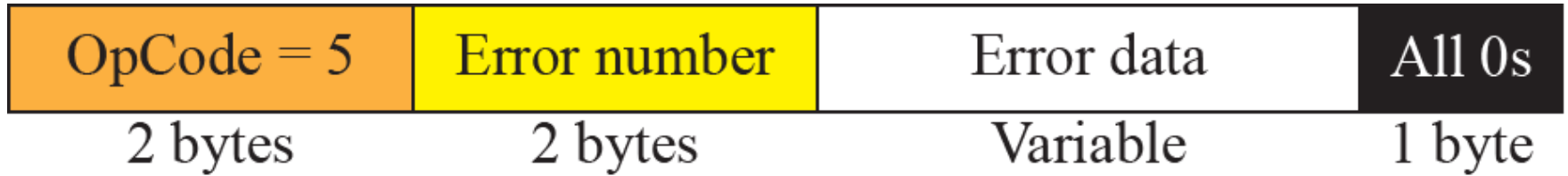


Figure 21.15 *ERROR format*



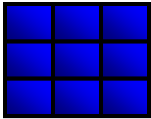


Table 21.8 *Error numbers and their meanings*

<i>Number</i>	<i>Meaning</i>	<i>Number</i>	<i>Meaning</i>
0	Not defined	5	Unknown port number
1	File not found	6	File already exists
2	Access violation	7	No such user
3	Disk full or quota exceeded		
4	Illegal operation		

Figure 21.16 *Connection establishment*

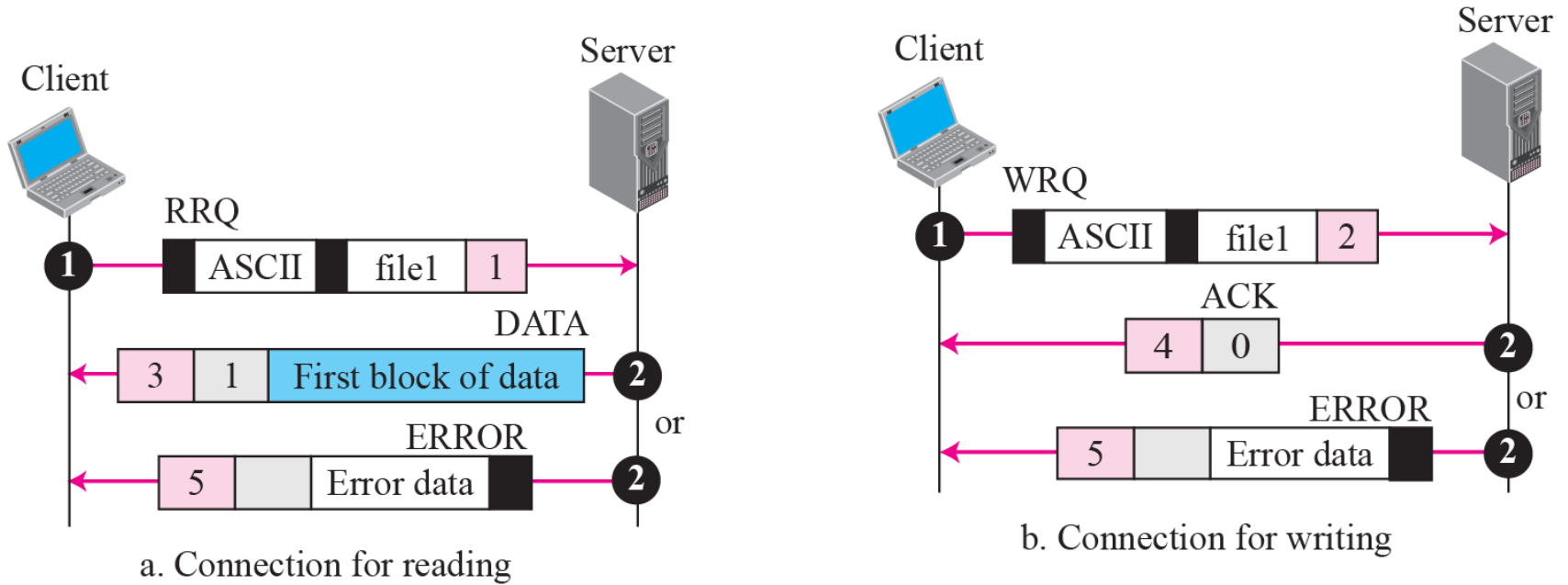


Figure 21.17 *Server's apprentice bug*

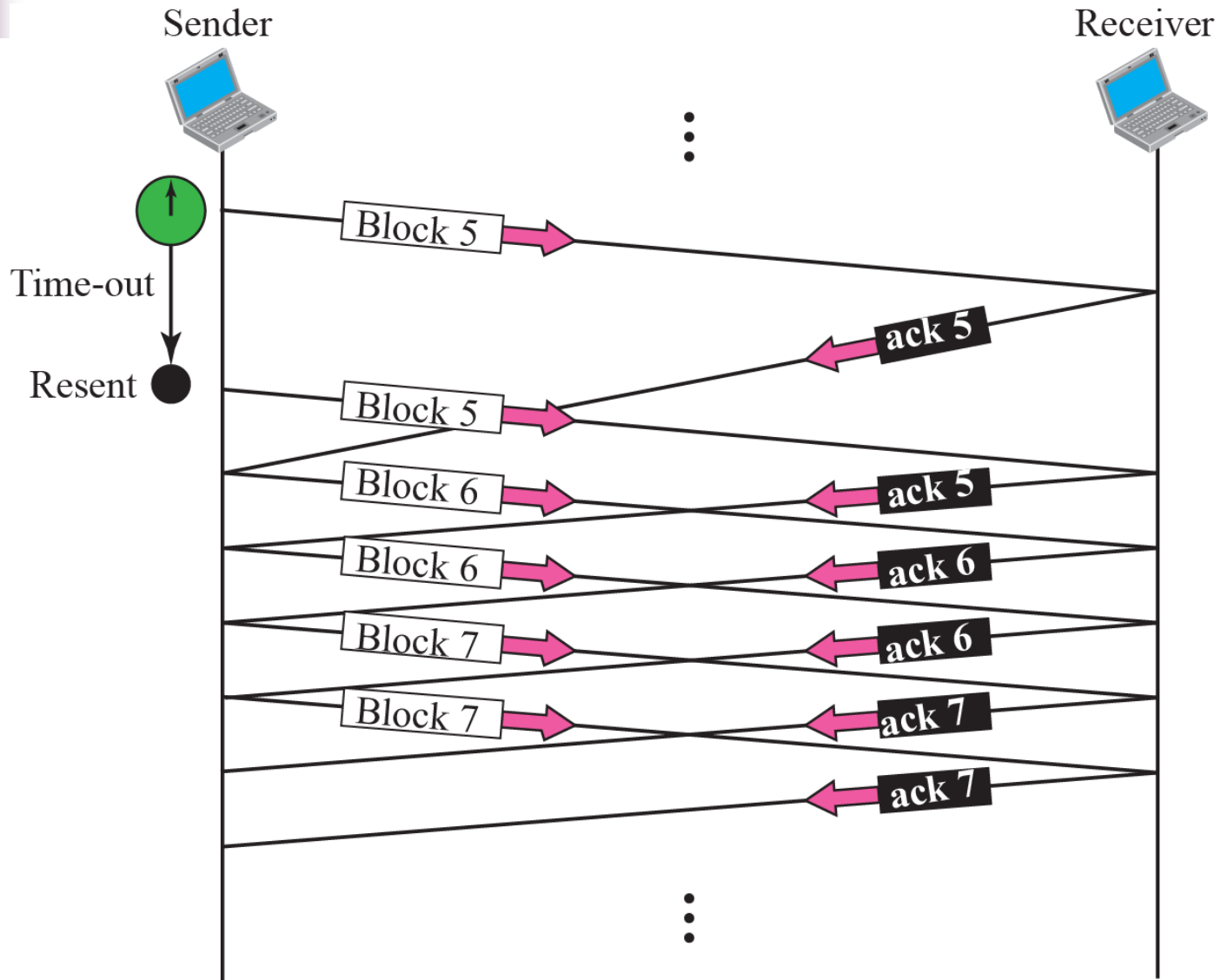


Figure 21.18 *UDP port numbers used by TFTP*

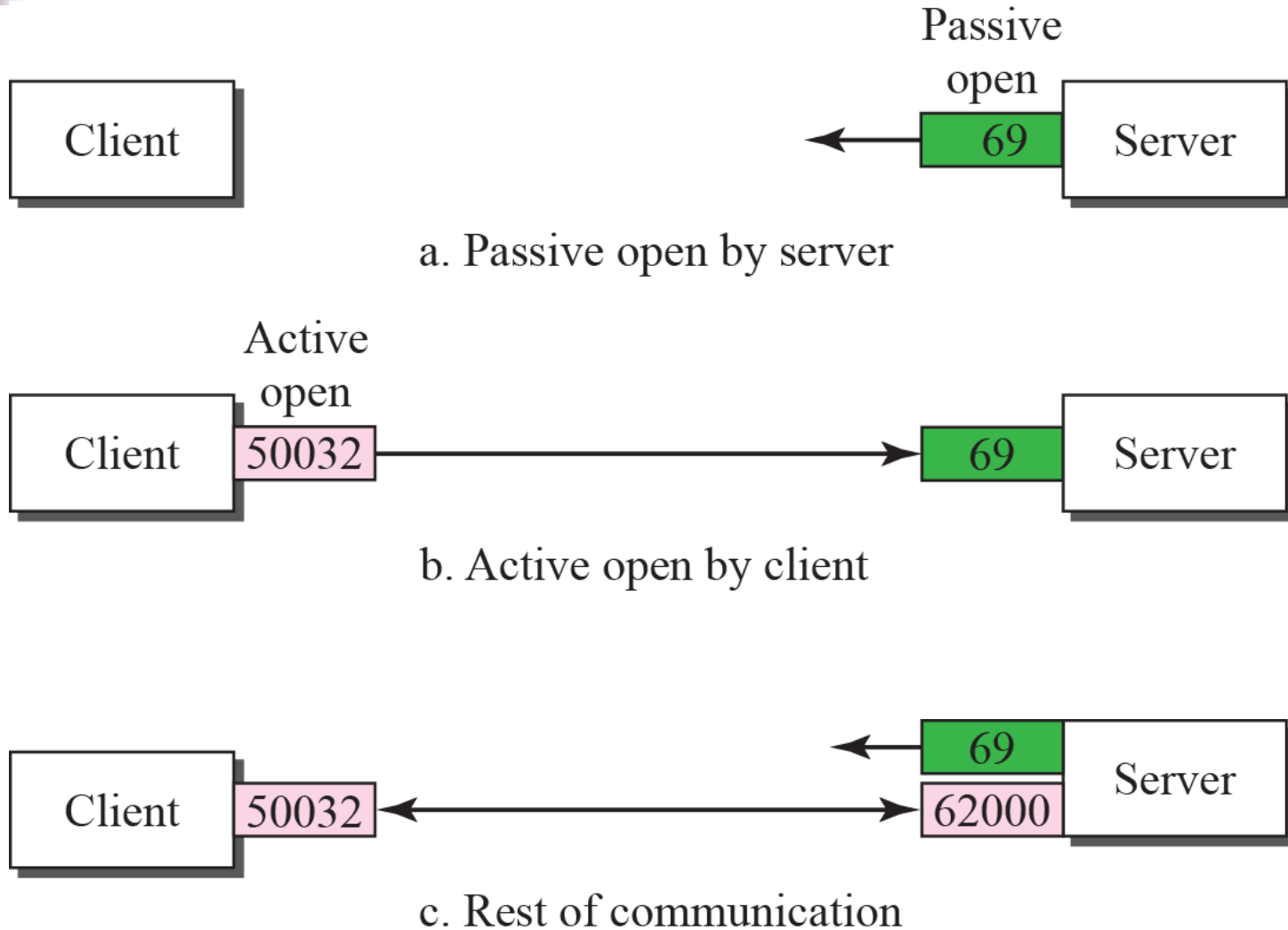


Figure 21.19 *TFTP example*

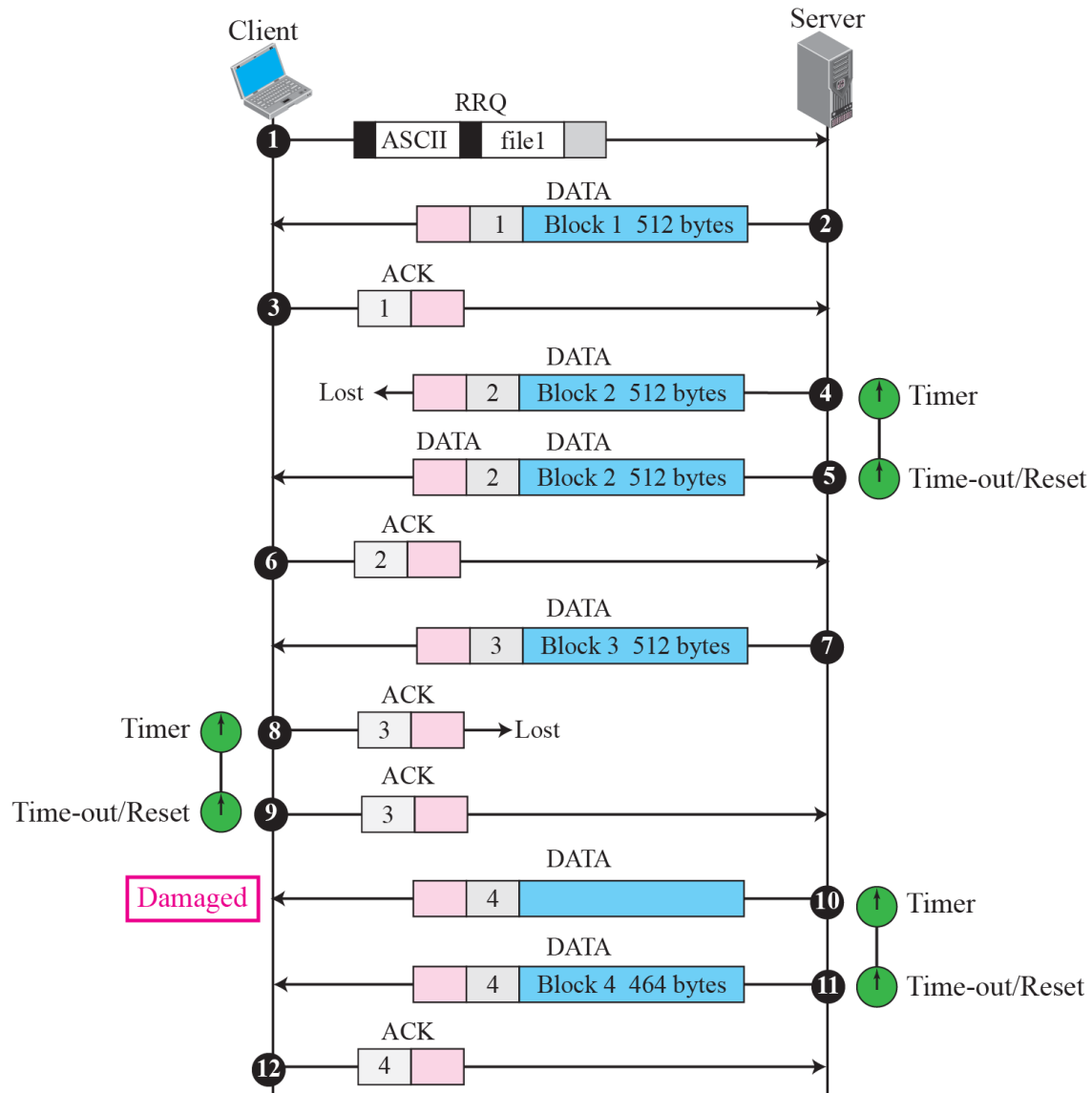


Figure 21.20 *Use of TFTP with DHCP*

