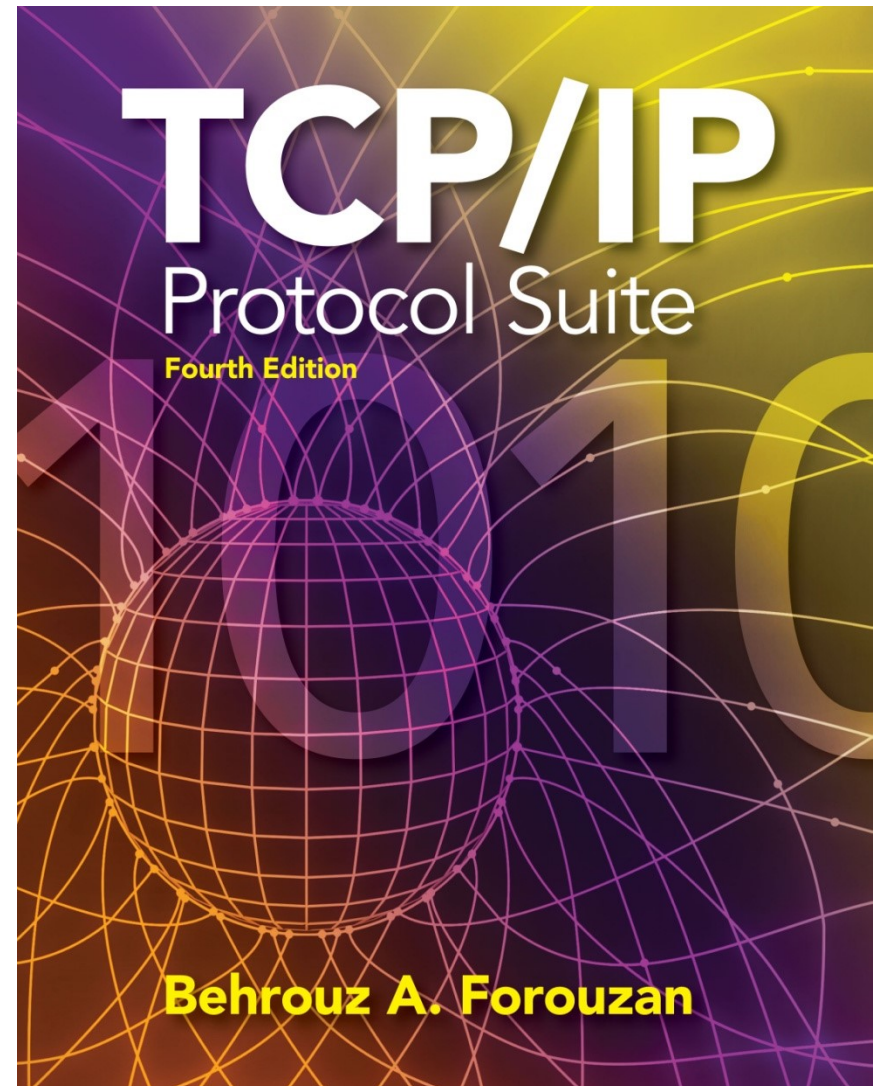


Chapter 20

Remote Login: TELNET and SSH



OBJECTIVES:

- ☐ **To introduce the TELNET protocol and show how it implements local and remote login.**
- ☐ **To discuss options and sub-options used in TELNET and how they are negotiated.**
- ☐ **To define out-of-band signaling in TELNET.**
- ☐ **To define different modes of operations in TELNET.**
- ☐ **To introduce SSH as an alternative to TELNET.**
- ☐ **To show how different components of SSH are combined to provide a secure connection over an insecure TCP connection.**
- ☐ **To discuss port-forwarding in SSH and how it can be used to provide security for other applications.**

Chapter Outline

20.1 TELNET

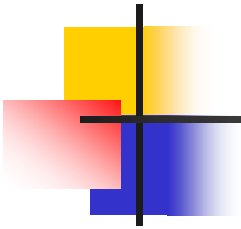
20.2 Secure Shell (SSH)

20-1 TELNET

TELNET is an abbreviation for TErminaL NETwork. It is the standard TCP/IP protocol for virtual terminal service as proposed by ISO. TELNET enables the establishment of a connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system.

Topics Discussed in the Section

- ✓ **Concepts**
- ✓ **Time-Sharing Environment**
- ✓ **Network Virtual Terminal (NVT)**
- ✓ **Embedding**
- ✓ **Options and Suboption Negotiation**
- ✓ **Controlling the Server**
- ✓ **Out-of-Band Signaling**
- ✓ **Escape Character**
- ✓ **Modes of Operation**
- ✓ **User Interface**
- ✓ **Security Issue**



Note

***TELNET is a general-purpose
client-server application program.***

Figure 20.1 *Local login*

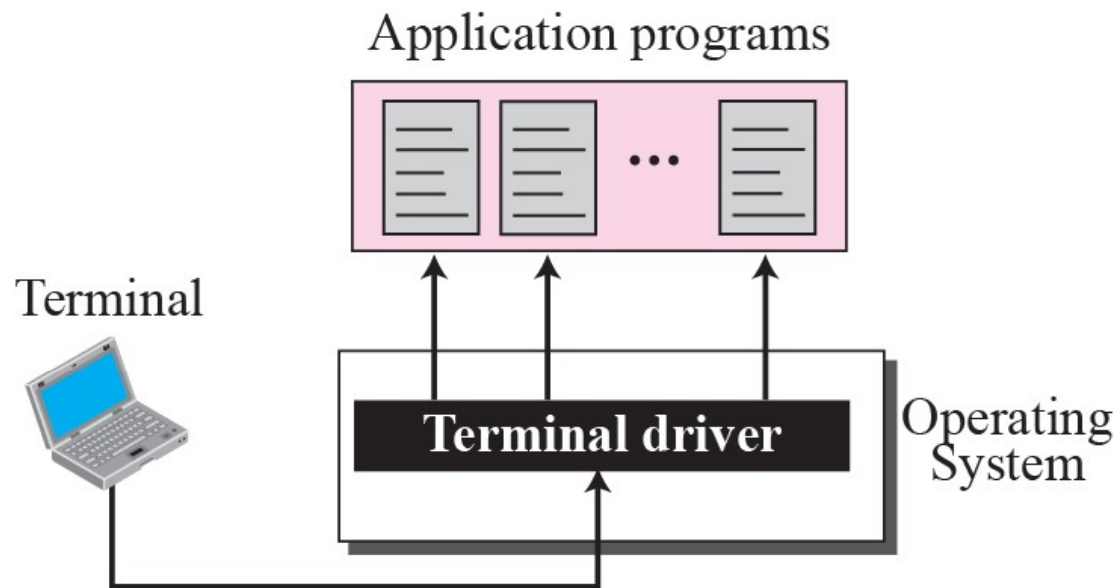


Figure 20.2 *Remote login*

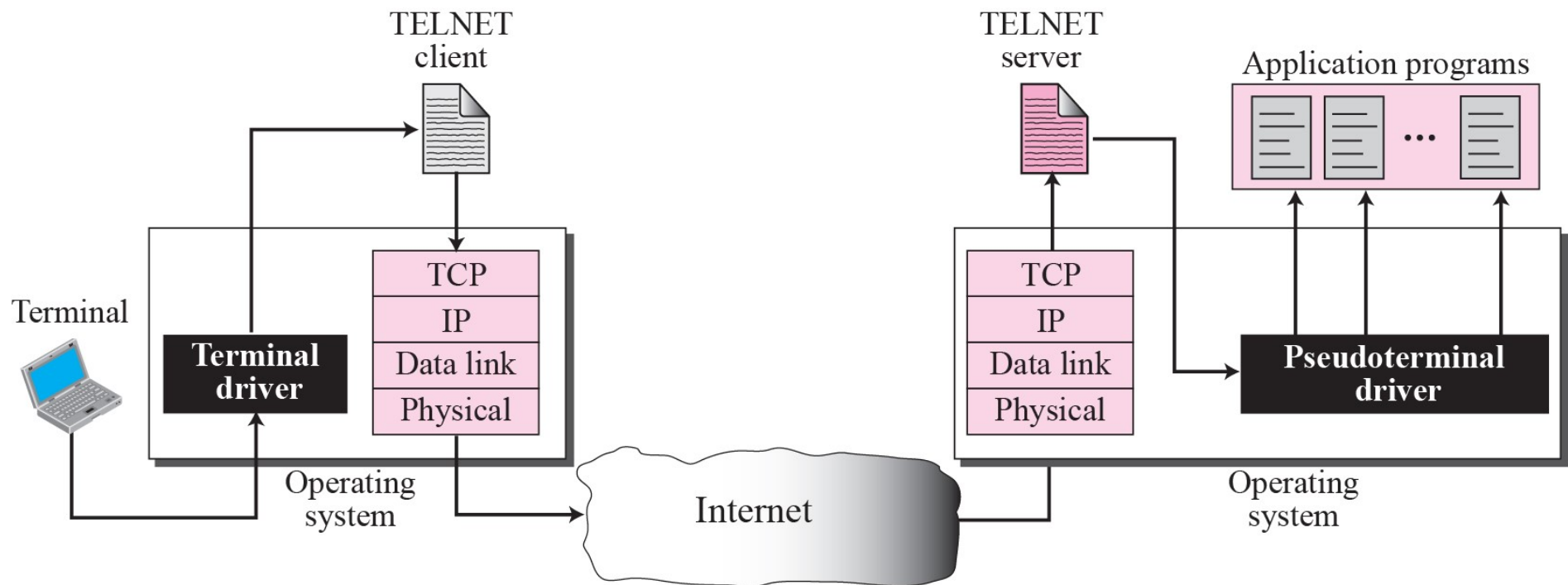


Figure 20.3 *Concept of NVT*

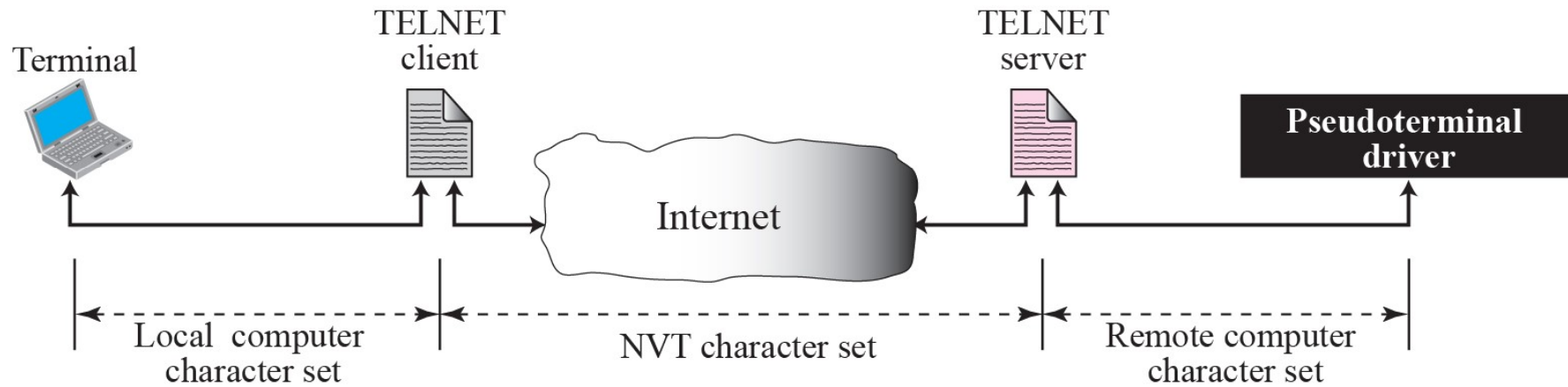
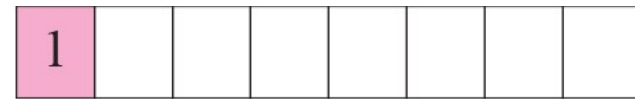




Figure 20.4 *Format of data and control characters*



a. Data Character



b. Control Character

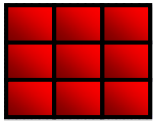
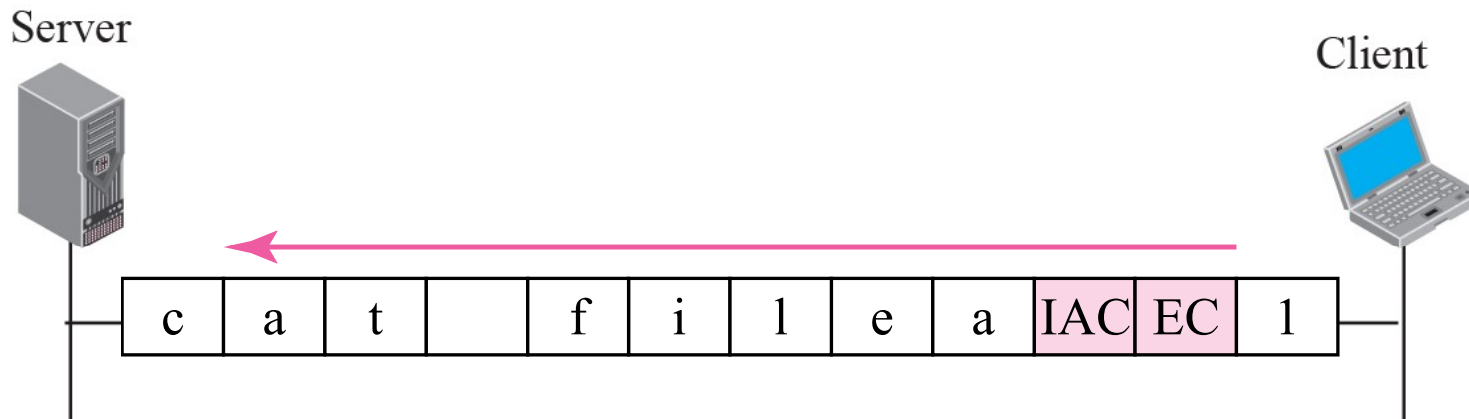


Table 20.1 *Some NVT control characters*

<i>Character</i>	<i>Decimal</i>	<i>Binary</i>	<i>Meaning</i>
EOF	236	11101100	End of file
EOR	239	11101111	End of record
SE	240	11110000	Suboption end
NOP	241	11110001	No operation
DM	242	11110010	Data mark
BRK	243	11110011	Break
IP	244	11110100	Interrupt process
AO	245	11110101	Abort output
AYT	246	11110110	Are you there?
EC	247	11110111	Erase character
EL	248	11111000	Erase line
GA	249	11111001	Go ahead
SB	250	11111010	Suboption begin
WILL	251	11111011	Agreement to enable option
WONT	252	11111100	Refusal to enable option
DO	253	11111101	Approval to option request
DONT	254	11111110	Denial of option request
IAC	255	11111111	Interpret (the next character) as control

Figure 20.5 *An example of embedding*



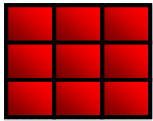


Table 20.2 *Options*

<i>Code</i>	<i>Option</i>	<i>Meaning</i>
0	Binary	Interpret as 8-bit binary transmission
1	Echo	Echo the data received on one side to the other
3	Suppress go-ahead	Suppress go-ahead signals after data
5	Status	Request the status of TELNET
6	Timing mark	Define the timing marks
24	Terminal type	Set the terminal type
32	Terminal speed	Set the terminal speed
34	Line mode	Change to line mode

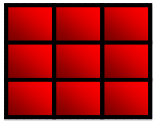


Table 20.3 *NVT character set for option negotiation*

<i>Character</i>	<i>Code</i>	<i>Meaning 1</i>	<i>Meaning 2</i>	<i>Meaning 3</i>
WILL	251	Offering to enable	Accepting to enable	
WONT	252	Rejecting to enable	Offering to disable	Accepting to disable
DO	253	Approving to enable	Requesting to enable	
DONT	254	Disapproving to enable	Approving to disable	Requesting to disable

Figure 20.6 *Offer to enable an option*

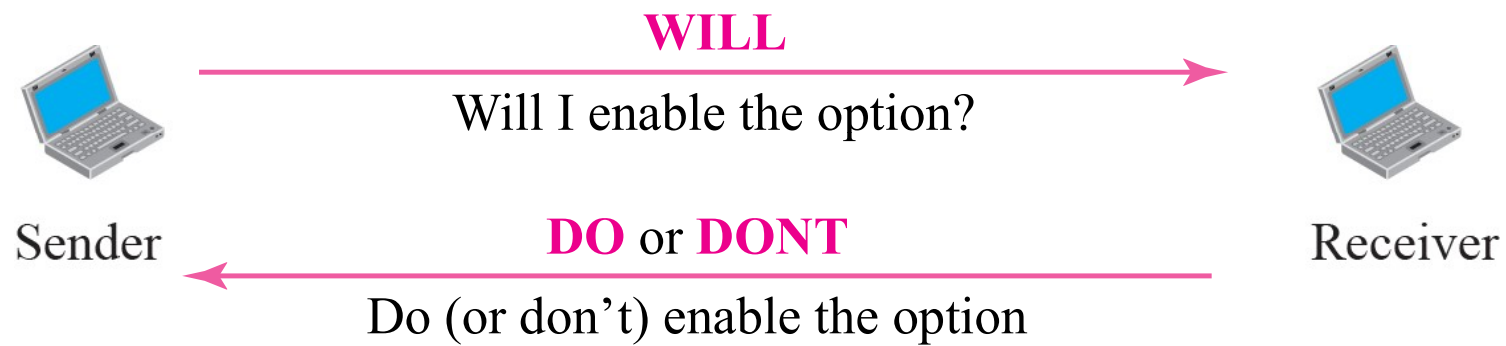


Figure 20.7 *Request to enable an option*

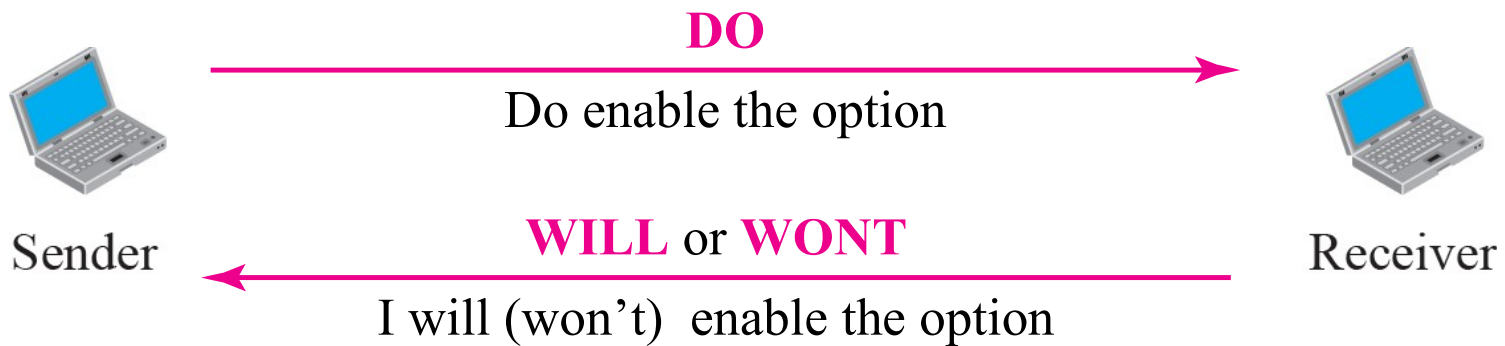


Figure 20.8 *Offer to disable an option*

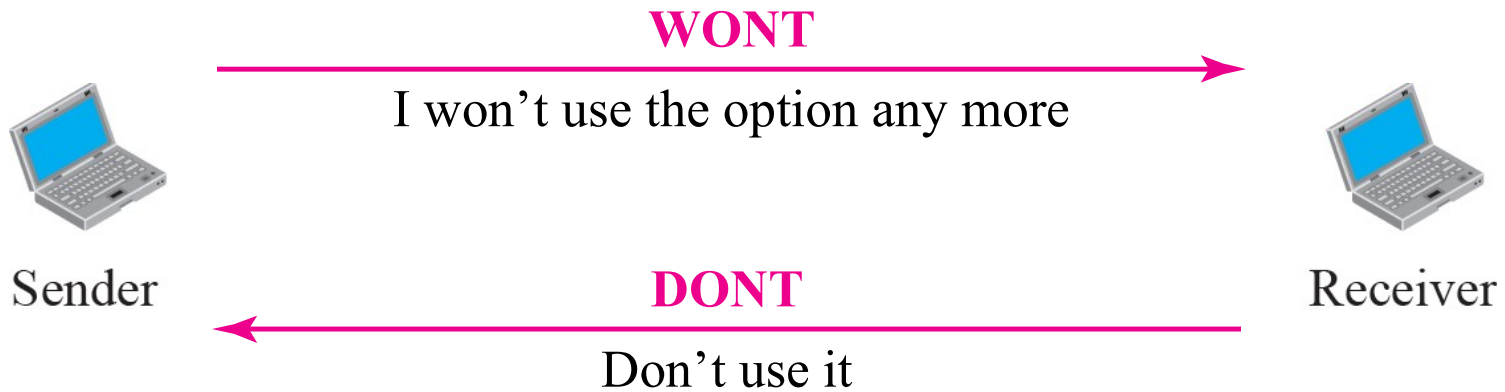
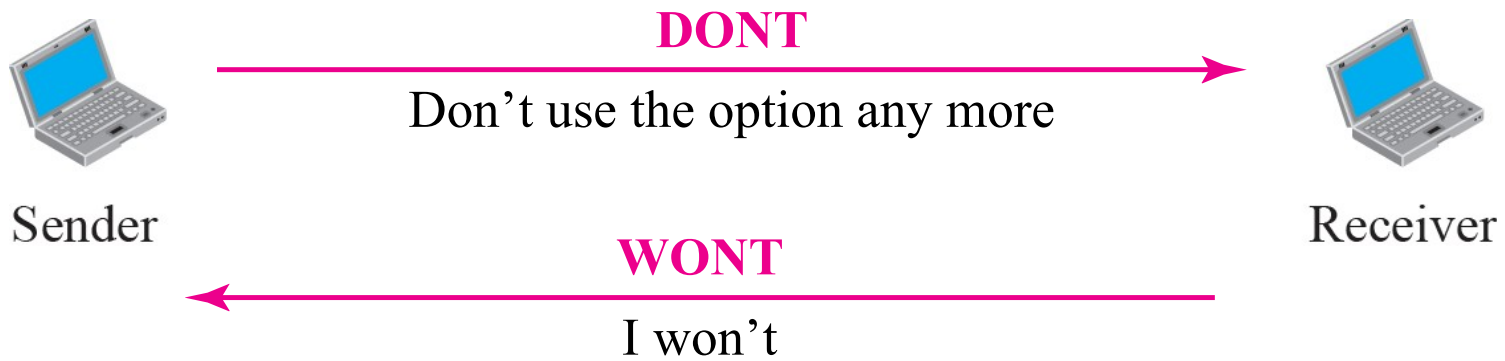


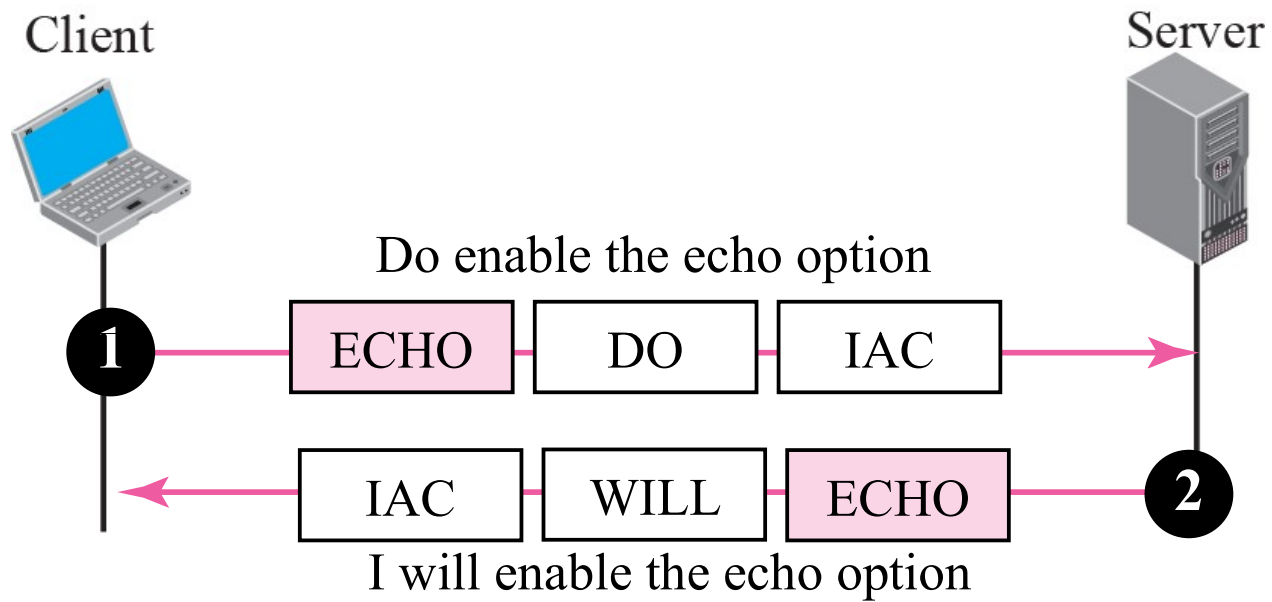
Figure 20.9 *Request to disable an option*



Example 20.1

Figure 20.10 shows an example of option negotiation. In this example, the client wants the server to echo each character sent to the server. In other words, when a character is typed at the user keyboard terminal, it goes to the server and is sent back to the screen of the user before being processed. The echo option is enabled by the server because it is the server that sends the characters back to the user terminal. Therefore, the client should request from the server the enabling of the option using DO. The request consists of three characters: IAC, DO, and ECHO. The server accepts the request and enables the option. It informs the client by sending the three-character approval: IAC, WILL, and ECHO.

Figure 20.10 *Example 20.1: Echo option*



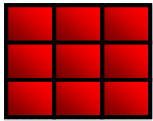
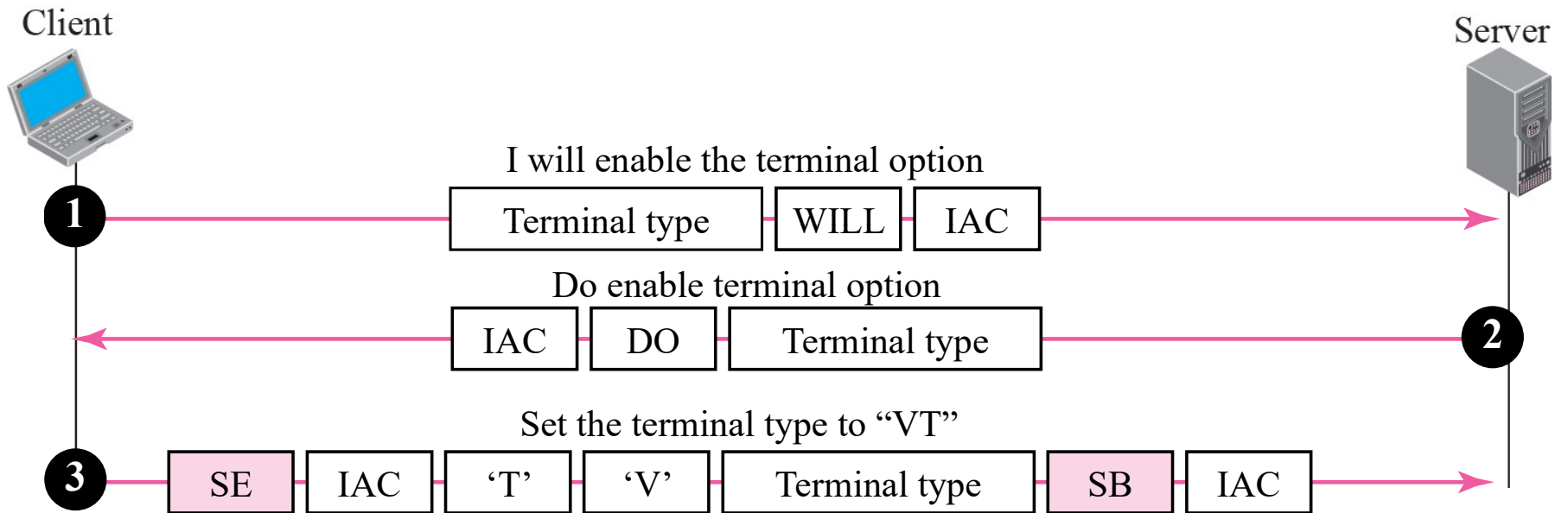


Table 20.4 *NVT character set for suboption negotiation*

<i>Character</i>	<i>Decimal</i>	<i>Binary</i>	<i>Meaning</i>
SE	240	11110000	Suboption end
SB	250	11111010	Suboption begin

Figure 20.11 *Example of sub-option negotiation*



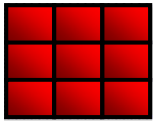


Table 20.5 *Characters used to control a program running on remote server*

<i>Character</i>	<i>Decimal</i>	<i>Binary</i>	<i>Meaning</i>
IP	244	11110100	Interrupt process
AO	245	11110101	Abort output
AYT	246	11110110	Are you there?
EC	247	11110111	Erase the last character
EL	248	11111000	Erase line

Figure 20.12 *Example of interrupting an application program*

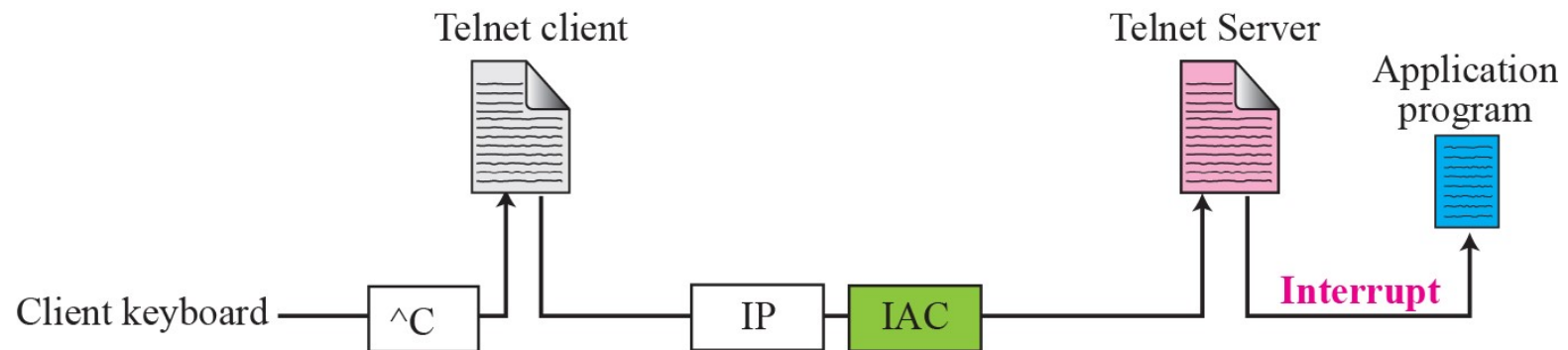


Figure 20.13 *Out-of-band signaling*

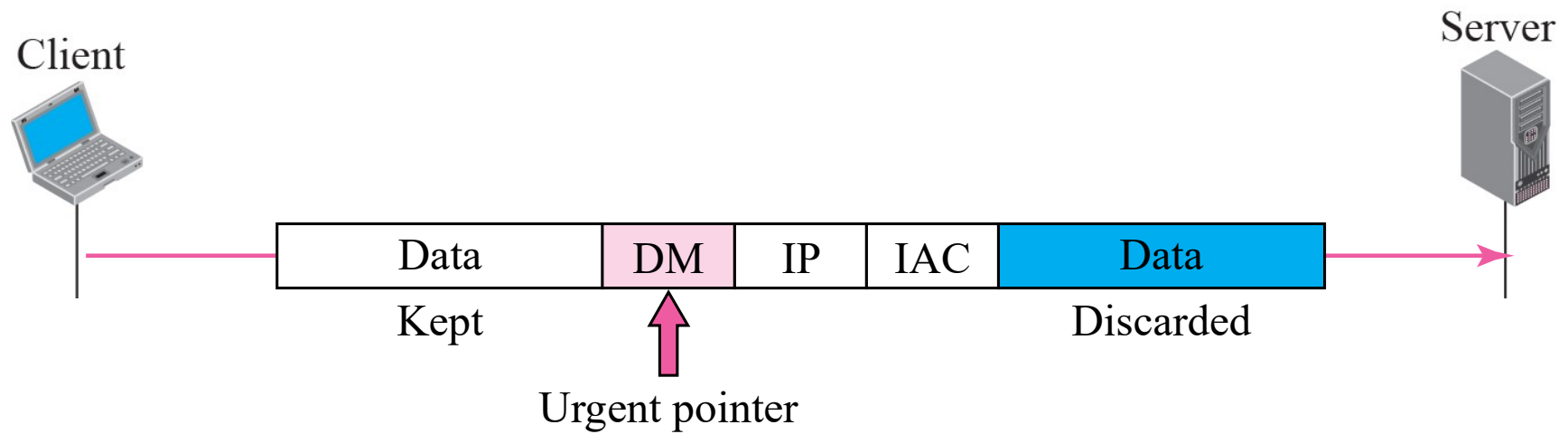
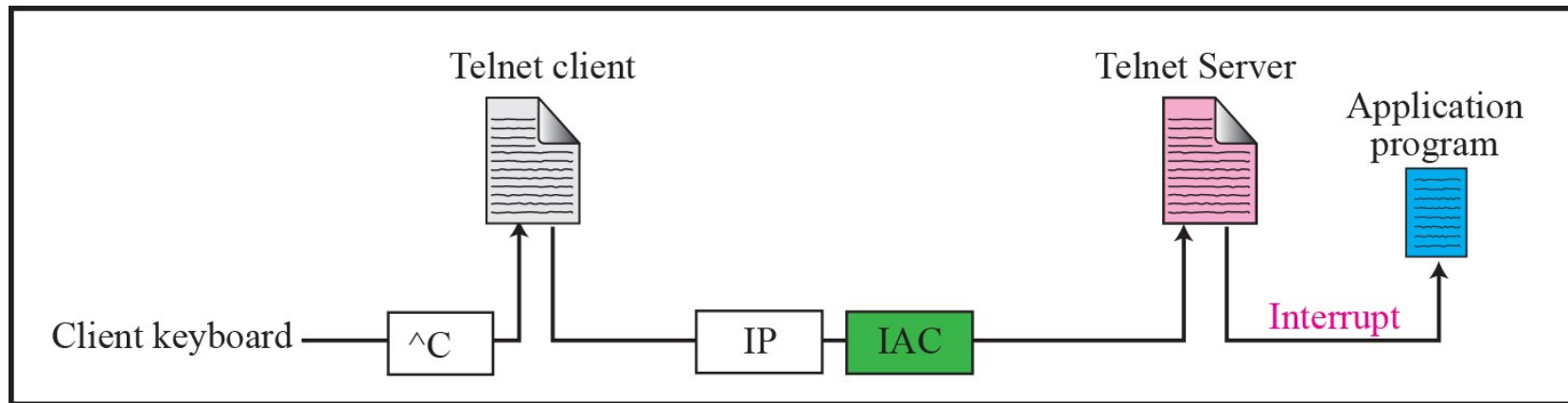
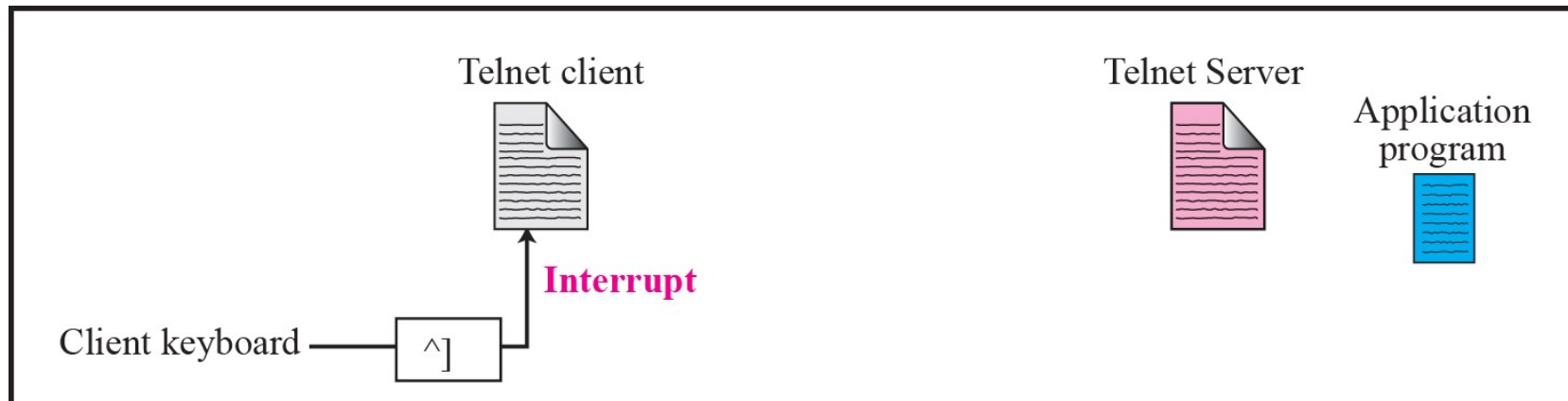


Figure 20.14 *Two different interruptions*



a. Interrupting the application program

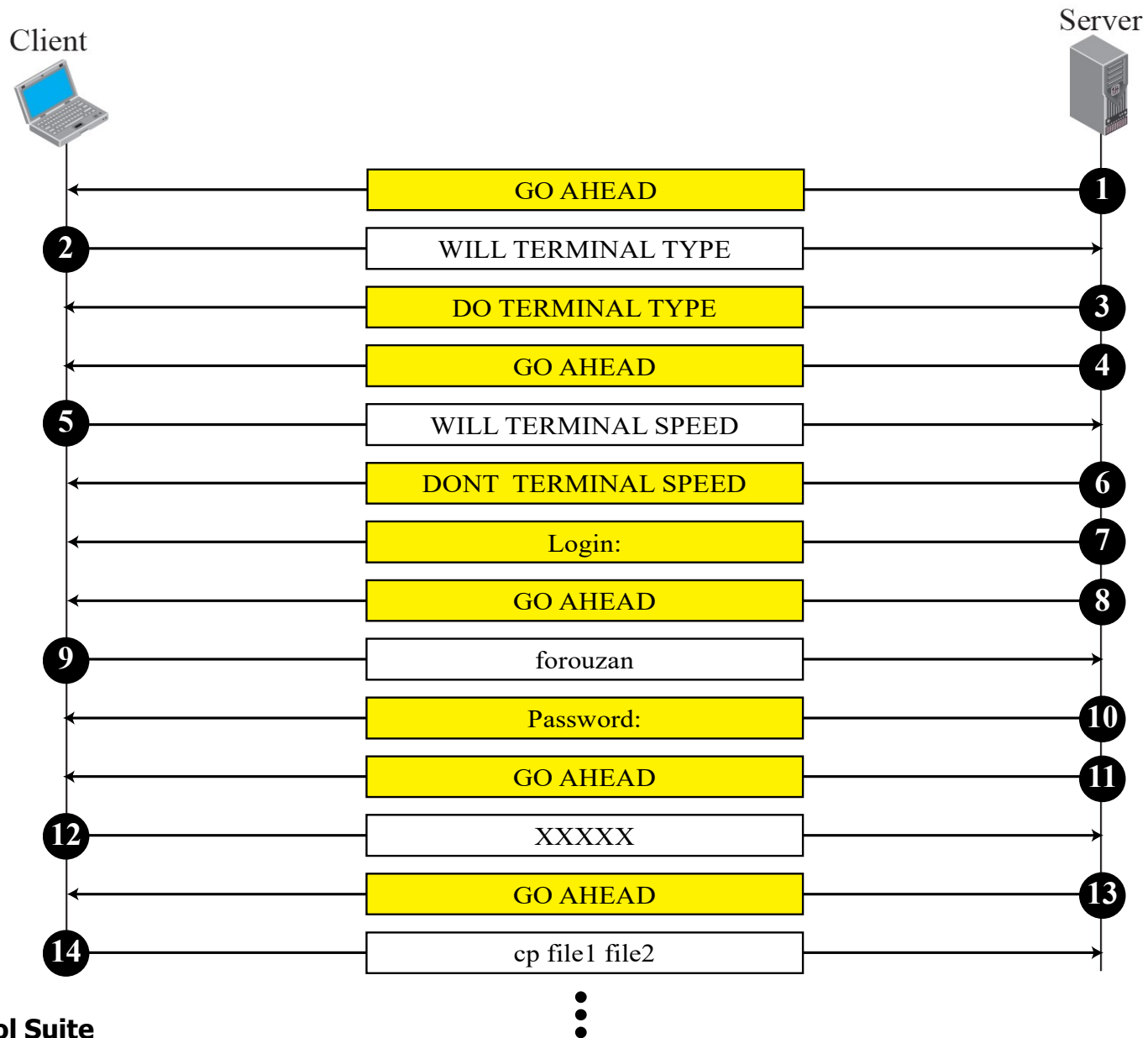


b. Interrupting the client

Example 20.2

In this example, we use the default mode to show the concept and its deficiencies even though it is almost obsolete today. The client and the server negotiate the terminal type and terminal speed and then the server checks the login and password of the user (see Figure 20.15).

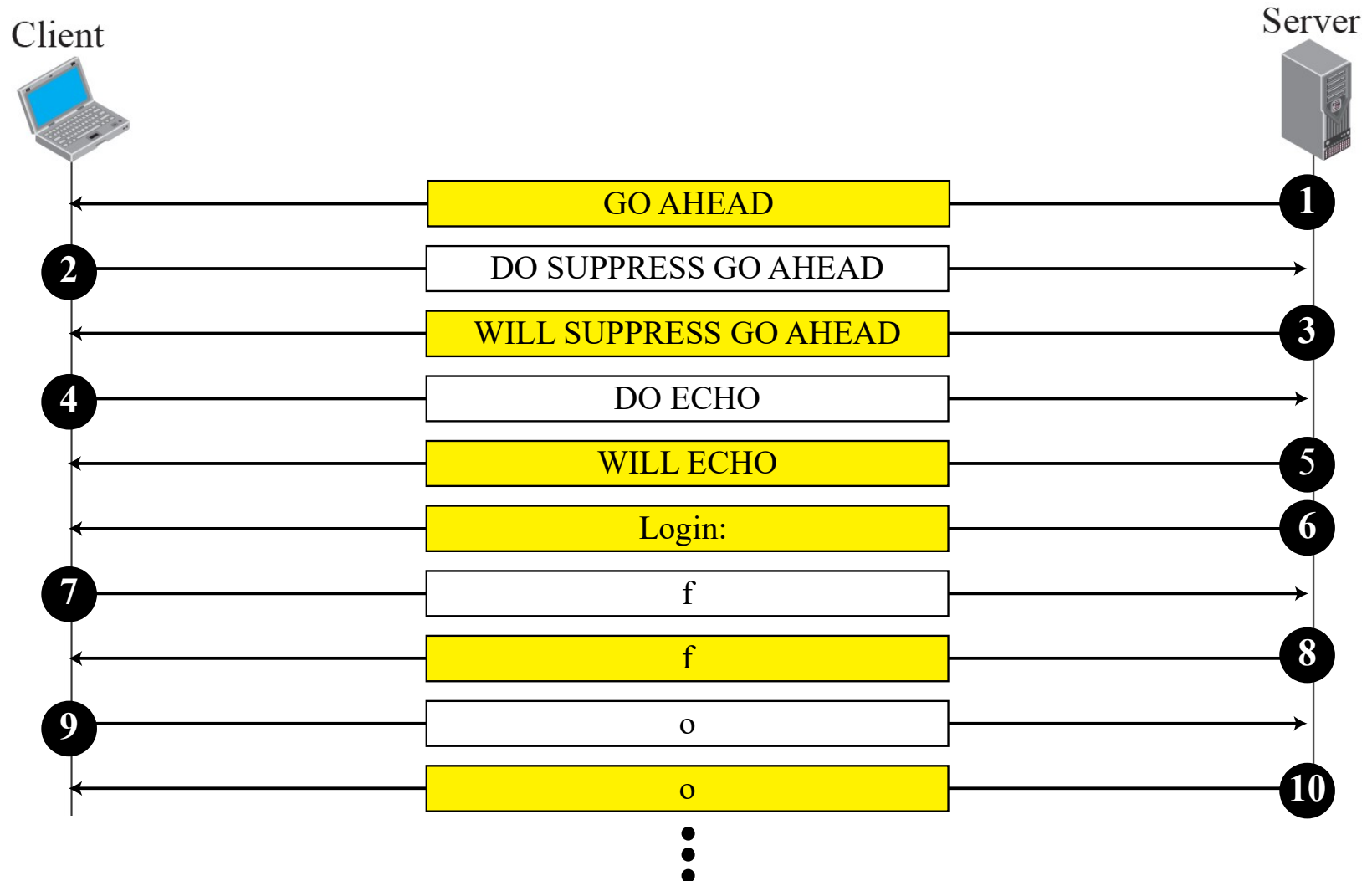
Figure 20.15 *Example 20.2*



Example 20.3

In this example, we show how the client switches to the character mode. This requires that the client request the server to enable the SUPPRESS GO AHEAD and ECHO options (see Figure 20.16).

Figure 20.16 *Example 20.3*



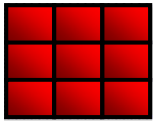


Table 20.6 *Examples of interface commands*

<i>Command</i>	<i>Meaning</i>	<i>Command</i>	<i>Meaning</i>
open	Connect to a remote computer	set	Set the operating parameters
close	Close the connection	status	Display the status information
display	Show the operating parameters	send	Send special characters
mode	Change to line or character mode	quit	Exit TELNET

20-2 SECURE SHELL (SSH)

Another popular remote login application program is Secure Shell (SSH). SSH, like TELNET, uses TCP as the underlying transport protocol, but SSH is more secure and provides more services than TELNET.

Topics Discussed in the Section

- ✓ **Versions**
- ✓ **Components**
- ✓ **Port Forwarding**
- ✓ **Format of the SSH Packet**

Figure 20.17 *Components of SSH*

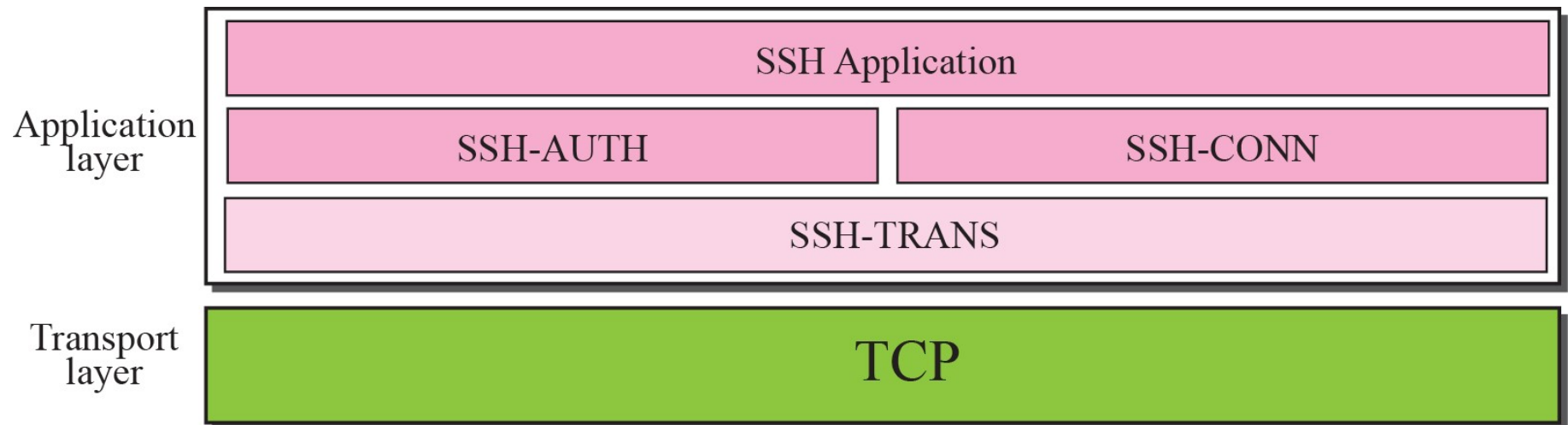


Figure 20.18 *Port forwarding*

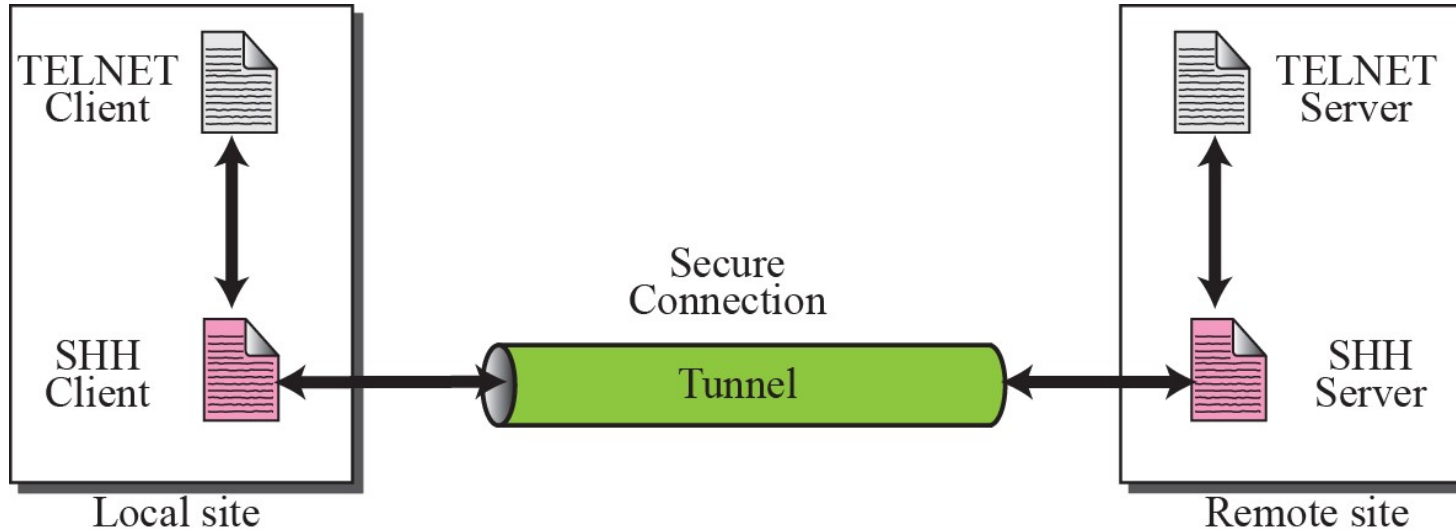


Figure 20.19 *SSH packet format*

