

# Module 2 Tutorials & Solutions

## Application Layer (Forouzan Chapter 23-23)

Daniel Franklin

August 13, 2020

**20.2** If TELNET is using the character mode, how many characters are sent back and forth between the client and server to copy a file named file1 to another file named file2 in Unix (`cp file1 file2`)?

A: With the TELNET protocol, each character is sent over a TCP connection one at a time, then echoed back by the server (and ACKed, in the same segment) and only then displayed on the client screen (and ACKed back to the server, confirming to both ends that the character has been received and the client knows that it has been received). This all happens much faster than the user can type. So each character of the above command (which should include a linefeed character as well, for a Unix operating system) results in the transmission of three TCP segments (client character to server, server echo/ack to client, client ack to server). Only the first two of these contain a 1-character payload. So the total number of characters sent in EACH direction in this case is 15 (including the line feed), and the total number of TCP segments (assuming the connection has been established and the initial TELNET terminal negotiation is complete) is 45.

**21.1** What do you think would happen if the control connection is accidentally severed during an FTP transfer?

A: This depends on the mode of severance, and whether the data connection was operational at the same time. If the control connection is killed ‘properly’ with a TCP RST packet, the server will shut down the control and any open data TCP connections (i.e. if data transfer is in progress). If the connection is severed without a shutdown (e.g. a cable is unplugged) and if no data transfer is in progress, eventually the control connection will time out, and the server will unilaterally terminate the TCP session - if data transfer is in progress, but the client is now unreachable, the operating system will realise that the other end is dead due to the lack of ACKs, at which point the data connection is terminated; the FTP server will then close the control connection endpoint with that client as well. If data transfer was underway, and the control connection was severed without a RST but the data connection was still alive, it would finish as normal and then terminate, and the control connection would eventually time out.

**21.2** Explain why the client issues an active open for the control connection and a passive open for the data connection.

A: Control is active from the client perspective since it is responsible for initiating the transfer. Data is passive because the port is only opened when the client is ready to send or receive data; the server initiates the connection. In passive mode, however, the server passively opens a second port for data transfer and the client makes a connection to that.

The use of two TCP sessions for FTP is one of its major shortcomings - the other being the lack of encryption and end-to-end system authentication.

**21.3** Why should there be limitations on anonymous FTP? What could an unscrupulous user do?

A: Allowing anonymous file uploads would be (and used to be) a great way to illegally distribute content via a 3rd party server. Allowing unrestricted access to other parts of the filesystem (even read-only) is also obviously a massive security problem.

HTTP is much better for most of what anonymous FTP was once used for (non-anonymous FTP has largely been replaced by sftp/rsync-over-ssh).

**21.4** Explain why FTP does not have a message format.

A: Each FTP command does exactly one thing - so all that is needed is a command (e.g. PORT) and argument (e.g. numerical value). There's no need to delimit more complex operations within an individual 'message' - more complex operations would be performed as a sequence of simpler atomic operations.

**22.11** Draw a diagram to show the use of a proxy server that is part of the client computer:

1. Show the transactions between the client, proxy server, and the target server when the response is stored in the proxy server.

A:

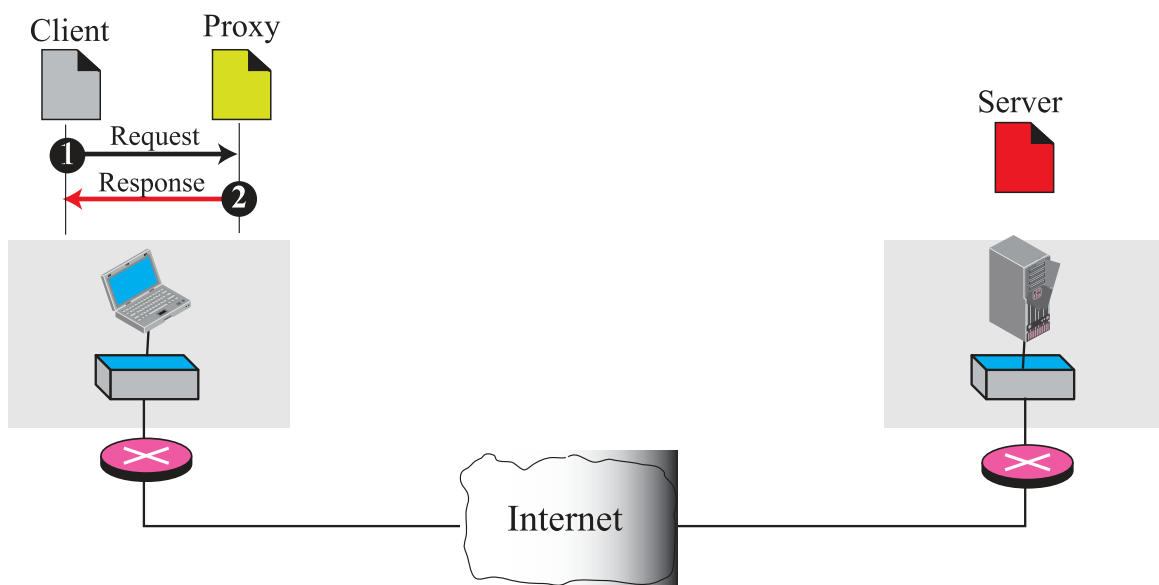


Figure 1: Solution for 22.11a

2. Show the transactions between the client, proxy server, and the target server when the response is not stored in the proxy server.

A:

**22.13** Draw a diagram to show the use of a proxy server that is installed in the ISP network:

1. Show the transactions between the client, proxy server, and the target server when the response is stored in the proxy server.

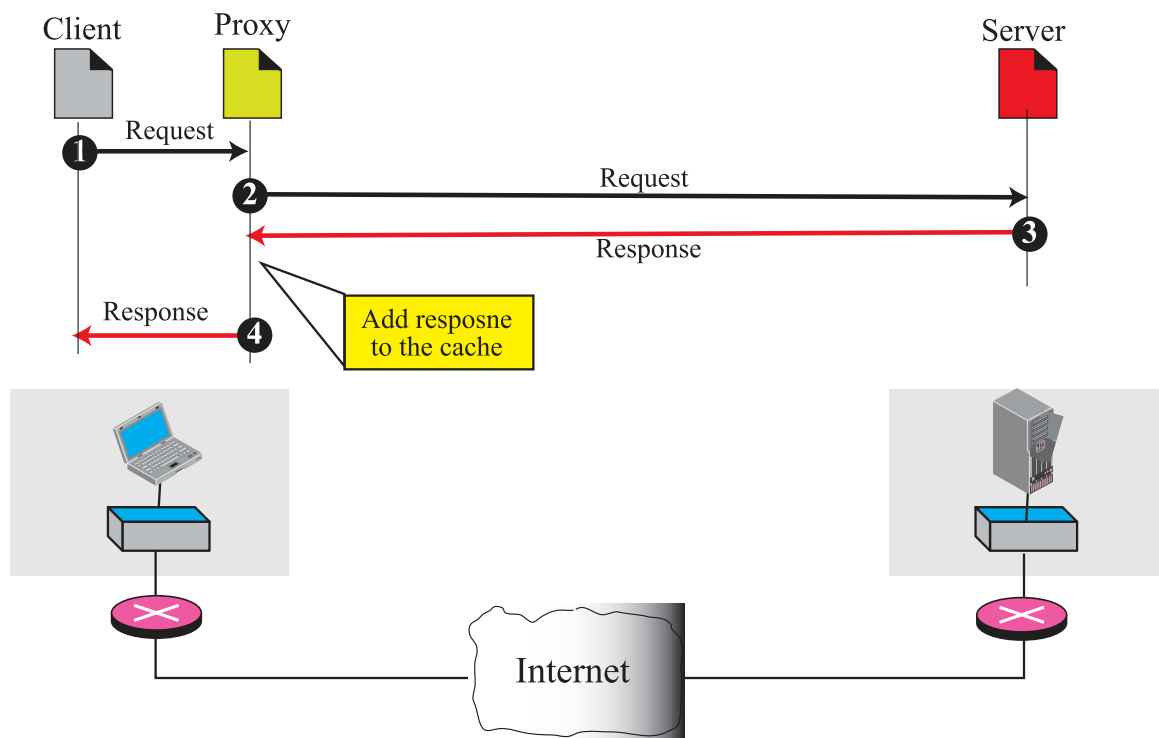


Figure 2: Solution for 22.11b

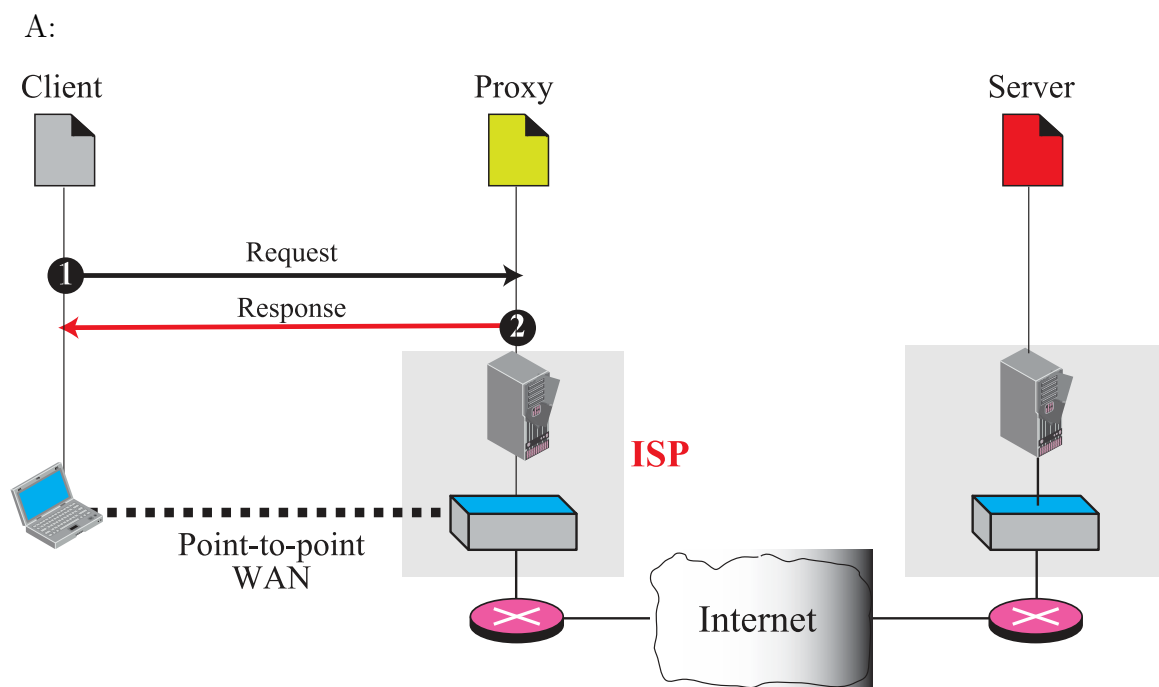


Figure 3: Solution for 22.13a

2. Show the transactions between the client, proxy server, and the target server when the response is not stored in the proxy server.

**23.10** Are the HELO and MAIL FROM commands both necessary? Why or why not?

A: The HELO command is used by the client to identify itself to the server (i.e. the client domain name); the MAIL FROM command identifies the sender of the e-mail. Of course, this

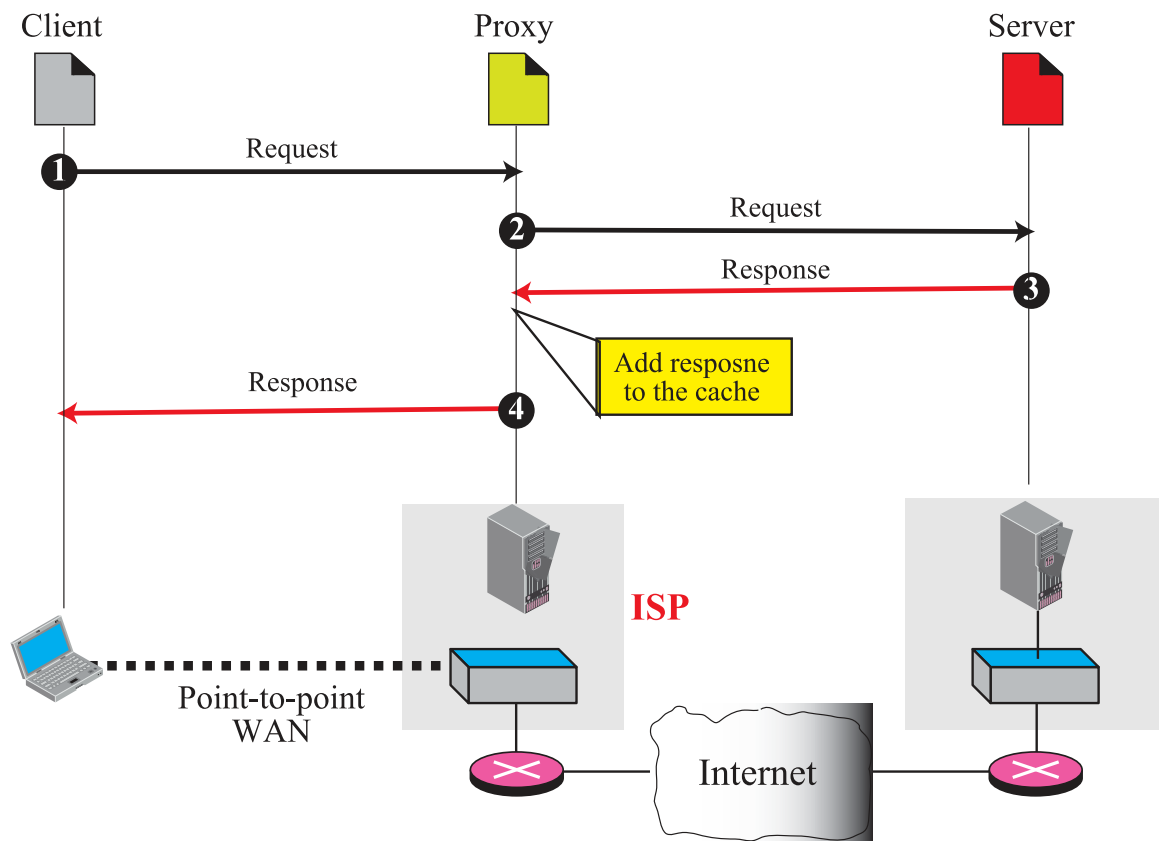


Figure 4: Solution for 22.13b

can be trivially falsified as by default SMTP does not include an authentication or message integrity mechanism (unless over an SSL/TLS connection).

**23.11** In Figure 23.11 what is the difference between MAIL FROM in the envelope and the From in the header?

A: The MAIL FROM: in the envelope contains the source e-mail address while the FROM in the header contains the name of the sender.

**23.12** Why is a connection establishment for mail transfer needed if TCP has already established a connection?

A: This delimits an individual e-mail (as opposed to a connection between the client and server which may be used to send many e-mails).

**23.18** In SMTP, if we send a one-line message between two users, how many lines of commands and responses are exchanged?

A: Here is a sample transaction. C denotes client; S denotes server:

```

S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.com
S: 250 smtp.example.com, I am glad to meet you
C: MAIL FROM:<bob@example.com>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok

```

C: DATA  
S: 354 End data with <CR><LF>.<CR><LF>  
C: From: "Bob Example" <bob@example.com>  
C: To: Alice Example <alice@example.com>  
C: Date: Tue, 15 Jan 2008 16:02:43 -0500  
C: Subject: Test message  
C:  
C: This is a test message with 4 header fields and 1 line in the message body.  
C: .  
S: 250 Ok: queued as 12345  
C: QUIT  
S: 221 Bye  
{The server closes the connection}