

CHAPTER 3 – REFERENCE TEXT

TRADITIONAL APPLICATIONS

Review Questions

- 3.1 What is a network virtual terminal?
- 3.2 What are the four phases of operation of a typical virtual terminal protocol?
- 3.3 What is the distinction between User Telnet and Server Telnet?
- 3.4 Briefly summarize the Telnet synch mechanism.
- 3.5 List and briefly define the categories of Telnet options.
- 3.6 Explain the use of the terms WILL, WONT, DO, DONT in Telnet.
- 3.7 What is the distinction between User FTP and Server FTP?
- 3.8 What data types are supported by FTP?
- 3.9 What file types are supported by FTP?
- 3.10 What transmission modes are supported by FTP?
- 3.11 How is the restart marker used in FTP?

Problems

Note: For some of the problems in this Chapter, you will need to consult the relevant RFCs.

- 3.1 When a Telnet implementation issues a WILL command, RFC 854 states that it must wait until receiving a DO command before enabling the option and that if a DONT command is received, the option remains disabled. Thus, the side issuing the WILL command must remember that it has issued that command and wait for a reply. A similar requirement applies to the DO command. However, RFC 854 does not explicitly state that a Telnet implementation must remember beginning a DONT or WONT negotiation. At first glance, it seems reasonable to simply issue the DONT or WONT and disable the option immediately and ignore the WONT/DONT response from the other side. This is so because the other side has no choice but to agree to disable the option. However, this strategy is incorrect because it could lead to an endless loop. Demonstrate the problem by showing a sequence of commands that lead to an endless loop if Telnet implementations do not remember beginning a DONT or WONT negotiation. *Hint:* Suppose one side decides to disable an option and then decides to reenable that option.
- 3.2 It is unclear from RFC 854 whether or not a TELNET implementation may allow new requests about an option that is currently under negotiation; it certainly seems limiting to prohibit “option typeahead.” Suppose an option is disabled, and we decide in quick succession to enable it, disable it, and reenable it. We send WILL WONT WILL and at the end remember that we are negotiating. Might this create a problem?