1. Routers (or “gateways”) only have to run a three-layer protocol, i.e. the transport layer is not needed. What are the three layers which are needed and what is the main function of each one?

2. The IP header has a field for “Datagram Number.” Since IP datagram delivery is unacknowledged, what is this field used for?

3. The TCP header has two 32-bit fields called “Sequence Number” and “Ack Number.” Explain precisely how these are used during transmission of bytes between sender and receiver over a tcp connection. Also explain what happens if a tcp-segment containing a single byte is lost.

4. In the example telnet dump we went over in class many of the tcp packets carried no data and had to be padded with 6 random bytes. Why did these frames carry no data? What was the purpose of sending them? Why were they padded?

5. How do statistical (or “learning”) bridges avoid passing unnecessary traffic across them and avoid loops? Give a complete answer and tell which address types are monitored by the bridges.

6. One routing option is called “Strict Source Routing,” and essentially includes (in the header) a complete path for the packet from source to destination. Under what special situations is this option useful, and why is it not used very often?

7. What precautions are taken on the Internet so that lost IP datagrams don’t circulate around the Internet forever?

8. In order to generate a new offspring process, Unix has the system call fork() whereas Windows NT uses the family of spawnxx() system calls. Explain the precise differences between these calls. For example, in each case what is the entry point for the offspring process, what is shared between the two processes (parent and offspring), what is not shared between the two processes, etc.?

9. The three most common transport protocols are icmp (1), udp (6), and tcp (17) (see the file /etc/protocols). Name three uses of icmp packets. How does the program traceroute use icmp packets?

10. In the pipesend.c program under Windows NT, we had to pass the pipe
descriptor to the offspring process on the command line. Why didn’t this have to be done in the case of Unix?

11. In wide area network topology one often has to find what is called the “shortest path.” Does this refer to the actual length of the transmission line? Why or why not?

12. In the daemon.c program under Unix, how was the parent process informed when one of its offspring terminated?

13. In the s.sh.c program, what are the advantages to using stat() rather than fopen() to see if a given pathname leads to a existing file?