1. Two blue armies are each poised on opposite hills preparing to attack a single red army in the valley. The red army can defeat either of the blue armies separately but will fail to defeat both blue armies if they attack simultaneously. The blue armies communicate via an unreliable communications system (a foot soldier). The commander, with one of the blue armies, would like to attack at noon. His problem is this: If he sends a message ordering the attack, he cannot be sure it will bet through. He could ask for acknowledgment, but that might not get through. Is there a protocol that the two blue armies can use to avoid defeat? (*See also Holtzmann 4-8 as a variation. This is a classic problem for a class in protocol analysis and design.*)

The problem is a classic problem. Two items of information need to be exchanged and each must be confirmed. The first item is the battle plan and the second item is the readiness to execute the plan. Since two items need to be exchanged, the problem is performed in two steps initiated by the commander. The first step would be the issue of the battle plan, which must be acknowledged. All parties must affirm the plan. The commander will expect the affirmations (or rejections) within a certain period of time. If one is missing, the commander will re-issue the order (to the missing units). Eventually all members will agree (or the plan will be discarded due to a disagreement). An accepted plan will then move to the execution phase. The commander will issue the order placing the plan into execution. Each unit of the army will acknowledge the start. If an acknowledgement is missing, the commander can re-issue the order (to the missing units). There is no infinite acknowledgement sequence; none is required.

http://ei.cs.vt.edu/~cs5204/fall99/distributedDBMS/duckett/tpcp.html

2. (Blankenship 1.1) Consider a slightly different problem. An individual wishes to withdraw an amount of money from her account using an ATM. Is there a protocol that will allow the transaction to be made? If so, how does it differ from the protocol suggested for problem one (above)?

This problem is also a two stage operation. The first stage is the agreement that the intent is to make a withdrawal. The second stage is the commit to perform the actual withdrawal. In this case there is the ATM machine with a money dispenser and the bank with account balance. The objective is to synchronize the ATM (credit from the account) and the bank (debit from the account). The two entities are separated by a hostile environment. Both problems demand that the contents of the message be hidden from a third party. Both problems demand that the contents of the message be authenticated. The protocol is much the same; just the names change.

In reality, this is the same problem. The difference between the two problems is the hostility of the environment. In the first problem, the environment is so hostile that there might be some extra protocol items to deal with an extremely hostile environment. Can something be added to help the message get through? One could talk about some sort of forward error correction or the use of message templates.

3. (Blankenship 1.2) Why did I waste your time exploring these two basic problems? What do they have to do with this course?

It is very easy to confuse the environment of a problem with its details. It is easy to confuse the woods, creeks, distance and physical hostility of the first problem with the objective. They are separate. The environment mandates procedures that are used to move messages; three are constraints on the media. It was my objective to reveal this basic truth.

4. (Holtzmann 1-1) The transmission code developed by Polybius for his torch telegraph divided the 24-letrter Greek alphabet into five groups. The first four groups had five letters each, and the fifth group had the remaining four. The telegraph worked with two groups of torches: one was used to encode the group number, the other to transmit the character number within that group. Transmission took place character by character, by raising and lowering torches in the two groups. There were no codes for spaces to separate words, or for any kind of punctuation. (Punctuation was not used yet in written Greek either.) There was, however, one additional control message to signal the start of a message: two torches raised simultaneously (see the quotation from Polybius on page 3). What are the possible synchronization problems, in the absence of proper agreement on the order in which the torches in the two groups are to be lowered and raised?

The end result of a synchronization problem would be confusion resulting in not being ale to properly read the message. Each letter is identified by a tuple (group, item in group). The first question is, "What does the phrase raise n torches mean?" Do they have to be raised together? Can they be raised over time? If so, what is the time? How long must they be up? How is a torch failure addressed? How do I deal with perspective? How do I tell right from left? Whose right? How is a visibility problem addressed? Finally, how does the sender know that the receiver has seen the torches? All of the issues deal with error control.

5. (Holtzmann 1-7) (Jon Bentley) If a telephone call is unexpectedly terminated, there is an informal "telephone protocol" which says that the caller should redial the call. If the called party is unaware of this protocol a curious problem results. A "Lover's Paradox" prevents contact from being made when both parties try to establish it simultaneously. What is the protocol flaw? Assume the callers are machines, how could the machines be programmed to prevent the problem from repeating itself ad infinitem? What happens to this protocol if both parties have a "call interrupt" feature (the ability to take an extra call when already off hook)?

The first problem is a clear definition of "unexpected terminated". Is the call "unexpected terminated", if the call is never really started (physically answered or personally answered)? The flaw of the protocol is a possible infinite loop of return calls by one or both parties?

6. Holtzmann 2-1) Identify the five protocol elements from Section 2.2 for the torch telegraph of Polybius, discussed in Chapter 1. List at least three cases of incompleteness in the protocol.

The five elements are listed in section 2.2 of the text book. Generally the service to be performed is the delivery of a character stream. The environment assumptions define the visibility of the torches (distance, obstructions, and anything that would make the torches not

readable). The vocabulary is the tuple definition. The encoding is the order of tuple presentation (synchronization, left then right). The procedure rules define the initial setup (send synchronization, receive synchronization) and little else. This is the area of problems. See problem 4 for a suggestion of residual problems.

7. (Holtzmann 2-6) Explain the difference between bit stuffing and character stuffing.

While they are fundamentally the same, the definition depends upon the perspective of the transmission stream. If one considers the transmission as a stream of characters, one would stuff characters; that is, any insertion or deletion must be a character or multiple characters. If one considers the transmission as a stream of bits, on would stuff bits. Character stuffing can be done by software and normally is done by software. Bit stuffing is difficult for software and normally done in hardware.

8. (Stallings 2.2) List the major disadvantages with the layered approach to protocols.

The disadvantages deal with inefficiency. Inefficiency could be interjected by the duplication of functions; for example, reliability could be an active feature of two layers. Inefficiency could be interjected by the inclusion of unnecessary functions; for example, a layer might supply reliability that a higher layer does not use.

9. (Tanenbaum 1.11) What are two reasons for using layered protocols?

There are multiple reasons. The two traditional ones are isolation of functionality and "building blocks".

A frame encapsulates a packet (or part of a packet).

10. (Tanenbaum 1.24) The Internet is roughly doubling in size every 18 months. Although no one really knows for sure, one estimate put the number of hosts on it at 100 million in 2001. Use these data to compute the expected number of Internet hosts in the year 2010. Do you believe this? Explain why or why not.

What will 6.4 G hosts do in the year 2010? There is lots of room for discussion. One side deals with the concentration of Internet access points; there are enough access points in many parts of the world. The other side of the issue deals with the large number of Internet appliances that each person would have.

11. (Tanenbaum 1.36) Go to IETF's Web site, www.ietf.org, to see what they are doing. Pick a project you like and write a half-page report on the problem and the proposed solution.

(I am looking for thought; there is no unique correct answer.)