

# Homework Problems and Questions

## Chapter 3 Review Questions

### SECTIONS 3.1–3.3

R1. Suppose the network layer provides the following service. The network layer in the source host accepts a segment of maximum size 1,200 bytes and a destination host address from the transport layer. The network layer then guarantees to deliver the segment to the transport layer at the destination host. Suppose many network application processes can be running at the destination host.

- a. Design the simplest possible transport-layer protocol that will get application data to the desired process at the destination host. Assume the operating system in the destination host has assigned a 4-byte port number to each running application process.
- b. Modify this protocol so that it provides a “return address” to the destination process.
- c. In your protocols, does the transport layer “have to do anything” in the core of the computer network?

R2. Consider a planet where everyone belongs to a family of six, every family lives in its own house, each house has a unique address, and each person in a given house has a unique name. Suppose this planet has a mail service that delivers letters from source house to destination house. The mail service requires that (1) the letter be in an envelope, and that (2) the address of the destination house (and nothing more) be clearly written on the envelope. Suppose each family has a delegate family member who collects and distributes letters for the other family members. The letters do not necessarily provide any indication of the recipients of the letters.

- a. Using the solution to Problem R1 above as inspiration, describe a protocol that the delegates can use to deliver letters from a sending family member to a receiving family member.
- b. In your protocol, does the mail service ever have to open the envelope and examine the letter in order to provide its service?

**R3.** Consider a TCP connection between Host A and Host B. Suppose that the TCP segments traveling from Host A to Host B have source port number  $x$  and destination port number  $y$ . What are the source and destination port numbers for the segments traveling from Host B to Host A?

R4. Describe why an application developer might choose to run an application over UDP rather than TCP.

R5. Why is it that voice and video traffic is often sent over TCP rather than UDP in today's Internet? (*Hint: The answer we are looking for has nothing to do with TCP's congestion-control mechanism.*)

R6. Is it possible for an application to enjoy reliable data transfer even when the application runs over UDP? If so, how?

R7. Suppose a process in Host C has a UDP socket with port number 6789. Suppose both Host A and Host B each send a UDP segment to Host C with destination port number 6789. Will both of these segments be directed to the same socket at Host C? If so, how will the process at Host C know that these two segments originated from two different hosts?

R8. Suppose that a Web server runs in Host C on port 80. Suppose this Web server uses persistent connections, and is currently receiving requests from two different Hosts, A and B. Are all of the requests being sent through the same socket at Host C? If they are being passed through different sockets, do both of the sockets have port 80? Discuss and explain.

### SECTION 3.4

R9. In our *rdt* protocols, why did we need to introduce sequence numbers?

R10. In our *rdt* protocols, why did we need to introduce timers?

R11. Suppose that the roundtrip delay between sender and receiver is constant and known to the sender. Would a timer still be necessary in protocol *rdt 3.0*, assuming that packets can be lost? Explain.

R12. Visit the Go-Back-N Java applet at the companion Web site.

- Have the source send five packets, and then pause the animation before any of the five packets reach the destination. Then kill the first packet and resume the animation. Describe what happens.
- Repeat the experiment, but now let the first packet reach the destination and kill the first acknowledgment. Describe again what happens.
- Finally, try sending six packets. What happens?

R13. Repeat R12, but now with the Selective Repeat Java applet. How are Selective Repeat and Go-Back-N different?

### SECTION 3.5

R14. True or false?

- Host A is sending Host B a large file over a TCP connection. Assume Host B has no data to send Host A. Host B will not send acknowledgments to Host A because Host B cannot piggyback the acknowledgments on data.

- b. The size of the TCP *rwnd* never changes throughout the duration of the connection.
- c. Suppose Host A is sending Host B a large file over a TCP connection. The number of unacknowledged bytes that A sends cannot exceed the size of the receive buffer.
- d. Suppose Host A is sending a large file to Host B over a TCP connection. If the sequence number for a segment of this connection is  $m$ , then the sequence number for the subsequent segment will necessarily be  $m+1$ .
- e. The TCP segment has a field in its header for *rwnd*.
- f. Suppose that the last *SampleRTT* in a TCP connection is equal to 1 sec. The current value of *TimeoutInterval* for the connection will necessarily be  $\geq 1$  sec.
- g. Suppose Host A sends one segment with sequence number 38 and 4 bytes of data over a TCP connection to Host B. In this same segment the acknowledgment number is necessarily 42.

R15. Suppose Host A sends two TCP segments back to back to Host B over a TCP connection. The first segment has sequence number 90; the second has sequence number 110.

- a. How much data is in the first segment?
- b. Suppose that the first segment is lost but the second segment arrives at B. In the acknowledgment that Host B sends to Host A, what will be the acknowledgment number?

R16. Consider the Telnet example discussed in [Section 3.5](#). A few seconds after the user types the letter 'C,' the user types the letter 'R.' After typing the letter 'R,' how many segments are sent, and what is put in the sequence number and acknowledgment fields of the segments?

## SECTION 3.7

R17. Suppose two TCP connections are present over some bottleneck link of rate  $R$  bps. Both connections have a huge file to send (in the same direction over the bottleneck link). The transmissions of the files start at the same time. What transmission rate would TCP like to give to each of the connections?

R18. True or false? Consider congestion control in TCP. When the timer expires at the sender, the value of *ssthresh* is set to one half of its previous value.

R19. In the discussion of TCP splitting in the sidebar in [Section 3.7](#), it was claimed that the response time with TCP splitting is approximately  $4 \cdot RTT_{FE} + RTT_{BE} + \text{processing time}$ . Justify this claim.

## Problems

**P1.** Suppose Client A initiates a Telnet session with Server S. At about the same time, Client B

also initiates a Telnet session with Server S. Provide possible source and destination port numbers for

- a. The segments sent from A to S.
- b. The segments sent from B to S.
- c. The segments sent from S to A.
- d. The segments sent from S to B.
- e. If A and B are different hosts, is it possible that the source port number in the segments from A to S is the same as that from B to S?
- f. How about if they are the same host?

P2. Consider [Figure 3.5](#) . What are the source and destination port values in the segments flowing from the server back to the clients' processes? What are the IP addresses in the network-layer datagrams carrying the transport-layer segments?

P3. UDP and TCP use 1s complement for their checksums. Suppose you have the following three 8-bit bytes: 01010011, 01100110, 01110100. What is the 1s complement of the sum of these 8-bit bytes? (Note that although UDP and TCP use 16-bit words in computing the checksum, for this problem you are being asked to consider 8-bit sums.) Show all work. Why is it that UDP takes the 1s complement of the sum; that is, why not just use the sum? With the 1s complement scheme, how does the receiver detect errors? Is it possible that a 1-bit error will go undetected? How about a 2-bit error?

P4.

- a. Suppose you have the following 2 bytes: 01011100 and 01100101. What is the 1s complement of the sum of these 2 bytes?
- b. Suppose you have the following 2 bytes: 11011010 and 01100101. What is the 1s complement of the sum of these 2 bytes?
- c. For the bytes in part (a), give an example where one bit is flipped in each of the 2 bytes and yet the 1s complement doesn't change.

P5. Suppose that the UDP receiver computes the Internet checksum for the received UDP segment and finds that it matches the value carried in the checksum field. Can the receiver be absolutely certain that no bit errors have occurred? Explain.

P6. Consider our motivation for correcting protocol [rdt2.1](#) . Show that the receiver, shown in [Figure 3.57](#) , when operating with the sender shown in [Figure 3.11](#) , can lead the sender and receiver to enter into a deadlock state, where each is waiting for an event that will never occur.

P7. In protocol [rdt3.0](#) , the ACK packets flowing from the receiver to the sender do not have sequence numbers (although they do have an ACK field that contains the sequence number of the packet they are acknowledging). Why is it that our ACK packets do not require sequence numbers?