CHAPTER 12

TCP

12.1 MULTIPLE-CHOICE QUESTIONS

1. c	2. a	3. a	4. d	5. b	6. c	7. b	8. b	9. a	10. d
11. c	12. c	13. c	14. d	15. a	16. a	17. b	18. d	19. b	20. a
21. d	22. c	23. b	24. a	25. d	26. a	27. b	28. d	29. c	30. a
31. b	32. c	33. c	34. b	35. b					

12.2 EXERCISES

36. RTT = 90% $\times$ 4 + 10% $\times$ 5 = 4.1 seconds
37. The RTT is not recalculated because, according to Karn's algorithm, it is only updated based on an acknowledged segment, not a retransmitted one..
38. The maximum size of the TCP header is 60 bytes.
39. The minimum size of the TCP header is 20 bytes.
40. The segment would include 8 bytes of options.
41.
    a. None of the control bits are set. The segment is part of a normal communication.
    b. The FIN bit is set. This is a request to terminate the connection.
    c. The ACK and the FIN bits are set. This is an acknowledgment of data received and a simultaneous request to close the connection in the other direction.
    d. The RST bit is set. The connection must be reset.
    e. The SYN bit is set. The client wishes to establish a connection with the server. The segment includes initialization information about the client end of the connection.
    f. The ACK and the SYN bits are set. This segment is sent in response to the segment in part e. It serves 2 purposes: it acknowledges the receipt of the connec-

tion request and includes initialization information about the server end of the connection.

42. The first 3 bits could be used at the same time while the last 3 bits would only be used one at a time. There are 8 permutations of the first 3 bits. These, with any one or none of the last 3 bits gives us 32 (4×8) possible combinations. The URG and PSH bits would never be used with the SYN bit which removes 3 possibilities (URG - SYN, PSH - SYN, and URG - PSH - SYN). So there are 29 viable combinations of control bits.
43. See Figure 12.1.

Figure 12.1 *Exercise 43*

Client's ephemeral port number			Server's well-known port number
14534			
0			
5	0	000010	Window size
Checksum			0

a. Segment 1: client to server

Server's newly generated ephemeral port number			Client's ephemeral port number
21732			
14535			
5	0	010010	Window size
Checksum			0

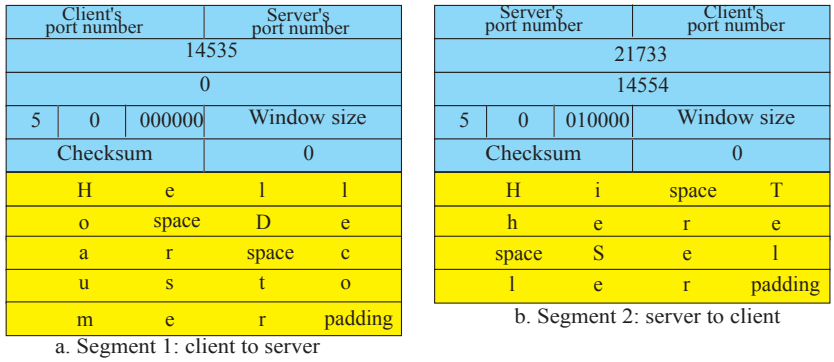
b. Segment 2: server to client

Client's ephemeral port number			Server's ephemeral port number
14535			
21733			
5	0	010000	Window size
Checksum			0

c. Segment 3: client to server

44. See Figure 12.2.
45. See Figure 12.3.
46. If we assume that 1 Megabyte is 1 million bytes (not 2^{20} bytes), the answer is (2^{31} - 7000) /1000,000 seconds or approximately 71 minutes.
47. See Figure 12.4.
48. See Figure 12.5.
49. See Figure 12.6.

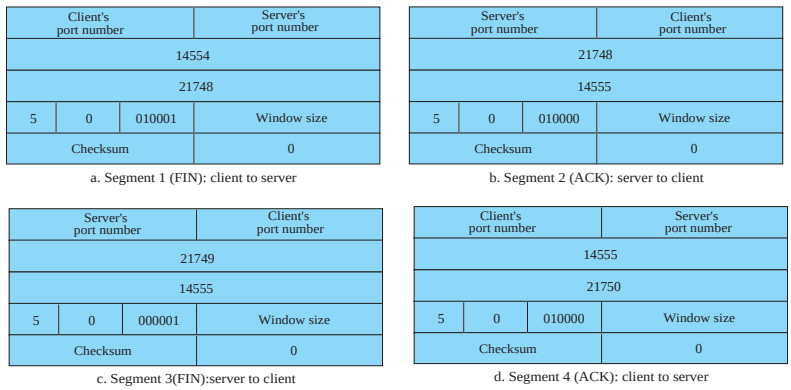
Figure 12.2 *Exercise 44*



a. Segment 1: client to server

b. Segment 2: server to client

Figure 12.3 *Exercise 45*



a. Segment 1 (FIN): client to server

b. Segment 2 (ACK): server to client

c. Segment 3(FIN):server to client

d. Segment 4 (ACK): client to server

50. 8000 bytes.

51.

a. The only side of the connection that can receive a FIN segment while it is in the ESTABLISHED state is the server, so we are talking about the server. When the
FIN segment is received, the server sends an ACK segment to the client and moves to the CLOSE-WAIT state.

b. When the “close” message is received from the application, the server sends a FIN segment to the client, moves to the LAST-ACK state and waits for the last acknowledgment to arrive from the client.

52.

a. The only side of the connection that can receive a “close” message while it is in the ESTABLISHED state is the client, so we are talking about the client. When

Figure 12.4 *Exercise 47*

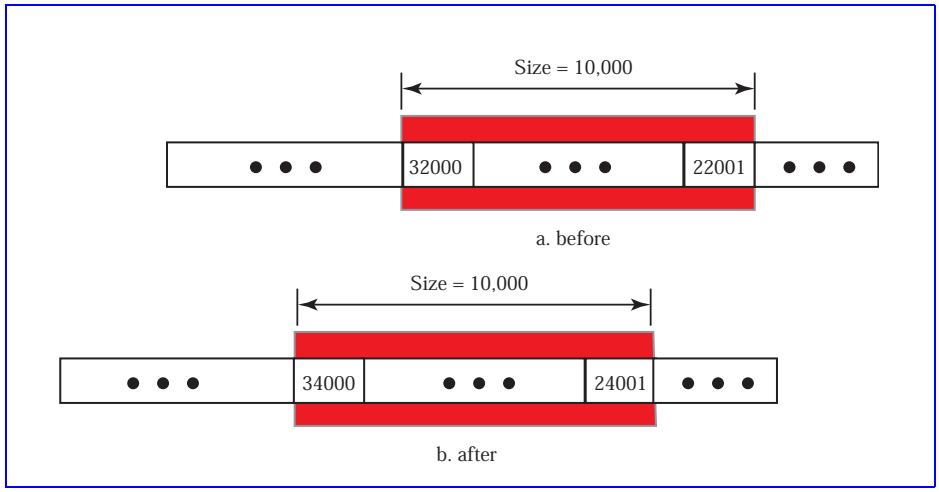
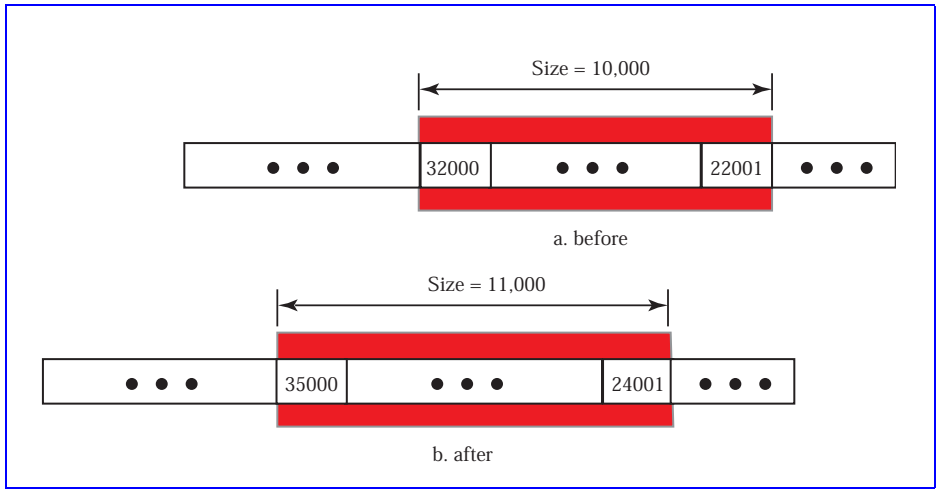


Figure 12.5 *Exercise 48*



the client receives the “close” message from the application, it sends a FIN segment to the server and moves to the FIN-WAIT-1 state.

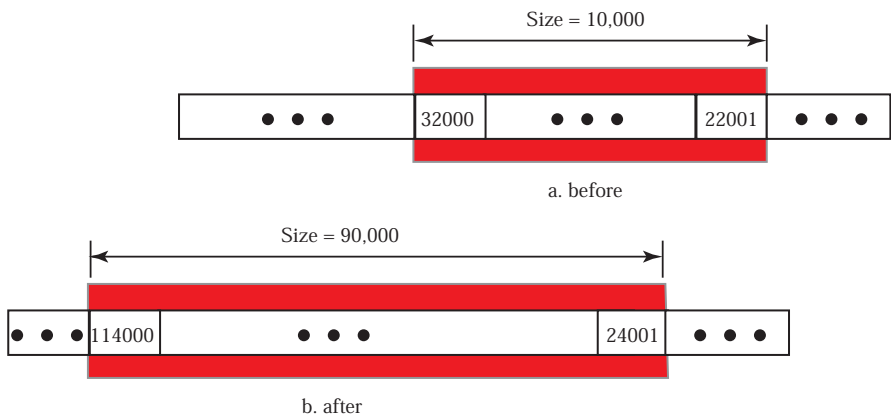
b. When the client receives an ACK segment from the server, it moves to the FIN-WAIT-2 state, and waits for the server to send a FIN segment.

53.

a. The only side of the connection that can receive a "close" message while in the SYN-RCVD state is the client, so we are talking about the client. When the client receives the "close" message from the application, it sends a FIN segment to the server and moves to the FIN-WAIT-1 state.

b. When the client receives the FIN segment from the server, it sends an ACK segment and moves to the CLOSING state.

Figure 12.6 *Exercise 49*



54.

a. The only side of the connection that cab be in SYN-SENT state is the client, so we are talking about the client. When the client receives the SYN+ACK segment from the server, it sends an ACK segment and moves to the ESTABLISHED state.

b. When the client receives the CLOSE message, it sends a FIN segment to the server and moves to the FIN-WAIT-1 state.

55.

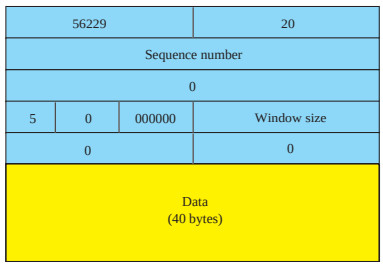
a. When the ACK segment is received, TCP sends nothing and moves from the FIN-WAIT-1 state to the FIN-WAIT-2 state.

b. When the FIN segment is received, TCP sends an ACK segment and moves to the TIME-WAIT state.

c. When the timeout occurs, TCP moves to the CLOSED state.

56. See Figure 12.7.

Figure 12.7 *Exercise 56*



57. 16 bytes of data / 36 bytes of header and data = 0.444
58. 16 bytes of data / 56 bytes of headers and data = 0.286
59. 16 bytes of data / (56 bytes of TCP/IP header and data + 19 bytes of Ethernet overhead) = 0.213
60.
    a. 1330
    b. 23
    c. 1
    d. 0
    e. 20 bytes
    f. A SYN segment used for connection establishment
    g. 2047 bytes