## CHAPTER 12 TCP

## **12.1 MULTIPLE-CHOICE QUESTIONS**

1. c	3. a	5. b	7. b	9. a
11. c	13. c	15. a	17. b	19. b
21. d	23. b	25. d	27. b	29. c
31. b	33. c	35. b		

## **12.2 EXERCISES**

- 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..
- 39. The minimum size of the TCP header is 20 bytes.

## 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 connection request and includes initialization information about the server end of the connection.
- 43. See Figure 12.1.

Cl	ient's epher port numb	neral er	Server's well-known port number		
		145	534		
		(			
5	0	000010	Window size		
	Checksu	ım	0		
	a. 9	Segment 1:	client to server		
Serv	er's newly g emeral port	generated number	Client's ephemeral port number		
		217	732		
		145	535		
5	0	010010	Window size		
	Checksu	ım	0		
	b. 1	Segment 2:	server to client		
Cl	ient's epher port numb	neral er	Server's ephemeral port number		
		145	535		
		21	733		
5	0	010000	Window size		
	Checksu	ım	0		
	c. (	Segment 3:	client to server		

Figure 12.1 Exercise 43

**45**. See Figure 12.2.

Figure 12.2 Exercise 45

	Client's port numbe	r	Server's port number		Server's port number			Client's port number
		145	54		21748			
21748			14555					
5	0	010001	Window size		5	0	010000	Window size
	Checksu	m	0			Checksu	ım	0
	a. Se	amont 1 (EIN	De all'end de la more			1.0		
		gment i (Piiv	): client to server			b. Se	gment 2 (AC	K): server to client
	Server port nur	r's	Client's port number			D. Se Client's port num	6	K): server to client Server's port number
	Server	r's nber	Client's			Client's	ber	Server's
	Server	r's nber 21'	Client's port number			Client's	s ber 14	Server's port number
5	Server	r's nber 21'	Client's port number 749	-	5	Client's	s ber 14	Server's port number

**47**. See Figure 12.3.

**49**. See Figure 12.4.

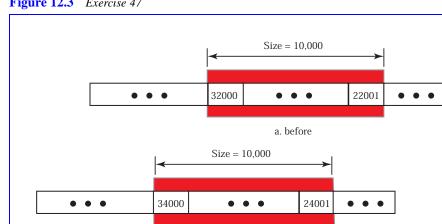
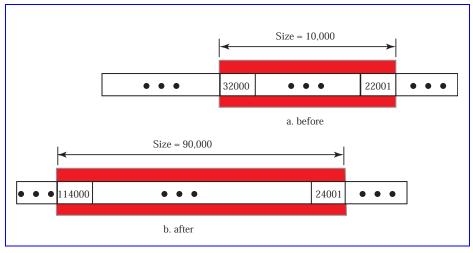


Figure 12.3 Exercise 47





b. after

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.

- 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.
- 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.
- 57. 16 bytes of data / 36 bytes of header and data = 0.444
- 59. 16 bytes of data / (56 bytes of TCP/IP header and data + 19 bytes of Ethernet overhead) = 0.213