CHAPTER 15

Transmission Control Protocol (TCP)

Exercises

1. See Table 15.E1.

Table 15.E1 *Solution to Exercise 1*

Fields	***UDP***	***TCP***	***Purpose***
Source Port Number	√	√	To define the source port number
Destination Port Number	√	√	To define the destination port number
Checksum	√	√	For error control
Total Length	√		It is not actually needed even in UDP
Sequence Number		√	For flow control
Acknowledgment Number		√	For flow control
Header Length		√	To define variable header length in TCP
Control Bits		√	To define different type of segments
Urgent Pointer		√	To define the end of urgent data
Options And Padding		√	To make TCP to use different options

Note that the only field that exists in UDP, but is missing in TCP, is the *total length* field. The designer of TCP did not feel that this field is needed because the size of the TCP segment can be determined from the size of the IP datagram that carries it.

3. The port is not listed in the transmission control block, which means no process is running associated to this port.

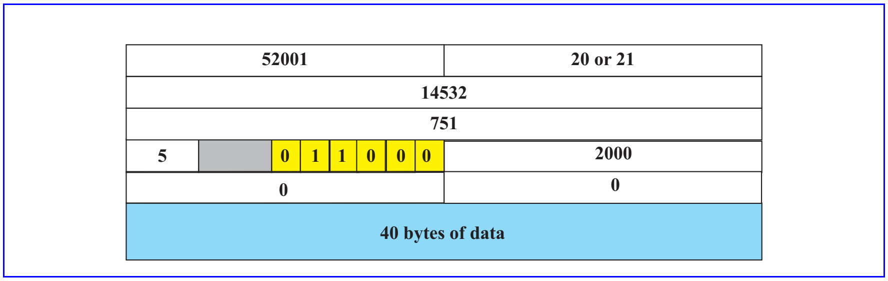
5.

a. The maximum size of the TCP header is **60** bytes (20 bytes of header and a maximum 40 bytes of options).

b. The minimum size of the TCP header is **20** bytes.

7. See Figure 15.E7.

Figure 15.E7 *Solution to Exercise 7*



9.

a. The source port number is 0532 in hex and **1330** in decimal.

b. The destination port number is 0017 in hex and **23** in decimal.

c. The sequence number is 00000001 in hex and **1** in decimal.

d. The acknowledgment number is 00000000 in hex and **0** in decimal.

e. The HLEN = 5. The header is 5 × 4 = 20 bytes long.

f. The control field is 002 in hex. This indicates a **SYN** segment used for connection establishment.

g. The window size field is 07FF in hex and 2047 in decimal. The window size is **2047** bytes.

11. Every second the counter is increment ed by 64,000 × 2 = 128,000. The sequence number field is 32 bits long and can hold only 2^{32}–1. So it takes (2^{32}–1)/(128,000) or **33,554** seconds to wrap around.

13. The window size is the smaller (3000, 5000) = 3000. Since 2000 bytes is already sent, only 3000 – 2000 = **1000** more bytes can be sent.

15. The data section is only 16 bytes. The TCP header is 20 bytes. IP header is 20 bytes. The header and trailer is 19 bytes (without preamble). We can calculate the efficiency at each layer:

a. At TCP level:

(16) / (16 + 20) = 0.444 → **44.4%**

b. At IP level:

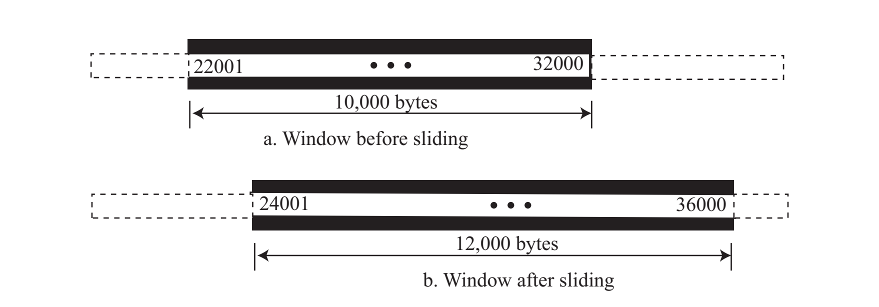
(16) / (16 + 20 + 20) = 28.57 → **28.57%**

c. At data link level:

(16) / (16 + 20 + 20 + 19) = 0.2133 → **21.33%**

17. See Figure 15.E17.

Figure 15.E17 *Solution to Exercise 17*



19.

a. The server can receive a FIN segment while it is in the ESTABLISHED state. 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 client TCP sends a FIN segment; the client goes to the FIN-WAIT-1 state and waits for an ACK.

21.

a. No ACK needed at 0:0:0:0:000, according to Rule 2

b. No ACK needed at 0:0:0:0:027, according to Rule 2

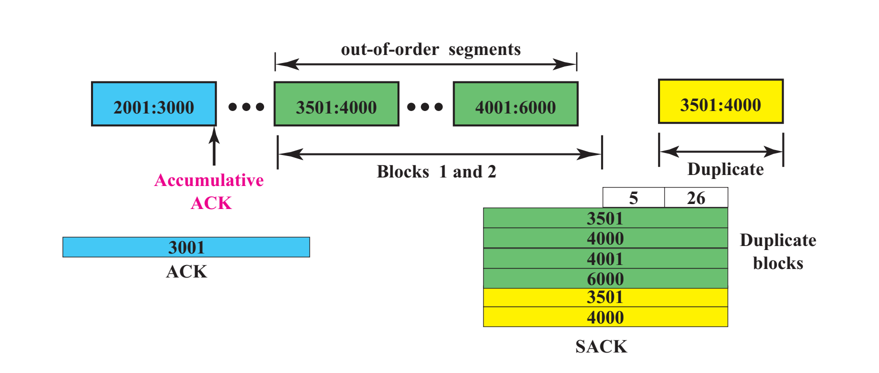
c. ACK: 4 can be sent at 0:0:0:0:500

d. ACK: 5 can be sent at 0:0:0:1:200

e. No ACK needed at 0:0:0:1:208, according to Rule 2

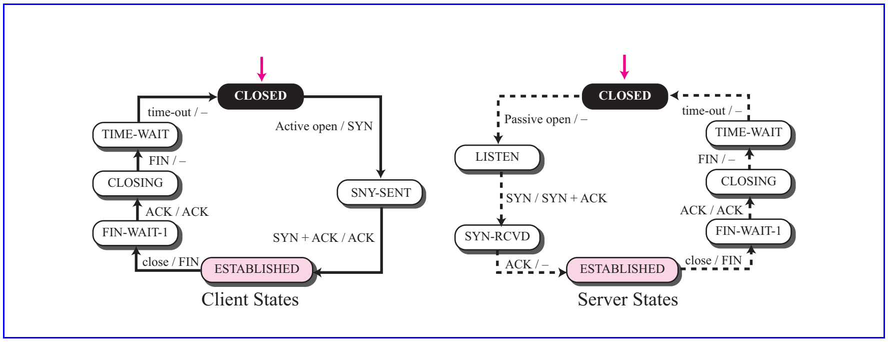
23. See Figure 15.E23.

Figure 15.E23 *Solution to Exercise 23*



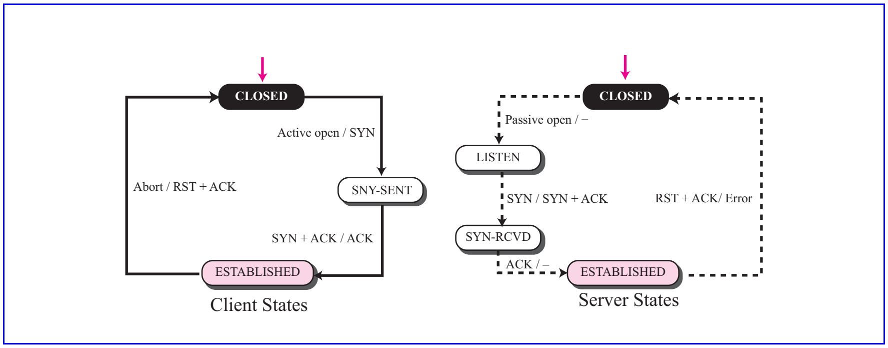
25. See Figure 15.E25.

Figure 15.E25 *Solution to Exercise 25*



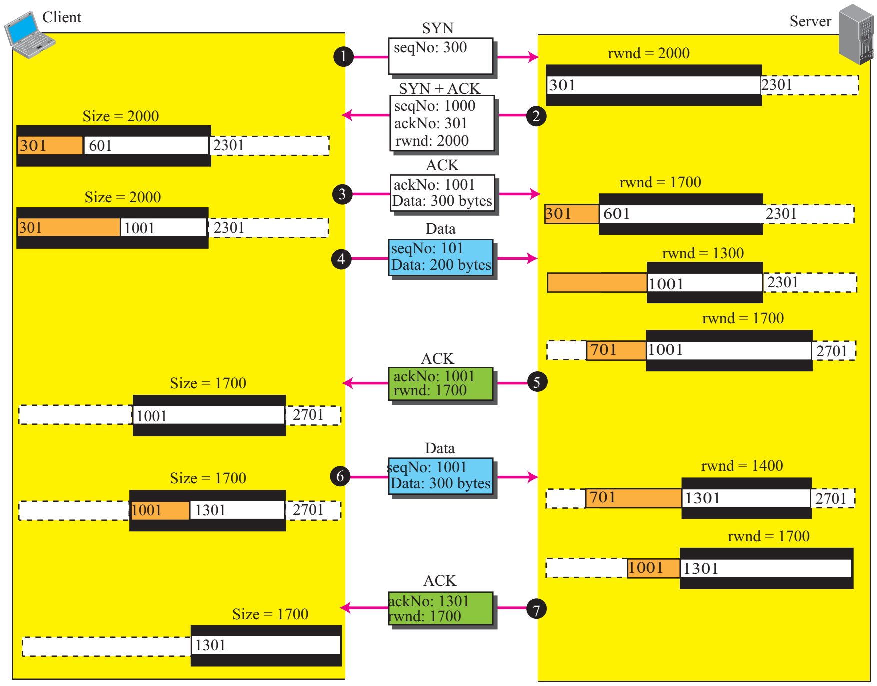
27. See Figure 15.E27.

Figure 15.E27 *Solution to Exercise 27*



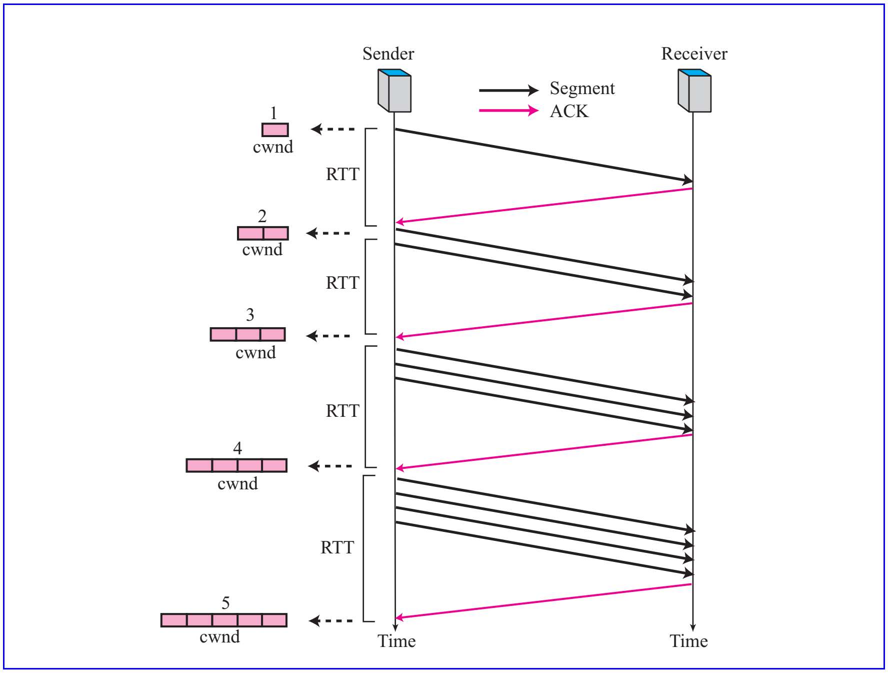
29. See Figure 15.E29.

Figure 15.E29 *Solution to Exercise 29*



31. See Figure 15.E31.

Figure 15.E31 *Solution to Exercise 31*


Sender
Receiver
Segment
ACK
1
cwnd
RTT
2
cwnd
RTT
3
cwnd
RTT
4
cwnd
RTT
5
cwnd
Time
Time