

第三章

R3: 源端口号 y. 目的端口号 x

R7: 是的, 两个段将指向同一套程序, 对于每个接收到的段, 在套接字接口上, 操作系统将为进程提供 IP 地址, 以确定各个段的来源.

R14: a. F. b. F c. T d. F
e. T f. F g. F

R15: a. 20 bit
b. ack number = 90.

P1: SPort DPort

a. A → S 467 23

b. B → S 513 23

c. S → A 23 467

d. S → B 23 513

e. yes. f. no

P2: 假设主机 A、B、C 的 IP 地址分别是 a、b、c

主机 A: 源端口 = 80. 源 IP 地址 = b. 目标端口 = 26145. IP = a.

c. 左 80 b 7532 c

c. 右 80 b 26145 c

P3: 将 8 比特符号和校验和相加, 结果全 1

$$\begin{array}{r} 01010011 \\ + 01100110 \\ \hline 1011001 \\ + 0110100 \\ \hline 00101101 \\ + 1 \\ \hline 00101110 \end{array}$$

和
校验和: 11010001

不可能.

可能

P4: a. 00111110
b. 10111111
c. 00101000 01101101

P5: 不能。如果包中两个16位字的对应位相加在一起是0和1,那么即使这些位分别翻转到1和0,和仍然保持不变。

P7: 对于接收器来说,一个重复的ACK是显而易见的。因为它接收并存储ACK时,它会转换到下一阶段。

P8: 略。

P11: 发送和接收将发生死锁。

P14: 不是,因为需要超时检测才能确认发送成功。
是,可以减少数据流量。
因为接收方判断丢失是依据数据包的上下文。
当丢失的包的下一个包被接收时,才会发现丢包。

P22: a. ① $k-1$ 及其之前都未到达 $k-4, k-3, k-2, k-1$
② $k-1$ 到达 $k, k+1, k+2, k+3$
b. $k-4, k-3, k-2, k-1$

P24: a. 可能。有一分组的ack还未到达,发送方就被判定为超时重发。
当它到达时,发送方窗口移动,刚才重发的分组ack就落在窗口外。
b. 可能。
c. 是。 d. 是。

P27: a. 序号207, 源端口号302, 目的端口号80
b. 序号207, 80, 302
c. 127
d. 略。

P28: 缓冲区已满时,主机B通过设置 $rcvWindow = 0$ 向主机A发出停止发送数据的信号。

P31: $EstimateRTT = (1-\alpha) EstimateRTT + \alpha SampleRTT$
 $DevRTT = (1-\beta) * DevRTT + \beta | SampleRTT - EstimateRTT$

第T ~~$ERTT = 106ms$~~

~~$DRTT = 0.75 \times 0 + 0.25 \times 0 = 0$~~

$ERTT = 87.5ms + 0.125 \times 106ms$

$DRTT = 0.75 \times 5 + 0.25 \times 6ms$

$TimeoutInterval = 100.75ms + 4 \times 5.15ms$

P32: 略

P39: 略