

1.

- P1. 假设客户 A 向服务器 S 发起一个 Telnet 会话。与此同时，客户 B 也向服务器 S 发起一个 Telnet 会话。给出下面报文段的源端口号和目的端口号：
- 从 A 向 S 发送的报文段。
 - 从 B 向 S 发送的报文段。
 - 从 S 向 A 发送的报文段。
 - 从 S 向 B 发送的报文段。
 - 如果 A 和 B 是不同的主机，那么从 A 向 S 发送的报文段的源端口号是否可能与从 B 向 S 发送的报文段的源端口号相同？
 - 如果它们是同一台主机，情况会怎么样？

假设 A 端口号为 101，B 端口号为 102，telnet 使用 22 端口

- 101 -> 22
- 102 -> 22
- 22 -> 101
- 22 -> 102
- 若 A, B 是不同主机，则源端口序号可能相同
- 若 A, B 是相同主机。则源端口序号不可能相同

2.

UDP 和 TCP 使用反码来计算它们的检验和。假设你有下面 3 个 8 比特字节：01010011，01100110，01110100。这些 8 比特字节和的反码是多少？（注意到尽管 UDP 和 TCP 使用 16 比特的字来计算检验和，但对于这个问题，你应该考虑 8 比特和。）写出所有工作过程。UDP 为什么要用该和的反码，即为什么不直接使用该和呢？使用该反码方案，接收方如何检测出差错？1 比特的差错将可能检测不出来吗？2 比特的差错呢？

$$01010011 + 01100110 + 01110100 = 00101110$$

反码：11010001

使用反码有以下好处：

- 不依赖系统是大端还是小端
- 计算检验和比较简单快速

接收方检验差错的方法是将三个字节与检验和相加，如果任何一个位为 0，说明出错

1 比特的差错肯定会导致结果不同

2 比特的差错可能会检测不出，比如题中第一、二字节变为 01010010，01100111，即最后一个比特反转

3.

- P4. a. 假定你有下列 2 个字节：01011100 和 01100101。这 2 个字节之和的反码是什么？
- b. 假定你有下列 2 个字节：11011010 和 01100101。这 2 个字节之和的反码是什么？
- c. 对于 (a) 中的字节，给出一个例子，使得这 2 个字节中的每一个都在一个比特反转时，其反码不会改变。

- a. $01011100 + 01100101 = 11000001$
反码: 00111110
- b. $11011010 + 01100101 = 01000000$
反码: 10111111
- c. 01101101 和 01010100

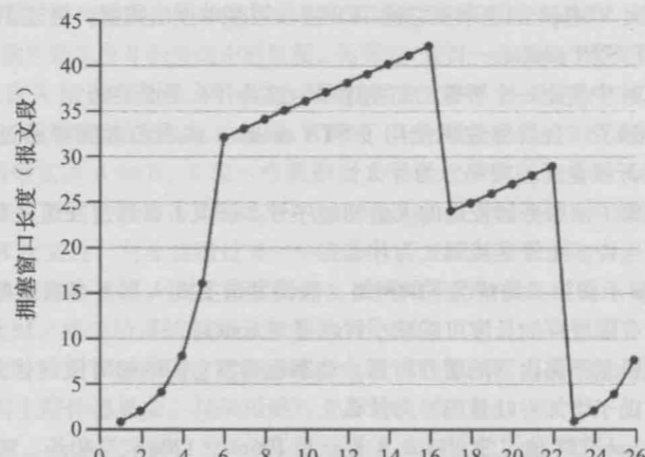
4.

P7. 在 rdt3.0 协议中, 从接收方向发送方流动的 ACK 分组没有序号 (尽管它们具有 ACK 字段, 该字段包括了它们正在确认的分组的序号)。为什么这些 ACK 分组不需要序号呢?

ACK 分组没有序号是因为接收方、发送方都不需要该序号

5.

P40. 考虑图 3-58。假设 TCP Reno 是一个经历如上所示行为的协议, 回答下列问题。在各种情况中, 简要地论证你的回答。



- a. 指出 TCP 慢启动运行时的时间间隔。
- b. 指出 TCP 拥塞避免运行时的时间间隔。
- c. 在第 16 个传输轮之后, 报文段的丢失是根据 3 个冗余 ACK 还是根据超时检测出来的?
- d. 在第 22 个传输轮之后, 报文段的丢失是根据 3 个冗余 ACK 还是根据超时检测出来的?
- e. 在第 1 个传输轮里, ssthresh 的初始值设置为多少?
- f. 在第 18 个传输轮里, ssthresh 的值设置为多少?
- g. 在第 24 个传输轮里, ssthresh 的值设置为多少?
- h. 在哪个传输轮内发送第 70 个报文段?
- i. 假定在第 26 个传输轮后, 通过收到 3 个冗余 ACK 检测出有分组丢失, 拥塞的窗口长度和 ssthresh 的值应当是多少?
- j. 假定使用 TCP Tahoe (而不是 TCP Reno), 并假定在第 16 个传输轮收到 3 个冗余 ACK。在第 19 个传输轮, ssthresh 和拥塞窗口长度是什么?
- k. 再次假设使用 TCP Tahoe, 在第 22 个传输轮有一个超时事件。从第 17 个传输轮到第 22 个传输轮 (包括这两个传输轮), 一共发送了多少分组?

a. [1, 6] 和 [23, 26]

b. [6, 16] 和 [17, 22]

c. 三个冗余 ACK

d. 超时

e. 32

f. $42/2 = 21$

g. $29/2 = 14$

h. 分组 1 (p1) 在传输轮回 1 (t1) 中发送, p2~p3 在 t2 中发送, p4~p7 在 t3 中发送, p8~p15 在 t4, p16~p31 在 t5, p32~p63 在 t6, p64~p96 在 t7, 因此分组 70 在第 7 个传输轮回内发送

i. $ssthresh = 4$, $cwnd = ssthresh + 3MSS = 7$

j. 在第 16 个传输轮回时, $ssthresh = cwnd/2 = 21$, $cwnd = 1$ 并进入慢启动状态, 则第 19 个传输轮回中, $ssthresh = 21$, $cwnd = 4$

k. t17: 1 个, t18: 2 个, t19: 4 个, t20: 8 个, t21: 16 个, t22: 21 个
 $1 + 2 + 4 + 8 + 16 + 21 = 52$ 个