

## 第三章习题

Ch3 R3 R7 R9 R10 R14 R15 R17 R18 P1 P3 P8 P15 P23 P24 P27 P32 P40 P45 P53

\*R3

R3. 考虑在主机 A 和主机 B 之间有一条 TCP 连接。假设从主机 A 传送到主机 B 的 TCP 报文段具有源端口号  $x$  和目的端口号  $y$ 。对于从主机 B 传送到主机 A 的报文段，源端口号和目的端口号分别是多少？

答：源端口号为  $y$ ，目的端口号为  $x$ 。

\*R7

R7. 假定在主机 C 上的一个进程有一个具有端口号 6789 的 UDP 套接字。假定主机 A 和主机 B 都用目的端口号 6789 向主机 C 发送一个 UDP 报文段。这两台主机的这些报文段在主机 C 都被描述为相同的套接字吗？如果是这样的话，在主机 C 的该进程将怎样知道源于两台不同主机的这两个报文段？

答：被描述为相同的套接字，根据源 IP 地址区分。

- 在接收端，UDP套接字用二元组标识 (目标IP地址、目标端口号)
- 当主机收到UDP报文段：
  - 检查报文段的目标端口号
  - 用该端口号将报文段定位给套接字
- 如果两个不同源IP地址/源端口号的数据报，但是有相同的目标IP地址和端口号，则被定位到相同的套接字

\*R9

R9. 在我们的 rdt 协议中，为什么需要引入序号？

答：序号可以用来区分到达的数据包是重传还是新数据，也可以用来判断数据包

顺序是否正确。

\*R10

R10. 在我们的 rdt 协议中，为什么需要引入定时器？

答：防止丢包；发送方发送分组后，启动定时器，如果定时结束前未收到 ACK，

则重传该分组，防止丢包现象发生。

R14. 是非判断题：

- a. 主机 A 经过一条 TCP 连接向主机 B 发送一个大文件。假设主机 B 没有数据发往主机 A。因为主机 B 不能随数据捎带确认，所以主机 B 将不向主机 A 发送确认。

\*R14

a) F.

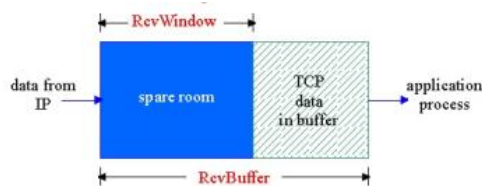
如果 B 没有数据发送给 A，B 将单独向 A 发送确认；

b. 在连接的整个过程中，TCP 的 rwnd 的长度决不会变化。

b) F.

课本 P165，接收窗口 rwnd 根据缓存可用空间的数量来设置，缓存可用空间随时间变化，rwnd 随时间变化,是动态的

$\text{rwnd} = \text{接收缓存} - \text{主机 B 中已接收但未读取的数据量}$



(假设TCP接收方丢弃乱序的报文段)

□ 缓存中的可用的空间

$= \text{RcvWindow}$

$= \text{RcvBuffer} - [\text{LastByteRcvd} - \text{LastByteRead}]$

- c. 假设主机 A 通过一条 TCP 连接向主机 B 发送一个大文件。主机 A 发送但未被确认的字节数不会超过接收缓存的大小。

c) T. 课本 P165

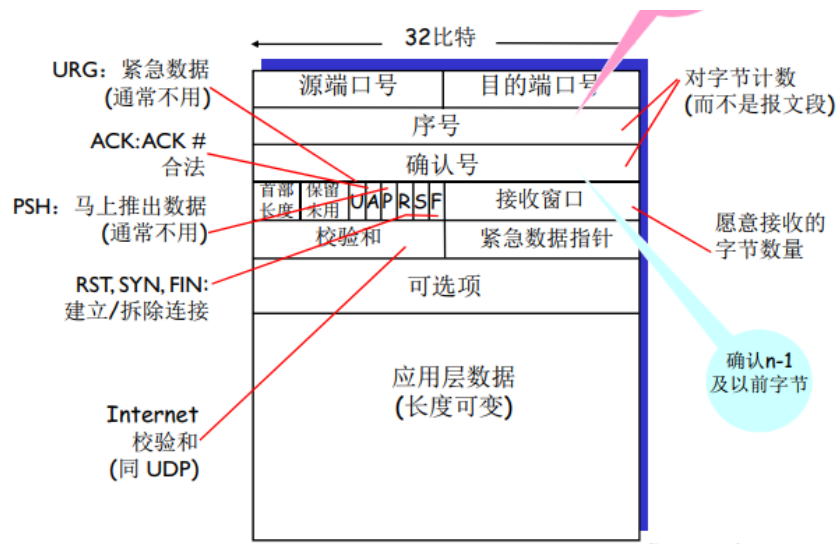
主机 A 轮流跟踪两个变量，LastByteSent 和 LastByteAcked，这两个变量的意义很明显。注意到这两个变量之间的差  $\text{LastByteSent} - \text{LastByteAcked}$ ，就是主机 A 发送到连接中但未被确认的数据量。通过将未确认的数据量控制在值 rwnd 以内，就可以保证主机 A 不会使主机 B 的接收缓存溢出。因此，主机 A 在该连接的整个生命周期须保证：

$$\text{LastByteSent} - \text{LastByteAcked} \leq \text{rwnd}$$

- d. 假设主机 A 通过一条 TCP 连接向主机 B 发送一个大文件。如果对于这条连接的一个报文段的序号为  $m$ ，则对于后继报文段的序号将必然是  $m+1$ 。
- e. TCP 报文段在它的首部中有一个 `rwnd` 字段。

d) F. TCP 连接为字节编号，后继报文段的序号为  $m$ +该段的字节数量

e) T 课本 p154 TCP 报文段结构



- f. 假定在一条 TCP 连接中最后的 `SampleRTT` 等于 1 秒，那么对于该连接的 `TimeoutInterval` 的当前值必定大于等于 1 秒。

f) F. 课本 P158-159

□ `SampleRTT` 会偏离 `EstimatedRTT` 多远:

$$\text{DevRTT} = (1-\beta) * \text{DevRTT} + \beta * |\text{SampleRTT} - \text{EstimatedRTT}|$$

(推荐值:  $\beta = 0.25$ )

超时时间间隔设置为:

$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$$



↑  
estimated RTT

↑  
"safety margin"

- g. 假设主机 A 通过一条 TCP 连接向主机 B 发送一个序号为 38 的 4 个字节的报文段。在这个相同的报文段中，确认号必定是 42。

g) F. 若 B 未成功收到序号为 38 的报文段，则确认号为 38

\*R15

R15. 假设主机 A 通过一条 TCP 连接向主机 B 发送两个紧接着的 TCP 报文段。第一个报文段的序号为 90, 第二个报文段序号为 110。

- 第一个报文段中有多少数据?
- 假设第一个报文段丢失而第二个报文段到达主机 B。那么在主机 B 发往主机 A 的确认报文中, 确认号应该是多少?

a.  $110 - 90 = 20$  bytes

b. 90 (ACK 为顺序收到的最后一个字节+1, 即为期望收到的字节号)

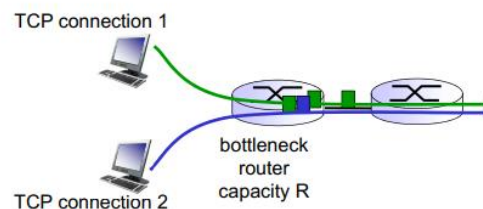
R17

R17. 假设两条 TCP 连接存在于一个带宽为  $R$  bps 的瓶颈链路上。它们都要发送一个很大的文件 (以相同方向经过瓶颈链路), 并且两者是同时开始发送文件。那么 TCP 将为每条连接分配什么样的传输速率?

答:  $R/2$

## TCP 公平性

公平性目标: 如果  $K$  个 TCP 会话分享一个链路带宽为  $R$  的瓶颈, 每一个会话的有效带宽为  $R/K$



\*R18

R18. 是非判断题。考虑 TCP 的拥塞控制。当发送方定时器超时, 其  $ssthresh$  的值将被设置为原来值的一半。

答: 错误, 当发送方定时器超时, 其  $ssthresh$  的值会被设置为当前拥塞窗口  $cwnd$  的一半。

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？

\*P1 答：

设 A 使用端口号为 x，B 使用端口号为 y，Telnet 使用端口号 23

A->S 源端口号：x；目的端口号：23

B->S 源端口号：y；目的端口号：23

S->A 源端口号：23；目的端口号：x

S->B 源端口号：23；目的端口号：y

若 A 和 B 是不同主机，则可以，因为能用 IP 地址区分报文。

若 A, B 是相同主机则不行，因为这样端口号和 IP 地址都相同，就无法区分报文了。

P3. 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？

\*P3 答：

```

  0 1 0 1 0 0 1 1
  0 1 1 0 0 1 1 0
  -----
  1 0 1 1 1 0 0 1
  ↓
  1 0 1 1 1 0 0 1
  0 1 1 1 0 1 0 0
  -----
  1 0 0 1 0 1 1 0 1
  ↘
  0 0 1 0 1 1 1 0
  -----
  1 1 0 1 0 0 0 1
  反码
  
```

①直接使用该和是做减法检测，而使用反码是做加法检测。而之前将 3 个数做相加操作，继续使用加法可以使得检测速度更快。

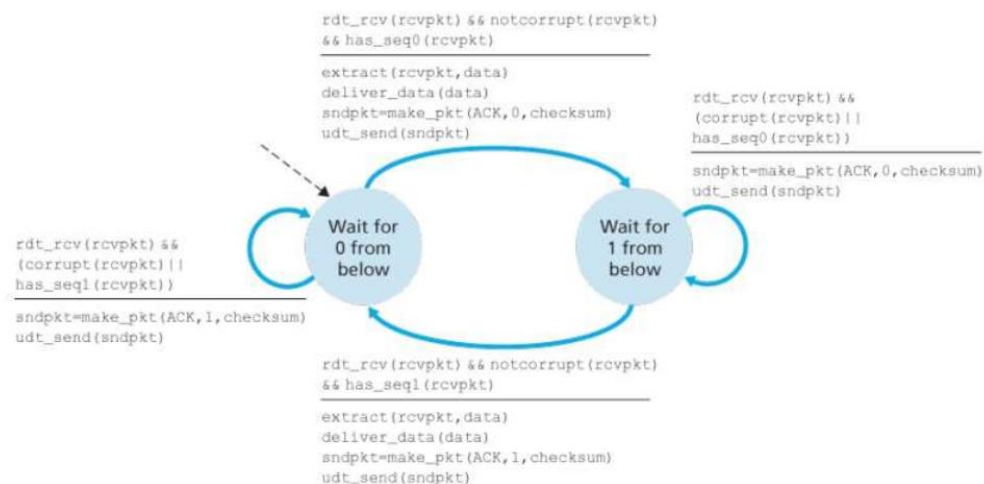
②错误检测方法，将四个数相加，正确时应出现 11111111，只要出现 0 则有错。

③1 比特差错可以检测，2 比特则不行。

(01010011 变为 01010010, 01100110 变为 01100111)。

\*P8.画出协议 rdt3.0 接收方中的 FSM.

答：rdt3.0 比 rdt2.2 多出超时重传机制，重传造成重复，rdt2.2 已引入序号机制，可以解决冗余数据分组问题，同 P140 rdt2.2 接收方



P15. 考虑显示在图 3-17 中的网络跨越国家的例子。窗口长度设置成多少时，才能使该信道的利用率超过 90%？假设分组的长度为 1500 字节（包括首部字段和数据）。

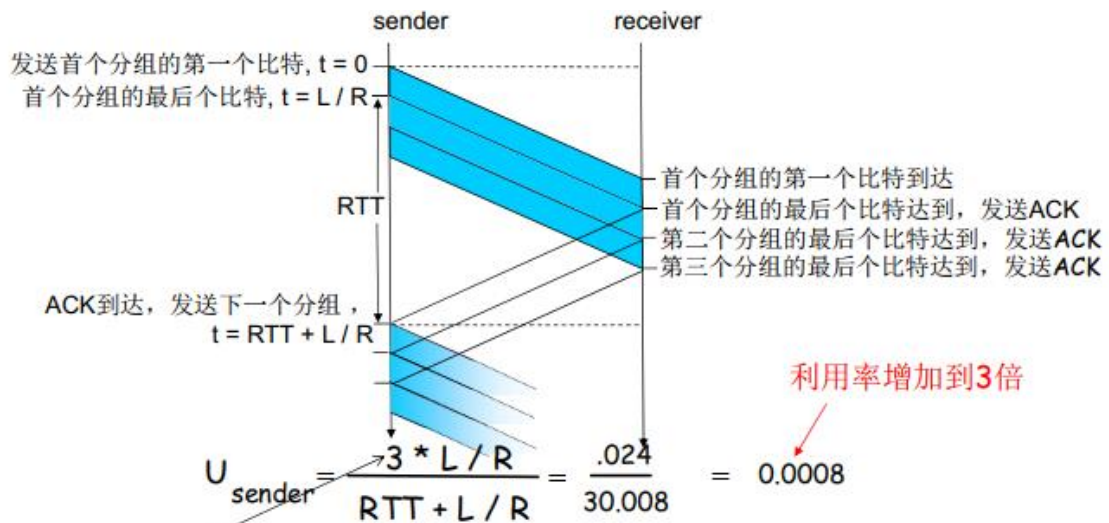
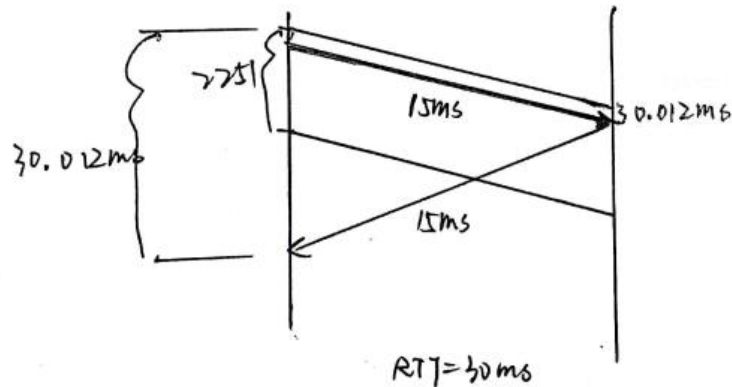
\*P15 答：

$$\text{传输时延 } t_{\text{trans}} = \frac{L}{R} = \frac{1500 \times 8 \text{ bit} / \text{pkt}}{10^9 \text{ bit} / \text{s}} = 12 \mu\text{s}, \text{ 窗口长度为 } n$$

$$U_{\text{sender}} = \frac{n * \frac{L}{R}}{RTT + \frac{L}{R}} = \frac{0.012n}{30.012} > 0.9 \quad (\text{课本 P143 公式})$$

$n > 2250.9$ ，窗口长度至少要 2251





- 增加n,能提高链路利用率
- 但当达到某个n,其 $u=100\%$ 时,无法再通过增加n, 提高利用率
- 瓶颈转移了->链路带宽

利用率增加到3倍

\*P23

P23. 考虑 GBN 协议和 SR 协议。假设序号空间的长度为  $k$ ，那么为了避免出现图 3-27 中的问题，对于这两种协议中的每一种，允许的发送方窗口最大为多少？

设序号字段的比特数为  $n$ ，则序号空间为  $2^n$ ，序号范围为  $[0, 2^n - 1]$

假设  $n=3$ ，则序号范围为  $[0, 7]$ ：

0 1 2 3 4 5 6 7 0 1 2 ...

对于 GBN，发送窗口最大为  $\omega_{\max} = 7$

对于 SR，发送窗口最大为  $\omega_{\max} = 4$

一般地：

对于 GBN，发送窗口最大为  $\omega_{\max} = 2^n - 1$

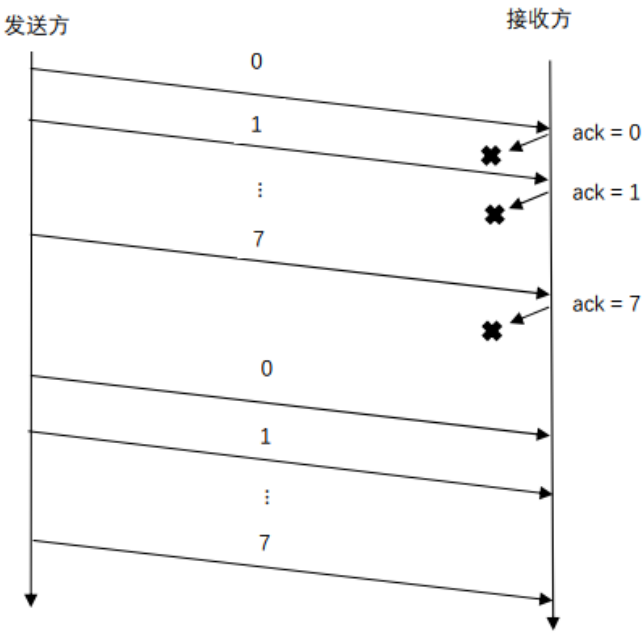
对于 SR，发送窗口最大为  $\omega_{\max} = 2^{n-1}$

本题中  $k$  为序号空间的长度，即：

对于 GBN，发送窗口最大为  $\omega_{\max} = k - 1$

对于 SR，发送窗口最大为  $\omega_{\max} = \lfloor k/2 \rfloor$

先看 GBN，假设发送窗口大小为 8，则：



接收方窗口：

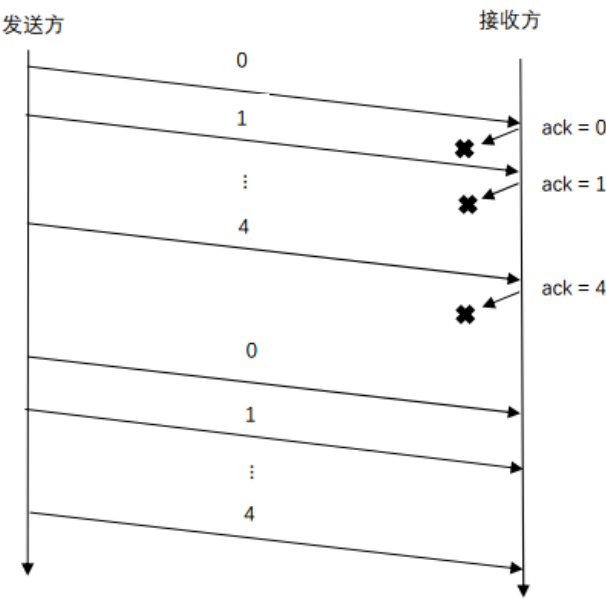
0 1 2 3 4 5 6 7 0 1 2 ...

发送方发送老的分组，接收方认为是新的分组，协议失败。  
如果发送窗口是 7，则：

0 1 2 3 4 5 6 7 0 1 2 ...

则不会出现这种情况。

对于 SR，假设发送窗口大小为 5，则：





接收方：

0 1 2 3 4 5 6 7 0 1 2 ...

老的窗口和新的窗口重叠, 发送方发送老的分组 0 和 1, 接收方认为是新的分组, 协议失败。

如果发送窗口是 4:

0 1 2 3 4 5 6 7 0 1 2 ...

则不会出现这种情况。

发送端滑动窗口的序号数目 + 接受端滑动窗口的序号数目  $\leq$  分组编号数目

对于 GBN, 发送窗口最大为  $\omega_{max} = 2^n - 1$

对于 SR, 发送窗口最大为  $\omega_{max} = 2^{n-1}$

本题中  $k$  为序号空间的长度, 即:

对于 GBN, 发送窗口最大为  $\omega_{max} = k - 1$

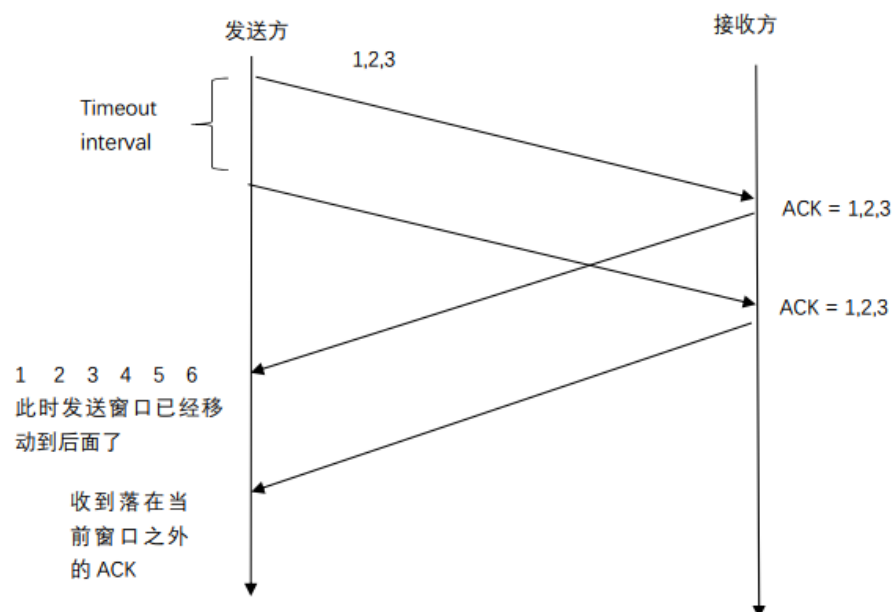
对于 SR, 发送窗口最大为  $\omega_{max} = \lfloor k/2 \rfloor$

\*P24

P24. 对下面的问题判断是非, 并简要地证实你的回答:

- a. 对于 SR 协议, 发送方可能会收到落在其当前窗口之外的分组的 ACK。
- b. 对于 GBN 协议, 发送方可能会收到落在其当前窗口之外的分组的 ACK。
- c. 当发送方和接收方窗口长度都为 1 时, 比特交替协议与 SR 协议相同。
- d. 当发送方和接收方窗口长度都为 1 时, 比特交替协议与 GBN 协议相同。

答: a, b, c, d 均正确。

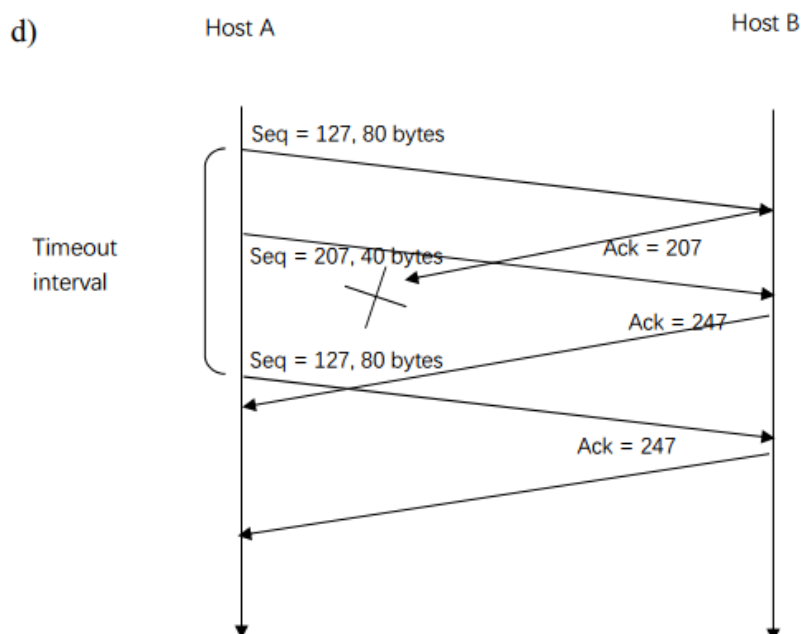


\*P27

P27. 主机 A 和 B 经一条 TCP 连接通信，并且主机 B 已经收到了来自 A 的最长为 126 字节的所有字节。假定主机 A 随后向主机 B 发送两个紧接着的报文段。第一个和第二个报文段分别包含了 80 字节和 40 字节的数据。在第一个报文段中，序号是 127，源端口号是 302，目的地端口号是 80。无论何时主机 B 接收到来自主机 A 的报文段，它都会发送确认。

- 在从主机 A 发往 B 的第二个报文段中，序号、源端口号和目的端口号各是什么？
- 如果第一个报文段在第二个报文段之前到达，在第一个到达报文段的确认中，确认号、源端口号和目的端口号各是什么？
- 如果第二个报文段在第一个报文段之前到达，在第一个到达报文段的确认中，确认号是什么？
- 假定由 A 发送的两个报文段按序到达 B。第一个确认丢失了而第二个确认在第一个超时间隔之后到达。画出时序图，显示这些报文段和发送的所有其他报文段和确认。（假设没有其他分组丢失。）对于图上每个报文段，标出序号和数据的字节数量；对于你增加的每个应答，标出确认号。

答：a) 序列号为 207，源端口号为 302，目的端口号为 80。  
 b) 确认号是 207，源端口号是 80，目的端口号是 302。  
 c) 确认号为 127，表示它仍在等待字节 127 及之后。



\*P32

P32. 考虑 TCP 估计 RTT 的过程。假设  $\alpha = 0.1$ ，令  $\text{SampleRTT}_1$  设置为最新样本 RTT，令  $\text{SampleRTT}_2$  设置为下一个最新样本 RTT，等等。

- 对于一个给定的 TCP 连接，假定 4 个确认报文相继到达，带有 4 个对应的 RTT 值： $\text{SampleRTT}_4$ 、 $\text{SampleRTT}_3$ 、 $\text{SampleRTT}_2$  和  $\text{SampleRTT}_1$ 。根据这 4 个样本 RTT 表示  $\text{EstimatedRTT}$ 。
- 将你得到的公式一般化到  $n$  个 RTT 样本的情况。
- 对于在 (b) 中得到的公式，令  $n$  趋于无穷。试说明为什么这个平均过程被称为指数移动平均。

答： 
$$\text{EstimatedRTT} = (1 - \alpha) \cdot \text{EstimatedRTT} + \alpha \cdot \text{SampleRTT}$$

以 SampleRTT1 为最新

a)

$$\begin{aligned} EstimatedRTT^{(4)} &= xSampleRTT_1 + \\ &\quad (1-x)[xSampleRTT_2 + \\ &\quad (1-x)[xSampleRTT_3 + (1-x)SampleRTT_4]] \\ &= xSampleRTT_1 + (1-x)xSampleRTT_2 \\ &\quad + (1-x)^2 xSampleRTT_3 + (1-x)^3 SampleRTT_4 \end{aligned}$$

代入  $x=0.1$  得:

$$EstimatedRTT^{(4)} = 0.1 SampleRTT_1 + 0.09 SampleRTT_2 + 0.081 SampleRTT_3 + 0.729 SampleRTT_4$$

b)

$$\begin{aligned} EstimatedRTT^{(n)} &= x \sum_{j=1}^{n-1} (1-x)^{j-1} SampleRTT_j \\ &\quad + (1-x)^{n-1} SampleRTT_n \end{aligned}$$

代入  $x=0.1$  得:

$$EstimatedRTT^{(n)} = 0.1 \times \sum_{j=1}^{n-1} (0.9)^{j-1} SampleRTT_j + (0.9)^{n-1} SampleRTT_n$$

c)

$$EstimatedRTT^{(\infty)} = \frac{x}{1-x} \sum_{j=1}^{\infty} (1-x)^j SampleRTT_j$$

代入  $x=0.1$  得:

$$\lim_{n \rightarrow \infty} EstimatedRTT^{(n)} = \frac{1}{9} \sum_{j=1}^{\infty} (0.9)^j SampleRTT_j$$

以 SampleRTT4 为最新，设初始为 SampleRTT0:

a)

$$\begin{aligned}\text{EstimatedRT}_{T1} &= (1 - \alpha) \cdot \text{EstimatedRT}_{T0} + \alpha \cdot \text{SampleRT}_{T1} \\ \text{EstimatedRT}_{T2} &= (1 - \alpha) \cdot \text{EstimatedRT}_{T1} + \alpha \cdot \text{SampleRT}_{T2} \\ \text{EstimatedRT}_{T3} &= (1 - \alpha) \cdot \text{EstimatedRT}_{T2} + \alpha \cdot \text{SampleRT}_{T3} \\ \text{EstimatedRT}_{T4} &= (1 - \alpha) \cdot \text{EstimatedRT}_{T3} + \alpha \cdot \text{SampleRT}_{T4}\end{aligned}$$

则:

$$\begin{aligned}\text{EstimatedRTT} &= \text{EstimatedRT}_{T4} \\ &= (1 - \alpha)^4 \text{EstimatedRT}_{T0} + \alpha(1 - \alpha)^3 \text{SampleRT}_{T1} \\ &\quad + \alpha(1 - \alpha)^2 \text{SampleRT}_{T2} + \alpha(1 - \alpha) \text{SampleRT}_{T3} + \alpha \text{SampleRT}_{T4}\end{aligned}$$

b)

$$\begin{aligned}\text{EstimatedRT}_{Tn} &= (1 - \alpha)^n \text{EstimatedRT}_{T0} + \alpha(1 - \alpha)^{n-1} \text{SampleRT}_{T1} + \cdots \\ &\quad + \alpha \text{SampleRT}_{Tn} \\ &= (1 - \alpha)^n \text{EstimatedRT}_{T0} + \alpha \sum_{j=1}^n (1 - \alpha)^{n-j} \text{SampleRT}_{Tj}\end{aligned}$$

c)

$$\begin{aligned}\lim_{n \rightarrow \infty} \text{EstimatedRT}_{Tn} &= \lim_{n \rightarrow \infty} \left( (1 - \alpha)^n \text{EstimatedRT}_{T0} + \alpha \sum_{j=1}^n (1 - \alpha)^{n-j} \text{SampleRT}_{Tj} \right)\end{aligned}$$

$0 < \alpha < 1$ ，即  $\lim_{n \rightarrow \infty} (1 - \alpha)^n \text{EstimatedRT}_{T0} \rightarrow 0$ ，则

$$\lim_{n \rightarrow \infty} \text{EstimatedRT}_{Tn} = \lim_{n \rightarrow \infty} \left( \alpha \sum_{j=1}^n (1 - \alpha)^{n-j} \text{SampleRT}_{Tj} \right)$$

J 越小，代表数据越旧，影响越小，EstimatedRTT 在更新的过程中，旧数据的权值呈指数型衰减，这个平均过程称为指数移动平均。

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。

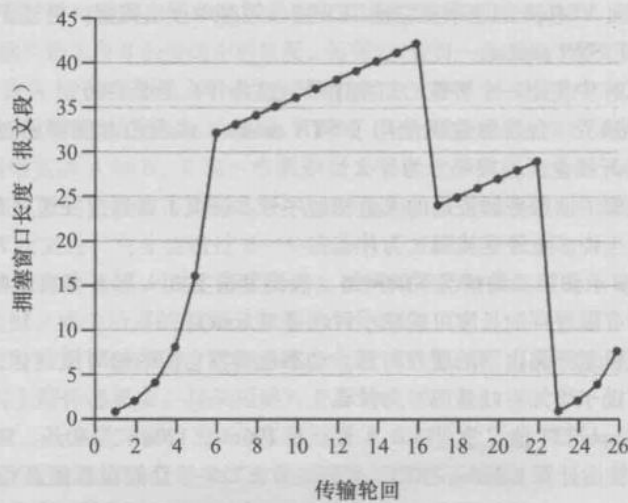


图 3-58 TCP 窗口长度作为时间的函数

- 指出 TCP 慢启动运行时的时间间隔。
- 指出 TCP 拥塞避免运行时的时间间隔。
- 在第 16 个传输轮回之后，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？
- 在第 22 个传输轮回之后，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？

a) 慢启动的时候，cwnd 的值会以指数的形式增长，运行时间间隔为[1, 6] 和 [23, 26]。

b) 拥塞避免的时候，cwnd 的值会以线性的形式增长，运行时间间隔为[6, 16] 和 [17, 22]。

c) 在第 16 个传输轮回之后，cwnd 的值没有变为 1MSS，由此可以推断为 3 个冗余 ACK。

d) 在第 22 个传输轮回之后，cwnd 的值变为 1MSS，由此可以推断为报文段丢失。

e. 在第 1 个传输轮回里，ssthresh 的初始值设置为多少？

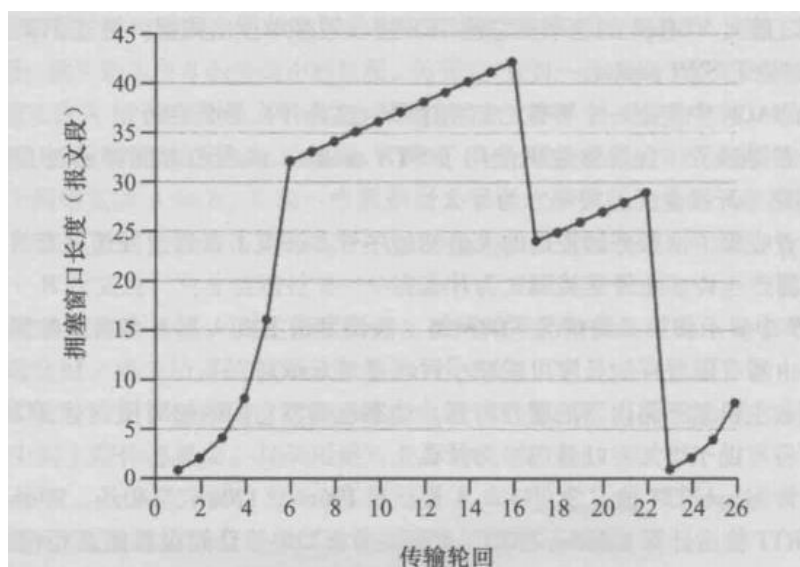
f. 在第 18 个传输轮回里，ssthresh 的值设置为多少？

g. 在第 24 个传输轮回里，ssthresh 的值设置为多少？

e) 在第 1 个传输轮回里，ssthresh 的初始值设置为 32MSS，因为慢启动在 32 处停止。

f) 因为上一个报文段丢失时，cwnd 的值为 42，所以在第 18 个传输轮回里，ssthresh 的值设为 cwnd 的一半，为 21。

g) 因为上一个报文段丢失时，cwnd 的值为 29，所以在第 24 个传输轮回里，ssthresh 的值设为 cwnd 的一半并向下取整，为 14。



h. 在哪个传输轮回内发送第 70 个报文段？

h) h 在第 1 到第 7 个传输轮回中属于慢启动阶段，发送的报文段呈指数形式增长。

在第 1 轮传输中，发送报文段 1；

报文段 2-3 在第 2 轮传输中发送；

报文段 4-7 在第 3 轮传输中发送；

报文段 8-15 在第 4 轮传输中发送；

报文段 16-31 在第 5 轮传输中发送；

报文段 32-63 在第 6 轮传输中发送；

报文段 64 - 96 在第 7 轮传输中发送。

因此，报文段 70 在第 7 轮传输中发送。

i. 假定在第 26 个传输轮回后，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？

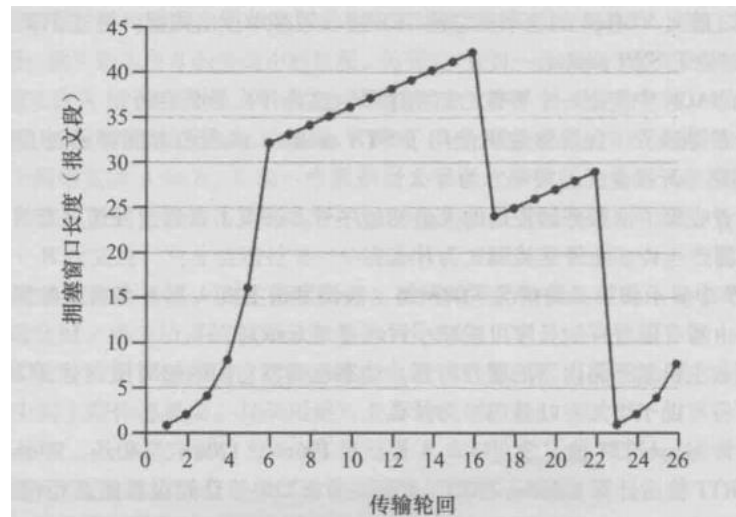
j. 假定使用 TCP Tahoe（而不是 TCP Reno），并假定在第 16 个传输轮回收到 3 个冗余 ACK。在第 19 个传输轮回，ssthresh 和拥塞窗口长度是什么？

i) 在第 26 个传输轮回时，cwnd 的值为 8，当收到 3 个冗余 ACK 时，ssthresh 变为 cwnd 的一半，为 4；cwnd 的值变为原来的一半并且加上 3 个 MSS，变为 7。

j) 在 TCP Tahoe 中，无论是接收到 3 个冗余 ACK 还是发生超时指示，都将 cwnd 变为 1。

在第 16 个传输轮回时，cwnd 的值为 42，接收到 3 个冗余 ACK，所以在第 17 个传输轮回里，ssthresh 的值为 21，cwnd 的值为 1，并进入慢启动阶段，cwnd 呈指数形式增长。故在 19 个传输轮回里，ssthresh 为 21，cwnd 为 4。





k. 再次假设使用 TCP Tahoe，在第 22 个传输轮回有一个超时事件。从第 17 个传输轮回到第 22 个传输轮回（包括这两个传输轮回），一共发送了多少分组？

- k) 在第 17 个传输轮回里发送 1 个分组；  
 在第 18 个传输轮回里发送 2 个分组；  
 在第 19 个传输轮回里发送 4 个分组；  
 在第 20 个传输轮回里发送 8 个分组；  
 在第 21 个传输轮回里发送 16 个分组；  
 在第 22 个传输轮回里发送 21 个分组（此时的 ssthresh 为 21）；  
 所以总数是 52。

P45

P45. 回想 TCP 吞吐量的宏观描述。在连接速率从  $W/(2 \times RTT)$  变化到  $W/RTT$  的周期内，只丢失了一个分组（在该周期的结束）。

a. 证明其丢包率（分组丢失的比率）等于：

$$L = \text{丢包率} = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

b. 如果一条连接的丢包率为  $L$ ，使用上面的结果，则它的平均速率近似由下式给出：

$$\text{平均速率} \approx \frac{1.22 * MSS}{RTT \sqrt{L}}$$

证明：

a TCP 通过每经过 1 个 RTT 将拥塞窗口增加 1 个 MSS 的带宽，在传输速率在  $\frac{W}{2 \times RTT}$

到  $\frac{W}{RTT}$  之间变化时，窗口大小增加  $\frac{W}{2}$ ，所以在这个周期内，传输的分组的大小为：

$$\sum_{n=0}^{\frac{W}{2}} \left( \frac{W}{2} + n \right) = \frac{3}{8}W^2 + \frac{3}{4}W$$

因为只丢失了一个分组，所以丢包率为：

$$L = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

B 对于 W 的大小  $\frac{3}{8}W^2 \gg \frac{3}{4}W$ , 所以  $L \approx \frac{1}{\frac{3}{8}W^2}$ , 即  $W \approx \sqrt{\frac{8}{3L}}$ , 带入公式平均速率 =

$$\frac{0.75 \times W}{RTT}$$

可以得到

$$\text{平均速率} = \frac{1.22MSS}{RTT\sqrt{L}}$$

P53

P53. 在 3.7 节对 TCP 未来的讨论中, 我们注意到为了取得 10Gbps 的吞吐量, TCP 仅能容忍  $2 \times 10^{-10}$  的报文段丢失率 (或等价于每 5 000 000 000 个报文段有一个丢包事件)。给出针对 3.7 节中给定的 RTT 和 MSS 值的对  $2 \times 10^{-10}$  值的推导。如果 TCP 需要支持一条 100Gbps 的连接, 所能容忍的丢包率是多少?

答: MSS=1500Bytes=1500×8bits=12000bits

RTT=100ms=0.1s

由公式平均速率 =  $\frac{1.22MSS}{RTT\sqrt{L}}$  可得

$$\sqrt{L} = \frac{1.22MSS}{RTT \times 10^{10}}$$

通过计算结果为  $L \approx 2 \times 10^{-10}$

当吞吐量为 100Gbps 时, 带入公式可得

$$L \approx 2.14 \times 10^{-12}$$