

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）。

为了评价该停等行为对性能的影响，可考虑一种具有两台主机的理想化场合，一台主机位于美国西海岸，另一台位于美国东海岸，如图 3-17 所示。在这两个端系统之间的光速往返传播时延 RTT 大约为 30 毫秒。假定彼此通过一条发送速率 R 为 1Gbps（每秒 10^9 比特）的信道相连。包括首部字段和数据的分组长 L 为 1000 字节（8000 比特），发送一个分组进入 1Gbps 链路实际所需时间是：

$$t_{\text{trans}} = \frac{L}{R} = \frac{8000\text{bit}/\text{pkt}}{10^9\text{bit/s}} = 8\mu\text{s}/\text{pkt}$$

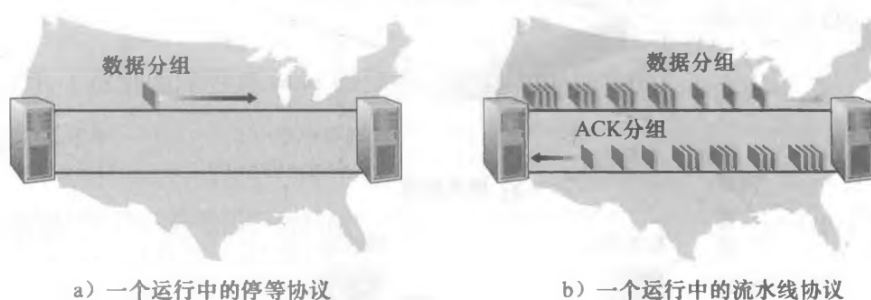


图 3-17 停等协议与流水线协议

$$\begin{aligned} L/R &= 12\mu\text{s} \\ n &= \frac{0.9 \times (RTT + L/R)}{L/R} = 2251 \end{aligned}$$

所以窗口大小大于等于2251

22

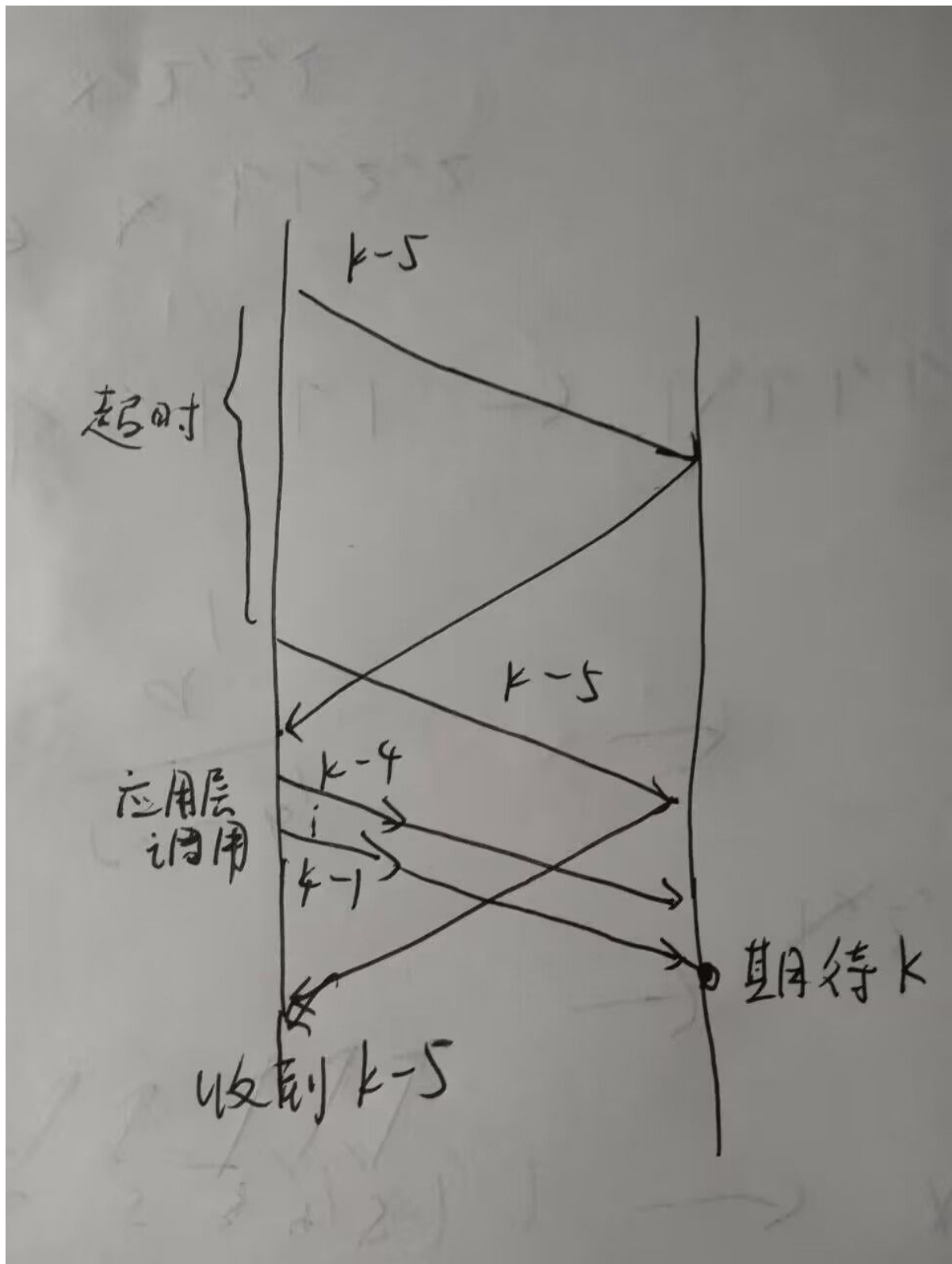
P22. 考虑一个 GBN 协议，其发送方窗口为 4，序号范围为 1024。假设在时刻 t ，接收方期待的下一个有序分组的序号是 k 。假设媒体不会对报文重新排序。回答以下问题：

- 在 t 时刻，发送方窗口内的报文序号可能是多少？论证你的回答。
- 在 t 时刻，在当前传播回发送方的所有可能报文中，ACK 字段的所有可能值是多少？论证你的回答。

a. 5种 起点为k-4 to k, 长度为4.

b. 5种 k-5 to k-1, 对应a种5种情况

其中k-5是由超时重传引起的，具体如下



23

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

对于GBN情况略有不同，因为GBN只会期望下一个包，而非一个窗口内的包。记初始窗口1~n，ack均丢失情况下，发送方发送1~n的包，接收方只期望接受n+1，其他包一概丢弃。所以只需要n+1和前面的n没有冲突，因此是k-1

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。

a) 可能。比如有一分组的ack没来得及返回发送方就被判定超时重发，当这个ack到达发送方时，发送方窗口移动，那么刚才重发的分组的ack就会落到窗口外。

b) 可能，原因同上

c) 是

d) 是

25

P25. 我们曾经说过，应用程序可能选择 UDP 作为运输协议，因为 UDP 提供了（比 TCP）更好的应用层控制，以决定在报文段中发送什么数据和发送时机。

- a. 应用程序为什么对在报文段中发送什么数据有更多的控制？
- b. 应用程序为什么对何时发送报文段有更多的控制？

a) 当你想发一个message的时候，UDP收到后直接将数据打包，而TCP可能将数据缓存，然后将更多或者更少的数据放到tcp segment里面。也就是说，如果应用给udp message，这个message就会是这个segment唯一的负载。因此udp有更大的权限去控制segment里面是什么数据。

b) TCP有流量控制以及拥塞控制，UDP没有。

27

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。

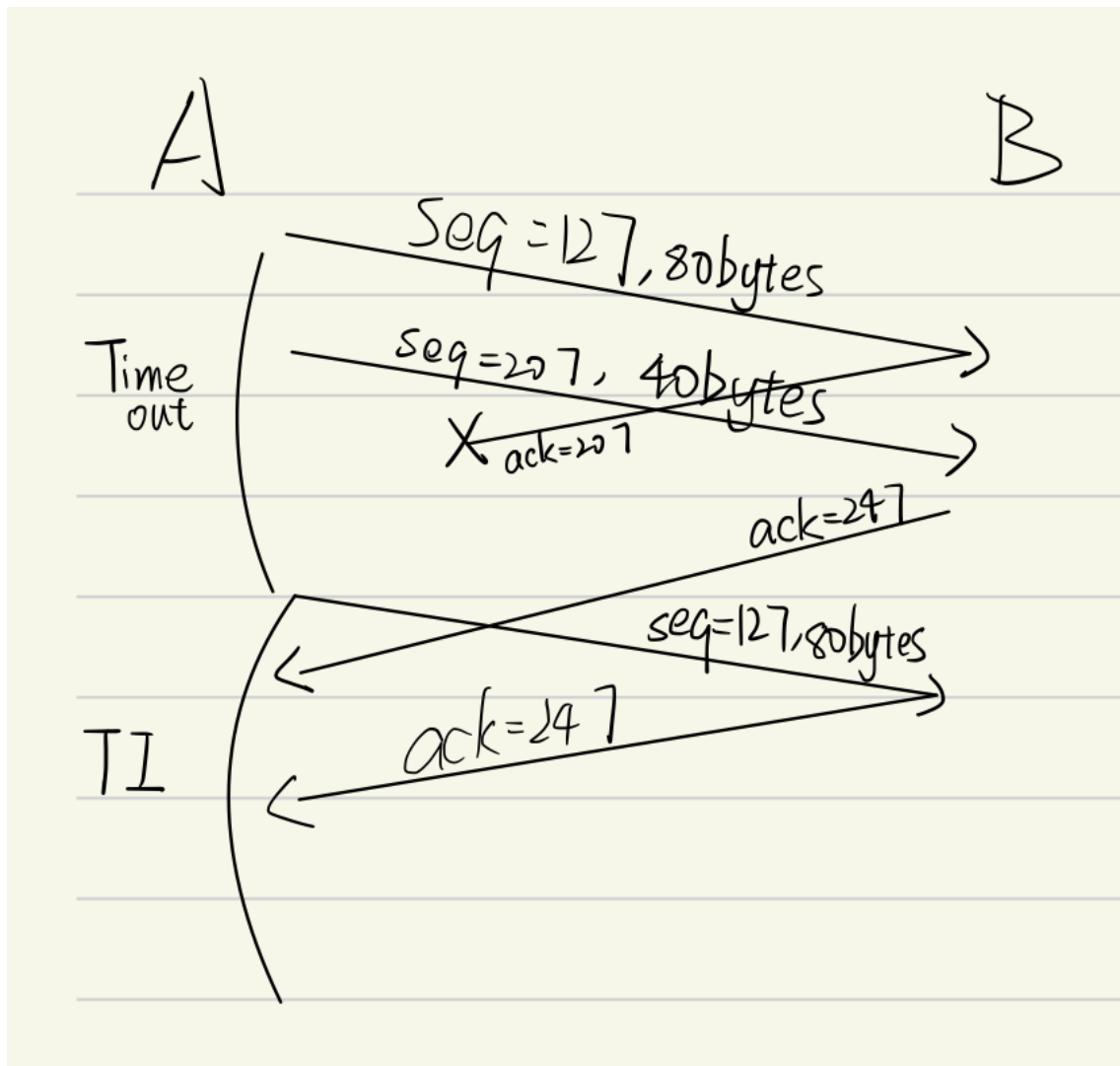
- a. 在从主机 A 发往 B 的第二个报文段中，序号、源端口号和目的端口号各是什么？
- b. 如果第一个报文段在第二个报文段之前到达，在第一个到达报文段的确认中，确认号、源端口号和目的端口号各是什么？
- c. 如果第二个报文段在第一个报文段之前到达，在第一个到达报文段的确认中，确认号是什么？
- d. 假定由 A 发送的两个报文段按序到达 B。第一个确认丢失了而第二个确认在第一个超时间隔之后到达。画出时序图，显示这些报文段和发送的所有其他报文段和确认。（假设没有其他分组丢失。）对于图上每个报文段，标出序号和数据的字节数量；对于你增加的每个应答，标出确认号。

1. 207, 302, 80

2. 207, 80, 302

3. 127

4.



37

P37. 比较 GBN、SR 和 TCP（无延时的 ACK）。假设对所有 3 个协议的超时值足够长，使得 5 个连续的数据报文段及其对应的 ACK 能够分别由接收主机（主机 B）和发送主机（主机 A）收到（如果在信道中无丢失）。假设主机 A 向主机 B 发送 5 个数据报文段，并且第二个报文段（从 A 发送）丢失。最后，所有 5 个数据报文段已经被主机 B 正确接收。

- 主机 A 总共发送了多少报文段和主机 B 总共发送了多少 ACK？它们的序号是什么？对所有 3 个协议回答这个问题。
- 如果对所有 3 个协议超时值比 5RTT 长得多，则哪个协议在最短的时间间隔中成功地交付所有 5 个数据报文段？

a

GBN:

A: 9 个 1 2 3 4 5 2 3 4 5

B: 8 个 1 1 1 1 2 3 4 5

SR:

A: 6 个 1 2 3 4 5 2

B: 5 个 1 3 4 5 2

TCP:

A: 6 个 1 2 3 4 5 2

b

tcp, 因为tcp收到3个冗余ACK就立即重传, 而不用等待超时, 而且只需要发送6个报文。

40

P39. 考虑图 3-46b。如果 λ'_{in} 增加超过了 $R/2$, λ_{out} 能够增加超过 $R/3$ 吗? 试解释之。现在考虑图 3-46c。

假定一个分组从路由器到接收方平均转发两次的话, 如果 λ'_{in} 增加超过 $R/2$, λ_{out} 能够增加超过 $R/4$ 吗? 试解释之。

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。

- 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?
- 在第 1 个传输轮回里, $ssthresh$ 的初始值设置为多少?
- 在第 18 个传输轮回里, $ssthresh$ 的值设置为多少?
- 在第 24 个传输轮回里, $ssthresh$ 的值设置为多少?
- 在哪个传输轮回内发送第 70 个报文段?
- 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?
- 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?

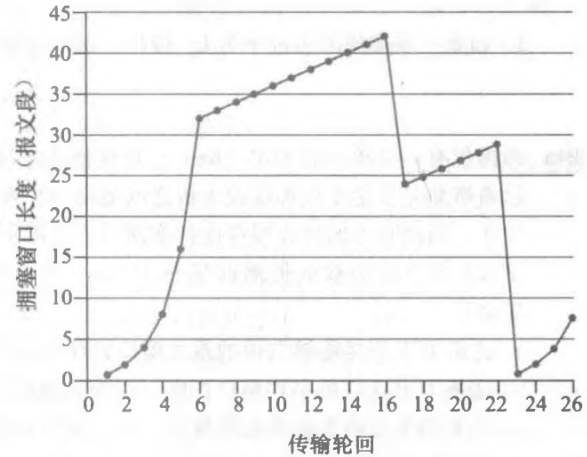


图 3-58 TCP 窗口长度作为时间的函数

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?

- [1,6], [23,26]
- [6,16], [17,22]
- 3个冗余ack, 否则cwnd应该是1而不是24
- 超时, cwnd被设置为1而不是+1 (快速重传时受到冗余ack会继续加窗口)
- 32
- 21
- 14 (14.5, 如果取下整则为14)
- 简单计算可知在第7个
- windows7 和 ssthresh 4
- threshold = 21 window = 4
- (round,packet):(17,1),(18,2),(19,4),(20,8),(21,16),(22,21) 一共32+21 = 52个packet

P44. 考虑从一台主机经一条没有丢包的 TCP 连接向另一台主机发送一个大文件。

- 假定 TCP 使用不具有慢启动的 AIMD 进行拥塞控制。假设每当收到一批 ACK 时, cwnd 增加 1 个 MSS, 并且假设往返时间大约恒定, cwnd 从 6MSS 增加到 12MSS 要花费多长时间 (假设没有丢包事件)?
- 对于该连接, 到时间 $= 6RTT$, 其平均吞吐量是多少 (根据 MSS 和 RTT)?

a) $6RTT$

b) $6+7+8+9+10+11=51$

$51/6=8.5MSS/RTT$

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)。

- 证明其丢包率 (分组丢失的比率) 等于:

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

- 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:

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

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

$$L(\text{LossRate}) = 1/packets = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

$$W \text{ 比较大时, } L \approx \frac{8}{3W^2}, W \approx \sqrt{\frac{8}{3L}}$$

$$\text{平均速率: } \frac{3}{4}W \frac{MSS}{RTT} = \frac{1.22 * MSS}{RTT * \sqrt{L}}$$

P46. 考虑仅有一条单一的 TCP (Reno) 连接使用一条 10Mbps 链路, 且该链路没有缓存任何数据。假设这条链路是发送主机和接收主机之间的唯一拥塞链路。假定某 TCP 发送方向接收方有一个大文件要发送, 而接收方的接收缓存比拥塞窗口要大得多。我们也做下列假设: 每个 TCP 报文段长度为 1500 字节; 该连接的双向传播时延是 150ms; 并且该 TCP 连接总是处于拥塞避免阶段, 即忽略了慢启动。

- 这条 TCP 连接能够取得的最大窗口长度 (以报文段计) 是多少?
- 这条 TCP 连接的平均窗口长度 (以报文段计) 和平均吞吐量 (以 bps 计) 是多少?
- 这条 TCP 连接在从丢包恢复后, 再次到达其最大窗口要经历多长时间?

$$W * MSS / RTT = 10Mbps$$

$$W = 125 (126 \text{ 也算对, 用时间间隔算的})$$

$$b. 0.75W = 94$$

$$\text{吞吐量} 94 * 1500 * 8 / 0.15 = 7.52Mbps$$

$$c \text{ 最小窗口 } W/2 = 62 \text{ 需要时间 } (125 - 62) * 150ms = 9.45s$$