Assignment II

Chapter IV

2021 年 5 月 7 日

谢远峰

网安一班

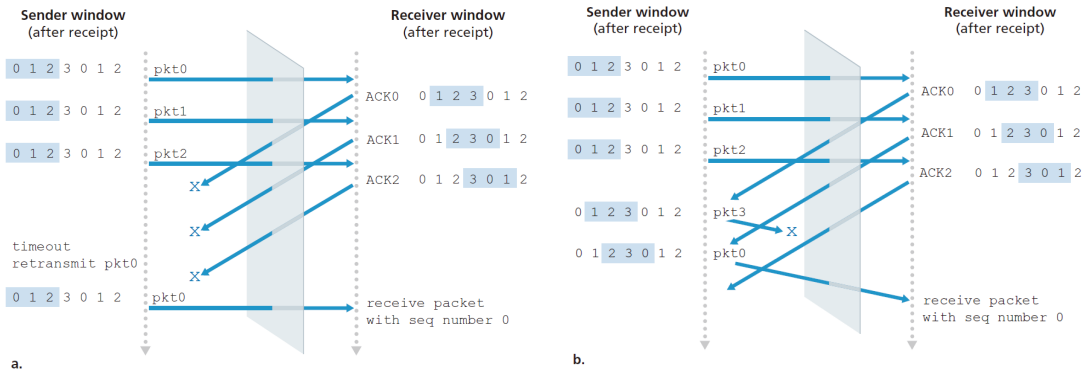
3019244283

Question1

设置计时器是有必要的

- 往返延迟时间将使发送方知道数据包或数据包的 ACK 是否丢失。
- 计时器用于跟踪数据包的传输时间。
- 发送方需要一个计时器来检测每个数据包的丢失。

Question2



(a) New Packet (b) Retransmission

图 1: SR 协议可能遇到的问题

发送方和接收方窗口间缺乏同步会产生严重的后果，有 4 个分组序列号且窗口长度为 3，假定发送了分组 0 至 2，并在接收方被正确接收且确认了。此时，接收方窗口落在第 4、5、6 个分组上，其序号分别为 3、0、1。现在考虑两种情况。在第一种情况下，如 a 所示，对前 3 个分组的 ACK 丢失，因此发送方重传这些分组。因此，接收方下一步要接收序号为 0 的分组即第一个发送分组的副本。在第二种情况下，如 b 所示，对前 3 个分组的 ACK 都被正确交付。因此发送方向前移动窗口并发送第 4、5、6 个分组其序号分别为 3、0、1。序号为 3 的分组丢失，但序号为 0 的分组到达（一个包含新数据的分组）。

考虑图中接收方的视角，在发送方和接收方之间有一个假想的帘子，因为接收方不能“看见”发送方采取的动作。接收方所能观察到的是它从信道中收到的以及它向信道中发出报文序列。就其所关注的而言，图中的两种情况是等同的。没有办法区分是第 1 个分组的重传还是第 5 个分组的初次传输。显然，窗口长度比序号空间小时协议无法工作。

为了避免出现图中显示的情况，需要避免接收方窗口的上界（接收方窗口的” 最新” 序列号）与发送方窗口下界（发送方窗口中” 最老” 序列号）重叠。因此，序列号空间必须足够大，以适应整个接收方窗口和整个发送方窗口。

假设接收方正在等待的最低序列号是数据包 m，在这种情况下，它的窗口 [m, m+w-1]，它已经接收数据包 m-1（并发送 ACK 信息）和之前的 w-1 数据包，其中 w 是窗口的大小。如果这些 w 个 ACK 都还没有被发送者收到，那么数值为 [m-w,m-1] 的 ACK 信息可能还在传播回来。如果发送方没有收到这些 ACK 号码，那么发送方的窗口将是 [m-w,m-1]。

因此，发送方窗口的下界是 m-w，而接收方窗口的上界是 m+w-1。为了使接收方窗口的上界不与发送方窗口的下界重叠，因此序列号长度至少为 2w，以同时容纳发送方窗口大小 + 接收方窗口大小，即 $w \leq k/2$

SR 协议发送方的窗口大小最大为 $\lfloor k/2 \rfloor$

Question3

a

用 $EstimatedRTT^{(n)}$ 表示对于第 n 个样例的预测

$$
\begin{aligned}
EstimatedRTT^{(1)} &= SampleRTT \\
EstimatedRTT^{(2)} &= x * SampleRTT_1 + (1-x) * SampleRTT_2 \\
EstimatedRTT^{(3)} &= x * SampleRTT_1 + (1-x)[x * SampleRTT_2 + (1-x) * SampleRTT_3] \\
&= x * SampleRTT_1 + (1-x)x * SampleRTT_2 + (1-x)^2 * SampleRTT_3 \\
EstimatedRTT^{(4)} &= x * SampleRTT_1 + (1-x) * SampleRTT_3 \\
&= x * SampleRTT_1 + (1-x)x * SampleRTT_2 + (1-x)^2 x * SampleRTT_3 \\
&\quad + (1-x)^3 * SampleRTT_4
\end{aligned}
$$

b

$$
EstimatedRTT^{(n)} = x \sum_{j=1}^{n-1} (1-x)^{j-1} * SampleRTT_j + (1-x)^{n-1} * SampleRTT_n
$$

c

$$
\begin{aligned}
EstimatedRTT^{\infty} &= \frac{\alpha}{1-\alpha} \sum_{j=1}^{\infty} (1-\alpha)^j * SampleRTT_j \\
&= \frac{1}{9} \sum_{j=1}^{\infty} (0.9)^j * SampleRTT_j
\end{aligned}
$$

因为基于先前样本的权重以指数形式衰减, 所以程序被称为指数移动平均。

Question4

a

GBN:

A 首先发送 5 个报文段（1,2,3,4,5），第二次重新发送 4 个报文段（2,3,4,5），总共发送 9 个报文段，A 发送的报文段顺序：1，2，3，4，5，2，3，4，5

B 首先发送 4 个 ACK（1,1,1,1），之后依次发送 4 个 ACK（2,3,4,5），B 总共发送 8 个报文段，B 发送的 ACK 顺序：1，1，1，1，2，3，4，5

SR:

A 首先发送 5 个报文段（1,2,3,4,5），第二次重新发送 1 个报文段（2），总共发送 6 个报文段，A 发送的报文段顺序：1，2，3，4，5，2

B 首先发送 4 个 ACK（1,3,4,5），之后发送一个 ACK（2），B 总共发送 5 个 ACK，B 发送的 ACK 顺序：1，3，4，5，2

TCP:

A 首先发送 5 个报文段（1,2,3,4,5），第二次重新发送报文段（2），总共发送 6 个报文段，A 发送的报文段顺序：1，2，3，4，5，2

B 首先发送 4 个 ACK（2,2,2,2），之后发送 ACK（6），B 总共发送 5 个 ACK，B 发送的 ACK 顺序：2，2，2，2，6

b

如果这三种协议的超时值都远长于 5 个 RTT，由于 TCP 使用快速重传，不需要等待时间，所以能在最短的时间间隔内成功传送所有五个数据段。

Question5

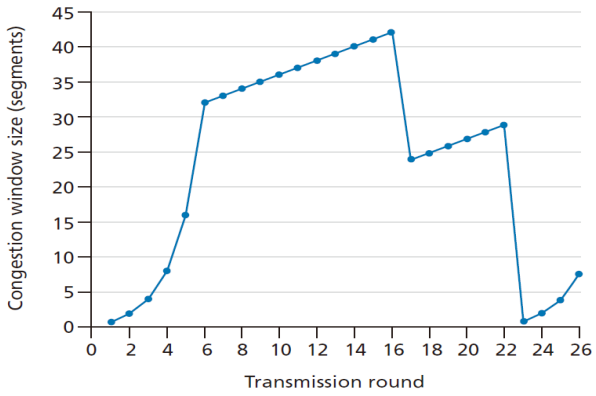


Figure 3.61 ♦ TCP window size as a function of time

a TCP 的慢启动区间在 [1,6] 和 [23,26]

b TCP 的拥塞避免区间在 [6,16] 和 [17,22]

c 由于重复接收到相同的 ACK 导致的丢包，如果为超时，发送窗口大小会降至 1

d 由于计时器超时导致的丢包，发送窗口大小降至 1

e 在第一个传输回合，慢启动阈值设置为 32

f 由于在第 16 个传输回合达到的窗口大小为 42，检测到报文段的丢失，慢启动阈值减半，因此在第 18 个传输回合，慢启动阈值为 21

g 由于在第 22 个传输回合达到的窗口大小为 29，计时器超时导致丢包事件出现，窗口大小归为 1，进入慢启动阶段，因此在第 23 个传输回合，慢启动阈值为 14

h 第一传输回合，包 1 发送；第二个传输回合，包 2-3 发送；第三个传输回合，包 4-7 发送；第四个回合，包 8-15 发送；第五个回合，包 16-31 发送，第六个回合，包 32-63 发送；第七个回合，包 64-96 发送。因此，包 70 在第七个传输回合内发送

i 第 21 个传输回合，检测到报文段丢失，当前拥塞窗口大小和慢启动阈值减半，设置为 4，

j 慢启动阈值为 21，拥塞窗口大小为 4

k 传输回合 17，1 个包；传输回合 18，2 个包；传输回合 19，4 个包，传输回合 20，8 个包，传输回合 21，16 个包；传输回合 22，21 个包。因此总共用发送了 52 包