

第3章-运输层(2)

231880038 张国良

- 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。

P40. 考虑图 3-61 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？
- 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？

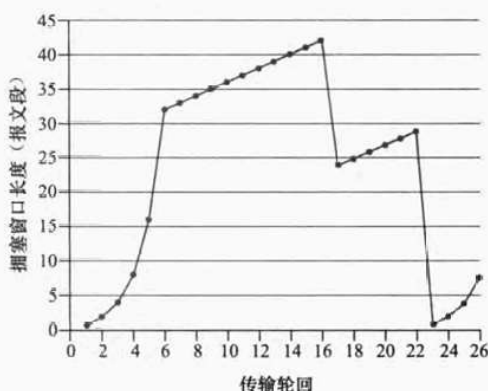


图 3-61 TCP 窗口长度作为时间的函数

- P48 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？

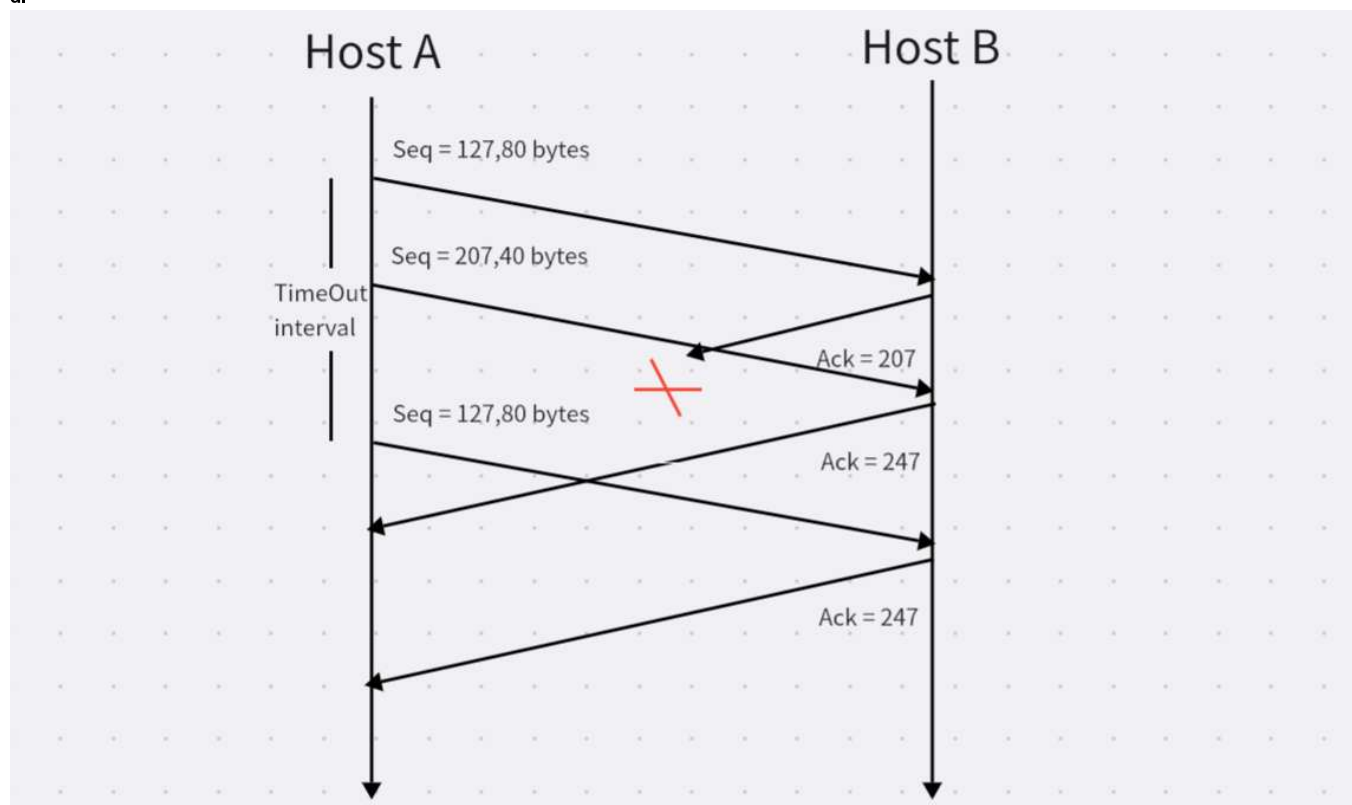
P52 考虑一种简化的 TCP 的 AIMD 算法，其中拥塞窗口长度用报文段的数量来度量，而不是用字节度量。在加性增中，每个 RTT 拥塞窗口长度增加一个报文段。在乘性减中，拥塞窗口长度减小一半（如果结果不是一个整数，向下取整到最近的整数）。假设两条 TCP 连接 C1 和 C2，它们共享一条速率为每秒 30 个报文段的单一拥塞链路。假设 C1 和 C2 均处于拥塞避免阶段。连接 C1 的 RTT 是 50ms，连接 C2 的 RTT 是 100ms。假设当链路中的数据速率超过了链路的速率时，所有 TCP 连接经受数据报文段丢失。

- 如果在时刻 t_0 ，C1 和 C2 具有 10 个报文段的拥塞窗口，在 1000ms 后它们的拥塞窗口为多长？
- 经长时间运行，这两条连接将取得共享该拥塞链路的相同的带宽吗？

P54 考虑修改 TCP 的拥塞控制算法。不使用加性增，使用乘性增。无论何时某 TCP 收到一个合法的 ACK，就将其窗口长度增加一个小正数 a ($0 < a < 1$)。求出丢包率 L 和最大拥塞窗口 W 之间的函数关系。论证：对于这种修正的 TCP，无论 TCP 的平均吞吐量如何，一条 TCP 连接将其拥塞窗口长度从 $W/2$ 增加到 W ，总是需要相同的时间。

Problem 1

- 序号是 207，源端口是 302，目的端口是 80
- 确认号是 207，源端口是 80，目的端口是 302
- 确认号是 127，因为还在等待序号 127 数据包的到来
-



Problem 2

- 在 1 ~ 6 和 23 ~ 26 慢启动
- 在 6 ~ 16 和 17 ~ 22 拥塞避免
- 根据三个冗余 ACK，因为在 16 时未超时，窗口长度大概 42，如果是超时检测，应该下降到 1，而实际上下降到 24，这是因为有三个冗余所以减半加上 3
- 第 22 次传输后，是由于超时而检测到段丢失，因为拥塞窗口大小设置为 1
- 阈值最初是 32，因为它处于慢启动停止的窗口大小，并且拥塞避免开始
- 当分组丢失时阈值被设置为拥塞窗口的值的一半。当在传输循环 16 期间检测到丢失时，拥塞 Windows 的大小为 42。因此在第 18 传输循环期间阈值为 21

- g. 当分组丢失时阈值被设置为拥塞窗口的值的一半。当在传输循环22期间检测到丢失时，拥塞Windows的大小为29。因此，在第24轮传输中阈值为14.5（取下限为14）
- h. 截止到6，共发送了63个报文段，在7将发送33个报文段，因此在7发送第70个报文段
- i. 收到三个冗余ACK，阈值减半，窗口长度在阈值加三，所以阈值设为4，窗口长度设为7
- j. 使用TCP Tahoe，阈值减半，下一轮回窗口长度设为1，随后指数增长，到19轮回，窗口长度为4，阈值为21
- k. 在第17，1个报，18发送2个报文段，19发送4个报文段，20发送8个报文段，21发送16个报文段，22发送21个报文段，一共发送了52个报文段

Problem 3

a.

$$\frac{W \times MSS}{RTT} = 10Mbps$$

代入得 $W = 125$

所以最大窗口长度为125段

b.

平均窗口长度： $0.75W = 93.75$

平均吞吐量： $\frac{0.75W}{RTT} \times MSS = 7500000bps$

c.

由题可知：该连接总是处于拥塞避免阶段

丢包恢复后窗口大小为 $\frac{W}{2}$ ，要增长到 W

耗时 $\frac{W}{2} \times RTT = 9.375s$

Problem 4

a.

时间	C1	C2
0ms	10	10
50ms	5	10
100ms	2	5
150ms	1	5
200ms	1	2
250ms	1	2
300ms	1	1
350ms	2	1
400ms	1	1
...	1/2变化	1
1000ms	1	1

1000ms后C1窗口长度为1段，C2窗口长度为1段

b.

在长期运行中，C1的带宽份额大约是C2的两倍，因为C1具有较短的RTT，仅是C2的一半，因此C1可以将其窗口大小调整为C2的两倍

Problem 5

使用乘性增，窗口从 $\frac{W}{2}$ 到 W 发出段数：

$$S = \frac{W}{2} + \frac{W}{2}(1 + \alpha) + \frac{W}{2}(1 + \alpha)^2 + ... + \frac{W}{2}(1 + \alpha)^k$$

$k = \log_{(1+\alpha)} 2$

得出： $S = \frac{W}{2} \times \frac{(1 + \alpha)^{k+1} - 1}{(1 + \alpha) - 1} = \frac{2\alpha + 1}{2\alpha}W$

$L = \frac{1}{S} = \frac{2\alpha}{(2\alpha + 1)W}$

消耗时间： $k \times RTT = RTT\log_{(1+\alpha)} 2$

仅和 RTT 与 α 大小有关，与吞吐量大小显然无关