

5-19

假设当前接收窗口正好在 $2^n - 1$ 号分组处

发送窗口和接收窗口必须满足以下两条：

- 发送窗口必须包含 $2^n - 1$ 号分组
- 若已经收到了分组 $2^n - 2$ 的确认，则发送窗口的左侧不能再含有分组 $2^n - 2$

以上两种情况对应了发送窗口的两种极端情况，其余情况对发送窗口的大小要求没有那么严格

设发送窗口为 W_T ，接收窗口为 W_R ，则满足：

$$\begin{aligned} W_T &\leq 2^n \\ W_T + W_R &\leq 2^n \end{aligned}$$

题目中 $W_R = 1$ ，因此发送窗口 $W_T \leq 2^n - 1$

5-22

(1) TCP的序号是每一个字节的编号，因此 L 与MSS无关。序号为4字节，能表示的最大值为 2^{32} ，因此 L 的最大值为 2^{32} 字节，即4GB

(2) $2^{32}/1460 = 2941758.422$ ，因此需要发送帧的数量为2941759，头部的额外开销为 $66 \times 2941759 = 194156094$ 字节

因此发送的总字节数为4489123390字节，数据率=10Mbit/s=1.25MB/s，因此发送时间
=4489123390/1250000=3591.3s

5-23

- (1) 第一个报文段的数据序号是70-99，共30个字节
- (2) B收到的第二个报文的第一个数据字节的序号是100，因此确认号是100
- (3) 有80个字节
- (4) 确认号应为70

5-24

设发送窗口的长度为 W ，发送端连续发送完窗口内的数据所需时间为 T

- (1) 若接收方再接收完一批数据后再发送确认：

这种情况下发送方经过 $(256\text{ms}+T)$ 后才能发送下一个发送窗口的数据，此时有：

$$\text{吞吐量} = \frac{W}{\frac{W}{256\text{kbit/s}} + 256\text{ms}} = 120\text{kbit/s}$$

解得 $W = 57825.88\text{bit}$ ，取整后约为7228Byte

- (2) 若接收方每收到一个很小的报文就发送确认：

这种情况下发送方每经过比256ms略长的时间就可以发送一个数据，因此可以看作每经过256ms就发送一个窗口的数据

$$\text{吞吐量} = \frac{W}{256\text{ms}} = 120\text{kbit/s}$$

解得 $W = 30720\text{bit}$, 即 3840Byte

5-35

(1) 由于 $1\text{MB} = 1024\text{KB} = 2^{10}\text{KB}$, 且不会发生网络拥塞, 因此经过 n 个 RTT 后, 发送窗口会增大到 2^nKB 。

所以, 经过 10 个 RTT 后, 发送窗口会增大到 1MB

(2) 经过 RTT 的数量与已传送的分组大小关系如下:

- 1 个 RTT: $1 = 2^1 - 1$ 个分组
- 2 个 RTT: $3 = 2^2 - 1$ 个分组
- 3 个 RTT: $7 = 2^3 - 1$ 个分组
-

可以看出再经历了 10 个 RTT 后, 已发送的分组数为 $2^{10} - 1$ 个, 大小为 $1\text{MB} - 1\text{KB}$, 此处的 1KB 可以暂时忽略

- 11 个 RTT: 2MB
- 12 个 RTT: 4MB
- 13 个 RTT: 8MB
- 14 个 RTT: 全部

因此, 把数据成功发送需要 14 个 RTT。再第 14 个 RTT **开始时**, 发送窗口大小为 $2^{13}\text{KB} = 2^{23}\text{B}$, 而使用了窗口扩大选项后的最大窗口为 $2^{30} - 1\text{B}$, 因此 TCP 扩大的窗口是够用的

(3)

14 个 RTT 占用的时间为: $14 \times 50\text{ms} = 0.7\text{s}$

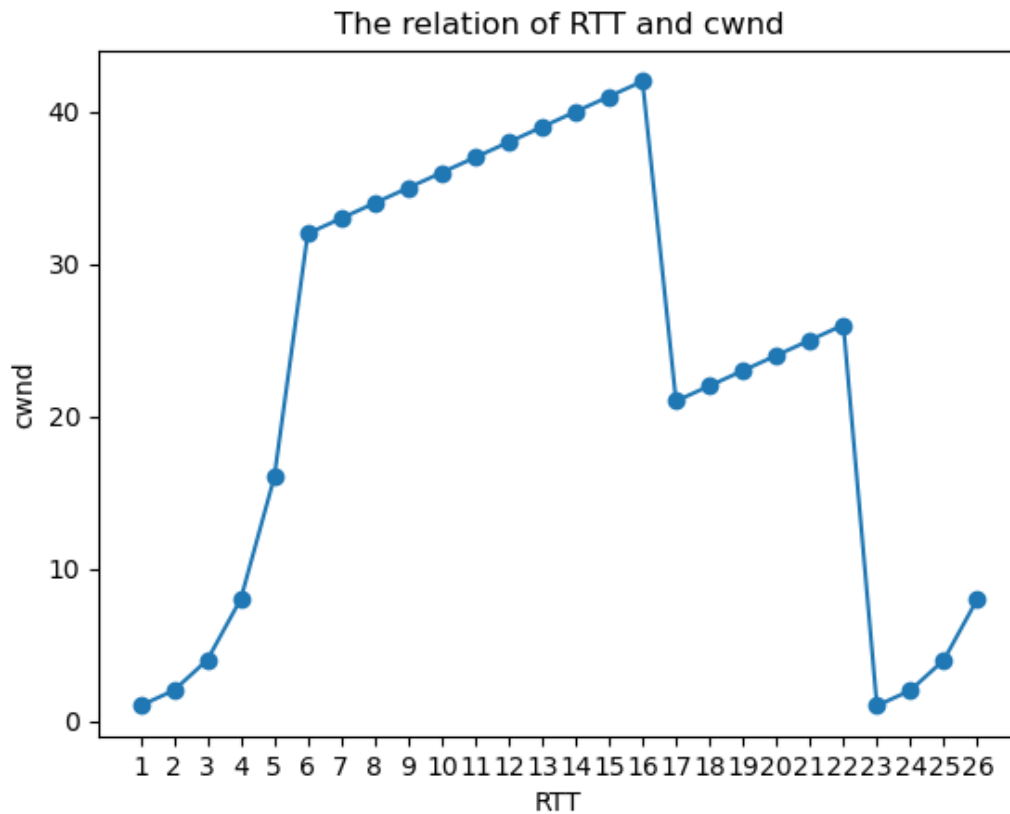
$10\text{MB} = 10 \times 2^{10} \times 2^{10} \times 8\text{bit}$

吞吐率 $= 10 \times 2^{10} \times 2^{10} \times 8 / 0.7\text{s} = 119.8 \times 10^6 \text{bit/s} = 119.8\text{Mbit/s}$

链路带宽的利用率 $= 119.8\text{Mbit/s} / 1000\text{Mbit/s} = 11.98$

5-39

(1)



(2) 慢开始阶段时间间隔: [RTT=1,RTT=6], [RTT=23,RTT=26]

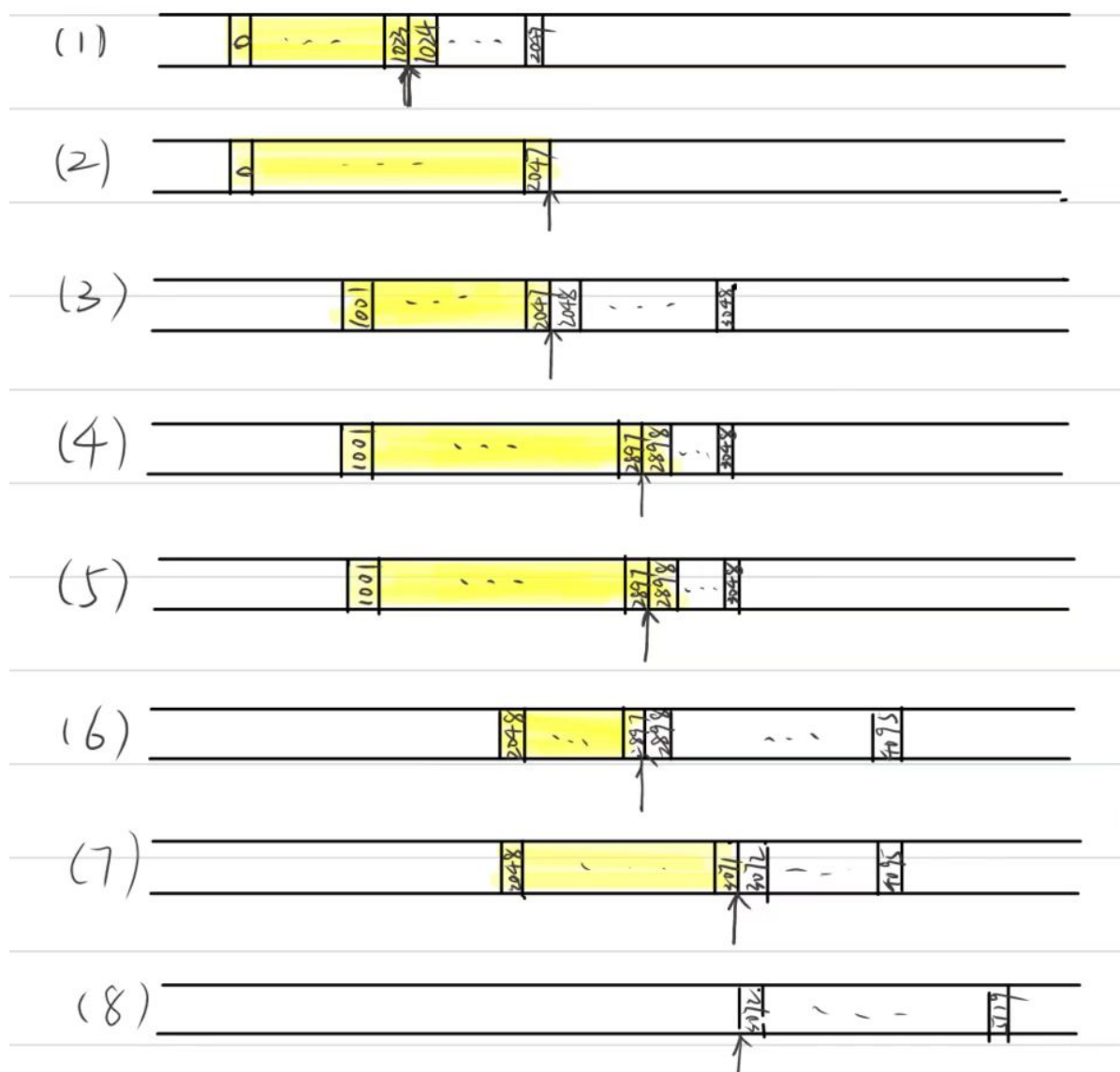
(3) 拥塞避免阶段时间间隔: [RTT=6,RTT=16], [RTT=17,RTT=22]

(4) 是通过收到了三个重复的确认报文, 因为拥塞窗口减为原来的一半而不是减为1

(5) RTT=1: ssthresh=32 RTT=18: ssthresh=21 RTT=24: ssthresh=13

(6) 当RTT=6结束时, 已发送63个报文段, 因此第70个报文段在第7个RTT发送

(7) 拥塞窗口设置为原来的一半, 即4; ssthresh设置为检测出拥塞时拥塞窗口的一般, 即4



涂黄色的区域表示已经使用的发送窗口，白色的区域表示仍可使用的发送窗口

5-64

- 收到ACK报文段：进入FIN-WAIT-2状态
- 收到FIN报文段：进入TIME-WAIT状态
- 发生了超时：进入到CLOSED状态

5-66

若这个报文段中的数据部分有n个字节，则下一个报文段的序号应当是x+n；只有当报文段中数据部分只有一个字节时，下一个报文段的序号才是x+1

5-70

(1) 在40Gbit/s的线路上传送数据，每秒可传送字节数为： 5×10^9

TCP的序号字段有32位，共有 2^{32} 个不同的序号，因此 $2^{32} / (5 \times 10^9) = 0.859s = 859ms$ 后会发生序号绕回

(2) $2^{32} \times 859 \times 10^{-6}s = 3.69 \times 10^6s = 42.7$ 天，因此经过42.7天后才会发生时间戳的绕回

流量控制：是在一条TCP连接中的接收端采用的措施，用于限制对方发送报文段的速率。因此流量控制只控制一个发送端

拥塞控制：拥塞控制会涉及到多个发送端发送报文段的速率，不过每个发送端只知道自己应该怎么调整发送速率

当接收窗口小于拥塞窗口时，发送窗口的大小取决于流量控制；而当拥塞窗口小于接收窗口时，发送窗口的大小取决于拥塞控制