

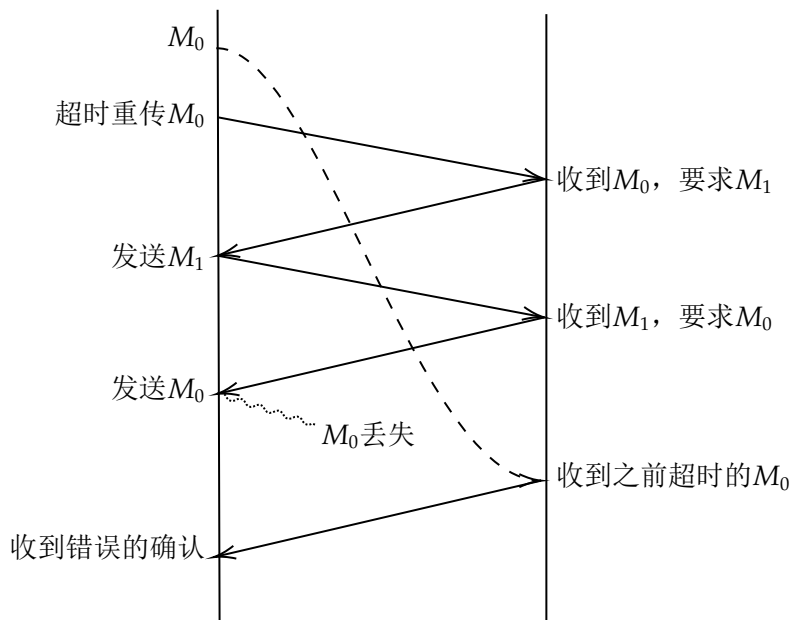
计算机网络课程 第五次作业

黄昊 20204205

5-08 UDP将从应用层接收到的报文加上首部后就直接交付给网络层。而TCP 对应用层交付下来的数据块当作字节流处理，不保证接收方和发送方的数据块大小一模一样，但能保证字节流完全一样，这就是面向字节流的含义。

5-17 不行，会导致发送方一直超时重传，一直传送重复的报文段。

5-18



5-21 (1) 要求收到5，说明5之前的都收到了，但考虑到ACK可能未到达接收方，因此 $[5 - m, 7 - m]$, $m \in \{3, 2, 1, 0\}$ 均有可能。

(2) 考虑窗口大小，再考虑捎带机制，可以发现2-4都有可能。

5-22 (1) 2^{32} 个

(2) 发送帧的个数 N 为 $\frac{2^{32}}{1460}$ ，总字节数 tot 为 $2^{32} + N \times 66$ ，所需时间 t 为 $\frac{tot}{1.25MB/s}$ ，计算得出时间为3591.3s，即59.85 min

5-27 65535-20-20=65495字节，其中不考虑可选项字段。

5-32 在计算平均往返时间 RTT 时，只要报文段重传了，就不采用其往返时间样本。这就是 Karn算法。

考虑是如何减少的：收到的确认是先发的（就是超时的）而不是超时重发的确认，这一时间会变少。这一调整导致后面重传的频率越来越频繁，估算的RRT越来越小，进而趋近于0。

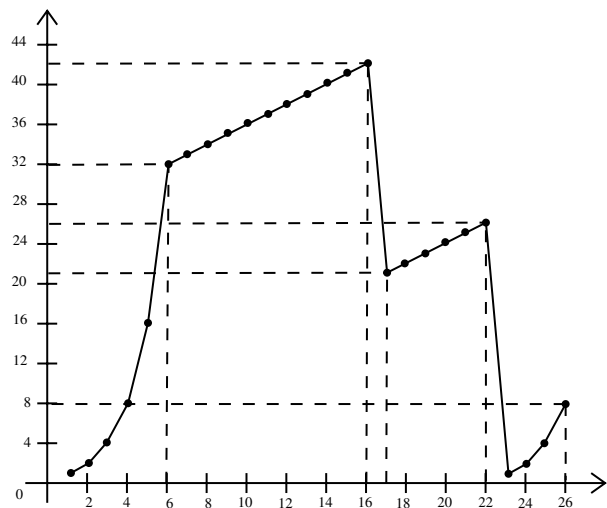
5-35 传输时延总和： $\frac{960}{48} \times 5 = 100ms$

传播时延总和： $250 \times 2 + \frac{1500}{150} \times 3 = 530ms$

总和：630ms

5-36 $630 - 20 + 150 = 760ms$

5-39 (1) 如下图所示：



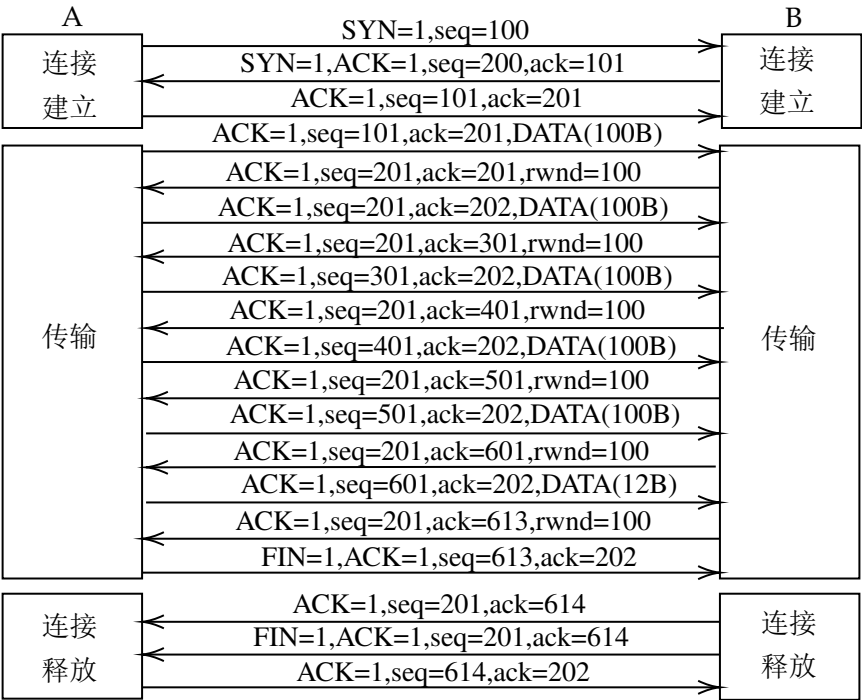
- (2) [1, 6], [23, 26]
- (3) [6, 16], [17, 22]
- (4) 三个重复的确认，因为下一轮的拥塞窗口减半了。
- (5) 分别为32， 21， 13
- (6) 求前缀和：

轮次	1	2	3	4	5	6	7
报文段总和	1	3	7	15	31	63	94

因此在第7轮发出。

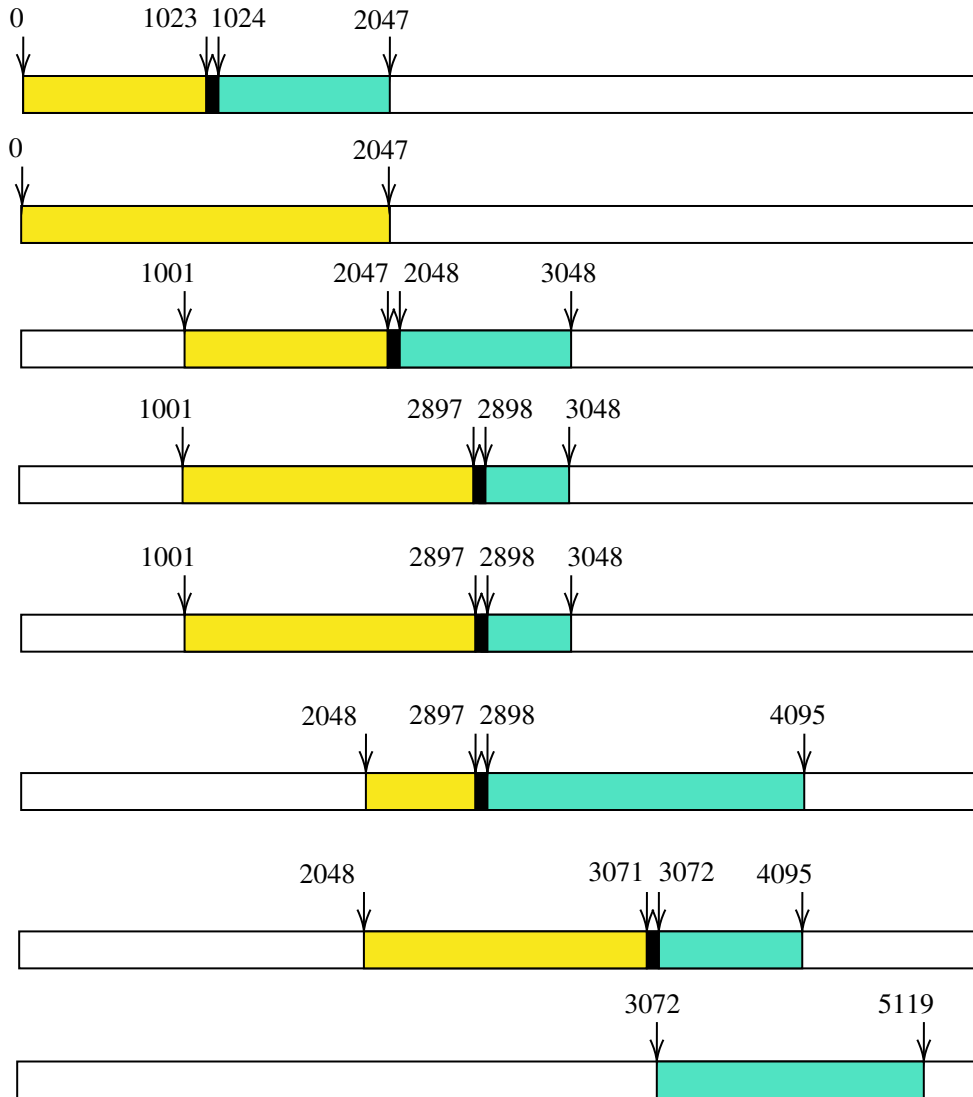
- (7) 均设置为4.

5-41 三个过程：连接建立，数据传输，连接释放。通信阶段，需要传输六次数据。下面是示意图。



5-46 考虑超时的问题。如果某个请求握手的报文超时了很久，以至于发送端早已关闭了连接，此时该报文到达接收端。接收端接收到后，会尝试与发送端建立连接。但发送端收到后应该知道这个连接不应该被建立。如果没有第三次握手，则将会有一个连接意外被建立。

5-61



从上到下的八个长条图依次是第（1）小问到第（8）小问的回答。其中黄色是已用窗口（已发送未收到确认），青色为可用窗口（未发送），其余部分（白色部分）为不可用窗口。

5-68 没有问题。因为第三次握手中，是对连接的建立进行再一次的确认，自此就可以建立可靠的TCP连接。