

计算机网络



第5章 习题课

第5章 作业

5-1 试说明运输层在协议栈中的地位和作用，运输层的通信和网络层的通信有什么重要区别？为什么运输层是必不可少的？

答：

运输层处于面向通信部分的最高层，同时也是用户功能中的最低层，向它上面的应用层提供服务。

运输层为应用进程之间提供端到端的逻辑通信，但网络层是为主机之间提供逻辑通信。

运输层为各种应用进程之间通信提供“可靠或尽力而为”的两类服务质量，同时实现了应用进程在运输层上的复用和分用。

5-21 假定使用连续ARQ协议中，发送窗口大小是3，而序列范围[0,15]，而传输媒体保证在接收方能够按序收到分组。在某时刻，接收方，下一个期望收到序号是5。

试问：

- (1) 在发送方的发送窗口中可能有出现的序号组合有哪几种？
- (2) 接收方已经发送出去的、但在网络中（即还未到达发送方）的确认分组可能有哪些？说明这些确认分组是用来确认哪些序号的分组。

答：(1) 序列号到4为止的分组都已收到。若这些确认都已到发送方，则发送窗口为[5,7]。假设所有的确认都丢失了，发送方没有收到确认，则发送窗口应为[2,4]。因此发送窗口可以是[2,4],[3,5],[4,6],[5,7]中的任何一个。

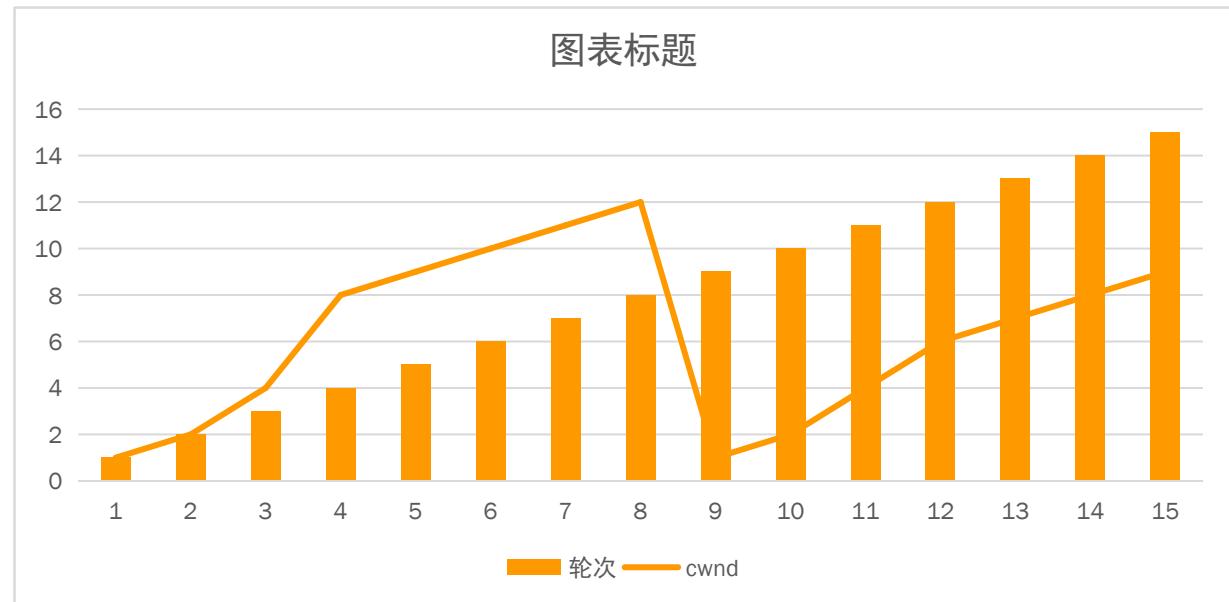
(2) 接收方期望接收5的分组，且发送窗口为3，所以正在传输确认分组可能有2, 3, 4。这些分组用以确认序号为2, 3, 4的分组。

- 5-23. 主机A向主机B连续发送了两个TCP报文段，其序号分别为70和100。试问：
 - (1) 第一个报文段携带了多少个字节的数据？
 - (2) 主机B收到第一个报文段后发回的确认中的确认号应当是多少？
 - (3) 如果主机B收到第二个报文段后发回的确认中的确认号是180，试问A发送的第二个报文段中的数据有多少字节？
 - (4) 如果A发送的第一个报文段丢失了，但第二个报文段到达了B。B在第二个报文段到达后向A发送确认。试问这个确认号应为多少？
- - 解： (1) 第一个报文段的数据序号是70到99，共30字节的数据。
 - (2) 确认号应为100.
 - (3) 80字节。
 - (4) 70

5-38 设TCP的SSthresh的初始值为8（单位为报文段）。当拥塞窗口上升到12时网络发生了超时，TCP使用慢开始和拥塞避免。试分别求出第1轮到第15轮次传输的各拥塞窗口大小。你能说明拥塞窗口每一次变化的原因吗？

答：

sstresh	轮次	cwnd
8	1	1
	2	2
	3	4
	4	8
	5	9
	6	10
	7	11
6	8	12
9	1	
10	2	
11	4	
12	6	
	13	7
	14	8
	15	9



5-39 TCP的拥塞窗口 cwnd 大小与传输轮次 n 的关系如下所示：

cwnd	1	2	4	8	16	32	33	34	35	36	37	38	39
<i>n</i>	1	2	3	4	5	6	7	8	9	10	11	12	13
cwnd	40	41	42	21	22	23	24	25	26	1	2	4	8
<i>n</i>	14	15	16	17	18	19	20	21	22	23	24	25	26

- (1) 试画出如图 5-25 所示的拥塞窗口与传输轮次的关系曲线。
- (2) 指明 TCP 工作在慢开始阶段的时间间隔。
- (3) 指明 TCP 工作在拥塞避免阶段的时间间隔。
- (4) 在第 16 轮次和第 22 轮次之后发送方是通过收到三个重复的确认还是通过超时检测到丢失了报文段？
- (5) 在第 1 轮次、第 18 轮次和第 24 轮次发送时，门限 ssthresh 分别被设置为多大？
- (6) 在第几轮次发送出第 70 个报文段？
- (7) 假定在第 26 轮次之后收到了三个重复的确认，因而检测出了报文段的丢失，那么拥塞窗口 cwnd 和门限 ssthresh 应设置为多大？

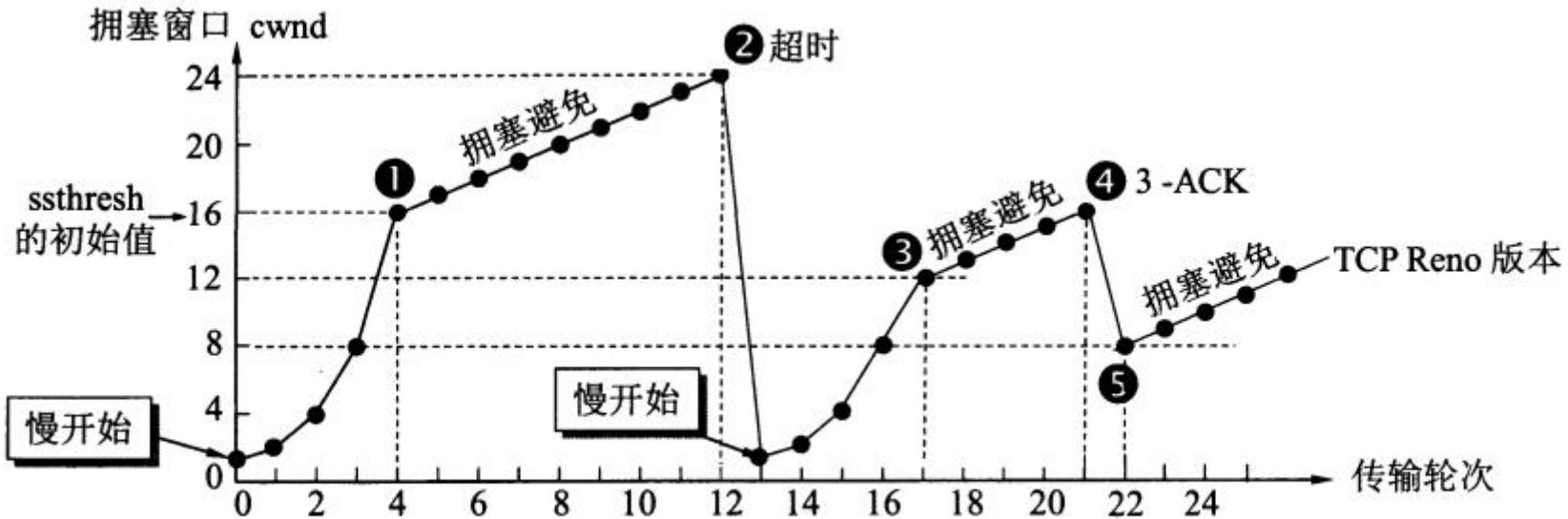
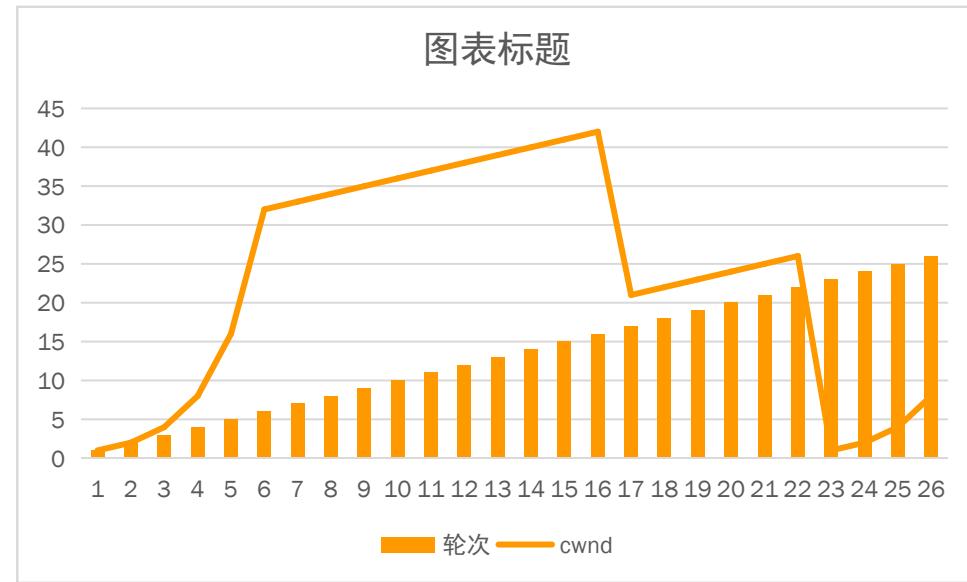


图 5-25 TCP 拥塞窗口 cwnd 在拥塞控制时的变化情况

sstresh	轮次	cwnd
	1	1
	2	2
	3	4
	4	8
	5	16
	6	32
	7	33
	8	34
	9	35
	10	36
	11	37
	12	38
	13	39
	14	40
	15	41
	16	42
21	17	21
	18	22
	19	23
	20	24
	21	25
	22	26
	23	1
	24	2
	25	4
	26	8



- 5-45 解释为什么突然释放运输连接就可能会丢失用户数据，而使用TCP的连接释放方法就可以保证不丢失数据。
-

5—46 试用具体例子说明为什么在运输连接建立时要使用三次握手。说明如不这样做可能会出现什么情况。

答：

3次握手完成两个重要的功能，既要双方做好发送数据的准备工作（双方都知道彼此已准备好），也要允许双方就初始序列号进行协商，这个序列号在握手过程中被发送和确认。

假定B给A发送一个连接请求分组，A收到了这个分组，并发送了确认应答分组。按照两次握手的协定，A认为连接已经成功地建立了，可以开始发送数据分组。可是，B在A的应答分组在传输中被丢失的情况下，将不知道A是否已准备好，不知道A建议什么样的序列号，B甚至怀疑A是否收到自己的连接请求分组，在这种情况下，B认为连接还未建立成功，将忽略A发来的任何数据分组，只等待连接确认应答分组。

而A发出的分组超时后，重复发送同样的分组。这样就形成了死锁。