

**5-01 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的区别？为什么运输层是必不可少的？**

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（面向主机，承担路由功能，即主机寻址及有效的分组交换）。各种应用进程之间通信需要“可靠或尽力而为”的两类服务质量，必须由运输层以复用和分用的形式加载到网络层。

**5-03 当应用程序使用面向连接的 TCP 和无连接的 IP 时，这种传输是面向连接的还是无连接的？**

都是。这要在不同层次来看，在运输层是面向连接的，在网络层则是无连接的。

**5-05 试举例说明有些应用程序愿意采用不可靠的 UDP,而不愿意采用可靠的 TCP。**

由于语音信息具有一定的冗余度，人耳对 VOIP 数据报损失由一定的承受度，但对传输时延的变化较敏感。有差错的 UDP 数据报在接收端被直接抛弃，TCP 数据报出错则会引起重传，可能带来较大的时延扰动。因此 VOIP 宁可采用不可靠的 UDP，而不愿意采用可靠的 TCP。

**5-09 端口的作用是什么？为什么端口号要划分为三种？**

端口的作用是对 TCP/IP 体系的应用进程进行统一的标志，使运行不同操作系统的计算机的应用进程能够互相通信。熟知端口，数值一般为 0~1023.标记常规的服务进程;登记端口号，数值为 1024~49151，标记没有熟知端口号的非常规的服务进程；

**5-11 某个应用进程使用运输层的用户数据报 UDP,然后继续向下交给 P 层后，又封装成 P 数据报。既然都是数据报，是否可以跳过 UDP 而直接交给 P 层？哪些功能 UDP 提供了但 P 没有提供？**

不可跳过 UDP 而直接交给 IP 层 IP 数据报 IP 报承担主机寻址，提供报头检错;只能找到目的主机而无法找到目的进程。UDP 提供对应用进程的复用和分用功能，以及提供对数据差分的差错检验。

**5-23 主机 A 向主机 B 连续发送了两个 TCP 报文段，其序号分别是 70 和 100。试问：**

**(1) 第一个报文段携带了多少字节的数据？**

第一个报文段的数据序号是 70 到 99，共 30 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？**

确认号应为 100。

**(3)如果 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？**

80 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？**

确认号应为 70

**5-24 一个 TCP 连接下面使用 256kbit/s 的链路，其端到端时延为 128ms。经测试，发现吞吐量只有 120kbit/s。试问发送窗口 W 是多少？（提示：可以有两种答案，取决于接收端发出确认的时机。）**

来回路程的时延等于  $256\text{ms}(=128\text{ms}\times 2)$ 。设窗口值为  $X$ (以字节为单位),假定一次最大发送量等于窗口值,且发射时间等于  $256\text{ms}$ ,那么,每发送一次都得停下来期待再次得到下一窗口的确认,以得到新的发送许可。这样,发射时间等于停止等待应答的时间结果,测到的平均吞吐率就等于发送速率的一半,即  $8X+(256\times 1000)=256\times 0.001X=8192$  所以,窗口值为  $8192$ 。

**5-28** 主机 A 向主机 B 发送 TCP 报文段,首部中的源端口是  $m$  而目的端口是  $n$ 。当 B 向 A 发送回信时,其 TCP 报文段的首部中的源端口和目的端口分别是什么?

分别是  $n$  和  $m$

**5-29** 在使用 TCP 传送数据时,如果有一个确认报文段丢失了,也不一定会引起与该确认报文段对应的数据的重传。试说明理由。

还未重传就收到了对更高序号的确认。

**5-33** 假定 TCP 在开始建立连接时,发送方设定超时重传时间  $\text{RTO}=6$  秒。

(1)当发送方收到对方的连接确认报文段时,测量出  $\text{RTT}$  样本值为  $1.5$  秒。试计算现在的  $\text{RTO}$  值。

据 RFC2988 建议,  $\text{RTO}=\text{RTTs}+4*\text{RTTd}$ 。其中  $\text{RTTd}$  是  $\text{RTTs}$  的偏差加权均值。初次测量时,

$$\text{RTTd}(1)=\text{RTT}(1)/2$$

$$\text{RTTd}(i)=(1-\text{Beta})*\text{RTTd}(i-1)+\text{Beta}*{\text{RTTs}-\text{RTT}(i)};$$

$$\text{Beta}=1/4$$

依题意,  $\text{RTT}(1)$  样本值为  $1.5$  秒,则

$$\text{RTTs}(1)=\text{RTT}(1)=1.5\text{s} \quad \text{RTTd}(1)=\text{RTT}(1)/2=0.75\text{s}$$

$$\text{RTO}(1)=\text{RTTs}(1)+4\text{RTTd}(1)=1.5+4*0.75=4.5(\text{s})$$

(2)当发送方发送数据报文段并收到确认时,测量出  $\text{RTT}$  样本值为  $2.5$  秒。试计算现在的  $\text{RTO}$  值。

$$\text{RTT}(2)=2.5$$

$$\text{RTTs}(1)=1.5\text{s}$$

$$\text{RTTd}(1)=0.75\text{s}$$

$$\text{RTTd}(2)=(1-\text{Beta})*\text{RTTd}(1)+\text{Beta}*{\text{RTTs}(1)-\text{RT}(2)}$$

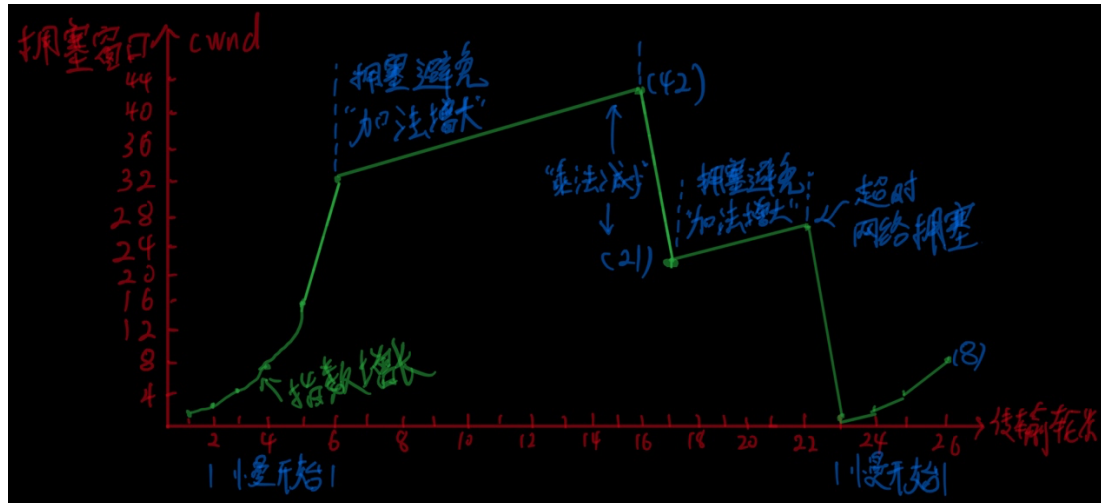
$$=0.75*3/4+\{1.5-2.5\}/4=13/16$$

$$\text{RTO}(2)=\text{RTTs}(1)+4\text{RTTd}(2)=1.5+4*13/16=4.75\text{s}$$

**5-39** TCP 的拥塞窗口  $\text{cwnd}$  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
$n$	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
$n$	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。

[1, 6]和[23, 26]

(3) 指明 TCP 工作在拥塞避免阶段的时间间隔。

[6, 16]和[17, 22]

(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？

在第 22 轮次之后发送方是通过超时检测到丢失了报文段，因为题目给出，下一个轮次的拥塞窗口下降到 1 了

(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？

在第 18 轮次发送时，门限  $ssthresh$  被设置为发生拥塞时拥塞窗口 42 的一半，即 21。

在第 24 轮次发送时，门限  $ssthresh$  被设置为发生拥塞时拥塞窗口 26 的一半，即 13。

(6) 在第几轮次发送出第 70 个报文段？

第 1 轮次发送报文段 1。(cwnd=1)

第 2 轮次发送报文段 2,3。(cwnd =2)

第 3 轮次发送报文段 4~7。(cwnd =4)

第 4 轮次发送报文段 8~15。(cwnd =8)

第 5 轮次发送报文段 16~31。(cwnd = 16)

第 6 轮次发送报文段 32~63。(cwnd = 32)

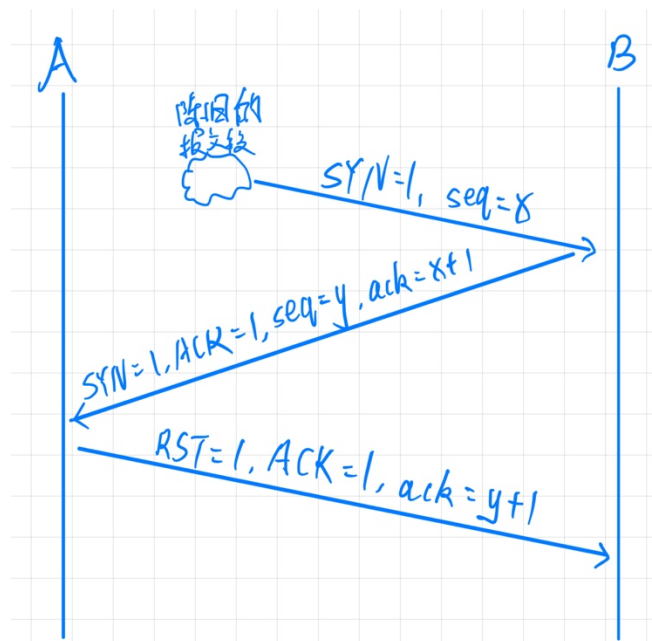
第 7 轮次发送报文段 64~94。(cwnd = 33)

因此第 70 报文段在第 7 轮次发送出。

(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？

检测出了报文段的丢失时拥塞窗口  $cwnd$  是 8，因此拥塞窗口  $cwnd$  的数值应当减半，等于 4，而门限  $ssthresh$  应设置为检测出报文段丢失时拥塞窗口 8 的一半，即 4。

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报文握手。说明如不这样做可能会出现什么情况。



在上一个 TCP 连接中，A 向 B 发送的连接请求 SYN 报文段滞留在网络中的某处。于是 A 超时重传，与 B 建立了 TCP 连接，交换了数据，最后也释放了 TCP 连接。但滞留在网络中某处的陈旧的 SYN 报文段，现在突然传送到 B 了。如果不使用三次握手，那么 B 就以为 A 现在请求建立 TCP 连接，于是就分配资源，等待 A 传送数据。但 A 并没有想要建立 TCP 连接，也不会向 B 传送数据。B 就白白等待着 A 发送数据。

如果使用三次握手，那么 B 在收到 A 发送的陈旧的 SYN 报文段后，就向 A 发送 SYN 报文段，选择自己的序号  $seq = y$ ，并确认收到 A 的 SYN 报文段，其确认号  $ack = x + 1$ 。当 A 收到 B 的 SYN 报文段时，从确认号就可得知不应当理睬这个 SYN 报文段(因为 A 现在并没有发送  $seq = x$  的 SYN 报文段)。这时，A 发送复位报文段。在这个报文段中， $RST = 1, ACK = 1$ ，其确认号  $ack = y + 1$ 。我们注意到，虽然 A 拒绝了 TCP 连接的建立(发送了复位报文段)，但对 B 发送的 SYN 报文段还是确认收到了。

B 收到 A 的 RST 报文段后，就知道不能建立 TCP 连接，不会等待 A 发送数据了。

**5-47** 一客户向服务器请求建立 TCP 连接。客户在 TCP 连接建立的三报文握手中的最后一个报文段中捎带上一些数据，请求服务器发送一个长度为字节的文件。假定：

- (1) 客户和服务器之间的数据传送速率是  $R$  字节/秒，客户与服务器之间的往返时间是  $RTT$ (固定值)。
- (2) 服务器发送的 TCP 报文段的长度都是  $M$  字节，而发送窗口大小是  $nM$  字节。
- (3) 所有传送的报文段都不会出现差错(无重传)，客户收到服务器发来的报文段后就及时发送确认。
- (4) 所有的协议首部开销都可忽略，所有确认报文段和连接建立阶段的报文段的长度都可忽略(即忽略这些报文段的发送时间)。

试证明，从客户开始发起连接建立到接收服务器发送的整个文件所需的时间  $T$  是：

$$T = 2RTT + L/R$$

$$\text{当 } nM > R(RTT) + M$$

$$\text{或 } T = 2RTT + L/R + (K - 1)[M/R + RTT - nM/R]$$

$$\text{当 } nM < R(RTT) + M$$

其中， $K = \lceil L/nM \rceil$ ，符号  $\lceil x \rceil$  表示若  $x$  不是整数，则把  $x$  的整数部分加 1。

(提示：求证的第一个等式发生在发送窗口较大的情况，可以连续把文件发送完。求证的第二个等式发生在发送窗口较小的情况，发送几个报文段后就必须停顿下来，等收到确认)

后再继续发送。建议先画出双方交互的时间图，然后再进行推导。)

(a)先看图

$M/R$  是一个报文段的发送时间(一个报文段的长度除以数据率)。

$nM$  是窗口大小,  $nM/R$  是把窗口内的数据都发送完所需的时间。

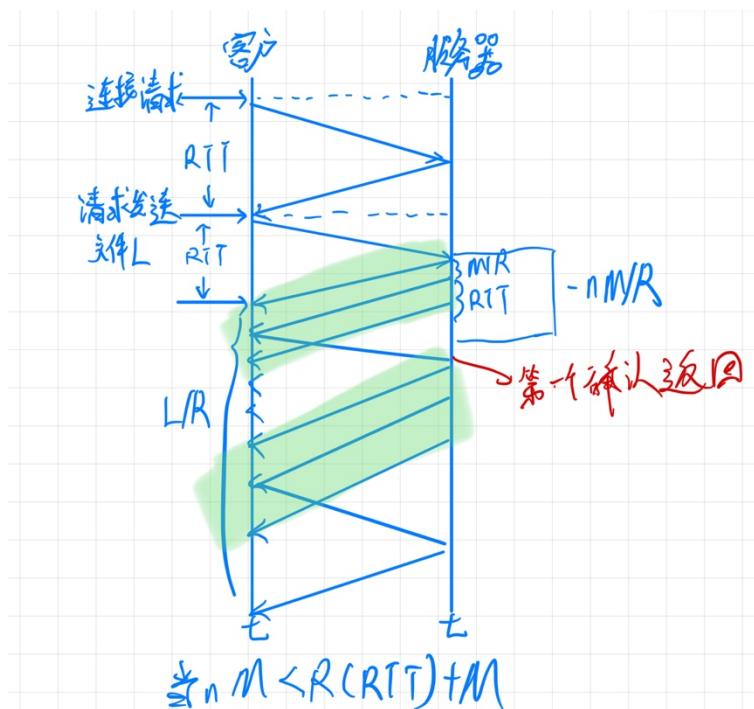
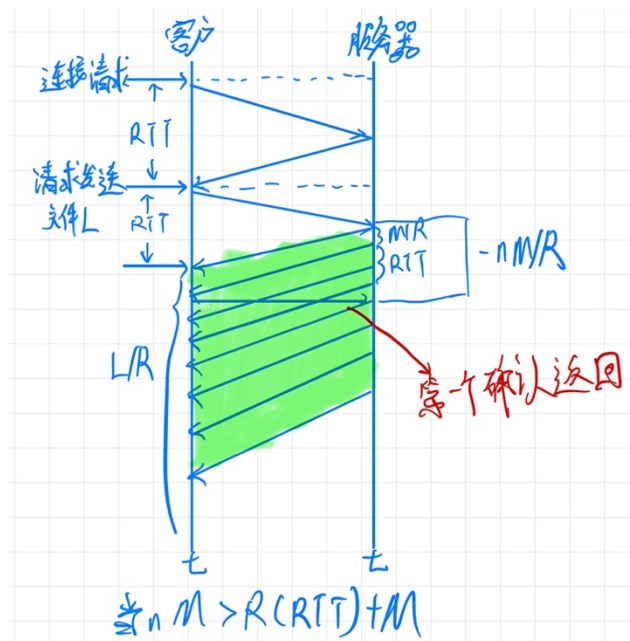
如果  $nM/R > M/R + RTT$ , 那么服务器在发送窗口内的数据还没有发送完, 就收到客户的确认, 因此。服务器可以连续发送, 直到全部数据发送完毕。

这个不等式两边都乘以  $R$ , 就得出等效的条件:

当  $nM > R(RTT) + M$  时, 发送窗口内的数据还没有发送完就收到确认, 因此。服务器可以连续发送, 直到全部数据发送完毕。

因此, 客户接收全部数据所需的时间是:

$T = 2RTT + L/R$  当  $nM > R(RTT) + M$



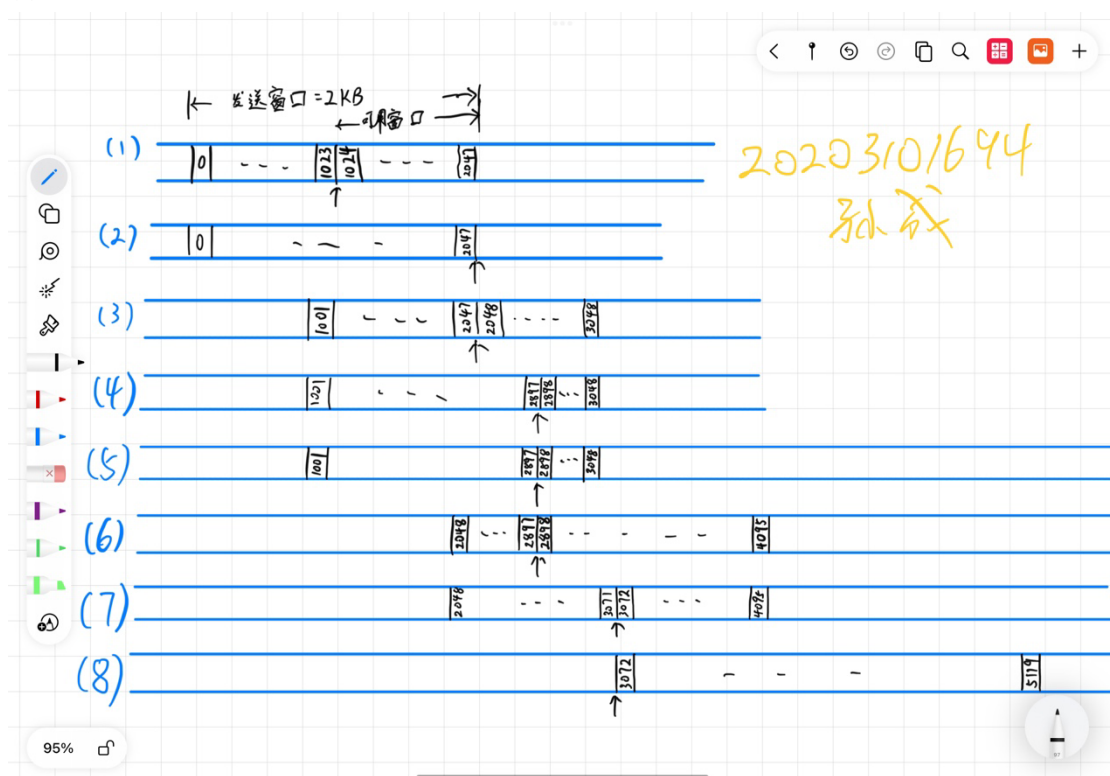
(b) 当  $nM < R(RTT) + M$  时, 服务器把发送窗口内的数据发送完毕时还收不到确认, 因此必须停止发送。从图 T-5-47(b) 可看出, 停止的时间间隔是  $M/R + RTT - nM/R$ 。

整个文件  $L$  要划分为  $K = \lceil L/nM \rceil$  次传送, 停止的时间间隔有  $(K-1)$  个。这样就证明了求证的公式:

$$T = 2RTT + L/R + (K-1)[M/IR + RTT - nM/IR] \quad \text{当 } nM < R(RTT) + M$$

**5-61** 在本题中列出的 8 种情况下, 画出发送窗口的变化, 并标明可用窗口的位置。已知主机 A 要向主机 B 发送 3KB 的数据。在 TCP 连接建立后, A 的发送窗口大小是 2KB。A 的初始序号是 0。

- (1) 一开始 A 发送 1KB 的数据。
- (2) 接着 A 就一直发送数据, 直到把发送窗口用完。
- (3) 发送方 a 收到对第 1000 号字节的确认报文段。
- (4) 发送方 A 再发送 850B 的数据。
- (5) 发送方 A 收到  $ack = 900$  的确认报文段。
- (6) 发送方 A 收到对第 2047 号字节的确认报文段。
- (7) 发送方 A 把剩下的数据全部都发送完。
- (8) 发送方 A 收到  $ack = 3072$  的确认报文段。



(1) 我们应当注意到, 发送窗口=2KB 就是  $2 \times 1024 = 2048$  字节。因此, 发送窗口应当是从 0 到第 2047 字节为止, 长度是 2048 字节。A 开始就发送了 1024 字节, 因此发送窗口中左边的 1024 个字节已经用掉了(窗口的这部分为灰色), 而可用窗口是白色的, 从第 1024 字节到第 2047 字节为止。请注意, 不是到第 2048 字节为止, 因为第一个字节的编号是 0 而不是 1。

(2) 发送方 A 一直发送数据, 直到把发送窗口用完。这时, 整个窗口都已用掉了, 可用窗口的大小已经是零了, 一个字节也不能再发送了。

(3) 发送方 A 收到对第 1000 号字节的确认报文段, 表明 A 收到确认号  $ack = 1001$  的确认报

文段。这时，发送窗口的后沿向前移动，发送窗口从第 1001 字节(不是从第 1000 字节)到第 3048 字节(不是第 3047 字节)为止。可用窗口从第 2048 字节到第 3048 字节。

(4) 发送方 A 再发送 850 字节，使得可用窗口的后沿向前移动 850 字节，即移动到 2898 字节。现在的可用窗口从第 2898 字节到第 3048 字节。

(5) 发送方 A 收到  $\text{ack} = 900$  的确认报文段，不会对其窗口状态有任何影响。这是个迟到的确认。

(6) 发送方 A 收到对第 2047 号字节的确认报文段。A 的发送窗口再向前移动。现在的发送窗口从第 2048 字节开始到第 4095 字节。可用窗口增大了，从第 2898 字节到第 4095 字节。

(7) 发送方 A 把剩下的数据全部都发送完。发送方 A 共有 3KB (即 3072 字节)的数据，其编号从 0 到 3071。因此现在的可用窗口变小了，从第 3072 字节到第 4095 字节。

(8) 发送方 A 收到  $\text{ack} = 3072$  的确认报文段，表明序号在 3071 和这以前的报文段都收到了，后面期望收到的报文段的序号是从 3072 开始。因此新的发送窗口的位置又向前移动，从第 3072 号到第 5119 号。整个发送窗口也就是可用窗口。

**5-65 假定主机 A 向 B 发送一个 TCP 报文段。在这个报文段中，序号是 50，而数据一共有 6 字节长。试问，在这个报文段中的确认字段是否应当写入 56?**

在这个报文段中的确认字段应当写入的是 A 期望下次收到 B 发送的数据中的第一个字节的编号，而这个数值是 A 已经收到的数据中的最后一个字节的编号加 1。然而这些在题目中并未给出。题目给出的是 A 向 B 发送的数据中第一个字节的编号是 50，并且在这个报文段中共有 6 字节的数据。这些都和此报文段中的确认字段是什么毫无关系。因此，现在无法知道在这个报文段中的确认字段应当写入的数值。