**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面向无连接的？**

答：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—06 接收方收到有差错的 UDP 用户数据报时应如何处理？**

答：简单地丢弃。

**5—09 端口的作用是什么？为什么端口要划分为三种？**

答：端口是用来标志进程的。端口也就是协议端口号。但这是软件端口，是应用层各种协议进程与运输实体进行层间交互的一种地址。这和路由器或交换机上的硬件端口是完全不同的概念，硬件端口是不同的硬件设备进行交互的接口。

两个计算机中的进程要互相通信，不仅必须知道对方的IP 地址（为了找到对方的计算机），而且还要知道对方的的端口号（为了找到对方计算机中的应用进程）。

端口号有三种。不同的端口类别有其特殊的用途。例如，客户端是通信的发起方，而服务器是服务的提供方。它们对端口的使用要求是不同的。这三种端口是：

（1）熟知端口或系统端口号，数值一般为 0 ~ 1023。

（2）登记端口号，数值为 1024 ~ 49151，为没有熟知端口号的应用程序使用的。使用这个范围的端口号必须在 IANA 登记，以防止重复。

以上两种端口号是服务器端使用的端口号。下面一种是客户端使用的端口号。

（3）短暂端口号，数值为 49152 ~ 65535，这类端口号仅在客户进程运行时才动态选择，是留给客户进程选择暂时使用。

**5—11 某个应用进程使用运输层的用户数据报 UDP，然而继续向下交给 IP 层后，又封装成 IP 数据报。既然都是数据报，可否跳过 UDP 而直接交给 IP 层？哪些功能 UDP提供了但 IP 没提提供？**

答：不可跳过 UDP 而直接交给 IP 层。IP 数据报只能找到目的主机而无法找到目的进程。如果应用进程直接把数据交给下面的IP层，那么在传送到对方IP层后，就只能交付目的主机，但不知道应当交付哪一个应用进程。UDP 提供对应用进程的复用和分用功能，以及提供对数据部分的差错检验。这些功能IP层没有提供。

**5—12 一个应用程序用 UDP，到 IP 层把数据报再划分为 4 个数据报片发送出去，结果前两个数据报片丢失，后两个到达目的站。过了一段时间应用程序重传UDP，而 IP**

**层仍然划分为 4 个数据报片来传送。结果这次前两个到达目的站而后两个丢失。试**

**问：在目的站能否将这两次传输的 4 个数据报片组装成完整的数据报？假定目的站第一次收到的后两个数据报片仍然保存在目的站的缓存中。**

答：不行。重传时，IP 数据报的标识字段会有另一个标识符。仅当标识符相同的 IP 数据报片才能组装成一个 IP 数据报。前两个 IP 数据报片的标识符与后两个 IP 数据报片的标识符不同， 因此不能组装成一个 IP 数据报。

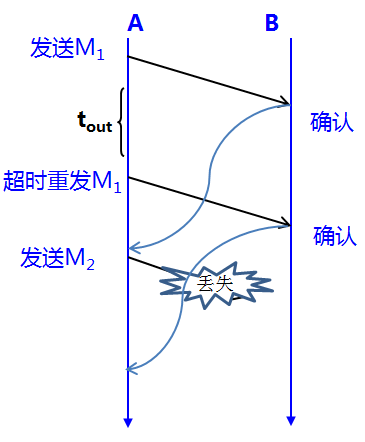
**5—16 在停止等待协议中如果不使用编号是否可行？为什么？**

答: 不行。见下图。

A发送报文段M1，B收到后发送确认（不编号）。但这个确认很晚才传送到A。A在没有等到确认时，超时重传了M1。

B发送的第一个确认最后到了A，于是A发送下一个报文段M2，但M2丢失了。

B收到A发送的重传的M1。但B并不知道是重传的，因为报文段没有编号。B无法判断是重传的老报文段，还是新的报文段。B只能把A发送的重传的M1收下，并发送确认。但这个确认使A认为是对其发送的M2的确认，于是以为发送的两个报文段B都收到了。实际上B收到了两个重复的报文段。所以在停止等待协议中，不使用编号是不行的。



**5—17 在停止等待协议中，如果收到重复的报文段时不予理睬（即悄悄地丢弃它而其他什么也没做）是否可行？试举出具体的例子说明理由。**

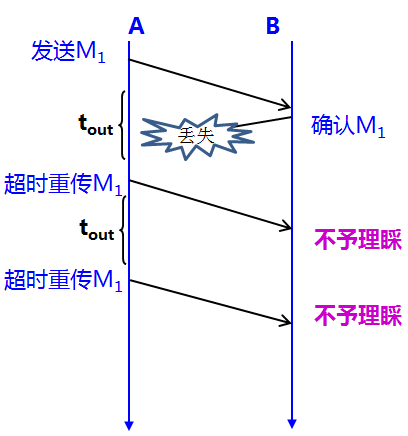
答：不行。见下图。

A发送报文段M1，B收到后发送确认。但这个确认丢失了。

A超时重传报文段M1。B收到后不予理睬。这就导致A再次超时重传M1.

B收到重复的报文段都不予理睬，A就一直超时重传M1。

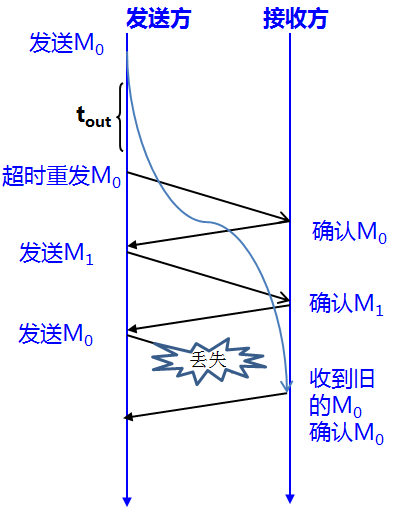
可见，如果收到重复的报文段不予理睬是不行的。



**5—18 假定在运输层使用停止等待协议。发送发在发送报文段 M0 后再设定的时间内未收到确认，于是重传M0，但 M0 又迟迟不能到达接收方。不久，发送方收到了迟到的对 M0 的确认，于是发送下一个报文段M1，不久就收到了对 M1 的确认。**

**接着发送方发送新的报文段M0，但这个新的M0 在传送过程中丢失了。正巧，一开始就滞留在网络中的 M0 现在到达接收方。 接收方无法分辨M0 是旧的。 于是收下M0，并发送确认。显然，接收方后来收到的M0 是重复的，协议失败了。**

答：



由图可见，旧的 M0 被当成新的 M0。可见运输层不能使用停止等待协议（编号只有0和1两种）。

**5—19 试证明：当用 n 比特进行分组的编号时，若接收到窗口等于 1（即只能按序接收分组） ，当仅在发送窗口不超过2n-1 时，连接 ARQ 协议才能正确运行。窗口单位是分组。**

解：课后答案有解析，也可见下面的解析。

假设发送窗口的大小为2n，发送方发送了0号帧，接收窗口发送ACK1(0号帧已收到，希望接收1号帧，但是ACK1丢失)，接着发送方发送了1号帧，接收窗口发送ACK2(1号帧已收到，希望接收2号帧，但是ACK2丢失)，以此类推，直到发送方发了第2n-1号帧，接收方发送ACK2 n(丢失)，此时不能再发送数据了

因为已经发送了2 n个帧，但一个确认都没有收到，所以过一段时间0号帧的计时器会到达预定时间进行重发，此时发过去接收方认为是新一轮的0号帧还是旧一轮重传的呢?接收方并不知道，很有可能接收方就把该0号帧当作新一轮的帧接收了，但实际上这个0号帧是重传的，所以出现了错误，即发送窗口的大小不可能为2 n。

现在假设发送窗口的大小为2 n-1，情况和上面一样，发送方发送了0~2 n -2号帧，接收方发送的确认帧都丢失了，如果没有丢失就应该接着传2 n -1号帧，但是丢失了，发送方应该发送0号帧。由于这种情况接收方可以判断出来(下一帧只要不是第2n-1号帧就是重传),因此不会发生错误。如果发送窗口尺寸小于2 n-1，那就更不会发生错误了。

**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）B期望收到下一个报文段的第一个数据字节的序号是100，因此确认号应为 100。

（3）A发送的第二个报文段中的数据中的字节数是180-100=80字节。

（4）B在第二个报文段到达后向A发送确认，确认号是70。

**5—26 为什么在TCP 首部中有一个首部长度字段，而UDP 的首部中就没有这个字段？**

答：TCP 首部除固定长度部分外，还有选项，因此TCP首部长度是可变的。 UDP首部长度是固定的，不需要这个字段。

**5—27一个 TCP 报文段的数据部分最多为多少个字节？为什么？如果用户要传送的数据的字节长度超过TCP 报文字段中的序号字段可能编出的最大序号，问还能否用TCP 来传送？**

答：一个TCP报文段的数据部分最多是65495 字节，此数据部分加上 TCP 首部的 20 字节，再加上 IP 首部的 20 字节，正好是 IP 数据报的最大长度 65535.（当然，若 IP 首部包含了选项，则 IP 首部长度超过 20 字节，这时 TCP 报文段的数据部分的长度将小于 65495 字节。 ）

如果用户要传送的数据的字节长度超过 TCP 报文段中的序号字段可能编出的最大序号， 通过循环使用序号，仍能用 TCP 来传送。

**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 报文段的首部中源端口和目的端口分别是什么？**

答：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。**

答：还未重传就收到了对更高序号的确认。

**5-74 流量控制和拥塞控制的最主要的区别是什么？发送窗口的大小取决于流量控制还是拥塞控制？**

简单地说，流量控制是在一条TCP连接中的接收端采用的措施，用来限制对方（发送端）发送报文段的速率，以使接收端来得及接收。流量控制只控制一个发送端。

拥塞控制是用来控制TCP连接中发送端发送报文段的速率，以免使互联网中的某处产生过载。拥塞控制可能会同时控制许多个发送端，限制它们的发送速率。不过每一个发送端只知道自己应当怎样调整发送速率，而不知道在互联网中还有哪些主机被限制了发送速率。

我们知道，发送窗口的上限值是Min[rwnd,cwnd]，即发送窗口的数值不能超过接收窗口和拥塞窗口中的较小的一个。接收窗口的大小体现了接收端对发送端施加的流量控制，而拥塞窗口的大小则是整个互联网的负载情况对发送端施加的拥塞控制。因此，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，即取决于接收端的接收能力。但当拥塞窗口小于接收窗口时，则发送窗口的大小取决于拥塞控制，即取决于整个网络的拥塞状况。