

TCP 也使用流水线,使得发送方在任意时刻都可以有多个已发出但还未被确认的报文段存在。我们在前面已经看到,当报文段长度与往返时延之比很小时,流水线可显著地增加一个会话的吞吐量。一个发送方可以具有的未被确认报文段的确切个数是由 TCP 的流量控制和拥塞控制机制决定的。TCP 流量控制将在本节后面讨论;TCP 拥塞控制将在 3.7 节中讨论。我们暂时只需知道 TCP 发送方使用了流水线即可。

### 3.5.4 可靠数据传输

前面讲过,因特网的网络层服务(IP 服务)是不可靠的。IP 不保证数据报的交付,不保证数据报的按序交付,也不保证数据报中数据的完整性。对于 IP 服务,数据报能够溢出路由器缓存而永远不能到达目的地,数据报也可能是乱序到达,而且数据报中的比特可能损坏(由 0 变为 1 或者相反)。由于运输层报文段是被 IP 数据报携带着在网络中传输的,所以运输层的报文段也会遇到这些问题。

TCP 在 IP 不可靠的尽力而为服务之上创建了一种可靠数据传输服务(reliable data transfer service)。TCP 的可靠数据传输服务确保一个进程从其接收缓存中读出的数据流是无损坏、无间隔、非冗余和按序的数据流;即该字节流与连接的另一方端系统发送出的字节流是完全相同。TCP 提供可靠数据传输的方法涉及我们在 3.4 节中所学的许多原理。

在我们前面研发可靠数据传输技术时,曾假定每一个已发送但未被确认的报文段都与一个定时器相关联,这在概念上是最简单的。虽然这在理论上很好,但定时器的管理却需要相当大的开销。因此,推荐的定时器管理过程[RFC 6298]仅使用单一的重传定时器,即使有多个已发送但还未被确认的报文段。在本节中描述的 TCP 协议遵循了这种单一定时器的推荐。

我们将以两个递增的步骤来讨论 TCP 是如何提供可靠数据传输的。我们先给出一个 TCP 发送方的高度简化的描述,该发送方只用超时来恢复报文段的丢失;然后再给出一个更全面的描述,该描述中除了使用超时机制外,还使用冗余确认技术。在接下来的讨论中,我们假定数据仅向一个方向发送,即从主机 A 到主机 B,且主机 A 在发送一个大文件。

图 3-33 给出了一个 TCP 发送方高度简化的描述。我们看到在 TCP 发送方有 3 个与发送和重传有关的主要事件:从上层应用程序接收数据;定时器超时和收到 ACK。一旦第一个主要事件发生,TCP 从应用程序接收数据,将数据封装在一个报文段中,并把该报文段交给 IP。注意到每一个报文段都包含一个序号,如 3.5.2 节所讲的那样,这个序号就是该报文段第一个数据字节的字节流编号。还要注意如果定时器还没有为某些其他报文段而运行,则当报文段被传给 IP 时,TCP 就启动该定时器。(将定时器想象为与最早的未被确认的报文段相关联是有帮助的。)该定时器的过期间隔是 TimeoutInterval,它是由 3.5.3 节中所描述的 EstimatedRTT 和 DevRTT 计算得出的。

第二个主要事件是超时。TCP 通过重传引起超时的报文段来响应超时事件。然后 TCP 重启定时器。

TCP 发送方必须处理的第三个主要事件是一个来自接收方的确认报文段(ACK)的到达(更确切地说是一个包含了有效 ACK 字段值的报文段)。当该事件发生时,TCP 将 ACK 的值  $y$  与它的变量 SendBase 进行比较。TCP 状态变量 SendBase 是最早未被确认的字节的