

文的确认号是 100，第二个确认报文的确认号是 120。现在假设在超时之前这两个报文段中没有一个确认报文到达主机 A。当超时事件发生时，主机 A 重传序号 92 的第一个报文段，并重启定时器。只要第二个报文段的 ACK 在新的超时发生以前到达，则第二个报文段将不会被重传。

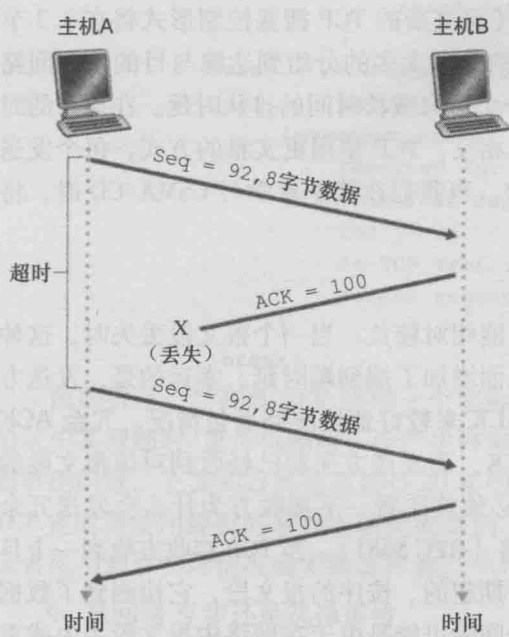


图 3-34 由于确认丢失而重传

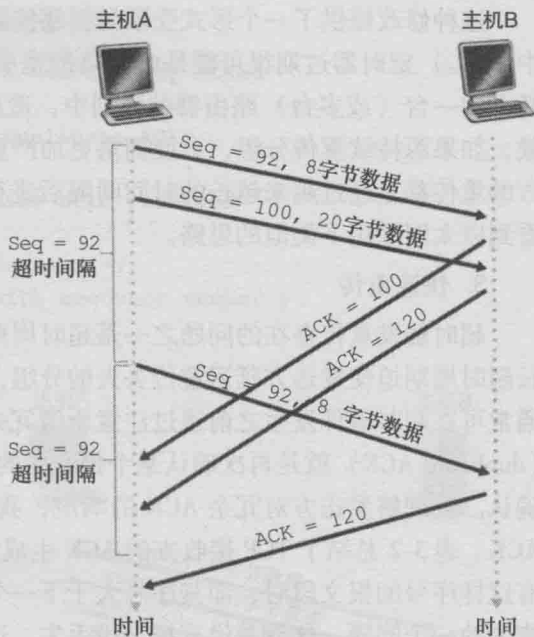


图 3-35 报文段 100 没有重传

在第三种也是最后一种情况中，假设主机 A 与在第二种情况中完全一样，发送两个报文段。第一个报文段的确认报文在网络丢失，但在超时事件发生之前主机 A 收到一个确认号为 120 的确认报文。主机 A 因而知道主机 B 已经收到了序号为 119 及之前的所有字节；所以主机 A 不会重传这两个报文段中的任何一个。这种情况在图 3-36 中进行了图示。

2. 超时间隔加倍

我们现在讨论一下在大多数 TCP 实现中所做的一些修改。首先关注的是在定时器时限过期后超时间隔的长度。在这种修改中，每当超时事件发生时，如前所述，TCP 重传具有最小序号的还未被确认的报文段。只是每次 TCP 重传时都会将下一次的超时间隔设为先前值的两倍，而不是用从 EstimatedRTT 和 DevRTT 推算出的值（如在 3.5.3 节中所描述的）。例如，假设当定时器第一次过期时，与最早的未被确认的报文段相关联的 TimeoutInterval 是 0.75 秒。TCP 就会重传该

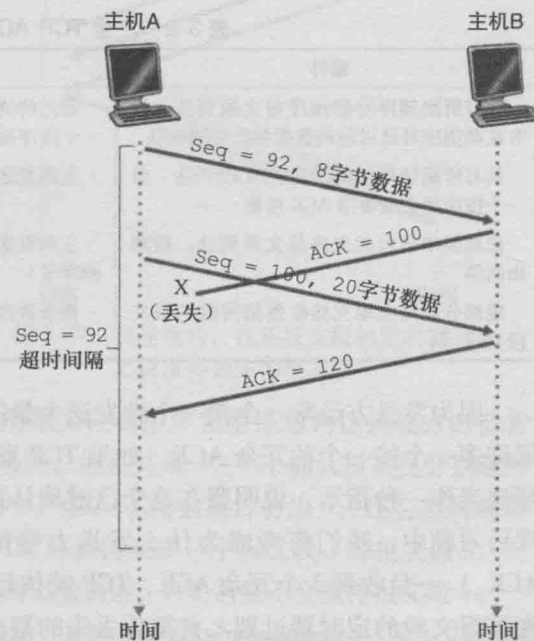


图 3-36 累积确认避免了第一个报文段的重传