

```
event: ACK received, with ACK field value of y
    if (y > SendBase) {
        SendBase=y
        if (there are currently any not yet
            acknowledged segments)
            start timer
    }
else { /* a duplicate ACK for already ACKed
    segment */
    increment number of duplicate ACKs
    received for y
    if (number of duplicate ACKs received
        for y==3)
        /* TCP fast retransmit */
        resend segment with sequence number y
    }
break;
```

前面讲过，当在如 TCP 这样一个实际协议中实现超时/重传机制时，会产生许多微妙的问题。上面的过程是在超过 20 年的 TCP 定时器使用经验的基础上演化而来的，读者应当理解实际情况确实是这样的。

4. 是回退 N 步还是选择重传

考虑下面这个问题来结束有关 TCP 差错恢复机制的学习：TCP 是一个 GBN 协议还是一个 SR 协议？前面讲过，TCP 确认是累积式的，正确接收但失序的报文段是不会被接收方逐个确认的。因此，如图 3-33 所示（也可参见图 3-19），TCP 发送方仅需维持已发送过但未被确认的字节的最小序号（SendBase）和下一个要发送的字节的序号（NextSeqNum）。在这种意义下，TCP 看起来更像一个 GBN 风格的协议。但是 TCP 和 GBN 协议之间有着一些显著的区别。许多 TCP 实现会将正确接收但失序的报文段缓存起来 [Stevens 1994]。另外考虑一下，当发

送方发送的一组报文段 1, 2, ..., N，并且所有的报文段都按序无差错地到达接收方时会发生的情况。进一步假设对分组  $n < N$  的确认报文丢失，但是其余  $N - 1$  个确认报文在分别超时以前到达发送端，这时又会发生的情况。在该例中，GBN 不仅会重传分组  $n$ ，还会重传所有后继的分组  $n + 1, n + 2, \dots, N$ 。在另一方面，TCP 将重传至多一个报文段，即报文段  $n$ 。此外，如果对报文段  $n + 1$  的确认报文在报文段  $n$  超时之前到达，TCP 甚至不会重传报文段  $n$ 。

对 TCP 提出的一种修改意见是所谓的选择确认（selective acknowledgment）[RFC 2018]，它允许 TCP 接收方有选择地确认失序报文段，而不是累积地确认最后一个正确接收的有序

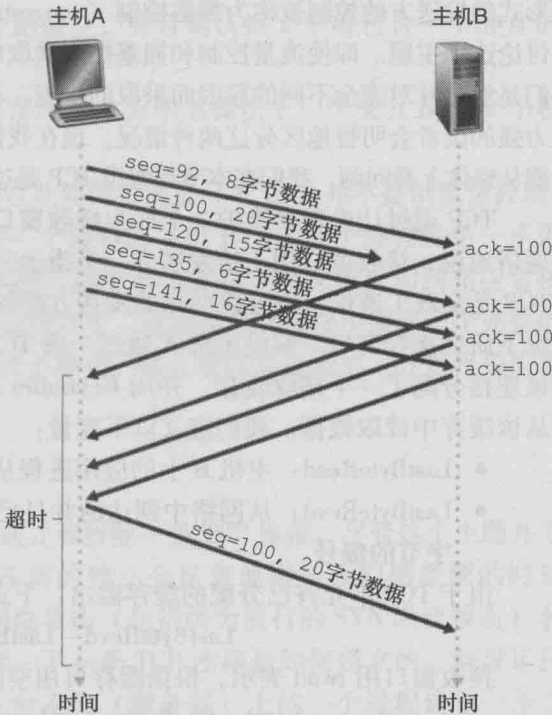


图 3-37 快速重传：在某报文段的定时器过期之前重传丢失的报文段