

SR 接收方将确认一个正确接收的分组而不管其是否按序。失序的分组将被缓存直到所有丢失分组（即序号更小的分组）皆被收到为止，这时才可以将一批分组按序交付给上层。图 3-25 详细列出了 SR 接收方所采用的各种动作。图 3-26 给出了一个例子以说明出现丢包时 SR 的操作。值得注意的是，在图 3-26 中接收方初始时缓存了分组 3、4、5，并在最终收到分组 2 时，才将它们一并交付给上层。

1. 序号在 $[rcv_base, rcv_base + N - 1]$ 内的分组被正确接收。在此情况下，收到的分组落在接收方的窗口内，一个选择 ACK 被回送给发送方。如果该分组以前没收到过，则缓存该分组。如果该分组的序号等于接收窗口的基序号（图 3-23 中的 rcv_base ），则该分组以及以前缓存的序号连续的（起始于 rcv_base 的）分组交付给上层。然后，接收窗口按向前移动分组的编号向上交付这些分组。举例来说，考虑一下图 3-26。当收到一个序号为 $rcv_base = 2$ 的分组时，该分组及分组 3、4、5 可被交付给上层。

2. 序号在 $[rcv_base - N, rcv_base - 1]$ 内的分组被正确收到。在此情况下，必须产生一个 ACK，即使该分组是接收方以前已确认过的分组。

3. 其他情况。忽略该分组。

图 3-25 SR 接收方的事件与动作

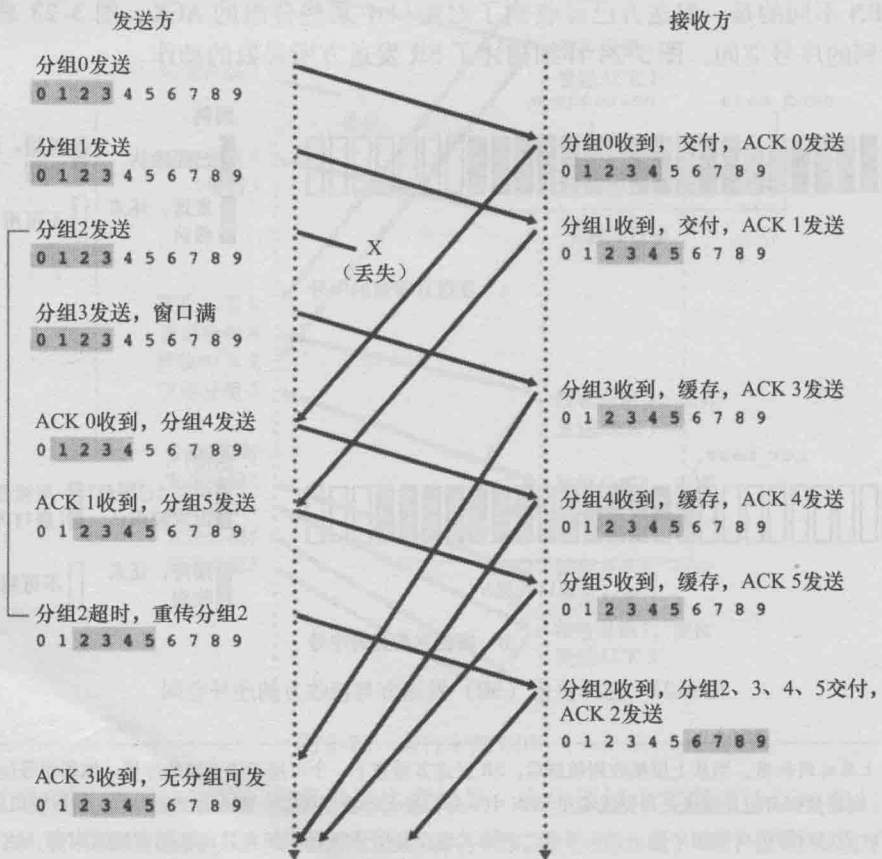


图 3-26 SR 操作

注意到图 3-25 中的第二步很重要，接收方重新确认（而不是忽略）已收到过的那些序号小于当前窗口基序号的分组。你应该理解这种重新确认确实是需要的。例如，给定在图 3-23 中所示的发送方和接收方的序号空间，如果分组 $send_base$ 的 ACK 没有从接收方传