

该做些什么。在 rdt 2.2 中已经研发的技术，如使用检验和、序号、ACK 分组和重传等，使我们能给出后一个问题的答案。为解决第一个关注的问题，还需增加一种新的协议机制。

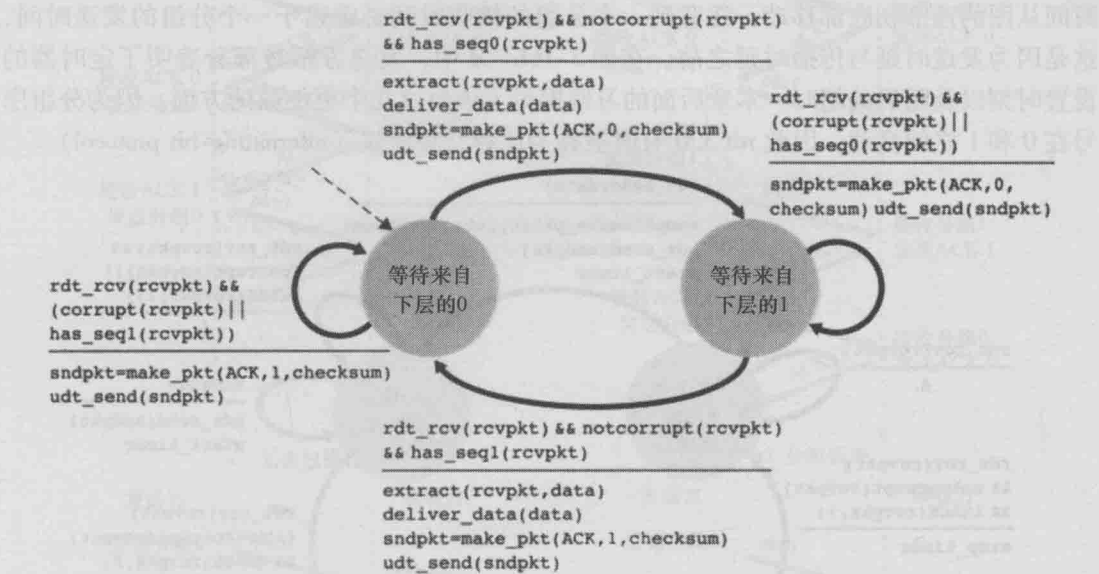


图 3-14 rdt 2.2 接收方

有很多可能的方法用于解决丢包问题（在本章结尾的习题中研究了几种其他方法）。这里，我们让发送方负责检测和恢复丢包工作。假定发送方传输一个数据分组，该分组或者接收方对该分组的 ACK 发生了丢失。在这两种情况下，发送方都收不到应当到来的接收方的响应。如果发送方愿意等待足够长的时间以便确定分组已丢失，则它只需重传该数据分组即可。你应该相信该协议确实有效。

但是发送方需要等待多久才能确定已丢失了某些东西呢？很明显发送方至少需要等待这样长的时间：即发送方与接收方之间的一个往返时延（可能会包括在中间路由器的缓冲时延）加上接收方处理一个分组所需的时间。在很多网络中，最坏情况下的最大时延是很难估算的，确定的因素非常少。此外，理想的协议应尽可能快地从丢包中恢复出来；等待一个最坏情况的时延可能意味着要等待一段较长的时间，直到启动差错恢复为止。因此实践中采取的方法是发送方明智地选择一个时间值，以判定可能发生了丢包（尽管不能确保）。如果在这个时间内没有收到 ACK，则重传该分组。注意到如果一个分组经历了一个特别大的时延，发送方可能会重传该分组，即使该数据分组及其 ACK 都没有丢失。这就在发送方到接收方的信道中引入了冗余数据分组（duplicate data packet）的可能性。幸运的是，rdt 2.2 协议已经有足够的功能（即序号）来处理冗余分组情况。

从发送方的观点来看，重传是一种万能灵药。发送方不知道是一个数据分组丢失，还是一个 ACK 丢失，或者只是该分组或 ACK 过度延时。在所有这些情况下，动作是同样的：重传。为了实现基于时间的重传机制，需要一个倒计时定时器（countdown timer），在一个给定的时间量过期后，可中断发送方。因此，发送方需要能做到：①每次发送一个分组（包括第一次分组和重传分组）时，便启动一个定时器。②响应定时器中断（采取适当的动作）。③终止定时器。

图 3-15 给出了 rdt 3.0 的发送方 FSM，这是一个在可能出错和丢包的信道上可靠传输