

们将一个 TCP 发送方的“丢包事件”定义为：要么出现超时，要么收到来自接收方的 3 个冗余 ACK。（回想我们在 3.5.4 节有关图 3-33 中的超时事件的讨论和收到 3 个冗余 ACK 后包括快速重传的后继修改。）当出现过度的拥塞时，在沿着这条路径上的一台（或多台）路由器的缓存会溢出，引起一个数据报（包含一个 TCP 报文段）被丢弃。丢弃的数据报接着会引起发送方的丢包事件（要么超时或收到 3 个冗余 ACK），发送方就认为在发送方到接收方的路径上出现了拥塞的指示。

考虑了拥塞检测问题后，我们接下来考虑网络没有拥塞这种更为乐观的情况，即没有出现丢包事件的情况。在此情况下，在 TCP 的发送方将收到对于以前未确认报文段的确认。如我们将看到的那样，TCP 将这些确认的到达作为一切正常的指示，即在网络上传输的报文段正被成功地交付给目的地，并使用确认来增加窗口的长度（及其传输速率）。注意到如果确认以相当慢的速率到达（例如，如果该端到端路径具有高时延或包含一段低带宽链路），则该拥塞窗口将以相当慢的速率增加。在另一方面，如果确认以高速率到达，则该拥塞窗口将会更为迅速地增大。因为 TCP 使用确认来触发（或计时）增大它的拥塞窗口长度，TCP 被说成是自计时（self-clocking）的。

给定调节 cwnd 值以控制发送速率的机制，关键的问题依然存在：TCP 发送方怎样确定它应当发送的速率呢？如果众多 TCP 发送方总体上发送太快，它们能够拥塞网络，导致我们在图 3-48 中看到的拥塞崩溃。事实上，为了应对在较早 TCP 版本下观察到的因特网拥塞崩溃 [Jacobson 1988]，研发了该版本的 TCP（我们马上将学习它）。然而，如果 TCP 发送方过于谨慎，发送太慢，它们不能充分利用网络的带宽；这就是说，TCP 发送方能够以更高的速率发送而不会使网络拥塞。那么 TCP 发送方如何确定它们的发送速率，既使得网络不会拥塞，与此同时又能充分利用所有可用的带宽？TCP 发送方是显式地协作，或存在一种分布式方法使 TCP 发送方能够仅基于本地信息设置它们的发送速率？TCP 使用下列指导性原则回答这些问题：

- 一个丢失的报文段意味着拥塞，因此当丢失报文段时应当降低 TCP 发送方的速率。回想在 3.5.4 节中的讨论，对于给定报文段，一个超时事件或四个确认（一个初始 ACK 和其后的三个冗余 ACK）被解释为跟随该四个 ACK 的报文段的“丢包事件”的一种隐含的指示。从拥塞控制的观点看，该问题是 TCP 发送方应当如何减小它的拥塞窗口长度，即减小其发送速率，以应对这种推测的丢包事件。
- 一个确认报文段指示该网络正在向接收方交付发送方的报文段，因此，当对先前未确认报文段的确认到达时，能够增加发送方的速率。确认的到达被认为是一切顺利的隐含指示，即报文段正从发送方成功地交付给接收方，因此该网络不拥塞。拥塞窗口长度因此能够增加。
- 带宽探测。给定 ACK 指示源到目的地路径无拥塞，而丢包事件指示路径拥塞，TCP 调节其传输速率的策略是增加其速率以响应到达的 ACK，除非出现丢包事件，此时才减小传输速率。因此，为探测拥塞开始出现的速率，TCP 发送方增加它的传输速率，从该速率后退，进而再次开始探测，看看拥塞开始速率是否发生了变化。TCP 发送方的行为也许类似于要求（并得到）越来越多糖果的孩子，直到最后告知他/她“不行！”，孩子后退一点，然后过一会儿再次开始提出请求。注意到网络中没有明确的拥塞状态信令，即 ACK 和丢包事件充当了隐式信号，并且每个 TCP 发送方根据异步于其他 TCP 发送方的本地信息而行动。