机可以在轻微拥塞时将经过的 RM 信元中的 NI 比特置为 1，在严重拥塞时，把 CI 比特置为 1。当目的主机收到一个 RM 信元时，它将把该 RM 信元发回给发送方，而保持 CI 与 NI 比特不变（除了 CI 比特也许会因为上面描述的 EFCI 机制而由目的端置为 1 之外）。

- ER 的设置。每一个 RM 信元还包含一个两字节的**显式速率**（Explicit Rate，ER）**字段**。一个拥塞的交换机也许会降低经过的 RM 信元中 ER 字段所包含的值。以这种方式，ER 字段将被设置为在源至目的地的路径上的所有交换机中的最小可支持速率。

一个 ATM ABR 源以返回的 RM 信元中的 CI、NI 及 ER 值为函数，来调整其发送信元的速率。进行速率调整的规则非常复杂而且繁琐，感兴趣的读者可以参考［Jain 1996］以得到详细信息。

3.7 TCP 拥塞控制

在本节中，我们再次来学习 TCP。如我们在 3.5 节所见，TCP 为运行在不同主机上的两个进程之间提供了可靠传输服务。TCP 的另一个关键部分就是其拥塞控制机制。如在前一节所指出，TCP 必须使用端到端拥塞控制而不是使网络辅助的拥塞控制，因为 IP 层不向端系统提供显式的网络拥塞反馈。

TCP 所采用的方法是让每一个发送方根据所感知到的网络拥塞程度来限制其能向连接发送流量的速率。如果一个 TCP 发送方感知从它到目的地之间的路径上没什么拥塞，则 TCP 发送方增加其发送速率；如果发送方感知沿着该路径有拥塞，则发送方就会降低其发送速率。但是这种方法提出了三个问题。第一，一个 TCP 发送方如何限制它向其连接发送流量的速率呢？第二，一个 TCP 发送方如何感知从它到目的地之间的路径上存在拥塞呢？第三，当发送方感知到端到端的拥塞时，采用何种算法来改变其发送速率呢？

我们首先分析一下 TCP 发送方是如何限制向其连接发送流量的。在 3.5 节中我们看到，TCP 连接的每一端都是由一个接收缓存、一个发送缓存和几个变量（LastByteRead、rwnd 等）组成。运行在发送方的 TCP 拥塞控制机制跟踪一个额外的变量，即**拥塞窗口**（congestion window）。拥塞窗口表示为 cwnd，它对一个 TCP 发送方能向网络中发送流量的速率进行了限制。特别是，在一个发送方中未被确认的数据量不会超过 cwnd 与 rwnd 中的最小值，即

$$\text{LastByteSent} - \text{LastByteAcked} \leqslant \min\{\text{cwnd}, \text{rwnd}\}$$

为了关注拥塞控制（与流量控制形成对比），我们后面假设 TCP 接收缓存足够大，以至可以忽略接收窗口的限制；因此在发送方中未被确认的数据量仅受限于 cwnd。我们还假设发送方总是有数据要发送，即在拥塞窗口中的所有报文段要被发送。

上面的约束限制了发送方中未被确认的数据量，因此间接地限制了发送方的发送速率。为了理解这一点，我们来考虑一个丢包和发送时延均可以忽略不计的连接。因此粗略地讲，在每个往返时间（RTT）的起始点，上面的限制条件允许发送方向该连接发送 cwnd 个字节的数据，在该 RTT 结束时发送方接收对数据的确认报文。因此，*该发送方的发送速率大概是 cwnd/RTT 字节/秒。通过调节 cwnd 的值，发送方因此能调整它向连接发送数据的速率。*

我们接下来考虑 TCP 发送方是如何感知在它与目的地之间的路径上出现了拥塞的。我