图 3-56 描绘了两条 TCP 连接实现的吞吐量情况。如果 TCP 要在这两条 TCP 连接之间平等地共享链路带宽，那么实现的吞吐量曲线应当是从原点沿 45°方向的箭头向外辐射（平等带宽共享）。理想情况是，两个吞吐量的和应等于 R。（当然，每条连接得到相同但容量为 0 的共享链路容量并非我们所期望的情况！）所以我们的目标应该是使取得的吞吐量落在图 3-56 中平等带宽共享曲线与全带宽利用曲线的交叉点附近的某处。

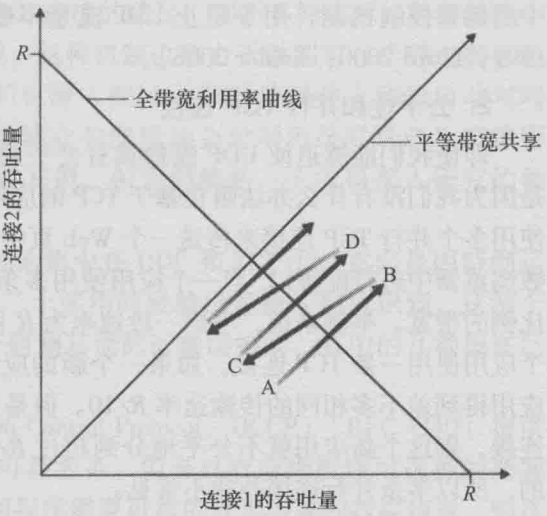

图 3-56 TCP 连接 1 和连接 2 实现的吞吐量

假定 TCP 窗口长度是这样的，即在某给定时刻，连接 1 和连接 2 实现了由图 3-56 中 A 点所指明的吞吐量。因为这两条连接共同消耗的链路带宽量小于 R，所以无丢包事件发生，根据 TCP 的拥塞避免算法的结果，这两条连接每过一个 RTT 都要将其窗口增加 1 个 MSS。因此，这两条连接的总吞吐量就会从 A 点开始沿 45°线前行（两条连接都有相同的增长）。最终，这两条连接共同消耗的带宽将超过 R，最终将发生分组丢失。假设连接 1 和连接 2 实现 B 点指明的吞吐量时，它们都经历了分组丢失。连接 1 和连接 2 于是就按二分之一减小其窗口。所产生的结果实现了 C 点指明的吞吐量，它正好位于始于 B 点止于原点的一个向量的中间。因为在 C 点，共同消耗的带宽小于 R，所以这两条连接再次沿着始于 C 点的 45°线增加其吞吐量。最终，再次发生丢包事件，如在 D 点，这两条连接再次将其窗口长度减半，如此等等。你应当搞清楚这两条连接实现的带宽最终将沿着平等带宽共享曲线在波动。还应该搞清楚无论这两条连接位于二维空间的何处，它们最终都会收敛到该状态！虽然此时我们做了许多理想化的假设，但是它仍然能对解释为什么 TCP 会导致在多条连接之间的平等共享带宽这个问题提供一个直观的感觉。

在理想化情形中，我们假设仅有 TCP 连接穿过瓶颈链路，所有的连接具有相同的 RTT 值，且对于一个主机 – 目的地对而言只有一条 TCP 连接与之相关联。实践中，这些条件通常是得不到满足的，客户 – 服务器应用因此能获得非常不平等的链路带宽份额。特别是，已经表明当多条连接共享一个共同的瓶颈链路时，那些具有较小 RTT 的连接能够在链路空闲时更快地抢到可用带宽（即较快地打开其拥塞窗口），因而将比那些具有较大 RTT 的连接享用更高的吞吐量［Laksman 1997］。

1. 公平性和 UDP

我们刚才已经看到，TCP 拥塞控制是如何通过拥塞窗口机制来调节一个应用程序的传输速率的。许多多媒体应用如因特网电话和视频会议，经常就因为这种特定原因而不在 TCP 上运行，因为它们不想其传输速率被扼制，即使在网络非常拥塞的情况下。相反，这些应用宁可在 UDP 上运行，UDP 是没有内置的拥塞控制的。当运行在 UDP 上时，这些应用能够以恒定的速率将其音频和视频数据注入网络之中并且偶尔会丢失分组，而不愿在拥塞时将其发送速率降至“公平”级别并且不丢失任何分组。从 TCP 的观点来看，运行在 UDP 上的多媒体应用是不公平的，因为它们不与其他连接合作，也不适时地调整其传输速