

TCP 继续演化的需求能够通过考虑网格和云计算应用所需要的高速 TCP 连接加以阐述。例如, 考虑一条具有 1500 字节报文段和 100ms RTT 的 TCP 连接, 假定我们要通过这条连接以 10Gbps 速率发送数据。根据 [RFC 3649], 我们注意到使用上述 TCP 吞吐量公式, 为了取得 10Gbps 吞吐量, 平均拥塞窗口长度将需要是 83 333 个报文段。对如此大量的报文段, 使我们相当关注这 83 333 个传输中的报文段也许会丢失。在丢失的情况下, 将会出现什么情况呢? 或者以另一种方式说, 这些传输的报文段能以何种比例丢失, 使得在图 3-52 中列出的 TCP 拥塞控制算法仍能取得所希望的 10Gbps 速率? 在本章的课后习题中, 要求读者推导出一条 TCP 连接的吞吐量公式, 该公式作为丢包率 (L)、往返时间 (RTT) 和最大报文段长度 (MSS) 的函数:

$$\text{一条连接的平均吞吐量} = \frac{1.22 \times \text{MSS}}{\text{RTT} \sqrt{L}}$$

使用该公式, 我们能够看到, 为了取得 10Gbps 的吞吐量, 今天的 TCP 拥塞控制算法仅能容忍 2×10^{-10} 的报文段丢失概率 (或等价地说, 对每 5 000 000 000 个报文段有一个丢包), 这是一个非常低的值。这种观察导致许多研究人员为这种高速环境特别设计新版 TCP, 对这些努力的讨论参见 [Jin 2004, RFC 3649, Kelly 2003, Ha 2008]。

公平性

考虑 K 条 TCP 连接, 每条都有不同的端到端路径, 但是都经过一段传输速率为 R bps 的瓶颈链路。(所谓瓶颈链路, 是指对于每条连接, 沿着该连接路径上的所有其他段链路都不拥塞, 而且与该瓶颈链路的传输容量相比, 它们都有充足的传输容量。) 假设每条连接都在传输一个大文件, 而且无 UDP 流量通过该段瓶颈链路。如果每条连接的平均传输速率接近 R/K , 即每条连接都得到相同份额的链路带宽, 则认为该拥塞控制机制是公平的。

TCP 的 AIMD 算法公平吗? 尤其是假定可在不同时间启动并因此在某个给定的时间点可能具有不同的窗口长度情况下, 对这些不同的 TCP 连接还是公平的吗? TCP 趋于在竞争的多条 TCP 连接之间提供对一段瓶颈链路带宽的平等分享, 其理由 [Chiu 1989] 给出了一个极好的、直观的解释。

我们考虑有两条 TCP 连接共享一段传输速率为 R 的链路的简单例子, 如图 3-55 中所示。我们将假设这两条连接有相同的 MSS 和 RTT (这样如果它们有相同的拥塞窗口长度, 就会有相同的吞吐量), 它们有大量的数据要发送, 且没有其他 TCP 连接或 UDP 数据报穿越该段共享链路。我们还将忽略 TCP 的慢启动阶段, 并假设 TCP 连接一直按 CA 模式 (AIMD) 运行。

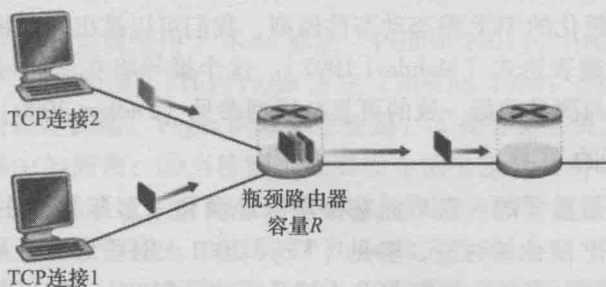


图 3-55 两条 TCP 连接共享同一条瓶颈链路