

率。因为 TCP 拥塞控制在面临拥塞增加（丢包）时，将降低其传输速率，而 UDP 源则不必这样做，UDP 源有可能压制 TCP 流量。当今的一个主要研究领域就是开发一种因特网中的拥塞控制机制，用于阻止 UDP 流量不断压制直至中断因特网吞吐量的情况 [Floyd 1999; Floyd 2000; Kohler 2006]。

2. 公平性和并行 TCP 连接

即使我们能够迫使 UDP 流量具有公平的行为，但公平性问题仍然没有完全解决。这是因为我们没有什么办法阻止基于 TCP 的应用使用多个并行连接。例如，Web 浏览器通常使用多个并行 TCP 连接来传送一个 Web 页中的多个对象。（多条连接的确切数目可以在多数浏览器中进行配置。）当一个应用使用多条并行连接时，它占用了一条拥塞链路中较大比例的带宽。举例来说，考虑一段速率为 R 且支持 9 个在线客户-服务器应用的链路，每个应用使用一条 TCP 连接。如果一个新的应用加入进来，也使用一条 TCP 连接，则每个应用得到差不多相同的传输速率 $R/10$ 。但是如果这个新的应用这次使用了 11 个并行 TCP 连接，则这个新应用就不公平地分到超过 $R/2$ 的带宽。Web 流量在因特网中是非常普遍的，所以多条并行连接并非不常见。

3.8 小结

本章我们首先学习了运输层协议能够向网络应用程序提供的服务。在一个极端，运输层协议非常简单，并向应用程序不提供不必要的服务，而仅向通信进程提供多路复用/分解的功能。因特网中的 UDP 协议就是这样一种不提供不必要服务的运输层协议。在另一个极端，运输层协议能够向应用程序提供各种各样的保证，例如数据的可靠交付、时延保证和带宽保证。无论如何，运输层协议能够提供的服务经常受下面网络层协议服务模型的限制。如果网络层协议不能向运输层报文段提供时延或带宽保证，那么运输层协议就不能向进程间发送的报文提供时延或带宽保证。

在 3.4 节中，我们学习了运输层协议能够提供可靠数据传输，即使下面的网络层是不可靠的。我们看到了提供可靠的数据传送会遇到许多微妙的问题，但都可以通过精心地结合确认、定时器、重传以及序号机制来完成任务。

尽管在本章中我们包含了可靠数据传送，但是我们应该理解在链路层、网络层、运输层或应用层协议中都可以提供可靠数据传送。该协议栈中上面 4 层的任意一层都可以实现确认、定时器、重传以及序号，能够向其上层提供可靠数据传送。事实上，在过去数年中，工程师以及计算机科学家们已经独立地设计并实现了提供可靠数据传送的链路层、网络层、运输层以及应用层协议（虽然这些协议中的许多已经销声匿迹了）。

在 3.5 节中，我们详细地研究了 TCP 协议，它是因特网中面向连接和可靠的运输层协议。我们知道 TCP 是非常复杂的，它涉及了连接管理、流量控制、往返时间估计以及可靠数据传送。事实上，TCP 比我们描述的要更为复杂，即我们有意地避而不谈在各种 TCP 实现版本中广泛实现的各种 TCP 补丁、修复和改进。然而，所有这些复杂性都对网络层应用隐藏了起来。如果某主机上的客户希望向另一台主机上的服务器可靠地发送数据，它只需要打开对该服务器的一个 TCP 套接字，然后将数据注入该套接字。客户-服务器应用程序则乐于对 TCP 的复杂性视而不见。

在 3.6 节中，我们从广泛的角度研究了拥塞控制，在 3.7 节中我们阐述了 TCP 是如何