实现拥塞控制的。我们知道了拥塞控制对于网络良好运行是必不可少的。没有拥塞控制，网络很容易出现死锁，使得端到端之间很少或没有数据能被传输。在3.7节中我们学习了TCP实现的一种端到端拥塞控制机制，即当TCP连接的路径上判断不拥塞时，其传输速率就加性增；当出现丢包时，传输速率就乘性减。这种机制也致力于做到每一个通过拥塞链路的TCP连接能平等地共享该链路带宽。我们也深入探讨了TCP连接建立和慢启动对时延的影响。我们观察到在许多重要场合，连接建立和慢启动会对端到端时延产生严重影响。我们再次强调，尽管TCP在这几年一直在发展，但它仍然是一个值得深入研究的领域，并且在未来的几年中还可能持续演化。

在本章中我们对特定因特网运输协议的讨论集中在UDP和TCP上，它们是因特网运输层的两匹“驮马”。然而，对这两个协议的二十多年的经验已经使人们认识到，这两个协议都不是完美无缺的。研究人员因此在忙于研制其他的运输层协议，其中的几种现在已经成为IETF建议的标准。

数据报拥塞控制协议（Datagram Congestion Control Protocol，DCCP）［RFC 4340］提供了一种低开销、面向报文、类似于UDP的不可靠服务，但是具有应用程序可选择的拥塞控制形式，该机制与TCP相兼容。如果某应用程序需要可靠的或半可靠的数据传送，则这将在应用程序自身中执行（也许使用我们已经在3.4节中学过的机制）。DCCP被设想用于诸如流媒体（参见第7章）等应用程序中，DCCP能够利用数据交付的预定时间和可靠性之间的折中，但是要对网络拥塞作出响应。

流控制传输协议（Stream Control Transmission Protocol，SCTP）［RFC 4960，RFC 3286］是一种可靠的、面向报文的协议，该协议允许几个不同的应用层次的“流”复用到单个SCTP连接上（一种称之为“多流”的方法）。从可靠性的角度看，在该连接中的不同流被分别处理，因此在一条流中的分组丢失不会影响其他流中数据的交付。当一台主机与两个或更多个网络连接时，SCTP也允许数据经两条出路径传输，还具有失序数据的选项交付和一些其他特色。SCTP的流控制和拥塞控制算法基本上与TCP中的相同。

TCP友好速率控制（TCP-Friendly Rate Control，TFRC）协议［RFC 5348］是一种拥塞控制协议而不是一种功能齐全的运输层协议。它定义了一种拥塞控制机制，该机制能被用于诸如DCCP等其他运输协议（事实上在DCCP中可供使用的两种应用程序可选的协议之一就是TFRC）。TFRC的目标是平滑在TCP拥塞控制中的“锯齿”行为（参见图3-54），同时维护一种长期的发送速率，该速率“合理地”接近TCP的速率。使用比TCP更为平滑的发送速率，TFRC非常适合诸如IP电话或流媒体等多媒体应用，这种平滑的速率对于这些应用是重要的。TFRC是一种“基于方程”的协议，这些协议使用测得的丢包率作为方程的输入［Padhye 2000］，即使用方程估计一个TCP会话在该丢包率下TCP的吞吐量将是多大。该速率则被取为TFRC的目标发送速率。

唯有未来才能告诉我们DCCP、SCTP或TFRC是否能得到广泛实施。虽然这些协议明确地提供了超过TCP和UDP的强化能力，但是多年来已经证明了TCP和UDP自身是“足够好”的。是否“更好”将胜出“足够好”，这将取决于技术、社会和商业考虑的复杂组合。

在第1章中，我们讲到计算机网络能被划分成“网络边缘”和“网络核心”。网络边缘包含了在端系统中发生的所有事情。既然已经覆盖了应用层和运输层，我们关于网络边缘的讨论也就结束了。接下来是探寻网络核心的时候了！我们的旅程从下一章开始，下一章将学习网络层，并且将在第5章继续学习链路层。