

的、拥塞受控的数据传输难以实现。此外，如我们前面所述，DNS 运行在 UDP 之上，从而避免了 TCP 的连接创建时延。

应用	应用层协议	下面的运输协议
电子邮件	SMTP	TCP
远程终端访问	Telnet	TCP
Web	HTTP	TCP
文件传输	FTP	TCP
远程文件服务器	NFS	通常 UDP
流式多媒体	通常专用	UDP 或 TCP
因特网电话	通常专用	UDP 或 TCP
网络管理	SNMP	通常 UDP
路由选择协议	RIP	通常 UDP
名字转换	DNS	通常 UDP

图 3-6 流行的因特网应用及其下面的运输协议

如图 3-6 所示，UDP 和 TCP 现在都用于多媒体应用，如因特网电话、实时视频会议、流式存储音频与视频。我们将在第 7 章仔细学习这些应用。我们刚说过，既然所有这些应用都能容忍少量的分组丢失，因此可靠数据传输对于这些应用的成功并不是至关重要的。此外，TCP 的拥塞控制会导致如因特网电话、视频会议之类的实时应用性能变得很差。由于这些原因，多媒体应用开发人员通常将这些应用运行在 UDP 之上而不是 TCP 之上。然而，TCP 被越来越多地应用于流式多媒体传输中。例如，[Sripanidkulchai 2004] 发现大约 75% 的按需和实况流式多媒体使用了 TCP。当分组丢包率低时，并且为了安全原因，某些机构阻塞 UDP 流量（参见第 8 章），对于流式媒体传输来说，TCP 变得越来越有吸引力了。

虽然目前通常这样做，但在 UDP 之上运行多媒体应用是有争议的。如我们前面所述，UDP 没有拥塞控制。但是，需要拥塞控制来预防网络进入一种拥塞状态，此时只能做很少的有用工作。如果每个人都启动流式高比特率视频而不使用任何拥塞控制的话，就会使路由器中有大量的分组溢出，以至于几乎没有 UDP 分组能成功地通过源到目的的路径传输。况且，由无控制的 UDP 发送方引入的高丢包率将引起 TCP 发送方（如我们将看到的那样，TCP 遇到拥塞将减小它们的发送速率）大大地减小它们的速率。因此，UDP 中缺乏拥塞控制能够导致 UDP 发送方和接收方之间的高丢包率，并挤垮了 TCP 会话，这是一个潜在的严重问题 [Floyd 1999]。很多研究人员已提出了一些新机制，以促进所有的数据源（包括 UDP 源）执行自适应的拥塞控制 [Mahdavi 1997; Floyd 2000; Kohler 2006; RFC 4340]。

在讨论 UDP 报文段结构之前，我们要提一下，使用 UDP 的应用是可以实现可靠数据传输的。这可通过在应用程序自身中建立可靠性机制来完成（例如，可通过增加确认与重传机制来实现，如采用我们将在下一节学习的一些机制）。但这并不是无足轻重的任务，它会使应用开发人员长时间地忙于调试。无论如何，将可靠性直接构建于应用程序中可以使其“左右逢源”。也就是说应用进程可以进行可靠通信，而无需受制于由 TCP 拥塞控制机制而无需受制于传输速率限制。