

进行一次查询时，它构造了一个 DNS 查询报文并将其交给 UDP。无须执行任何与运行在目的端系统中的 UDP 实体之间的握手，主机端的 UDP 为此报文添加首部字段，然后将形成的报文段交给网络层。网络层将此 UDP 报文段封装进一个 IP 数据报中，然后将其发送给一个名字服务器。在查询主机中的 DNS 应用程序则等待对该查询的响应。如果它没有收到响应（可能是由于底层网络丢失了查询或响应），则要么试图向另一个名字服务器发送该查询，要么通知调用的应用程序它不能获得响应。

现在你也许想知道，为什么应用开发人员宁愿在 UDP 之上构建应用，而不选择在 TCP 上构建应用？既然 TCP 提供了可靠数据传输服务，而 UDP 不能提供，那么 TCP 是否总是首选的呢？答案是否定的，因为有许多应用更适合用 UDP，原因主要以下几点：

- 关于何时、发送什么数据的应用层控制更为精细。采用 UDP 时，只要应用进程将数据传递给 UDP，UDP 就会将此数据打包进 UDP 报文段并立即将其传递给网络层。在另一方面，TCP 有一个拥塞控制机制，以便当源和目的主机间的一条或多条链路变得极度拥塞时来遏制运输层 TCP 发送方。TCP 仍将继续重新发送数据报文段直到目的主机收到此报文并加以确认，而不管可靠交付需要用多长时间。因为实时应用通常要求最小的发送速率，不希望过分地延迟报文段的传送，且能容忍一些数据丢失，TCP 服务模型并不是特别适合这些应用的需要。如后面所讨论的，这些应用可以使用 UDP，并作为应用的一部分来实现所需的、超出 UDP 的不提供不必要的报文段交付服务之外的额外功能。
- 无需连接建立。如我们后面所讨论的，TCP 在开始数据传输之前要经过三次握手。UDP 却不需要任何准备即可进行数据传输。因此 UDP 不会引入建立连接的时延。这可能是 DNS 运行在 UDP 之上而不是运行在 TCP 之上的主要原因（如果运行在 TCP 上，则 DNS 会慢得多）。HTTP 使用 TCP 而不是 UDP，因为对于具有文本数据的 Web 网页来说，可靠性是至关重要的。但是，如我们在 2.2 节中简要讨论的那样，HTTP 中的 TCP 连接建立时延对于与下载 Web 文档相关的时延来说是一个重要因素。
- 无连接状态。TCP 需要在端系统中维护连接状态。此连接状态包括接收和发送缓存、拥塞控制参数以及序号与确认号的参数。我们将在 3.5 节看到，要实现 TCP 的可靠数据传输服务并提供拥塞控制，这些状态信息是必要的。在另一方面，UDP 不维护连接状态，也不跟踪这些参数。因此，某些专门用于某种特定应用的服务器当应用程序运行在 UDP 之上而不是运行在 TCP 上时，一般都能支持更多的活跃客户。
- 分组首部开销小。每个 TCP 报文段都有 20 字节的首部开销，而 UDP 仅有 8 字节的开销。

图 3-6 列出了流行的因特网应用及其所使用的运输协议。如我们所期望的那样，电子邮件、远程终端访问、Web 及文件传输都运行在 TCP 之上。因为所有这些应用都需要 TCP 的可靠数据传输服务。无论如何，有很多重要的应用是运行在 UDP 上而不是 TCP 上。UDP 被用于 RIP 路由选择表的更新（参见 4.6.1 节）。因为这些更新被周期性地发送（通常每 5 分钟一次），更新的丢失能被最近的更新所替代，因此丢包、过期的更新是无用的。UDP 也用于承载网络管理数据（SNMP，参见第 9 章）。在这种场合下，UDP 要优于 TCP，因为网络管理应用程序通常必须在该网络处于重压状态时运行，而正是在这个时候可靠