

样人们就能够创造协同工作的系统和产品。这正是标准发挥作用的地方。因特网标准 (Internet standard) 由因特网工程任务组 (Internet Engineering Task Force, IETF) [IETF 2012] 研发。IETF 的标准文档称为请求评论 (Request For Comment, RFC)。RFC 最初是作为普通的请求评论 (因此而得名), 以解决因特网先驱者们面临的网络和协议问题 [Allman 2011]。RFC 文档往往是技术性很强并相当详细的。它们定义了 TCP、IP、HTTP (用于 Web) 和 SMTP (用于电子邮件) 等协议。目前已经有将近 6000 个 RFC。其他组织也在制定用于网络组件的标准, 最引人注目的是针对网络链路的标准。例如, IEEE 802 LAN/MAN 标准化委员会 [IEEE 802 202] 制定了以太网和无线 WiFi 的标准。

1.1.2 服务描述

前面的讨论已经辨识了构成因特网的许多部件。但是我们也能从一个完全不同的角度, 即从为应用程序提供服务的基础设施的角度来描述因特网。这些应用程序包括电子邮件、Web 冲浪、即时讯息、社交网络、IP 语音 (VoIP)、流式视频、分布式游戏、对等 (peer-to-peer, P2P) 文件共享、因特网电视、远程注册等等。这些应用程序称为**分布式应用程序** (distributed application), 因为它们涉及多台相互交换数据的端系统。重要的是, 因特网应用程序运行在端系统上, 即它们并不运行在网络核心中的分组交换机中。尽管分组交换机促进端系统之间的数据交换, 但它们并不关心作为数据的源或宿的应用程序。

我们稍深入地探讨一下为应用程序提供服务的基础设施的含义。为此, 假定你对分布式因特网应用程序有一个激动人心的新思想, 它可能大大地造福于人类, 或者它可能直接使你富有和出名。你将如何将这种思想转换成为一种实际的因特网应用程序呢? 因为应用程序运行在端系统上, 所以你将需要编写运行在端系统上的一些软件。例如, 你可能用 Java、C 或 Python 编写软件。此时, 因为你在研发一种分布式因特网应用程序, 运行在不同端系统上的软件将需要互相发送数据。在这里我们碰到一个核心问题, 这导致了另一种描述因特网的方法, 即将因特网描述为应用程序的平台。运行在一个端系统上的应用程序怎样才能指令因特网向运行在另一个端系统上的软件发送数据呢?

与因特网相连的端系统提供了一个**应用程序编程接口** (Application Programming Interface, API), 该 API 规定了运行在一个端系统上的软件请求因特网基础设施向运行在另一个端系统上的特定目的地软件交付数据的方式。因特网 API 是一套发送软件必须遵循的规则集合, 因此因特网能够将数据交付给目的地。我们将在第 2 章详细讨论因特网 API。此时, 我们做一个简单的类比, 在本书中我们将经常使用这个类比。假定 Alice 使用邮政服务向 Bob 发一封信。当然, Alice 不能只是写了这封信 (相关数据) 然后把该信丢出窗外。相反, 邮政服务要求 Alice 将信放入一个信封中; 在信封的中央写上 Bob 的全名、地址和邮政编码; 封上信封; 在信封的右上角贴上邮票; 最后将该信封丢进一个邮局的邮政服务邮箱中。因此, 该邮政服务有自己的“邮政服务 API”或一套规则, 这是 Alice 必须遵循的, 这样邮政服务才能将自己的信件交付给 Bob。同理, 因特网也有一个发送数据的程序必须遵循的 API, 使因特网向接收数据的程序交付数据。

当然, 邮政服务向顾客提供了多种服务, 如特快专递、挂号、普通服务等。同样的, 因特网向应用程序提供了多种服务。当你研发一种因特网应用程序时, 也必须为你的应用程序选择其中的一种因特网服务。我们将在第 2 章中描述因特网服务。