

年1月1日见证了 TCP/IP 作为 ARPAnet 新的标准主机协议的正式部署，替代了 NCP 协议。从 NCP 到 TCP/IP 的迁移 [RFC 801] 是一个标志性事件，所有主机被要求在那天转移到 TCP/IP 上去。在 20 世纪 80 年代后期，TCP 进行了重要扩展，以实现基于主机的拥塞控制 [Jacobson 1988]。还研制出了 DNS（域名系统），用于将人可读的因特网名字（例如 gaia. cs. umass. edu）映射到它的 32 比特 IP 地址 [RFC 1034]。

在 20 世纪 80 年代初期，在 ARPAnet（这绝大多数是美国的成果）发展的同时，法国启动了 Minitel 项目，这个雄心勃勃的计划是让数据网络进入每个家庭。在法国政府的支持下，Minitel 系统由公共分组交换网络（基于 X.25 协议集）、Minitel 服务器和具有内置低速调制解调器的廉价终端组成。Minitel 于 1984 年取得了巨大的成功，当时法国政府向每个需要的住户免费分发一个 Minitel 终端。Minitel 站点包括免费站点（如电话目录站点）以及一些专用站点，这些专用站点根据每个用户的使用来收取费用。在 20 世纪 90 年代中期的鼎盛时期，Minitel 提供了 20 000 多种服务，涵盖从家庭银行到特殊研究数据库的广泛范围。Minitel 在大量法国家庭中存在于 10 年后，大多数美国人才听说因特网。

1.7.4 因特网爆炸：20 世纪 90 年代

20 世纪 90 年代出现了许多事件，这些事件标志着因特网持续革命和很快到来的商业化。作为因特网祖先的 ARPAnet 已不复存在。1991 年，NSFNET 解除了对 NSFNET 用于商业目的的限制。NSFNET 自身于 1995 年退役，这时因特网主干流量则由商业因特网服务提供商负责承载。

然而，20 世纪 90 年代的主要事件是万维网（World Wide Web）应用程序的出现，它将因特网带入世界上数以百万计的家庭和商业中。Web 作为一个平台，也引入和配置了数百个新的应用程序，其中包括搜索（如谷歌和 Bing）、因特网商务（如亚马逊和 eBay）以及社交网络（如脸书），对这些应用程序我们今天已经习以为常了。

Web 是由 Tim Berners-Lee 于 1989 ~ 1991 年间在 CERN 发明的 [Berners-Lee 1989]，最初的想法源于 20 世纪 40 年代 Vannevar Bush [Bush 1945] 和 20 世纪 60 年代以来 Ted Nelson [Xanadu 2012] 在超文本方面的早期工作。Berners-Lee 和他的同事研制了 HTML、HTTP、Web 服务器和浏览器的初始版本，这是 Web 的 4 个关键部分。到了 1993 年年底前后，大约有 200 台 Web 服务器在运行，而这些只是正在出现的 Web 服务器的冰山一角。就在这个时候，几个研究人员研制了具有 GUI 接口的 Web 浏览器，其中的 Marc Andreessen 和 Jim Clark 一起创办了 Mosaic Communications 公司，该公司就是后来的 Netscape 通信公司 [Cusmano 1998; Quittner 1998]。到了 1995 年，大学生们每天都在使用 Netscape 浏览器在 Web 上冲浪。大约在这段时间，大大小小的公司都开始运行 Web 服务器，并在 Web 上处理商务。1996 年，微软公司开始开发浏览器，这导致了 Netscape 和微软之间的浏览器之战，并以微软公司在几年后获胜而告终 [Cusumano 1998]。

20 世纪 90 年代的后 5 年，随着主流公司和数以千计的初创公司创造了大量因特网产品和服务，因特网到了飞速增长和创新的时期。到了 2000 年末，因特网已经支持数百流行的应用程序，包括以下 4 种备受欢迎的应用程序：

- 电子邮件，包括附件和 Web 可访问的电子邮件。
- Web，包括 Web 浏览和因特网商务。
- 即时讯息（instant messaging），具有联系人列表。
- MP3 的对等（peer-to-peer）文件共享，由 Napster 开创。

值得一提的是，前两个应用程序出自专业研究机构，而后两个却由一些年轻创业者所发明。

1995 ~ 2001 年，这段时间也是因特网在金融市场上急转突变的时期。在成为有利可图的公司之前，数以百计的因特网初创公司靠首次公开募股（IPO）并在股票市场上交易起家。许多公司身价数十亿美元，却没有任何主要的收入渠道。因特网的股票在 2000 ~ 2001 年崩盘，导致许多初创公司倒闭。不过，也有许多公司成为因特网世界的大赢家，包括微软、思科、雅虎、e-Bay、谷歌和亚马逊。

1.7.5 最新发展

计算机网络中的变革继续以急促的步伐前进。所有的前沿研究正在取得进展，包括部署更快的路由器和在接入网和网络主干中提供更高的传输速率。但下列进展值得特别关注：

- 自 2000 年开始，我们见证了家庭宽带因特网接入的积极部署——不仅有电缆调制解调器和 DSL，而且有光纤到户，这些在 1.2 节中讨论过。这种高速因特网为丰富的视频应用创造了条件，包括用户生成的视频的分发（例如 YouTube）、电影和电视节目的按需流（例如 Netflix）以及多人视频会议（例如 Skype、Facetime 和 Google Hangouts）。
- 高速（54Mbps 及更高）公共 WiFi 网络和经过 4G 蜂窝电话网的中速（几十 Mbps）因特网接入越来越普及，不仅使在运动中保持持续连接成为可能，也产生了新型特定位置应用，如 Yelp、Tinder、Yik Yak 和 Waz。2011 年，与因特网连接的无线设备的数量超过了有线设备的数量。高速无线接入为手持计算机（iPhone、安卓手机、iPad 等）的迅速出现提供了舞台，这些手持计算机具有对因特网持续不断和无拘束接入的优点。
- 诸如脸书、Instagram、推特（Twitter）和微信（在中国极为流行）这样的在线社交网络已经在因特网之上构建了巨大的人际网络。这些社交网络，许多广泛用于发送消息以及照片分享。许多因特网用户今天主要“生活”在一个或多个社交网络中。通过他们的 API，在线社交网络为新的联网应用和分布式游戏创建了平台。
- 如在 1.3.3 节中所讨论的，在线服务提供商如谷歌和微软已经广泛部署了自己的专用网络。该专用网络不仅将它们分布在全球的数据中心连接在一起，而且通过直接与较低层 ISP 对等连接，能够尽可能绕过因特网。因此，谷歌几乎可以瞬间提供搜索结果和电子邮件访问，仿佛它们的数据中心运行在自己的计算机中一样。
- 许多因特网商务公司在“云”（如亚马逊的 EC2、谷歌的应用引擎、微软的 Azure）中运行它们的应用。许多公司和大学也已经将它们的因特网应用（如电子邮件和 Web 托管）迁移到云中。云公司不仅可以为应用提供可扩展的计算和存储环境，也可为应用提供对其高性能专用网络的隐含访问。

1.8 小结

在本章中，我们涉及了大量的材料！我们已经看到构成特别的因特网以及一般的计算机网络的各种硬件和软件。我们从网络的边缘开始，观察端系统和应用程序，以及运行在

端系统上为应用程序提供的运输服务。接着我们也观察了通常能够在接入网中找到的链路层技术和物理媒体。然后我们进入网络核心更深入地钻研网络，看到分组交换和电路交换是通过电信网络传输数据的两种基本方法，并且探讨了每种方法的长处和短处。我们也研究了全球性因特网的结构，知道了因特网是网络的网络。我们看到了因特网的由较高层和较低层 ISP 组成的等级结构，允许该网络扩展为包括数以千计的网络。

在这个概述性章节的第二部分，我们研究了计算机网络领域的几个重要主题。我们首先研究了分组交换网中的时延、吞吐量和丢包的原因。我们研究得到传输、传播和排队时延以及用于吞吐量的简单定量模型；我们将在整本书的课后习题中多处使用这些时延模型。接下来，我们研究了协议分层和服务模型、网络中的关键体系结构原则，我们将在本书多处引用它们。我们还概述了在今天的因特网中某些更为流行的安全攻击。我们用计算机网络的简要历史结束我们对网络的概述。第 1 章本身就构成了计算机网络的小型课程。

因此，第 1 章中的确涉及了大量的背景知识！如果你有些不知所云，请不要着急。在后继几章中我们将重新回顾这些概念，更为详细地研究它们（那是承诺，而不是威胁！）。此时，我们希望你完成本章内容的学习时，对构建网络的众多元素的直觉越来越敏锐，对网络词汇越来越精通（不妨经常回过头来查阅本章），对更加深入地学习网络的愿望越来越强烈。这些也是在本书的其余部分我们将面临的任务。

本书的路线图

在开始任何旅行之前，你总要先查看路线图，以便更为熟悉前面的主要道路和交叉路口。对于我们即将开启的这段“旅行”而言，其最终目的地是深入理解计算机网络“是什么、怎么样和为什么”等内容。我们的路线图是本书各章的顺序：

第 1 章 计算机网络和因特网

第 2 章 应用层

第 3 章 运输层

第 4 章 网络层：数据平面

第 5 章 网络层：控制平面

第 6 章 链路层和局域网

第 7 章 无线网络和移动网络

第 8 章 计算机网络中的安全

第 9 章 多媒体网络

第 2~6 章是本书的 5 个核心章。应当注意的是，这些章都围绕 5 层因特网协议栈上面的 4 层而组织，其中一章对应一层。要进一步注意的是，我们的旅行将从因特网协议栈的顶部即应用层开始，然后向下面各层进行学习。这种自顶向下旅行背后的基本原理是，一旦我们理解这些应用程序，就能够理解支持这些应用程序所需的各种网络服务。然后能够依次研究可能由网络体系结构实现的服务的各种方式。较早地涉及应用程序，也能够对学习本课程其余部分提供动力。

第 7~9 章关注现代计算机网络中的 3 个极为重要的（并且在某种程度上是独立的）主题。在第 7 章中，我们研究了无线网络和移动网络，包括无线 LAN（其中有 WiFi 和蓝牙）、蜂窝电话网（包括 GSM、3G 和 4G）以及移动性（在 IP 网络和 GSM 网络中的）。在第 8 章中，我们首先学习加密和网络安全的基础知识，然后研究基础理论如何应用于各种各样的因特网环境。在最后一章（第 9 章）中，我们研究了音频和视频应用，例如因特网

电话、视频会议和流式存储媒体。此外，还讨论如何设计分组交换网络以对音频和视频应用程序提供一致的服务质量。

课后习题和问题



复习题

1.1 节

R1. “主机”和“端系统”之间有什么不同？列举几种不同类型的端系统。Web 服务器是一种端系统吗？

R2. “协议”一词常被用于描述外交关系。维基百科是怎样描述外交协议的？

R3. 标准对于协议为什么重要？

1.2 节

R4. 列出 6 种接入技术。将它们分类为住宅接入、公司接入或广域无线接入。

R5. HFC 传输速率在用户间是专用的还是共享的？在下行 HFC 信道中，可能出现碰撞吗？为什么？

R6. 列出你所在城市中的可供使用的住宅接入技术。对于每种类型的接入方式，给出所宣称的下行速率、上行速率和每月的价格。

R7. 以太 LAN 的传输速率是多少？

R8. 能够运行以太网的一些物理媒体是什么？

R9. 拨号调制解调器、HFC、DSL 和 FTTH 都用于住宅接入。对于这些技术，给出每种技术的传输速率的范围，并讨论它们的传输速率是共享的还是专用的。

R10. 描述今天最为流行的无线因特网接入技术。对它们进行比较和对照。

1.3 节

R11. 假定在发送主机和接收主机间只有一台分组交换机。发送主机和交换机间以及交换机和接收主机间的传输速率分别是 R_1 和 R_2 。假设该交换机使用存储转发分组交换方式，发送一个长度为 L 的分组的端到端总时延是什么？（忽略排队时延、传播时延和处理时延。）

R12. 与分组交换网络相比，电路交换网络有哪些优点？在电路交换网络中，TDM 比 FDM 有哪些优点？

R13. 假定用户共享一条 2Mbps 链路。同时假定当每个用户传输时连续以 1Mbps 传输，但每个用户仅传输 20% 的时间。

a. 当使用电路交换时，能够支持多少用户？

b. 作为该题的后继问题，假定使用分组交换。为什么如果两个或更少的用户同时传输的话，在链路前面基本上没有排队时延？为什么如果 3 个用户同时传输的话，将有排队时延？

c. 求出某指定用户正在传输的概率。

d. 假定现在有 3 个用户。求出在任何给定的时间，所有 3 个用户在同时传输的概率。求出队列增长的时间比率。

R14. 为什么等级结构中级别相同的两个 ISP 通常互相对等？某 IXP 是如何挣钱的？

R15. 某些内容提供商构建了自己的网络。描述谷歌的网络。内容提供商构建这些网络的动机是什么？

1.4 节

R16. 考虑从某源主机跨越一条固定路由向某目的主机发送一分组。列出端到端时延中的时延组成成分。这些时延中的哪些是固定的，哪些是变化的？

R17. 访问配套 Web 网站上有关传输时延与传播时延的 Java 小程序。在速率、传播时延和可用的分组长度之中找出一种组合，使得该分组的第一个比特到达接收方之前发送方结束了传输。找出另一种组合，使得发送方完成传输之前，该分组的第一个比特到达了接收方。

R18. 一个长度为 1000 字节的分组经距离为 2500km 的链路传播，传播速率为 2.5×10^8 m/s 并且传输速率为 2Mbps，它需要用多长时间？更为一般地，一个长度为 L 的分组经距离为 d 的链路传播，传播速

- 率为 s 并且传输速率为 R bps, 它需要用多长时间? 该时延与传输速率相关吗?
- R19. 假定主机 A 要向主机 B 发送一个大文件。从主机 A 到主机 B 的路径上有 3 段链路, 其速率分别为 $R_1 = 500\text{kbps}$, $R_2 = 2\text{Mbps}$, $R_3 = 1\text{Mbps}$ 。
- 假定该网络中没有其他流量, 该文件传送的吞吐量是多少?
 - 假定该文件为 4MB。用吞吐量除以文件长度, 将该文件传输到主机 B 大致需要多长时间?
 - 重复 (a) 和 (b), 只是这时 R_2 减小到 100kbps。
- R20. 假定端系统 A 要向端系统 B 发送一个大文件。在一个非常高的层次上, 描述端系统怎样从该文件生成分组。当这些分组之一到达某分组交换机时, 该交换机使用分组中的什么信息来决定将该分组转发到哪一条链路上? 因特网中的分组交换为什么可以与驱车从一个城市到另一个城市并沿途询问方向相类比?
- R21. 访问配套 Web 站点的排队和丢包 Java 小程序。最大发送速率和最小的传输速率是多少? 对于这些速率, 流量强度是多大? 用这些速率运行该 Java 小程序并确定出现丢包要花费多长时间? 然后第二次重复该实验, 再次确定出现丢包花费多长时间。这些值有什么不同? 为什么会有这种现象?

1.5 节

- R22. 列出一个层次能够执行的 5 个任务。这些任务中的一个 (或两个) 可能由两个 (或更多) 层次执行吗?
- R23. 因特网协议栈中的 5 个层次有哪些? 在这些层次中, 每层的主要任务是什么?
- R24. 什么是应用层报文? 什么是运输层报文段? 什么是网络层数据报? 什么是链路层帧?
- R25. 路由器处理因特网协议栈中的哪些层次? 链路层交换机处理的是哪些层次? 主机处理的是哪些层次?

1.6 节

- R26. 病毒和蠕虫之间有什么不同?
- R27. 描述如何产生一个僵尸网络, 以及僵尸网络是怎样被用于 DDoS 攻击的。
- R28. 假定 Alice 和 Bob 经计算机网络互相发送分组。假定 Trudy 将自己安置在网络中, 使得她能够俘获由 Alice 发送的所有分组, 并发送她希望给 Bob 的东西; 她也能够俘获由 Bob 发送的所有分组, 并发送她希望给 Alice 的东西。列出在这种情况下 Trudy 能够做的某些恶意的事情。



习题

- P1. 设计并描述在自动柜员机和银行的中央计算机之间使用的一种应用层协议。你的协议应当允许验证用户卡和口令, 查询账目结算 (这些都在中央计算机中进行维护), 支取账目 (即向用户支付钱)。你的协议实体应当能够处理取钱时账目中钱不够的常见问题。通过列出自动柜员机和银行中央计算机在报文传输和接收过程中交换的报文和采取的动作来定义你的协议。使用类似于图 1-2 所示的图, 拟定在简单无差错取钱情况下该协议的操作。明确地阐述在该协议中关于底层端到端运输服务所做的假设。
- P2. 式 (1-1) 给出了经传输速率为 R 的 N 段链路发送长度 L 的一个分组的端到端时延。对于经过 N 段链路一个接一个地发送 P 个这样的分组, 一般化地表示出这个公式。
- P3. 考虑一个应用程序以稳定的速率传输数据 (例如, 发送方每 k 个时间单元产生一个 N 比特的数据单元, 其中 k 较小且固定)。另外, 当这个应用程序启动时, 它将连续运行相当长的一段时间。回答下列问题, 简要论证你的回答:
- 是分组交换网还是电路交换网更为适合这种应用? 为什么?
 - 假定使用了分组交换网, 并且该网中的所有流量都来自如上所述的这种应用程序。此外, 假定该应用程序数据传输速率的总和小于每条链路的各自容量。需要某种形式的拥塞控制吗? 为什么?
- P4. 考虑在图 1-13 中的电路交换网。回想在每条链路上有 4 条链路, 以顺时针方向标记四台交换机 A、

B、C 和 D。

- 在该网络中，任何时候能够进行同时连接的最大数量是多少？
- 假定所有连接位于交换机 A 和 C 之间。能够进行同时连接的最大数量是多少？
- 假定我们要在交换机 A 和 C 之间建立 4 条连接，在交换机 B 和 D 之间建立另外 4 条连接。我们能够让这些呼叫通过这 4 条链路建立路由以容纳所有 8 条连接吗？

P5. 回顾在 1.4 节中的车队的类比。假定传播速度为 100km/h。

- 假定车队旅行 150km：在一个收费站前面开始，通过第二个收费站，并且正好在第三个收费站后面结束。其端到端时延是多少？
- 重复 (a)，现在假定车队中有 8 辆汽车而不是 10 辆。

P6. 这个习题开始探讨传播时延和传输时延，这是数据网络中的两个重要概念。考虑两台主机 A 和 B 由一条速率为 R bps 的链路相连。假定这两台主机相隔 m 米，沿该链路的传播速率为 s m/s。主机 A 向主机 B 发送长度 L 比特的分组。

- 用 m 和 s 来表示传播时延 d_{prop} 。
- 用 L 和 R 来确定该分组的传输时间 d_{trans} 。
- 忽略处理和排队时延，得出端到端时延的表达式。
- 假定主机 A 在时刻 $t=0$ 开始传输该分组。在时刻 $t=d_{\text{trans}}$ ，该分组的最后一个比特在什么地方？
- 假定 d_{prop} 大于 d_{trans} 。在时刻 $t=d_{\text{trans}}$ ，该分组的第一个比特在何处？
- 假定 d_{prop} 小于 d_{trans} 。在时刻 $t=d_{\text{trans}}$ ，该分组的第一个比特在何处？
- 假定 $s=2.5 \times 10^8$ ， $L=120$ 比特， $R=56\text{kbps}$ 。求出使 d_{prop} 等于 d_{trans} 的距离 m 。

P7. 在这个习题中，我们考虑从主机 A 向主机 B 通过分组交换网发送语音 (VoIP)。主机 A 将模拟语音转换为传输中的 64kbps 数字比特流。然后主机 A 将这些比特分为 56 字节的分组。A 和 B 之间有一条链路：它的传输速率是 2Mbps，传播时延是 10ms。一旦 A 收集了一个分组，就将它向主机 B 发送。一旦主机 B 接收到一个完整的分组，它将该分组的比特转换成模拟信号。从比特产生（从位于主机 A 的初始模拟信号起）的时刻起，到该比特被解码（在主机 B 上作为模拟信号的一部分），花了多少时间？

P8. 假定用户共享一条 3Mbps 的链路。又设每个用户传输时要求 150kbps，但是每个用户仅有 10% 的时间传输。（参见 1.3 节中关于“分组交换与电路交换的对比”的讨论。）

- 当使用电路交换时，能够支持多少用户？
- 对于本习题的后续小题，假定使用分组交换。求出某给定用户正在传输的概率。
- 假定有 120 个用户。求出在任何给定时刻，实际有 n 个用户在同时传输的概率。（提示：使用二项式分布。）
- 求出有 21 个或更多用户同时传输的概率。

P9. 考虑在 1.3 节“分组交换与电路交换的对比”的讨论中，给出了一个具有一条 1Mbps 链路的例子。用户在忙时以 100kbps 速率产生数据，但忙时仅以 $p=0.1$ 的概率产生数据。假定用 1Gbps 链路替代 1Mbps 的链路。

- 当采用电路交换技术时，能被同时支持的最大用户数量 N 是多少？
- 现在考虑分组交换和有 M 个用户的情况。给出多于 N 用户发送数据的概率公式（用 p 、 M 、 N 表示）。

P10. 考虑一个长度为 L 的分组从端系统 A 开始，经 3 段链路传送到目的端系统。令 d_i 、 s_i 和 R_i 表示链路 i 的长度、传播速度和传输速率 ($i=1, 2, 3$)。该分组交换机对每个分组的时延为 d_{proc} 。假定没有排队时延，用 d_i 、 s_i 、 R_i ($i=1, 2, 3$) 和 L 表示，该分组总的端到端时延是什么？现在假定该分组是 1500 字节，在所有 3 条链路上的传播时延是 2.5×10^8 m/s，所有 3 条链路的传输速率是 2Mbps，分组交换机的处理时延是 3ms，第一段链路的长度是 5000km，第二段链路的长度是 4000km，并且最后一段链路的长度是 1000km。对于这些值，该端到端时延为多少？

P11. 在上述习题中，假定 $R_1=R_2=R_3=R$ 且 $d_{\text{proc}}=0$ 。进一步假定该分组交换机不存储转发分组，而是

在等待分组到达前立即传输它收到的每个比特。这时端到端时延为多少？

- P12. 一台分组交换机接收一个分组并决定该分组应当转发的出链路。当某分组到达时，另一个分组正在该出链路上被发送到一半，还有 4 个其他分组正等待传输。这些分组以到达的次序传输。假定所有分组是 1500 字节并且链路速率是 2Mbps。该分组的排队时延是多少？在更一般的情况下，当所有分组的长度是 L ，传输速率是 R ，当前正在传输的分组已经传输了 x 比特，并且已经在队列中有 n 个分组，其排队时延是多少？
- P13. a. 假定有 N 个分组同时到达一条当前没有分组传输或排队的链路。每个分组长为 L ，链路传输速率为 R 。对 N 个分组而言，其平均排队时延是多少？
b. 现在假定每隔 LN/R 秒有 N 个分组同时到达链路。一个分组的平均排队时延是多少？
- P14. 考虑某路由器缓存中的排队时延。令 I 表示流量强度；即 $I = La/R$ 。假定排队时延的形式为 $IL/R(1-I)$ ，其中 $I < 1$ 。
a. 写出总时延即排队时延加上传输时延的公式。
b. 以 L/R 为函数画出总时延的图。
- P15. 令 a 表示在一条链路上分组的到达率（以分组/秒计），令 μ 表示一条链路上分组的传输率（以分组/秒计）。基于上述习题中推导出的总时延公式（即排队时延加传输时延），推导出以 a 和 μ 表示的总时延公式。
- P16. 考虑一台路由器缓存前面的一条出链路。在这个习题中，将使用李特尔（Little）公式，这是排队论中的一个著名公式。令 N 表示在缓存中的分组加上被传输的分组的平均数。令 a 表示到达该链路的分组速率。令 d 表示一个分组历经的平均总时延（即排队时延加传输时延）。李特尔公式是 $N = a \times d$ 。假定该缓存平均包含 10 个分组，并且平均分组排队时延是 10ms。该链路的传输速率是 100 分组/秒。使用李特尔公式，在没有丢包的情况下，平均分组到达率是多少？
- P17. a. 对于不同的处理速率、传输速率和传播时延，给出 1.4.3 节中式 (1-2) 的一般表达式。
b. 重复 (a)，不过此时假定在每个节点有平均排队时延 d_{queue} 。
- P18. 在一天的 3 个不同的小时内，在同一个大陆上的源和目的地之间执行 Traceroute。
a. 在这 3 个小时的每个小时中，求出往返时延的均值和方差。
b. 在这 3 个小时的每个小时中，求出路径上的路由器数量。在这些时段中，该路径发生变化了吗？
c. 试图根据源到目的地 Traceroute 分组通过的情况，辨明 ISP 网络的数量。具有类似名字和/或类似的 IP 地址的路由器应当被认为是同一个 ISP 的一部分。在你的实验中，在相邻的 ISP 间的对等接口处出现最大的时延了吗？
d. 对位于不同大陆上的源和目的地重复上述内容。比较大陆内部和大陆之间的这些结果。
- P19. a. 访问站点 www.traceroute.org，并从法国两个不同的城市向位于美国的相同的目的主机执行 Traceroute。在这两个 Traceroute 中，有多少条链路是相同的？大西洋沿岸国家的链路相同吗？
b. 重复 (a)，但此时选择位于法国的一个城市和位于德国的另一个城市。
c. 在美国挑选一个城市，然后向位于中国的两个不同城市的主机执行 Traceroute。在这两次 Traceroute 中有多少链路是相同的？在到达中国前这两个 Traceroute 分开了吗？
- P20. 考虑对应于图 1-20b 吞吐量的例子。现在假定有 M 对客户-服务器而不是 10 对。用 R_s 、 R_c 和 R 分别表示服务器链路、客户链路和网络链路的速率。假设所有的其他链路都有充足容量，并且除了由这 M 对客户-服务器产生的流量外，网络中没有其他流量。推导出由 R_s 、 R_c 、 R 和 M 表示的通用吞吐量表达式。
- P21. 考虑图 1-19b。现在假定在服务器和客户之间有 M 条路径。任两条路径都不共享任何链路。路径 k ($k=1, \dots, M$) 由传输速率为 $R_1^k, R_2^k, \dots, R_N^k$ 的 N 条链路组成。如果服务器仅能够使用一条路径向客户发送数据，则该服务器能够取得的最大吞吐量是多少？如果该服务器能够使用所有 M 条路径发送数据，则该服务器能够取得的最大吞吐量是多少？
- P22. 考虑图 1-19b。假定服务器与客户之间的每条链路的丢包概率为 p ，且这些链路的丢包率是独立的。一个（由服务器发送的）分组成功地被接收方收到的概率是多少？如果在从服务器到客户的路径上

分组丢失了，则服务器将重传该分组。平均来说，为了使客户成功地接收该分组，服务器将要重传该分组多少次？

P23. 考虑图 1-19a。假定我们知道沿着从服务器到客户的路径的瓶颈链路是速率为 R_1 bps 的第一段链路。假定我们从服务器向客户发送紧密相连的一对分组，且沿这条路径没有其他流量。假定每个分组的长度为 L 比特，两条链路具有相同的传播时延 d_{prop} 。

- 在目的地，分组的到达间隔时间有多大？也就是说，从第一个分组的最后一个比特到达第二个分组最后一个比特到达所经过的时间有多长？
- 现在假定第二段链路是瓶颈链路（即 $R_2 < R_1$ ）。第二个分组在第二段链路输入队列中排队是可能的吗？请解释原因。现在假定服务器在发送第一个分组 T 秒之后再发送第二个分组。为确保在第二段链路之前没有排队， T 必须要有多长？试解释原因。

P24. 假设你希望从波士顿向洛杉矶紧急传送 40×10^{12} 字节数据。你有一条 100Mbps 专用链路可用于传输数据。你是愿意通过这条链路传输数据，还是愿意使用 FedEx 夜间快递来交付？解释你的理由。

P25. 假定两台主机 A 和 B 相隔 20 000km，由一条直接的 $R = 2\text{Mbps}$ 的链路相连。假定跨越该链路的传播速率是 $2.5 \times 10^8 \text{ m/s}$ 。

- 计算带宽-时延积 $R \cdot t_{\text{prop}}$ 。
- 考虑从主机 A 到主机 B 发送一个 800 000 比特的文件。假定该文件作为一个大的报文连续发送。在任何给定的时间，在链路上具有的比特数量最大值是多少？
- 给出带宽-时延积的一种解释。
- 在该链路上一个比特的宽度（以米计）是多少？它比一个足球场更长吗？
- 用传播速率 s 、带宽 R 和链路 m 的长度表示，推导出一个比特宽度的一般表示式。

P26. 对于习题 P25，假定我们能够修改 R 。对什么样的 R 值，一个比特的宽度能与该链路的长度一样长？

P27. 考虑习题 P25，但此时链路的速率是 $R = 1\text{Gbps}$ 。

- 计算带宽-时延积 $R \cdot d_{\text{prop}}$ 。
- 考虑从主机 A 到主机 B 发送一个 800 000 比特的文件。假定该文件作为一个大的报文连续发送。在任何给定的时间，在链路上具有的比特数量最大值是多少？
- 在该链路上一个比特的宽度（以米计）是多少？

P28. 再次考虑习题 P25。

- 假定连续发送，发送该文件需要多长时间？
- 假定现在该文件被划分为 20 个分组，每个分组包含 40 000 比特。假定每个分组被接收方确认，确认分组的传输时间可忽略不计。最后，假定前一个分组被确认后，发送方才能发送分组。发送该文件需要多长时间？
- 比较（a）和（b）的结果。

P29. 假定在同步卫星和它的地球基站之间有一条 10Mbps 的微波链路。每分钟该卫星拍摄一幅数字照片，并将它发送到基站。假定传播速率是 $2.4 \times 10^8 \text{ m/s}$ 。

- 该链路的传播时延是多少？
- 带宽-时延积 $R \cdot d_{\text{prop}}$ 是多少？
- 若 x 表示该照片的大小。对于这条微波链路，能够连续传输的 x 最小值是多少？

P30. 考虑 1.5 节中我们在分层讨论中对航空旅行的类比，随着协议数据单元向协议栈底层流动，首部在增加。随着旅客和行李移动到航线协议栈底部，有与上述首部信息等价的概念吗？

P31. 在包括因特网的现代分组交换网中，源主机将长应用层报文（如一个图像或音乐文件）分段为较小的分组并向网络发送。接收方则将这些分组重新装配为初始报文。我们称这个过程为报文分段。图 1-27 显示了一个报文在报文不分段或报文分段情况下的端到端传输。考虑一个长度为 8×10^6 比特的报文，它在图 1-27 中从源发送到目的地。假定在该图中的每段链路是 2Mbps。忽略传播、排队和处理时延。

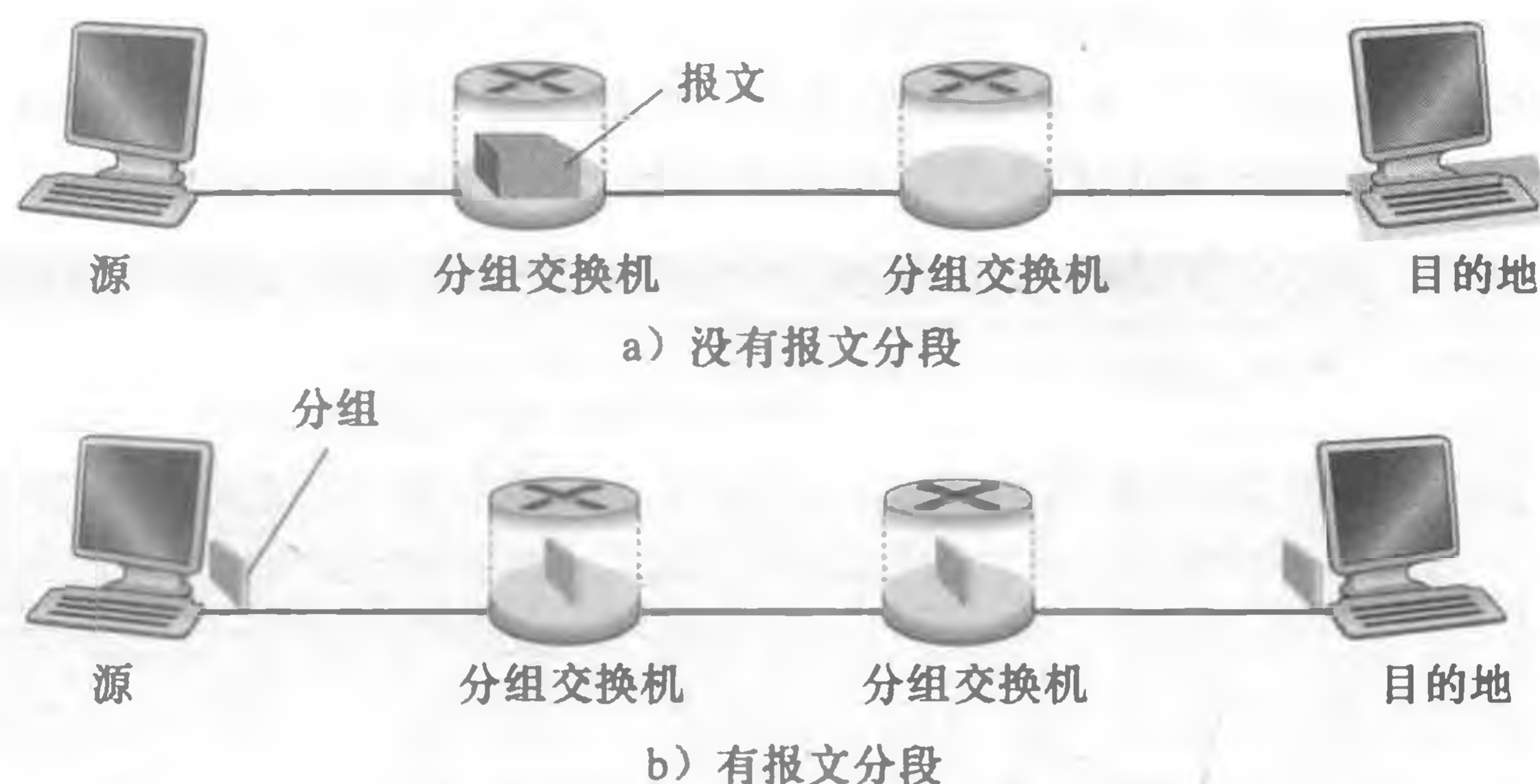


图 1-27 端到端报文传输

- 考虑从源到目的地发送该报文且没有报文分段。从源主机到第一台分组交换机移动报文需要多长时间？记住，每台交换机均使用存储转发分组交换，从源主机移动该报文到目的主机需要多长时间？
- 现在假定该报文被分段为 800 个分组，每个分组 10 000 比特长。从源主机移动第一个分组到第一台交换机需要多长时间？从第一台交换机发送第一个分组到第二台交换机，从源主机发送第二个分组到第一台交换机各需要多长时间？什么时候第二个分组能被第一台交换机全部收到？
- 当进行报文分段时，从源主机向目的主机移动该文件需要多长时间？将该结果与 (a) 的答案进行比较并解释之。
- 除了减小时延外，使用报文分段还有什么原因？
- 讨论报文分段的缺点。

P32. 用本书的 Web 网站上的报文分段小 Java 小程序进行实验。该 Java 程序中的时延与前一个习题中的时延相当吗？链路传播时延是怎样影响分组交换（有报文分段）和报文交换的端到端总时延的？

P33. 考虑从主机 A 到主机 B 发送一个 F 比特的大文件。A 和 B 之间有三段链路（和两台交换机），并且该链路不拥塞（即没有排队时延）。主机 A 将该文件分为每个为 S 比特的报文段，并为每个报文段增加一个 80 比特的首部，形成 $L = 80 + S$ 比特的分组。每条链路的传输速率为 R bps。求出从 A 到 B 移动该文件时延最小的值 S 。忽略传播时延。

P34. Skype 提供了一种服务，使你能用 PC 向普通电话打电话。这意味着语音呼叫必须通过因特网和电话网。讨论这是如何做到的。



Wireshark 实验

“不闻不若闻之，闻之不若见之，见之不若知之，知之不若行之。”

——中国谚语

一个人通常能够通过以下方法加深对网络协议的理解：观察它们的动作和经常摆弄它们，即观察两个协议实体之间交换的报文序列，钻研协议运行的细节，使协议执行某些动作，观察这些动作及其后果。这能够在仿真环境下或在如因特网这样的真实网络环境下完成。在本书配套 Web 站点上的 Java 小程序采用的是第一种方法。在 Wireshark 实验中，我们将采用后一种方法。你可以在家中或实验室中使用桌面计算机在各种情况下运行网络应用程序。在你的计算机上观察网络协议，它是如何与在因特网别处执行的协议实体交互和交换报文的。因此，你与你的计算机将是这些真实实验的有机组成部分。你将通过动手来观察和学习。

用来观察执行协议实体之间交换的报文的基本工具称为分组嗅探器（packet sniffer）。顾名思义，一个分组嗅探器被动地拷贝（嗅探）由你的计算机发送和接收的报文；它也能显示出这些被捕获报文的各个协议字段的内容。图 1-28 中显示了 Wireshark 分组嗅探器的屏幕快照。Wireshark 是一个运行在 Win-

dows、Linux/Unix 和 Mac 计算机上的免费分组嗅探器。贯穿全书，你将发现 Wireshark 实验能让你探索在该章中学习的一些协议。在这第一个 Wireshark 实验中，你将获得并安装一个 Wireshark 的副本，访问一个 Web 站点，捕获并检查在你的 Web 浏览器和 Web 服务器之间交换的协议报文。

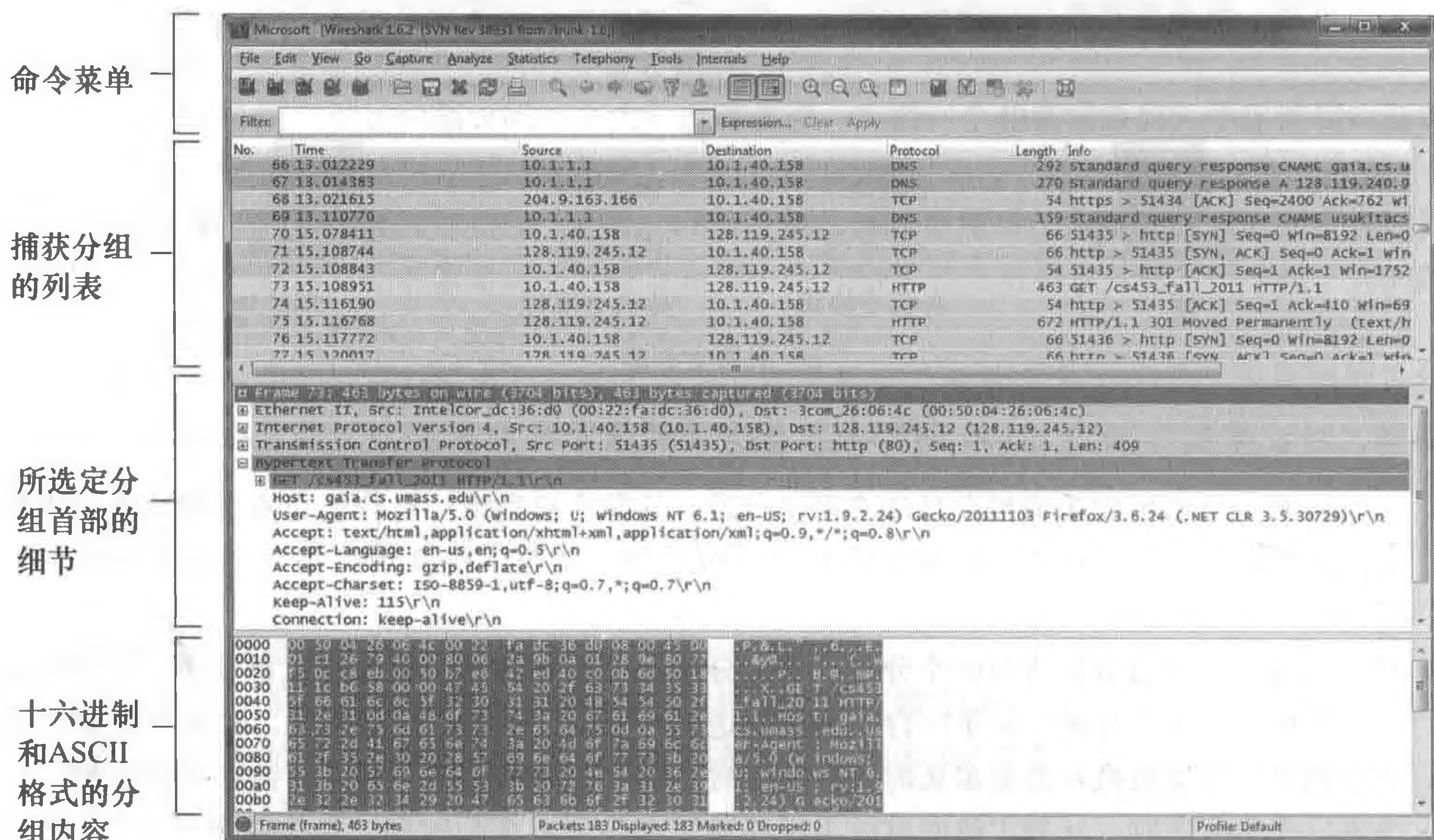


图 1-28 一个 Wireshark 屏幕快照（打印的 Wireshark 屏幕快照得到了 Wireshark 基金会的许可）

你能够在 Web 站点 <http://www.pearsonhighered.com/cs-resources/> 上找到有关该第一个 Wireshark 实验的全部材料（包括如何获得并安装 Wireshark 的指导）。

人物专访

Leonard Kleinrock 是加州大学洛杉矶分校（UCLA）的计算机科学教授。1969 年，他在 UCLA 的计算机成为因特网的第一个节点。1961 年，他创造的分组交换原理成为因特网的支撑技术。他在纽约城市大学（City College of New York, CCNY）获得电气工程学士学位，并在麻省理工学院（MIT）获得电气工程硕士和博士学位。

- 是什么使得您决定专门研究网络/因特网技术的？

当我于 1959 年在 MIT 读博士时，我发现周围的大多数同学正在信息理论和编码理论领域做研究。在 MIT，那时有伟大的研究者 Claude Shannon，他已经开创这些领域，并且已经解决了许多重要的问题。留下来的研究问题既难又不太重要。因此我决定开始新的研究领域，而该领域其他人还没有想到。回想那时在 MIT 我的周围有许多计算机，我很清楚很快这些计算机将有相互通信的需求。在那时，却没有有效的办法来做到这一点，因此我决定研发能够创建有效的数据网络的技术。

- 您在计算机产业的第一份工作是什么？它使您有哪些收益？

1951~1957 年，我为了获得电子工程学士学位在 CCNY 读夜大。在那段时间里，我在一家称为 Photobell 的工业电子小公司工作，先是当技术员，然后当工程师。在那里，我在它们的产品线上引入了数字技术。我们主要使用光电子设备来检测某些物体（盒子、人等）的存在，一种称为双稳态多频振荡器的电路的使用正是我们需要的技术类型，它能将数字处理引入检测领域。这些电路恰好是计算机的基本模块，用今天的话说就是触发电路或交换器。

- 当您发送第一个主机到主机报文（从 UCLA 到斯坦福研究院）时，您心中想到了什么？



Leonard Kleinrock

坦率地说，我们当时并没有想到那件事的重要性。我们没有准备具有历史意义的豪言壮语，就像昔日许多发明家所做的那样（如塞缪尔·莫尔斯的“上帝创造了什么（What hath God wrought）”，亚历山大·格瑞汉姆·贝尔的“Watson 先生，请来这里！我想见你”，或尼尔·阿姆斯特朗的“个人的一小步，人类的一大步”）。多么聪明的人哪！他们明白媒体和公众的关系。我们要做的所有工作是向斯坦福研究院的计算机进行注册。当我们键入“L”，它被正确收到，当我们键入“o”又被正确收到，而当我们键入“g”则引起斯坦福研究院主机的崩溃！因此，这将我们的报文转换为最短的，也许是最有预测性的报文，“Lo!”即为“真想不到（Lo and behold）!”。

那年早些时候，UCLA 新闻稿引用我的话说，一旦该网络建立并运行起来，将可能从我们的家中和办公室访问计算机设施，就像我们获得电力和电话连接那样容易。因此那时我的美好愿望是，因特网将是一个无所不在的、总是运行的、总是可用的网络，任何人从任何地方用任何设备将能够与之相连，并且它将是不可见的。然而，我从没有期待我的 99 岁的母亲将能够上因特网，但她确实做到了这一点。

- 您对未来网络的展望是什么？

我的展望中最容易的部分是预测基础设施本身。我预见我们看到移动计算、移动设备和智能空间的大量部署。轻量级、廉价、高性能、便携的计算和通信设备（加上因特网的无处不在）的确使我们成为游牧一员。游牧计算是指使从一个地方旅行到另一个地方的端用户，以透明方式访问因特网服务，无论他们旅行到何处，无论他们携带什么设备或获得何种接入。展望中最困难的部分是预测应用和服务，它们以引人注目的方式不断地带给我们惊喜（电子邮件、搜索技术、万维网、博客、社交网络、用户一代以及音乐、照片和视频等的共享）。我们正面临一种新的惊奇和创新，即移动应用装载于手持设备中。

下一步将使我们从信息空间虚拟世界（netherworld）移动到智能空间的物理世界。我们的环境（办公桌、墙壁、车辆、钟表、腰带等）将因技术而栩栩如生，这些技术包括执行器（actuator）、传感器、逻辑、处理、存储、照相机、麦克风、话筒、显示器和通信。这种嵌入式技术将使得环境能提供我们需要的 IP 服务。当我走进一间房间时，该房间知道我的到来。我将能够与环境自然地通信，如同说英语一样；我的请求产生的响应将从墙上的显示器通过我的眼镜以 Web 网页的形式呈现给我，就像说话、全息照相等一样。

再向前看一点，我看到未来的网络包括了下列附加的关键组件。我看到在网络各处部署的智能软件代理，它的功能是挖掘数据，根据数据采取动作，观察趋势，并能动态地、自适应地执行任务。我看到相当多的网络流量并不是由人产生的，而是由这些嵌入式设备和这些智能软件代理产生的。我看到大批的自组织系统控制这个巨大、快速的网络。我看到巨量的信息瞬间通过网络立即得到强力处理和过滤。因特网最终将是一个无所不在的全球性神经系统。当我们急速在 21 世纪进发时，我将看到这些东西和更多的东西。

- 是谁激发了您的职业灵感？

到目前为止，是麻省理工学院的 Claude Shannon。他是一名卓越的研究者，具有以高度直觉的方式将他的数学理念与物理世界关联起来的能力。他是我的博士论文答辩委员会的成员。

- 您对进入网络/因特网领域的学生有什么忠告吗？

因特网和由它使能的所有东西是一个巨大的新前沿，充满了令人惊奇的挑战，为众多创新提供了广阔空间。不要受今天技术的束缚，开动大脑，想象能够做些什么，并去实现它。

应用层

网络应用是计算机网络存在的理由，如果我们不能构想出任何有用的应用，也就没有任何必要去设计支持它们的网络协议了。自因特网全面发展以来，的确已开发出众多有用的、有趣的网络应用。这些应用程序已经成为因特网成功的驱动力，激励人们在家庭、学校、政府和商业中利用网络，使因特网成为他们日常活动的密不可分的一部分。

因特网应用包括：20 世纪 70 年代和 80 年代开始流行的经典的基于文本的应用，如文本电子邮件、远程访问计算机、文件传输和新闻组；20 世纪 90 年代中期引入的招人喜爱的应用——万维网，包括 Web 冲浪、搜索和电子商务；20 世纪末引入的两个招人喜爱的应用——即时讯息和对等（P2P）文件共享。自 2000 年以来，新型和极其引人入胜的应用持续出现，包括：IP 电话（VoIP）、视频会议（如 Skype、Facetime 和 Google Hangouts）；用户生成的视频（如 YouTube）和点播电影（如 Netflix）；多方在线游戏（如《第二人生》（Second Life）和《魔兽世界》（World of Warcraft））。在这段时期，我们看到了新一代社交网络应用，如 Facebook、Instagram、Twitter 和微信，它们在因特网的网络或路由器和通信链路之上创建了引人入胜的人类网络。近年来，随着智能手机的发展，出现了大量基于位置的移动应用程序，包括流行的签到、约会和道路流量预测应用（如 Yelp、Tinder、Waz 和 Yik Yak）。显然，新型和令人兴奋的因特网应用的步伐并没有减缓。也许本书的一些读者将会创建下一代招人喜爱的因特网应用。

在本章中，我们学习有关网络应用的原理和实现方面的知识。我们从定义关键的应用层概念开始，其中包括应用程序所需要的网络服务、客户和服务端、进程和运输层接口。我们详细考察几种网络应用程序，包括 Web、电子邮件、DNS、对等文件分发和视频流（第 9 章关注多媒体应用，包括流式视频和 VoIP）。然后我们将涉及开发运行在 TCP 和 UDP 上的网络应用程序。特别是，我们学习套接字接口，并浮光掠影地学习用 Python 语言写的一些简单的客户-服务器应用程序。在本章结尾，我们也将提供几个有趣、有意思的套接字编程作业。

应用层是我们学习协议非常好的起点，它最为我们所熟悉。我们熟悉的很多应用就是建立在这些将要学习的协议基础上的。通过对应用层的学习，将有助于我们很好地感受到协议的方方面面，将使我们了解到很多问题，这些问题在我们学习运输层、网络层及数据链路层协议时也同样会碰到。

2.1 应用层协议原理

假定你对新型网络应用有了一些想法。也许这种应用将为人类提供一种伟大的服务，或者将使你的教授高兴，或者将带给你大量的财富，或者只是在开发中获得乐趣。无论你的动机是什么，我们现在考察一下如何将你的想法转变为一种真实世界的网络应用。

研发网络应用程序的核心是写出能够运行在不同的端系统和通过网络彼此通信的程序。例如，在 Web 应用程序中，有两个互相通信的不同的程序：一个是运行在用户主机（桌面

机、膝上机、平板电脑、智能电话等) 上的浏览器程序; 另一个是运行在 Web 服务器主机上的 Web 服务器程序。另一个例子是 P2P 文件共享系统, 在参与文件共享的社区中的每台主机中都有一个程序。在这种情况下, 在各台主机中的这些程序可能都是类似的或相同的。

因此, 当研发新应用程序时, 你需要编写将在多台端系统上运行的软件。该软件能够用如 C、Java 或 Python 来编写。重要的是, 你不需要写在网络核心设备如路由器或链路层交换机上运行的软件。即使你要为网络核心设备写应用程序软件, 你也不能做到这一点。如我们在第 1 章所知, 以及如图 1-24 所显示的那样, 网络核心设备并不在应用层上起作用, 而仅在较低层起作用, 特别是在网络层及下面层次起作用。这种基本设计, 即将应用软件限制在端系统 (如图 2-1 所示) 的方法, 促进了大量的网络应用程序的迅速研发和部署。

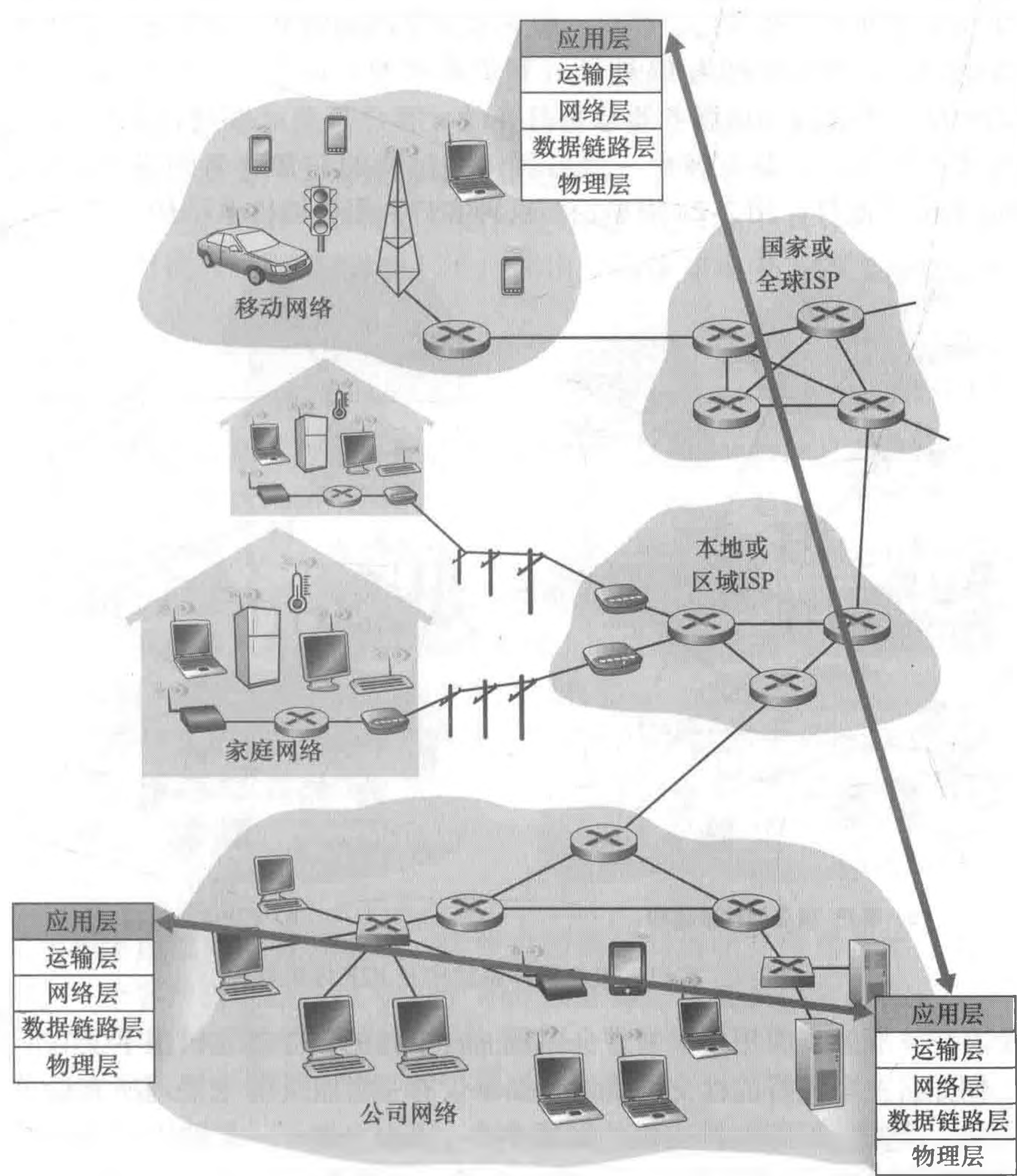


图 2-1 在应用层的端系统之间的网络应用的通信

2. 1. 1 网络应用程序体系结构

当进行软件编码之前, 应当对应用程序有一个宽泛的体系结构计划。记住应用程序的体系结构明显不同于网络的体系结构 (例如在第 1 章中所讨论的 5 层因特网体系结构)。从应用程序研发者的角度看, 网络体系结构是固定的, 并为应用程序提供了特定的服务集

合。在另一方面，应用程序体系结构（application architecture）由应用程序研发者设计，规定了如何在各种端系统上组织该应用程序。在选择应用程序体系结构时，应用程序研发者很可能利用现代网络应用程序中所使用的两种主流体系结构之一：客户-服务器体系结构或对等（P2P）体系结构。

在客户-服务器体系结构（client-server architecture）中，有一个总是打开的主机称为服务器，它服务于来自许多其他称为客户的主机的请求。一个典型的例子是 Web 应用程序，其中总是打开的 Web 服务器服务于来自浏览器（运行在客户主机上）的请求。当 Web 服务器接收到来自某客户对某对象的请求时，它向该客户发送所请求的对象作为响应。值得注意的是利用客户-服务器体系结构，客户相互之间不直接通信；例如，在 Web 应用中两个浏览器并不直接通信。客户-服务器体系结构的另一个特征是该服务器具有固定的、周知的地址，该地址称为 IP 地址（我们将很快讨论它）。因为该服务器具有固定的、周知的地址，并且因为该服务器总是打开的，客户总是能够通过向该服务器的 IP 地址发送分组来与其联系。具有客户-服务器体系结构的非常著名的应用程序包括 Web、FTP、Telnet 和电子邮件。图 2-2a 中显示了这种客户-服务器体系结构。

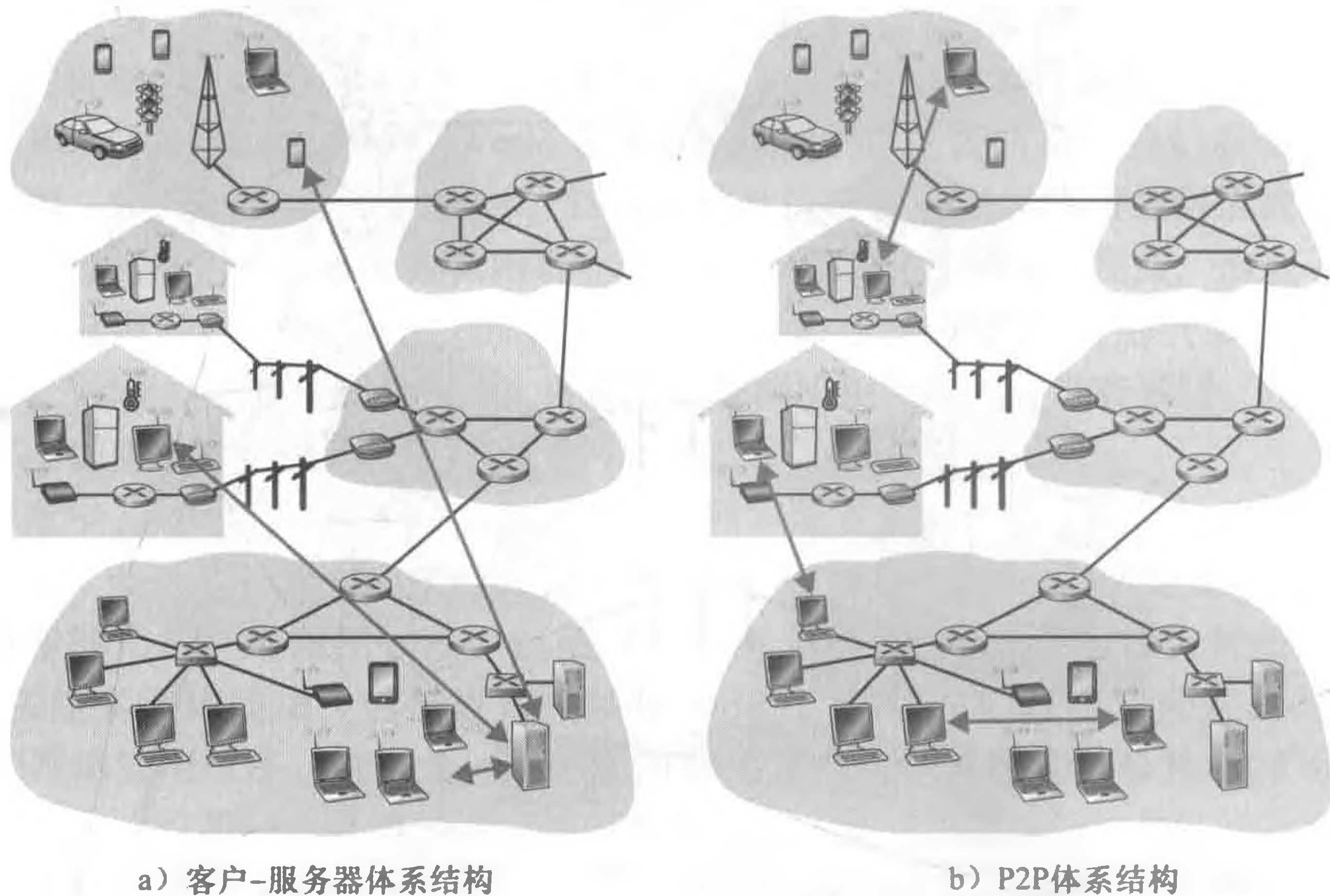


图 2-2 客户-服务器体系结构及 P2P 体系结构

在一个客户-服务器应用中，常常会出现一台单独的服务器主机跟不上它所有客户请求的情况。例如，一个流行的社交网络站点如果仅有一台服务器来处理所有请求，将很快变得不堪重负。为此，配备大量主机的数据中心（data center）常被用于创建强大的虚拟服务器。最为流行的因特网服务——如搜索引擎（如谷歌、Bing 和百度）、因特网商务（如亚马逊、e-Bay 和阿里巴巴）、基于 Web 的电子邮件（如 Gmail 和雅虎邮件）、社交网络（如脸书、Instagram、推特和微信），就应用了一个或多个数据中心。如在 1.3.3 节中所讨论的那样，谷歌有分布于全世界的 30~50 个数据中心，这些数据中心共同处理搜索、YouTube、Gmail 和其他服务。一个数据中心能够有数十万台服务器，它们必须要供电和维护。此外，服务提供商必须支付不断出现的互联和带宽费用，以发送和接收到达/来自数据中心的数据。

在一个 P2P 体系结构 (P2P architecture) 中, 对位于数据中心的专用服务器有最小的 (或者没有) 依赖。相反, 应用程序在间断连接的主机对之间使用直接通信, 这些主机对被称为对等方。这些对等方并不为服务提供商所有, 相反却为用户控制的桌面机和膝上机所有, 大多数对等方驻留在家庭、大学和办公室。因为这种对等方通信不必通过专门的服务器, 该体系结构被称为对等方到对等方的。许多目前流行的、流量密集型应用都是 P2P 体系结构的。这些应用包括文件共享 (例如 BitTorrent)、对等方协助下载加速器 (例如迅雷)、因特网电话和视频会议 (例如 Skype)。图 2-2b 中显示了 P2P 的体系结构。需要提及的是, 某些应用具有混合的体系结构, 它结合了客户 - 服务器和 P2P 的元素。例如, 对于许多即时讯息应用而言, 服务器被用于跟踪用户的 IP 地址, 但用户到用户的报文在用户主机之间 (无须通过中间服务器) 直接发送。

P2P 体系结构的最引人入胜的特性之一是它们的自扩展性 (self-scalability)。例如, 在一个 P2P 文件共享应用中, 尽管每个对等方都由于请求文件产生工作负载, 但每个对等方通过向其他对等方分发文件也为系统增加服务能力。P2P 体系结构也是有成本效率的, 因为它们通常不需要庞大的服务器基础设施和服务器带宽 (这与具有数据中心的客户 - 服务器设计形成鲜明对比)。然而, 未来 P2P 应用由于高度非集中式结构, 面临安全性、性能和可靠性等挑战。

2.1.2 进程通信

在构建网络应用程序前, 还需要对运行在多个端系统上的程序是如何互相通信的情况有一个基本了解。用操作系统的术语来说, 进行通信的实际上是进程 (process) 而不是程序。一个进程可以被认为是运行在端系统中的一个程序。当多个进程运行在相同的端系统上时, 它们使用进程间通信机制相互通信。进程间通信的规则由端系统上的操作系统确定。而在本书中, 我们并不特别关注同一台主机上的进程间的通信, 而关注运行在不同端系统 (可能具有不同的操作系统) 上的进程间的通信。

在两个不同端系统上的进程, 通过跨越计算机网络交换报文 (message) 而相互通信。发送进程生成并向网络中发送报文; 接收进程接收这些报文并可能通过回送报文进行响应。图 2-1 显示了驻留在 5 层协议栈的应用层进程互相通信的情况。

1. 客户和服务进程

网络应用程序由成对的进程组成, 这些进程通过网络相互发送报文。例如, 在 Web 应用程序中, 一个客户浏览器进程与一台 Web 服务器进程交换报文。在一个 P2P 文件共享系统中, 文件从一个对等方中的进程传输到另一个对等方中的进程。对每对通信进程, 我们通常将这两个进程之一标识为客户 (client), 而另一个进程标识为服务器 (server)。对于 Web 而言, 浏览器是一个客户进程, Web 服务器是一台服务器进程。对于 P2P 文件共享, 下载文件的对等方标识为客户, 上载文件的对等方标识为服务器。

你或许已经观察到, 如在 P2P 文件共享的某些应用中, 一个进程能够既是客户又是服务器。在 P2P 文件共享系统中, 一个进程的确既能上载文件又能下载文件。无论如何, 在任何给定的一对进程之间的通信会话场景中, 我们仍能将一个进程标识为客户, 另一个进程标识为服务器。我们定义客户和服务进程如下:

在一对进程之间的通信会话场景中, 发起通信 (即在该会话开始时发起与其他进程的联系人) 的进程被标识为客户, 在会话开始时等待联系的进程是服务器。