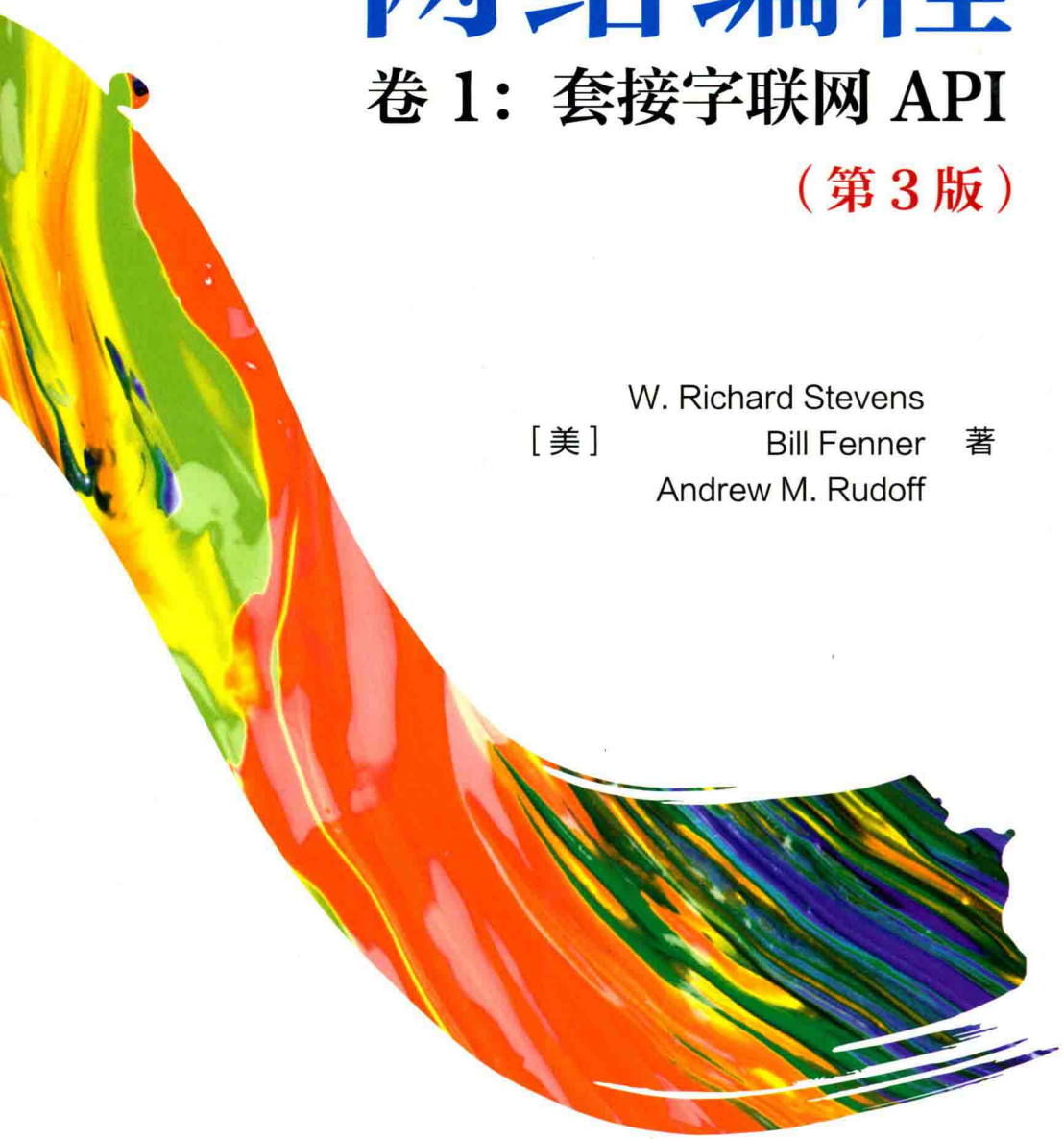
PEARSON

UNIX 网络编程

卷 1：套接字联网 API

（第 3 版）

[美] W. Richard Stevens
Bill Fenner
Andrew M. Rudoff 著



中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

Unix Network Programming, Volume 1: The Sockets Networking API Third Edition

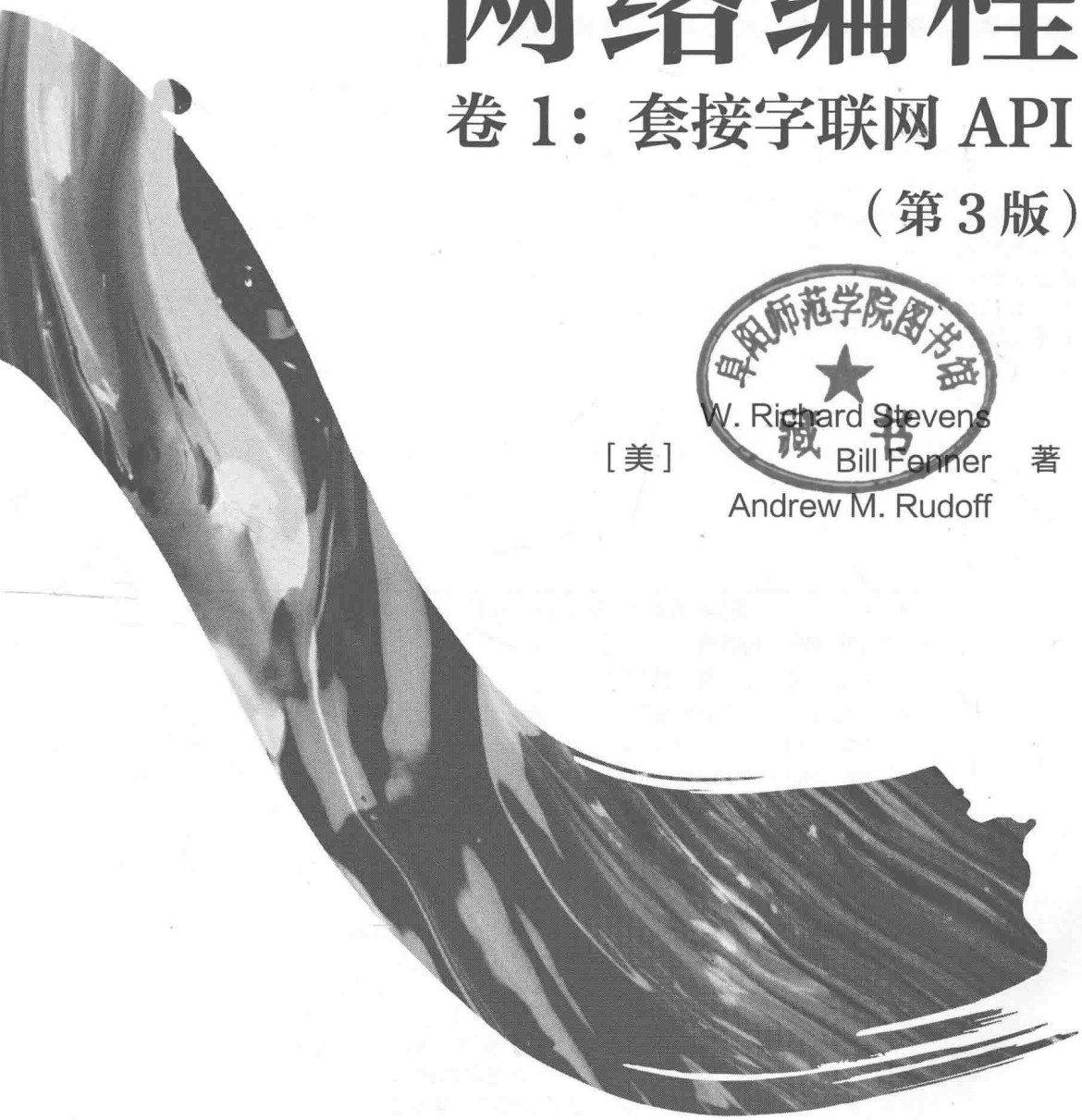
PEARSON

UNIX 网络编程

卷 1：套接字联网 API

（第 3 版）

[美] W. Richard Stevens
Bill Fenner 著
Andrew M. Rudoff



人民邮电出版社
北京

图书在版编目（CIP）数据

UNIX网络编程 : 第3版. 第1卷, 套接字联网API / (美) 史蒂文斯 (Stevens,W.R.) , (美) 芬纳 (Fenner,B.) , (美) 鲁道夫 (Rudoff,A.M.) 著 ; 匿名译. -- 2版. -- 北京 : 人民邮电出版社, 2015.8
书名原文: Unix Network Programming, Volume 1: The Sockets Networking API, Third Edition
ISBN 978-7-115-36719-8

Ⅰ. ①U… Ⅱ. ①史… ②芬… ③鲁… ④匿… Ⅲ. ①UNIX操作系统－程序设计 Ⅳ. ①TP316.81

中国版本图书馆CIP数据核字(2015)第143512号

内 容 提 要

本书是一部 UNIX 网络编程的经典之作！书中全面深入地介绍了如何使用套接字 API 进行网络编程。全书不但介绍了基本编程内容，还涵盖了与套接字编程相关的高级主题，对于客户/服务器程序的各种设计方法也作了完整的探讨，最后还深入分析了流这种设备驱动机制。

本书内容详尽且具权威性，几乎每章都提供精选的习题，并提供了部分习题的答案，是网络研究和开发人员理想的参考书。

◆ 著 [美] W. Richard Stevens Bill Fenner Andrew M. Rudoff
责任编辑 杨海玲
责任印制 张佳莹 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：787×1092 1/16
印张：51.5
字数：1 363 千字 2015 年 8 月第 2 版
印数：1-4 500 册 2015 年 8 月北京第 1 次印刷

著作权合同登记号 图字：01-2009-5715 号

定价：129.00 元

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第 0021 号

版 权 声 明

序

本书的第1版于1990年问世，并迅速成为程序员学习网络编程的权威参考书。时至今日，计算机网络技术已发生了翻天覆地的变化。只要看看第1版给出的用于征集反馈意见的地址（“uunet!hsi!netbook”）就一目了然了。（有多少读者能看出这是20世纪80年代很流行的UUCP拨号网络的地址？）

现在UUCP网络已经很罕见了，而无线网络等新技术则变得无处不在！在这种背景下，新的网络协议和编程范型业已开发出来，但程序员却苦于找不到一本好的参考书来学习这些复杂的新技术。

这本书填补了这一空白。拥有本书旧版的读者一定想要一个新的版本来学习新的编程方法，了解IPv6等下一代协议方面的新内容。所有人都非常期待本书，因为它完美地结合了实践经验、历史视角以及在本领域浸淫多年才能获得的透彻理解。

阅读本书是一种享受，我收获颇丰。相信大家定会有同感。

Sam Leffler

前　言

概述

本书面向的读者是那些希望自己编写的程序能使用称为套接字（socket）的API进行彼此通信的人。有些读者可能已经非常熟悉套接字了，因为这个模型几乎已经成了网络编程的同义词，但有些读者可能仍需要从头开始学习。本书想达到的目标是向大家提供网络编程指导。这些内容不仅适用于专业人士，也适用于初学者；不仅适用于维护已有代码，也适用于开发新的网络应用程序；此外，还适用于那些只是想了解一下自己系统中网络组件的工作原理的人。

书中的所有示例都是在Unix系统上测试通过的真实的、可运行的代码。但是，考虑到许多非Unix的操作系统也支持套接字API，因而我们选取的示例与所讲述的一般性概念，在很大程度上是与操作系统无关的。几乎每种操作系统都提供了大量的网络应用程序，如网页浏览器、电子邮件客户端、文件共享服务器等。我们按常规的划分方法把这些应用程序分为客户程序和服务器程序，并在书中多次编写了相应的小型示例。

面向Unix介绍网络编程自然免不了要介绍Unix本身和TCP/IP的相关背景知识。需要更详尽的背景知识时，我们会指引读者查阅其他书籍。本书中经常提到以下4本书，我们将其简记如下。

- APUE：*Advanced Programming in the UNIX Environment* [Stevens 1992]。
- TCPv1：*TCP/IP Illustrated, Volume 1* [Stevens 1994]。
- TCPv2：*TCP/IP Illustrated, Volume 2* [Wright and Stevens 1995]。
- TCPv3：*TCP/IP Illustrated, Volume 3* [Stevens 1996]。

其中TCPv2包含了与本书内容密切相关的细节，它描述并给出了套接字API中网络编程函数（`socket`、`bind`、`connect`等）的真实4.4BSD实现。如果已经理解某个特性的实现，那么在应用程序中使用该特性就更有意义了。

与第 2 版的区别

从20世纪80年代开始，套接字就差不多是现在这个样子了。时至今日，套接字仍然是网络API的首选，其最初的设计的确值得称道。因此，当读者发现我们对出版于1998年的第2版又做了不少改动时，可能会觉得惊讶。本书中所做的改动归纳如下。

- 新版本包含了IPv6的最新信息。在第2版出版时，IPv6尚处于草案阶段，这些年来已经有所发展。
- 更新了全部函数和示例的描述，以反映最新的POSIX规范（POSIX 1003.1-2001），即*Single Unix Specification Version 3*。
- 删去了X/Open传输接口（XTI）的内容。这个API已经不常用了，连最新的POSIX 规范也不再提到。

- 删去了事务TCP协议（T/TCP）的内容。
- 新增了三章用于描述一种相对较新的传输协议——SCTP。这个可靠的面向消息的协议能够在两个端点之间提供多个流，并为多归属技术提供传输层支持。该协议最初是为了在因特网上传输电话信号而设计的，但它的一些特性可以用于许多应用。
- 新增一章描述密钥管理套接字，该套接字可用于网际协议安全（IPsec）和其他网络安全服务。
- 第2版中使用的机器及Unix变体都按最新版本更新，示例也根据机器的特性做了修改。许多情况下，修改示例是因为操作系统厂商修正了程序缺陷或者新增了特性。但读者可以想见，新的缺陷总能不时地被发现。本书中用于测试示例的机器如下：
  - 运行MacOS/X 10.2.6的Apple Power PC；
  - 运行HP-UX 11i 的HP PA-RISC；
  - 运行AIX 5.1的IBM Power PC；
  - 运行FreeBSD 4.8的Intel x86；
  - 运行Linux 2.4.7的Intel x86；
  - 运行FreeBSD 5.1的Sun SPARC；
  - 运行Solaris 9的Sun SPARC。

  这些机器的具体用法见图1-16。

本系列的第2卷（《UNIX网络编程 卷2：进程间通信》）基于本卷的内容进一步讨论了消息传递、同步、共享内存及远程过程调用。

如何使用本书

本书既可以作为网络编程的教程，也可以作为有经验的程序员的参考书。用作网络编程的教程或入门级教材时，重点应放在第二部分（第3章至第11章），然后可以看看其他感兴趣的主题。第二部分包含了TCP和UDP的基本套接字函数，以及SCTP、I/O多路复用、套接字选项和基本名字与地址的转换。所有读者都应该阅读第1章，尤其是1.4节，介绍了一些贯穿全书的包裹函数。读者可以根据自身的知识背景，选读第2章，或许还有附录A。第三部分的多数章节可以彼此独立地进行阅读。

为了方便读者把本书作为参考书，本书提供了完整的全文索引，并在最后几页总结了每个函数和结构的详细描述在正文中的哪里可以找到。为了给不按顺序阅读本书的读者提供方便，我们在全书中为相关主题提供了大量的交叉引用。

源代码与勘误

书中所有示例的源代码可以从www.unpbook.com获得[①]。学习网络编程的最好方法就是下载这些程序，对其进行修改和改进。只有这样实际编写代码才能深入理解有关概念和方法。每章末尾提供了大量的习题，大部分在附录E中给出答案。

本书的最新勘误表也可以在上述网站获取。

① 书中所有示例的源代码也可以从图灵网站（www.turingbook.com）本书网页免费注册下载。——编者注

致谢

本书第1版和第2版由W. Richard Stevens独立撰写，他不幸于1999年9月1日去世。Richard的著作体现了非常高的水准，被公认为是精练、翔实且极具可读性的艺术作品。在撰写这一修订版的过程中，我们力图保持Richard之前版本的高质量和全面性，这方面的任何不足都完全是新作者的过错。

任何作者的著作离不开家人与朋友的支持。Bill Fenner在此感谢爱妻Peggy（沙滩1/4英里赛冠军）与好友Christopher Boyd在本书撰写过程中承担了全部的家务，还要感谢朋友Jerry Winner，他的激励是无价的。同样地，Andy Rudoff要特别感谢他的妻子Ellen和两个女儿Jo、Katie自始至终的理解与鼓励。没有你们的支持，我们不可能完成本书。

思科公司的Randall Stewart提供了许多SCTP的材料，非常感谢他的巨大贡献。如果缺少了他的工作，本书就不能涵盖这一新颖而有趣的主题。

本书的审稿人给出了宝贵的反馈意见。他们发现了一些错误，指出了一些需要更多解释的地方，并对文字和代码示例提出了一些改进建议。作者在这里对如下审稿人表示感谢：James Carlson、Wu-Chang Feng、Rick Jones、Brian Kernighan、Sam Leffler、John McCann、Craig Metz、Ian Lance Taylor、David Schwartz和Gary Wright。

许多个人及其单位为本书中一些示例的测试提供了帮助，他们义务向我们出借系统、软件或为我们提供系统访问权限。

- IBM奥斯汀实验室的Jessie Haug提供了AIX系统和编译器。
- 惠普公司的Rick Jones和William Gilliam为我们提供了运行HP-UX的多个系统的访问权限。

与Addison Wesley出版社的员工合作非常愉快，他们是Noreen Regina、Kathleen Caren、Dan DePasquale和Anthony Gemellaro。要特别感谢本书的编辑Mary Franz。

为了延续Rich Stevens的风格（不过该风格与流行的风格相反），我们用James Clark编写的优秀的Groff包为本书排版，用gpic程序绘制插图（其中用到了许多由Gary Wright编写的宏），用gtbl程序生成了表格，我们为全书添加了索引，并设计了最终的版式。录入源代码时用到了Dave Hanson的loom程序和Gary Wright写的一些脚本。在生成最终索引的过程中，还用到了Jon Bentley与Brian Kernighan编写的一组awk脚本。

欢迎读者以电子邮件的方式反馈意见、提出建议或订正错误。

Bill Fenner
加利福尼亚州伍德赛德市
Andrew M. Rudoff
科罗拉多州博尔德市
2003年10月
authors@unpbook.com
http://www.unpbook.com

目　录

第一部分　简介和TCP/IP

第二部分　基本套接字编程

第三部分 高级套接字编程

第一部分

简介和TCP/IP

第1章 简 介

1.1 概述

要编写通过计算机网络通信的程序，首先要确定这些程序相互通信所用的协议（protocol）。在深入设计一个协议的细节之前，应该从高层次决断通信由哪个程序发起以及响应在何时产生。举例来说，一般认为Web服务器程序是一个长时间运行的程序（即所谓的*守护程序*，daemon），它只在响应来自网络的请求时才发送网络消息。协议的另一端是Web客户程序，如某种浏览器，与服务器进程的通信总是由客户进程发起。大多数网络应用就是按照划分成客户（client）和服务器（server）[①]来组织的。在设计网络应用[②]时，确定总是由客户发起请求往往能够简化协议和程序[③]本身。当然一些较为复杂的网络应用还需要异步回调（asynchronous callback）通信，也就是由服务器向客户发起请求消息。然而坚持采纳图1-1所示的基本客户/服务器模型的网络应用毕竟要普遍得多。

图1-1 网络应用：客户和服务器

通常客户每次只与一个服务器通信，不过以使用Web浏览器为例，我们也许在10分钟内就可以与许多不同的Web服务器通信。从服务器的角度来看，一个服务器同时与多个客户通信并不稀奇，见图1-2。本书后面将介绍若干种让一个服务器同时处理多个客户请求的方法。

可认为客户与服务器之间是通过某个网络协议通信的，但实际上，这样的通信通常涉及多个网络协议层。本书的焦点是TCP/IP协议族，也称为网际协议族。举例来说，Web客户与服务

① 本书英文原文通篇频繁使用client（客户）和server（服务器）这两个术语。实际上它们的具体含义随上下文而变化，有时指静态的源程序或可执行程序（客户程序和服务器程序），有时指动态进程（客户进程和服务器进程），有时指运行进程的主机（客户主机和服务器主机）。在不致引起混淆的前提下，我们简单地称客户进程为客户，称服务器进程为服务器。——译者注

② 应用（application）这个术语的具体含义随上下文而变化，有时指程序（应用程序），有时指进程（应用进程），有时作为名词性修饰词译为应用。本书有时把同处应用层的客户和服务器对也用应用表示，我们称之为应用系统、网络应用或应用。——译者注

③ Unix系统中程序（program）和进程（process）是在系统调用exec上衔接的。exec既可以由shell隐式调用（直接输入命令行执行程序属于这种情况），也可以在用户程序中显式调用。显式exec调用执行的程序在本书中称为新程序，以示与exec调用所在程序的区别。exec调用前后两个程序实际上在同一个进程环境下执行，不过往往使用新程序的名字来称呼这个进程。exec调用往往跟在某个fork调用之后，这样新程序将在新的进程环境中执行。客户程序和迭代服务器程序运行时通常只有一个进程，并发服务器程序运行时除主进程外，通常还为每个客户派生一个进程。程序和进程的密切关系使得两者有时相互渗透使用，不易区分。——译者注

器之间使用TCP（Transmission Control Protocol，传输控制协议）通信。TCP又转而使用IP（Internet Protocol，网际协议）通信，IP再通过某种形式的数据链路层通信。如果客户与服务器处于同一个以太网，就有图1-3所示的通信层次。

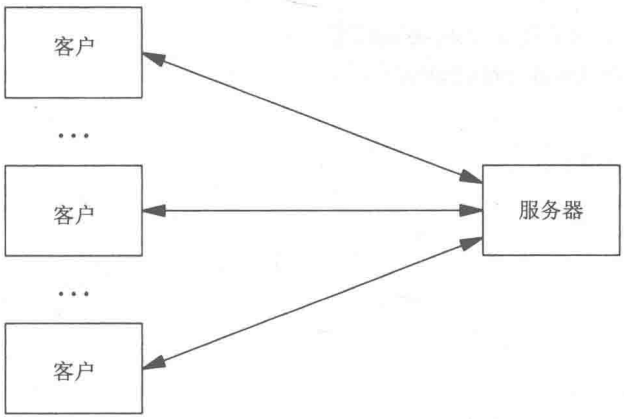


图1-2 一个服务器同时处理多个客户的请求

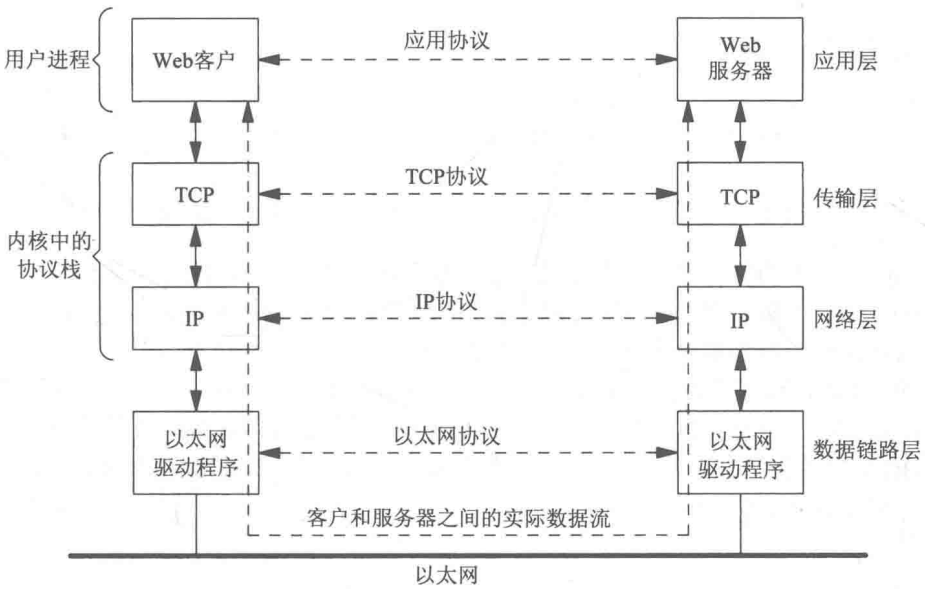


图1-3 客户与服务器使用TCP在同一个以太网中通信

尽管客户与服务器之间使用某个应用协议通信，传输层却使用TCP通信。注意，客户与服务器之间的信息流在其中一端是向下通过协议栈的，跨越网络后，在另一端则是向上通过协议栈的。另外注意，客户和服务器通常是用户进程，而TCP和IP协议通常是内核中协议栈的一部分。我们在图1-3右边标出了4个层。

本书讨论的协议不限于TCP和IP。有些客户和服务器改用UDP（User Datagram Protocol，用户数据报协议）而不是TCP，第2章将详细介绍这两个协议。此外，本书使用术语“IP”来称谓的那个协议，自20世纪80年代早期以来一直在使用，其实其正式名称是IPv4（IP version 4，IP