



LARRY L. PETERSON and BRUCE S. DAVIE

Computer Networks

A Systems Approach

EDITION

3



The Morgan Kaufmann Series in Networking

Series Editor, David Clark, M.I.T.

- Computer Networks: A Systems Approach, 3e*
Larry L. Peterson and Bruce S. Davie
- Network Architecture, Analysis, and Design, 2e*
James D. McCabe
- MPLS Network Management: MIBs, Tools, and Techniques*
Thomas D. Nadeau
- Developing IP-Based Services: Solutions for Service Providers and Vendors*
Monique Morrow and Kateel Vijayananda
- Telecommunications Law in the Internet Age*
Sharon K. Black
- Optical Networks: A Practical Perspective, 2e*
Rajiv Ramaswami and Kumar N. Sivarajan
- Internet QoS: Architectures and Mechanisms*
Zheng Wang
- TCP/IP Sockets in Java: Practical Guide for Programmers*
Michael J. Donahoo and Kenneth L. Calvert
- TCP/IP Sockets in C: Practical Guide for Programmers*
Kenneth L. Calvert and Michael J. Donahoo
- Multicast Communication: Protocols, Programming, and Applications*
Ralph Wittmann and Martina Zitterbart
- MPLS: Technology and Applications*
Bruce Davie and Yakov Rekhter
- High-Performance Communication Networks, 2e*
Jean Walrand and Pravin Varaiya
- Internetworking Multimedia*
Jon Crowcroft, Mark Handley, and Ian Wakeman
- Understanding Networked Applications: A First Course*
David G. Messerschmitt
- Integrated Management of Networked Systems: Concepts, Architectures,
and their Operational Application*
Heinz-Gerd Hegering, Sebastian Abeck, and Bernhard Neumair
- Virtual Private Networks: Making the Right Connection*
Dennis Fowler
- Networked Applications: A Guide to the New Computing Infrastructure*
David G. Messerschmitt
- Modern Cable Television Technology: Video, Voice, and Data Communications*
Walter Ciciora, James Farmer, and David Large
- Switching in IP Networks: IP Switching, Tag Switching, and Related Technologies*
Bruce S. Davie, Paul Doolan, and Yakov Rekhter
- Wide Area Network Design: Concepts and Tools for Optimization*
Robert S. Cahn
- Frame Relay Applications: Business and Technology Case Studies*
James P. Cavanagh

For further information on these books and for a list of forthcoming titles, please visit our website at <http://www.mkp.com>

THIRD EDITION

Larry L. Peterson & Bruce S. Davie

COMPUTER NETWORKS

A Systems Approach

Morgan Kaufmann Publishers

An Imprint of Elsevier

Computer Networks: A Systems Approach
Peterson & Davie

MORGAN KAUFMANN PUBLISHERS
An Imprint of Elsevier
340 Pine Street, Sixth Floor
San Francisco, CA 94104-3205, USA

© 2003 by Elsevier

Original ISBN : 0-55860-833-8

All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means—electronic or mechanical, including photocopy, recording, or any information storage and retrieval system—without permission in writing from the publisher.

First Printed in India 2003
Reprinted 2004
Reprinted 2005
Reprinted 2006

Indian Reprint ISBN-13 : 978-81-8147-206-9
Indian Reprint ISBN-10 : 81-8147-206-3

This edition is for sale in Bangladesh, Bhutan, India, Maldives, Nepal, Pakistan, Sri Lanka and designated countries in South-East Asia through Elsevier (Singapore) Pte Ltd. Sale and purchase of this book outside of these countries is unauthorised by the publisher.

Published by Elsevier, a division of Reed Elsevier India Private Limited,
Shri Pratap Udyog, 274, Captain Gaur Marg, Srinivaspuri,
New Delhi-110065 (India).

Printed and bound at Saurabh Printers Pvt. Ltd.,
A-16, Sector IV, NOIDA-201 301.

implementing the protocols running within the network. In many respects, network applications and network protocols are very similar—the way an application engages the services of the network is pretty much the same as the way a high-level protocol invokes the services of a low-level protocol. As we will see later in the section, however, there are a couple of important differences.

1.4.1 Application Programming Interface (Sockets)

The place to start when implementing a network application is the interface exported by the network. Since most network protocols are implemented in software (especially those high in the protocol stack), and nearly all computer systems implement their network protocols as part of the operating system, when we refer to the interface “exported by the network,” we are generally referring to the interface that the OS provides to its networking subsystem. This interface is often called the network *application programming interface* (API).

Although each operating system is free to define its own network API (and most have), over time certain of these APIs have become widely supported; that is, they have been ported to operating systems other than their native system. This is what has happened with the *socket interface* originally provided by the Berkeley distribution of Unix, which is now supported in virtually all popular operating systems. The advantage of industrywide support for a single API is that applications can be easily ported from one OS to another. It is important to keep in mind, however, that application programs typically interact with many parts of the OS other than the network; for example, they read and write files, fork concurrent processes, and output to the graphical display. Just because two systems support the same network API does not mean that their file system, process, or graphic interfaces are the same. Still, understanding a widely adopted API like Unix sockets gives us a good place to start.

Before describing the socket interface, it is important to keep two concerns separate in your mind. Each protocol provides a certain set of *services*, and the API provides a *syntax* by which those services can be invoked in this particular OS. The implementation is then responsible for mapping the tangible set of operations and objects defined by the API onto the abstract set of services defined by the protocol. If you have done a good job of defining the interface, then it will be possible to use the syntax of the interface to invoke the services of many different protocols. Such generality was certainly a goal of the socket interface, although it’s far from perfect.

The main abstraction of the socket interface, not surprisingly, is the *socket*. A good way to think of a socket is as the point where a local application process attaches to the network. The interface defines operations for creating a socket, attaching the socket to the network, sending/receiving messages through the socket, and closing the

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.