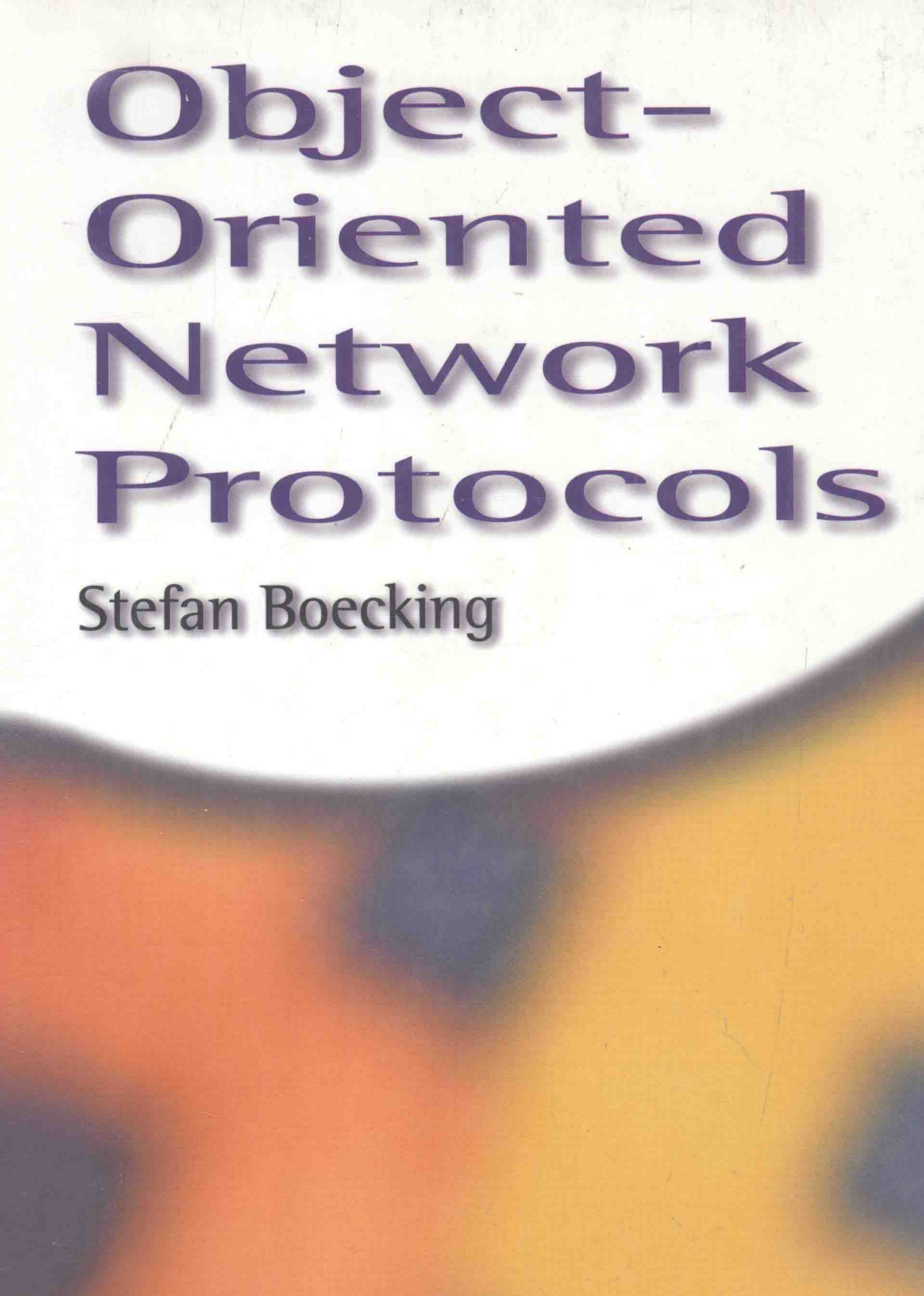
Object-Oriented Network Protocols

Stefan Boecking



Object-Oriented Network Protocols

Stefan Boecking

An imprint of PEARSON EDUCATION
Harlow, England · London · New York · Reading, Massachusetts · San Francisco · Toronto · Don Mills, Ontario · Sydney
Tokyo · Singapore · Hong Kong · Seoul · Taipei · Cape Town · Madrid · Mexico City · Amsterdam · Munich · Paris · Milan

PEARSON EDUCATION LIMITED
Head Office:
Edinburgh Gate
Harlow CM20 2JE
England
Tel: +44 (0)1279 623623
Fax: +44 (0)1279 431059

London Office:
128 Long Acre
London WC2E 9AN
Tel: +44 (0)171 477 2000
Fax: +44 (0)171 240 5771

First published in Great Britain 2000

The right of Stefan Boecking to be identified as author of this work has been asserted by him in accordance with the Copyright, Designs, and Patents Act 1988.

ISBN 0-201-17789-7

British Library Cataloguing-in-Publication Data
A CIP catalogue record for this book can be obtained from the British Library.

Library of Congress Cataloging in Publication Data
Applied for.

10 9 8 7 6 5 4 3 2 1

Typeset by Pantek Arts, Maidstone, Kent.
Printed and bound in Great Britain by Biddles Ltd, Guildford and Kings Lynn.

The Publishers' policy is to use paper manufactured from sustainable forests.

Preface

The growing diversity of network technologies such as Ethernet, FDDI and ATM, and networked applications such as file transfer, multimedia conferencing and WWW calls for a multitude of network communication services and protocols. To counter this challenge, the conventional approach of designing completely new services and protocols for each new class of application requirements and network properties is becoming more and more unsuitable and inefficient, because common functionality will be specified and implemented redundantly. For instance the Internet today comprises more than 100 network protocols and still does not satisfy all requirements of newer applications, such as the support of reliable group communications. Moreover, many of these protocols possess redundant functions, such as those for connection, error and flow control which have been designed and implemented many times.

An attractive alternative for modern network protocols is the concept of a modular and object-oriented architecture. Object orientation is a widely accepted and used software engineering technique that has been introduced successfully for the design and implementation of networked applications and middleware software such as CORBA. Surprisingly, it is scarcely used for the development of network protocols such as TCP/IP.

The benefits of the object orientation method can be used to design and implement modular network protocols that are composable, reusable and extendable. Programmers are then able to implement protocols and services by composing (rather than programming) them from a set of already specified, implemented and tested elementary protocol objects according to the application needs and the network environment. Therefore it is expected that object orientation will generally ease and accelerate the design and implementation of new services and protocols.

Furthermore, thinking of Java as one promising candidate for a *de facto* network programming language and JavaBeans as the related composition framework, a scenario becomes possible with object-oriented network protocols where network applications are downloaded including their individually customized and performance-optimized protocol stack.

In addition, we will use a case study to show that modular network protocols can provide improved performance compared to monolithic ones due to their ability to compose only those functions that are really needed to provide the requested service. This may also reassure protocol programmers, who may think that applying modular and object-oriented design or implementation methods will decrease the overall performance of their protocols.

Objectives of the book

The general intention of this book is to provide a foundation for the object-oriented design and implementation of network communication protocols.

Important for the design of network protocols is a reference framework that provides a common terminology and modeling technique for the description, specification and implementation of protocols and protocol stacks. Well known frameworks such as the ISO, B-ISDN/ATM or Internet Reference Model provide inadequate modelling support for object-oriented network protocols and modern protocols with group communications and multimedia functionality. Thus, the first objective of this book is to introduce a modern framework for systems comprising modular network communication protocols. It provides an object-oriented modelling technique and combines well-accepted and still valid concepts found in the ISO, B-ISDN/ATM and Internet world. A deliberate attempt has been made to keep the terminology used as familiar as possible to the well known reference models.

The second objective is to look into the development of object-oriented network protocols. One case study discusses how the well known Internet protocol TCP (Transmission Control Protocol) can be transformed from a monolithic protocol to a set of elementary protocol objects which can subsequently be used to compose various TCP variants. Another case study discusses a new object-oriented protocol family (Transport and Internetworking Package, TIP) which has been purposefully designed to be composable. TIP comprises a repository of protocol building blocks (object classes), an application programming interface enabling applications to compose and access their individual communication service, and a protocol operating system (similar to the X-kernel) which is required to implement dynamic composable network protocols. Furthermore, other modular systems are briefly presented to show the reader other approaches. We also show that well known protocol design principles such as the Integrated Layer Processing (ILP) and Application Level Framing (ALF), can also be applied for systems consisting of object-oriented network protocols.

The third objective of this book is to investigate whether object-oriented network protocols are able to improve communications performance and whether they can reduce the costs of protocol implementation. The performance and software measures are obtained from the object-oriented protocol family (TIP) presented as a case study.

In addition, this book provides a fundamental introduction to network technologies and networked applications to sensitize the reader for the demands on modern network communication protocols and services.

Who should read this book

This book can be used as a text for students in computer sciences and related disciplines to learn about network communication protocols such as TCP/IP from a new (object-oriented) perspective. This book is also intended for computer professionals who are interested in understanding and developing reusable protocol software.

A general knowledge of networking is not required because a basic introduction in network applications and network technologies is provided.

In addition, state-of-the-art existing network protocol mechanisms, such as those for routing, security and error control, are briefly described. This also makes the book interesting as a reference book.

Outline of the book

This book is organized into three parts: fundamentals of networks and applications, a reference framework for modular communication systems (that is, systems of modular network communication protocols), and case studies of system designs and implementations.

Part I: Fundamentals

We describe the environment of communication systems given by networks on which communication systems are based and applications for which the communication services are intended.

Chapter 1 briefly introduces the terms networks, applications and communication systems. It defines the difference between monolithic and modular communication systems.

Chapter 2 provides an overall survey of network technologies and their properties that must be taken into account by communication systems.

Chapter 3 provides an overall survey of networked applications and their requirements as regards communication systems' capabilities.

Chapter 4 gives an overview of the OSI framework comprising a reference model, and service and protocol specifications. Although the OSI services and procotols are hardly used today, the reference model and its terminology is widely used and adopted in conceptual and architectual issues.

Chapter 5 summarizes the *de facto* standard of data communication systems, the Internet protocol suite. Protocols of this suite, such as TCP and IP, typically possess a monolithic design.

Part II: Reference framework

This part introduces a framework for modular communication systems. A modular communication system comprises a collection of network communication

protocols selected to provide a particular communication service. The proposed reference framework is the Modular Communication Systems (MCS) Framework.

The intention of the MCS framework is to provide a common terminology and modeling technique for the description, design, specification and implementation of modular communication systems. The terminology used is in line with that defined in the OSI, B-ISDN/ATM, and Internet Framework, with some modifications and extensions.

One novelty introduced with the MCS framework is its object orientation. Object-oriented modelling is an efficient and well accepted software engineering method whose modelling properties of information hiding, encapsulation, abstraction and inheritance ease the design of composable, extendable and reusable systems in general, and modular communication systems in particular.

Another novelty that comes with the MCS framework is its universal communication model providing the abstraction of a multipoint connection. Its universality enables modular communication systems to compose a variety of communication services with different capabilities while being based on the same communication model.

Chapter 6 explains the fundamentals in object-oriented modelling based on objects, classes, operations and inheritance.

Chapter 7 breaks down communication systems into a collection of service objects structured in layers and planes.

Chapter 8 defines the abstraction of a service on which the interaction between objects of adjacent layers is based and defines users, providers, primitives and actions of services.

Chapter 9 explains the universal communication model of a multipoint connection, including aspects concerning communication addressing, enhancement facilities and quality of service.

For service provision, service objects co-operate in accordance with a specific protocol. Chapter 10 details the description model for protocols and summarizes typical protocol functions.

Chapter 11 describes the overall process for composing communication services. The MCS model therefore introduces a composition service that is able to compose a tailored communication service according to a given service description.

Finally, Chapter 11 discusses the communalities and differences of the MCS framework and the OSI Framework.

Part III: Case studies

This part looks into the design and implementation of modular communication systems and related components, such as protocol operating systems and application programming interfaces.

The introductory chapter 13 presents a simplified model of a network node on which a communication system's software runs. This information is provided in order to have a common understanding of which components are

involved in implementing a communication system. Furthermore, we discuss general design objectives that we consider to be important for the development of modern communication systems. This chapter also gives some guidelines on how to construct modular communication systems and design protocols in an object-oriented way.

Chapter 14 discusses a possible modularization of the TCP protocol to show ways in which existing monolithic protocols can migrate to a modular and object-oriented future.

Chapter 15 gives an overview of the system architecture of TIP and other modular communication systems, for example HOPS, F-CSS, Micro Protocols and Da CaPo. To ease the assessment of these systems, their philosophy and architecture are described uniquely using the MCS framework.

Chapter 16 presents TIP's protocol operating system which has been designed to support in particular the implementation of dynamic configurable protocol software.

Chapter 17 studies TIP's application programming interface (API), which provides access from an application program to TIP's configuration and communication services. It also gives an overview of other API approaches.

Chapter 18 presents measurement results as regards the overall performance of TIP and benchmarks between TIP and TCP/IP, which finally answers the questions whether modular communication systems are able to improve communication performance compared to monolithic ones. Furthermore, the impact of modularization and configuration is investigated.

The last chapter evaluates specific software engineering aspects of TIP to prove that modular communication systems are able to reduce the costs of protocol implementation.

Appendices

The appendices contain details and more information about topics presented in this book. Appendix A maps the Internet terminology with our reference framework. Appendix B contains network details. Appendix C explains how TCP operates.

Acknowledgements

This book would not have been possible without the help and encouragement of my family, friends, colleagues and company. It has evolved from my post-doctoral thesis, which has been supervised by Professor Eike Jessen and Professor Heinz-Gerd Hegering, both from the Technical University of Munich. Their suggestions and expertise were invaluable and I want to thank both of them for the help that shepherded me through the post-doctoral university lecturing qualification. I would also like to thank Professor Martina Zitterbart from the Technical University of Braunschweig for evaluating my post-doctoral thesis.

I also want to thank the people who reviewed drafts of various chapters: Professor Ernst Biersack from Eurecom in France, and Bernd Petri, Michael Munzert, Lutz Kasselmann and Alexander Pilger, all from Siemens, Munich.

This book contains a case study of a running object-oriented communication system, called TIP. TIP has been implemented, improved and evaluated by colleagues from Siemens, the Technical University of Braunschweig, the Technical University of Berlin, and Eurecom. Especially, I want to thank my colleagues at Siemens, Per Vindeby, Vera Seidel and Alexander Pilger, who designed and developed several versions of TIP with me over the past years. Also thanks to Professor Martina Zitterbart and her students Ralph Wittmann, Florian Schröcker, and Marcus Briesen, who helped to implement and improve TIP. Professor Adam Wolisz and his students from the Technical University of Berlin make use of TIP for educational and training purposes in their classes and have given valuable feedback. Professor Ernst Biersack and his students from Eurecom have evaluated TIP for a Java/JavaBeans-based implementation.

I would also like to thank Siemens for supporting my research over the past years in the area of object-oriented network protocols.

Contents

Part I

Fundamentals

1. Introduction
2. Networks
3. Applications
4. The OSI framework
5. Internet protocol suite

This part describes well-known networks from Ethernet to asynchronous transfer mode (ATM) and popular applications from file transfer to video- on-demand in order to get an idea of the broad diversity of their properties and requirements.

The listed networks and applications are those which we have investigated and considered for the design of our reference framework for modular communication systems described in the next part.

It is important for the design of a reference framework for (modular) communication systems to consider as much as possible of the existing and expected application requirements and network properties, otherwise the framework may be used only to design communication systems for a limited scope of networks and applications. For instance, if applications that require multipoint communications support are not considered, the framework will probably provide only a point-to-point communication model. Or, if networks with resource reservation are left out, the quality of service model may not provide performance guarantee support.

This part starts with an introductory chapter defining the terms application, network, and communication system as used throughout this book. After presenting a variety of networks and applications, this part will end with a brief overview of the open systems interconnection (OSI) framework and the Internet protocol suite, which are fundamental for the subsequent discussions of our framework for object-oriented network protocols (Part II) and the case studies (Part III).