
Preface

THIS BOOK IS INTENDED TO FILL a long-standing gap in the documentation and literature of the Java™ programming language and platform, by providing fundamental and in-depth coverage of TCP/IP and UDP networking from the point of view of the Java API, and by discussing advanced networking programming techniques.¹ The new I/O and networking features introduced in JDK 1.4 provide further justification for the appearance of this text. Much of the information in this book is either absent from or incorrectly specified in the Java documentation and books by other hands, as I have noted throughout.

In writing this book, I have drawn on nearly twenty years' experience in network programming, over a great variety of protocols, APIs, and languages, on a number of platforms (many now extinct), and on networks ranging in size from an Ethernet a few inches in length, to a corporate WAN between cities thousands of miles apart, to the immense geographic spread of the Internet.

This book covers both 'traditional' Java stream-based I/O and so-called 'new I/O' based on buffers and channels, supporting non-blocking I/O and multiplexing, for both 'plain' and secure sockets, specifically including non-blocking TLS/SSL and GSS-API.

Server and client architectures, using both blocking and non-blocking I/O schemes, are discussed and analysed from the point of view of scalability and with a particular emphasis on performance analysis.

An extensive list of TCP/IP platform dependencies, not documented in Java, is provided, along with a handy reference to the various states a TCP/IP port can assume.

1. Sun, Java, and many Java-related terms and acronyms are trademarks of Sun Microsystems Incorporated, Santa Clara, California. These and all other trademarks referred to in this book remain property of their respective owners.

ABOUT THE BOOK

Audience

I have assumed a competent reader familiar with the fundamentals of the Java programming language, specifically the concepts of *class*, *object*, *interface*, *method*, *parameter*, *argument*, *result*, and *exception*; with the basic principles of object-oriented programming: *inheritance* and *polymorphism*; and with the standard Java I/O, utility, and exception classes.

I have also assumed a reader who is able to digest short passages of simple Java code without requiring every line explained, and to turn English prose into Java code without requiring a code sample at every turn. A very basic knowledge of TCP programming with clients and servers is assumed, although I have provided a brief review. Finally, I assume that the reader either knows about the Internet, hosts, and routers, or has the initiative and the resources to look them up.

I have used some of the more standardized vocabulary of design patterns, as first seen in Gamma *et al.*, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995, specifically the terms *adapter*, *delegate*, *facade*, and *factory*, which are now in such common use as not to require explanation. I have also used UML sequence diagrams without definition or comment, as these are fairly self-explanatory.

Scope

The book covers TCP downwards from the Java API, through socket options and buffers, to the TCP segment level, including the connection and termination sequences, RST segments, and—to a small extent—windowing, but excluding sequence numbering, pacing, acknowledgements, and retries.

Similarly, it covers UDP downwards from the Java API, through socket options and buffers, to the UDP datagram level, including unicast, broadcast, and multicast, and including material on reliable UDP, erasure codes, and higher-level multicasting protocols.

I have paid particular attention to the neglected and widely misunderstood topic of multi-homed hosts, particularly in relation to UDP unicast, broadcast, and multicast, where multi-homing presents special difficulties.

The TCP, UDP, and TLS/SSL protocols are all covered both in blocking and non-blocking mode, via both traditional Java streams and channel- and buffer-oriented ‘NIO’ (new I/O). Secure sockets via SSL and TLS are covered in detail, and the JGSS-API is discussed as an alternative.

I have devoted an entire chapter to a reduction-to-practice of the JDK 1.5 SSLEngine, with sample code for a complete and correct SSLEngineManager, making this bizarre apparition actually useable for writing non-blocking SSL servers and clients.

The organization of the book is described in section 1.2.

Exclusions

The book excludes `IP` at the packet level altogether, as well as associated protocols such as `ICMP`, `ARP`, `RARP`, `DHCP`, *etc.*, although `IGMP` does appear fleetingly in the discussion of multicasting. These topics are definitively covered in Stevens & Wright, *TCP/IP Illustrated*, Volumes I and II, Addison-Wesley, 1994–5, whose completeness and authoritativeness I have not attempted to duplicate.

I have deliberately omitted any mention of the defunct 7-layer `OSI` Reference Model,² into which `TCP/IP` cannot be shoehorned.

I have excluded all higher-level protocols such as `HTTP`, `HTTPS`, and `FTP`. I have also excluded `J2EE` in its entirety, as well as `Java RMI` (Remote Method Invocation), with the exception of `RMI` socket factories which present special, undocumented difficulties. Kathleen McNiff and I have described `Java RMI` in detail in our book *java.rmi: The Guide to Remote Method Invocation*, Addison-Wesley 2001.³

I have resisted without apology the recent tendency to re-present all of computer science as design patterns, even in Chapter 12, ‘Server and client models’, for which design patterns do exist. The relevant parts of `Java` and the `Java Class Library` themselves constitute design patterns which subsume many existing patterns for network programming.

This book is about networking, and so is the sample code. `Java` program code which is not directly relevant to network programming does not appear. Not a line of `AWT` or `Swing` code is to be found in these pages, nor are screen shots, console outputs, or examples of streaming audio-visuals or 3D animations. Nor have I presented the ‘complete source code’ for some arbitrary application of limited relevance.

ACKNOWLEDGEMENTS

I am primarily indebted to the many people who researched and developed the `TCP/IP` protocol suite over several decades, and whose names appear in the various `IETF` formal standards and RFCs which define the suite: some of these are listed in the bibliography.

Any serious writer on `TCP` and `UDP` owes practically everything to the late W. Richard Stevens, with whom I was privileged to exchange a few e-mails. Stevens documented the entire protocol suite, both the specification and the `BSD 4.4` implementation, in his *TCP/IP Illustrated*, 3 volumes, and described the Berkeley Sockets API in all its gruesome details in his *Unix Network Programming*, 2 volumes. These are now fundamental references for anyone who really wants to understand `IP` network programming in any language.

2. for which see e.g. Piscitello & Chapin, *Open Systems Networking: OSI @ TCP/IP*.

3. Much of the present chapter on firewalls first appeared there, and is used by permission.

This book started life in 1993 as a 25-page paper written in collaboration with my brother and colleague David Pitt: the paper was privately distributed to employees and clients, and has subsequently turned up in all sorts of surprising places.

Several anonymous reviewers contributed significantly to the final form and content of this book. All errors however remain mine.

My thanks go to Sun Microsystems Inc. for producing Java and supplying it free of charge, and to Sun Microsystems Ltd, Melbourne, Australia, for providing Solaris and Linux testing facilities.

Thanks also to my long-standing colleague Neil Belford for advice, assistance, and encouragement. Finally, thanks to Tilly Stoové and all the Pitt family for their understanding and support during the writing of this book.

Esmond Pitt, Melbourne, June 2005.



<http://www.springer.com/978-1-84628-030-6>

Fundamental Networking in Java

Pitt, E.

2006, XVIII, 382 p. 23 illus., Hardcover

ISBN: 978-1-84628-030-6