



图灵原版计算机科学系列



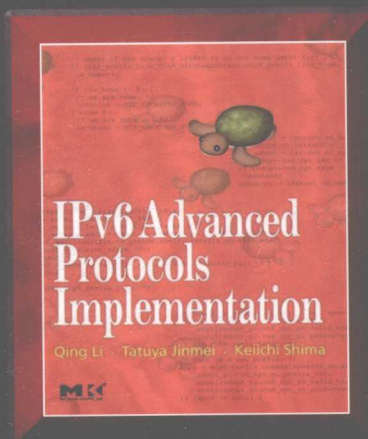
IPv6 Advanced Protocols
Implementation

IPv6详解

卷2：高级协议实现

(英文版)

[美] Qing Li
[日] Tatuya Jinmei 著
[日] Keiichi Shima



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵原版计算机科学系列

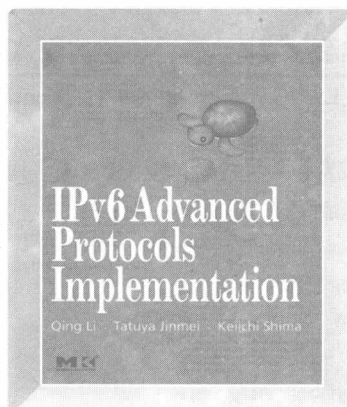
IPv6 Advanced Protocols
Implementation

IPv6详解

卷2：高级协议实现

(英文版)

[美] Qing Li
[日] Tatuya Jinmei 著
[日] Keiichi Shima



人民邮电出版社
北京

图书在版编目(CIP)数据

IPv6详解. 第2卷. 高级协议实现 = IPv6 Advanced
Protocols Implementation: 英文/ (美) 李清等著. —北
京: 人民邮电出版社, 2009.2

(图灵原版计算机科学系列)

ISBN 978-7-115-19519-7

I. I… II. 李… III. 计算机网络—传输控制协议—英
文 IV. TN915.04

中国版本图书馆CIP数据核字 (2008) 第201262号

内 容 提 要

本书全面讲解IPv6及相关协议实现的事实标准KAME, 揭示了KAME IPv6协议栈的所有细节, 对每行代码到底做了什么, 以及为什么要这样设计都进行了解释。全书共分6章, 分别介绍IPv6单播路由选择协议、IPv6多播技术、IPv6的DNS DHCPv6、移动IPv6、IPv6与IP安全。书中每章都包含两个主要部分, 第一部分是相关规范的综述, 第二部分则逐行代码地描述和分析实际的实现。

本书是IPv6的权威参考书, 适合网络设计和开发人员阅读。此外, 本书还适合作为高校相关专业网络课程的教学参考书。

图灵原版计算机科学系列

IPv6详解, 卷2: 高级协议实现 (英文版)

◆ 著 [美]Qing Li [日]Tatuya Jinmei [日]Keiichi Shima
责任编辑 杨海玲

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京铭成印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 62.75

字数: 1205千字 2009年2月第1版

印数: 1-2 000册 2009年2月北京第1次印刷

著作权合同登记号 图字: 01-2008-5824号

ISBN 978-7-115-19519-7/TP

定价: 128.00元

读者服务热线: (010) 88593802 印装质量热线: (010) 67129223

反盗版热线: (010) 67171154

版 权 声 明

IPv6 Advanced Protocols Implementation, First Edition by Qing Li, Tatuya Jinmei, and Keiichi Shima,
ISBN: 978-0-12-370479-5.

Copyright © 2007 by Elsevier. All rights reserved.

Authorized English language reprint edition published by the Proprietor.

ISBN: 978-981-272-318-5

Copyright © 2009 by Elsevier (Singapore) Pte Ltd. All rights reserved.

Elsevier (Singapore) Pte Ltd.

3 Killiney Road

#08-01 Winsland House I

Singapore 239519

Tel: (65)6349-0200

Fax: (65)6733-1817

First Published 2009

2009年初版

Printed in China by POSTS & TELECOM PRESS under special arrangement with Elsevier (Singapore) Pte Ltd. This edition is authorized for sale in China only, excluding Hong Kong SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书英文影印版由Elsevier (Singapore) Pte Ltd. 授权人民邮电出版社在中华人民共和国境内（不包括香港特别行政区和台湾地区）出版发行。未经许可之出口，视为违反著作权法，将受法律之制裁。

*To Huaying, Jane and Adalia
in Him*
—Qing Li

*To my colleagues at KAME: working with you talented geeks was an exciting experience and
has even made this derivative project possible.*
—Tatuya Jinmei

*To all KAME developers, all people who developed the Internet, and all people who
will develop the future Internet.*
—Keiichi Shima

Preface

This book is the second installment of our series detailing IPv6 and related protocols through the KAME implementation. KAME is a widely deployed de facto reference implementation for IPv6 and IP security protocols developed on multiple variants of the BSD operating systems.

The first installment of this series is titled *IPv6 Core Protocols Implementation*, which is referred to as the *Core Protocols* book below, and it focuses on the fundamentals of IPv6 and the essential protocols that are supported by most implementations. These essential protocols operate in IPv6-capable devices, large or small. Our *Core Protocols* book also describes IPv6 implication on higher layer protocols, such as TCP and UDP, and covers IPv6 related application programming interfaces.

This second book discusses those protocols that are found in more capable IPv6 devices, are commonly deployed in more complex IPv6 network environments, or are not specific to IPv6 but are extended to support IPv6. Specifically, this book engages the readers in more advanced topics, such as routing, multicasting, DNS, mobility, and security.

The general structure and style of this book is the same as that of the *Core Protocols* book; each chapter begins with a summary of the relevant specifications followed by line-by-line code description and analysis of the actual implementation.

We hope to help the readers establish a solid and empirical understanding of IPv6 with our book series. Our two books together cover a wide spectrum of the IPv6 technology and are paralleled by none.

This book consists of the following chapters:

- Chapter 1 (“IPv6 Unicast Routing Protocols”) discusses general routing concepts and the fundamentals of various types of unicast routing protocols. This chapter details RIPng, a simple routing protocol for IPv6, and summarizes IPv6-specific extensions defined for the BGP4+ and OSPFv3 routing protocols. Comparisons are made among these protocols in regards to protocol complexity, stability, and the operational issues and solutions

offered by each. This chapter also provides the necessary background to implement IPv6 routing protocols on BSD variants through descriptions of the routing API for IPv6 and code narrations of KAME's RIPng implementation, the **route6d** daemon. This chapter concludes with configuration examples of **route6d** for some typical scenarios.

- Chapter 2 ("IPv6 Multicasting") discusses details about IPv6 multicasting, especially on multicast routing mechanisms. It first provides the basics of a host-to-router protocol and multicast routing protocols, specifically the Multicast Listener Discovery protocol version 1 (MLDv1) and Protocol Independent Multicast (PIM), focusing on IPv6 specific issues. The latter part of this chapter describes the KAME kernel implementation of MLDv1 and IPv6 multicast forwarding.
- Chapter 3 ("DNS for IPv6") describes IPv6 extensions to the DNS (Domain Name System) protocol specification and implementation. It begins with a general description of the DNS protocol and its extensions that support IPv6. It then describes KAME's DNS client (called a *resolver*) implementation, and highlights the support for IPv6. This section also gives a complete view of the `getaddrinfo()` library function, which was partially described in the *Core Protocols* book. The latter half of this chapter shows how to operate the BIND9 DNS server to support IPv6 with notes about common pitfalls and issues specific to IPv6-related operations.
- Chapter 4 ("DHCPv6") details DHCPv6 (Dynamic Host Configuration Protocol for IPv6) both on the protocol specification and on KAME's implementation. Although the basic concept of the protocol is largely derived from DHCP for IPv4 (DHCPv4), DHCPv6 has introduced various improvements in its design and the expected usage model differs from that of DHCPv4; this chapter clarifies such major differences. The implementation descriptions cover all protocol functionalities, that is, clients, servers, and relay agents, and will provide an in-depth understanding of how the protocol works. This chapter also provides how to operate DHCPv6 with the KAME implementation for some common usage scenarios.
- Chapter 5 ("Mobile IPv6") covers the IPv6 host mobility protocol known as Mobile IPv6. The chapter begins with a basic description of Mobile IPv6, and then details protocol specifications and data structures. The actual implementation is discussed in the middle of the chapter. The KAME Mobile IPv6 implementation supports both home agent and mobile node functions. The code description section will discuss all data structures and functions in detail. This chapter also provides a brief instruction of Mobile IPv6 operation with sample configuration files using the KAME Mobile IPv6 implementation at the end of the chapter.
- Chapter 6 ("IPv6 and IP Security") begins with an introduction of the IPsec protocols and the concept of keying in the context of the Internet Key Exchange (IKE) protocol. The remainder of this chapter then focuses on describing the popular **racoona** IKE daemon. Its configuration and operation are thoroughly explained. This chapter concludes with some practical examples of using **racoona**. Unlike other chapters, this chapter does not provide any code description because the basic mechanism of IP Security and most of its implementation are not specific to IPv6; including non-IPv6 specific code description would change the main objective of this book.

Intended Audience

In general, this book is intended for the same class of readers as was the *Core Protocols* book: developers implementing IPv6 and related protocols, and students who are going to start a project on these protocols, especially on top of or using the KAME/BSD implementation. Unlike the *Core Protocols* book, however, this book discusses more advanced topics, such as protocols that have been standardized relatively recently, so it can also be used as a reference to these protocols per se; DHCPv6 and Mobile IPv6 are two specific examples of this.

As in the *Core Protocols* book, it is assumed that readers are fluent in the C programming language. In addition, this book assumes knowledge of the basic notions of IPv6 and related protocols described in the *Core Protocols* book, though other references within this book will help those who cannot refer to the *Core Protocols* book to understand the contents. Chapters 2 and 5 also require general understanding of the BSD kernel implementation.

Unlike the *Core Protocols* book, each chapter of this book is quite independent; although there are several cross references among the chapters, readers can generally start from any chapter based on their interest.

Typographical Conventions

This book adopts the same typographical conventions as those for the *Core Protocols* book, which is summarized as follows:

Variable, function, or structure names, structure members, and programming language keywords are represented in a constant-width font when referred to in the code descriptions. Function names are in a constant-width font followed by parentheses, as in `ip6_mforward()`, and structure names are in a constant-width font followed by braces, as in `ip6_mh{}`.

Program names are displayed in bold fonts, as in **route6d**. The command line input and the output of a program are displayed in a constant-width font.

Accompanying CD-ROM

This book comes with two CD-ROMs^①. The first CD-ROM is an ISO image of FreeBSD4.8-RELEASE, which is the base operating system covered in Chapters 1, 2, 3, and 6. It is a bootable CD-ROM and includes installation files. The installation procedure is started by turning on the computer with the CD-ROM loaded. The detailed installation procedure can be found in the `INSTALL.TXT` file located in the root directory of the CD-ROM.

Similarly, the second CD-ROM is a bootable ISO image of FreeBSD4.9-RELEASE, which is the base operating system covered in Chapter 5.

Note: FreeBSD 4.8 and 4.9 RELEASEs are known to have several security flaws and are no longer supported by the FreeBSD project. Therefore, these systems should only be used for reference on learning the KAME implementation as part of reading this book. It is not advisable to use these versions of FreeBSD in a production environment connected to the Internet.

① 本影印版出版时并未随书附带CD-ROM，文中提到的CD-ROM中的FreeBSD4.8-RELEASE和FreeBSD4.9-RELEASE可以从<http://www.freebsd.org/releases/4.8R/installation.html>和<http://www.freebsd.org/releases/4.9R/installation.html>下载，KAME源代码可以从<http://www.kame.net/>下载，以上内容也可从图灵网站（www.turingbook.com）直接下载。

The first CD-ROM also contains the KAME source code discussed in this book. It is accessed via the `appendix` directory located at the root directory, which has two subdirectories, `kame-snap` and `rtadd6`.

The `kame-snap` subdirectory contains the following archive files:

- `kame-20030421-freebsd48-snap.tgz`
A KAME snapshot for FreeBSD 4.8 taken on April 21, 2003.
- `kame-20040712-freebsd49-snap.tgz`
A KAME snapshot for FreeBSD 4.9 taken on July 12, 2004. This is referred to in Chapter 5, and should be used with the FreeBSD 4.9 system contained in the second CD-ROM.
- `kame-dhcp6-20050509.tgz`
KAME's DHCPv6 implementation included in a KAME snapshot taken on May 9, 2005, which is referred to in Chapter 4.

To install the KAME snapshot, unpack the archive, go down to the top level directory named `kame` (which is also referred to as `/${KAME}` throughout this book), and see the `INSTALL` file located in the directory. For those who have the *Core Protocols* book, its Chapter 1 provides a more detailed description of the usage. Chapter 4 of this book explains how to install the DHCPv6 implementation.

The other subdirectory, `rtadd6`, contains the source code of the `rtadd6` program referred to in Chapter 1, which was newly written for this book.

Source Code Copyright

This book presents many parts of the source code developed by the KAME project and external contributors. It also refers to system header files that are part of the FreeBSD distributions. All of the source code has copyright notices, which are available in the copy of the code contained in the CD-ROM discs.

Reporting Errors and Errata Page

The authors are happy to receive error reports on the content of this book, and plan to provide an error correction page on the Internet. It will be available at the following web page: <http://books.elsevier.com/companions/9780123704795>.

Acknowledgments

The authors, first and foremost, thank all KAME developers. As in our first book, this book is half-filled with the KAME source code, which means they are the shadow authors of this book.

We are also deeply indebted to technical reviewers who read selected chapters of this book and provided many valuable comments and suggestions, as well as error corrections: Mark Andrews, David Borman, Francis Dupont, Daniel Hartmeier, Jeffrey Hsu, Akira Kato, T. J. Kniveton, Ted Lemon, Tsuyoshi Momose, George Neville-Neil, Yasuhiro Ohara, Shawn Routhier, Shoichi Sakane, Shigeya Suzuki, Shinsuke Suzuki, Christian Vogt, and Carl Williams. As with our first book, reviewing this book required thorough knowledge of the related protocol specifications, as well as high level programming skills. We knew very few people have such talents, and we were very lucky to have the world's best reviewer team.

The book cover is based on the well-known KAME turtle image, which was designated as a project mascot, and was designed by Manabu Higashida and Chizunu Higashida.

Next, we would like to thank our editors Rick Adams, Rachel Roumeliotis, Dawnmarie Simpson, and the editorial staff at Morgan Kaufmann/Elsevier for their continuing patience and encouragement over the three and a half years of this project.

Finally, we are grateful to Gary R. Wright and W. Richard Stevens. Their work inspired us to start our own project and kept us confident about the value of this work.

LI, Qing—I would like to thank Rick Adams for his keen understanding of the importance of this book, as it fulfills a market void. His prompt acceptance of my book proposal has been an invaluable motivation. I want to thank my wife Huaying Cheng for her understanding and support of me during this book project. I would like to thank VMware Inc. for its donation of a single license for the VMware Workstation 4 software. I would also like to thank MKS Software for its donation of a single license for the MKS Toolkit for Enterprise Developers version 8.6 software.

JINMEI, Tatuya—I would like to thank my current and former managers at Toshiba for their approval and support of this work: Yukio Kamatani, Toshio Murai, Yasuhiro Katsube, and Atsushi Inoue. My thanks also go to my “supervisors” at the WIDE project, Jun Murai and Hiroshi Esaki.

SHIMA, Keiichi—I thank all of the people who worked hard to publish this book and those who supported this work, especially my manager Eiiti Wada at Internet Initiative Japan, Inc. Also my thanks go to all operators, engineers, and researchers of the Internet.

About the Authors

Li, Qing is a senior architect at Blue Coat Systems, Inc. leading the design and development efforts of the next-generation IPv6 enabled secure proxy appliances. Prior to joining Blue Coat Systems, Qing spent 8 years at Wind River Systems, Inc. as a senior architect in the Networks Business Unit, where he was the lead architect of Wind River's embedded IPv6 products since the IPv6 program inception at the beginning of 2000. Qing holds multiple U.S. patents. Qing is a contributing author of the book titled *Handbook of Networked and Embedded Control Systems* published in June of 2005 by Springer-Verlag. He is also the author of the embedded systems development book titled *Real-Time Concepts for Embedded Systems* published in April of 2003 by CMP Books. Qing participates in open source development projects and is an active FreeBSD src committer.

Jinmei, Tatuya, PhD, is a research scientist at Corporate Research & Development Center, Toshiba Corporation (Jinmei is his family name, which he prefers is presented first according to the Japanese convention). He had been a core developer of the KAME project since the launch of the project through its conclusion. In 2003, he received his PhD degree from Keio University, Japan, based on his work at KAME. He also coauthored three RFCs on IPv6 through his activity in KAME. His research interests spread over various fields of the Internet and IPv6, including routing, DNS, and multicasting.

Shima, Keiichi is a senior researcher at Internet Initiative Japan Inc. His research area is IPv6 and IPv6 mobility. He was a core developer of the KAME project from 2001 to the end of the project and developed Mobile IPv6/NEMO Basic Support protocol stack. He is now working on the new mobility stack (the SHISA stack) for BSD operating systems that is a completely restructured mobility stack.

Contents

1

IPv6 Unicast Routing Protocols 1

- 1.1 Introduction 1
- 1.2 Overview of Routing Concepts 2
- 1.3 Overview of Vector-based Algorithms and Link-State Algorithm 5
 - 1.3.1 Distance-Vector Algorithm 5
 - 1.3.2 Path-Vector Algorithm 7
 - 1.3.3 Link-State Algorithm 7
- 1.4 Introduction to RIPng 10
 - 1.4.1 RIPng Message Formats 11
 - 1.4.2 RIPng Operation 14
 - 1.4.3 Problems with RIPng 15
- 1.5 Introduction to BGP4+ 17
 - 1.5.1 BGP4+ Operation 19
 - 1.5.2 BGP4+ Messages 21
 - 1.5.3 Path Attributes 27
 - 1.5.4 IPv6 Extensions for BGP4+ 29
 - 1.5.5 BGP4+ Route Selection Process 31

- 1.6 Introduction to OSPFv3 33
 - 1.6.1 Router Adjacency and LSDB Synchronization 33
 - 1.6.2 Area Types and Router Classification 35
 - 1.6.3 Link State Advertisement and LSA Types 35
 - 1.6.4 LSA Formats 37
 - 1.6.5 OSPF Tree Construction and Route Computation 46
- 1.7 Code Introduction 49
- 1.8 IPv6 Routing Table in the BSD Kernel 50
 - 1.8.1 Scope Zone Representation in the Routing Table 53
- 1.9 Routing API 55
 - 1.9.1 Routing Sockets 55
 - 1.9.2 Dumping Routing Table via `sysctl()` 62
- 1.10 Overview of **route6d** Daemon 65
- 1.11 Common Data Structures, Routines and Global Variables 65
 - 1.11.1 Structures for RIPng Messages 65
 - 1.11.2 **route6d**'s Routing Table 67
 - 1.11.3 Structures for Local Interfaces 68
 - 1.11.4 **route6d** Route Filter Entry 70
 - 1.11.5 Subroutines and Global Variables 72
- 1.12 Interface Configuration 74
 - 1.12.1 `ifconfig()` Function 74
 - 1.12.2 `ifconfig1()` Function 77
- 1.13 RIPng Protocol Operation 81
 - 1.13.1 `sendrequest()` Function 82
 - 1.13.2 `riprecv()` Function 83
 - 1.13.3 `riprequest()` Function 96
 - 1.13.4 `ripsend()` Function 97
 - 1.13.5 `ripalarm()` Function 104
- 1.14 Routing Operation Using **route6d** 105
 - 1.14.1 A Leaf Network 105
 - 1.14.2 A Simple Loop Network 108
 - 1.14.3 A Hierarchical Network 111

2

IPv6 Multicasting 113

- 2.1 Introduction 113
- 2.2 IPv6 Multicast Address to Layer-2 Multicast Address Mapping 114

| | | |
|-------|---|-----|
| 2.3 | Multicast Listener Discovery Protocol | 114 |
| 2.3.1 | MLD Protocol Message Format | 115 |
| 2.3.2 | Router Alert Option | 116 |
| 2.3.3 | Source Address Selection | 116 |
| 2.3.4 | Destination Address Selection | 116 |
| 2.3.5 | MLD Querier | 116 |
| 2.3.6 | Operational Variables | 117 |
| 2.3.7 | MLD Join Process | 118 |
| 2.3.8 | MLD Leave Process | 119 |
| 2.4 | Multicast Routing Fundamentals | 120 |
| 2.4.1 | Reverse Path Forwarding | 120 |
| 2.4.2 | Multicast Routing Models | 121 |
| 2.4.3 | Protocol Independent Multicast | 125 |
| 2.4.4 | IPv6 Specific Issues about PIM | 128 |
| 2.4.5 | IPv6 Multicast Future—MLDv2 and SSM | 130 |
| 2.5 | Code Introduction | 131 |
| 2.6 | MLD Implementation | 133 |
| 2.6.1 | Types and Structures | 133 |
| 2.6.2 | mld6_init() Function | 136 |
| 2.6.3 | Joining a Group: mld6_start_listening() Function | 137 |
| 2.6.4 | Leaving a Group: mld6_stop_listening() Function | 139 |
| 2.6.5 | Input Processing: mld6_input() Function | 140 |
| 2.6.6 | mld6_fasttimeo() Function | 144 |
| 2.6.7 | mld6_sendpkt() Function | 146 |
| 2.6.8 | mld_allocbuf() Function | 149 |
| 2.7 | IPv6 Multicast Interface: mif6{} Structure | 150 |
| 2.8 | IPv6 Multicast Routing API | 152 |
| 2.8.1 | ip6_mrouter_set() Function | 152 |
| 2.8.2 | ip6_mrouter_init() Function | 155 |
| 2.8.3 | ip6_mrouter_get() Function | 156 |
| 2.8.4 | set_pim6() Function | 157 |
| 2.8.5 | add_m6if() Function | 157 |
| 2.8.6 | del_m6if() Function | 160 |
| 2.8.7 | ip6_mrouter_done() Function | 161 |
| 2.8.8 | mrt6_ioctl() Function | 164 |
| 2.8.9 | get_mif6_cnt() Function | 164 |

- 2.9 IPv6 Multicast Forwarding Cache 165
 - 2.9.1 `add_m6fc()` Function 166
 - 2.9.2 `del_m6fc()` Function 171
 - 2.9.3 `expire_upcalls()` Function 172
 - 2.9.4 `get_sg_cnt()` Function 173
- 2.10 IPv6 Multicast Forwarding 174
 - 2.10.1 `ip6_mforward()` Function 175
 - 2.10.2 `ip6_mdq()` Function 183
 - 2.10.3 `phyint_send()` Function 189
 - 2.10.4 `register_send()` Function 192
 - 2.10.5 `socket_send()` Function 194
 - 2.10.6 `pim6_input()` Function 195
- 2.11 IPv6 Multicast Operation 202
 - 2.11.1 **ifmcstat** Command 202
 - 2.11.2 Enable IPv6 Multicast Routing 203
 - 2.11.3 **pim6dd** and **pim6sd** Routing Daemons 203
 - 2.11.4 **pim6stat** Output 203
 - 2.11.5 **netstat** Command 206

3

DNS for IPv6 207

- 3.1 Introduction 207
- 3.2 Basics of DNS Definitions and Protocols 208
 - 3.2.1 DNS, Domains, and Zones 208
 - 3.2.2 Resource Records and Zone Files 210
 - 3.2.3 DNS Transaction and Packet Format 212
 - 3.2.4 Name Resolution and Caching 214
- 3.3 IPv6-Related Topics about DNS 217
 - 3.3.1 AAAA Resource Record 217
 - 3.3.2 DNS Reverse Tree for IPv6 217
 - 3.3.3 IPv6 Transport for DNS 219
 - 3.3.4 Packet Size Issue and EDNS0 219
 - 3.3.5 Misbehaving DNS Servers against AAAA 222
 - 3.3.6 Obsolete Standards 225
- 3.4 Implementation of IPv6 DNS Resolver 226
 - 3.4.1 `_dns_getaddrinfo()` Function 229
 - 3.4.2 `getanswer()` Function 235
 - 3.4.3 `res_queryN()` Function 243
 - 3.4.4 Resolver State Structure 245

- 3.4.5 `res_init()` Function 248
- 3.4.6 `res_send()` Function 250
- 3.4.7 IPv6 Reverse Lookup: `_dns_ghbyaddr()` Function 260
- 3.5 IPv6 DNS Operation with BIND 264
 - 3.5.1 Overview of BIND9 265
 - 3.5.2 Getting BIND9 266
 - 3.5.3 Building and Installing BIND9 266
 - 3.5.4 Configuring BIND9 for IPv6 Operation 267
 - 3.5.5 Implementation-Specific Notes 274
 - 3.5.6 Complete Configuration Example 282
 - 3.5.7 **dig** and **host** Utilities 286

4 DHCPv6 289

- 4.1 Introduction 289
- 4.2 Overview of the DHCPv6 Protocol 290
 - 4.2.1 Cases for DHCPv6 290
 - 4.2.2 Definitions about DHCPv6 293
 - 4.2.3 DHCPv6 Message Exchanges 297
 - 4.2.4 Summary of DHCPv6 Options 310
 - 4.2.5 Interaction with Neighbor Discovery 319
 - 4.2.6 Comparison to DHCPv4 319
- 4.3 Code Introduction 320
 - 4.3.1 Common Data Structures and Routines 320
- 4.4 Client Implementation 326
 - 4.4.1 Client-Specific Data Structures 328
 - 4.4.2 `client6_mainloop()` Function 332
 - 4.4.3 `client6_timo()` Function 333
 - 4.4.4 `client6_send()` Function 338
 - 4.4.5 `client6_recv()` Function 344
 - 4.4.6 `client6_recvadvert()` Function 346
 - 4.4.7 `client6_recvreply()` Function 352
 - 4.4.8 Processing Identity Association 357
 - 4.4.9 `update_ia()` Function 359
 - 4.4.10 `update_address()` Function 365
 - 4.4.11 `reestablish_ia()` Function 369
 - 4.4.12 `ia_timo()` Function 374
 - 4.4.13 Release Resources 379

- 4.5 Server Implementation 382
 - 4.5.1 server6_mainloop() Function 386
 - 4.5.2 server6_recv() Function 387
 - 4.5.3 process_relayforw() Function 391
 - 4.5.4 react_solicit() Function 396
 - 4.5.5 react_request() Function 401
 - 4.5.6 make_ia() Function 406
 - 4.5.7 react_renew() Function 417
 - 4.5.8 react_rebind() Function 419
 - 4.5.9 binding_time() Function 426
 - 4.5.10 react_release() Function 428
 - 4.5.11 react_informreq() Function 432
 - 4.5.12 server6_send() Function 434
- 4.6 Relay Agent Implementation 439
 - 4.6.1 relay6_loop() Function 439
 - 4.6.2 relay6_recv() Function 441
 - 4.6.3 relay_to_server() Function 444
 - 4.6.4 relay_to_client() Function 450
- 4.7 Implementation of DHCPv6 Authentication 454
 - 4.7.1 Data Structures Related to DHCPv6 Authentication 454
 - 4.7.2 set_auth() Function 455
 - 4.7.3 process_auth() Function (Client Side) 458
 - 4.7.4 process_auth() Function (Server Side) 462
- 4.8 DHCPv6 Operation 468
 - 4.8.1 Building the DHCPv6 Implementation 468
 - 4.8.2 Configuring a DUID 469
 - 4.8.3 Configuring the DHCPv6 Server 469
 - 4.8.4 Configuring the DHCPv6 Client 470
 - 4.8.5 Configuring the DHCPv6 Relay Agent 474
 - 4.8.6 Configuring DHCPv6 Authentication 475
 - 4.8.7 Configuring Control Command Keys 476
 - 4.8.8 Operation of DHCPv6 Services 476

5

Mobile IPv6 485

- 5.1 Introduction 485
- 5.2 Mobile IPv6 Overview 486
 - 5.2.1 Types of Nodes 487
 - 5.2.2 Basic Operation of Mobile IPv6 488