Linux Kernel Networking

Implementation and Theory

Rami Rosen

Apress®

Linux Kernel Networking: Implementation and Theory

Copyright © 2014 by Rami Rosen

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

ISBN-13 (pbk): 978-1-4302-6196-4

ISBN-13 (electronic): 978-1-4302-6197-1

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

President and Publisher: Paul Manning

Lead Editor: Michelle Lowman

Technical Reviewer: Brendan Horan

Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Louise Corrigan, James DeWolf,

Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham,

Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft,

Gwenan Spearing, Matt Wade, Steve Weiss, Tom Welsh

Coordinating Editor: Kevin Shea

Copy Editor: Corbin Collins

Compositor: SPi Global

Indexer: SPi Global

Artist: SPi Global

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales-eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code.

*To Dr. Joseph Shapira, Qualcomm Israel Founder and Ex-President, coauthor of
“CDMA Radio with Repeaters”(Springer, 2007).*

To Dr Ruth Shapira.

Iris & Dr. Shye Shapira, made of the stuff dreams are made of.

—Rami Rosen

Contents at a Glance

About the Author	xxv
About the Technical Reviewer	xxvii
Acknowledgments	xxix
Preface	xxxix
■ Chapter 1: Introduction	1
■ Chapter 2: Netlink Sockets	13
■ Chapter 3: Internet Control Message Protocol (ICMP).....	37
■ Chapter 4: IPv4	63
■ Chapter 5: The IPv4 Routing Subsystem	113
■ Chapter 6: Advanced Routing	141
■ Chapter 7: Linux Neighbouring Subsystem	165
■ Chapter 8: IPv6	209
■ Chapter 9: Netfilter	247
■ Chapter 10: IPsec	279
■ Chapter 11: Layer 4 Protocols	305
■ Chapter 12: Wireless in Linux	345
■ Chapter 13: InfiniBand.....	373
■ Chapter 14: Advanced Topics	405

■ **Appendix A: Linux API483**

■ **Appendix B: Network Administration571**

■ **Appendix C: Glossary.....589**

Index.....599

Contents

About the Author	xxv
About the Technical Reviewer	xxvii
Acknowledgments	xxix
Preface	xxxix
■ Chapter 1: Introduction	1
The Linux Network Stack	2
The Network Device	4
New API (NAPI) in Network Devices	5
Receiving and Transmitting Packets.....	5
The Socket Buffer	7
The Linux Kernel Networking Development Model	10
Summary.....	12
■ Chapter 2: Netlink Sockets	13
The Netlink Family.....	13
Netlink Sockets Libraries.....	14
The sockaddr_nl Structure	15
Userspace Packages for Controlling TCP/IP Networking	15
Kernel Netlink Sockets	16
The Netlink Message Header.....	19
NETLINK_ROUTE Messages	22
Adding and Deleting a Routing Entry in a Routing Table	24

Generic Netlink Protocol.....	25
Creating and Sending Generic Netlink Messages.....	29
Socket Monitoring Interface	31
Summary.....	32
■ Chapter 3: Internet Control Message Protocol (ICMP).....	37
ICMPv4	37
ICMPv4 Initialization	38
ICMPv4 Header	39
Receiving ICMPv4 Messages.....	42
Sending ICMPv4 Messages: “Destination Unreachable”	44
ICMPv6	47
ICMPv6 Initialization	48
ICMPv6 Header	49
Receiving ICMPv6 Messages.....	50
Sending ICMPv6 Messages	53
ICMP Sockets (“Ping sockets”)	56
Summary	57
Quick Reference	57
Methods.....	57
Tables	58
procfs entries	60
Creating “Destination Unreachable” Messages with iptables	61
■ Chapter 4: IPv4.....	63
IPv4 Header	64
IPv4 Initialization	66
Receiving IPv4 Packets	66
Receiving IPv4 Multicast Packets.....	70
IP Options	72
Timestamp Option	74
Record Route Option.....	77

IP Options and Fragmentation	86
Building IP Options	87
Sending IPv4 Packets	88
Fragmentation	94
Fast Path.....	95
Slow Path.....	97
Defragmentation	100
Forwarding	104
Summary.....	107
Quick Reference	107
Methods.....	107
Macros.....	110
■ Chapter 5: The IPv4 Routing Subsystem	113
Forwarding and the FIB.....	113
Performing a Lookup in the Routing Subsystem	115
FIB Tables	118
FIB Info	119
Caching.....	123
Nexthop (fib_nh).....	124
Policy Routing.....	126
FIB Alias (fib_alias)	127
ICMPv4 Redirect Message.....	130
Generating an ICMPv4 Redirect Message.....	131
Receiving an ICMPv4 Redirect Message	132
IPv4 Routing Cache.....	133
Summary.....	135
Quick Reference	135
Methods.....	135
Macros.....	136

Tables	137
Route Flags.....	139
■ Chapter 6: Advanced Routing	141
Multicast Routing	141
The IGMP Protocol	142
The Multicast Routing Table	143
The Multicast Forwarding Cache (MFC).....	144
Multicast Router	146
The Vif Device	147
IPv4 Multicast Rx Path.....	148
The ip_mr_forward() Method	151
The ipmr_queue_xmit() Method.....	154
The ipmr_forward_finish() Method	156
The TTL in Multicast Traffic.....	157
Policy Routing.....	157
Policy Routing Management.....	158
Policy Routing Implementation.....	158
Multipath Routing.....	159
Summary.....	160
Quick Reference	160
Methods.....	160
Macros.....	163
Procfs Multicast Entries.....	163
Table	164
■ Chapter 7: Linux Neighbouring Subsystem	165
The Neighbouring Subsystem Core	165
Creating and Freeing a Neighbour.....	172
Interaction Between Userspace and the Neighbouring Subsystem.....	174
Handling Network Events	175

The ARP protocol (IPv4)	175
ARP: Sending Solicitation Requests.....	177
ARP: Receiving Solicitation Requests and Replies	181
The NDISC Protocol (IPv6)	187
Duplicate Address Detection (DAD).....	187
NDISC: Sending Solicitation Requests	189
NDISC: Receiving Neighbour Solicitations and Advertisements	193
Summary	200
Quick Reference	200
Methods.....	200
Macros.....	204
The neigh_statistics Structure	206
Table	207
■ Chapter 8: IPv6	209
IPv6 – Short Introduction.....	209
IPv6 Addresses	210
Special Addresses	210
Multicast Addresses	212
IPv6 Header	213
Extension Headers.....	215
IPv6 Initialization	217
Autoconfiguration	217
Receiving IPv6 Packets	218
Local Delivery	222
Forwarding	224
Receiving IPv6 Multicast Packets.....	228
Multicast Listener Discovery (MLD).....	230
Joining and Leaving a Multicast Group	230
MLDv2 Multicast Listener Report	233
Multicast Source Filtering (MSF)	234

Sending IPv6 Packets	239
IPv6 Routing	240
Summary	240
Quick Reference	240
Methods	240
Macros	244
Tables	245
Special Addresses	246
Routing Tables Management in IPv6	246
■ Chapter 9: Netfilter	247
Netfilter Frameworks	247
Netfilter Hooks	248
Registration of Netfilter Hooks	249
Connection Tracking	250
Connection Tracking Initialization	251
Connection Tracking Entries	255
Connection Tracking Helpers and Expectations	259
IPTables	262
Delivery to the Local Host	265
Forwarding the Packet	265
Network Address Translation (NAT)	266
NAT Hook Callbacks and Connection Tracking Hook Callbacks	268
NAT Hook Callbacks	271
Connection Tracking Extensions	273
Summary	274
Quick Reference	274
Methods	274
MACRO	276
Tables	277

■ Chapter 10: IPsec	279
General	279
IKE (Internet Key Exchange)	279
IPsec and Cryptography	280
The XFRM Framework	281
XFRM Initialization	282
XFRM Policies	282
XFRM States (Security Associations)	285
ESP Implementation (IPv4)	288
IPv4 ESP Initialization	290
Receiving an IPsec Packet (Transport Mode)	291
Sending an IPsec Packet (Transport Mode)	294
XFRM Lookup	295
NAT Traversal in IPsec	298
NAT-T Mode of Operation	299
Summary	299
Quick Reference	299
Methods	299
Table	302
■ Chapter 11: Layer 4 Protocols	305
Sockets	305
Creating Sockets	306
UDP (User Datagram Protocol)	310
UDP Initialization	311
Sending Packets with UDP	313
Receiving Packets from the Network Layer (L3) with UDP	316
TCP (Transmission Control Protocol)	318
TCP Header	319
TCP Initialization	321
TCP Timers	322

TCP Socket Initialization	323
TCP Connection Setup	323
Receiving Packets from the Network Layer (L3) with TCP	324
Sending Packets with TCP	325
SCTP (Stream Control Transmission Protocol)	326
SCTP Packets and Chunks	328
SCTP Chunk Header	328
SCTP Chunk	329
SCTP Associations	330
Setting Up an SCTP Association	331
Receiving Packets with SCTP	332
Sending Packets with SCTP	332
SCTP HEARTBEAT	332
SCTP Multistreaming	333
SCTP Multihoming	333
DCCP: The Datagram Congestion Control Protocol	333
DCCP Header	334
DCCP Initialization	336
DCCP Socket Initialization	337
Receiving Packets from the Network Layer (L3) with DCCP	338
Sending Packets with DCCP	338
DCCP and NAT	339
Summary	340
Quick Reference	340
Methods	340
Macros	342
Tables	342
■ Chapter 12: Wireless in Linux	345
Mac80211 Subsystem	345
The 802.11 MAC Header	346
The Frame Control	347

The Other 802.11 MAC Header Members.....	348
Network Topologies	349
Infrastructure BSS	349
IBSS, or Ad Hoc Mode	350
Power Save Mode.....	350
Entering Power Save Mode	350
Exiting Power Save Mode	351
Handling the Multicast/Broadcast Buffer	351
The Management Layer (MLME)	353
Scanning.....	353
Authentication	353
Association	353
Reassociation	353
Mac80211 Implementation	354
Rx Path	356
Tx Path.....	356
Fragmentation	357
Mac80211 debugfs.....	358
Wireless Modes	359
High Throughput (ieee802.11n).....	359
Packet Aggregation.....	360
Mesh Networking (802.11s)	362
HWMP Protocol	364
Setting Up a Mesh Network.....	365
Linux Wireless Development Process.....	366
Summary	366
Quick Reference	366
Methods.....	366
Table	371

■ Chapter 13: InfiniBand	373
RDMA and InfiniBand—General	373
The RDMA Stack Organization	374
RDMA Technology Advantages	375
InfiniBand Hardware Components	375
Addressing in InfiniBand	375
InfiniBand Features	376
InfiniBand Packets	376
Management Entities	377
RDMA Resources	378
RDMA Device	378
Protection Domain (PD)	380
Address Handle (AH)	380
Memory Region (MR)	381
Fast Memory Region (FMR) Pool	382
Memory Window (MW)	382
Completion Queue (CQ)	382
eXtended Reliable Connected (XRC) Domain	384
Shared Receive Queue (SRQ)	384
Queue Pair (QP)	386
Work Request Processing	391
Supported Operations in the RDMA Architecture	392
Multicast Groups	396
Difference Between the Userspace and the Kernel-Level RDMA API	396
Summary	397
Quick Reference	397
Methods	397
■ Chapter 14: Advanced Topics	405
Network Namespaces	405
Namespaces Implementation	406
UTS Namespaces Implementation	414

Network Namespaces Implementation	416
Network Namespaces Management	423
Cgroups	426
Cgroups Implementation	427
Cgroup Devices Controller: A Simple Example	430
Cgroup Memory Controller: A Simple Example	430
The net_prio Module	431
The cls_cgroup Classifier	432
Mounting cgroup Subsystems	432
Busy Poll Sockets	433
Enabling Globally	435
Enabling Per Socket	435
Tuning and Configuration	435
Performance	436
The Linux Bluetooth Subsystem	436
HCI Layer	439
HCI Connection	441
L2CAP	441
BNEP	442
Receiving Bluetooth Packets: Diagram	443
L2CAP Extended Features	444
Bluetooth Tools	444
IEEE 802.15.4 and 6LoWPAN	445
Neighbor Discovery Optimization	446
Linux Kernel 6LoWPAN	447
Near Field Communication (NFC)	450
NFC Tags	450
NFC Devices	451
Communication and Operation Modes	451
Host-Controller Interfaces	451
Linux NFC support	452

Userspace Architecture	456
NFC on Android	457
Notifications Chains	458
The PCI Subsystem.....	461
Wake-On-LAN (WOL).....	463
Teaming Network Device.....	464
The PPPoE Protocol	465
PPPoE Header.....	465
PPPoE Initialization	467
Sending and Receiving Packets with PPPoE	468
Android.....	472
Android Networking.....	472
Android internals: Resources.....	473
Summary	474
Quick Reference	474
Methods.....	474
Macros.....	482
■ Appendix A: Linux API	483
The sk_buff Structure	483
struct skb_shared_info	492
The net_device structure	493
RDMA (Remote DMA).....	518
RDMA Device.....	518
The ib_register_client() Method.....	518
The ib_unregister_client() Method.....	519
The ib_get_client_data() Method.....	519
The ib_set_client_data() Method	519
The INIT_IB_EVENT_HANDLER macro	520
The ib_register_event_handler() Method.....	520
The ib_event_handler struct:.....	520

The <code>ib_event</code> Struct	520
The <code>ib_unregister_event_handler()</code> Method.....	522
The <code>ib_query_device()</code> Method	522
The <code>ib_query_port()</code> Method	526
The <code>rdma_port_get_link_layer()</code> Method	529
The <code>ib_query_gid()</code> Method.....	530
The <code>ib_query_pkey()</code> Method	530
The <code>ib_modify_device()</code> Method.....	530
The <code>ib_modify_port()</code> Method.....	531
The <code>ib_find_gid()</code> Method.....	532
The <code>ib_find_pkey()</code> Method	532
The <code>rdma_node_get_transport()</code> Method	532
The <code>rdma_node_get_transport()</code> Method	532
The <code>ib_mtu_to_int()</code> Method	533
The <code>ib_width_enum_to_int()</code> Method.....	533
The <code>ib_rate_to_mult()</code> Method	533
The <code>ib_rate_to_mbps()</code> Method.....	534
The <code>ib_rate_to_mbps()</code> Method.....	534
Protection Domain (PD)	534
The <code>ib_alloc_pd()</code> Method	534
The <code>ib_dealloc_pd()</code> Method	534
eXtended Reliable Connected (XRC).....	535
The <code>ib_alloc_xrca()</code> Method	535
The <code>ib_dealloc_xrca_cq()</code> Method.....	535
Shared Receive Queue (SRQ)	535
The <code>ib_create_srca()</code> Method.....	536
The <code>ib_modify_srca()</code> Method	536
The <code>ib_query_srca()</code> Method.....	537
The <code>ib_destory_srca()</code> Method.....	537
The <code>ib_post_srca_rcv()</code> Method	537

Address Handle (AH)	538
The <code>ib_create_ah()</code> Method	539
The <code>ib_init_ah_from_wc()</code> Method	539
The <code>ib_create_ah_from_wc()</code> Method	540
The <code>ib_modify_ah()</code> Method	540
The <code>ib_query_ah()</code> Method	540
The <code>ib_destory_ah()</code> Method	540
Multicast Groups	541
The <code>ib_attach_mcast()</code> Method	541
The <code>ib_detach_mcast()</code> method	541
Completion Queue (CQ)	541
The <code>ib_create_cq()</code> Method	541
The <code>ib_resize_cq()</code> Method	542
The <code>ib_modify_cq()</code> Method	542
The <code>ib_peek_cq()</code> Method	542
The <code>ib_req_notify_cq()</code> Method	543
The <code>ib_req_ncomp_notif()</code> Method	543
The <code>ib_poll_cq()</code> Method	543
The <code>ib_destory_cq()</code> Method	547
Queue Pair (QP)	547
The <code>ib_qp_cap</code> Struct	547
The <code>ib_create_qp()</code> Method	547
The <code>ib_modify_qp()</code> Method	549
The <code>ib_query_qp()</code> Method	553
The <code>ib_open_qp()</code> Method	554
The <code>ib_close_qp()</code> Method	554
The <code>ib_post_recv()</code> Method	555
The <code>ib_post_send()</code> Method	555
Memory Windows (MW)	559
The <code>ib_alloc_mw()</code> Method	559
The <code>ib_bind_mw()</code> Method	560
The <code>ib_dealloc_mw()</code> Method	560

Memory Region (MR)	561
The <code>ib_get_dma_mr()</code> Method	561
The <code>ib_dma_mapping_error()</code> Method	561
The <code>ib_dma_map_single()</code> Method	561
The <code>ib_dma_unmap_single()</code> Method	562
The <code>ib_dma_map_single_attrs()</code> Method	562
The <code>ib_dma_unmap_single_attrs()</code> Method	562
The <code>ib_dma_map_page()</code> Method	563
The <code>ib_dma_unmap_page()</code> Method	563
The <code>ib_dma_map_sg()</code> Method	564
The <code>ib_dma_unmap_sg()</code> Method	564
The <code>ib_dma_map_sg_attr()</code> Method	564
The <code>ib_dma_unmap_sg()</code> Method	565
The <code>ib_sg_dma_address()</code> Method	565
The <code>ib_sg_dma_len()</code> Method	565
The <code>ib_dma_sync_single_for_cpu()</code> Method	565
The <code>ib_dma_sync_single_for_device()</code> Method	566
The <code>ib_dma_alloc_coherent()</code> Method	566
The <code>ib_dma_free_coherent()</code> method	566
The <code>ib_reg_phys_mr()</code> Method	567
The <code>ib_rereg_phys_mr()</code> Method	567
The <code>ib_query_mr()</code> Method	568
The <code>ib_dereg_mr()</code> Method	569
■ Appendix B: Network Administration	571
arp	571
arping	571
arp tables	571
arpwatch	571
ApacheBench (ab)	572
brctl	572
conntrack-tools	572

crtools	572
ebtables.....	572
ether-wake	572
ethtool	573
git	573
hciconfig.....	574
hcidump	574
hcitool.....	574
ifconifg	574
ifenslave	574
iperf	575
Using iperf	575
iproute2.....	575
iptables and iptables6.....	579
ipvsadm.....	579
iw	579
iwconfig.....	579
libreswan Project	580
l2ping	580
lowpan-tools	580
lshw	580
lscpu.....	580
lspci.....	580
mrouted.....	580
nc	580
ngrep	581
netperf.....	581
netsniff-ng.....	581
netstat	581

nmap (Network Mapper)	582
openswan	582
OpenVPN	582
packeth	582
ping	582
pimd	583
poptop	583
ppp	583
pktgen	583
radvd	583
route	583
RP-PPPoE	584
sar	584
smcroute	584
snort	584
suricata	584
strongSwan	584
sysctl	584
taskset	585
tcpdump	585
top	585
tracepath	585
tracert	585
tshark	585
tunctl	586
udevadm	586
unshare	587
vconfig	587

wpa_suplicant.....	587
wireshark	588
XORP.....	588
■ Appendix C: Glossary.....	589
Index.....	599

About the Author

Rami Rosen is a software engineer, a computer science graduate of the Technion, Israel High Institute of Technology. In the last 17 years he has been a software developer for three innovative startups and a semiconductor company. Rami lives in Israel and he has participated in highly advanced Linux kernel projects, in particular those related to networking. He has published several articles and given lectures about Linux kernel networking and virtualization.

About the Technical Reviewer

Brendan Horan is a hardware fanatic, with a full high rack of all types of machine architectures in his home. He has more than ten years of experience working with large UNIX systems and tuning the underlying hardware for optimal performance and stability. Brendan's love for all forms of hardware has helped him throughout his IT career, from fixing laptops to tuning servers and their hardware in order to suit the needs of high-availability designs and ultra low-latency applications. Brendan takes pride in the open source movement and is happy to say that every computer in his house is powered by open source technology. He resides in Hong Kong with his wife, Vikki, who continues daily to teach him more Cantonese.

Acknowledgments

Thanks to my editors for giving me the honor of writing this book; to Michelle Lowman, the lead editor, for believing in this book while it was still just an idea; to Kevin Shea, the coordinating editor, who guided and supported me from the initial stages until the book was fully realized; to Brendan Horan, the technical reviewer, for his helpful comments that helped me to improve the book by a lot; to Troy Mott, the development editor, for his many suggestions and for his hard work; to Corbin Collins and Roger LeBlanc, the copy editors, for shaping up the text; and to Kumar Dhaneesh from the production team.

I would like to thank the Linux kernel networking maintainer, David Miller, for the great work he has done over all these years and all the developers who continue to participate and contribute to the networking subsystem. I would like also to say thanks to the Linux kernel networking community and all its members who helped me by reviewing my text: Julian Anastasov, Timo Teras, Steffen Klassert, Gerrit Renker, Javier Cardona, Gao feng, Vlad Yasevich, Cong Wang, Florian Westphal, Reuben Hawkins, Pekka Savola, Andreas Steffen, Daniel Borkmann, Joachim Nilsson, David Hauweele, Maxime Ripard, Alexandre Belloni, Benjamin Zores, and too many others to mention. Thanks to Donald Wood and Eliezer Tamir from Intel for their help with the “Busy Polling Sockets” section, and to Samuel Ortiz from Intel for his advice in preparing the NFC section. Thanks for Dotan Barak, an InfiniBand expert, for contributing Chapter 13, “InfiniBand.”

Rami Rosen

Preface

This book takes you on a guided, in-depth tour of the current Linux kernel networking implementation and the theory behind it. For almost a decade, no new book about Linux networking has been written. A decade of dynamic and fast-paced Linux kernel development is quite a long time. There are important kernel networking subsystems that are not described in any other book; for example, IPv6, IPsec, Wireless (IEEE 802.11), IEEE 802.15.4, NFC, InfiniBand, and more. There is also very little information on the Web about the implementation details of these subsystems. For all these reasons, I have written this book.

About ten years ago I made my first steps in kernel programming. I was a developer in a startup taking part in a VoIP project for a Linux-based set-top box (STB). There were crashes in the USB stack with some USB cameras, and we had to delve into the code to try to find a solution, because the vendors of that STB did not want to spend time to solve the problem. In fact, it was not that they did not want to, they simply did not know how to. In these days, there was almost no documentation about the USB stack. The *Linux Device Drivers* book from O'Reilly in those days was only in its second edition (the USB chapter was added only in the third edition). Success in that project was crucial for us as a startup. I had learned much about kernel programming in the process of solving the USB crash. Later on we had a project where a NAT traversal solution was needed. The userspace solution was so heavy that the device quickly crashed. When I suggested a kernel solution, my managers were very skeptical, but they did let me try. The kernel solution proved to be very stable and took much less CPU than the userspace solution. Since then I have taken part in many kernel networking projects. This book is a result of my many years of development and research.

Who This Book Is For

This book is intended for computer professionals, including developers, software architects, designers, project managers, and CTOs, who are working on networking-related projects. These projects can be in a wide range of professional areas, such as communication, data centers, embedded devices, virtualization, security, and more. In addition, students and academy researchers and theorists who deal with networking projects or networking research or operating systems research will find a lot of help in this book.

How This Book Is Structured

In Chapter 1 you will find a general overview of the Linux kernel and the Linux network stack. Other topics in this chapter include the implementation of the network device, the socket buffer, and the Rx and Tx paths. Chapter 1 concludes with a section about the Linux Kernel Networking Development Model.

In chapter 2 you will learn about netlink sockets, which provide a mechanism for bidirectional communication between userspace and the kernel, and which are used by the networking subsystem as well as by other subsystems. You will also find a section in this chapter about generic netlink sockets, which can be perceived as advanced netlink sockets, and which you will encounter in Chapter 12 and while browsing the kernel networking source code.

In Chapter 3 you will learn about the ICMP protocol, which helps to keep the system behaving correctly by sending error and control messages about the network layer (L3). You will learn about the implementation of the ICMP protocol both in IPv4 and in IPv6.

Chapter 4 delves into the IPv4 protocol—the Internet and modern life cannot be described without it. You will learn about the structure of IPv4 header, about the Rx and Tx path, about IP options, about fragmentation and defragmentation and why they are needed, and about forwarding packets, which is one of the important tasks of IPv4.

Chapters 5 and 6 are devoted to the IPv4 Routing Subsystem. In chapter 5 you will learn how a lookup in the routing subsystem is performed, how the routing tables are organized, which optimizations are used in the IPv4 routing subsystem and about the removal of the IPv4 routing cache. Chapter 6 discusses advanced routing topics such as Multicast Routing, Policy Routing, and Multipath Routing.

Chapter 7 endeavors to explain the neighbouring subsystem. You will learn about the ARP protocol, which is used in IPv4, and about the the NDISC protocol used in IPv6, and about some of the differences between the two protocols. You will also learn about the Duplicate Address Detection (DAD) mechanism in IPv6.

Chapter 8 discusses the IPv6 protocol, which seems to be the inevitable solution to the shortage of IPv4 addresses. This chapter describes the implementation of IPv6 and discusses topics such as IPv6 addresses, the IPv6 header and extension headers, autoconfiguration in IPv6, Rx path, and forwarding. It also describes the MLD protocol.

Chapter 9 deals with the netfilter subsystem. You will learn about netfilter hooks and how they are registered, about Connection Tracking, about IP tables and Network Address Translation (NAT), and about callback used by Connection Tracking and NAT.

Chapter 10 deals with IPsec, one of the most complex networking subsystems. Topics like the IKE protocol (which is implemented in userspace) and cryptography aspects of IPsec are discussed briefly (full treatment is beyond the scope of the book). You will learn about the XFRM framework, which is the basis of the Linux IPsec subsystem, and about its two most important structures: XFRM policy and XFRM state. The ESP protocol is briefly described, as well as the IPsec Rx path and Tx path in transport mode. The chapter concludes with a section about XFRM lookup and a short section about NAT traversal.

Chapter 11 describes four Layer 4 protocols, starting with the most commonly used protocols, UDP and TCP, and concluding with two newer protocols, SCTP and DCCP.

Chapter 12 deals with wireless in Linux (IEEE 802.11). You will learn about the mac80211 subsystem and its implementation, about various wireless network topologies, about power save mode, and about IEEE 802.11n and packet aggregation. There is also a section devoted to Wireless Mesh networks in this chapter.

Chapter 13 delves into the InfiniBand subsystem, a technology enjoying a rising popularity in datacenters. You will learn about the RDMA stack organization, about addressing in InfiniBand, about the organization of InfiniBand packets, and about the RDMA API.

Chapter 14 concludes the book with a discussion of advanced topics such as Linux namespaces and network namespaces in particular, Busy Poll Sockets, the Bluetooth subsystem, the IEEE 802.15.4 subsystem, the Near Field Communication (NFC) subsystem, the PCI subsystem, and more.

Appendices A, “Linux API,” and C, “Glossary,” provide complete reference information for many topics discussed in the book. Appendix B, “Network Administration,” provides information about various tools which you will need while working with Linux kernel networking.

Conventions

Throughout the book, I’ve kept a consistent style. All code snippets, whether inside text paragraphs or on lines of their own, along with library paths, shell commands, URLs, and other code-related elements, are set in monospaced font, like this. New terms are set off in *italics*, and other emphasis may be given in **bold**.