TURING 图灵程序设计丛书

Apress®

Linux Kernel Networking

Implementation and Theory

精通Linux内核网络

最详尽的Linux内核网络专著

深入剖析IPsec、Wireless、InfiniBand等重要内核网络子系统

【以色列】Rami Rosen 著

袁国忠 译

中国工信出版集团

人民邮电出版社 POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

Linux Kernel Networking
Implementation and Theory

精通Linux内核网络

【以色列】Rami Rosen 著
袁国忠 译

人民邮电出版社
北京

图书在版编目（CIP）数据

精通Linux内核网络 /（以）罗森（Rosen,R.）著；
袁国忠译. -- 北京 : 人民邮电出版社, 2015.6
（图灵程序设计丛书）
ISBN 978-7-115-39293-0

Ⅰ. ①精… Ⅱ. ①罗… ②袁… Ⅲ. ①Linux操作系统
Ⅳ. ①TP316.89

中国版本图书馆CIP数据核字(2015)第098093号

内 容 提 要

本书讨论 Linux 内核网络栈的实现及其原理，深入而详尽地分析网络子系统及其架构，主要内容包括：内核网络基础知识、Netlink 套接字、ARP、邻居发现和 ICMP 等重要协议的实现、IPv4 和 IPv6 的深入探索、Linux 路由选择、Netfilter 和 IPsec 的实现、Linux 无线网络、InfiniBand 等。

本书不仅适合从事网络相关项目的专业人员参考，也能为相关研究人员和学生提供极大帮助。

◆ 著 [以色列] Rami Rosen
译 袁国忠
责任编辑 朱 巍
责任印制 杨林杰

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：35
字数：827千字 2015年6月第1版
印数：1-4 000册 2015年6月北京第1次印刷
著作权合同登记号 图字：01-2014-6528号

定价：99.00元

读者服务热线：**(010)51095186转600** 印装质量热线：**(010)81055316**
反盗版热线：**(010)81055315**
广告经营许可证：京崇工商广字第 **0021** 号

站在巨人的肩上

Standing on Shoulders of Giants

iTuring.cn

版权声明

献　　词

献给高通公司以色列分公司的创建者和前总裁、*CDMA Radio with Repeaters* 合著者 Joseph Shapira 博士。

也献给 Ruth Shapira 博士，筑梦者 Iris 和 Shye Shapira 博士。

——Rami Rosen

前　言

本书将引领你完成一次深入探索 Linux 内核网络实现和理论的旅程。最近 10 年，始终没有讨论 Linux 网络的新书上市。对于快速发展的 Linux 内核来说，10 年时间可谓相当漫长。很多重要的内核网络子系统都没有人著书介绍，其中包括 IPv6、IPsec、Wireless（IEEE 802.11）、IEEE 802.15.4、NFC、InfiniBand 等。网上讨论这些子系统实现细节的资源也是凤毛麟角。有鉴于此，我编写了本书。

大约在 10 年前，我向 Linux 内核编程迈出了第一步。当时我是一家创业公司的开发人员，参与了一个基于 Linux 的机顶盒（STB）的 VoIP 项目。这个项目涉及 USB 摄像机，USB 栈经常崩溃。鉴于该 STB 厂商不想花时间解决这种问题，我们不得不深入研究源代码，试图找到解决方案。事实上，不是厂商不想解决问题，而是根本不知道如何解决。当时，几乎找不到任何有关 USB 栈的文档。那时 O'Reilly 出版的 *Linux Device Drivers* 还是第二版，而讨论 USB 的章节是第三版才增补的。作为一家创业公司，成功完成这个项目对我们来说生死攸关。在解决 USB 崩溃问题的过程中，我不得不大量地学习 Linux 内核编程知识。后来，我们又做了一个需要实现 NAT 穿越解决方案的项目。由于用户空间解决方案过于庞大，设备很快就崩溃了。有鉴于此，我提出了一种内核解决方案。项目经理对这种想法深表怀疑，但还是决定让我试试。事实证明，内核解决方案非常稳定，占用的 CPU 周期比用户空间解决方案少得多。从那以后，我参与了很多内核网络项目。本书正是我多年开发和研究工作的结晶。

针对的读者

本书是为计算机专业人员编写的，包括从事网络相关项目的开发人员、软件架构师、设计人员、项目经理和 CTO。这些项目涉及的专业领域非常广泛，包括通信、数据中心、嵌入式设备、虚拟化、安全等。另外，对于从事网络项目、网络研究或操作系统研究的学生、学术研究人员和理论研究者，本书也可提供极大的帮助。

组织结构

第 1 章首先概述了 Linux 内核和 Linux 网络栈，然后介绍了网络设备、套接字缓存区、接收路径和传输路径的实现，最后概述了 Linux 内核网络开发模型。

第 2 章讨论了 Netlink 套接字。这种套接字提供了一种在用户空间和内核之间进行双向通信

的机制，为网络子系统及其他一些子系统所采用。另外，本章还讨论了通用 Netlink 套接字。这是一种高级 Netlink 套接字，第 12 章也有介绍，内核网络源代码中也能见到。

第 3 章讨论了 ICMP 协议。它通过发送有关网络层（L3）的错误和控制消息来帮助确保系统正确地运行。本章还介绍了 IPv4 和 IPv6 中的 ICMP 实现。

第 4 章深入讨论了 IPv4 协议。如果没有它，Internet 和当代人的生活都不会是现在的样子。具体内容包括 IPv4 报头的结构、接收和传输路径、IP 选项、分段和重组及这样做的原因、数据包转发（这是 IPv4 最重要的任务之一）。

第 5 和 6 章讨论了 IPv4 路由选择子系统。第 5 章介绍了路由选择子系统查找是如何进行的，路由选择表是如何组织的，IPv4 路由选择子系统使用了哪些优化方法，以及为何将 IPv4 路由选择缓存删除。第 6 章讨论了高级路由选择主题，如组播路由选择、策略路由选择和多路径路由选择。

第 7 章阐述了邻接子系统。主要内容有：IPv4 使用的 ARP 协议、IPv4 使用的 NDISC 协议以及这两种协议之间的一些差别、IPv6 使用的重复地址检测（DAD）机制。

第 8 章讨论了 IPv6 协议，看起来它终将成为 IPv4 地址短缺的解决方案。本章介绍了 IPv6 的实现，讨论了 IPv6 地址、IPv6 报头和扩展报头、IPv6 自动配置、接收路径和转发等主题，还将介绍 MLD 协议。

第 9 章讨论了 Netfilter 子系统，包括 Netfilter 钩子回调函数及其注册、连接跟踪、IP 表和网络地址转换（NAT）以及连接跟踪和 NAT 使用的回调函数。

第 10 章讨论了 IPsec，这是最复杂的网络子系统之一。本章将简要地讨论 IKE 协议（它是在用户空间中实现的）和 IPsec 加密方面的内容（全面讨论它们超出了本书的范围）。你将学习 XFRM 框架（它是 Linux IPsec 子系统的基础）及其两个最重要的结构——XFRM 策略和 XFRM 状态。本章还将简要地讨论 ESP 协议以及传输模式中的 IPsec 接收路径和传输路径。最后，本章将介绍 XFRM 查找和 NAT 穿越。

第 11 章阐述了 4 种第 4 层协议。首先介绍最常用的协议 UDP 和 TCP，然后是较新的协议 SCTP 和 DCCP。

第 12 章讨论了 Linux 无线子系统（IEEE 802.11）。你将学习 mac80211 子系统及其实现、各种无线网络拓扑、省电模式、IEEE 802.11n 和数据包聚合。本章还专辟一节探讨了无线网状网络。

第 13 章深入讨论了 InfiniBand 子系统，这是一种在数据中心中使用得越来越广泛的技术。你将学习 RDMA 栈的组织结构、InfiniBand 编址、InfiniBand 数据包的结构以及 RDMA API。

第 14 章是本书的最后一章，将讨论一些高级主题，如 Linux 命名空间（尤其是网络命名空间）、频繁轮询套接字、蓝牙子系统、IEEE 802.15.4 子系统、近场通信（NFC）子系统、PCI 子系统等。

附录 A 和附录 C 提供了本书讨论的众多主题的完整参考信息。附录 B 介绍了使用 Linux 内核网络时需要的各种工具。

排版约定

本书始终采用一致的排版风格。所有代码段（无论包含在正文中还是单独列出）都使用等宽字体，新术语使用*楷体*，其他需要突出的内容使用**粗体**。

致　　谢

感谢诸位编辑们给我这个机会，让我得以有幸编写本书。感谢责任编辑 Michelle Lowman 在本书尚属思路雏形时就对它充满信心。感谢协调编辑 Kevin Shea 在本书编写过程中始终如一的指导和支持。感谢技术审阅 Brendan Horan 提供有益的评论，让本书的质量改善良多。感谢开发编辑 Troy Mott 提供的大量建议及所做的艰苦工作。感谢文字编辑 Corbin Collins 和 Roger LeBlanc 为文字润色。感谢印刷团队成员 Kumar Dhaneesh。

这里要感谢 Linux 内核网络子系统维护者 David Miller 多年来的出色工作，还有一直以来为该子系统贡献代码的所有开发人员。还要感谢 Linux 内核网络社区及帮助审阅本书的成员，他们是：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，等等。感谢 Intel 公司的 Donald Wood 和 Eliezer Tamir 在我编写 14.3 节时提供的帮助，还有 Samuel Ortiz 为我编写 NFC 方面的内容提供的建议。感谢 InfiniBand 专家 Dotan Barak 协助撰写了本书第 13 章。

——Rami Rosen

目　录

第 1 章　绪论 ……1
1.1 Linux 网络栈 ……2
1.2 网络设备 ……4
1.2.1 网络设备中的 NAPI ……5
1.2.2 数据包的收发 ……5
1.2.3 套接字缓冲区 ……7
1.3 Linux 内核网络开发模型 ……10
1.4 总结 ……12

第 2 章　Netlink 套接字 ……13
2.1 Netlink 簇 ……13
2.1.1 Netlink 套接字库 ……15
2.1.2 结构 sockaddr_nl ……15
2.1.3 用于控制 TCP/IP 联网的用户空间包 ……15
2.1.4 内核 Netlink 套接字 ……16
2.1.5 Netlink 消息报头 ……20
2.1.6 NETLINK_ROUTE 消息 ……22
2.1.7 在路由选择表中添加和删除路由选择条目 ……24
2.2 通用 Netlink 协议 ……25
2.2.1 创建和发送通用 Netlink 消息 ……29
2.2.2 套接字监视接口 ……31
2.3 总结 ……32
2.4 快速参考 ……32

第 3 章　Internet 控制消息协议（ICMP） ……36
3.1 ICMPv4 ……36
3.1.1 ICMPv4 的初始化 ……37
3.1.2 ICMPv4 报头 ……38
3.1.3 接收 ICMPv4 消息 ……42
3.1.4 发送 ICMPv4 消息：目的地不可达 ……43
3.2 ICMPv6 ……47
3.2.1 ICMPv6 初始化 ……47
3.2.2 ICMPv6 报头 ……48
3.2.3 接收 ICMPv6 消息 ……49
3.2.4 发送 ICMPv6 消息 ……52
3.3 ICMP 套接字（ping 套接字） ……55
3.4 总结 ……56
3.5 快速参考 ……56
3.5.1 方法 ……56
3.5.2 表格 ……57
3.5.3 procfs 条目 ……58
3.5.4 使用 iptables 创建“目的地不可达”消息 ……59

第 4 章　IPv4 ……61
4.1 IPv4 报头 ……62
4.2 IPv4 的初始化 ……63
4.3 接收 IPv4 数据包 ……64
4.4 接收 IPv4 组播数据包 ……67
4.5 IP 选项 ……69
4.5.1 时间戳选项 ……71
4.5.2 记录路由选项 ……74
4.5.3 IP 选项和分段 ……82
4.5.4 创建 IP 选项 ……84
4.6 发送 IPv4 数据包 ……85
4.7 分段 ……89
4.7.1 快速路径 ……90
4.7.2 慢速路径 ……93
4.8 重组 ……94

4.9 转发……99
4.10 总结……101
4.11 快速参考……101
4.11.1 方法……102
4.11.2 宏……104
第 5 章 IPv4 路由选择子系统……105
5.1 转发和 FIB……105
5.2 在路由选择子系统中进行查找……107
5.3 FIB 表……110
5.3.1 FIB 信息……110
5.3.2 缓存……115
5.3.3 下一跳……115
5.3.4 策略路由选择……117
5.3.5 FIB 别名……118
5.4 ICMPv4 重定向消息……121
5.4.1 生成 ICMPv4 重定向消息……122
5.4.2 接收 ICMPv4 重定向消息……123
5.4.3 IPv4 路由选择缓存……125
5.5 总结……126
5.6 快速参考……126
5.6.1 方法……127
5.6.2 宏……128
5.6.3 表……128
5.6.4 路由标志……129
第 6 章 高级路由选择……131
6.1 组播路由选择……131
6.1.1 IGMP……132
6.1.2 组播路由选择表……133
6.1.3 组播转发缓存（MFC）……134
6.1.4 组播路由器……136
6.1.5 vif 设备……137
6.1.6 IPv4 组播接收路径……138
6.1.7 方法 ip_mr_forward()……141
6.1.8 方法 ipmr_queue_xmit()……143
6.1.9 方法 ipmr_forward_finish()……145
6.1.10 组播流量中的 TTL……146
6.2 策略路由选择……146
6.2.1 策略路由选择的管理……147
6.2.2 策略路由选择的实现……147
6.3 多路径路由选择……148
6.4 总结……149
6.5 快速参考……149
6.5.1 方法……149
6.5.2 宏……151
6.5.3 procfs 组播条目……152
6.5.4 表……152
第 7 章 Linux 邻接子系统……153
7.1 邻接子系统的核心……153
7.1.1 创建和释放邻居……160
7.1.2 用户空间和邻接子系统之间的交互……161
7.1.3 处理网络事件……163
7.2 ARP 协议（IPv4）……163
7.2.1 ARP：发送请求……165
7.2.2 ARP：接收请求和应答……168
7.3 NDISC 协议（IPv6）……174
7.3.1 重复地址检测（DAD）……174
7.3.2 NIDSC：发送请求……176
7.3.3 NDISC：接收邻居请求和通告……179
7.4 总结……185
7.5 快速参考……186
7.5.1 方法……186
7.5.2 宏……189
7.5.3 结构 neigh_statistics……190
7.5.4 表……191
第 8 章 IPv6……192
8.1 IPv6 简介……192
8.2 IPv6 地址……193
8.2.1 特殊地址……193
8.2.2 组播地址……194
8.3 IPv6 报头……195
8.4 扩展报头……197
8.5 IPv6 初始化……199
8.6 自动配置……200
8.7 接收 IPv6 数据包……201

8.7.1 本地投递 ……204
8.7.2 转发 ……206
8.8 接收 IPv6 组播流量 ……210
8.9 组播侦听者发现（MLD） ……211
8.9.1 加入和退出组播组 ……212
8.9.2 MLDv2 组播侦听者报告 ……215
8.9.3 组播源过滤 ……215
8.10 发送 IPv6 数据包 ……220
8.11 IPv6 路由选择 ……221
8.12 总结 ……221
8.13 快速参考 ……221
8.13.1 方法 ……221
8.13.2 宏 ……224
8.13.3 表 ……224
8.13.4 特殊地址 ……225
8.13.5 IPv6 路由选择表的管理 ……226

第 9 章 Netfilter ……227

9.1 Netfilter 框架 ……227
9.2 Netfilter 挂接点 ……228
注册 Netfilter 钩子回调函数 ……229
9.3 连接跟踪 ……230
9.3.1 连接跟踪的初始化 ……231
9.3.2 连接跟踪条目 ……234
9.3.3 连接跟踪辅助方法和期望连接 ……238
9.3.4 iptables ……241
9.3.5 投递到当前主机 ……243
9.3.6 转发数据包 ……245
9.3.7 网络地址转换（NAT） ……245
9.3.8 NAT 钩子回调函数和连接跟踪钩子回调函数 ……247
9.3.9 NAT 钩子回调函数 ……250
9.3.10 连接跟踪扩展 ……252
9.4 总结 ……253
9.5 快速参考 ……253
9.5.1 方法 ……253
9.5.2 宏 ……255
9.5.3 表 ……255
9.5.4 工具和库 ……256

第 10 章 IPsec ……257

10.1 概述 ……257
10.2 Internet 密钥交换（IKE） ……257
10.3 IPsec 和加密 ……259
10.4 XFRM 框架 ……259
10.4.1 XFRM 的初始化 ……260
10.4.2 XFRM 策略 ……260
10.4.3 XFRM 状态（安全关联） ……263
10.5 IPv4 ESP 的实现 ……266
10.6 接收 IPsec 数据包（传输模式） ……268
10.7 发送 IPsec 数据包（传输模式） ……271
10.8 XFRM 查找 ……272
10.9 IPsec 的 NAT 穿越功能 ……275
10.10 总结 ……276
10.11 快速参考 ……276
10.11.1 方法 ……276
10.11.2 表 ……278

第 11 章 第 4 层协议 ……280

11.1 套接字 ……280
11.2 创建套接字 ……281
11.3 用户数据包协议（UDP） ……285
11.3.1 UDP 的初始化 ……286
11.3.2 发送 UDP 数据包 ……287
11.3.3 接收来自网络层（L3）的 UDP 数据包 ……290
11.4 传输控制协议（TCP） ……293
11.4.1 TCP 报头 ……293
11.4.2 TCP 的初始化 ……295
11.4.3 TCP 定时器 ……296
11.4.4 TCP 套接字的初始化 ……297
11.4.5 TCP 连接的建立 ……297
11.4.6 接收来自网络层（L3）的 TCP 数据包 ……298
11.4.7 发送 TCP 数据包 ……299
11.5 流控制传输协议（SCTP） ……300
11.5.1 SCTP 数据包和数据块 ……301
11.5.2 SCTP 块头 ……302
11.5.3 SCTP 块 ……302
11.5.4 SCTP 关联 ……303

11.5.5 建立 SCTP 关联 ····· 305
11.5.6 接收 SCTP 数据包 ····· 305
11.5.7 发送 SCTP 数据包 ····· 306
11.5.8 SCTP 心跳 ····· 306
11.5.9 SCTP 多流 ····· 306
11.5.10 SCTP 多宿主 ····· 307
11.6 数据报拥塞控制协议（DCCP） ····· 307
11.6.1 DCCP 报头 ····· 307
11.6.2 DCCP 的初始化 ····· 309
11.6.3 DCCP 套接字的初始化 ····· 310
11.6.4 接收来自网络层（L3）的 DCCP 数据包 ····· 311
11.6.5 发送 DCCP 数据包 ····· 311
11.6.6 DCCP 和 NAT ····· 312
11.7 总结 ····· 313
11.8 快速参考 ····· 313
11.8.1 方法 ····· 313
11.8.2 宏 ····· 315
11.8.3 表 ····· 315

第 12 章 无线子系统 ····· 317

12.1 mac80211 子系统 ····· 317
12.2 802.11 MAC 帧头 ····· 318
12.3 802.11 MAC 帧头的其他成员 ····· 320
12.4 网络拓扑 ····· 321
12.4.1 基础设施 BSS ····· 321
12.4.2 IBSS（对等模式） ····· 322
12.5 省电模式 ····· 322
12.5.1 进入省电模式 ····· 322
12.5.2 退出省电模式 ····· 322
12.5.3 处理组播/广播缓冲区 ····· 323
12.6 管理层 ····· 325
12.6.1 扫描 ····· 325
12.6.2 身份验证 ····· 325
12.6.3 关联 ····· 325
12.6.4 重新关联 ····· 325
12.7 mac80211 的实现 ····· 326
12.7.1 接收路径 ····· 328
12.7.2 传输路径 ····· 328
12.7.3 分段 ····· 329
12.7.4 mac80211 debugfs ····· 330
12.7.5 无线模式 ····· 331
12.8 高吞吐量（IEEE 802.11n） ····· 331
12.9 网状网络（802.11s） ····· 334
12.9.1 HWMP ····· 335
12.9.2 组建网状网络 ····· 336
12.10 Linux 无线开发流程 ····· 337
12.11 总结 ····· 337
12.12 快速参考 ····· 338
12.12.1 方法 ····· 338
12.12.2 表 ····· 341

第 13 章 InfiniBand ····· 343

13.1 RDMA 和 InfiniBand 概述 ····· 343
13.1.1 RDMA 栈的组织结构 ····· 344
13.1.2 RDMA 技术的优点 ····· 345
13.1.3 InfiniBand 硬件组件 ····· 345
13.1.4 InfiniBand 中的编址 ····· 345
13.1.5 InfiniBand 的功能 ····· 346
13.1.6 InfiniBand 数据包 ····· 346
13.1.7 管理实体 ····· 347
13.2 RDMA 资源 ····· 348
13.2.1 RDMA 设备 ····· 348
13.2.2 PD ····· 350
13.2.3 AH ····· 350
13.2.4 MR ····· 350
13.2.5 FMR 池 ····· 351
13.2.6 MW ····· 352
13.2.7 CQ ····· 352
13.2.8 XRC ····· 353
13.2.9 SRQ ····· 353
13.2.10 QP ····· 355
13.2.11 工作请求的处理 ····· 360
13.2.12 RDMA 架构支持的操作 ····· 361
13.2.13 组播组 ····· 365
13.2.14 用户空间 RDMA API 和内核级 RDMA API 的差别 ····· 365
13.3 总结 ····· 366
13.4 快速参考 ····· 366

第 14 章　高级主题……372
14.1　网络命名空间……372
14.1.1　命名空间的实现……373
14.1.2　UTS 命名空间的实现……381
14.1.3　网络命名空间的实现……383
14.1.4　网络命名空间的管理……388
14.2　cgroup……392
14.2.1　cgroup 的实现……393
14.2.2　cgroup 设备控制器：一个简单示例……395
14.2.3　cgroup 内存控制器：一个简单示例……396
14.2.4　net_prio 模块……396
14.2.5　分类器 cls_cgroup……397
14.2.6　挂载 cgroup 子系统……398
14.3　频繁轮询套接字……399
14.3.1　全局启用……400
14.3.2　对特定套接字启用……401
14.3.3　调整和配置……401
14.3.4　性能……401
14.4　Linux 蓝牙子系统……401
14.4.1　HCI 层……404
14.4.2　HCI 连接……406
14.4.3　L2CAP……407
14.4.4　BNEP……407
14.4.5　蓝牙数据包接收示意图……408
14.4.6　L2CAP 扩展功能……409
14.4.7　蓝牙工具……409
14.5　IEEE 802.15.4 和 6LoWPAN……410
14.5.1　邻居发现优化……411
14.5.2　Linux 内核的 6LoWPAN 实现……412
14.6　NFC……415
14.6.1　NFC 标签……415
14.6.2　NFC 设备……416
14.6.3　通信模式和操作模式……416
14.6.4　主机控制器接口……417
14.6.5　Linux 对 NFC 的支持……417
14.6.6　用户空间架构……421
14.6.7　Android NFC……421
14.7　通知链……422
14.8　PCI 子系统……425
14.9　组合网络设备……428
14.10　PPPoE 协议……428
14.10.1　PPPoE 报头……429
14.10.2　PPPoE 的初始化……430
14.10.3　PPPoE 数据包的收发……432
14.11　Android……435
14.11.1　Android 联网技术……436
14.11.2　Android 内部原理：资料……437
14.12　总结……438
14.13　快速参考……438
14.13.1　方法……438
14.13.2　宏……443
附录 A　Linux API……444
附录 B　网络管理……520
附录 C　术语表……537

第1章 绪 论

本书讨论Linux内核网络栈的实现及其原理，深入而详尽地分析网络子系统及其架构。为减轻读者压力，这里将不讨论在阅读内核网络栈源代码过程中可能遇到的但与网络没有直接关系的主题，如加锁与同步、SMP、原子操作等。有关这些主题的资料浩如烟海，然而，专门探讨内核网络的最新资料却少之又少。本书将重点讲解数据包在Linux内核网络栈中的传输过程，阐述其与网络各层及各子系统之间的交互，探讨各种网络协议的实现方法。

本书也不会不厌其烦地逐行解读代码，而将专注于各网络协议实现技术的精髓及其遵循的指导方针和原则。近年来的情况表明，Linux是一款成功、可靠、稳定而深受欢迎的操作系统，且受欢迎程度正稳步提升。Linux版本众多，有用于大型机、数据中心、核心路由器和Web服务器的版本，有用于无线路由器、机顶盒、医疗仪器、导航设备（如GPS设备）等嵌入式设备的版本，还有用于消费电子产品的版本。很多半导体厂商开发的板级支持包（Board Support Package，BSP）都基于Linux。Linux操作系统肇始于芬兰人Linus Torvalds于1991年开发的一个基于UNIX操作系统的项目。事实证明，它已成为一款严谨而可靠的操作系统，可与老牌专用操作系统相媲美。

Linux最初只是一款基于Intel x86的操作系统，现已移植到包括ARM、PowerPC、MIPS、SPARC等在内的各种处理器。Android操作系统是当前常见的平板电脑和智能手机操作系统，未来有望在智能电视领域大行其道，而这款操作系统正是基于Linux内核的。除Android操作系统外，Google还开发了一些内核网络功能。这些功能已纳入主流内核中。

Linux是个开源项目，因此相比于其他专用操作系统具有如下优势：遵照通用公共许可证（General Public License，GPL）条款，用户可免费获得其源代码。相对而言，其他开源操作系统（如各种类型的BSD）的普及程度则要低得多。这里有必要说说OpenSolaris项目。该项目基于通用开发与发布许可（Common Development and Distribution License，CDDL）协议，由Sun公司发起，但其受欢迎程度不可与Linux同日而语。在Linux开发大军中，有些人以公司的名义贡献代码，有些人自发地贡献代码。所有内核开发过程都可通过内核邮件列表获悉。Linux内核邮件列表（Linux Kernel Mailing List，LKML）为其核心邮件列表，很多子系统也都有专用的邮件列表。要贡献代码，可将补丁发送至相应的内核邮件列表及维护人员。这些补丁将通过邮件列表得到相关成员的讨论。

Linux内核网络栈是Linux内核中一个极其重要的子系统。在基于Linux的系统中，不使用任何网络功能的很少，无论是台式机、服务器、移动设备还是其他嵌入式设备都如此。即便在机器没

有任何硬件网络设备这种极其罕见的情况下，在用户使用X-Windows时也将使用到网络功能（虽然用户没有意识到这一点），因为X-Windows本身就是基于客户端-服务器网络的。与Linux网络栈相关的项目很多，从核心路由器到小型嵌入式设备。其中，有些项目致力于添加厂商特定的功能。例如，有些硬件厂商在一些网络设备中实现了通用分段延后处理功能（Generic Segmentation Offload，GSO）。GSO是内核网络栈的一项网络功能，由内核网络栈在传输路径中将大型数据包划分成小型数据包。很多硬件厂商都在其网络设备硬件中实现了校验和功能。校验和是一种验证机制。它计算数据包的散列值并将其附加到数据包中，以核实数据包在传输过程中未受损。很多项目都对Linux做了安全改进。其中的一些改进要求对网络子系统进行修改。在第3章讨论项目Openwall GNU/*/Linux时，你将看到这一点。在嵌入式设备领域，很多无线路由器都基于Linux。例如，Linksys WRT54GL路由器运行的就是Linux。这种设备（以及其他设备）还可运行基于Linux的开源操作系统OpenWrt。这款操作系统拥有庞大而活跃的开发人员社区，其网址为https://openwrt.org/。要更深入地了解Linux内核网络栈，必须明白它是如何实现各种协议的，同时还要熟悉主要的数据结构以及数据包在其中的主要传输路径。

1.1　Linux网络栈

开放系统互联（OSI）模型定义了7个逻辑网络层。最下面是物理层，即硬件环境。最上面是应用层，其中运行着用户空间软件进程。下面来说说这7层。

(1) *物理层*：提供电信号和一些底层的细节。

(2) *数据链路层*：处理端点间的数据传输。最常见的数据链路层标准是以太网。Linux以太网网络设备驱动程序就位于这一层。

(3) *网络层*：负责数据包转发和主机编址。本书讨论Linux内核网络子系统实现的最常见网络层协议：IPv4和IPv6。Linux还实现了其他不那么常见的网络层协议，如DECnet，但本书将不对其作出讨论。

(4) *协议层/传输层*：完成结点间的数据发送。TCP和UDP是最著名的传输层协议。

(5) *会话层*：处理端点间的会话。

(6) *表示层*：处理数据传送和格式设置。

(7) *应用层*：向最终用户应用程序提供网络服务。

图1-1显示了OSI模型定义的7个逻辑网络层。

图1-2显示了Linux内核网络栈所涉及的3层。其中，L2、L3和L4这三层分别对应于OSI 7层模型中的数据链路层、网络层和传输层。从本质上说，Linux内核栈的任务就是将接收到的数据包从L2（网络设备驱动程序）传递给L3（网络层，通常为IPv4或IPv6）。接下来，如果数据包目的地为当前设备，Linux内核网络栈就将其传递给L4（传输层，应用TCP或UDP协议侦听套接字）；如果数据包需要转发，就将其交还给L2进行传输。对于本地生成的出站数据包，将从L4依次传递给L3和L2，再由网络设备驱动程序进行传输。这个过程分很多阶段，期间可能会发生如下行为。