

Linux Routers: A Primer for Network Administrators (2nd Edition)

By Tony Mancill

Linux Routers, Second Edition shows you exactly how to reduce your costs and extend your network with Linux-based routing. You'll find step-by-step coverage of software/hardware selection, configuration, management, and troubleshooting for today's key internetworking applications, including LANs, Internet/intranet/extranet routers, Frame Relay, VPNs, remote access, and firewalls. Extensive new coverage includes dynamic routing, Quality of Service, the current Linux kernel - even next-generation IPv6 routing.

- [Entrepreneurial Financial Management](#)
- [Logistics](#)
- [Microeconomic Theory: An Integrated Approach](#)
- [Marketing Management: Millennium Edition \(10th Edition\)](#)
- [Internet Security Protocols: Protecting IP Traffic](#)
- [Complete Y2K Home Preparation Guide, The](#)
- [Linux Shells by Example \(with CD-ROM\)](#)
- [Modern Spectral Estimation: Theory and Application](#)
- [Parents and Families of Children with Disabilities: Effective School-Based Support Services](#)
- [Probability and Random Processes with Applications to Signal Processing \(3rd Edition\)](#)
- [Quantitative Analysis for Management \(7th Edition\)](#)
- [Robust Modulation Methods and Smart Antennas in Wireless Communications](#)
- [Wideband Wireless Digital Communications](#)
- [Ranking Task Exercises in Physics](#)
- [Win32 System Services: The Heart of Windows 98 and Windows 2000 \(3rd Edition\)](#)
- [PACEMAKER WORLD HISTORY TEACHER'S EDITION 2002C \(Pacemaker World History Pacemaker\)](#)
- [PACEMAKER UNITED STATES HISTORY AUDIO CD PROGRAM AND TEACHER EDITION 2004 \(Fearon Us History\)](#)
- [PACEMAKER AMERICAN LITERATURE CLASSROOM RESOURCE BINDER 2005C \(Fearon's American Literature\)](#)
- [Red Hat Linux System Administration Handbook](#)
- [Principles of Field Crop Production \(4th Edition\)](#)

Linux Routers: A Primer for Network Administrators (2nd Edition) Summary Details

Linux Routers: A Primer for Network Administrators (2nd Edition) by By Tony Mancill ebook read online.
pdetail:

- Rank: #1291845 in Books
- Published on: 2002-06-21
- Released on: 2002-06-11
- Original language: English
- Number of items: 1
- Dimensions: 9.06" h x 1.02" w x 6.93" l, 1.65 pounds
- Binding: Paperback
- 464 pages

editorial:

From the Back Cover

Save a fortune: Use Linux to deliver cost-effective, reliable routing services.

Linux routers are inexpensive, flexible, stable, adaptable, expandable, easy to manage, and based on proven technology. In *Linux Routers, Second Edition*, Tony Mancill shows you exactly how to configure, administer, and troubleshoot Linux routing for today's most common internetworking applications. Thoroughly updated for the latest technologies and version 2.4 of the Linux kernel, this edition includes practical coverage of dynamic routing, Quality of Service (QoS)—even next-generation IPv6 routing. Mancill's step-by-step explanations walk you through:

- Choosing the right Linux software distribution and hardware platform
- Building Ethernet LAN routers, including coverage of the Linux Router Project (LRP)
- Implementing Internet, extranet, and Frame Relay routing
- Creating VPN tunneling routers with iproute2 and FreeS/WAN
- Providing remote access to satellite offices while conserving bandwidth
- Using Linux IP firewalling features to improve network security
- Providing routing services alongside Web and DNS services on a single platform
- Monitoring Linux routers and maximizing availability in production environments

Whether your goal is to reduce network costs, add applications, solve problems you can't solve with "traditional" routers, or simply learn about routing hands-on without investing expensive, proprietary network gear, *Linux Routers, Second Edition* is all you need to get the job done.

About the Author

TONY MANCILL works for Vesta, a stored-value services company in Portland, Oregon. He has worked in several large IT shops as a UNIX systems administrator and systems programmer, including Bank of America and BellSouth, and he has been running Linux in production corporate environments since 1996.

Mancill is also active in the Free Software community as a volunteer developer for the Debian project (<http://www.debian.org/>). He is a graduate of Georgia Tech with a degree in Electrical Engineering.

Excerpt. © Reprinted by permission. All rights reserved.

Introduction

Router configuration has long been an arcane art possessed by the few (and the lucky—at least in the eyes of the interested yet unchosen). One reason for this is that routers were expensive, and required specialized training. They were also found only in small numbers in larger companies. Unix, while not quite as inaccessible in terms of cost, has also frequently been considered a black art for the few. Linux, of course, has turned all of this on its ear and delivered a GNU operating system to the masses. There used to be only a few big-city bookstores that had a decent selection of Unix books. Now every mall bookstore has a Unix/Linux section.

Routers, although not quite ubiquitous, are also much more prevalent in recent times too. The explosive growth of the Internet has every business, large and small, scrambling to get connected. Even completely nontechnical organizations POP their email from their ISP over a dial-up link several times a day. And people use more networked computers in general nowadays. People even have routers in their homes, perhaps to provide Internet connectivity via a cable modem to multiple computers in the home.

Interestingly (but not surprisingly), Linux delivers a router platform accessible by the many, almost as easily as it has resurrected the word "Unix." Although what folks think of as "traditional" routers are special-purpose hardware running embedded real-time operating systems, there has always been a group using Unix-based operating systems running on general-purpose hardware. Linux has helped expand this group immensely by providing a full-featured and robust TCP/IP stack and hundreds of device drivers—all of which run on commonly available computing hardware. Another important factor is the GNU/Linux community's basic tenet of sharing ideas and knowledge instead of hoarding them. The truly amazing part is that all of this is available without expensive licensing costs or nondisclosure agreements. With these good tools and some know-how, anyone can build an inexpensive, stable router platform capable of performing a variety of functions within any organization.

If you're asking yourself, *What's a router?*, do not despair! This book will cover most topics in enough depth and with enough references that you will not be left stumped. If it fails to do this, please let me know. Traditionally, a router is defined as a layer 3 switch. That is, it looks at the layer 3 packet headers and decides where to forward them. (Layer 3 refers to the OSI "network" layer.) For TCP/IP networks, the router examines the IP headers to determine the destination IP address of the packet and the TTL (Time To Live), among other things. In this strict definition, the type of packet—TCP, UDP, ICMP, etc.—is immaterial. However, as routers accumulate functionality, they can now take into account the layer 4 (transport layer) headers (as well as those above layer 4 and below layer 2) when deciding what to do with a packet. Some other things that routers do:

- Transfer traffic between two different types of physical networks. An example of this is forwarding packets from Ethernet to Frame Relay and vice versa, as is done by a WAN router.
- Restrict the flow of traffic to/from certain addresses or networks. This is commonly known as **packet filtering**. Often the router will examine more than just the IP headers to make filtering decisions. TCP and UDP headers and other layer 4+ packet fields are fair game.
- Rewrite source addresses so that traffic appears to originate from an address other than that of the original sender. Under Linux, this is accomplished through **masquerading** and **network address translation** (or

NAT). The task includes unmasquerading replies flowing in the other direction.

- Act as a BOOTP or DHCP server issuing IP addresses and other configuration parameters for workstations on your network. Throughout this book we will use Linux routers to perform these tasks and much more.

The focus of this book is admittedly not on academic definitions, but on how to accomplish (or at least get started on) a certain task. You may find that knowing the correct technical definition for something is not the same as successfully configuring and administering it in a production environment. If you're just getting started as a network administrator, be confident that a lot of technical know-how can be learned only through experience (a.k.a. "the hard way"). Do not be intimidated by the lack of a formal body of knowledge; just admit it to yourself when you don't know the answer and strive to figure it out.

From Whence Linux?

The progress on operating-systems implementation, in my opinion, has been quite dramatic during the 1990s thanks to Linux. If you are interested in the "early" history of Linux, the Ur-history, and can stomach discussions of OS design, you should check out the following:

http://www.educ.umu.se/A_bjorn/mhonarc-files/obsolete/

This is a thread that features a heated exchange between Linus Torvalds and Andy Tanenbaum, a well-established researcher, professor, father of MINIX, and author of several (good) OS-design books. (I read somewhere that Linus Torvalds isn't particularly proud of the bravado he displayed in these messages. I find the characteristic of being passionate about something and then later being able to admit having been caught up in the heat of the moment, and to rise above it, to be magnanimous indeed.)

http://www.educ.umu.se/A_bjorn/linux/misc/linux-history.html

This is a collection of Linux-relevant postings made to comp.os.minix about building an OS.

The way I understand the early history of Linux, it boils down to this: Linus Torvalds had an Intel 80386-based PC and some free time. He set out to write an operating system that would capitalize on the new features of this processor, ignoring backward compatibility with the 8088 and the 80286. (Other PC operating systems of that era were struggling with some design trade-offs that were imposed by the limitations of the 386's predecessors. In all fairness, they did this to maintain backward compatibility with software written to run on those systems.)

Another incentive for writing his own OS from scratch was to take advantage of the large body of excellent "free" software being developed by the Free Software Foundation. This is the (first) connection between Linux and Richard Stallman, who founded the GNU project in 1984 and has lent both philosophy and code to the effort. The system software which Linus used for his new kernel is commonly known as the GNU toolset. These tools implement most of what is commonly thought of as standard "Unix" system tools. Remember, "GNU's Not Unix." This collection of software had the tremendous advantage of being available in source-code form and being delivered with a license that allowed early Linux developers to modify that source and redistribute it.

In the early 1990s, the Internet made the transition from an academic/scientific tool to the most powerful communications medium ever available to the general public. The ability to communicate, collaborate, share, discuss, and debate has never before been available to such a large percentage of the world's population. Why the sentimentality about the Internet? If Linus Torvalds conceived Linux, and it spent a while in the "womb" of academia, then the Internet is its home—where it will play, grow, and mature. But Linux has become more than a user of the Internet; it has become part of it, and thus is now being used to propagate

itself.

Why a Linux Router?

At this point, you may be thinking that this is all well and good and apropos in some sort of cosmic sense—the child has grown up to take care of the parent, etc., and so on—but what makes a Linux router any better or different than other routers? As with most interesting questions, the answer is: "It depends." Primarily, it depends upon your definition of "better" is.

- *Linux routers are inexpensive.* For about \$500 plus a meagerly equipped PC, you have a WAN router capable of running several different protocols at T1 (1.5Mbps) speeds and higher. For a bit more, you can route multiple T3s. License costs are typically nil. Ongoing hardware maintenance costs are nil to minimal. Linux runs on an amazing variety of hardware platforms, from the latest low-cost commodity gear to systems that haven't been manufactured in a decade.
- *Linux routers are flexible.* Try running a copy of Apache (the world's most popular HTTP daemon) on a "traditional" router, or integrating a "traditional" router into your in-house-developed security infrastructure. At best, you will pay your router vendors handsomely for the privilege of letting them port your code to their systems. Otherwise, you're probably out of luck.
- *Linux routers are stable.* Whereas the TCP/IP implementation of many others has been reviewed by, at most, a few hundred capable programmers, Linux's TCP/IP stack has been reviewed by literally thousands. (It even has its own book; see *TCP/IP and Linux Protocol Implementation* in the Bibliography.) And because Linux is a "labor of love" for most of its contributors, its quality is based on pride and technical excellence, not hampered by deadlines and office politics.
- *Linux routers are easy to administer.* If you can navigate a shell (you can pick the shell of your choice), you have already completed level one of your router training. With other solutions, you need to learn a new command environment and set of tools. Linux is the same whether it's acting as a web server, workstation, or router.
- *Linux routers are based on proven, widely available technology.* Because the system hardware and adapters are being produced for such an enormous market, costs are low and time-to-market cycles are as short as possible.
- *Linux routers provide investment protection beyond that of their "traditional" counterparts.* If you've worked in computer support for any length of time, you've probably experienced a vendor's discontinuing a product line, phasing out support for a particular version, failing to add features that you desperately need, or simply not responding to bug reports. With Linux, there is no vendor who can strong-arm you into accepting any of these frustrations. You always have the option of modifying the source yourself or choosing not to upgrade and remaining at the version which suits you best.
- *Linux routers are expandable.* Because you can use almost anything that you can attach to a general-purpose computer, you never have to worry about whether or not your router chassis can support a third BRI ISDN adapter. Just add a serial expansion card and attach the third adapter.
- *Linux routers are adaptable.* One thing the computer industry has shown us by now (or taught us the hard way) is to expect change. Where is IBM's MCA? Whatever happened to 100VG-LAN? Did VoIP (Voice over IP) really ever catch on? The next big thing to hit the market will be ?? Since a Linux router is not a proprietary "point-in-time" solution, you can retrofit to meet your technology needs without replacing your entire router hardware.

Goals of This Book

This book aims to promote Linux as a router by telling you how to set up and manage your own Linux router based on my experiences using them in production environments. Different "types" of routers are illustrated through a series of examples, each of which introduces the functionality and configuration of that

router. For instance, Chapter 3 describes a router used to route traffic between two LAN subnets, while Chapter 5 details the configuration of a Frame Relay (WAN) router. Because the fine details of "how" change continually, every attempt is made to maintain a level of detail that will be useful for more than just the next point-release of the kernel. (However, the specific examples in this, the second edition of the book, are for the 2.4.x kernel series. See Chapter 10 for information on prior kernels.)

Alongside the configurations are general discussions about running Linux in production, as well as coverage of some applications that help support the network infrastructure (like traffic analysis and system monitoring). For me, Linux is about enabling people to do things that they would not have been able to do or afford to do otherwise. Really. I mean that quite sincerely. Sometimes "afford" means more than having sufficient money—it means having sufficient time. Linux can help you craft tailor-made solutions by building on the work of others. The book also contains some background information suited for "newbie" administrators who have never done network administration. While this information will be familiar to experienced network administrators, I feel that it's important, because technical elegance is pointless without sound administration practices, or if you don't even know where to start.

To exhibit what I believe is the "true spirit" of Linux—sharing information, ideas, and accomplishments instead of hiding them—references to publicly available resources will be made whenever I am aware of such sources. Several of these are collected as a list of links in Appendix . Much of what you read in this book is made possible by the existence of openly available documentation. Another element of this spirit is the attitude that "whatever it is I need to accomplish, Linux is a sufficient toolset to succeed." Having faith in your toolset and in your ability to use it effectively to solve business problems is crucial to your performing your work well and enjoying it, too. This book aims to help you establish some faith in the toolset by offering knowledge about how to use it. It will help you find ways in which you can:

- Reduce the total cost of ownership (TCO) of your current environment. Very few businesses are so wealthy that they disregard the bottom line.
- Deploy Linux routers in organizations which might not be able to afford the "traditional" hardware or the associated license and maintenance costs.
- Solve problems not addressed by "traditional" routers.
- Keep your Linux production environment healthy, saving yourself time and unexpected trips into the office.

The Linux Documentation Project

The Linux Documentation Project, or LDP as it's commonly known, is an invaluable resource for Linux documentation. The project aims to make high-quality documentation freely available to all. This means not only covering a wide range of topics at various depths and skill levels, but also making sure that the documentation is current and correct.

This documentation covers a wide variety of topics, from general hardware compatibility to specialized software configurations. The documentation takes several forms, from the informal FAQs and mini-HOWTOs to the regular HOWTOs and the longer guides. If you're not already familiar with this site, bookmark it—I will refer throughout this book to documentation found there.

<http://www.linuxdoc.org>

Nongoes of This Book

This book does not aspire to incite the masses to riot and replace all "traditional" routers with Linux systems.

(Throughout this book the term "traditional" should be taken to mean router product offerings based on a closed, proprietary system interfaces or "operating systems." This would include the fine family of Cisco products, Nortel, Ascend Communications, et al.) These products have their place, and they have paved the way for the existence of Linux and the Linux router. In many instances, depending upon either the specific application or your corporate culture, traditional routers are the right choice for your production environment. On the other hand, a sound business principle is to pick the least expensive tool capable of satisfying all of the operational requirements—the tool that will result in the lowest total cost of ownership. Realistically, cost is the overriding factor for most business ventures, and Linux-based solutions will often have the advantage.

This book does not attempt to thoroughly teach TCP/IP or fully explain routing protocols used in TCP/IP networks. Nor does the book thoroughly address firewall configuration or security. These topics are simply beyond the scope of this book, and other people have done fantastic jobs of covering them (much better than I could). I do heartily recommend a thorough understanding of TCP/IP and networking in general for anyone considering administering a TCP/IP network, regardless of the equipment used. For an authoritative treatment of networking and routing, see *Interconnections* by Radia Perlman. Finally, the book you now hold covers only routing with TCP/IP (although Linux is capable of handling other protocols). If you need to support something other than IP, know that you are not alone, and that much of the software and hardware specifically supports protocols besides IP.

Layout of This Book

Chapters and discuss topics which are generic and applicable to all Linux routers, while Chapters - each cover a type of Linux router as I have used it in production and highlight what I learned in the process. (There should probably be a chapter called "Franken-router" for the router configurations that did not work—a HOWNOTTO of sorts. Adventurous readers will be able to write this chapter themselves.... :)) Of the latter seven chapters, the first three are basic LAN, extranet, and WAN routers, while the remaining four address more advanced topics such as connecting to the Internet and making routers do more than just route.

Within each router chapter, first the router is introduced by its function, and then technical details of configuration are discussed. Throughout the book, several threads of related topics are presented and developed alongside the routers themselves. These topics include choice of hardware, choice of software, strategies for high availability, TCP/IP niceties of Linux available in the kernel, and monitoring configurations and software. Different chapters have different components and requirements, but generally have the following layout:

- hardware
- software required and configuration
- kernel configuration
- support/high availability
- monitoring

The appendices address topics either too general or too specific for the main text, from how to compile the Linux kernel (Appendix) to using VMware to augment your testing strategies (Appendix) to ethical considerations of a network admin. Finally, there may not be a single "flavor" of router presented in this book that fits your application. If that is the case, do not shy away from combining configurations presented here into something new.

A Note to 2.0.x and 2.2.x Kernel Users

Initially, the first edition of this text encompassed the 2.0.x and the 2.2.x versions of the kernel. As that book went through various stages of editing (and time passed), it became clear that the main text should appeal to the broadest userbase without forcing the reader to switch back and forth between the details of two different kernel versions. This is particularly true for the sections on masquerading and firewalling (Chapters 4, 7, and 9).

In the second edition the focus is on the 2.4 kernel, since all of the major Linux distributions at the time of writing run on 2.4 kernels. However, notes for 2.0 and 2.2 kernel users remain and are scattered throughout the text, including many of the first edition materials for the 2.2.x kernels collected in Chapter 10. If you support an environment running on an older kernel and need assistance, check <http://mancill.com/linuxrouters/>

Naming of Chapters

You might have already noticed a common thread in the names of the chapters in this book. Maybe they make you want to check the cover to make sure that this isn't a science textbook. Selecting a naming motif for Unix systems is a tradition that seems to have been around for quite a while. Perhaps one of the hacker subculture deities knows its origin. (See *The New Hackers Dictionary* by Eric Raymond if this sort of trivia interests you.) If you haven't already guessed, I've named my routers after elements in the periodic table.

The periodic table has characteristics that render it useful as a system naming scheme. First of all, there is a large pool of names, many of which sound cool. Next, every name has a short yet unique "symbol name" associated with it, e.g., *gold* is **Au**, *carbon* is **C**. These can be used as a sort of shorthand, along with aliases (CNAMEs) in DNS to save keystrokes.

Linux Routers: A Primer for Network Administrators (2nd Edition) by By Tony Mancill epub PDF read Online Download.

Linux Routers: A Primer for Network Administrators (2nd Edition) by By Tony Mancill Reader Review Online

Linux Routers, Second Edition shows you exactly how to reduce your costs and extend your network with Linux-based routing. You'll find step-by-step coverage of software/hardware selection, configuration, management, and troubleshooting for today's key internetworking applications, including LANs, Internet/intranet/extranet routers, Frame Relay, VPNs, remote access, and firewalls. Extensive new coverage includes dynamic routing, Quality of Service, the current Linux kernel - even next-generation IPv6 routing.

Linux Routers: A Primer for Network Administrators (2nd Edition) by By Tony Mancill ebook PDF online