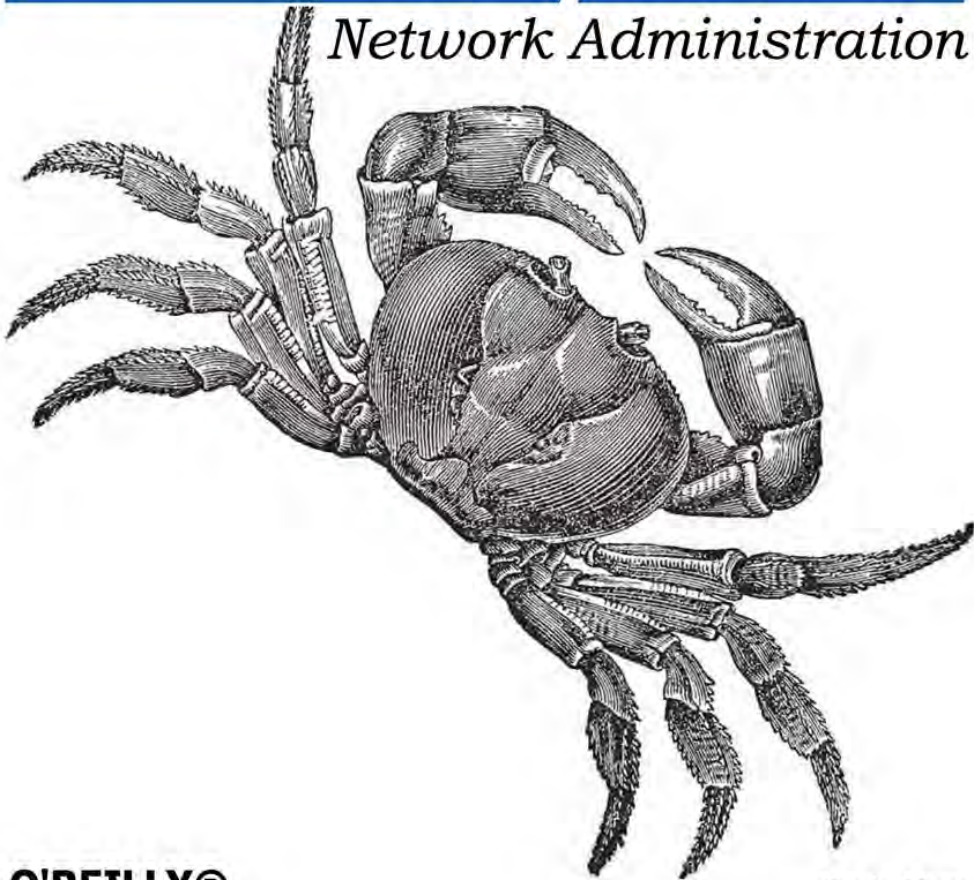


Help for Unix System Administrators

3rd Edition

TCP/IP

Network Administration



O'REILLY®

Craig Hunt

O'REILLY'

TCP/IP Network Administration



TCP/IP Network Administration, Third Edition, is a complete guide to setting up and running a TCP/IP network, and is geared toward system administrators as well as users of home systems that access the Internet. It starts with the fundamentals: what protocols do and how they work, how addresses and routing are used to move data through the network, and how to set up your network connection.

Beyond basic setup, this book discusses advanced routing protocols (RIPv2, OSPF, and BGP) and the gated software package that implements them. It provides a tutorial on configuring important network services, including DNS, Apache, sendmail, Samba, PPP, and DHCP. There are chapters on troubleshooting and security. In addition, this book contains a command and syntax reference for important packages such as gated, pppd, named, dhcpcd, and sendmail.

This new edition includes a section on configuring Samba to provide file and print sharing on networks that integrate Unix and Windows, and a new chapter dedicated to the important task of configuring the Apache web server. Network security coverage is expanded to include details on OpenSSH, stunnel, gpg, iptables, and the access control mechanism in xinetd. This book also contains updated information about DNS, including details on BIND 8 and BIND 9, the role of classless IP addressing and network prefixes, and the changing role of registrars.

This book covers Linux, Solaris, BSD, and System V TCP/IP implementations.

Praise for previous editions:

"The book you reach for first..."

—Marshall Rose, *ConneXions*

"...the definitive volume on the subject."

—Tom Yager, *BYTE*

"...probably the best single Unix TCP/IP system administrator's handbook in print..."

—Anthony M. Rutkowski, *SprintLink*

"The second edition of Hunt's superb book is even more useful and informative than the original edition... an extraordinary and outstanding revision of a classic and indispensable reference."

—Elizabeth Zinkann, *Sys Admin*

www.oreilly.com

US \$44.95

CAN \$69.95

ISBN: 978-0-596-00297-8



TCP/IP Network Administration

Download from Wow! eBook <www.wowebook.com>

THIRD EDITION

TCP/IP Network Administration

Craig Hunt

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

TCP/IP Network Administration, Third Edition

by Craig Hunt

Copyright © 2002, 1998, 1992 Craig Hunt. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly Media, Inc. books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (safari.oreilly.com). For more information contact our corporate/institutional sales department: (800) 998-9938 or corporate@oreilly.com.

Editors: Mike Loukides and Debra Cameron

Production Editor: Emily Quill

Cover Designer: Edie Freedman

Interior Designer: Melanie Wang

Printing History:

August 1992: First Edition.

January 1998: Second Edition.

April 2002: Third Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *TCP/IP Network Administration, Third Edition*, the image of a land crab, and related trade dress are trademarks of O'Reilly Media, Inc. Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.



This book uses RepKover[™], a durable and flexible lay-flat binding.

ISBN: 978-0-596-00297-8

[C]

[10/08]

Table of Contents

Preface	xi
1. Overview of TCP/IP	1
TCP/IP and the Internet	2
A Data Communications Model	6
TCP/IP Protocol Architecture	9
Network Access Layer	11
Internet Layer	12
Transport Layer	18
Application Layer	22
Summary	23
2. Delivering the Data	24
Addressing, Routing, and Multiplexing	24
The IP Address	25
Internet Routing Architecture	35
The Routing Table	37
Address Resolution	43
Protocols, Ports, and Sockets	44
Summary	50
3. Network Services	51
Names and Addresses	51
The Host Table	52
DNS	54
Mail Services	62
File and Print Servers	75
Configuration Servers	76
Summary	82

4. Getting Started	84
Connected and Non-Connected Networks	85
Basic Information	86
Planning Routing	97
Planning Naming Service	101
Other Services	104
Informing the Users	106
Summary	107
5. Basic Configuration	108
Kernel Configuration	108
Startup Files	124
The Internet Daemon	129
The Extended Internet Daemon	132
Summary	133
6. Configuring the Interface	134
The ifconfig Command	134
TCP/IP Over a Serial Line	150
Installing PPP	153
Summary	169
7. Configuring Routing	170
Common Routing Configurations	170
The Minimal Routing Table	171
Building a Static Routing Table	173
Interior Routing Protocols	178
Exterior Routing Protocols	188
Gateway Routing Daemon	191
Configuring gated	193
Summary	204
8. Configuring DNS	205
BIND: Unix Name Service	205
Configuring the Resolver	207
Configuring named	211
Using nslookup	228
Summary	232

9. Local Network Services	233
The Network File System	233
Sharing Unix Printers	252
Using Samba to Share Resources with Windows	259
Network Information Service	268
DHCP	272
Managing Distributed Servers	277
Post Office Servers	280
Summary	283
10. sendmail	285
sendmail's Function	285
Running sendmail as a Daemon	286
sendmail Aliases	288
The sendmail.cf File	290
sendmail.cf Configuration Language	297
Rewriting the Mail Address	309
Modifying a sendmail.cf File	319
Testing sendmail.cf	323
Summary	332
11. Configuring Apache	333
Installing Apache Software	334
Configuring the Apache Server	338
Understanding an httpd.conf File	341
Web Server Security	361
Managing Your Web Server	378
Summary	380
12. Network Security	381
Security Planning	382
User Authentication	387
Application Security	402
Security Monitoring	404
Access Control	409
Encryption	418
Firewalls	425
Words to the Wise	433
Summary	434

13. Troubleshooting TCP/IP	435
Approaching a Problem	435
Diagnostic Tools	438
Testing Basic Connectivity	440
Troubleshooting Network Access	443
Checking Routing	450
Checking Name Service	456
Analyzing Protocol Problems	471
Protocol Case Study	474
Summary	478
A. PPP Tools	479
B. A gated Reference	503
C. A named Reference	548
D. A dhcpd Reference	586
E. A sendmail Reference	599
F. Solaris httpd.conf File	661
G. RFC Excerpts	679
<u>Index</u>	687

Transport Layer

The protocol layer just above the Internet Layer is the *Host-to-Host Transport Layer*, usually shortened to *Transport Layer*. The two most important protocols in the Transport Layer are *Transmission Control Protocol* (TCP) and *User Datagram Protocol* (UDP). TCP provides reliable data delivery service with end-to-end error detection and correction. UDP provides low-overhead, connectionless datagram delivery service. Both protocols deliver data between the Application Layer and the Internet Layer. Applications programmers can choose whichever service is more appropriate for their specific applications.

User Datagram Protocol

The User Datagram Protocol gives application programs direct access to a datagram delivery service, like the delivery service that IP provides. This allows applications to exchange messages over the network with a minimum of protocol overhead.

UDP is an unreliable, connectionless datagram protocol. As noted, “unreliable” merely means that there are no techniques in the protocol for verifying that the data reached the other end of the network correctly. Within your computer, UDP will deliver data correctly. UDP uses 16-bit *Source Port* and *Destination Port* numbers in word 1 of the message header to deliver data to the correct applications process. Figure 1-8 shows the UDP message format.

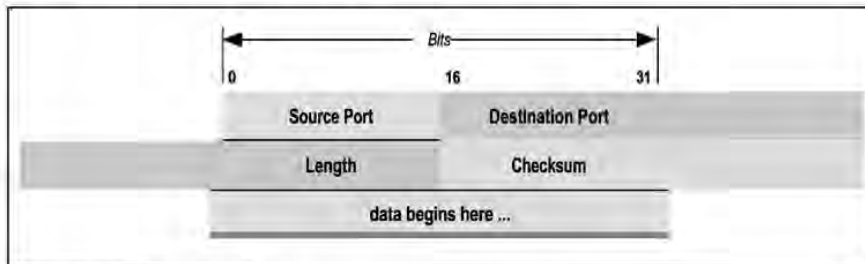


Figure 1-8. UDP message format

Why do applications programmers choose UDP as a data transport service? There are a number of good reasons. If the amount of data being transmitted is small, the overhead of creating connections and ensuring reliable delivery may be greater than the work of re-transmitting the entire data set. In this case, UDP is the most efficient choice for a Transport Layer protocol. Applications that fit a *query-response* model are also excellent candidates for using UDP. The response can be used as a positive acknowledgment to the query. If a response isn't received within a certain time period, the application just sends another query. Still other applications provide their own techniques for reliable data delivery and don't require that service from the

Transport Layer protocol. Imposing another layer of acknowledgment on any of these types of applications is inefficient.

Transmission Control Protocol

Applications that require the transport protocol to provide reliable data delivery use TCP because it verifies that data is delivered across the network accurately and in the proper sequence. TCP is a *reliable, connection-oriented, byte-stream* protocol. Let's look at each of these characteristics in more detail.

TCP provides reliability with a mechanism called *Positive Acknowledgment with Retransmission* (PAR). Simply stated, a system using PAR sends the data again *unless* it hears from the remote system that the data arrived OK. The unit of data exchanged between cooperating TCP modules is called a *segment* (see Figure 1-9). Each segment contains a checksum that the recipient uses to verify that the data is undamaged. If the data segment is received undamaged, the receiver sends a *positive acknowledgment* back to the sender. If the data segment is damaged, the receiver discards it. After an appropriate timeout period, the sending TCP module re-transmits any segment for which no positive acknowledgment has been received.

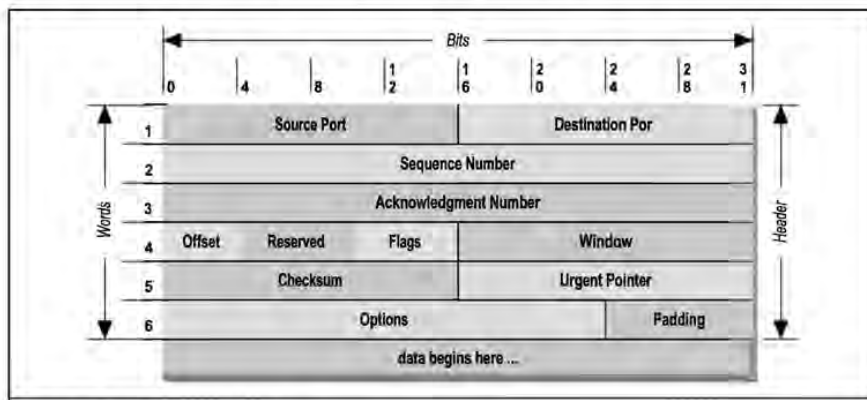


Figure 1-9. TCP segment format

TCP is connection-oriented. It establishes a logical end-to-end connection between the two communicating hosts. Control information, called a *handshake*, is exchanged between the two endpoints to establish a dialogue before data is transmitted. TCP indicates the control function of a segment by setting the appropriate bit in the Flags field in word 4 of the segment header.

The type of handshake used by TCP is called a *three-way handshake* because three segments are exchanged. Figure 1-10 shows the simplest form of the three-way handshake. Host A begins the connection by sending host B a segment with the "Synchronize sequence numbers" (SYN) bit set. This segment tells host B that A wishes to set

up a connection, and it tells *B* what sequence number host *A* will use as a starting number for its segments. (Sequence numbers are used to keep data in the proper order.) Host *B* responds to *A* with a segment that has the “Acknowledgment” (ACK) and SYN bits set. *B*’s segment acknowledges the receipt of *A*’s segment, and informs *A* which sequence number host *B* will start with. Finally, host *A* sends a segment that acknowledges receipt of *B*’s segment, and transfers the first actual data.

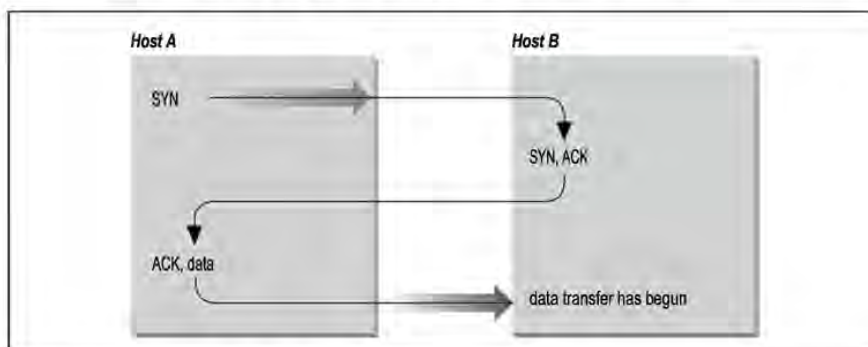


Figure 1-10. Three-way handshake

After this exchange, host *A*’s TCP has positive evidence that the remote TCP is alive and ready to receive data. As soon as the connection is established, data can be transferred. When the cooperating modules have concluded the data transfers, they will exchange a three-way handshake with segments containing the “No more data from sender” bit (called the FIN bit) to close the connection. It is the end-to-end exchange of data that provides the logical connection between the two systems.

TCP views the data it sends as a continuous stream of bytes, not as independent packets. Therefore, TCP takes care to maintain the sequence in which bytes are sent and received. The Sequence Number and Acknowledgment Number fields in the TCP segment header keep track of the bytes.

The TCP standard does not require that each system start numbering bytes with any specific number; each system chooses the number it will use as a starting point. To keep track of the data stream correctly, each end of the connection must know the other end’s initial number. The two ends of the connection synchronize byte-numbering systems by exchanging SYN segments during the handshake. The Sequence Number field in the SYN segment contains the *Initial Sequence Number* (ISN), which is the starting point for the *byte-numbering system*. For security reasons the ISN should be a random number.

Each byte of data is numbered sequentially from the ISN, so the first real byte of data sent has a Sequence Number of ISN+1. The Sequence Number in the header of a data segment identifies the sequential position in the data stream of the first data byte in

the segment. For example, if the first byte in the data stream was sequence number 1 (ISN=0) and 4000 bytes of data have already been transferred, then the first byte of data in the current segment is byte 4001, and the Sequence Number would be 4001.

The Acknowledgment Segment (ACK) performs two functions: *positive acknowledgment* and *flow control*. The acknowledgment tells the sender how much data has been received and how much more the receiver can accept. The Acknowledgment Number is the sequence number of the next byte the receiver expects to receive. The standard does not require an individual acknowledgment for every packet. The acknowledgment number is a positive acknowledgment of all bytes up to that number. For example, if the first byte sent was numbered 1 and 2000 bytes have been successfully received, the Acknowledgment Number would be 2001.

The Window field contains the *window*, or the number of bytes the remote end is able to accept. If the receiver is capable of accepting 6000 more bytes, the window would be 6000. The window indicates to the sender that it can continue sending segments as long as the total number of bytes that it sends is smaller than the window of bytes that the receiver can accept. The receiver controls the flow of bytes from the sender by changing the size of the window. A zero window tells the sender to cease transmission until it receives a non-zero window value.

Figure 1-11 shows a TCP data stream that starts with an Initial Sequence Number of 0. The receiving system has received and acknowledged 2000 bytes, so the current Acknowledgment Number is 2001. The receiver also has enough buffer space for another 6000 bytes, so it has advertised a window of 6000. The sender is currently sending a segment of 1000 bytes starting with Sequence Number 4001. The sender has received no acknowledgment for the bytes from 2001 on, but continues sending data as long as it is within the window. If the sender fills the window and receives no acknowledgment of the data previously sent, it will, after an appropriate timeout, send the data again starting from the first unacknowledged byte.

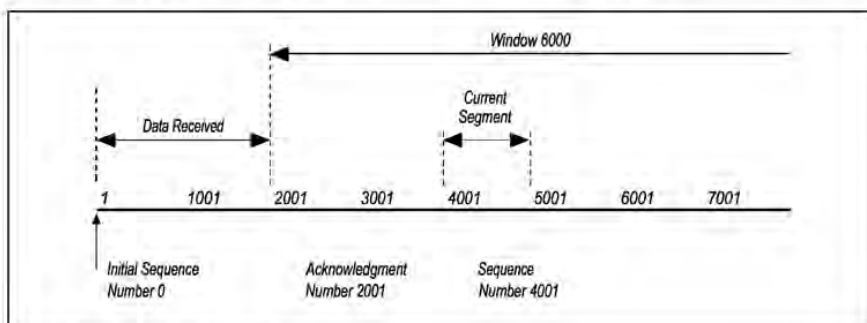


Figure 1-11. TCP data stream

In Figure 1-11 re-transmission would start from byte 2001 if no further acknowledgments are received. This procedure ensures that data is reliably received at the far end of the network.

TCP is also responsible for delivering data received from IP to the correct application. The application that the data is bound for is identified by a 16-bit number called the *port number*. The *Source Port* and *Destination Port* are contained in the first word of the segment header. Correctly passing data to and from the Application Layer is an important part of what the Transport Layer services do.

Application Layer

At the top of the TCP/IP protocol architecture is the *Application Layer*. This layer includes all processes that use the Transport Layer protocols to deliver data. There are many applications protocols. Most provide user services, and new services are always being added to this layer.

The most widely known and implemented applications protocols are:

Telnet

The Network Terminal Protocol, which provides remote login over the network.

FTP

The File Transfer Protocol, which is used for interactive file transfer.

SMTP

The Simple Mail Transfer Protocol, which delivers electronic mail.

HTTP

The Hypertext Transfer Protocol, which delivers web pages over the network.

While HTTP, FTP, SMTP, and Telnet are the most widely implemented TCP/IP applications, you will work with many others as both a user and a system administrator. Some other commonly used TCP/IP applications are:

Domain Name System (DNS)

Also called *name service*, this application maps IP addresses to the names assigned to network devices. DNS is discussed in detail in this book.

Open Shortest Path First (OSPF)

Routing is central to the way TCP/IP works. OSPF is used by network devices to exchange routing information. Routing is also a major topic of this book.

Network File System (NFS)

This protocol allows files to be shared by various hosts on the network.

Some protocols, such as Telnet and FTP, can be used only if the user has some knowledge of the network. Other protocols, like OSPF, run without the user even knowing that they exist. As the system administrator, you are aware of all these