

Understanding Network Packets, Protocols, and Architectures



Packet Guide to

Core Network Protocols

O'REILLY®

Bruce Hartpence

www.allitebooks.com

Packet Guide to Core Network Protocols

Take an in-depth tour of core Internet protocols and learn how they work together to move data packets from one network to another. With this concise book, you'll delve into the aspects of each protocol, including operation basics and security risks, and learn the function of network hardware such as switches and routers.

Ideal for beginning network engineers, each chapter in this book includes a set of review questions, as well as practical, hands-on lab exercises.

- Understand basic network architecture, and how protocols and functions fit together
- Learn the structure and operation of the Ethernet protocol
- Examine TCP/IP, including the addressing, protocol fields, and operations used for networks
- Explore the address resolution process in a typical IPv4 network
- Become familiar with switches, access points, routers, and other network components that process packets
- Discover how the Internet Control Message Protocol (ICMP) provides error messages during network operations
- Learn about the network mask (subnetting) and how it helps determine the network

Twitter: @oreillymedia
facebook.com/oreilly

US \$24.99

CAN \$28.99

ISBN: 978-1-449-30653-3



O'REILLY[®]
oreilly.com

Packet Guide to Core Network Protocols

Packet Guide to Core Network Protocols

Bruce Hartpence

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

www.allitebooks.com

Packet Guide to Core Network Protocols

by Bruce Hartpence

Copyright © 2011 Bruce Hartpence. 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 books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://my.safaribooksonline.com>). For more information, contact our corporate/institutional sales department: (800) 998-9938 or corporate@oreilly.com.

Editor: Mike Hendrickson

Production Editor: Jasmine Perez

Copyeditor: Amy Thomson

Proofreader: Jasmine Perez

Cover Designer: Karen Montgomery

Interior Designer: David Futato

Illustrator: Robert Romano

Printing History:

June 2011: First Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *Packet Guide to Core Network Protocols*, the image of a helmetshrike, 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.

ISBN: 978-1-449-30653-3

[LSI]

1306936758

*To my wonderful wife, Christina, and our three
great kids, Brooke, Nick, and Sydney. A husband
couldn't be happier or a dad more proud.*

Table of Contents

Preface	xi
 1. Networking Models	 1
What Is a Model?	1
Why Use a Model?	3
OSI Model	4
OSI—Beyond the Layers	7
OSI/ITU-T Protocols	8
Introducing TCP/IP	8
TCP/IP and the RFCs	11
The Practical Side of TCP/IP	13
Encapsulation	14
Addressing	15
Equipment	15
Reading	17
Summary	17
Review Questions	18
Review Answers	18
Lab Exercises	19
Activity 1—Examining Encapsulation	19
Activity 2—Protocol Distribution	19
Activity 3—Developing a Protocol/Architecture	19
 2. Ethernet	 21
Remember the Models	22
Structure	23
Preamble	23
Source and Destination MAC Addresses	24
Control Field (Type)	24
Data Field	24
Frame Check Sequence	25

Ethernet Type II vs. 802.3	25
MAC Addresses—Another Look	27
Ethernet Operation	29
Shared Media	30
Physical Layer	32
Cabling	33
Encoding	37
10Base-T	37
100Base-T	38
1000Base-T	39
Other Types of Signaling	39
Link Pulse	39
Autonegotiation	40
Topologies	40
Final Thoughts on Ethernet	41
Reading	41
Summary	42
Review Questions	42
Review Answers	42
Lab Exercises	43
Activity 1—Basic Framing	43
Activity 2—Control Field Values	43
Activity 3—Addressing	44
Activity 4—Destination Addresses	44
Activity 5—Logical Link Control	44
 3. Internet Protocol	 45
Protocol Description	45
Structure	46
Addressing	53
Sample Host Configuration	55
Operation	56
Digging a Little Deeper...What Addressing is Sufficient?	57
Security Warning	58
Organizations for Assigning Addresses and Names	58
Standards and RFCs	59
Summary	60
Review Questions	60
Review Answers	60
Lab Exercises	61
Activity 1—Determining IP Address Components	61
Activity 2—IP Packet Capture	61
Activity 3—Header Checksum	61

Activity 4—Fragmentation	62
Activity 5—Special Address Capture	62
4. Address Resolution Protocol	63
The Problem	63
Techniques	64
Protocol Description	64
Structure	65
Addressing in the ARP Request	66
Addressing in the ARP Reply	67
Operation	68
Example 1—Sender and Target on the Same LAN	68
Example 2—Sender and Target on Separate LANs	69
Additional Operations	70
The Return ARP	71
Gratuitous ARP	71
Security Warning	72
IPv6	72
Digging a Little Deeper	72
Standards and RFCs	73
Summary	74
Review Questions	74
Review Answers	74
Lab Activities	75
Activity 1—Determining Your IP Address and Your Default Gateway	75
Activity 2—Examining the ARP Table	75
Activity 3—Packet Capture	75
Activity 4—Gratuitous ARP	76
Activity 5—How Long Does an ARP Table Entry Live?	76
5. Network Equipment	77
Tables and Hosts	77
Hubs or Repeaters	79
Switches and Bridges	80
Access Points	85
Routers	88
Another Gateway	91
Multilayer Switches and Home Gateways	91
Security	93
Summary	93
Review Questions	94
Review Answers	95
Lab Activities	95

Activity 1—Traffic Comparison	95
Activity 2—Layer-2 Trace	95
Activity 3—Tables	96
Activity 4—Layer-3 Trace	96
Activity 5—Traffic Comparison	96
6. Internet Control Message Protocol	97
Structure	98
Operations and Types	100
Echo Request (Type 0) and Echo Reply (Type 8)	100
Redirect (Type 5)	104
Time to Live Exceeded (Type 11)	106
Tracing a Route	107
Destination Unreachable (Type 3)	108
Router Solicitation (Type 10) and Router Advertisements (Type 9)	110
Digging a Little Deeper—the One’s Complement	111
IPv6	112
Summary	114
Additional Reading	114
Review Questions	114
Review Answers	115
Lab Activities	115
Activity 1—Ping	115
Activity 2—Tracert	115
Activity 3—Start Up Packet Capture	116
Activity 4—Destination Unreachable From the OS	116
Activity 5—Destination Unreachable From the Router	116
7. Subnetting and Other Masking Acrobatics	117
How Do We Use the Mask?	118
What Is a Subnet?	122
Subnet Patterns	123
Subnet IP Addressing	124
A Shorthand Technique	125
The Effect on Address Space	126
Theory vs. Reality	126
Supernetting	127
The Supernetted Network	129
Classless Inter-Domain Routing	130
CIDR and Aggregation Implementation	133
RFC 4632	134
Summary	134
RFCs and Reading	135

Review Questions	135
Review Answers	136
Lab Activities	136
Activity 1—What Is My Network?	136
Activity 2—Change Your Network	136
Activity 3—What Is the Address Given to You by Your ISP?	137
Activity 4—Subnet Calculator	137

Preface

Trying to find the perfect networking resource or textbook can be a real challenge. Sometimes they are extremely focused on one technology, and thus miss the mark. Or they are extremely broad, covering every networking idea known to man. This book is about something that all networks have in common—the core protocols. Networks have a couple of basic building blocks: routers, switches, access points, and hosts. These building blocks use a particular set of rules when forwarding bits of information from one side of the network to another.

These bits are wrapped up in a neat little package called a packet. Packets have many qualities, but one thing they never do is lie. If a packet is present, it is there because some device or network host put it there. By looking at the packets running on a network and understanding the forces (sometimes good, sometimes evil) that put them there, we can gain a deep understanding of how networks operate and what is happening at a given moment.

This book provides the structure (a.k.a. model) used to formulate network transmissions and then dives into the major protocols populating almost every single network today: Ethernet, Internet Protocol (IP), Address Resolution Protocol (ARP), and the Internet Control Message Protocol (ICMP). But this is not simply a description of the foundation protocols. In each chapter, the protocol is analyzed by examining topologies and the packets generated on actual networks. Wireshark is the tool of choice. It is not only powerful but the folks out at wireshark.org continue to provide it free of charge.

Almost all network devices and hosts use tables to make decisions. The packets are on the network because a table was consulted and the result indicated that a transmission be sent. So the packets are the end result. Inside these pages you will find discussion and examples of the ARP tables, routing tables, and source address tables. Tying it all together will be step-by-step descriptions of the processes used so that the reader will be able to completely trace and understand the content of the packets and the events within the communications architecture.

Other key components of this book include addressing and equipment operation. Since lists of addresses are not much use to someone wishing to understand actual behavior,

each chapter describes variation and application of these addresses. A chapter on masks has also been included because it is such an integrated part of every single network. Just for fun, there is a section on cabling to provide an explanation of why we connect things the way we do.

The sources used in this book are the actual standards as described by the IEEE and ITU-T. Wherever possible, RFCs are directly referenced. So, if you see it here, it came from either the original source or an operational network.

In a nutshell, this book will describe the core protocols, tables, and equipment used on contemporary networks. Each chapter will take topologies and packets from actual networks and explain why the packets were generated and the purpose of the content found in each. The goal is to provide an in-depth understanding of these components, security concerns, and operation.

Audience

For those not familiar with O'Reilly books, they commonly do two things: provide lots of solid information and help with the real world. I've tried to do the same thing here. So this book is terrific for anyone trying to understand networks for the first time and anyone who works with them on a regular basis.

If you have never run packet capture or analysis software, the first time is always an eye opener. All those packets whizzing around the network and each one chock full of arcane information. With this book as a guide you will be able to interpret what is seen and understand why it is there.

For the professional out there, well, we forget things and sometimes get lost in the weeds. When that happens or if you need a refresher, this book is a great reference, not only for the chapter content but for the decomposition of the standards as well. The expert in the field will also find many details not explained elsewhere.

Contents of This Book

Chapter 1, Networking Models

Many networking texts start with models, but this is models with a twist. This book focuses on the TCP/IP model, and this chapter gives them a place in the universe and describes where the focus should be. Backed up with captures and standards, the models are populated with protocols, equipment, and addresses.

Chapter 2, Ethernet

Ethernet provides the basis for a very large percentage of the networks deployed today. This chapter discusses the choices of the network administrator while providing significant details about operation and configuration. Topologies and cabling are two other focal points providing further details into actual networking practice.

Chapter 3, *Internet Protocol*

Leaving Layer 2, we arrive at Layer 3 and the domain of IP. This chapter takes us through the structure and operation of IP, backed up by packet captures of course. Every field is given an example. Particular attention is paid to the addressing and how it is deployed, including the entries standard to the host routing table.

Chapter 4, *Address Resolution Protocol*

ARP is arguably the simplest protocol on a network, but it is also a very neat troubleshooting tool and a point of attack for the bad guys. This chapter discusses the operation and particular addressing associated with ARP. It also covers storage of learned information and network behavior.

Chapter 5, *Network Equipment*

It is easy to outline network device responsibilities and assign them to the various layers of the networking model. This chapter goes a step further to include tables used in making forwarding decisions and guides the reader through a series of step-by-step examples.

Chapter 6, *Internet Control Message Protocol*

This protocol defines a large collection of error and informational message types. However, contemporary networks utilize a subset of this collection. For every one of these practical messages types, a topology is provided and sample packet captures analyzed to give the reader a complete understanding of the situations resulting in their transmission.

Chapter 7, *Subnetting and Other Masking Acrobatics*

A network cannot be built without using network masks. This chapter describes the subnetting and supernetting procedures and provides two methods for arriving at the correct answer when dividing up address space. Several examples are provided and explained completely.

Conventions Used in This Book

The following typographical conventions are used in this book:

Plain text

Indicates menu titles, menu options, menu buttons, and keyboard accelerators (such as Alt and Ctrl).

Italic

Indicates new terms, URLs, email addresses, filenames, file extensions, pathnames, directories, and Unix utilities.

Constant width

Indicates commands, options, switches, variables, attributes, keys, functions, types, classes, namespaces, methods, modules, properties, parameters, values, objects, events, event handlers, XML tags, HTML tags, macros, the contents of files, or the output from commands.

Constant width bold

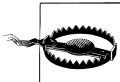
Shows commands or other text that should be typed literally by the user.

Constant width italic

Shows text that should be replaced with user-supplied values.



This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Packet Guide to Core Network Protocols* by Bruce Hartpence (O'Reilly). Copyright 2011 Bruce Hartpence, 978-1-449-30653-3."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

Safari® Books Online



Safari Books Online is an on-demand digital library that lets you easily search over 7,500 technology and creative reference books and videos to find the answers you need quickly.

With a subscription, you can read any page and watch any video from our library online. Read books on your cell phone and mobile devices. Access new titles before they are available for print, and get exclusive access to manuscripts in development and post feedback for the authors. Copy and paste code samples, organize your favorites, download chapters, bookmark key sections, create notes, print out pages, and benefit from tons of other time-saving features.

O'Reilly Media has uploaded this book to the Safari Books Online service. To have full digital access to this book and others on similar topics from O'Reilly and other publishers, sign up for free at <http://my.safaribooksonline.com>.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

<http://www.oreilly.com/catalog/0636920020516>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

Acknowledgments

When networking, I live for packets. While I would have used any tool available, the folks out at Wireshark sure made it easy with a terrific tool and a nice set of resources.

Thanks to all of my students, who after realizing the coolness and importance of understanding packets, reaffirmed my own belief in those magical little packages. Thanks also to my colleagues for the kind words of encouragement. Especially those of you that helped me get started in the early days. Ten Hungry Writers and Bill Stallings—you know who you are. Special thanks to Jim Leone, who not only followed my style changes, but kept up with the editing when I needed it—and all for the price of a burrito. By the way Jim—e4.

Networking Models

Mod-el: noun: 1—structural design, 2—a miniature representation, 3—an example for emulation or imitation

—The Merriam-Webster Dictionary

Basic network architecture and construction is a good starting point when trying to understand how communication systems function, even though the topic is a bit dull. Architectures are typically based on a model showing how protocols and functions fit together. Historically, there have been many models used for this purpose, including, but not limited to, Systems Network Architecture (SNA-IBM), AppleTalk, Novell Netware (IPX/SPX), and the Open System Interconnection (OSI). Most of these have gone the way of the dodo due to the popularity of TCP/IP. TCP/IP stands for Transmission Control Protocol/Internet Protocol, and represents a suite of protocols used on almost all modern communication systems. As the name suggests, this is the language of the Internet. This chapter focuses on the practical TCP/IP model, using the OSI model as a reference point.

What Is a Model?

A model is a way to organize a system's functions and features to define its structural design. A design can help us understand how a communication system accomplishes tasks to form a protocol suite. To help wrap our heads around models, communication systems are often compared to the postal system ([Figure 1-1](#)). Imagine writing a letter and taking it to the post office. At some point, the mail is sorted and then delivered via some transport system to another post office. From there, it is sorted and given to a mail carrier for delivery to the destination. The letter is handled at several points along the way. Each part of the system is trying to accomplish the same thing—deliver the mail. But each section has a particular set of rules to obey. While in transit, the truck follows the rules of the road as the letter is delivered to the next point for processing. In between, inspectors and sorters ensure the mail is metered and safe, without much concern for traffic lights or turn signals.

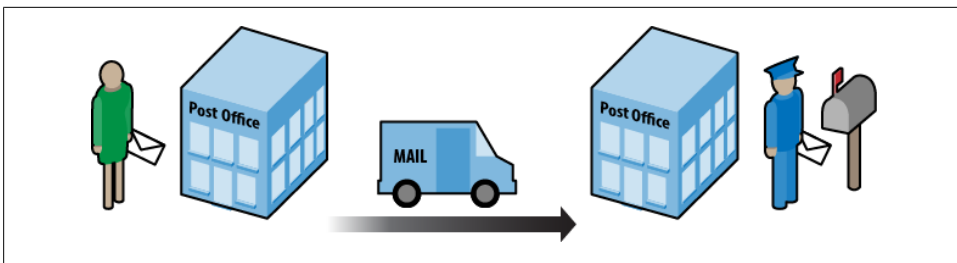


Figure 1-1. Postal system

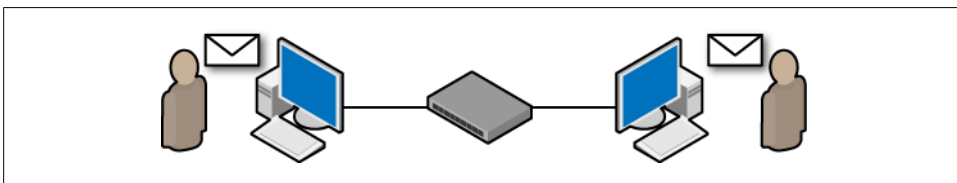


Figure 1-2. Small communication network

A communication system is not much different, since messages created on a computer are processed and delivered, with each piece of equipment performing some function and obeying certain rules for transmission. [Figure 1-2](#) depicts a typical scenario in which two computers are connected by their network cards via a networking device. Two people are communicating using an application such as instant message or email. At some point we have to decide exactly how to handle this communication. After all, when you mail that letter, you cannot address the envelope in some arbitrary language or ignore zip codes, just as the mail truck driver cannot drive on the wrong side of the road.

So, how is the job of each device or connection determined? An application at the user level should not be responsible for choosing the encoding sequence or the signal type used between the client and server. The letter doesn't decide to go by air or boat. Similarly, the network interface card (NIC) is not in the business of message header construction, just as the mail sorting system doesn't care if you use pen or pencil when writing a letter.

Models are routinely organized in a hierarchical or layered structure. Each layer has a set of functions to perform. Protocols are created to handle these functions and, therefore, protocols are also associated with each layer. The protocols are collectively referred to as a *protocol suite*. The lower layers are often linked with hardware and the upper with software. For example, Ethernet operates at Layers 1 and 2, while the File Transfer Protocol (FTP) operates at the very top of the model. This is true for both the TCP/IP and OSI models. Network traffic can also be viewed in terms of these layers, many of which can actually be seen using a packet capture tool like Wireshark. In [Figure 1-3](#), the major layers of the TCP/IP model are displayed in a message going to a web server.

```
Ethernet II, Src: Western0_89:ba:fa (00:00:c0:89:ba:fa), Dst: Cisco_2c:0c:80 (00:11:21:2c:0c:80)
Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.254 (192.168.1.254)
Transmission Control Protocol, Src Port: optima-vnet (1051), Dst Port: http (80), Seq: 1, Ack: 1, Len: 336
Hypertext Transfer Protocol
```

Figure 1-3. Packet showing layers

Why Use a Model?

Before we get too far, let's do a little reality check. A model describes the entire structure. At the beginning of the chapter, I stated that many of these models “have gone the way of the dodo.” There may have been good ideas in each, but everyone ended up using one model in particular—TCP/IP. For example, both Apple and IBM initially developed their own protocol suites, but converted to TCP/IP due to its popularity. This section explains the historical use of models and provides a more modern viewpoint.

Even a simple communication system is a complicated environment in which thousands or even millions of transactions occur daily. Interconnected systems are considerably more complex. A single electrical disturbance or software configuration error can prevent completion of these transactions. Models provide a starting point for determining what must be done to enable communication or to figure out how systems using different protocols might connect to each other. They also help in troubleshooting problems. For example, how would a Novell Netware client running IPX/SPX communicate with an IBM AS400 over a TCP/IP based network? [Figure 1-4](#) depicts a scenario in which several different platforms might be required to interact with each other. Windows nodes are based on the TCP/IP protocol suite, but, if required, can run Novell Netware client software for network authentication. Novell developed internetworking and transport protocols, called IPX and SPX. At the other end of the network, the IBM mainframe communicates via the protocols used in the SNA model. Imagine the programming and extra effort required to maintain all of the transactions between these separate architectures.

Another example is a network of Apple computers running Appletalk while connecting to a network of Windows machines running TCP/IP.

As I've said, TCP/IP is the prevalent architecture today. The complexities of interplatform communication are dramatically reduced with TCP/IP. Protocol systems such as Appletalk, Netware, and SNA are considered legacy. However, understanding protocol layers on a particular communications device or how processes might interact on the network are still critically important ideas. When troubleshooting standard problems or potential security threats, the models and their associated layers offer logical reference points to begin the process. One would not start looking at the routing protocols if the link light was dark.

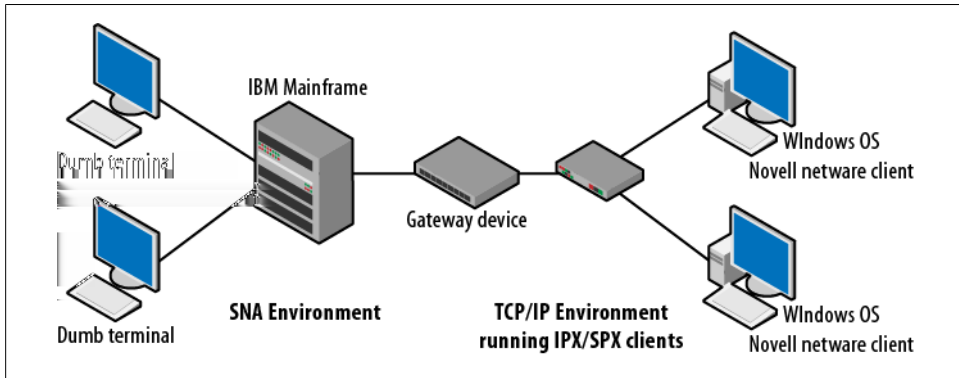


Figure 1-4. Mixed architecture topology

OSI Model

The OSI model is called a reference model. This means this particular model provides a method by which standards and protocols can be compared in order to assist in connectivity and consistency. Developers can use a reference model to understand how transmissions are framed and create methods to translate between systems.

The OSI basic model is standardized in ISO/IEC (International Standards Organization/International Electrotechnical Commission) 7498, which includes most of the definitions used here. These two organizations have actually created a Joint Technical Committee (JTC) that handles the issues associated with information technology. This model was developed in collaboration with the ITU-T and has also been printed as ITU-T Recommendation X.200. The ITU-T is the International Telecommunications Union—Telecom sector. Now that we’ve had our fill of acronyms, on to the standard.

The first version of ISO/IEC 7498 was written in 1984. This was replaced in 1994 by version 2, with some additional corrections after that date. ISO/IEC 7498 actually has four parts:

- Part 1 – The Basic Model
- Part 2 – Security Architecture
- Part 3 – Naming and Addressing
- Part 4 – Management Framework

This section examines the basic model, which is defined in section six of 7498-1 as having seven layers: *Application*, *Presentation*, *Session*, *Transport*, *Network*, *Data Link*, and *Physical*. [Figure 1-5](#) depicts these layers and the connection to a similarly structured *open* system. An open system is one that adheres to this architecture.

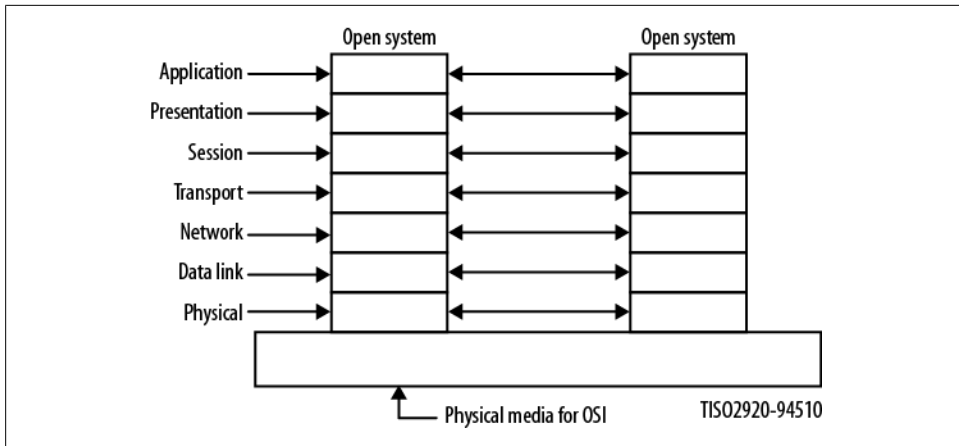


Figure 1-5. The OSI layers

Section 6 of the document also includes the guiding principles for layers, such as:

- Not creating so many as to make the engineering of the system difficult
- Reducing the number of interactions across a layer boundary
- Collecting similar functions and separating fundamentally different functions
- Identifying those that may receive a benefit by being based in hardware or software

Additionally, the OSI framework includes as its goals the improvement of current standards, flexibility, and the quality of being “open,” which simply means that systems have mutually adopted accepted standards for the exchange of information. Each of the layers defined in 7498-1 includes specific details about the functions and processes occurring at that layer. It is worth noting that this ISO/IEC/ITU-T document specifically states:

It is not the intent of this reference model to either serve as an implementation specification or to be a basis for appraising the conformance of actual implementations or to provide a sufficient level of detail to define precisely the services and protocols of the interconnection architecture.

So the OSI model does not specify protocols, services, or rules to be used in a communication system, but it does detail the ideas and processes that may be required. Section 7 of 7498-1 provides some particulars for each of the layers, and these are summarized below. If you are driving heavy machinery, be careful in the next section.

Application

Provides the sole means for the application to access the OSI environment (OSIE) with no layer above it. The functions are divided into *connection mode* and *connectionless mode*. Connection mode facilities will include quality of service (QoS), security, identification of the parties, error control, and mode of dialog.

Connectionless facilities are essentially a subset of those already mentioned, without error control and some security.

Presentation

This layer provides for the representation and preservation of the data provided by the Application Layer entities. Specifically, the Presentation Layer is focused on syntax that is acceptable to both ends and access to the layers above and below.

Session

Specifies both full-duplex and half-duplex modes of operation. This layer provides the means (setup and teardown) for communicating nodes to synchronize and manage the data between them. A mapping is provided between the Transport Layer and Session Layer (session service access point) addresses. Support is present for many-to-one session-to-transport addresses. The bulk of the responsibilities at this layer involve connection-oriented transmissions, but connectionless transmissions are also supported.

Transport

Protocols at this layer are end-to-end between communicating OSI nodes and deal primarily with low cost and reliable transfer of data. A single Transport Layer address may be associated with many session addresses and provides performance required by each session entity. Basic functions include transport connection establishment, release, data transfer, and QoS. While this layer is not responsible for routing, it does map to Network Layer addressing. All modes handle error control, but when running in connection-oriented mode, sequence control is required.

Network

Provides the means for managing network connections between open systems. This layer is not responsible for negotiating QoS settings, but rather focuses on routing between networks and subnetworks. Network Layer addresses uniquely identify transport entities. The Network Layer is also responsible for error control, sequencing, and mapping to the data link addresses.

Data Link

Responsible for the construction of the data link connection between Network Layer entities. The addresses used are unique within the open system set of devices. Like most of the OSI layers, connectionless and connection-oriented modes are utilized. In addition to interfacing with the Network Layer, the data link connection can be built upon one or more Physical Layer interfaces.

Physical

Like most models, this OSI Physical Layer contains the electrical, mechanical, and functional means to establish physical connections between Layer-2 devices. The interface is largely determined by the medium, but the bit-level transmissions must be organized into their physical service data units.

OSI—Beyond the Layers

It is common to limit the discussion of the OSI reference model to the seven layer specifications. While these ideas have been discussed here, the OSI model also provides a potentially valuable insight into the design and implementation of networking models and protocols. The architects of this model spent a lot of time thinking about and enumerating those items demanded at each layer and what is necessary to communicate with the layers immediately above and below. As an example, section five of 7498-1 includes discussion on the various aspects of layering. These include but are not limited to the following:

- Communication between peer entities, including the following:
 - Modes of communication (connection or connectionless)
 - Relationships between services provided at each adjacent layer boundary
 - Mode conversion functions (transport and network layers primarily)
- Identifiers such as N-address—unambiguous names used to identify a set of service access points at a particular layer
- Properties of service access points
- Definitions and descriptions of data units
- Elements of layer operation
 - Connections to/from
 - Multiplexing
 - Flow control
 - Segmentation
 - Sequencing
 - Acknowledgment
 - Protocol selection
 - Negotiation mechanisms
 - Connection establishment and release
 - Quality of service
 - Error detection

Along with these generalized aspects of communication within a layer model, each individual layer adds further discussion where appropriate. For example, the section dedicated to the Transport Layer details connection establishment/release, data transfer, functions within the layer, addressing, multiplexing/splitting, and management. Where necessary (where a one-to-one mapping between services/addresses is not always present), a layer description will include details about the negotiation of the connections between the layers or even sublayers.

OSI/ITU-T Protocols

So far we've examined a layered model and outlined the responsibilities of each layer. What about the actual protocols? For every protocol used in the TCP/IP model, there is a corresponding (and perhaps more complex) version in the OSI/ITU-T architecture. For ease of access to the reference material, this section refers to the ITU-T X series of standards.

As stated previously, the model itself is described in ITU-T X.200. While the layers are also described, more detailed specifications are contained within X.211-X.217bis. These additional documents are similar to RFCs for individual protocols in that they provide the rules and guidelines for those actually developing protocols, including state diagrams and primitive definitions. For both the Network and Transport Layers, special attention is paid to connectionless- and connection-based communication. One of the major differences between these two forms of transmission is controlling the flow of information between endpoints. It is interesting that the two Layer-4 protocols used today—TCP and UDP (User Datagram Protocol)—are differentiated from each other in the exact same way, with TCP characterized as connection-oriented while UDP is connectionless. TCP is very concerned with sequence numbers and ensuring that all packets arrive at the destination. UDP is not.

[Figure 1-6](#) is from ITU-T X.220 and shows the actual protocols to be used. The original diagram is quite large, so only a portion of it is shown here. While a bit old (written in 1993), it does provide some insight into the structure of the model. Many of the protocols are outdated, but we can see the modularity of the protocol stack that aids in subsystem replacement. For example, at the lower layers, X.25 has been replaced by Frame Relay and ATM. These, in turn, have been replaced by the transmission standards we use today.

As a practical matter, OSI/ITU-T protocols are not seen nearly as often as the protocols specified for use in the TCP/IP model, although there are exceptions. Some WAN connections continue to use these specifications and, of course, we still have traditional telephony systems. Perhaps one of the best examples of an ITU-T standard that continues to survive even as more and more communications shift to TCP/IP is in the area of VoIP (Voice over IP). H.323, Q.931, and G.711 are still a big part of contemporary VoIP transmissions, as shown in [Figure 1-7](#). H.225 is part of H.323.

Introducing TCP/IP

The Internet and almost all networks in use today have standardized on the TCP/IP model. It is often referred to as the language of the Internet, because applications are typically built around this protocol suite. [Figure 1-8](#) shows the TCP/IP model and some of the more well-known protocols and corresponding layers. At Layer 4, there are actually two protocols present—TCP and UDP. While this model shares its name with the former, many operations are based on UDP, so Layer 4 is actually shared by the

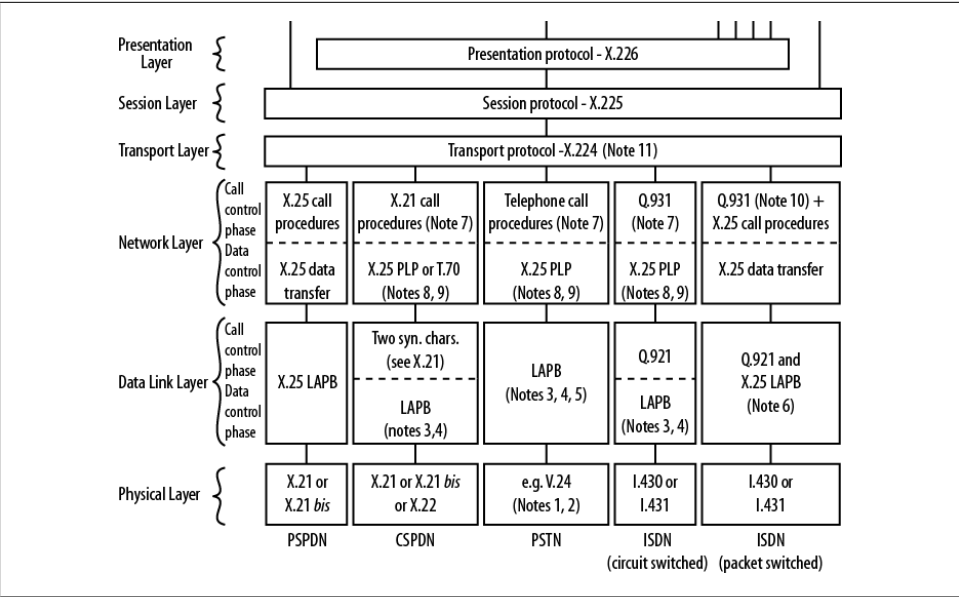


Figure 1-6. OSI/ITU-T protocols

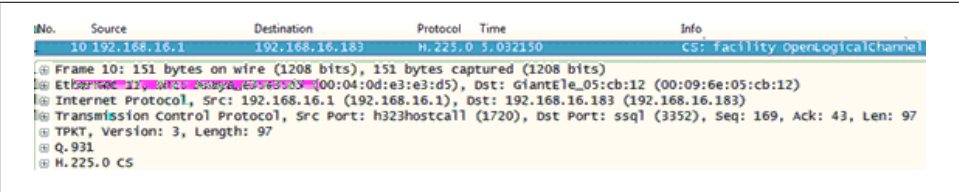


Figure 1-7. ITU-T protocols used in VoIP

Application	FTP, telnet, email, games, printing, http
Transport	Transmission control Protocol (TCP), User Datagram Protocol (UDP)
Internet (Internetwork)	Internet Protocol (IP), ICMP, IGMP
Link layer (Network)	Ethernet, 802.11
Physical	Ethernet, 802.11

Figure 1-8. The TCP/IP model and protocols

two protocols. Layers 1 and 2 are governed by the Local Area Network Protocol, but Layer 3 belongs to IP with Internet Control Message Protocol (ICMP) and Internet Group Membership Protocol (IGMP) components of IP-based operations.

The TCP/IP model does not specify any particular protocol to be run at the lower (LAN) layers. Historically, networks have been built upon many technologies, including fiber distributed data interface (FDDI), Localtalk, Token Ring, Ethernet, and wireless protocols from the 802.11 family of standards. Today, only Ethernet and 802.11 in their various forms survive and even these have eliminated certain variations. For example, Ethernet based on coaxial cable and 802.11 frequency-hopping are almost nonexistent.

In a typical network, most of the decisions regarding protocols, at least for Layers 1–4, are made for you and the real variation is in the applications you chose to deploy. This argument can even be made for advanced technologies such as voice communications, where traditional circuit-switched telephone systems are quickly being replaced by VoIP. The dominance of the TCP/IP model can be demonstrated using a tool within Wireshark. By examining the protocol distribution for a particular network segment, a picture (shown in Figure 1-9) emerges regarding the protocols in use.

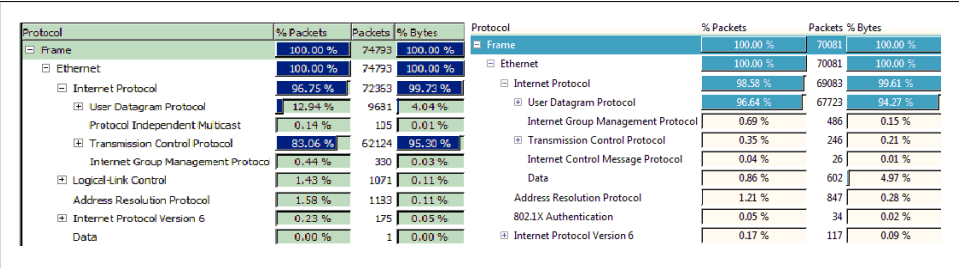


Figure 1-9. Windows and Apple protocol distribution

On the left is packet capture data from a Windows machine and on the right from a Mac OS platform. Nearly 100 percent of the Layer-3 traffic caught is IPv4, with a small amount of IPv6. At Layer 4, TCP and UDP dominate, with ARP and 802.1x contributing. *Missing from the data collected are protocols from any other model.*

Comparing the TCP/IP and OSI models, it can be said that the functions are the same but the structure is different. Figure 1-10 shows a side-by-side comparison of the OSI and TCP/IP model layers. Layers 5–7 of the OSI model map to Layer 5 of the TCP/IP model.

Application	7	Application
	6	Presentation
	5	Session
Transport	4	Transport
Internetwork	3	Network
Link/Network	2	Data Link
Physical	1	Physical

Figure 1-10. The TCP/IP and OSI networking models

TCP/IP and the RFCs

The TCP/IP model is presented in RFCs 1122 and 1123. These documents, released in 1989, provide the same level of detail encompassed in the ISO/IEC standard. An examination of the standard dates, historical deployments, and the difference in complexity between the two models provides insight into the adoption of TCP/IP over the OSI model. If you survived reading the OSI sections above, you might be left with the impression that the OSI model is very complex in comparison to TCP/IP. A quick look at the “companion” RFCs that go along with RFCs 1122 and 1123 also show a significant level of complexity. To quote from RFC 1122:

This RFC enumerates standard protocols that a host connected to the Internet must use, and it incorporates by reference the RFCs and other documents describing the current specifications for these protocols. It corrects errors in the referenced documents and adds additional discussion and guidance for an implementor.

For each protocol, this document also contains an explicit set of requirements, recommendations, and options. The reader must understand that the list of requirements in this document is incomplete by itself; the complete set of requirements for an Internet host is primarily defined in the standard protocol specification documents, with the corrections, amendments, and supplements contained in this RFC.

As with the OSI model, each layer of the TCP/IP model has a particular set of responsibilities. While most of these are defined in RFC 1122, those for the Application Layer come from RFC 1123. One significant difference between the two models is that RFC 1122 does specify particular protocols at the various layers. What follows are some of the major requirements that *must* occur at each layer.

Application

The top TCP/IP layer combines the OSI Application and Presentation Layers and includes user-based and support/management protocols. Items at this layer must do the following:

- Support flexibility (naming and length) in hostnames
- Map domain names appropriately
- Handle DNS errors

Telnet, FTP, Trivial FTP, Simple Mail Transport Protocol, and Domain Name Service all have more specific additional requirements.

Transport

This layer provides end-to-end communication services based either on TCP (connection-oriented) or UDP (connectionless). TCP is much more concerned with sequence numbers and handshaking than UDP. Items at this layer must do the following:

- Pass IP options and Internet Control Message Protocol messages to the Application Layer
- Be able to handle and manipulate checksums

- Support IP addresses, local and wildcards, such as broadcast, multicast, and unicast destinations
- Treat window size as an unsigned number
- Manage window size effectively and allow 0 window size
- Support urgent data and the pointer that points to last octet of the urgent data
- Support TCP options
- Gracefully handle opening of connections
- Silently discard improper connection requests
- Handle retransmissions per recommended algorithms
- Follow recommended procedures when generating ACKs (acknowledgments)
- Gracefully handle connection failures

Internet

The Internet Layer specifies the use of IP, ICMP, and Internet Group Management Protocol. Operationally, this is a connectionless “best effort” protocol concerned with addressing, type of service, security, and fragmentation. It relies on upper-layer protocols for accurate delivery. Items in this layer must do the following:

- Handle remote multihoming
- Meet appropriate gateway specification, if used
- Discard improper IP and ICMP packets
- Properly handle all forms of addressing including subnets
- Maintain packet IDs
- Support ToS and reassembly
- Support source routing options

Link Layer

This is the network interface, and includes framing and media access to communicate directly with the network to which it is attached. Items at this layer must:

- Clear the ARP cache
- Prevent ARP floods
- Send and receive RFC 894 encapsulation (should also support IEEE 802)
- Use ARP on Ethernet and IEEE 802 networks
- Report link layer broadcasts to Internetwork Layer.
- Receive IP ToS values

Physical

Typically, the network interface card or port defines the Physical Layer. Each LAN protocol has within its specification the electrical and mechanical characteristics for communication on the link. These include items such as voltage level, encoding, pin assignments, and shape of the interface.

Not all of the requirements were actually implemented in protocol suites running on hosts and network equipment. For example, IP hosts were intended to be much more active in detecting gateway or next hop failures. The reality is that if a gateway fails, hosts simply can't communicate to destinations outside their networks.

The Practical Side of TCP/IP

The models discussed in this chapter are usually drawn from the top layer (application) down. Wireshark displays them in reverse order, as shown in [Figure 1-11](#), where the protocols corresponding to Layers 1–5 are identified. This packet happens to be from a VoIP conversation. Starting from the bottom of the model, we see Ethernet Type II (also called Ethernet Type 2 or Ethernet II). Ethernet as a protocol exists at Layers 1 and 2, with Layer 2 defining the frame (error control, addressing, etc.) and the Ethernet network interface defining the Physical Layer characteristics.

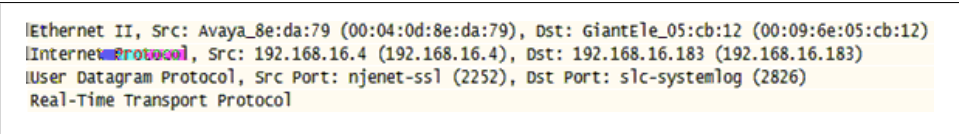


Figure 1-11. RTP example showing TCP/IP protocols

In addition, Layer 2 LAN protocols can be split into two major areas: logical link control (LLC) and media access control (MAC). These become *sublayers* within the model. LLC functions include frame construction, error control, and addressing. The MAC layer defines line discipline and network transmission. Specifically, this includes a method for determining which node is in line to communicate and for how long. [Figure 1-12](#) depicts these TCP/IP sublayers. Sublayers are also found in the OSI model.

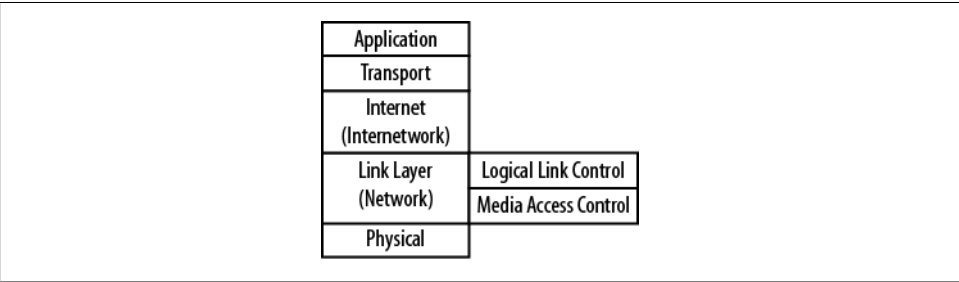


Figure 1-12. Layer-2 sublayers

Encapsulation

Encapsulation is the method by which the various layers interact and pass information up and down the protocol stack. A message generated by an application must be formatted for transmission. On the sending node, the upper layer places packaging around the message that describes the application used to generate the message. This is called the *header*. The package and header are then passed to the next layer down. Each layer completes its own required encapsulation operation that includes a header.

By the time the message reaches the bottom of the protocol stack, it has several of these wrappers. This process is the encapsulation. For ease of processing, each header contains information regarding the contents. Thus, the Ethernet header provides some indication that it has encapsulated IP. The IP header indicates that it was carrying TCP, and so on. Basic encapsulation and the encapsulation specific to the packet in [Figure 1-11](#) are shown in [Figure 1-13](#).

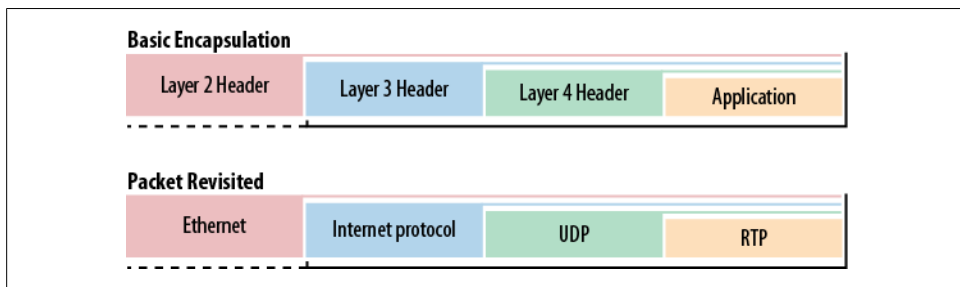


Figure 1-13. Encapsulation

At the receiving node, the same process occurs, but in reverse. Each layer, beginning with the lowest, processes the appropriate information and strips off the outermost wrapper before handing the message to the next layer up. This continues as the message travels up the stack until the last bit of packaging is removed. This process is called *de-encapsulation*.

The packet shown in [Figure 1-14](#) has been expanded to reveal the headers. Each layer contains a code identifying the content of the encapsulated data. Ethernet uses the code 0800 to indicate that IP is encapsulated. At Layer 3, IP uses code 17 to show that it has encapsulated UDP. At Layer 4, UDP uses port numbers to direct the data to the proper process or application. The receiver uses this information to properly parse and de-encapsulate the data.

```

Ethernet II, Src: Ibm_43:49:97 (00:11:25:43:49:97), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  Destination: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  Source: Ibm_43:49:97 (00:11:25:43:49:97)
  Type: IP (0x0800)
Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 239.255.255.250 (239.255.255.250)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  Total Length: 125
  Identification: 0xb23a (45626)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 1
  Protocol: UDP (17)
  Header checksum: 0x5592 [correct]
  Source: 192.168.1.1 (192.168.1.1)
  Destination: 239.255.255.250 (239.255.255.250)
User Datagram Protocol, Src Port: blockade-bpsp (2574), Dst Port: ssdp (1900)
  Source port: blockade-bpsp (2574)
  Destination port: ssdp (1900)
  Length: 105
  Checksum: 0x0118 [validation disabled]
Hypertext Transfer Protocol

```

Figure 1-14. HTTP packet

Addressing

Addressing used within networked systems can also be tied to layers and equipment, as shown in [Figure 1-14](#). Some protocols require addressing as part of their basic operation. For example, an Ethernet switch processes Layer-2 frames, which contain addresses called hardware or MAC addresses. For example, the source MAC address in [Figure 1-14](#) is 00:11:25:43:49:97. The Internet Protocol uses IP addresses at Layer 3, and these are processed by routers. In this case, the source is 192.168.1.1 and the destination is 239.255.255.250. Both TCP and UDP communicate via port numbers at Layer 4. Thus, a TCP/UDP message not only possesses the port numbers necessary to communicate with an application, but also the IP and MAC addresses necessary to complete the transmission. Understanding relationships like these better prepares the network administrator to build, troubleshoot, or secure the infrastructure. For example, if the administrator is concerned about port scanning at Layer 4, it is unlikely that solutions will be found by working with the network switches down at Layer 2.

Equipment

Layering the model can also provide a picture of device responsibilities. Each device on a network is designed for a particular task. They have different levels of intelligence and process traffic in a variety of ways. By applying the layers to equipment, the impact on traffic and capabilities of the device at that particular location are easier to understand. Routers and switches form the building blocks of almost any network. While they have many individual features and can be configured for a variety of functions, they all provide the same basic services when you plug them, regardless of the vendor. The relationship between devices, addressing, and the layers is outlined in [Figure 1-15](#).

Layer	Device	Addressing
Application		
Transport	Gateway	TCP/UDP Ports
Internetwork	Router	IP addresses
Link/Network	Switch	MAC addresses
Physical	Hub	Bits

Figure 1-15. Equipment and addressing layers

Switches operate at Layer 2 and forward LAN frames based upon the MAC addresses contained within those frames. They also perform error checking for each frame. Switches also provide some measure of network segmentation, since the processing of MAC addresses will result in network traffic control. Switches have a variety of management features such as support for Simple Network Management Protocol (SNMP) and virtual local area networks (VLANs).

Routers operate at Layer 3 and process IP packets. They will read Layer-2 frames when their MAC addresses appear in the frames, but their main function is to get IP packets to the proper destination. In so doing, the router calculates the IP header checksum and can act on any QoS or fragmentation information the packet contains. Many routers also support advanced features such as firewalling, virtual private networks (VPNs) termination, authentication, and network address translation (NAT).

The term *gateway* has several meanings. Routers and network hosts are configured with a *default gateway*, but this is actually a router. It is called a default gateway because this is the network path to the rest of the world. The more traditional gateway existing at Layer 4 is a device used to convert between systems that do not share the same networking model. This type of environment is depicted in [Figure 1-4](#), and requires protocol translation for network nodes to communicate. We might say this sort of thing is another legacy item, but with the emergence of VoIP, the gateway is making a comeback. The language of the public switched telephone network (PSTN) is Signaling System 7 (SS7). A gateway that understands both TCP/IP and SS7 is required if an IP-based VoIP phone is to communicate with a traditional telephone.

Not all devices fit neatly into boxes. An *access point* is sometimes referred to as a *wireless hub* because it broadcasts certain kinds of traffic everywhere. However, like an Ethernet switch, the access point is not only aware of MAC addresses, it uses them to make some forwarding decisions. More recently, the emergence of multilayer switching has blurred the line between processing frames at Layer 2 and some of the higher level functions like routing. [Figure 1-16](#) provides an example of how these devices and addresses interact within the confines of the protocol layers.

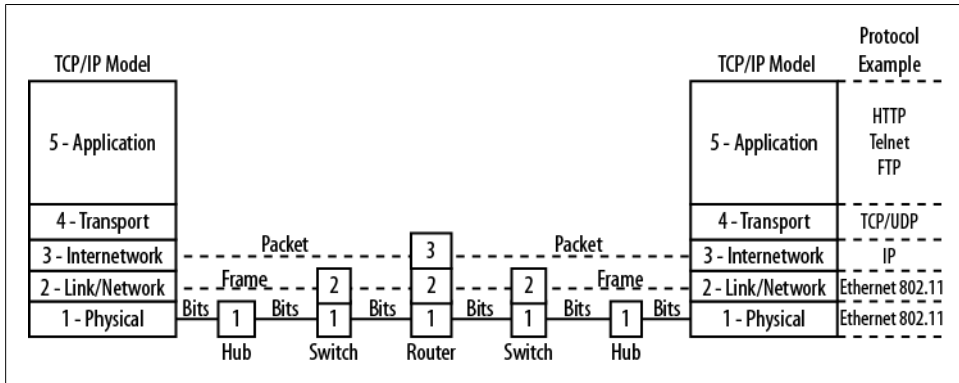


Figure 1-16. Equipment and addressing relationship

Reading

ISO/IEC 7498-1: “Information Technology — Open Systems Interconnection — Basic Reference Model: The Basic Model” (Joint Technical Committee)

ITU-T X.200: “Information Technology — Open Systems Interconnection — Basic Reference Model: The Basic Model” (International Telecommunication Union)

ITU-T X.220: Open Systems Interconnection — General Connection Mode Protocol Specifications “Use of X200-Series Protocols in CCITT Applications —” (International Telecommunication Union)

RFC 1122: “Requirements for Internet Hosts — Communication Layers” (Internet Engineering Task Force)

RFC 1123: “Requirements for Internet Hosts — Application and Support” (Internet Engineering Task Force)

Summary

The OSI and TCP/IP models provide an architecture in which the functions and rules for communication, addressing, and equipment are organized. The OSI model is standardized in ITU-T X.200, and the companion documents X.211-X.217 and X.220 provide the specifications for the developers and protocols. The TCP/IP model is described in RFC 1122, which lists the required protocols for hosts that communicate on the network.

The OSI model, now primarily considered a reference model, is in limited use. The TCP/IP model is the language of the Internet. Adoption of IP-based communication in many systems such as VoIP will further marginalize the OSI model’s use, a factor not lost on major companies like Apple and Novell, who have adopted the TCP/IP model.

Modern practitioners and researchers will be well served by an understanding of standards for communication requirements in widely varying conditions. These architectures help us to better understand the evolution of communication standards.

Review Questions

1. What is the name of the process by which an upper-layer protocol is wrapped up in a lower-layer protocol?
2. Name four communication models.
3. What two documents specify the standardization of the OSI model?
4. How many layers are in the OSI and TCP/IP models, respectively?
5. Name the layers of the OSI model.
6. Name the layers of the TCP/IP model.
7. What two documents present and detail the use of the TCP/IP model?
8. There are two forms of communication described in both standards. These forms are predominantly part of Layers 3 and 4. What are they?
9. Network administrators typically have the ability to use whatever protocols they wish, regardless of the layer in question. True or False?
10. One big difference between the documentation of the TCP/IP and OSI standards is that the TCP/IP RFC specifies the protocols to be used and the OSI ITU-T model documentation does not. True or False?
11. For each address type or device, specify its proper layer (Layer 2, Layer 3, or Layer 4).
 - Switch
 - Router
 - Gateway
 - MAC address
 - IP address
 - Port number

Review Answers

1. Encapsulation
2. TCP/IP, OSI, SNA, Appletalk, Novell (IPX/SPX)
3. ISO/IEC 7498 and ITU-T X.200
4. 7, 5
5. Application, Presentation, Session, Transport, Network, Data Link, Physical

6. Application, Transport, Internet, Data Link, Physical
7. RFCs 1122 and 1123
8. Connection-mode (oriented) and connectionless-mode
9. False
10. True
11. Layer 2, Layer 3, Layer 4, Layer 2, Layer 3, Layer 4

Lab Exercises

Activity 1—Examining Encapsulation

Materials: Wireshark and a network connection

1. Start a capture.
2. Complete several different transactions from your computer.
3. Stop the capture and examine the individual packets.
4. Find examples of the following: ARP, ICMP, TCP, UDP, and IPv6.
5. Describe these packets in terms of their encapsulation or protocol stacks.

Activity 2—Protocol Distribution

Materials: Wireshark and a network connection

1. Start a capture and allow it to run for several minutes, the longer the better.
2. Complete as many different transactions from your computer as possible.
3. From the Statistics menu in Wireshark, select Protocol Hierarchy.
4. Examine the distribution of protocols and attempt to determine the models used and the level of traffic specific to each protocol. What is the most common upper-layer protocol? What caused it to be generated?

Activity 3—Developing a Protocol/Architecture

Using the models discussed in this chapter as references, develop a series of rules or parameters that describe a conversation between two people who have never met. Things to consider might include the mode of communication, language, nonverbal communication, access method, body language, expressions, speed, and parameter negotiation.

CHAPTER 2

Ethernet

Computers cabled together in a network are almost certainly going to be connected via Ethernet. Ethernet is a technology that describes the rules used for communication between LAN-based systems and is considered a Layer-2 protocol. This chapter discusses the structure and operation of the Ethernet protocol, the differences between Ethernet Type II and 802.3, cabling types, and deployment considerations.

A historical review of the current standards can be a little confusing. The story begins in the 1970s with Bob Metcalfe, who envisioned a cable-based network, which later evolved into Ethernet Type II. Shortly after Metcalfe's ideas were disseminated, the IEEE standards committee developed 802.3 Ethernet. Both versions are in use today and are described more fully below. For the information-hungry, some interesting documents to read include:

- “Ethernet: Distributed Packet Switching for Local Computer Networks” (Metcalfe and Boggs)
- “The Ethernet: A Local Area Network Physical Layer and Data Link Layer Protocol Specifications” (DEC, Intel, and Xerox)
- “802.3-1985 IEEE Standard for Local and Metropolitan Area Networks: Carrier Sense Multiple Access with Collision Detection (Original 10Mb/s Standard)” (IEEE Standards Association)

The first paper, written by Bob Metcalfe and David Boggs, describes Ethernet as a LAN system with such characteristics as shared communication, broadcast packet switching (all nodes hear the transmission), extension via repeaters, distributed control for packet transmission, and controlled behavior in the presence of interference or collisions.

While their paper describes operation on a coaxial-based line, these properties have also been central to noncoaxial Ethernet systems. The 802.3 standard describes the communication on a network employing the carrier sense multiple access with collision detection (CSMA/CD) access method. It includes sections for aggregation, multiple speeds, and full/half duplex operation.

There have been many versions of this ubiquitous protocol, including 10Base5, 10Base2, 10Base-FL, 10Base-T, 100Base-T, and 100Base-FX. Of these, 10Base-T, 100Base-T, and 1000Base-T (gigabit) are the most common and are the focus of this chapter. Network equipment is sometimes referred to as 10/100/1000 (or 10/100), Ethernet having the capability of running at 10, 100, or 1000Mbps. The popularity of Ethernet as a LAN protocol has forced personal computer and laptop vendors to include Ethernet ports and/or auxiliary slots for Ethernet cards on all their products.

Remember the Models

Ethernet governs the two lowest layers (Physical and Network) of the TCP/IP model. [Chapter 1](#) introduced RFC 1122, which requires nodes operating on a TCP/IP-based network to support the Ethernet encapsulation scheme described in RFC 894. Nodes on the network should also be able to receive frames described by RFC 1042 (IEEE 802.3 frames) and may support transmission of these frames. Today we see both types of frames coexisting on the network, but typically RFC 894 (Ethernet II) frames are generated by hosts. When an IEEE 802.3 frame appears, it is almost always from a network communication device such as a router, switch, or access point.

As previously stated, Ethernet resides at Layers 1 and 2 of the TCP/IP (or OSI) model. Layer 2 is further subdivided as shown in [Figure 2-1](#).

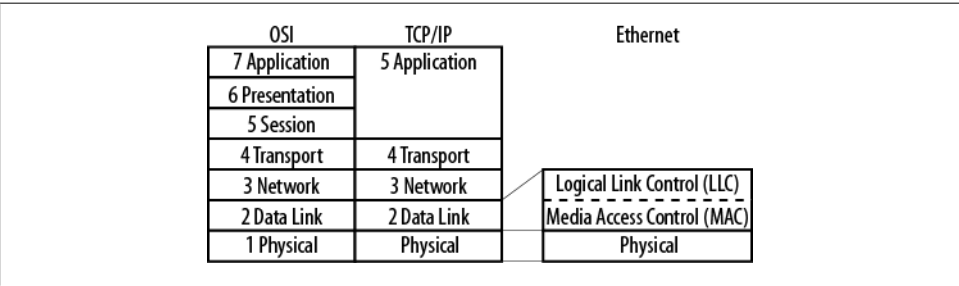


Figure 2-1. Models and Ethernet

The two sublayers are called Logical Link Control (LLC) and Media Access Control (MAC). The LLC sublayer is where the Ethernet frame and its associated fields are assembled and is similar to the IEEE 802.2 structure. The MAC sublayer is responsible for what is called the *access method*. Discussed in detail later in the chapter, the MAC sublayer detects the carrier, transmits and receives from the media, and passes frames to/from the LLC sublayer.

We know that encapsulation causes user data to be wrapped in headers from each layer in our networking model. As an example, a DHCP (BOOTP) packet is encapsulated first in UDP, followed by IP. This packet must then be placed in a LAN frame. This is true whether the network is wireless (802.11) or wired via Ethernet. [Figure 2-2](#) shows

the encapsulation concept and [Figure 2-3](#) shows an example of an actual Ethernet Type II frame carrying the information.

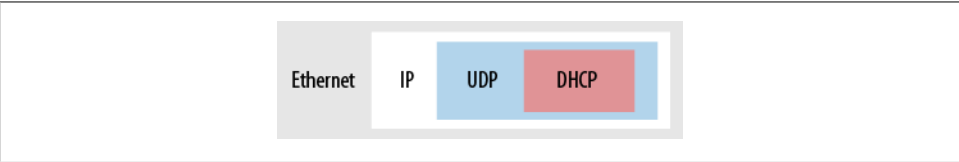


Figure 2-2. Encapsulation

```

Ethernet II, Src: Standard_44:12:65 (00:e0:29:44:12:65), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  Source: Standard_44:12:65 (00:e0:29:44:12:65)
    Type: IP (0x0800)
  Internet Protocol, Src: 192.168.16.195 (192.168.16.195), Dst: 255.255.255.255 (255.255.255.255)
  User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
  Bootstrap Protocol

```

Figure 2-3. Ethernet frame encapsulating DHCP

Structure

The Ethernet frame (Layer 2) in [Figure 2-3](#) is expanded in [Figure 2-4](#) to show several fields used to control various aspects of the transmission. When you capture traffic using Wireshark, protocols above Layer 2 are shown in their entirety. However, several parts or fields of the Layer 2 are not shown. [Figure 2-4](#) shows a simple Ethernet frame as defined in the standard.

Preamble	Destination MAC addr	Source MAC addr	Control	Data	FCS
8 bytes	6 bytes	6 bytes	2 bytes	46-1500 bytes	4 bytes

Figure 2-4. Ethernet fields

Preamble

The preamble is a series of alternating 1s and 0s that provide timing for the receiving interface. The Ethernet II preamble is eight bytes in length, with each successive byte repeating the 1-0-1-0 sequence. The 802.3 frame has a seven-byte preamble with the alternating 1-0-1-0 pattern, but the eighth byte is slightly different (10101011) and is referred to as a *start frame delimiter*, or SFD. The preamble and the SFD are invisible to packet analyzers.

Source and Destination MAC Addresses

A MAC address (also known as the hardware, Ethernet, or physical address) is the six-byte address encoded into the network interface card (NIC) of a particular machine. The Ethernet frame has two addresses—destination and source—with the destination transmitted first. MAC addresses are used to send frames to the correct recipients on the LAN. MAC addresses have no significance beyond a computer's own network, so the MAC addresses of machines beyond the local network are unknown. When transmitting outside the network, the MAC address of the default gateway is placed in the destination field.

Control Field (Type)

This is a two-byte field that describes what is contained in the data field. [Figure 2-3](#) shows a value of 0x0800 following the two MAC addresses. The “0x” means that it is a hexadecimal (hex) number. The hex decode of a Wireshark capture will only show “0800,” leaving out the “0x.” The value 0800 is the most common value for this field, indicating that an IP packet is encapsulated. Another common value for this field is 0806, which indicates an Address Resolution Protocol (ARP) message. This same two-byte field in an 802.3 frame indicates the length of the data field in bytes.

Data Field

All higher layers of the protocol stack are encapsulated in the data field, or payload. All of the traffic to be sent over the network must be encapsulated into the data field of an Ethernet frame. As indicated in [Figure 2-4](#), the minimum payload is 46 bytes and the maximum is 1500. These values are directly related to Ethernet operation. A payload of less than 46 bytes requires trailing zeros to be added to obtain the minimum of 46 bytes. An example of this is shown in [Figure 2-5](#). However, the padding is not to be included in the length calculation for the IP packet. If the data chunk is greater than 1500 bytes, it is split up into two or more frames to be sent across the network separately.

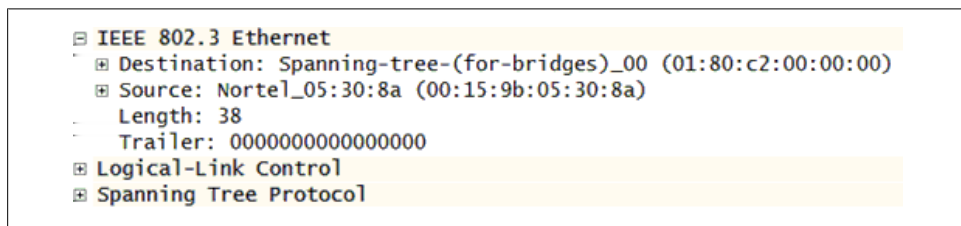


Figure 2-5. Use of trailer zeros

Frame Check Sequence

The Frame Check Sequence (FCS) is the last field of the Ethernet frame and is used for error checking. A 32-bit cyclical redundancy check (CRC-32) algorithm is computed using the entire frame's binary sequence, excluding the FCS field itself. This algorithm does error detection, but not error correction. In fact, it only checks for single bit errors. When a frame is created, the CRC calculation result is appended to the frame. Any node receiving or forwarding the frame will also calculate the CRC. The two values are compared and if they are different, an error has occurred. By default, switches calculate the CRC of each frame. A CRC error will result in the switch discarding the frame. Typically, error repair is left to the upper layers of the protocol stack or model. When errored frames are dropped, the TCP conversation will be missing packets, as indicated by out-of-order sequence numbers. Clients will ask for retransmission of the missing data.

Bit error rate (BER) refers to the number of bits that we can expect to transmit without a problem. BERs can range from 1 in 10^9 bits to less than 1 in 10^{12} on today's high-performance networks. This means that for every 10^9 or 10^{12} bits transmitted, only 1 will be in error, causing a CRC test to fail. Even in the early Ethernet standards, the BERs were limited to 1 in 10^9 .

Ethernet Type II vs. 802.3

As mentioned previously, Ethernet frames conform to two different formats. NICs and devices understand these variations and they coexist. Ethernet Type II is the standard used for IP-based data packets. IEEE 802.3 is often used with management protocols such as spanning tree. [Figure 2-6](#) depicts the two frame types for comparison.

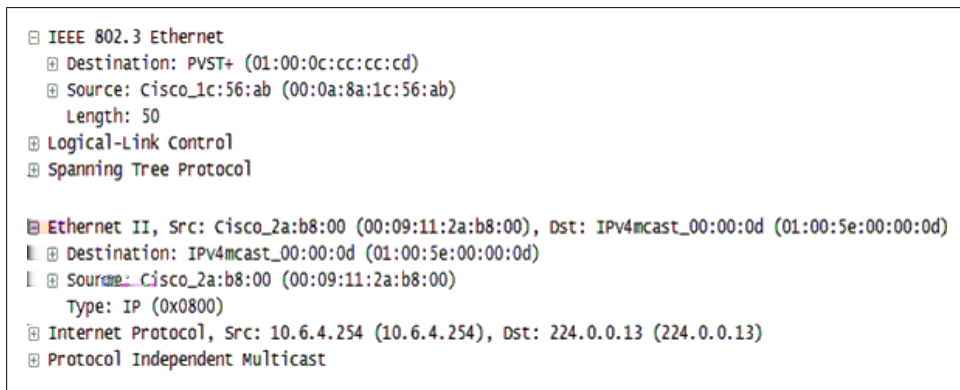


Figure 2-6. 802.3 vs. Ethernet Type II

The major difference between these two standards that can be seen with packet capture software is the two-byte control field. Ethernet Type II uses this as a type or protocol identifier, while 802.3 uses this field as a length value. All Ethertypes are greater than a base 10 value of 1536 (0x0600 in hexadecimal) and the most common of these are shown in [Table 2-1](#).

Table 2-1. Control field values

Hex value	Base 10 value	Meaning
0x0800	2048	IP packet
0x0806	2054	ARP packet
0x86DD	35425	IPv6

A complete list can be found at the IEEE Registration Authority. The website address is given at the end of this chapter. If the base 10 value of the control field is 1500 or less, the frame is 802.3 and the control field is a length value, as shown in [Figure 2-7](#). The standard actually considers any value below 1536 (0600 in hex) to be a length. However, RFC 1122 tells us that the maximum size of an Ethernet data field is 1500 and the maximum transmission unit (MTU) for 802.3 is 1492 bytes.

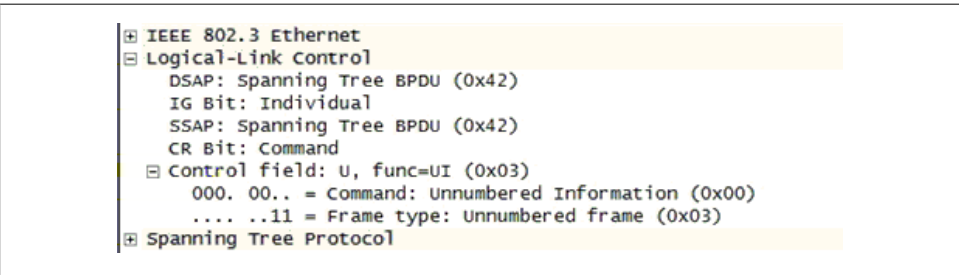


Figure 2-7. 802.2 LLC

While we're here, we might as well take a little closer look at the LLC portion of an 802.3 frame. Expanded, the LLC header provides a little more detail about the frame's functionality. Ethernet Type II frames use the control field for this purpose. The full details of the LLC can be found in the IEEE 802.2 standard.

The general form of an LLC packet includes four fields:

Destination Service Access Point (DSAP)

This first byte indicates the ending point or target for this frame. This address includes the individual or group bit at the beginning. An address of all ones is reserved for the global address. All zeros is a null address. A value (DSAP or SSAP) often seen on networks is AA, referring to the Subnetwork Access Protocol, or SNAP. SNAP provides support mechanisms for multiple protocols on the same Ethernet sublayer.

Source Service Access Point (SSAP)

The second byte indicates the starting point or reason for the frame. This address includes the command/response bit at the beginning. All zeros indicates a null address.

Control

These are values for the command and response functions, depending on the operation, as well as possible sequence numbers.

Information

This field is the data carried by the frame. [Figure 2-8](#) shows that the frame is carrying a Spanning Tree Protocol packet, and shows the general format for 802.3 frames with 802.2 LLC. Spanning Tree is a standard management protocol used on Ethernet networks.

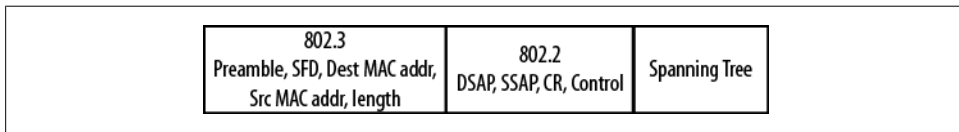


Figure 2-8. 802.3 frame with 802.2 header

MAC Addresses—Another Look

Before we get into Ethernet operation, it is worth spending some time on the various MAC addresses seen on a network. These addresses can drastically affect how frames are processed. Generally, MAC addresses are divided into two parts: a three-byte vendor code and a three-byte host ID. In the example shown in [Figure 2-6](#), the Ethernet Type II source MAC address is 00:09:11:2a:b8:00. The vendor code is 00:09:11, which corresponds to Cisco. Wireshark can interpret this value for us because the value is registered and this list is publicly available at the IEEE site. A handy lookup tool is available at <http://standards.ieee.org/develop/regauth/oui/public.html>. The unique value for the network node is 2a:ba:00. So, a single vendor code can have as many as 2^{24} possible host addresses. These values are almost always written in hexadecimal notation, though the frame is transmitted in binary.

There are three different types of MAC addresses on an Ethernet network: unicast, broadcast, and multicast. *Unicast* MAC addresses are those assigned to individual nodes, and the first byte is always 00. In an Ethernet frame, the source MAC address is always a unicast address. [Figure 2-9](#) shows a unicast-to-unicast communication. That means the source and destination addresses both begin with 00 and correspond to individual machines. We call this a *unicast frame*. In the case of the destination, the vendor code is 00:19:55 and the unique node ID is 35:1a:d0.

```
[-] Ethernet II
    [+] Destination: 00:19:55:35:1a:d0
    [+] Source: 00:11:25:43:49:97
        Type: IP (0x0800)
    [+] Internet Protocol
    [+] Internet Control Message Protocol
```

Figure 2-9. Unicast Ethernet frame

Broadcast frames are those sent by a single node (unicast address) to everyone on the local network. This special hexadecimal address (ff-ff-ff-ff-ff-ff) is used for several different messages, but a couple of the most common are ARP and DHCP requests. A hexadecimal value of ff corresponds to a base 10 value of 255. [Figure 2-10](#) shows a broadcast frame.

```
[-] Ethernet II
    [+] Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    [+] Source: D-Link_c4:40:7f (00:50:ba:c4:40:7f)
        Type: IP (0x0800)
    [+] Internet Protocol
```

Figure 2-10. Broadcast Ethernet frame

Broadcast frames are read by all nodes on a network. They are also forwarded everywhere by Layer-2 networking equipment. When a switch receives a broadcast frame, it is sent out every port except the one through which the frame arrived. Routers will not forward broadcast frames. Routers are said to be the boundary of the *broadcast domain*. In fact, routers generally do not forward Layer-2 frames to other networks. Stated another way, no network but your own will ever see the MAC addresses used on your network.

Multicast frames are created by a single host (unicast) but destined for a subset of the entire network. Multicast is important when a message must be sent to a particular process or group of nodes. One example might be the wireless equipment on a network. A controller or node might send out a multicast frame that reaches all devices with a particular vendor code. Another example is the Spanning Tree Protocol. Switches engaging in the Spanning Tree Protocol send and receive frames with a particular reserved multicast address. Nodes ignore this address. Multicast frames will have 01 as the first byte of the MAC address.

The example in [Figure 2-11](#) is a spanning tree frame called a *bridge protocol data unit*. The source address is a unicast and the destination is 01:80:2c:00:00:00, which corresponds to the MAC address reserved for spanning tree. This is an 802.3 Ethernet frame rather than an Ethernet Type II frame.


```
IEEE 802.3 Ethernet
  Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
  Source: Nortel_05:30:8a (00:15:9b:05:30:8a)
  Length: 38
  Trailer: 0000000000000000
  Logical-Link Control
  Spanning Tree Protocol
```

Figure 2-11. Multicast Ethernet frame

Ethernet Operation

If you imagine yourself as a computer trying to communicate on a network fully populated with a large number of other computers, you might discover that there are a significant number of issues associated with trying to be understood. For example:

- How would you ensure the correct destination received the transmission?
- How would you determine how fast or how slow you were supposed to send the data?
- How would you decide which computers had permission to speak or transmit?

These issues are similar to having a conversation with a group of your friends. Who is talking? For how long? Can we interrupt? Like many conversations with our friends, some topologies are easier to manage than others. For these reasons, every single LAN protocol has a set of rules.

As a protocol, Ethernet is pretty straightforward. As for the first issue, Ethernet uses MAC addresses to uniquely identify the network nodes. When a frame is transmitted, the recipient is specified in the Ethernet header. Moving to the second issue, data rate is a function of the NIC. NICs are normally capable of two or three speeds, including 10Mbps, 100Mbps, and 1Gbps. So either the NIC negotiates an acceptable data rate with the network or the NIC will use an incoming frame's preamble to help it sync up with the incoming transmission.

The third issue is a little more complicated. Before a node can transmit, it must determine if the wire or medium is clear for transmission. This is handled via the access method. Specifically, Ethernet uses CSMA/CD along with a truncated binary exponential random backoff algorithm. It's quite a mouthful, but simpler than it sounds. First, the node will listen for other transmissions. "Hearing" none, it will assume the medium is clear and begin its own transmission. If the line is not clear, the node waits for the transmission to complete and then sends its own frame.

Shared Media

Early Ethernet operated on a bus topology, meaning every node on the network can hear what you transmitted and vice versa. In fact, early generations such as 10Base5 and 10Base2 actually connect nodes together via coaxial cables and tap into the central shared conductor. The next generation was called 10Base-T and dropped the use of coaxial cable in favor of unshielded twisted pair (UTP) as the media, but it is still a bus. The naming convention (10Base-T) is described in 802.3 and indicates the speed (10, 100, or 1000Mbps), type of transmission (baseband or broadband), and the type of media or distance, T indicating twisted pair. Earlier standards such as 10Base5 use the trailing number as a distance value to indicate the maximum network diameter. For example, 10Base5 has a maximum collision domain distance of 500m.

In a UTP-based bus topology, all of the transmissions use the same pair of wires. The transmission eventually winds up on the receive pair of wires for every single node. The central node is a hub or repeater. With bus topologies, if two separate transmissions are initiated at the same time, they will collide somewhere on the network, causing a spike in voltage or power (Figure 2-12). Just like vehicular collisions, Ethernet frame collisions are bad. In the event of a collision, the two nodes involved must back off from the transmission and try again later.

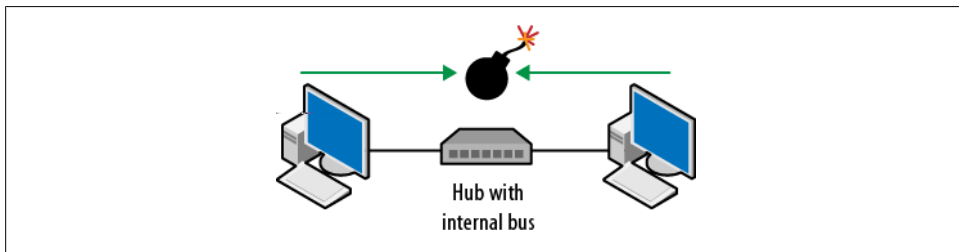


Figure 2-12. Collision on a hub-based topology

Network nodes can detect when a collision occurs, because their NICs have transmit and receive wire pairs. Thus, if a signal is detected on your receive wires while you were transmitting, another node must be transmitting at the same time. The frames have collided with yours and the collision now propagates to the receive pair. Normally, a frame from a single node will effectively “fill up” the entire network for the duration of the transmission, ensuring all other network nodes will remain quiet.

To ensure we can detect all possible collisions, regardless of when the transmission started, we have rules that must be obeyed regarding frame size, bit rate, and maximum network diameter. Ethernet frames must be a minimum of 64 bytes in length (recall that the data field can range in size from 46 to 1500 bytes) measured from the destination MAC address field to the CRC field, as shown in Figure 2-13.

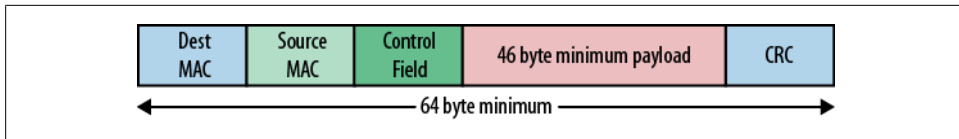


Figure 2-13. Ethernet minimum frame size

The minimum size restriction prevents the generation and transmission of very small frames. A 10Base-T network is limited to 500m end to end. The goal is to prevent a node from transmitting the last bit of a frame before the first bit arrives at the destination. If we allowed the network size to exceed the limit, or if we transmitted really small frames, it would be possible for the entire frame to leave the NIC and then get destroyed in the network. Violation of these restrictions makes it possible for a transmitting node to finish putting a frame onto the network before the leading edge of the frame reaches its destination. Should a collision occur, the transmitting node will have no idea that its own frame was involved in the collision. At the other extreme, nodes with large frames may monopolize the network.

Any node capable of detecting a collision is a member of the same *collision domain*. A collision domain is the distance that collision electrical noise travels. Hubs will forward collisions, but switches and routers will not. More on these devices can be found in the [Chapter 5](#).

The Ethernet standard includes guidelines for the construction of shared (bus topology) Ethernet networks. A 10Base-T network should follow a 3-4-5 rule—three populated segments, four repeaters or hubs, and a maximum distance of five hundred meters ([Figure 2-14](#)). The rule imposes a maximum distance between any two Ethernet UTP endpoints of 100 meters.

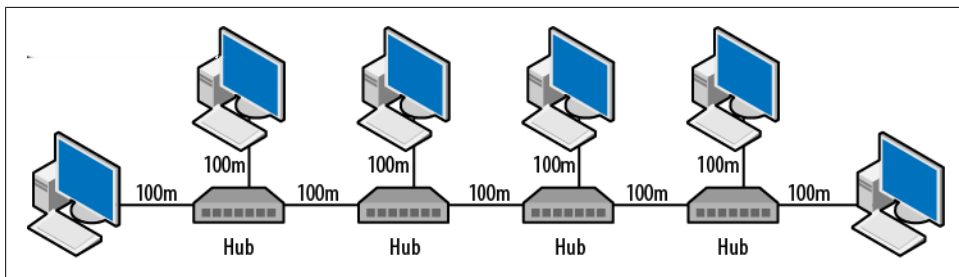


Figure 2-14. 3-4-5 rule for 10Base-T Ethernet

For 100Base-T (the 100Mbps upgrade for 10Base-T), we use the 1-2 rule, which means that only one repeater or hub is allowed and the maximum network diameter is two hundred meters. This rule allows either translational or nontranslational hubs. Non-translational hubs do not convert between media types. An exception to the 1-2 rule is adding a second hub if it is nontranslational (Class II) and placed close by the other

hub, as in a wiring closet. The Ethernet standard specifies that, “It must be possible [for] any two DTEs on that network to contend for the network at the same time.”

Remember that these rules are for shared Ethernet, meaning hubs, bus topologies, and a broadcast system. The minute a device like a switch is used, these rules go out the window, because the network behavior changes. On a practical note, today’s networks are usually built with switches, so these rules serve as background. However, Ethernet default configuration settings are typically aligned with the shared network mentality.

We know that Ethernet has a minimum frame size of 64 bytes, or 512 bits. A frame of this size on a 10Base-T network (10 million bits per second) takes 51.2μsec to transmit. This is referred to as the *slot time*. If a node has a frame to transmit, it must listen to the media for at short period of time called the *interframe gap*. This minimum wait time prevents transmitting into a frame already on the network and allows station a chance to reset. An interframe gap (time distance between two frames) of 96 bits (9.6μsec) also prevents a node from transmitting one frame after another. For 100Base-T, the interframe gap is shortened to .96μsec. The slot time is also shorter, based on the faster transmission time of each bit. Just for fun, 1000Base-T has a slot time of .096μsec and a corresponding drop in slot time.

When the media is free and the wait is over, a frame may be sent via the transmit pair. If a spike in voltage or power (a collision) is detected on the receive pair while the frame is being transmitted, the stations involved in the collision know that their frames were destroyed, because they were obeying the minimum frame size rules. The transmitters now have two tasks—first, they issue a 32-bit jam signal to ensure all devices are aware of the collision, then they back off and wait for another chance to transmit.

Here is where the truncated binary exponential backoff algorithm comes in. The back-off time is dependent upon the following formula, where *r* is a random integer between the two values and *n* is the number of transmission attempts:

$$(0 > r > 2^k) * \text{slot time} \\ k = \min(n, 10)$$

If, after sixteen attempts, a node does not manage a transmission, it stops trying.

Physical Layer

So far we’ve discussed the operation and framing of Ethernet from Layer 2 of the protocol stack. The Physical Layer specifies the electrical and mechanical properties such as voltage levels, encoding schemes, and the connectors. On a practical note, users are given limited options when configuring a network. We no longer use NICs that have different types of connector. Connectors are almost always RJ45 terminations for UTP. Before we look at the electrical details, it is worth spending some time on the cabling and connectors. It turns out that this is where we make many of our mistakes.

Cabling

UTP wiring is the most common network media type and is used for VoIP phones, Token Ring, FDDI, Ethernet, and many others. UTP describes the construction of the actual wiring. That is, eight conductors, twisted into four pairs, with nothing but plastic protecting the copper from the environment. The conductor size is American Wire Gauge (AWG) 22-26. The RJ45 jack (male or female) has eight pins for these connections and a locking tab to keep it secure in the outlet.

Figure 2-15 shows both the male and female RJ45 jacks. On the male, pins 1 is at the top and is connected to the orange pair along with pin 2. In the female jack, pin 1 is on the right. UTP cabling is tested for performance and given a rating, or *category*, based on this performance. The standard categories and some of their performance ratings are given in Table 2-2.

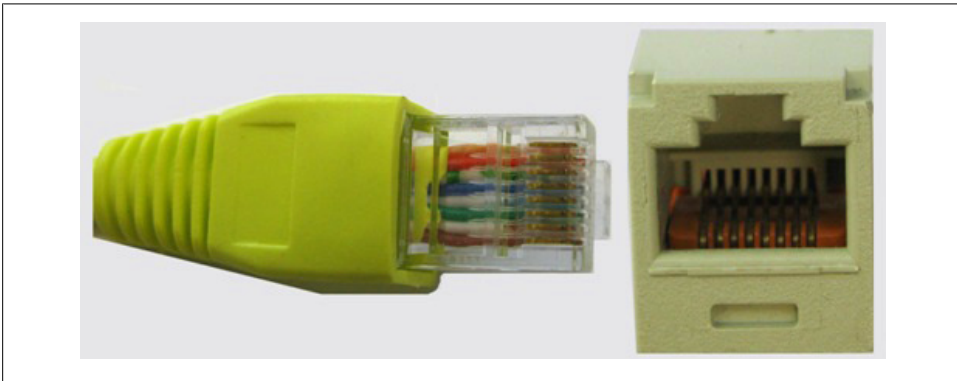


Figure 2-15. RJ45 male and female ends

Table 2-2. Cable specifications

	Attenuation (insertion loss)	Reflection	Near end crosstalk	Max frequency
Category 5	22dB	15.1dB	32.3dB	100MHz
Category 5e	22dB	20.1dB	35.3dB	100MHz
Category 6	19.8dB	20.1dB	44.3dB	250MHz



Values given in Table 2-2 are for 100MHz operation and there are differences between cable types. Additionally, Category 3 cables are not rated for performance at these frequencies.

As the category number increases, so does the range of frequencies that can be sent over the cable. This means a greater data rate and improved performance. These improvements are due largely to the change in the construction of the cable. The biggest physical difference is in the number of twists per inch in each pair of conductors. The higher the number of twists, the better the cable is at avoiding crosstalk and interference problems. [Figure 2-16](#) shows the differences between early Category 3 (Cat 3) cable through Category 5e (Cat 5e) to Category 6 (Cat 6).

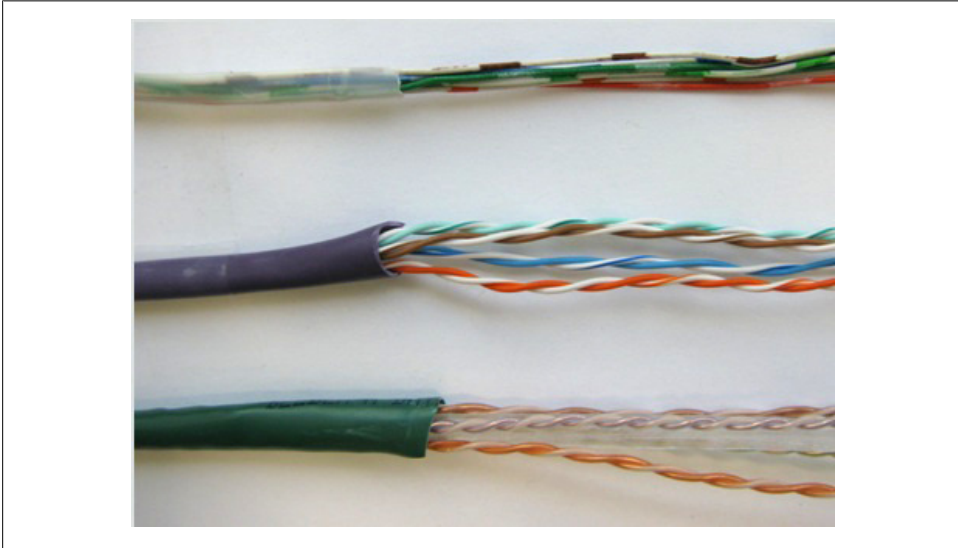


Figure 2-16. Cable construction comparison

A close look at [Figure 2-16](#) reveals several interesting characteristics about the cable construction. First, Cat 3 cables (top) have very few twists when compared to the others. In fact, 3–5 twists per foot is the standard. Cat 5e has 3–5 twists per inch and if we look closely we can see that the individual pairs have different twist densities. Cat 6 cable (bottom) can have even more twists, but may add the plastic separator and the bond the pairs together. These make the cable an excellent transmission medium, although it makes life more difficult for the cable installer when terminating cables manually, as the pairs typically have to be separated.

In addition to categories, individual conductors can be manufactured differently. Permanent cabling (horizontal cabling) is installed in the wall connecting the female jack to the wiring closet distribution frame. This cabling is fabricated using solid copper for each of the eight individual conductors. Patch cables connect your computer to the wall jack and make connections in the data closet. Patch cables are constructed using stranded copper. If you were to take one of the eight conductors and strip off the insulation, you would see many small strands of copper rather than one thicker piece. This stranding makes the cable more flexible and therefore less likely to break than

solid core, but it increases the attenuation of the signal it carries. For this reason, we usually like to keep the patch cables as short as possible. These differences can be seen in [Figure 2-17](#).



Figure 2-17. Stranded vs. solid core cable

Both permanent cabling and patch cables are straight-through, which means that pin 1 on one end goes to pin 1 on the other and so on. These connections follow the EIA/TIA 568 cabling standards. For data networks, we are actually concerned with EIA568B, while telecommunications networks generally employ EIA568A. A straight-through data cable is terminated on both ends using 568B.

At the Physical Layer, Ethernet uses only pins 1, 2, 3, and 6 for transmission and reception ([Table 2-3](#)).

Table 2-3. Ethernet pin usage

1	Tx+
2	Tx-
3	Rx+
6	Rx-

Pins 4 and 5 can be used to handle traditional telephone connections and the telephone jack (RJ11 style) fits right into the center of an RJ45. In addition, modern communications often deploy power over Ethernet, or PoE. This also runs over unused connectors. Power over Ethernet is standardized in IEEE 802.3af and 802.3at. These standards describe the method and electrical characteristics used to energize devices such as VoIP phones and access points over the unused pairs of the Ethernet cable. This is a much more convenient solution than installing power outlets everywhere.

In addition to categories and construction, there are several different nonpermanent cable types used on a network, including straight-through, crossover, and rollover. They all use the same stranded copper, but the pins don't always start and end in the same place. Standard straight-through patch cables are used to connect a computer to a hub or switch. However, they do not work when connecting computers directly together or when interconnecting some network devices. A standard patch cable is likely a stranded, Category 5e, straight-through (568B to 568B) data cable terminated with RJ45 jacks. But most people just call it an Ethernet cable, even though the exact same cable might be used for a Token Ring network or a VoIP phone.

A crossover cable allows two computers or two switches to be connected directly together by crossing the receive wires to the transmit lines. A crossover cable is terminated using 568B on one end and 568A on the other. To simplify things, we can look at the orange and green pairs. [Table 2-4](#) depicts the mapping, or *pinouts*, of 568A and 568B.

Table 2-4. 568A and B pins

T568A for patch cable		T568B for patch cable	
Pin	Wire color	Pin	Wire color
1	White/green	1	White/orange
2	Green	2	Orange
3	White/orange	3	White/green
4	Blue	4	Blue
5	White/blue	5	White/blue
6	Orange	6	Green
7	White/brown	7	White/brown
8	Brown	8	Brown

This crossing is normally handled by the switch or hub.

When a network administrator configures a device such as a router, it is often accomplished via a *rollover cable* connected to a console port. A rollover cable maps pin 1 to pin 8, pin 2 to pin 7, pin 3 to pin 6, and pin 4 to pin 5. We use these cables to connect from a computer serial (COM) port to a switch/router console port, and EIA232 is used for communication. When connecting a DB9 serial port to a router console port, the rollover cable is connected to a terminal adapter that converts the RJ45 wiring to DB9. Often, these cables come premade with the conversion built in. [Figure 2-18](#) shows networks utilizing the various cable types.

The top image in [Figure 2-18](#) shows two computers directly connected to each other. The required cable is an Ethernet crossover. The same is true of the switch-to-switch connection if an uplink port is not present. The straight-through Ethernet patch cables are running between the computer and the switch. There is also a straight-through cable between the router and the switch. Finally, the management station is connected to the router console port via a rollover cable. Of course, ports on newer devices negotiate many of the cross connections for us, but we're better off knowing how things work.

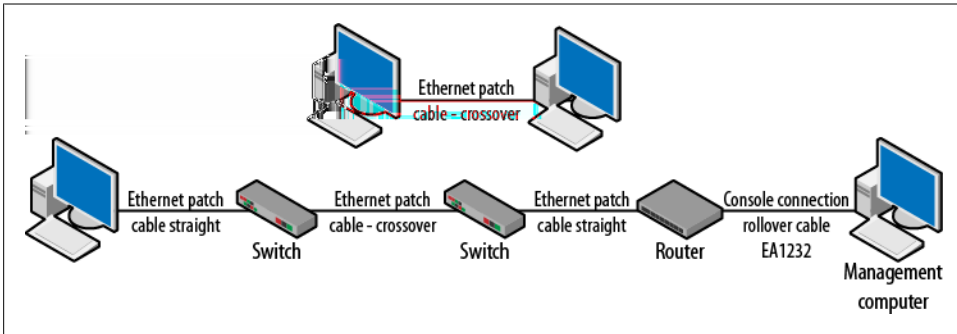


Figure 2-18. Connection and cable types

Encoding

Now that you understand framing and operation, and have connected the computer to the network, what is actually going on over the orange and green pairs? When the frame is sent, the NIC generates an electrical signal conveying the binary 1s and 0s (encoding) that are read by the receiver on the same network. The main electrical features of the Physical Layer, including encoding, are outlined in this section.

10Base-T

- Connector type and media—RJ45, UTP
- Encoding—Manchester

Manchester encoding specifies that a binary 1 is indicated when the voltage transitions from a low point to a high point in a single bit interval and a binary 0 is the exact opposite. This transition also aids in timing of the signal and it means that when a series of 1s (or 0s) is transmitted, a single bit interval may have two transitions. Examples are shown in [Figure 2-19](#).

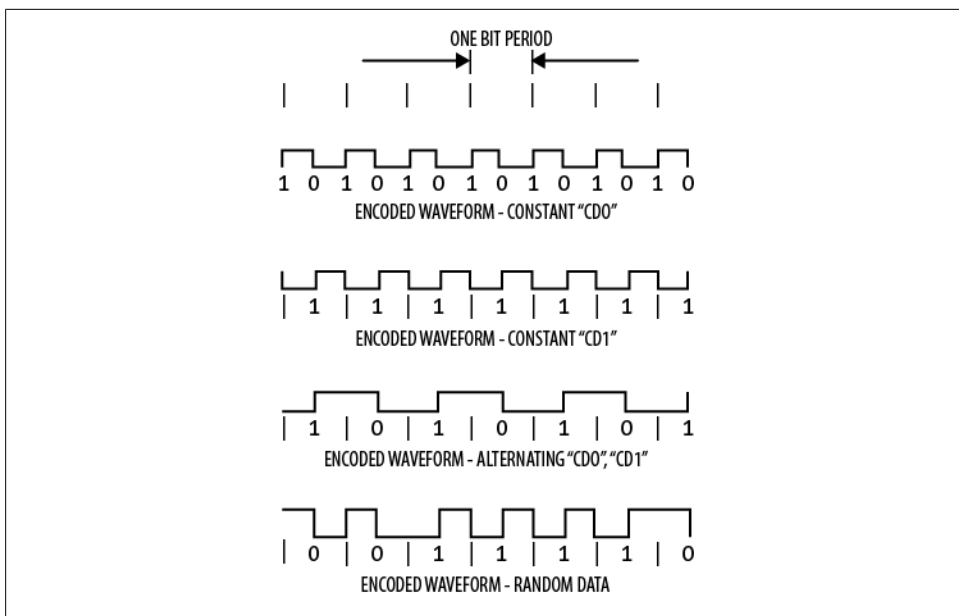


Figure 2-19. Manchester encoding (source: IEEE 802.3-2002)

100Base-T

- Connector type and media—RJ45, UTP
- Encoding—NRZI

For the most part, 100Base-T (also known as Fast Ethernet or 802.3u Ethernet) is functionally the same at Layer 2 as 10Base-T. However, when we move from the slower speeds to the increased data rates, we have shorter bit times or intervals, less time used to transmit frames, and the rate at which bits are produced is faster. So the real changes are at the Physical Layer.

An interesting point is that 100Base-X (the general Fast Ethernet specification) imports the signaling used on Fibre Distributed Data Interface (FDDI) networks. Specifically, the signaling interface runs at 125Mbps but exchanges four-bit (nibble) data chunks for five-bit patterns. This is called *4b/5b substitution*. This also means that even though the signaling is 125Mbps, the effective throughput is 100Mbps.

The actual signaling is called NRZI, or *nonreturn to zero inverted*. With NRZI, a binary 1 is indicated by a polarity transition and a binary 0 is indicated by the absence of a transition (Figure 2-20). A basic problem with this encoding is that when a series of 0s is transmitted, synchronization can be lost. The 4b/5b substitution provides for additional transitions by ensuring that 1s are injected into the data stream.

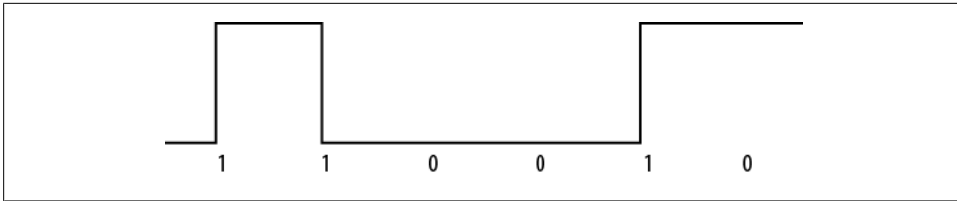


Figure 2-20. NRZI encoding

1000Base-T

- Connector type and media—RJ45, UTP
- Encoding—4D, five-level Pulse Amplitude Modulation (PAM5)

With the move to 1Gbps speed (1000Base-T), we have similar changes to the frame transmission rates. It turns out that supporting this data rate on Cat 5 copper is very difficult, so the standard calls for the use of additional wire pairs. Each pair actually transmits 250Mbps with the aggregate result of 1000Mbps. The signaling rate is still 125Mbaud. To manage these values, the encoding scheme is quite complex, utilizing what is called a four-dimensional symbol and five different voltage levels. While this complexity is a little beyond what we want to accomplish here, the important thing for us is that these symbols allow the receipt of data even while transmitting on the same pair, preserving full duplex operation.



All of the Ethernet standards discussed here have fiber transmission specifications. The use of fiber is typically used for high speed uplinks or backbone transmissions. Fiber to the desktop, while certainly not unheard of, is rare due to expense and the additional management.

Other Types of Signaling

In addition to the Ethernet frame, there are a couple of signals used to indicate health/status of the line or the capabilities of the link. This form of signaling is considered out of band because it occurs between the frames flowing across the network cabling.

Link Pulse

What activates the link light? Answer: the link pulse. There are two types of link pulse signals—normal and fast. 10Base-T nodes support only the normal link pulse, or NLP. NLP indicates link status. The NLP is simply a blip sent every $16 \pm 8\text{ms}$ while the link is idle. Devices on either end of a link send each other these pulses to indicate that the link is up.

Autonegotiation

Autonegotiation uses the same NLP, but for a different purpose. In this chapter we have discussed full and half duplex and 10/100/1000Mbps systems. The question is, how do we determine the correct communication parameters? Autonegotiation replaces a single NLP blip with the fast link pulse (FLP) signal. There are 33 pulse positions in the time reserved for a NLP pulse, and these are divided into 17 odd and 16 even positions. The FLP blips are spaced every $62.5 \pm 7\mu\text{sec}$. The odd positions are for link pulse and the clock. The even positions are used for autonegotiation data. If a pulse is present, it indicates a binary 1; if not, the position is a binary 0.

The first 13 of the 16 data bits do most of the work. They are broken up into five-bit selector and eight-bit technology fields. The selector bits provide information on protocol type, which will almost always be set to 802.3. The technology bits provide details on Ethernet flavor (10Base-T, 100Base-T, etc.) and support for full duplex. This list is prioritized so that the highest common denominator is chosen. The last three bits are remote fault, acknowledge (receipt of partner link code), and next page. In the event that an NLP is received in response to the FLP, it is assumed that a 10Base-T node is on the other end.

Topologies

In the previous sections, we established that traditional Ethernet is a broadcast, shared media that uses a “listen then transmit” approach. In addition, if a node transmits without obeying the rules, it destroys any other transmission. These characteristics make Ethernet a *half-duplex* system. Hubs and repeaters have an internal bus. So, while they work to extend the network, the half-duplex operation remains unchanged.

With the addition of a switch as the central node, or if Ethernet nodes are connected directly together, the devices may negotiate full-duplex operation. In these situations, the shared portion of the network is removed. This means that simultaneous transmissions can exist and a node can both transmit and receive at the same time.

Traditional Ethernet configurations such as 10Base5 are called bus topologies, based on their operation. We might go one step further and say that a network of this type is physically wired as a bus and logically acts like a bus. With the addition of a hub, the network still acts like a bus in that every node can “hear” all that is transmitted on the network, but the wiring looks like a star or tree. For this reason, we often call 10Base-T shared Ethernet a *star wired bus* (Figure 2-21).

When we move to contemporary networks, it is far more common to have a switch as the central point. The network is no longer a bus topology and collisions are almost a thing of the past. With a switch, the network is a physical star as well as a logical star. Capturing packets is a bit more difficult, as we must set up monitor sessions or mirrored ports on the switches.

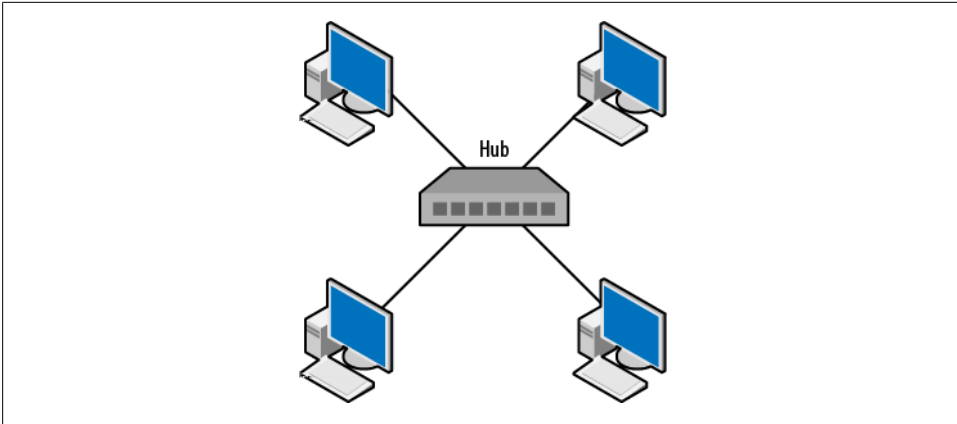


Figure 2-21. Star wired bus with a central hub

Final Thoughts on Ethernet

Almost anyone can build an Ethernet network. The fact of the matter is that because of features like autonegotiation, and since we do not have much choice in the selection of signaling, most of the configuration options are handled without any interference from us. In many cases, network administrators can simply take a switch (no hubs, please) out of the box, plug it in, and presto, instant network. Today, many ports will even autoconfigure themselves if you plug in the wrong cable. Of course, we prefer not to test these features, especially if someone is watching.

The real work comes in trying to understand what is happening when things go wrong, optimizing performance, or improving security. It is then that knowledge of protocol structure and operation become critical. There are times when human interference is essential, and preferably it is informed interference.

Reading

RFC 894: “A Standard for the Transmission of IP Datagrams over Ethernet Networks,” C. Hornig

RFC 895: “A Standard for the Transmission of IP Datagrams over Experimental Ethernet Networks,” J. Postel

RFC 1042: “A Standard for the Transmission of IP Datagrams over IEEE 802 Networks,” J. Postel, J. Reynolds

ISO 9314-3:1990: “Information Processing Systems – Fibre Distributed Data Interface (FDDI)-Part 3 Physical Layer Medium Dependent (PMD)”

IEEE 802.3-2002: “Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications”

IEEE 802.3u: “Supplement to CSMA/CD Access Method and Physical Layer Specifications – MAC Parameters, Physical Layer, MAU and Repeater for 100Mbps Operation”

IEEE 802.3ab: “Supplement to CSMA/CD Access Method and Physical Layer Specifications – Physical Layer Parameters and Specifications for 1000Mbps Operation over 4-pair Category 5 Balanced Copper Cabling”

IEEE Registration Authority: <http://standards.ieee.org/develop/regauth/general.html>

For Ethertypes: <http://standards.ieee.org/develop/regauth/ethertype/eth.txt>

Summary

Ethernet is by far the most used LAN standard for contemporary networks. Existing at Layers 1 and 2 of our networking models, Ethernet drives many of the decisions we make regarding network equipment. The signaling, physical characteristics, framing, low-level error checking, and operation of LANs are all determined by this protocol.

Historically, Ethernet has used several media types and topologies. Today, we consistently see the use of UTP for end nodes wired in switched star or tree configurations, with fiber typically reserved for network backbones.

Ethernet is standardized in the IEEE 802.3 series.

Review Questions

1. What are the two sublayers for Ethernet at Layer 2?
2. What are the two main parts of a MAC address?
3. What are the three types of destination MAC address?
4. A collision domain ends at what device?
5. What are two differences between Ethernet II and 802.3 frames?
6. What is stranded UTP used for?
7. What is the name of the Ethernet access method and the “wait” algorithm?
8. Counting the preamble and the CRC, what is the maximum size for an Ethernet frame?
9. How many data bits are part of the autonegotiation fast link pulse?
10. Ethernet has error-detection capability, but not error-correction capability. True or false?

Review Answers

1. Logical link control (LLC) and media access control (MAC)

2. Vendor code and unique node ID
3. Unicast, broadcast, multicast
4. Switch or router
5. Preamble (preamble and start frame delimiter) and control (length)
6. Patch cables
7. Carrier sense multiple access with collision detection (CSMA/CD), truncated binary exponential random backoff
8. 1526 bytes
9. 16
10. True

Lab Exercises

Activity 1—Basic Framing

Materials: Wireshark and a computer with an active connection

1. Open up a command shell and run `ipconfig /all`.
2. Identify the IP mask, default gateway, and DNS.
3. Start Wireshark.
4. In the command shell, ping your default gateway or other nodes on the network.
5. Within Wireshark, examine the packets that result from this command.
6. Identify the individual fields in the Ethernet header.
7. What are the values for each field? Does the value found in the control field match the payload?
8. What fields of the Ethernet frame are not displayed? Can you verify the addresses from step 2?

Activity 2—Control Field Values

Materials: Wireshark and a computer with an active connection

1. Capture packets continuously.
2. Examine the control field of the Ethernet frame until you identify at least one other control field type.
3. What type of frame is this?
4. Why does it have a different value from the previous frame?

Activity 3—Addressing

Materials: Wireshark and a computer with an active connection

1. Capture packets continuously.
2. Identify a frame sent by your computer by matching the MAC found in the source address field.
3. What type of address is this?
4. What is your vendor code?
5. What is your unique ID?

Activity 4—Destination Addresses

Materials: Wireshark and a computer with an active connection

1. As you capture, start examining the destination MACs in the frames you see.
2. What are the three types of destination address?
3. Collect frames matching each of these destination types.
4. Can you identify the purpose of these frames?

Activity 5—Logical Link Control

Materials: Wireshark and a computer with an active connection

1. Begin capturing with Wireshark.
2. Examine the captures until you find an Ethernet Type II frame.
3. Continuing capturing until you find an 802.3 frame.
4. What are the differences between these frame types? What does Wireshark show you?
5. What was the purpose of the 802.3 frame?
6. Decode the 802.2 header within the 802.3 frame. What do the subfields mean?

Internet Protocol

“During my service in the United States Congress, I took the initiative in creating the Internet.”

—Al Gore

“He [Al Gore] is indeed due some thanks and consideration for his early contributions.”

—Vint Cerf

As stated in [Chapter 2](#), the language of the Internet and of the networks connected to the Internet is TCP/IP. This chapter examines the later part of this protocol pair. The Internet Protocol (IP) exists at Layer 3, regardless of which model you are using as a reference. It is often referred to as a “best effort” protocol, which simply means that IP provides very little in the way of connection or error control. Communication networks rely on upper-layer protocols such as TCP and the associated applications to handle these issues. However, all applications and processes running on the network have one thing in common—they all use IP. So, it is critical that we understand the operation of this ubiquitous protocol. This chapter takes an in-depth look at the protocol fields and their uses, operations, and the addressing used for networks today.

Protocol Description

IP has been around for more than three decades. Perhaps the easiest and best place to start is with RFC 791, titled “Internet Protocol DARPA Internet Protocol Specification.” This RFC was written in 1981, and the following quote gives some indication of its roots and age:

This document is based on six earlier editions of the ARPA Internet Protocol Specification, and the present text draws heavily from them.

This early document also describes IP as the protocol that “provides for transmitting blocks of data called datagrams from sources to destinations.” Today, we use the terms

packets and *datagrams* interchangeably, but the goal is the same. Every time a node connected to a network tries to communicate with another node, the transmission is broken up into these datagrams or packets. For example, the request to see a webpage and the delivery of the web content returned to the desktop are accomplished via IP packets. The size and number of the packets depends on the amount of information. The device primarily responsible for getting these packets to the correct destination is a router. This also means that every single IP packet must have all of the information necessary to be routed independently from all other packets.

Structure

IP packets are encapsulated in whatever Layer-2 protocol is running. Today, this would mostly commonly be either Ethernet or 802.11. IP packets have a payload field into which the upper-layer protocols and data are inserted, as shown in [Figure 3-1](#).

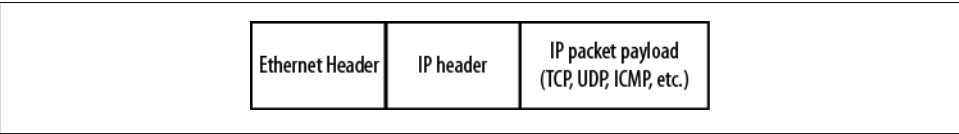


Figure 3-1. Basic IP encapsulation

[Figure 3-2](#) shows the basic form of an IP packet and is taken directly from RFC 791. This is perhaps the most common image used to describe this structure, but it can be difficult to understand. The figure is read from left to right and is in rows of four bytes (32 bits) each. Counting begins at zero, so the scale at the top ranges from 0–31. [Figure 3-3](#) depicts an actual IP packet. These two figures will help you gain an understanding of the fields used in IP packets.

[Figure 3-3](#) shows this encapsulation for an ICMP packet. As you can see, the Layer-2 protocol is Ethernet and the IP header has been expanded. Whatever follows the IP header is actually considered to be in the IP packet payload.

The IP packet begins with the highlighted line “Internet Protocol.” This line is not actually part of the datagram. It is important to start packet analysis at the correct point. For this reason, the hexadecimal values for the IP packet have been included and are also highlighted when the section is selected.

Version

This is a straightforward indicator of the format currently used. Today, the only variation would be IPv6, which would significantly change the structure of the packet. The Wireshark decode describes this, and since 4 in base 10 numbers is also 4 in base 16, the hexadecimal also shows the value of 4.

Header length

Internet Header Length (IHL) is the number of four byte “words” at the beginning of the IP packet. The RFC specifies that the minimum number of bytes will be 20

field are single bits to be set for delay, throughput, and reliability, as shown in [Table 3-1](#). While there is built-in capacity for varying degrees of quality of service (QoS) and the values are included below, this field is almost never used. This means that this one-byte field typically has a value of 0, as is the case in [Figure 3-3](#). In addition, when QoS is implemented, it is usually done without utilizing the precedence specified in the RFC.

Table 3-1. IP ToS bits

Bits 0–2: precedence	Bit 3: delay	Bit 4: throughput	Bit 5: reliability	Bits 6–7: reserved for future use
000—routine	0 = normal delay	0 = normal throughput	0 = normal reliability	Always 0
001—priority	1 = low delay	1 = high throughput	1 = high reliability	
010—immediate				
011—flash				
100—flash override				
101—CRITICAL/ECP				
110—internetwork control				
111—network control				

Instead of ToS values, the Differentiated Services and Diff Serv Code Points (DSCP) are used to mark and handle IP-based network traffic. An edge device such as a router will modify the IP header to include a nonzero value where the ToS is found. Subsequent routers will act on this value based on the treatment configured for that value. An example of an IP packet that has been modified is shown in [Figure 3-4](#).

Total length

This two-byte field is the size of the data in bytes including the header. In the case of the packet shown in [Figure 3-3](#) and [Figure 3-4](#), the total length is 60 bytes. The maximum size of an IP packet is 65,535 bytes (2 bytes yields a range of 0–65,535), but most of the traffic seen on communication networks is made up of smaller packets. RFC 791 focuses on 576-byte packets (512-byte payload plus 64 bytes of header) and while the applications used on networks have changed quite a bit, a great deal of traffic still consists of datagrams of this size or smaller.

Identification

Every IP packet receives an identification value to aid in reassembly of packets. Packets too large for the network must be segmented into smaller chunks for transmission. For example, the maximum transmission unit (MTU) for an Ethernet network is 1500 bytes. Thus, while an IP packet can extend to 65,535 bytes, the packets must be broken up in order to fit into Ethernet frames. The ID field de-

```
Internet Protocol, Src: 192.168.16.253 (192.168.16.253), Dst: 192.168.16.2 (192.168.16.2)
  Version: 4
  Header Length: 20 bytes
  Differentiated Services Field: 0x20 (DSCP 0x08: Class Selector 1; ECN: 0x00)
  Total Length: 100
  Identification: 0x0087 (135)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 255
  Protocol: ICMP (0x01)
  Header checksum: 0x18a2 [correct]
  Source: 192.168.16.253 (192.168.16.253)
  Destination: 192.168.16.2 (192.168.16.2)
Internet Control Message Protocol
```

Figure 3-4. IP header showing DSCP values

scribes which fragments belong to the same packet. The value is chosen at random, but since it is a two-byte field, the values will be reused. However, in any particular conversation, the packets will typically have sequential IDs.

Flags

This small, three-bit field describes how the packet fragmentation is to be handled. Table 3-2 includes the possible values.

Table 3-2. Flag values

Bit value	Function
Bit 0	Reserved (always 0)
Bit 1	If 0, the packet may be fragmented; if 1, do not fragment
Bit 2	If 0, this is the last fragment; if 1, there are more fragments to come

With small packets, the value of this field will be 000. If a packet is fragmented, the fragments will vary the flags depending on the order. A node may specify that a packet is not to be fragmented. In this case, should the packet exceed the MTU for the network, there is a very good chance it will be dropped.

Fragment Offset

This 13-bit field is used in conjunction with the identification field (or fragment ID). Once a packet has been broken up, each part is given the same ID. However, when the packets are collected together, some method must be used to determine their proper order. The fragment offset provides the value (in bytes) of a particular fragment's position. An example of this fragmentation is shown in Figure 3-5. Pinging the address 10.1.1.253 with a payload slightly larger than 1500 bytes results in the conversation shown in Figure 3-6.

Figure 3-6 is an ICMP echo request exchange between the two endpoints. Notice that two packets are sent in one direction before the corresponding response is received. The first two packets are then decoded. By examining the IP header in Figure 3-5, you can see that the two packets have the same ID number (7526), but the fragment offsets

```
[-] Ethernet II, Src: Intel_c8:ad:30 (00:0c:f1:c8:ad:30), Dst: Cisco_b5:05:40 (00:07:50:b5:05:40)
[-] Internet Protocol, Src: 10.1.1.1 (10.1.1.1), Dst: 10.1.1.253 (10.1.1.253)
    Version: 4
    Header length: 20 bytes
    [-] Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 1500
    Identification: 0xd66 (7526)
    [-] Flags: 0x02 (More Fragments)
    Fragment offset: 0
    Time to live: 128
    Protocol: ICMP (0x01)
    [-] Header checksum: 0xe0bb [correct]
    Source: 10.1.1.1 (10.1.1.1)
    Destination: 10.1.1.253 (10.1.1.253)
    Reassembled IP in frame: 17
[-] Data (1480 bytes)
[-] Ethernet II, Src: Intel_c8:ad:30 (00:0c:f1:c8:ad:30), Dst: Cisco_b5:05:40 (00:07:50:b5:05:40)
[-] Internet Protocol, Src: 10.1.1.1 (10.1.1.1), Dst: 10.1.1.253 (10.1.1.253)
    Version: 4
    Header length: 20 bytes
    [-] Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 548
    Identification: 0xd66 (7526)
    [-] Flags: 0x00
    Fragment offset: 1480
    Time to live: 128
    Protocol: ICMP (0x01)
    [-] Header checksum: 0x03bb [correct]
    Source: 10.1.1.1 (10.1.1.1)
    Destination: 10.1.1.253 (10.1.1.253)
    [-] [IP Fragments (2008 bytes): #16(1480), #17(528)]
[-] Internet Control Message Protocol
```

Figure 3-5. Sequential fragmented packets

16	18.673969	10.1.1.1	10.1.1.253	IP	Fragmented IP protocol
17	18.673990	10.1.1.1	10.1.1.253	ICMP	Echo (ping) request
18	18.677451	10.1.1.253	10.1.1.1	IP	Fragmented IP protocol
19	18.677766	10.1.1.253	10.1.1.1	ICMP	Echo (ping) reply

Figure 3-6. Fragmentation conversation

are different. This is because they are two parts of the same message. The first packet has an offset of 0 because it is the start of the packet. The second has an offset of 1480. The Ethernet MTU is 1500. Subtracting 20 bytes for the IP header in the second packet fixes the offset at 1480. The offset of this first packet must be a multiple of eight.

Examining the flags field, the first (packet 16) has a value of 0x02 or more fragments, while the second (packet 17) has a value of 0x00, indicating that there are no further packets. The value 0x02 may be a bit confusing, but recall that this is a three-bit field. Since hexadecimal numbers are four bits long, Wireshark is simply borrowing the neighboring bit, which is 0.

Time to live

At Layer 3, to provide some protection from routing loops and to remove continuously circulating datagrams, we use the time to live (TTL) value. This is the length of time or the number of hops that this packet is permitted to make on the network. Each router decrements this field by 1 and once the value reaches 0, the packet must not be forwarded. RFC 791 actually refers to this as an actual lifetime in seconds, with each router taking approximately 1 second to process the packet. However, since routers process packets very quickly and no router will decrement the field by less than 1 (or more than 1), this value really falls to the hop count.

Protocol

This eight-bit field provides an indication as to what is being carried by the IP packet. This is necessary for the next process to correctly parse the subsequent header information. RFC 790 contains most of the assigned numbers used in networks today, including the values used in the protocol field. However, the most common values seen will be hexadecimal 0x01 (1—ICMP), 0x11 (17—UDP) and 0x06 (6—TCP). This value is shown in [Figure 3-3](#) and indicates that ICMP is encapsulated.

Header checksum

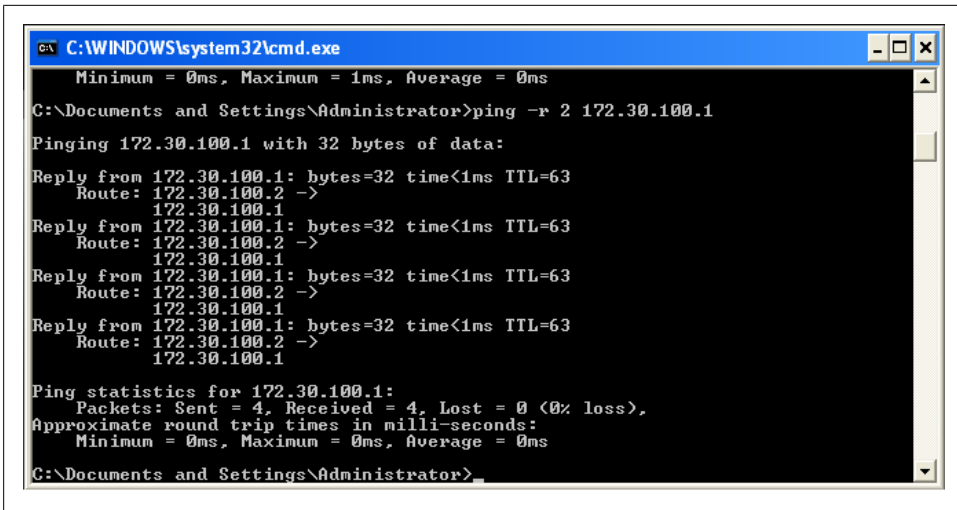
This is a 16-bit one's complement of the one's complement sum. This is applied to the 16-bit words in the header. The packet capture in [Figure 3-3](#) returns the value 0x949d. Refer to [Chapter 6](#) for an example of the one's complement addition.

Source and destination IP addresses

The last two fields normally used are the IP addresses of the nodes involved in the transmission. In this case, the four-byte source is 192.168.15.103 and the destination is 192.168.15.1. Looking at the hexadecimal, the values c0a80f67 and c0a80f01 can be seen as the last part of the header. The “Addressing” section provides greater insight into the various IP addresses seen in these fields.

Options

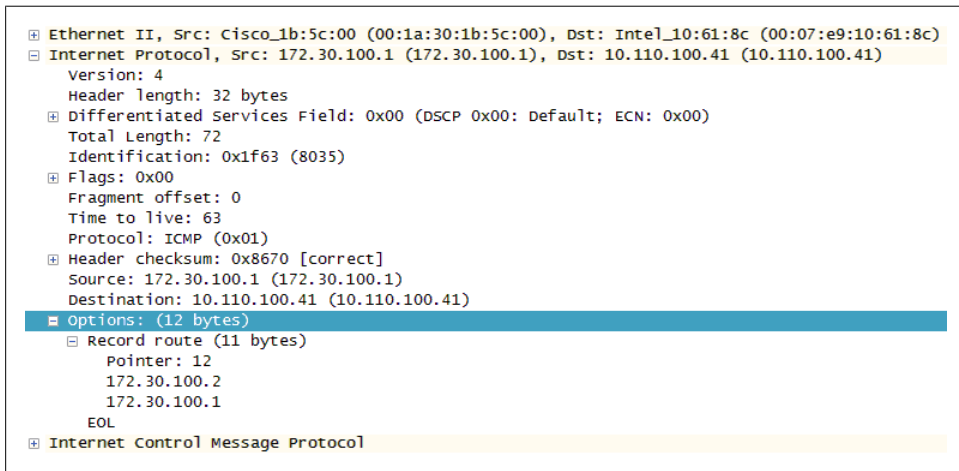
In comparing [Figure 3-2](#) and [Figure 3-3](#), the options field is absent from the live packet. This is normal for standard network transmissions. RFC 791 specifies that while the options field must be supported, it may or may not appear in packets. This is a variable-length field consisting of the option type, option parameters, and any additional data required. Options can be used in the case where a particular feature or test is desired. For example, if an administrator wishes to use a particular pathway between two hosts, the specific routers to be followed can be specified in the IP packet. Additionally, there may be some security requirements for traffic that can be described in the options field. Options may also be configured to be part of any fragments that are created. In [Figure 3-7](#), ping was used to recover the route taken by packets for the first couple of hops. Notice that the number of hops desired can vary and so the field size also varies. [Figure 3-7](#) shows the result of running the command `ping -r 2 172.30.100.1`, which asks that the first two hops encountered be recorded.



```
C:\WINDOWS\system32\cmd.exe
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\Documents and Settings\Administrator>ping -r 2 172.30.100.1
Pinging 172.30.100.1 with 32 bytes of data:
Reply from 172.30.100.1: bytes=32 time<1ms TTL=63
Route: 172.30.100.2 ->
172.30.100.1
Reply from 172.30.100.1: bytes=32 time<1ms TTL=63
Route: 172.30.100.2 ->
172.30.100.1
Reply from 172.30.100.1: bytes=32 time<1ms TTL=63
Route: 172.30.100.2 ->
172.30.100.1
Reply from 172.30.100.1: bytes=32 time<1ms TTL=63
Route: 172.30.100.2 ->
172.30.100.1
Ping statistics for 172.30.100.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Documents and Settings\Administrator>
```

Figure 3-7. Record route output

The accompanying packet capture shown in [Figure 3-8](#) has been expanded to show the options field. This packet comes from the destination, meaning that the source host requested that the information be included in the response. In addition, the header length, normally 20 bytes, is now 32. The extra twelve bytes include the option type field (the record route has a value of 7), length field (1 byte with a value of 11), a pointer indicating the next route data entry to be processed, and the two addresses returned (8 bytes). The final byte is the end flag.



```
Ethernet II, Src: Cisco_1b:5c:00 (00:1a:30:1b:5c:00), Dst: Intel_10:61:8c (00:07:e9:10:61:8c)
Internet Protocol, Src: 172.30.100.1 (172.30.100.1), Dst: 10.110.100.41 (10.110.100.41)
  Version: 4
  Header length: 32 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  Total Length: 72
  Identification: 0x1f63 (8035)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 63
  Protocol: ICMP (0x01)
  Header checksum: 0x8670 [correct]
  Source: 172.30.100.1 (172.30.100.1)
  Destination: 10.110.100.41 (10.110.100.41)
  Options: (12 bytes)
    Record route (11 bytes)
      Pointer: 12
      172.30.100.2
      172.30.100.1
      EOL
  Internet Control Message Protocol
```

Figure 3-8. IP header with options

Addressing

IP addresses are written in what is called *dotted quad* four-byte addressing. This simply means that there are four numbers in every IPv4 address and that these four values are each one byte in length and separated by a decimal. From [Figure 3-3](#), this is 192.168.15.103. Because each number is a single byte in length, the range for each is 0-255. The corresponding binary and hexadecimal ranges are 00000000-11111111 and 00-ff, respectively. While it is not obvious, each IP address has a network portion and a host portion that is determined by the mask. This will be covered in greater detail later on.

IP addresses are also grouped into classes. These classes vary in the size of the network and the number of hosts in each. [Table 3-3](#) provides the ranges for each class.

Table 3-3. IP classes

Class	Range of the first octet	Value of the first bits in binary	Number of possible networks	Number of possible hosts per network
A	0–127	0	128	16,777,216
B	128–191	10	16,364	65,636
C	192–223	110	2,097,152	256
D	224–239	1110	NA	NA
E	240–255	1111	NA	NA

For example, RIT has a network address of 129.21.0.0. This means all of the network hosts within RIT will begin with the same network ID, 129.21. In addition, based on the information in [Table 3-3](#), there are 65,536 possible hosts within the RIT network and that there are fewer than 17,000 networks of this size or larger. Examining the binary for the first octet (129) returns 10000001, which corresponds to the binary pattern for a class B network.

Each class of address has a mask associated with it. The purpose of the mask is to determine the network for a particular address. [Table 3-4](#) provides these values.

Table 3-4. IP class masks

Class	Mask	Network addressing	Host addressing
A	255.0.0.0	7 bits	24 bits
B	255.255.0.0	14 bits	16 bits
C	255.255.255.0	21 bits	8 bits

The method used to determine the network is called *ANDing*. By taking the IP address and ANDing it with the mask associated with the network, you can calculate the network ID. The logical AND operation takes two inputs and compares them. Anytime a

value is ANDed with a 0, the result is 0. Converting an RIT address to binary results in the following:

```
129.21.199.200    10000001.00010101.11000111.11001000
255.255.0.0        11111111.11111111.00000000.00000000
After ANDing       10000001.00010101.00000000.00000000
```

The reason the network ID must be calculated is that forwarding decisions, for hosts and routers, are made based on the network ID. Every network device performs these calculations. Taken another way, the binary 1s indicate the network portion and the 0s indicate the host portion. This process is covered in greater detail in the [Chapter 7](#). The network and host portions are also referred to as the *prefix* and *suffix*. It is important to remember that the prefix and suffix are determined by the mask.

In addition to the classes of address, there are many special IP addresses that are reserved. These are described in [Table 3-5](#).

Table 3-5. Reserved IP addresses

Binary prefix	Binary suffix	Type and example	Purpose
All zeros	All zeros	Identifies the host	Used for DHCP to obtain a working IP address
00000...	00000...	0.0.0.0	
IP address (network portion)	All zeros	Network ID	Specifies a particular network
	00000...	129.21.0.0	
IP address (network portion)	All ones	Directed broadcast	Broadcast packet to a particular network
	11111...	129.21.255.255	
All ones	All ones	Limited broadcast	Broadcast packet to the current network
11111...	11111...		
127	Anything	Loopback	Used for testing or identifying the localhost
		127.0.0.1	

There are a couple of other special addresses that must be included in this discussion. Anyone running a network in their own home or small office probably recognizes the address 192.168.1.1. This address is part of a collection of addresses (one for each class) specified for use with network address translation, or NAT. While NAT is a subject for another chapter, the basic idea is that private addressing or addresses not present on the public Internet are used whenever NAT is deployed. [Table 3-6](#) provides the complete list.

Table 3-6. Private IP address ranges

Class	Address range
A	10.0.0.0–10.255.255.255
B	172.16.0.0–172.31.255.255
C	192.168.0.0–192.168.255.255

For more information about NAT, a good place to start is RFC 1918 for the addressing and RFC 1631 for NAT structure and operation.

There is one other address that is quite common, but it is often confused with what might be default settings or allocated to Microsoft. The address range in question is 169.254.0.0–169.254.255.255. The set of addresses is set aside for the IETF Zero Configuration standard. This standard describes operation and requirements for a network running without infrastructure support. This means that in the absence of DHCP or DNS systems, the network will still be operational (to a certain extent), because the nodes will still have IP addresses automatically assigned. Figure 3-9 depicts an example of ZeroConf addressing.

```
Ethernet adapter Local Area Connection 1:

Connection-specific DNS Suffix . . : 
Autoconfiguration IP Address. . . : 169.254.200.104
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . :
```

Figure 3-9. IP ZeroConf example

Sample Host Configuration

When operating on a network, four numbers are typically required: IP address, mask, default gateway (router), and DNS. These values are mostly commonly acquired via a DHCP server. The output shown in Figure 3-10 is a sample configuration from a Windows host. To do this yourself, open a command prompt (type **cmd** in the Run box and press Enter) and issue the command **ipconfig /all**.

If any one of these numbers is missing, some portion of network communications will be hobbled. For example, without the DNS entry, names such as www.rit.edu or www.google.com couldn't be found. The IP addresses, if known, would have to be typed into the web browser. Without the default gateway entry, there would be no way to communicate with nodes outside of the current network, because the host would not know the address of the router used to send packets externally.

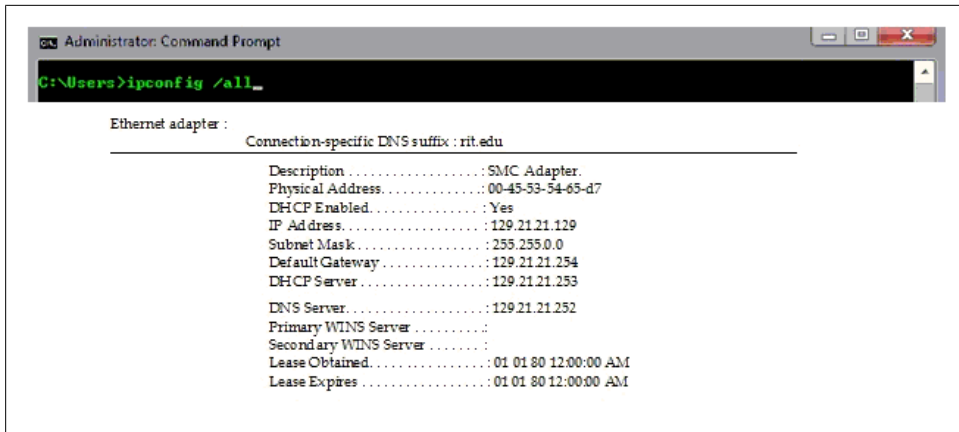


Figure 3-10. Sample host configuration

Operation

IP is one of the protocols that can really make you wonder at the success of the Internet, at least in terms of its operation. IP packets have no knowledge of the pathway to the endpoint, little in the way of error control, and nothing to ensure their delivery. To quote from RFC 791:

The internet protocol treats each internet datagram as an independent entity unrelated to any other internet datagram. There are no connections or logical circuits (virtual or otherwise).

Yet, as packets are cast into the void of interconnected networks, somehow they reach their destinations. The key is that all routers and hosts follow the same set of rules. The amazing part is that these devices are run by humans who typically *do not* follow the rules.

Once a host initiates a communication via an application, some chunk of data, such as an email message, document or a Facebook post, is inserted into a TCP or UDP datagram. This datagram is in turn placed inside of an IP packet. The result of this process can be seen in any of the TCP-based captures in this book or in the encapsulation examples. The IP packet is created after examining the host routing table. This step is critical because the sending host must determine whether or not the packet is to be sent to a node on the local network or one that is outside, thus changing the content of the IP header. This particular process will be covered in a later chapter.

If the packet belongs to the current network and the appropriate header is constructed, no more has to be done with the IP header except when it is received by the destination. Packets destined for the current network get to the destination via a local forwarding based on MAC addresses and ARP. Recall that ARP maps IP addresses to MAC addresses. When received, the header checksum will be calculated and the packet will be delivered to the proper application.

If the packet is to be sent off of the current network, the packet must be routed, so it is sent to the host's default gateway. The default gateway is actually a router interface that is reachable by the host. A simple example is a home gateway such as a Linksys. Linksys provides (among other things) routing services for the internal hosts, meaning that computers on the home network send packets to the router for forwarding to either the next router in line (the next hop) or the final destination. Like hosts, routers participate in the ARP protocol, understand ICMP, and have a routing table. However, the router's routing table has a slightly different use than the host's routing table—it is filled with information about other networks rather than information about the current network.

In their basic form and operation, routers process the IP header and simply forward the packets on—they do not modify the content of either the header or the IP packets. With advanced operations, certainly there are times when this is not the case.

Digging a Little Deeper...What Addressing is Sufficient?

There are many different types of addressing used in a modern communication network. These addresses may exist at different layers of our networking models, be encoded in hardware, or be easily manipulated in software. Addressing can even be dynamically generated as needed, as with TCP/UDP ports. But what addressing is actually required? Why does a typical communication use at least three different addresses?

To understand how we arrived at the current state of affairs, it might be appropriate to imagine scenarios in which networks try to get by with less. For example, eliminating either the MAC (hardware) or the IP (software) address would simplify things a bit. Headers and, therefore, packets would be smaller, improving network throughput. But is one address enough? Giving each host a single unique address might mean that all hosts exist on the same network. This also means that broadcast and multicast messages have the potential to propagate much farther. Node or service discovery could be much more time consuming as the hierarchy imposed by IP would now be missing. It turns out that in our current architecture, the additional addresses are required in order to provide some indication of location, or at least forwarding. MAC addresses do not contain any location information and IP addresses do not have a mechanism for locating individual hosts. IP also insulates the packet from the possible differences in Layer-2 technologies, providing the ability to move easily between Ethernet, Token, or 802.11 networks.

But not everyone believes that this is the way it must be. The telephone system provides a single address for every telephone, and the location information is stored within the telephone number itself. The telephone network has built in redundancy for many of its pathways and has successfully kept people connected for decades, albeit over very low bandwidth lines. New ideas have begun to surface about Internet topologies following this sort of approach. IP addresses provide routing information, but a specific address may physically exist. By creating a system in which geographic regions have

assigned addresses, the routing becomes very similar to the telephone system. Forwarding may be much faster and routing tables simpler to parse. If the test data works out, IPv6 may have a competitor or two. You can find some light reading on the subject at NSF NeTS FIND Initiative (www.nets-find.net/) and the European Future Internet Initiative (<http://initiative.future-internet.eu/>).

Security Warning

There are a number of security concerns with basic IP headers and operation. IP addresses are logical addresses configured within the software of the operating system. Between this and the ability to capture traffic, it is extraordinarily simple to pick a target IP address and change yours to match the target. This is called *spoofing an address*. The purpose of spoofing is to pass your system off as a valid node for the purpose of bypassing security or receiving packets destined for someone else.

The IP header is almost always clear text. Even when operating in a system deploying virtual private networks (VPNs) such as those based on IPSec, SSL, or the elderly PPTP, the IP header is clearly visible. This gives attackers the ability to read portions of any conversation in order to determine the probable locations of servers, hosts, and network devices. By understanding the network device types, attacks can be tailored for the appropriate target.

The basic operation of IP also lends itself to attack. Since IP does not have any inherent error or security checks, packets are forwarded according to the rules. This is the case for all traffic; thus, there is often no distinction between good traffic and bad traffic. A router will forward packets regardless of where they came from or where they are going to, unless it is specifically configured not to do so. But that is another subject entirely.

Organizations for Assigning Addresses and Names

The most notable and oldest group associated with the Internet as we know it is the Internet Assigned Numbers Authority, or IANA. The following information is taken from the IANA website at www.iana.org:

- IANA is responsible for the operation and maintenance of a number of key aspects of the DNS, including the root zone, and the .int and .arpa domains.
- IANA is responsible for global coordination of the Internet Protocol addressing systems, as well as the Autonomous System Numbers used for routing Internet traffic.
- IANA is responsible for maintaining many of the codes and numbers contained in a variety of Internet protocols, enumerated below. We provide this service in coordination with the Internet Engineering Task Force (IETF).

IANA is, in turn, operated by the Internet Corporation for Assigned Names and Numbers (ICANN), which is a nonprofit partnership that organizes the public IP address space for the entire world. It is the central repository for all of the used and unused addresses. Formed in 1998, ICANN also has the responsibility for organizing all of the names and conventions used in the domain name servers. ICANN is the forum in which stakeholders have a voice into the administration process and policy building.

Interaction between ICANN and the other organizations can be a bit confusing, as there are several components. In addition to ICANN, there are five nonprofit Regional Internet Registries (RIRs) that manage the numeric resources and help develop policy:

- AfriNIC—Africa
- APNIC—Asia/Pacific
- ARIN—North America
- LACNIC—Latin America/Caribbean
- RIPE NCC—Europe, Middle East, Central Asia

The Number Resource Organization (NRO) was created in 2005 and brings these five groups together for joint projects and for coordinating the number allocation/protection and policy work.

DNS provides the mapping between human-readable names (such as those used in web pages) and IP addresses, which are horrible to remember. IANA/ICANN manages significant parts of DNS. DNS is a very complex collection of servers, each providing answers related to the resolution of IP addresses. There are 13 main or root servers, each of which has a copy of the index for what amounts to the Internet phone book. Further down the chain are the servers running the top-level domains, such as those ending with .com or .edu. A registry is in charge of each top-level domain, and domain names such as `whatshouldicallmywebsite.com` are purchased from registrars. In practice, someone wishing to create his own domain might purchase the domain name from any number of companies (registrars), which will handle registering the new domain with the DNS. These companies are, in turn, charged by the registry.

You can find a wealth of information regarding DNS at either IANA (www.iana.org/) or InterNIC (www.internic.net/). Run by ICANN, InterNIC is a website that provides searching for accredited registrars, operational information, and filing complaints.

Standards and RFCs

RFC 790: Assigned Numbers, J. Postel

RFC 791: Internet Protocol DARPA Internet Program Protocol Specification

RFC 796: Address Mapping, J. Postel

RFC 1122: Requirements for Internet Hosts—Communication Layers, IETF

Summary

While IPv6 is beginning to see increased deployment, IPv4 continues to be the basis for contemporary networks. This chapter details the fields and addressing used within IP and provides several examples in which the fields have been modified. However, it is not enough to simply understand the structure of the packets. IP is integrated into network operations and forwarding of these packets throughout the global Internet. For this reason, this chapter also introduces some of the operational and security considerations that are part of any network. There are many agencies that work together to manage the addressing and naming concerns associated with IP. This chapter provides the introduction and structure for their interaction.

Review Questions

1. What is the length of a typical IP header?
2. IP ToS is commonly used to provide QoS to IP packets. True or false?
3. IP packets belonging to the same “conversation” are generally routed together. True or false?
4. The protocol field in the IP header uses the same values as the Ethernet control field. True or false?
5. Each class C network can contains 256 possible addresses. True or false?
6. An IP address consisting of a network ID followed by all 1s is used for what type of message?
7. What is the address space specified for use with the Zero Configuration Protocol?
8. What are the four dotted-quad numbers required for hosts operating on a network?
9. IP headers are almost always transmitted in the clear regardless of the security employed. True or false?
10. What organization is responsible for managing critical DNS components such as the root servers?

Review Answers

1. 20 bytes
2. False
3. False
4. False
5. True
6. This is a directed broadcast at a particular network
7. 169.254.0.0–169.254.255.255

8. IP address, DNS, mask, and default gateway
9. True
10. IANA

Lab Exercises

Activity 1—Determining IP Address Components

Materials: A computer with an active Internet connection

1. Open up a command shell and run **ipconfig /all**.
2. Identify the IP mask, default gateway, and DNS.
3. Where did these numbers come from?
4. Calculate the network ID for your computer.

Activity 2—IP Packet Capture

Materials: Wireshark and an active connection

1. Start Wireshark.
2. In the command shell, ping your default gateway.
3. Within Wireshark, examine the packets that result from this command.
4. Identify the individual fields in the IP packets.
5. What are the values for each field? Does the protocol ID match the payload?
6. Were other packets captured while you were doing this activity? If so, how are they different from the ICMP traffic?

Activity 3—Header Checksum

Materials: Wireshark and an active connection

1. If you haven't already, open Wireshark and capture an IP packet.
2. Find the value of the header checksum.
3. Take a look at the hexadecimal values in the bottom frame within Wireshark.
4. Using these values and the one's complement, calculate your own header checksum.
5. Did your value match that seen in Wireshark?

Activity 4—Fragmentation

Materials: Wireshark and a Windows active connection

1. Within the command shell, run the `ping` command without an argument. This will provide you with the options available with `ping`.
2. Using the `-l` option, change the length of your next ping to the gateway to 2000 bytes.
3. While capturing, issue the `ping` command.
4. Examine the packets generated and match the packets with the same identification value. Calculate the fragment offsets.
5. Do the fragment offsets match your calculation?
6. Would `ping -l 1000` accomplish the same goal of following the fragmentation? Why or why not?

Activity 5—Special Address Capture

Materials: Wireshark and an active connection

1. Start a packet capture on your local segment or your home network.
2. [Table 3-5](#) identifies several special (reserved) addresses used on IP-based networks. Can you capture packets that use each one of these special addresses? Hint: what does a Windows host send out when trying to bring up a network connection?
3. Once you have captured the packets, can you determine the conversation or activity that generated the packets?
4. Convert these addresses to binary. Do they match the values shown in [Table 3-5](#)?

Address Resolution Protocol

“The world is a jungle in general, and the networking game contributes many animals.”

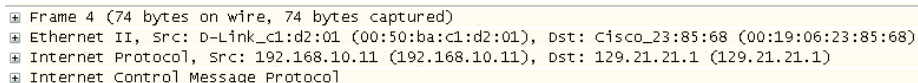
—RFC 826

The operation of an IPv4 network requires the use of several different kinds of addresses at different layers of the networking model, but also the resolution of these addresses to one another. This chapter describes the address resolution process, gives real-world examples of the messaging used, and provides insight into potential security risks associated with its use.

The Problem

A vast majority of IP packet-based data transmission begins and ends on a LAN. This is true regardless of whether the IP packet is going to a neighbor on the same LAN or to the other side of the world. [Chapter 3](#) describes how IP packets are encapsulated in LAN frames that use Layer-2 MAC addressing for both the source and the destination nodes. The source MAC address is easy to determine. The problem is the determination of the destination MAC address.

With Ethernet as a LAN infrastructure, a frame is constructed using the sender’s own address as the source at Layer 2 and IP address as the source at Layer 3. The destination IP address (or at least the name) is usually known, leaving only the determination of the destination MAC address. [Figure 4-1](#) is a packet-capture review of these addresses shown in an encapsulated ICMP message.



```
⊞ Frame 4 (74 bytes on wire, 74 bytes captured)
⊞ Ethernet II, Src: D-Link_c1:d2:01 (00:50:ba:c1:d2:01), Dst: Cisco_23:85:68 (00:19:06:23:85:68)
⊞ Internet Protocol, Src: 192.168.10.11 (192.168.10.11), Dst: 129.21.21.1 (129.21.21.1)
⊞ Internet Control Message Protocol
```

Figure 4-1. Addressing layers

This is an example of a transmitted frame where the source and destination MAC addresses have been previously determined.

Techniques

Methods for the determination of the destination MAC address include closed-form computation, table lookup, and message exchange. Some of these options are listed in RFC 894, which describes Ethernet encapsulation.

Closed-form computation calculates the unknown MAC address from the known IP address. The sending node fills in the destination MAC in the Ethernet frame from the calculated value. This method is very quick and does not require outside resources or communication. It also allows reasonably tight control over the address space. However, it does require configurable MAC addresses and some level of management, as the addresses must all be assigned to the various hosts.

Table lookup provides each host with a list of MAC addresses and the corresponding IP addresses. This is also very fast, as the sender needs to consult the table only before building the Ethernet frame. Replacing even a single network card mandates that all tables be updated.

These methods have an advantage in terms of speed, but impose heavy management oversight. Individual host addresses must be configured and the hosts will have to be notified of any changes. For this reason, networks today (with the exception of some WAN connections) rely on the distributed approach or message exchange using the address resolution protocol, or ARP. Message exchange does add extra traffic to the network and is slower than the other methods. However, this message exchange technique is totally automated and therefore very attractive.

Protocol Description

ARP is built into the IP configuration of every node. This means that developers at Microsoft, Sun, Google, and in the open source community develop their operating systems for operation on an IPv4 network, and code for ARP is included.

The nice thing about ARP is that for basic operation, there are only two messages defined: an ARP request and an ARP reply. When a host must find the MAC address of the destination, it will send out an ARP request. This is after the node consults its ARP table and determines that the address is in fact unknown.

Upon receipt of the ARP request message, the destination will send back an ARP reply. Basically, the ARP request asks, “Can I have your MAC address?” and the reply says, “Sure, here it is.” Hosts never say no if they can help it. [Figure 4-2](#) shows this message exchange.

No. .	Time	Source	Destination	Protocol	Info
299	426.491695	Ibm_43:49:97	Broadcast	ARP	who has 192.168.1.254? Tell 192.168.1.1
300	426.492283	Cisco_35:1a:d0	Ibm_43:49:97	ARP	192.168.1.254 is at 00:19:55:35:1a:d0

Figure 4-2. ARP exchange

Wireshark interprets this conversation as a question followed by an answer. In the first line, one node (192.168.1.1) is asking about 192.168.1.254 and in the response, 192.168.1.254 gives its location as 00:19:55:35:1a:d0, which is a MAC address.

Structure

The construction of the two ARP message types is shown in [Figure 4-3](#) and later in [Figure 4-5](#). Consider the details of the two message types, paying special attention to the addressing used in both the frame and the ARP fields.

* Frame 299 (42 bytes on wire, 42 bytes captured)	
⊞ Ethernet II, Src: Ibm_43:49:97 (00:11:25:43:49:97), Dst: Broadcast (ff:ff:ff:ff:ff:ff)	
⊞ Address Resolution Protocol (request)	
Hardware type: Ethernet (0x0001)	
Protocol type: IP (0x0800)	
Hardware size: 6	
Protocol size: 4	
opcode: request (0x0001)	
[Is gratuitous: False]	
Sender MAC address: Ibm_43:49:97 (00:11:25:43:49:97)	
Sender IP address: 192.168.1.1 (192.168.1.1)	
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)	
Target IP address: 192.168.1.254 (192.168.1.254)	

Figure 4-3. ARP request

The ARP message format is straightforward and consists of the following fields:

Hardware type

The type of MAC address being sought

Protocol type

The Layer-3 protocol in use

Hardware size

The length of the MAC address

Protocol size

The length of the protocol address

OpCode

The type of ARP message

Sender MAC address

The MAC address of the machine sending the request

Sender IP address

The protocol address of the machine sending the ARP request

Target MAC address

The MAC address being sought

Target IP address

The protocol address of the destination

The terms *hardware address* and *protocol address* are used as general descriptions, but operationally these will almost always be Ethernet six-byte hardware addresses and IP four-byte addresses. The OpCode will be either a request or a reply.

Addressing in the ARP Request

Three of the four addresses in an ARP request packet are known: the source and destination IP and the source MAC. This leaves only the destination MAC unknown. The request packet is completed by padding the unknown address field with 0s. The reply will fill in the correct value.

Line 2 of [Figure 4-3](#) shows that the Ethernet frame source MAC is the machine sending the request, but the frame destination MAC is a broadcast address. This ensures all nodes pay attention, thereby guaranteeing that if the destination is connected and powered up, it will respond.

While there are IP or protocol addresses used in this message, it does not actually have an IP header. The IP addresses seen are simply part of the ARP header. This means that ARP messages are not routable and that routers will not pass ARP traffic on to another network. Consequently, the MAC address of a node not on the source node's LAN cannot be determined.

It also means that the Ethertype in an Ethernet frame carrying an ARP message is different than standard data traffic. This difference is shown in [Figure 4-4](#).

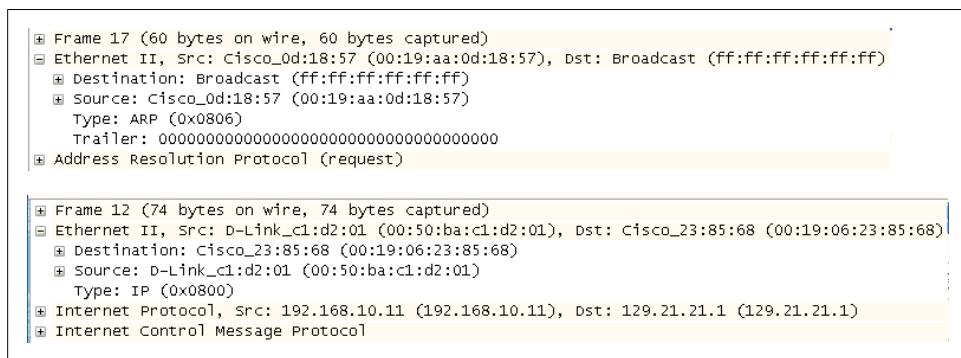


Figure 4-4. Ethertypes

Frame 17 in [Figure 4-4](#) has a hexadecimal type value of 0x0806 and lacks an IP header. Frame 12 has a hexadecimal type value of 0x0800 and does have an IP header. This difference can affect packet filtering or the firewall rules in place depending on the information sought.

Addressing in the ARP Reply

The ARP reply depicted in [Figure 4-5](#) is the response to the request sent in [Figure 4-3](#), with the missing MAC address filled in. The reply is heading in the opposite direction. Thus, the sender and target addresses are now reversed. The code field has also changed to a reply.

```
⊞ Frame 300 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: Cisco_35:1a:d0 (00:19:55:35:1a:d0), Dst: Ibm_43:49:97 (00:11:25:43:49:97)
⊞ Address Resolution Protocol (reply)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (0x0002)
    [Is gratuitous: False]
    Sender MAC address: Cisco_35:1a:d0 (00:19:55:35:1a:d0)
    Sender IP address: 192.168.1.254 (192.168.1.254)
    Target MAC address: Ibm_43:49:97 (00:11:25:43:49:97)
    Target IP address: 192.168.1.1 (192.168.1.1)
```

Figure 4-5. ARP reply

In the Ethernet frame itself, instead of a broadcast destination, *both* MAC addresses are now unicast. The reply goes directly to the original sender from the target and other nodes will ignore the frame.

Upon receiving this message, the original source host will do two things:

1. Build the data frame using the newly determined MAC address information in the destination field.
2. Populate the local ARP table.

Step 1 satisfies the original goal of sending a message to the destination. The second step populates an ARP table to save time during the next transmission to the same destination. The ARP table is a collection of recently learned MAC addresses and corresponding IP addresses. The next time the host must transmit a frame, it will search for the address in local memory and use the addresses found there instead of issuing more ARP requests. An example of an ARP table is shown in [Figure 4-6](#).

This output was obtained on a Windows machine with the command `arp -a` issued from the command shell. Notice the two types of entries—static and dynamic. The normal entry will be a dynamic entry. Static entries are uncommon.

```
C:\>arp -a

Interface: 129.21.152.158 --- 0x10005
Internet Address      Physical Address      Type
11.11.11.11           23-34-45-56-67-78    static
129.21.152.172        00-11-d8-d6-06-91    dynamic
129.21.152.254        00-00-0c-07-ac-01    dynamic
```

Figure 4-6. ARP table

The dynamic nature of these entries indicates that they are not permanent. Regardless of the underlying operating system, all nodes will age out ARP table entries in a matter of minutes. Windows, for example, removes these entries after approximately two minutes. If a node is to be addressed, but has been aged out of the ARP table, the ARP process must be repeated for that node.

The time that an ARP table entry should be allowed to live has been debated, as there are differing opinions as to the perfect time. If the value is too short, the hosts will be reARPing at an increased rate and generating more network traffic. If the time is too long, bad or erroneous information may stick around longer and prevent hosts from reaching the proper destination.

Operation

With an understanding of what takes place under the hood, two examples will help illustrate ARP packet formation for near and far destinations when ARP table information is nonexistent.

Example 1—Sender and Target on the Same LAN

A common troubleshooting technique is to ping a target IP as “proof of life.” Ping generates an ICMP echo request packet that is encapsulated in an IP packet, which, in turn, is encapsulated in an Ethernet frame, as shown in [Figure 4-7](#).

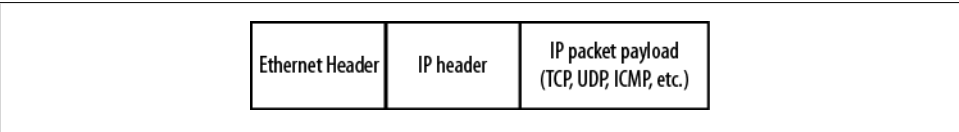


Figure 4-7. Basic frame encapsulation

Packet capture activity of the frame depicted in [Figure 4-7](#) is shown in [Figure 4-8](#).

1	0.000000	Micro-St_eb:78:84	Broadcast	ARP	who has 192.168.15.1? Tell 192.168.15.100
2	0.000324	Cisco-Li_7f:fb:9d	Micro-St_eb:78:84	ARP	192.168.15.1 is at 00:14:bf:7f:fb:9d
3	0.000347	192.168.15.100	192.168.15.1	ICMP	Echo (ping) request
4	0.000862	192.168.15.1	192.168.15.100	ICMP	Echo (ping) reply

Frame 3 (74 bytes on wire, 74 bytes captured)

Ethernet II, Src: Micro-St_eb:78:84 (00:0c:76:eb:78:84), Dst: Cisco-Li_7f:fb:9d (00:14:bf:7f:fb:9d)

Destination: Cisco-Li_7f:fb:9d (00:14:bf:7f:fb:9d)

Source: Micro-St_eb:78:84 (00:0c:76:eb:78:84)

Type: IP (0x0800)

Internet Protocol, Src: 192.168.15.100 (192.168.15.100), Dst: 192.168.15.1 (192.168.15.1)

Internet Control Message Protocol

Figure 4-8. ARP and ICMP on the same network

The MAC address requested in frame 1 is returned in frame 2. It is then used in frame 3 to build the Ethernet frame carrying the ping (ICMP echo), with Node A attempting to contact the router on its LAN (Figure 4-9). While the example here uses ping with the associated ICMP echo request/reply messages, the same ARP request and reply would have been required had the sender issued a Telnet, FTP, or HTTP request to the target.

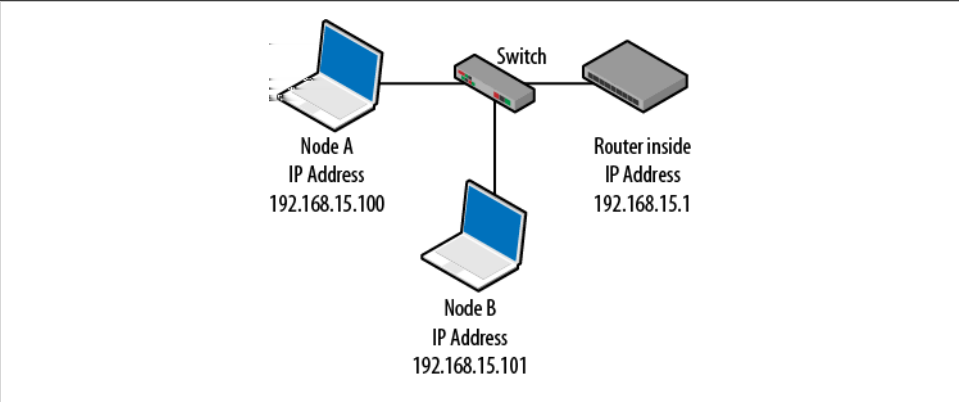


Figure 4-9. Single LAN topology

Example 2—Sender and Target on Separate LANs

As with our first example, when the sender and target are on separate LANs, the Ethernet frame’s destination MAC address must be determined. In this case, the destination node is on a remote LAN. Since Layer-2 MAC addressing is restricted to the local network, assistance is required from the designated default gateway that will route the frame to the destination network. Router ARP behavior is similar to that of hosts. They respond to ARP messages and have to locate locally connected nodes.

To accomplish this, the sending node determines the gateway’s MAC address and places it in the destination field, as shown in Figure 4-10. As before, frame 3 is expanded to show that in the ICMP echo request, the router MAC address is used.

1	0.000000	Micro-St_eb:78:84	Broadcast	ARP	who has 192.168.15.1? Tell 192.168.15.100
2	0.000479	Cisco-Li_7f:fb:9d	Micro-St_eb:78:84	ARP	192.168.15.1 is at 00:14:bf:7f:fb:9d
3	0.000503	192.168.15.100	129.21.3.17	ICMP	Echo (ping) request
4	0.036549	129.21.3.17	192.168.15.100	ICMP	Echo (ping) reply

+ Frame 3 (74 bytes on wire, 74 bytes captured)	
+ Ethernet II, Src: Micro-St_eb:78:84 (00:0c:76:eb:78:84), Dst: Cisco-Li_7f:fb:9d (00:14:bf:7f:fb:9d)	
+ Destination: Cisco-Li_7f:fb:9d (00:14:bf:7f:fb:9d)	
+ Source: Micro-St_eb:78:84 (00:0c:76:eb:78:84)	
+ Type: IP (0x0800)	
+ Internet Protocol, Src: 192.168.15.100 (192.168.15.100), Dst: 129.21.3.17 (129.21.3.17)	
+ Internet Control Message Protocol	

Figure 4-10. ARP and ICMP exchange for different networks

To summarize, the sender is attempting to determine the target MAC address, but the ICMP echo request is heading for a destination on another network. So the ICMP echo request uses the default gateway MAC address (00:14:bf:7f:fb:9d), but the IP address is for the distant node. Shown in [Figure 4-11](#), Node A is now trying to contact Node C.

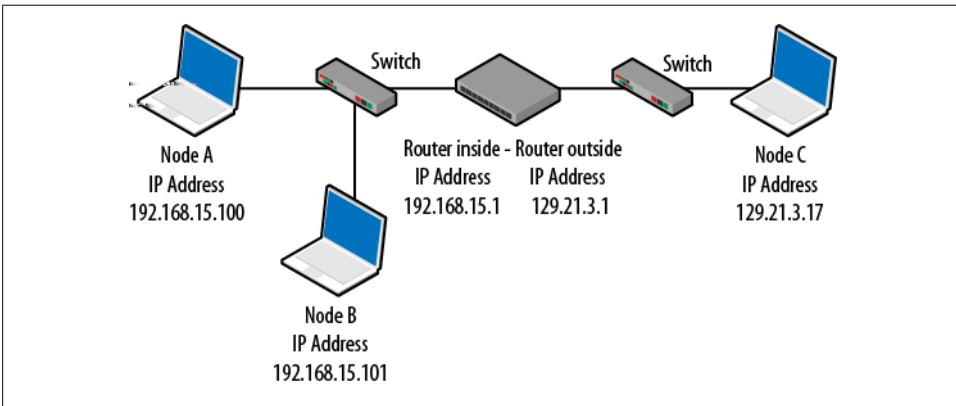


Figure 4-11. Two-network topology

The question to ask at this point is, “How did the original source node know that it had to replace the MAC address of the destination host with the MAC address of the router?” Hosts first process their own routing tables to determine if the host is on the same LAN. Then the ARP process takes over. The algorithm the hosts use is discussed in [Chapter 7](#).

Additional Operations

The standard operation of ARP is pretty simple: broadcast a message requesting the MAC address for a particular IP address and receive an answer. However, there are a couple of key “helper” tasks accomplished by ARP that either add a little security or improve performance of the network.

The Return ARP

The conversation shown in [Figure 4-12](#) illustrates another important facet of ARP—only the host originating the conversation (generating the ARP request) will place an entry for the destination host in its local ARP table. That is, other stations hearing the exchange, even if they are receiving the ARP request, will not add these stations to their own ARP tables. However, many hosts (especially routers) are aggressive when it comes to populating their tables and, upon hearing ARP traffic or being involved in ARP messages, will subsequently generate their own ARP requests to populate their tables.

1	0.000000	Micro-St_eb:78:84	Broadcast	ARP	who has 192.168.15.1? Tell 192.168.15.100
2	0.000479	Cisco-Li_7f:fb:9d	Micro-St_eb:78:84	ARP	192.168.15.1 is at 00:14:bf:7f:fb:9d
3	0.000503	192.168.15.100	129.21.3.17	ICMP	Echo (ping) request
4	0.035549	129.21.3.17	192.168.15.100	ICMP	Echo (ping) reply
5	0.993031	192.168.15.100	129.21.3.17	ICMP	Echo (ping) request
6	1.045927	129.21.3.17	192.168.15.100	ICMP	Echo (ping) reply
7	1.993023	192.168.15.100	129.21.3.17	ICMP	Echo (ping) request
8	2.030664	129.21.3.17	192.168.15.100	ICMP	Echo (ping) reply
9	2.992966	192.168.15.100	129.21.3.17	ICMP	Echo (ping) request
10	3.029049	129.21.3.17	192.168.15.100	ICMP	Echo (ping) reply
11	5.034065	Cisco-Li_7f:fb:9d	Micro-St_eb:78:84	ARP	who has 192.168.15.100? Tell 192.168.15.1
12	5.034091	Micro-St_eb:78:84	Cisco-Li_7f:fb:9d	ARP	192.168.15.100 is at 00:0c:76:eb:78:84

Figure 4-12. Return ARP exchange

The packet capture sequence shown in [Figure 4-12](#) shows the original host using ARP to determine its default gateway when attempting to send to an offsite host. After the conversation has been routed, the router (default gateway) issues its own ARP request for the original (sending) host. In this way, it populates its table with what it believes is a valid host address. This improves routing efficiency for future traffic forwarding.

Gratuitous ARP

When a host boots up, it either receives an IP address via DHCP or has one statically configured. But the host must make sure no other network node is using the same address. For this reason, network hosts will often ARP for themselves. If a device answers, the sender is alerted that another node is using the same IP address. [Figure 4-13](#) shows the target IP address and sender with gratuitous ARP.

⊞ Ethernet II, Src: Avaya_70:cf:66 (00:04:0d:70:cf:66), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Address Resolution Protocol (request/gratuitous ARP)
Hardware type: Ethernet (0x0001)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (0x0001)
[Is gratuitous: True]
Sender MAC address: Avaya_70:cf:66 (00:04:0d:70:cf:66)
Sender IP address: 192.168.16.2 (192.168.16.2)
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.16.2 (192.168.16.2)

Figure 4-13. Gratuitous ARP

Security Warning

The distributed approach to address resolution can be subject to attackers. Although hosts should only populate their tables with information they have requested, not all operating systems are programmed this way. Some older systems will allow unsolicited ARP traffic to fill a host's cache, accepting an ARP response even if it was not requested. This allows attackers to populate the ARP table with bogus data, resulting in hosts forwarding traffic based on erroneous information.

An attacker can also take advantage of a device's desire to populate its ARP table by providing an answer for every address on the network. In this way, it claims to have a valid MAC address for all hosts on the network, so hosts and routers on the network will believe that the attacker address is to be used for all destinations. The effect is that the valid network hosts send their traffic to the attacker, who then makes copies of the data and sends the traffic on to the correct destination.

This is called a *man-in-the-middle* attack because the attacker has placed herself between the proper source and destination and is effectively invisible. The technique of inserting bad data into unsuspecting host ARP tables is called *ARP poisoning*.

You can diagnose this type of attack by examining the ARP tables on the host machines and the routers, looking for multiple entries with identical MAC address. Security heuristics will also look for excessive ARP messages on the network. While these tables are easy to access, overworked network administrators do have to look, so this information is often missed.

IPv6

ARP is absent in IPv6. Rather, networks hosts use a series of messages called redirects, solicitations, and advertisements in a process called *neighbor discovery*. Instead of using an approach that requires hosts to discover MAC addresses when they are needed, IPv6 adopts a slightly different process. Neighbor solicitation and advertisement messages help discover information about the network before it is needed. These messages are multicast out to all IPv6 nodes. Examples of these packets are given in [Chapter 6](#).

Digging a Little Deeper

ARP, a distributed approach to address resolution and discovery, is not without problems. Consider the traffic generated in a 100-node network where each host must discover every address on the network. If nodes do not cache information as a result of a transmission from a neighbor, every node has the potential to send 99 messages. Adding another 99 messages for the corresponding replies brings the total to 198 for that single requesting node. For n nodes, each node will generate $2(n-1)$ messages or a total of $n * 2(n-1)$ packets or $2n(n-1)$.

Half of the $2n(n-1)$ messages, $n(n-1)$, are broadcast frames traveling throughout the entire Layer-2 network (wired and wireless) and all of them are necessary, but are considered overhead because they do not carry user data. It is unlikely that most of these frames will be generated at the same time, but there are times (for example, at the beginning and end of the work day) when a large number of network hosts will be transmitting concurrently. Complicating matters is the fact that ARP tables age out for nodes that are not routinely participating in message exchanges. Refreshing those tables further adds to network traffic.

Routers are burdened with the additional problem of resolving the addresses next hop routers. Thus, when a router receives a message to be sent to a distant host, it must first determine the MAC address of the neighboring router. At the other end, the router receiving an IP packet may have to ARP for the destination host, further adding delays to the message traffic. As a result, it is not uncommon for the first packet of a transmission to be delayed or lost while addresses are being resolved. For this reason, routers will aggressively populate their ARP tables with known hosts.

IPv6 alleviates some of this, but creates other traffic issues, as the discovery process uses several different types of message, some of which are multicast. Switch behavior with multicast is similar in that multicast frames are sent everywhere throughout the Layer-2 domain. While routers, switches, and hosts have some ability to filter multicast traffic, we have increased the number of message types (redirects, router advertisements, router solicitations, neighbor advertisements, and neighbor solicitations), arguably increasing the overhead on the network.

Standards and RFCs

This chapter has taken you through the operation and structure of ARP. This information will be about all you will need to handle ARP on almost any network. However, there are some operations or standards that you should familiarize yourself with, even though you are not likely to run into them very often. Some of these are listed below.

RFC 826: “Ethernet Address Resolution Protocol”

This is the base address resolution standard. While not very descriptive, current operation is based on this RFC.

RFC903: “A Reverse Address Resolution Protocol”

This RFC approaches the issue of address resolution from the opposite direction. Instead of trying to learn a MAC address, RFC 903 describes how a host can discover a protocol (IP) address if it knows only the MAC address of the destination.

RFC1293: “Inverse Address Resolution Protocol”

This RFC allows a host to request a particular protocol address for a given hardware address.

RFC 1868: “ARP Extension — UNARP (Proxy ARP)”

This RFC suggests some solutions for potential limits of the original ARP RFC.

Summary

In this chapter, we examined the problem of Layer-2 address resolution. After examining the packets themselves and the addressing used, you should now have a solid understanding of ARP. We have also examined several of the operations used and the security threat represented by this distributed approach.

Review Questions

1. How many addresses are defined in ARP?
2. Is an ARP message routable?
3. Describe the Ethernet addressing used in the standard ARP request. Are the source and destination addresses unicast, broadcast, or multicast?
4. Describe the Ethernet addressing used in the standard ARP reply. Are the source and destination addresses unicast, broadcast, or multicast?
5. What is a gratuitous ARP?
6. What information is stored in an ARP table?
7. Can we send standard ARP messages directly to computers that are not on our own network?
8. Is ARP included in IPv6?
9. Is ARP a secure protocol?
10. What is the Ethertype hexadecimal value for an ARP message?

Review Answers

1. 2
2. No, the messages do not contain an IP header.
3. The ARP request uses a unicast address for the source and a broadcast address for the destination.
4. The ARP request uses a unicast address for the source and a unicast address for the destination.
5. This is a node sending an ARP request out for its own IP address in order to determine if another node is using the same address.
6. The ARP table contains a mapping between host MAC and IP addresses. It also shows whether each entry is static or dynamic.
7. No, ARP is not routable.
8. No

9. No. False ARP messages can be created to fool ARP tables. Hosts then make incorrect forwarding decisions. ARP transmissions are also sent in the clear.
10. 0806

Lab Activities

Activity 1—Determining Your IP Address and Your Default Gateway

Materials: A Windows computer with a network connection

1. In Windows, click the Start button.
2. In the run box, type **cmd** and press Enter. A command window opens.
3. Type **ipconfig /all**. This will display the IP address of your computer. The output will be similar to the following. This shows your IP address and the address of the default gateway.

Windows IP Configuration

Mini-PCI Express Adapter

```
Physical Address. . . . . : 00-22-68-90-D5-DB
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
IPv4 Address. . . . . : 192.168.15.100(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.15.1
DHCP Server . . . . . : 192.168.15.1
DNS Servers . . . . . : 24.56.123.4
                        106.12.34.56
NetBIOS over Tcpip. . . . . : Enabled
```

Activity 2—Examining the ARP Table

Materials: A Windows computer with a network connection

1. In the command window, type **arp -a**. This will provide the same output shown in [Figure 4-6](#). This gives an idea about nodes on the network with whom the computer has recently communicated.
2. Record the IP addresses you see in this table, as you'll need them later.

Activity 3—Packet Capture

Materials: A Windows computer with a network connection and packet capture software

1. To capture the ARP traffic, first clear the ARP table or cache. To do this, type **arp -d *** in the command window, then type **arp -a** to verify there are no entries.

2. In Wireshark, select your adapter and start a capture.
3. Back in the command window, ping one of the nodes previously listed in the ARP table. In the capture window, you should see the ARP request and ARP reply. These will be followed by the ICMP traffic. In pinging the default gateway, you may see the return ARP. That is, after pinging the gateway and seeing the associated traffic, the gateway generates its own ARP request directed back to you.

Activity 4—Gratuitous ARP

Materials: A Windows computer with a network connection, packet capture software, and a DHCP server like a Linksys router

To see a node ARPing for itself, typically the best time is right after an exchange with the DHCP server. This can be done on startup or by forcing the node to go through the IP address release and renewal process.

1. Start another capture.
2. In the command window type **ipconfig /release**. This forces the node to give up its IP address.
3. In the command window type **ipconfig /renew**. This causes the node to ask for an IP address again.
4. After the DHCP exchange has completed, you should see your node ARP for the very IP address it was assigned during the exchange. This is the gratuitous ARP.

Activity 5—How Long Does an ARP Table Entry Live?

Materials: A Windows computer with a network connection

1. In the command window, type **arp -a** to show the other nodes on the network.
2. Ping one of these nodes to refresh the ARP table entry.
3. At an interval of about 30 seconds, repeat the command **arp -a** until the entry disappears from the ARP table. How long did it take?

Network Equipment

Every network needs a certain amount of equipment facilitating the transmission of data. Equipment selection is based on the task at hand. To select the best possible device, we have to understand operations and ask questions regarding the interconnection of networks and computers. In addition to handling a specific set of tasks, infrastructure devices are designed to operate at a particular layer of the TCP/IP model. This means that hubs (also called repeaters), switches (or bridges), routers, access points (APs), and gateways can be inserted into the model just like protocols. While contemporary networks continue to use devices that are clearly defined, newer equipment can cross layer boundaries.

This chapter examines the components of a typical network in terms of their operation and behavior. We will also take a look at some of the security concerns associated with each. This includes changes for contemporary equipment, small office home office (SOHO) networks and the ubiquitous home gateway.

[Figure 5-1](#) provides the basis for most of this chapter. Both the OSI and TCP/IP models are represented along with the major protocols at each layer. The hubs, switches, and the router interconnect the two sides. The solid line at the bottom represents the physical connection between the devices, or what we might call the transmission path. However, devices may make decisions at upper layers, so they are logically connected via the dotted line. For example, switches process Layer-2 frames, which include MAC addresses.

Tables and Hosts

While this chapter is about network equipment, it wouldn't be complete without some mention of the hosts or nodes sitting on the network. Nodes have IP and MAC addresses, which are used to communicate over the network infrastructure. Almost without exception, devices use these addresses to forward packets or frames. It is also helpful to realize that within the network, almost everything follows a step-by-step process based on the same set of rules. The process can be traced through a series of tables and

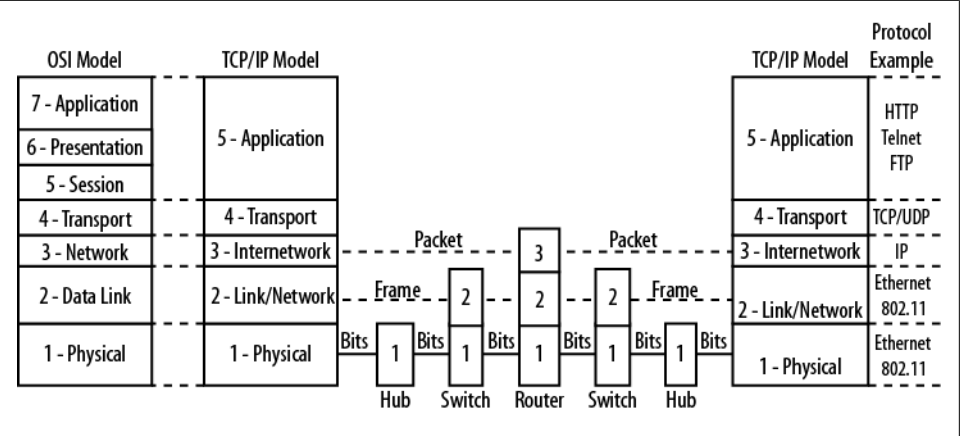


Figure 5-1. Models and equipment

packets. All of the devices on the network, including the hosts, have these tables and are capable of generating and/or forwarding packets. In the following sections, we will discuss the major types of equipment and the tables they use to process frames. [Table 5-1](#) lists these tables, along with each one's purpose and where it can be found on the network.

Table 5-1. Networking tables and purposes

Table	Location	Purpose
ARP table	Router and host	Maps IP addresses to MAC addresses
Source address table	Switch/bridge	Maps MAC addresses to switch ports
Routing table	Router and host	Determines correct interface and next hop
AP forwarding database	Access point	Collection nodes managed by the AP

Many of these operations are covered in greater depth in other chapters in this book, but let's examine a general example. [Figure 5-2](#) shows a basic network, illustrating the components covered in this chapter.

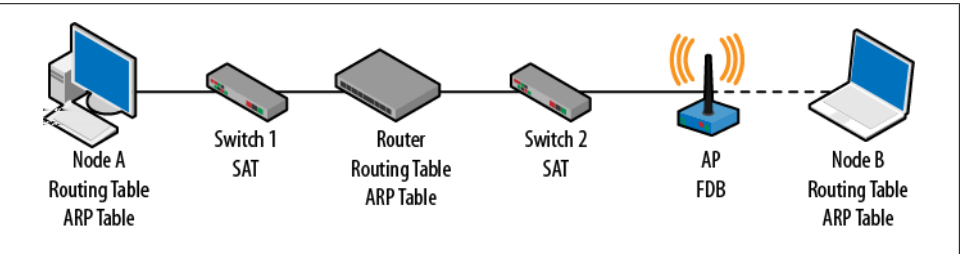


Figure 5-2. Basic network

If we assume that Node A is sending data to Node B, the following basic step-by-step process is followed:

1. Node A consults its host routing table to determine if the packet is for the local network or offsite.
2. Node A builds a frame for the Layer-2 destination by pulling the proper MAC address from its ARP table or by sending an ARP request and receiving the appropriate answer.
3. The frame is sent from Node A to the router via Switch 1.
4. Upon receiving the frame, Switch 1 consults its source address table (SAT) to determine the proper port for the destination.
5. The router receives the frame from Switch 1 and examines the IP header to determine the destination network.
6. The router processes its routing table to determine the correct interface to use for the destination.
7. The router builds a frame for the Layer-2 destination by pulling the proper MAC address from its ARP table or by sending an ARP request and receiving the appropriate answer.
8. The frame is sent from the router to Node B via Switch 2 and the AP.
9. Upon receiving the frame, Switch 2 consults its SAT to determine the proper port for the destination.
10. The AP receives the frame and, upon determining that Node B is in its forwarding database, sends the frame out to the wireless network to Node B.

It may seem like an awful lot of processing just to get a packet across two networks, but every single network transmission can follow a similar process.

Hubs or Repeaters

Starting from the bottom of the TCP/IP protocol stack and working our way up, the first device we come across is a hub. Hubs, or at least the need for hubs, are defined right along with the Layer-2 protocol standards. To clarify, let's start with the term *repeater*. IEEE 802.3 describes repeating as “the means used to connect segments of network medium together, thus allowing larger topologies and a larger MAU base than are allowed by the rules governing individual segments.”

The idea of a repeater has been around for a long time. The basic problem for a signal is that it tends to degrade over distance. A repeater is a point in the network where a weak but still readable signal can be cleaned up and retransmitted, thus extending the length of the network. According to the standards, repeaters “improve signal amplitude, waveform, and timing applied to the normal data and collision signals.” However, it is important to remember that this is not without boundaries, as protocols such as

Ethernet have rules regarding network size, particularly in the case of the collision domain. Repeaters may also permit the interconnection of dissimilar physical layers, such as UTP and fiber, and are only used in half-duplex environments. Hubs handle the same sort of operations and are defined in 802.3 as “A device used to provide connectivity between DTEs. Hubs perform the basic functions of restoring signal amplitude and timing, collision detection, and notification and signal broadcast to lower level hubs and DTEs.”

So what is a DTE? Data terminal equipment (DTE) typically refers to devices generating/terminating transmissions and, in this case, nodes. One other distinction is that hubs can generate some of the control signals used on the network. But with statements like “repeater sets are used as the hub in a star topology,” it can get a little confusing. A little reality can help us here. First, you don’t buy repeaters anymore, you buy hubs. Lots of networking folks think of hubs as multiport repeaters. Second, we don’t like to buy hubs. Most organizations have moved away from hubs, so this discussion is included here only to be complete.

So what have we got against hubs? Generally, they do not possess a great deal of intelligence. Early managed versions like those from 3Com actually had a nice collection of tools for controlling network traffic, but this was unusual. Most hubs are not much smarter than your toaster. Hubs act like repeaters—they forward traffic out all ports except the source interface. Thus, any transmission is sent to anyone connected to the same collision domain. This makes the hub a significant security concern. This behavior can vary between manufacturers. For example, some vendors isolate slower connections, but the hub broadcast behavior makes jacks installed in conference rooms, seating areas, or spare offices a real threat to network security.

One positive aspect of hubs is that they are very fast. In a small network with a few nodes, it is tough to beat the performance of a hub. In some scenarios, they will actually outperform a switch. But, as the number of nodes increases, we start to get collisions, which destroy the performance. As the network size increases, hub performance decreases, so hubs do not *scale* well. For these performance and security reasons, hubs have largely been replaced with switches.

Switches and Bridges

Moving up to the next TCP/IP layer, we have switches. Switches are the workhorses of modern networks. Where previously we might have used hubs to extend the network and add more nodes, we now use bridges and switches. The term *bridge* is used to describe a device that interconnects collision domains. Collisions that appear on one side of a switch are not allowed to propagate to the other. In [Figure 5-3](#), a collision on Hub 4 will propagate to all of the nodes in the same collision domain. This includes PC 4, PC 5, PC 6, and the switch port itself. However, the switch/bridge will prevent further transmission. PC 1 and PC 2 will be blissfully unaware of the collision. Switches and bridges also filter out traffic that should not be forwarded. For example, if PC 1

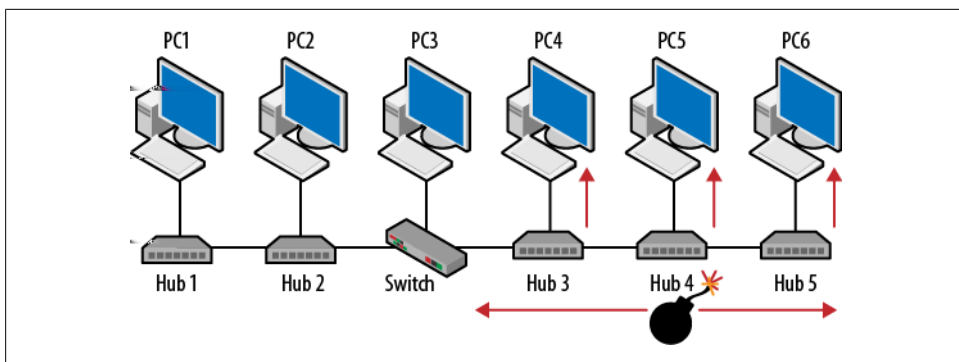


Figure 5-3. Collision boundary

and PC 2 are having a conversation, there is no reason to force the other nodes to listen, so PC 3, PC 4, PC 5, and PC 6 will not hear the frames.

Generally, switches are considered to be newer, high-powered versions of bridges, performing the same functions and bringing some extras to the table. In fact, we don't buy bridges anymore except in the case of wireless bridges. But these are not typical and aren't used if it can be avoided. Early versions of bridges and bridging standards did not include many of the advanced features we have come to rely on. For these reasons, our discussion will center on switches.

As a replacement for hubs, switches have done very well. The purchase price (cost per port) of a switch has come down considerably and switches have many features that hubs (or early bridges) never possessed, including changes to the forwarding behavior, support for virtual LANs (VLANs), basic port security, and 802.1X.

The key difference between switches and hubs is that switches forward based on MAC addresses. To accomplish this, the switch consults a SAT before transmitting a frame to the destination. This means that for a significant portion of network traffic, only the proper destination receives the transmission. The operation of switches and bridges is defined in IEEE 802.1D, titled "IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges." 802.1D provides the guidelines for support of the MAC layer, including interconnecting network segments, support for several different Layer-2 protocols, handling of errors, and, of course, forwarding of frames. In addition, the standard describes the behavior of other Layer-2 protocols such as spanning tree.

So, how does a switch work? Aside from a couple of rules for specialized frames, Ethernet switches operate in a very straightforward way: receive a frame, read the addresses, error check, and forward to the correct port. We will work through a couple of examples to explore the details. [Figure 5-4](#) depicts a typical topology with a switch at the center.

Switches keep track of the location of network nodes via the SAT or MAC address table. Remember that all network nodes have a unique MAC address and each Ethernet frame identifies the source and destination by these MAC addresses. The table is a mapping

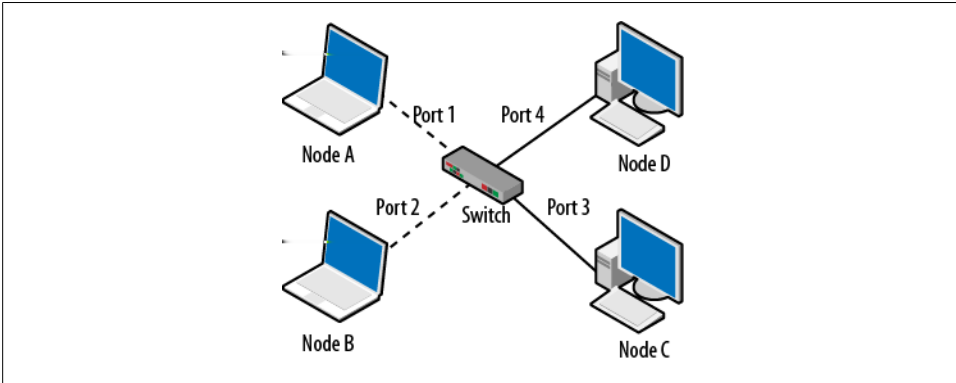


Figure 5-4. Basic switch topology

between the MAC addresses and the switch ports. This table also keeps track of the VLANs configured on the switch. The SAT for the network shown in [Figure 5-4](#) might look like [Table 5-2](#). Typically, nodes all start out in VLAN 1.

Table 5-2. Basic switch SAT

MAC address	VLAN	Port
Node A MAC	1	1
Node B MAC	1	2
Node C MAC	1	3
Node D MAC	1	4

A switch has some basic procedures to follow:

1. When a frame is received, buffer the frame and perform the frame error check. If there are problems, discard the frame.
2. Copy the source address and port number into the SAT.
3. Look in the SAT for the destination MAC address.
4. If the address is known, forward to the correct port. If the address is not known, send the frame everywhere except the source port. This is called *flooding*.
5. If the destination is a broadcast address (ff:ff:ff:ff:ff:ff), send the frame everywhere except the source port. In many cases, this is also the behavior for multicast frames. Recall that multicast frames commonly begin with 01. VLANs can reduce the effect of flooding because they can be used to segment the switch into smaller logical network segments, but this is a story for another day.



Switches will continue to forward broadcast frames from one another until the frame hits a Layer-3 boundary, like a router. The distance that a broadcast frame will travel is called the *broadcast domain*.

Let’s take the example of a ping between Node A and Node B in [Figure 5-4](#) and add a little more detail. For this example, we will assume the SAT entries in [Table 5-2](#) are not present. SAT entries usually timeout in five minutes or so. For this example, we will use the frame shown in [Figure 5-5](#).

```
.Ethernet II, Src: Ibm_43:49:97 (00:11:25:43:49:97), Dst: Cisco_35:1a:d0 (00:19:55:35:1a:d0)
  Destination: Cisco_35:1a:d0 (00:19:55:35:1a:d0)
  Source: Ibm_43:49:97 (00:11:25:43:49:97)
  Type: IP (0x0800)
  Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.254 (192.168.1.254)
  Internet Control Message Protocol
```

Figure 5-5. Ethernet frame with ICMP echo request

When the ICMP echo request is received at the switch, the switch buffers the entire frame and calculates the CRC. If there are no problems with the error check, the switch places the MAC address of Node A (Src) into the SAT and notes the port number and VLAN ID.

MAC Address	VLAN	Port
00:11:25:43:49:97	11	

Next, the destination MAC address is examined. Node B is currently missing from the SAT, so the switch forwards the frame out of all ports except the original source port. So, the ICMP echo request is sent out ports 2, 3, and 4. Node B receives the frame and answers back. When the switch receives the ICMP echo reply (shown in [Figure 5-6](#)), the switch buffers the entire frame and calculates the CRC.

```
Ethernet II, Src: Cisco_35:1a:d0 (00:19:55:35:1a:d0), Dst: Ibm_43:49:97 (00:11:25:43:49:97)
  Destination: Ibm_43:49:97 (00:11:25:43:49:97)
  Source: Cisco_35:1a:d0 (00:19:55:35:1a:d0)
  Type: IP (0x0800)
  Internet Protocol, Src: 192.168.1.254 (192.168.1.254), Dst: 192.168.1.1 (192.168.1.1)
  Internet Control Message Protocol
```

Figure 5-6. Ethernet frame with ICMP echo reply

Take a look at the source and destination MAC addresses in [Figure 5-6](#). They have flipped, indicating that this is a reply. The IP addresses have also flipped. If there are no problems with the error check, the switch places the MAC address of Node B into the SAT and notes the port number and VLAN ID.

MAC Address	VLAN	Port
00:11:25:43:49:97	1	1
00:19:55:35:1a:d0	1	2

As the destination is examined, we find that Node A has an entry in the SAT, so the frame can be directed to Port 1 only. This learning process is what makes a switch *transparent*. This is also what allows the switch to filter network traffic, prevent errors, and stop the propagation of collisions. Figure 5-7 shows the SAT from an operating Cisco switch. The term “dynamic” means that the switch learned the address.

Non-static Address Table:			
Destination Address	Address Type	VLAN	Destination Port
0004.9b4b.5701	Dynamic	1	FastEthernet0/1
0004.9b4b.5701	Dynamic	2	FastEthernet0/1
0004.9b4b.5701	Dynamic	3	FastEthernet0/1
000e.0c76.5ad4	Dynamic	2	FastEthernet0/7
000e.0c77.20e4	Dynamic	2	FastEthernet0/1
000e.0c77.2322	Dynamic	3	FastEthernet0/1
0011.212c.15e0	Dynamic	3	FastEthernet0/23
0011.212c.15e1	Dynamic	2	FastEthernet0/13

Figure 5-7. Cisco switch SAT

In this particular case, there are three VLANs, and we can see that Port 1 (FastEthernet0/1) has several associated MAC addresses. This is because another switch was connected at that point. This is reflected in the topology shown in Figure 5-8. Two switches are interconnected via Port 3 on Switch 1 and Port 3 on Switch 2. As normal traffic flows, the switches will learn where all of the MAC destinations are by recording the source MACs from Ethernet transmissions.

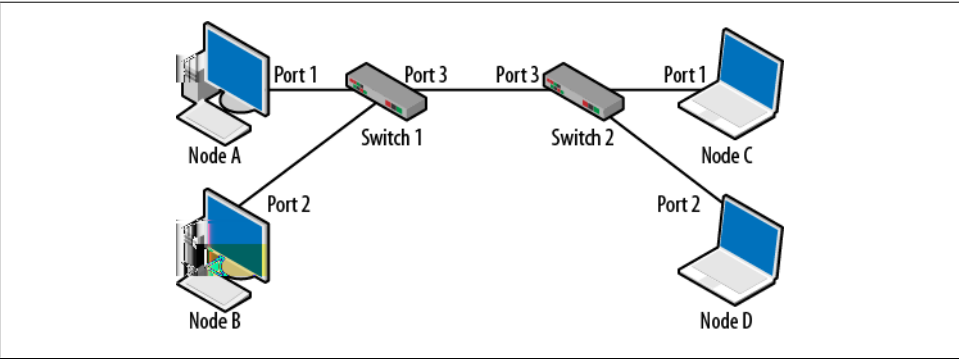


Figure 5-8. Two-switch topology

In topologies such as this, it is impossible for a switch to connect directly to each destination. The only piece of information a switch will possess is the source, from its perspective. So, from the perspective of Switch 2, all frames appear to have come from the single port connected to Switch 1. The reverse is also true. Building on what we know of SATs and the learning process, the SATs for the two switches would look like [Table 5-3](#).

Table 5-3. SATs for a two-switch topology

Switch 1 SAT			Switch 2 SAT		
Node A MAC address	VLAN 1	Port 1	Node A MAC address	VLAN 1	Port 3
Node B MAC address	VLAN 1	Port 2	Node B MAC address	VLAN 1	Port 3
Node C MAC address	VLAN 1	Port 3	Node C MAC address	VLAN 1	Port 2
Node D MAC address	VLAN 1	Port 3	Node D MAC address	VLAN 1	Port 1

Node A sends traffic to Node D, and Switch 1 forwards the traffic out Port 3. Switch 2 receives the frame and forwards the frame to Port 1. If these nodes cease to generate traffic, their addresses will be removed from the SAT.

Ethernet nodes obey the CSMA/CD access method. By default, switches also engage in CSMA/CD because, until link negotiation is complete, the switch has no indication as to what might be downstream of a particular port. Finally, switches read and process the Ethernet frames, but they are not supposed to change them in any way. This means a frame that is forwarded by a switch looks exactly the same as it did when it was received by the switch. We will see that this is not always true when dealing with network devices.

Access Points

Before we leave Layer 2, let's talk about APs. APs are sometimes called *wireless hubs* because the medium is shared. While calling them hubs this isn't exactly accurate, we can understand the confusion. Just like a hub, APs broadcast traffic to anyone capable of hearing it. But, again, this is more due to the type of media than the operation of the AP. Let's take a closer look at what an AP is supposed to do. The 802.11 standard describes several major responsibilities:

- Notifying network users of its presence and negotiating connections
- Forwarding traffic between the wired and wireless sections of the network
- Handling traffic for all of the wireless nodes currently connected
- Encrypting of data traffic if configured
- Handling nodes in power save mode

There have been several modifications to this standard, including 802.11b, 802.11a, 802.11g, and 802.11n. However, with the exception of 802.11n, there have not been very many changes to the Layer-2 behavior. Most of the changes have been to signaling and modulation, so the responsibilities outlined above haven't changed.

Nodes use a three-step process when joining a wireless network. First, the network must be found either by passive or active scan. Second, the node must authenticate with the network. Third, the node is associated with the network. Actually, a node associates with an AP possessing the service set identifier (SSID) for the desired network. Once this has been completed, the node is entered into the AP's forwarding database. The association is a key relationship, because the nodes understand which AP is theirs, and the AP takes responsibility for the associated nodes. This means the AP will not forward traffic for nonassociated nodes, and when wireless laptops have traffic to send, one of the MAC addresses included in the frame is that of the AP. A wireless data frame is shown in [Figure 5-9](#).

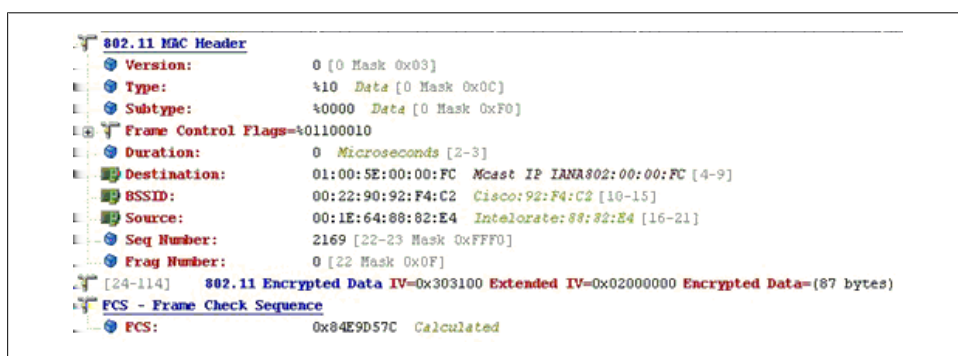


Figure 5-9. Wireless data frame

This frame was captured using Omnipeek. While the details of 802.11 operation and frame content are a little beyond the scope of this text, we can see that three addresses are used in an 802.11 data frame: destination, BSSID, and source. The destination and source MAC addresses are exactly the same as in an Ethernet frame and serve the same purpose. The BSSID is the MAC address of the AP. This allows the AP to determine which frames to process.

This brings us to the behavior of the AP. Based on these decisions, the AP is forwarding traffic after examining the MAC addresses used in the frame, so it operates similar to a switch. We already know that traffic sent out by the AP is broadcast to anyone listening. This includes frames such as beacons and other wireless management traffic. In the case of [Figure 5-10](#), the AP handles transmissions from Node C and Node D, because they are associated with the AP. When two wireless nodes communicate, as long as they are connected to the same AP, the transmission is limited to the wireless segment and does not cross to the wired side. The same can be said of two wired nodes (Node A and Node B) because the frames stay on their side of the network. When traffic from the wired-side switch arrives at the AP, it is buffered and sent out to the wireless network. If we know how a switch operates, we can understand how traffic finds its way to the AP.

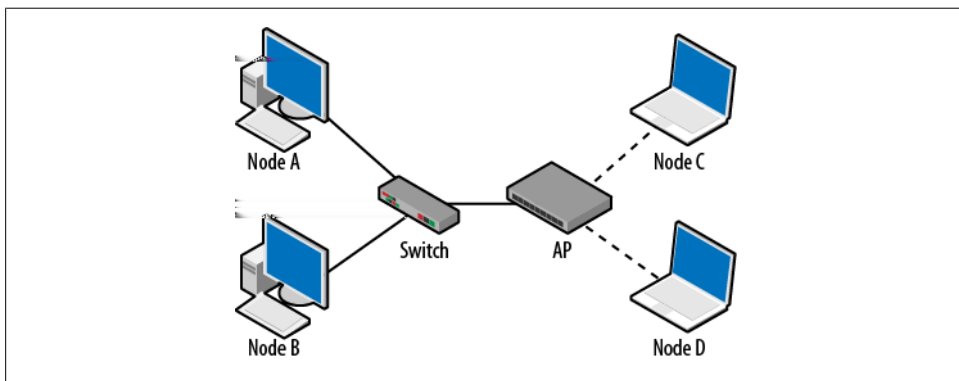


Figure 5-10. Small wireless topology

When Node A and Node B generate broadcast or multicast traffic, it will be forwarded by the switch. It turns out that the AP does the same thing. In addition, any traffic that is destined for Node C and Node D will be forwarded to the AP from the switch because of the switch SAT. Again, close examination of this behavior reveals that most of the forwarding decisions are based on MAC addresses or handling of broadcast/multicast frames.

There are differences between the Layer-2 behavior seen on a switch and that seen on an AP. 802.11 frames have a greater number of control fields compared to Ethernet. In addition, 802.11 frames are larger. So, an AP is one network device that must modify the Layer-2 frame. However, like a switch, the AP does not care about Layer-3 addresses or headers.

Routers

As we move up the layers in our networking model, the biggest differences in device operation revolve around addressing and what the device “cares” about. Routers live at Layer 3, as they deal primarily with IP addresses. Building on [Figure 5-1](#), we can see the area of concern for each type of device and the type of addressing processed. Unlike any other device, routers will forward traffic between IP based networks after examining the Layer-3 header ([Figure 5-11](#)).

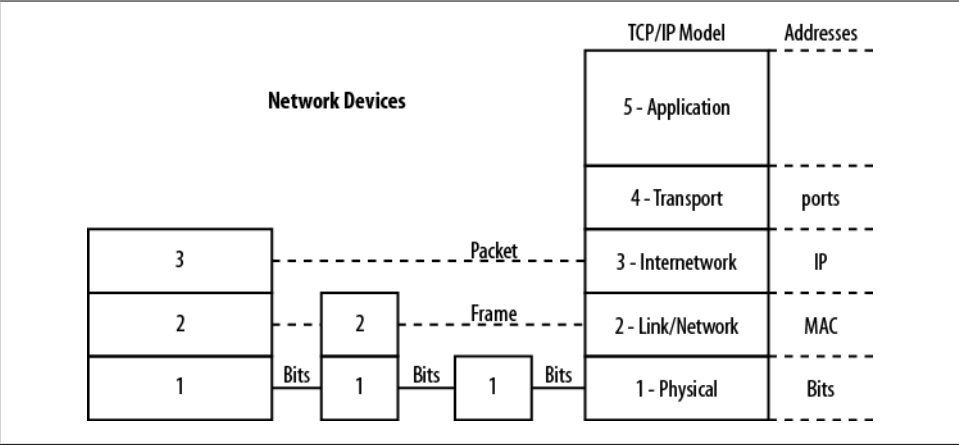


Figure 5-11. Device addressing

Though routers act on IP addresses, that doesn’t mean they are not actively engaged in the network. A router can act much like a host in some situations. They require IP addresses in order to operate (switches and APs do not), they use and respond to ARP messages, and listen to (but will not forward) all broadcast frames. This means that not only will a router forward traffic for hosts, it can be contacted directly. Routers are also known by another name—*default gateway*. To be fully functional, a host must be able to communicate off of its LAN. So, when a host is configured, either statically or via DHCP, it has a default gateway, as shown in [Figure 5-12](#). This default gateway is actually the router that will receive transmissions from the network nodes when sending traffic offsite.

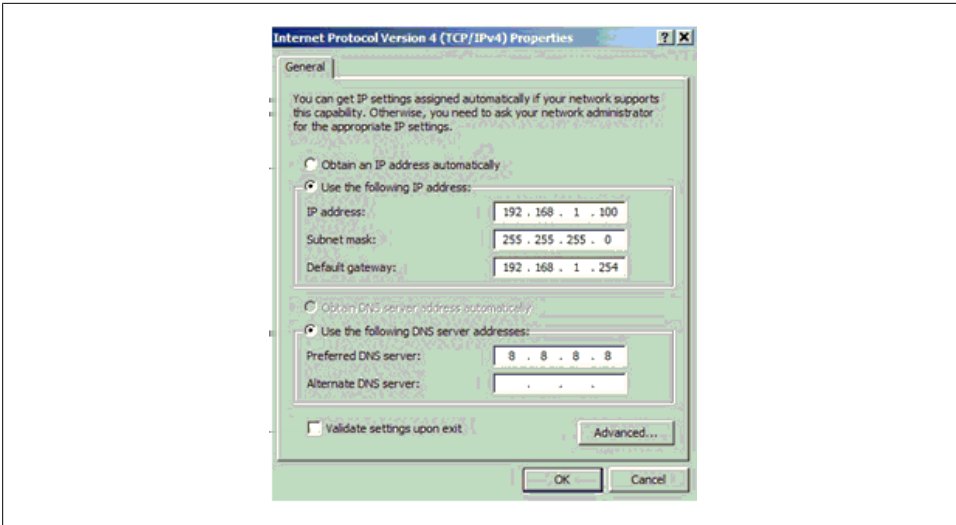


Figure 5-12. Host IP settings

There are three major operations/objects to think about on a router:

- Routing process
- Routing protocols
- Routing table

The routing process is the actual movement of IP packets from one port to another, the routing table holds the information used by the routing process, and routing protocols such as RIP or OSPF might be used to communicate with other routers. Hosts, switches, and APs do not participate in these processes, although the SAT and forwarding data-base (FDB) do behave in similar fashion. A simple routing table appears in [Figure 5-13](#).

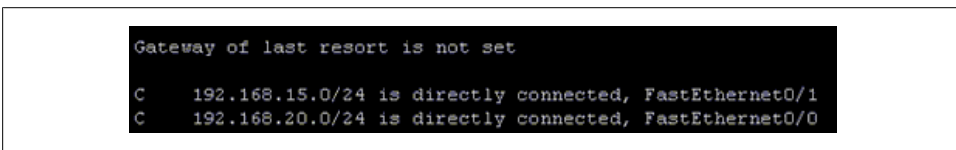


Figure 5-13. Router routing table

Another major difference between router behavior and the operation of lower-layer devices is that routers change the Layer-2 frames. When a transmission is destined for an offsite location, the frame on the incoming side is completely removed and replaced with an appropriate frame on the outgoing side. Consider the small network shown in [Figure 5-14](#).

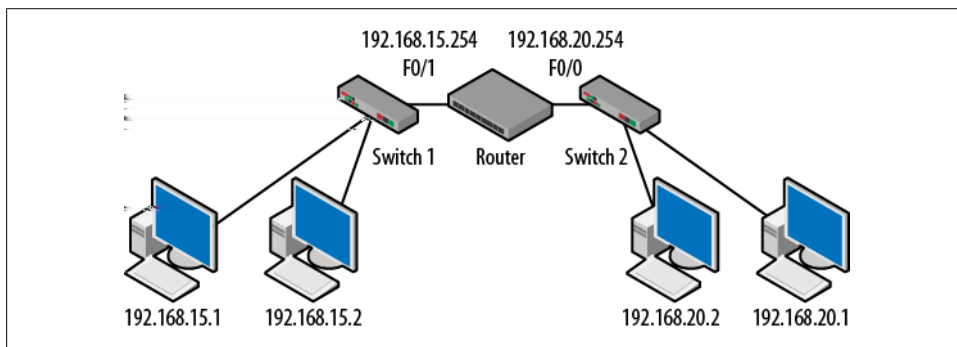


Figure 5-14. Routed topology

The two networks shown in [Figure 5-14](#) have been interconnected via the router. Addresses for the nodes and router interfaces have been filled in. Node A and Node B use the leftmost router interface (192.168.15.254) as their default gateway. Node C and Node D use 192.168.20.254. We already know that if Node A and Node B communicate, the entire process is handled by these two nodes and Switch 1. Switch 1 will examine the MAC addresses. An important detail that we often gloss over is that the Ethernet frames will be addressed from the MAC address of Node A to the MAC address of Node B.

This changes as we go to the network on the other side of the router. Nodes on a particular network will never know the MAC addresses for nodes residing on distant networks. Instead, Layer-2 frames are addressed to router interfaces and this is actually how we ask a router to forward something for us. So, when Node A wishes to communicate with Node D, it sends an IP packets encapsulated in an Ethernet frame to the router. The IP packet is addressed from Node A to Node D, but the Ethernet frame is addressed from the MAC address of Node A to the MAC address of the left router interface, or F0/1. As the router forwards the IP packet to the other network (192.168.20.0), the frame is rebuilt for the new network. The IP packet is still addressed from Node A to Node D, but the Ethernet frame is addressed from the router interface F0/0 to the MAC address of Node D. The MAC addresses are learned via ARP. If we now go back to the process described at the beginning of the chapter, the whole thing begins to come together.

Another Gateway

At the beginning of this chapter, there was mention of a device called a gateway. This is not to be confused with the default gateway or the home gateway. In this case, *gateway* refers to a device that understands and converts between two different networking models. For example, a gateway is necessary when connecting IPX/SPX and TCP/IP and Appletalk networks together. Since TCP/IP is the dominant model, the usage of this type of gateway is much diminished. However, VoIP breathes new life into this old term because now TCP/IP networks must talk directly with Signaling System 7 on the telephone network.

Multilayer Switches and Home Gateways

The network devices described so far behave in a very stratified manner. They rarely leave their networking model layers and do not share responsibilities. But, just as hubs and bridges are fading away, so too are the straightforward single-use switch and router. Multilayer switches are an attempt to achieve performance gains while collapsing the chassis a bit. If we consider the routed topology shown in [Figure 5-14](#), we can see that there are three network devices and that the router takes up a port on each switch. However, if the switches understood a little more about routing, we could reduce the number of devices and power outlets, recover some network ports, use less A/C cooling, and save some space. Most vendors have a collection of products that accomplish exactly that. In fact, it is getting more and more difficult to find a device that is “just a router” or “just a switch.” Another nice feature is that now a single device can route between VLANs. However, there is a bit more to the story than these improvements alone. Multilayer switches are also trying to improve the forwarding efficiency of the network.

The idea is that if the device understands something about the topology of the network (i.e., MAC addresses in addition to IP networks), this knowledge might be leveraged to improve forwarding times. For example, in [Figure 5-14](#), a transmission from Node A to Node D would normally be processed by two SATs and a router routing table. With multilayer switching, one SAT and one routing table would be in use. Finally, if we take advantage of the topology information, it is possible that the routing table would not be necessary. With these modifications, the topology shown in [Figure 5-14](#) might be rebuilt with a multilayer switch as shown in [Figure 5-15](#).

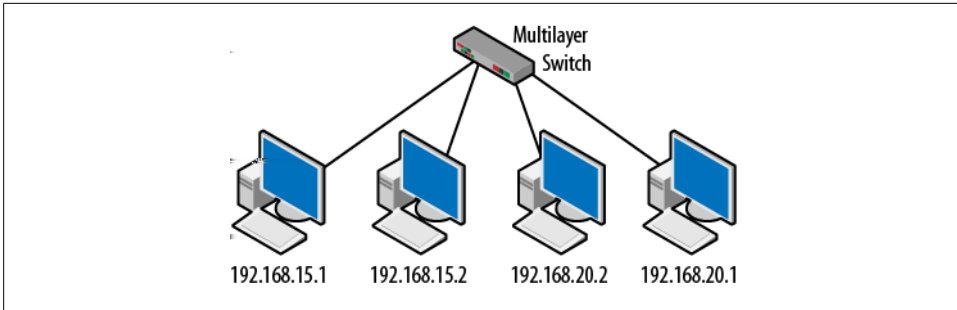


Figure 5-15. Multilayer switch topology

Home gateways are another interesting type of device, because they combine so many features from so many layers of the TCP/IP networking model. A typical unit includes four to eight switch ports and a wireless interface. All of the nodes connected to these interfaces are on the same network and receive IP addresses from the gateway. A common network address is 192.168.1.0. Since the gateway is providing addresses, it is a DHCP server. The gateway also routes traffic to the outside world, but before it forwards the traffic, the source IP addresses are translated by the gateway. The 192.168 address space is considered a private network and cannot be used on the public Internet. This translation is called network address translation (NAT) and is primarily concerned with the conservation of IP addresses. SOHO networks can use the same network addresses because of this translation. Finally, outside requests will not be forwarded to the internal network, because the gateway is preconfigured to block unsolicited packets. Thus, the gateway is performing the functions of a basic firewall. Figure 5-16 depicts a functional diagram of the gateway.

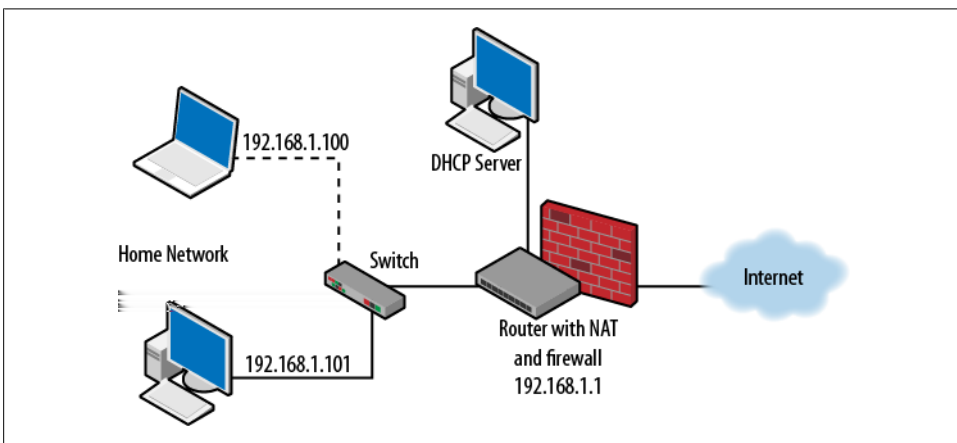


Figure 5-16. Home gateway

Security

If we look at these devices and their topologies from a security standpoint, we can begin to understand potential attack vectors that might take advantage of their basic operation. Every single device and protocol that is part of the network can be attacked or used as an attack vector. If we imagine that an attacker might either plug into our network or be listening on the wireless side, we place the threat at the front door. Starting with Layer-1 hubs, the weakness is very clear—they operate in a shared media and make no effort to filter traffic. In Layer 2, switches filter traffic based on MAC address, but they are willing to forward broadcast frames everywhere. This means that over time, an attacker can learn quite a bit via passive eavesdropping. Common improvements are the addition of VLANs to reduce exposure and port-based security to prevent unauthorized MAC addresses from gaining access to the network.

Wireless networks present a completely different problem in that the traffic is broadcast out to the world. However, if the wireless APs are connected back to a switch configured with VLANs, we can minimize the amount of traffic that is broadcast. Wireless networks should also be protected with encryption. Historically, we had the wired equivalent privacy (WEP), followed by WiFi Protected Access with a Pre-Shared Key (WPA-PSK) for small networks. Since both of these have been cracked, the recommended minimum encryption for these network types is WPA2-PSK with a 20-character passphrase using number and letter combinations. This goes a long way to prevent dictionary or brute-force attacks.

Routers provide the greatest amount of inherent filtering because they will not forward anything unless the packets are destined for another network. So, broadcast frames are not allowed to pass. The problem is that, by default, routers are also very accommodating. If you send a router a packet for forwarding, it will do its best to see that it gets there, whether you are a bad guy or not. In addition, routers have an IP address. That is not to say that switches and APs never do—switches and APS will often be given IP addresses for the purpose of remote access (Telnet, SSH) and management such as Simple Network Management Protocol (SNMP). But, routers always have IP addresses and are often directly connected to the outside world. Like hosts, routers can be attacked, because they can be reached via the IP address. This also means routers, like hosts, must be secured in the areas of accounts, patch levels, shutting down access to services, and firewalls or filter rules. Finally, any protocol (such as a routing protocol) that involves the router should also be secured.

Summary

When building a network, you are going to use the same building blocks as most other networks: switches, APs, and routers. Newer implementations will see increased use of multilayer switches, though it is tough to justify throwing out perfectly good routers and switches. In this chapter, you have hopefully developed an understanding of not

only the purpose of each device, but how it operates. Each device has a particular set of functions as it processes packets and frames.

When packets traverse a network, it is possible to track each and every decision along the way in order to determine the best device to use. The tables used to make these decisions can also be used to diagnose problems, optimize performance, and understand potential security threats.

Review Questions

1. Match the following devices to the proper networking model layer.

- | | |
|-----------------|-----------------|
| a. Hub | A. Application |
| b. Switch | B. Transport |
| c. Access Point | C. Internetwork |
| d. Router | D. Network/link |
| | E. Physical |

2. Match the following devices to the type of addressing processed.

- | | |
|-----------------|----------------|
| a. Hub | A. Port |
| b. Switch | B. IP address |
| c. Access Point | C. MAC address |
| d. Router | D. 1s and 0s |

3. Match the following devices to the tables used to process packets or frames. For some devices you may select more than one table.

- | | |
|-----------------|-------------------------|
| a. Host | A. Routing Table |
| b. Switch | B. Source Address Table |
| c. Access Point | C. Forwarding Database |
| d. Router | D. ARP Table |

- Switches stop collisions from propagating. True or false?
- APs stop broadcast frames from propagating. True or false?
- A SAT is a mapping between MAC address and IP addresses. True or false?
- While routers process packets, they do not change or manipulate the frame or packet itself. True or false?
- A default gateway is a router interface. True or false?
- A multilayer switch is simply a switch that understands how to forward packets between VLANs. True or false?
- One of the reasons switches are very fast even when connected to a wireless network is that Ethernet frames and 802.11 frames are almost identical. True or false?

Review Answers

1. E, D, D, C
2. D, C, C, B
3. Host: A, D; Switch: B; Access Point: C; Router: A, D
4. True
5. False
6. False
7. False
8. True
9. False
10. False

Lab Activities

Activity 1—Traffic Comparison

Materials: Wireshark, switch, three computers

1. Configure the IP addresses for the computers.
2. Connect all three computers to the switch.
3. On all three machines, start Wireshark.
4. Ping between two of the computers.
5. Compare the traffic that is seen on the computers. What is the difference in the traffic? Why is there a difference?

Activity 2—Layer-2 Trace

Materials: Two computers, Wireshark, switch

1. Configure the IP addresses for the computers.
2. Connect both computers to the switch.
3. Start a capture on each machine.
4. Access the SAT on the switch.
5. Access the ARP (`arp -a`) and routing tables (`route print`) on the hosts.
6. Ping from one computer to another.
7. Using the tables and packet capture, explain exactly how the packet gets from one node to another. Include every device in your explanation.

Activity 3—Tables

Materials: AP, router, switch, wired host, wireless host (use devices that are available, not all are required)

1. Add a router and AP to the topology you created in Activity 2. This will give you two IP-based networks.
2. Connect a node to the AP.
3. Configure the IP addresses and ensure all nodes can ping each other.
4. Establish a management connection to each of the devices used.
5. Pull up the tables discussed in this chapter for each of the devices.
6. Explain the content of these tables and how it got there.

Activity 4—Layer-3 Trace

Materials: Two computers, Wireshark, two switches, router (one switch may be used; however, configuring VLANs may add confusion)

1. Configure a topology similar to the one shown in [Figure 5-14](#).
2. Start a capture on each machine.
3. Access the SAT on the switch.
4. Access the ARP (`arp -a`) and routing tables (`route print`) on the hosts.
5. Access the routing table for the router.
6. Ping from one computer to another.
7. Using the tables and packet capture, explain exactly how the packet gets from one node to another. Include every device in your explanation.

Activity 5—Traffic Comparison

Materials: Two computers, Wireshark, two switches, router (one switch may be used; however, configuring VLANs may cause confusion)

1. Using the topology from Activity 4, start a capture on both computers.
2. Ping from one node to another.
3. In the packet captures, compare the packets and frames seen on one side of the router to those seen on the other.
4. What are the differences? Why are they different?

Internet Control Message Protocol

“ICMP error messages signal network error conditions that were encountered while processing an internet datagram. Depending on the particular scenario, the error conditions being reported might or might not get solved in the near term.”

—RFC 4443

The Internet Control Message Protocol (ICMP) provides error messages and feedback during network operations. These messages provide insight into the current state of the network, making it simpler to troubleshoot network connectivity problems. An ICMP error message is often sent in response to a failed transmission attempt. In addition, ICMP messages allow us to ask the network for information. This chapter will explain ICMP by taking an in-depth look at several of the most common message types and the conditions that cause them. The basic tools used herein include captures from Wireshark, the output from the Windows command (DOS) shell, and output seen from a Cisco router.

RFC 792 defines several different ICMP message types and forms the basis of our discussion. Contemporary networks typically use a handful of these message types to deal with standard issues. Several other RFCs have contributed to ICMP, most of which are designed to handle very particular situations. A list of these RFCs is included at the end of the chapter.

ICMP exists within the Internetwork Layer (Layer 3) of the TCP/IP model and is encapsulated in an IP datagram (shown in [Figure 6-1](#)). All IP-based nodes, regardless of operating system or device type, use ICMP for error and notification messages. However, the behavior can vary between systems.

According to the RFC, to keep the error messages to a minimum, no ICMP error messages are sent about ICMP error messages and, in the case of fragmented IP packets, error messages are sent only regarding the first fragment (fragment 0) of the IP packet. So, if an ICMP error message is generated because of a problem, this message cannot

```
Ethernet II, Src: Cisco_28:1b:e1 (00:05:5e:28:1b:e1), Dst: HonHaiPr_12:1c:a9 (00:1f:e2:12:1c:a9)
Internet Protocol, Src: 192.168.2.254 (192.168.2.254), Dst: 192.168.2.1 (192.168.2.1)
Internet Control Message Protocol
```

Figure 6-1. ICMP encapsulation

inspire the creation of other error messages. However, in the case where the ICMP message is informational, other ICMP messages can result. For example, if you were to ping a node on another network via an ICMP echo request, and the intervening routers could not forward the traffic to the destination, the router may respond with an ICMP destination unreachable message. These ICMP message types are explained more fully later in this chapter.

Security is not often associated with ICMP, as many of the message types defined provide clear text information about the network or its operation. In addition, many network devices and hosts are configured to answer ICMP questions without hesitation. In the case of traceroute, packets are normally permitted to traverse the network regardless of the source or purpose. Finally, the ICMP echo request is often used in packet injection attacks designed at breaking encryption schemes, because the alphabet is contained in the payload. This provides an easily recognizable pattern for the before-encryption and after-encryption snapshots.

Structure

Generally, ICMP messages have similar structure. The next series of figures are Wireshark views of the same ICMP packet, each pointing out a different aspect. The ICMP echo request is the payload for the IP packet and is contained within the data field. ICMP does not possess a TCP or UDP header. Expanding the IP header shows us that the IP packet payload type is 01 for ICMP. This is highlighted in [Figure 6-2](#).

```
Ethernet II, Src: Cisco_28:1b:e1 (00:05:5e:28:1b:e1), Dst: HonHaiPr_12:1c:a9 (00:1f:e2:12:1c:a9)
Internet Protocol, Src: 192.168.2.254 (192.168.2.254), Dst: 192.168.2.1 (192.168.2.1)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  Total Length: 60
  Identification: 0x026d (621)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 255
  Protocol: ICMP (0x01)
  Header checksum: 0x3304 [correct]
  Source: 192.168.2.254 (192.168.2.254)
  Destination: 192.168.2.1 (192.168.2.1)
Internet Control Message Protocol
```

Figure 6-2. IP payload type ICMP

In [Figure 6-3](#), an expanded the ICMP header depicts the general format for ICMP messages. This message happens to be an echo reply.

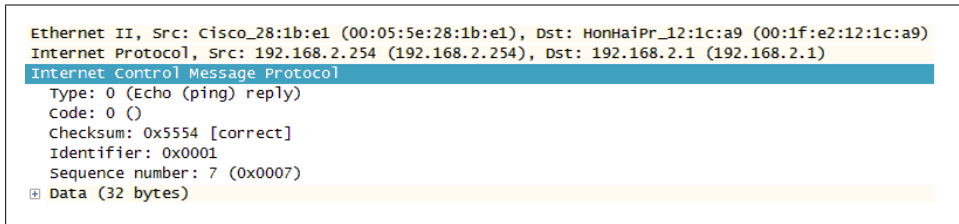


Figure 6-3. ICMP general format

Type

Defines each kind of message, common types include:

- 0—Echo reply
- 3—Destination unreachable
- 4—Source quench
- 5—Redirect
- 8—Echo request
- 9—Router advertisement
- 10—Router solicitation message
- 11—Time exceeded
- 12—Parameter problem
- 13—Timestamp
- 14—Timestamp reply
- 15—Information request
- 16—Information reply

Code

Each type of message defines one or more codes that are used to indicate the operation or variation in the message type. If there is only one code, its value will typically be 0.

Checksum

16-bit one's complement of the one's complement sum of the ICMP message starting with the ICMP type. When computing the checksum, the checksum field should be 0.

Identifier

This field is not always present. In this case ([Figure 6-3](#)), it provides a reference point for matching the correct echo reply to the original echo request.

Sequence number

This value is also used in matching requests and replies.

Internet header + 64 bits of data datagram

In the case where an ICMP message is generated in response to a message from a network node, the ICMP message will include a copy of the original IP header plus 64 bits of the original message payload. This is to help the source host in identifying the reason for the ICMP message. As shown in [Figure 6-3](#), not all ICMP messages contain this field.

Payload

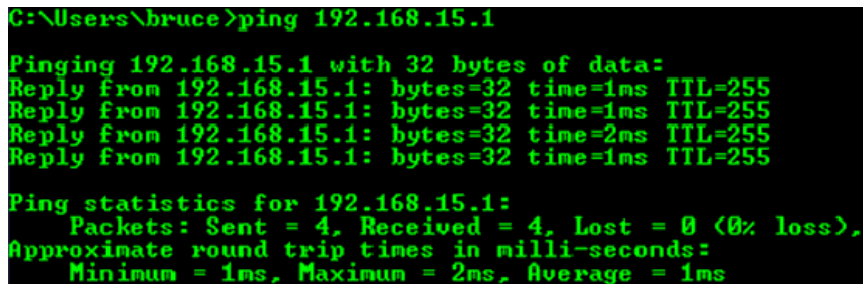
In the case of an echo request, this is the data generated by the ICMP process. It is present only in some of the ICMP message types.

Operations and Types

This section covers the common ICMP protocol behavior, message purpose, and the scenarios in which they are produced.

Echo Request (Type 0) and Echo Reply (Type 8)

A common tool for testing connectivity is the ping program, which generates ICMP echo request packets. By sending an ICMP echo request, the sender is asking the receiving node to send an ICMP echo reply back. This is called the *responder function*. Ping program behavior varies between operating systems, but the basic rules are the same. An example of the Windows exchange is shown in [Figure 6-4](#). By default, Windows ping sends four ICMP echo requests, each having a payload size of 32 bytes.



```
C:\Users\bruce>ping 192.168.15.1

Pinging 192.168.15.1 with 32 bytes of data:
Reply from 192.168.15.1: bytes=32 time=1ms TTL=255
Reply from 192.168.15.1: bytes=32 time=1ms TTL=255
Reply from 192.168.15.1: bytes=32 time=2ms TTL=255
Reply from 192.168.15.1: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.15.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

Figure 6-4. Ping command-line output

The packets resulting from this command are shown in [Figure 6-5](#). In this case, the entire conversation includes the ARP request, ARP reply, four ICMP echo requests (type 8), and the four corresponding ICMP replies (type 0). The ARP messages are generated because the ARP table on the sender does not include the address of the destination.

20	26.889800	HonHaiPr_90:d5:db	Broadcast	ARP	who has 192.168.15.1? Tell 192.168.15.103
21	26.892093	Cisco-Li_7f:fb:9d	HonHaiPr_90:d5:db	ARP	192.168.15.1 is at 00:14:bf:7f:fb:9d
22	26.892134	192.168.15.103	192.168.15.1	ICMP	Echo (ping) request
23	26.893278	192.168.15.1	192.168.15.103	ICMP	Echo (ping) reply
24	27.892528	192.168.15.103	192.168.15.1	ICMP	Echo (ping) request
25	27.894209	192.168.15.1	192.168.15.103	ICMP	Echo (ping) reply
26	28.906559	192.168.15.103	192.168.15.1	ICMP	Echo (ping) request
27	28.908002	192.168.15.1	192.168.15.103	ICMP	Echo (ping) reply
28	29.920400	192.168.15.103	192.168.15.1	ICMP	Echo (ping) request
29	29.921921	192.168.15.1	192.168.15.103	ICMP	Echo (ping) reply

Figure 6-5. ICMP echo conversation

The ARP messages are not specific to the ping program and are generated any time a network device communicates with another when corresponding ARP table entries are absent. Figure 6-6 is an expansion of the echo request and reply messages, and a direct comparison shows their differences and commonalities.

+	Ethernet II, Src: HonHaiPr_90:d5:db (00:22:68:90:d5:db), Dst: Cisco-Li_7f:fb:9d (00:14:bf:7f:fb:9d)
+	Internet Protocol, Src: 192.168.15.103 (192.168.15.103), Dst: 192.168.15.1 (192.168.15.1)
+	Internet Control Message Protocol
	Type: 8 (Echo (ping) request)
	Code: 0 ()
	Checksum: 0x4d42 [correct]
	Identifier: 0x0001
	Sequence number: 25 (0x0019)
+	Data (32 bytes)
	Data: 6162636465666768696A6B6C6D6E6F707172737475767761...
	[Length: 32]
+	Ethernet II, Src: Cisco-Li_7f:fb:9d (00:14:bf:7f:fb:9d), Dst: HonHaiPr_90:d5:db (00:22:68:90:d5:db)
+	Internet Protocol, Src: 192.168.15.1 (192.168.15.1), Dst: 192.168.15.103 (192.168.15.103)
+	Internet Control Message Protocol
	Type: 0 (Echo (ping) reply)
	Code: 0 ()
	Checksum: 0x5542 [correct]
	Identifier: 0x0001
	Sequence number: 25 (0x0019)
+	Data (32 bytes)
	Data: 6162636465666768696A6B6C6D6E6F707172737475767761...
	[Length: 32]

Figure 6-6. ICMP echo request and reply packets

First, the echo request is a type 8, while the echo reply is a type 0. Notice that the identifiers and sequence numbers for these two packets are the same, since they are part of the same conversation. Finally, the data, or payload, is a series of numbers from 61 to 76 repeating. These are the hexadecimal values for the ASCII table characters. As an example, the hexadecimal value 61 corresponds to the base 10 value of 97, which in turn corresponds to a lowercase “a” in the ASCII table. Windows actually sends the letters a through w in the payload. These values will repeat for larger payload sizes. Other devices send different characters. However, with the responder function, whatever is sent is supposed to be returned. The payload decode for this example is shown in Figure 6-7.

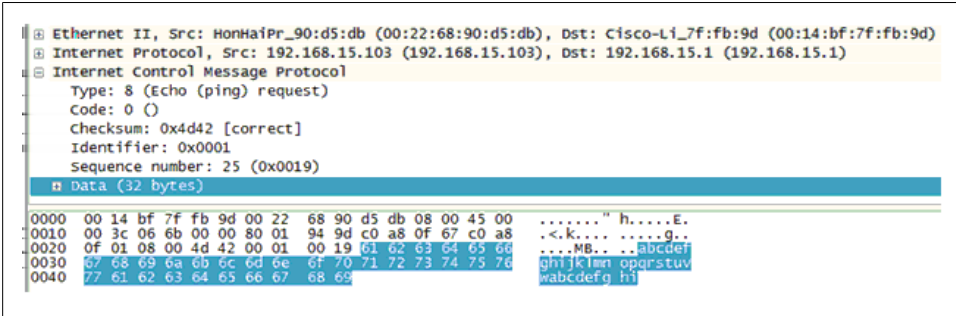


Figure 6-7. ICMP echo payload

Echo fun

Ping is a favored diagnostic tool and is used everywhere. However, using ping while sitting behind a router performing network address translation (NAT) brings up an interesting question. With basic NAT, the router modifies the IP header of outbound packets by changing the source IP address to the outside interface of the router. In effect, all of the traffic appears to have come from the router rather than the original source host. [Figure 6-8](#) depicts a topology in which the middle router is running NAT. All of the traffic from the 192.168.1.0 and 192.168.2.0 networks is translated on the outside interface of this router. If the right side of the NAT router has an IP address of 192.168.3.253, all of the traffic from these two networks will appear to have come from this one IP address. When traffic comes back in response, the router remaps the IP header back to the original IP address and forwards it on.

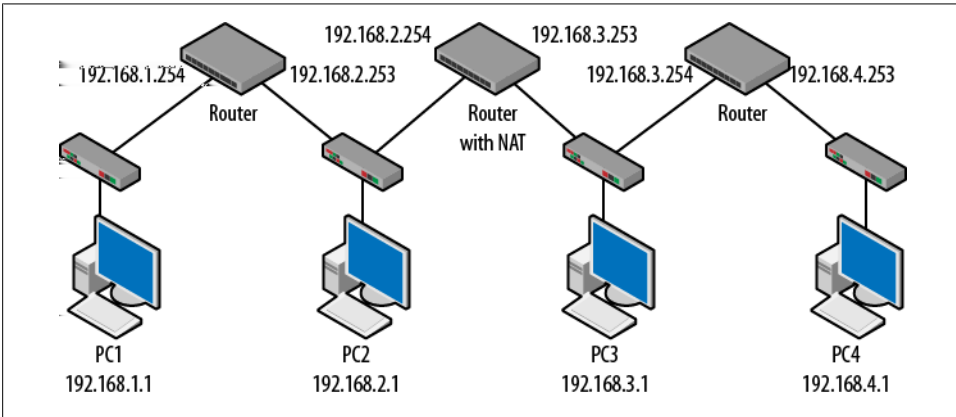


Figure 6-8. Three-router topology with NAT

The usual way to keep track of all these conversations is by the source and destination IP addresses, in addition to the Layer-4 port numbers (TCP and UDP). But ping can be used through a NAT device and, in fact, a node can run several instances of ping at the same time. However, ICMP does not have a Layer-4 header. In other words, the port numbers are missing. If this is true, how are the different ICMP conversations tracked through the NAT router? How do the packets return to the correct source?

The answer lies in the header of the ICMP packet. In this case, we are actually going to use the identifier value along with the IP addresses to keep track of the conversations. An example of an ICMP echo and the corresponding NAT translation table is shown in [Figure 6-8](#).

PC 1 has an IP address of 192.168.1.1 and attempts to ping PC 3, which has an IP address of 192.168.3.1. Since the 2621 is running NAT, it translates this packet so that it appears to have come from 192.168.3.253. The translated ICMP echo request is shown in [Figure 6-9](#).

```
Ethernet II, Src: Cisco_28:1b:e0 (00:05:5e:28:1b:e0), Dst: Standard_08:e0:27 (00:e0:29:08:e0:27)
Internet Protocol, Src: 192.168.3.253 (192.168.3.253), Dst: 192.168.3.1 (192.168.3.1)
Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0 ()
  Checksum: 0x195c [correct]
  Identifier: 0x0400
  Sequence number: 12288 (0x3000)
  Data (32 bytes)
```

Figure 6-9. ICMP echo after translation

The identifier for this packet has a value of 0x0400, which is in hexadecimal, and converting this to base 10 numbers returns 1024. The translation table for the 2621 router is shown in [Figure 6-10](#), where we can see the original source, destination, and the addresses used in the translation. In addition, the identifier value is clearly visible as the replacement for the TCP or UDP port number.

```
Router#
Router#sh ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
icmp 192.168.3.253:1024 192.168.1.1:1024 192.168.3.1:1024 192.168.3.1:1024
Router#
```

Figure 6-10. Cisco translation table

Redirect (Type 5)

The goal of a redirect message is to inform a host that there is a better path to the destination than the one just tried. The diagram below depicts a topology in which a redirect often occurs, but according to RFC 792, a couple of conditions must first be satisfied:

- The new forwarding router and the host identified by the Internet source address of the datagram must be on the same network.
- The datagram in question must not be using the IP source route options and the gateway address is not in the destination address field.

The topology used earlier presents a scenario that will result in a redirect. Instead of NAT, the interesting characteristic is that PC 2 and PC 3 are connected between routers. This means that possible destinations can be found in both directions from these machines. In the case of PC 1 and PC 4, all destinations can be reached by traveling in one direction ([Figure 6-11](#)).

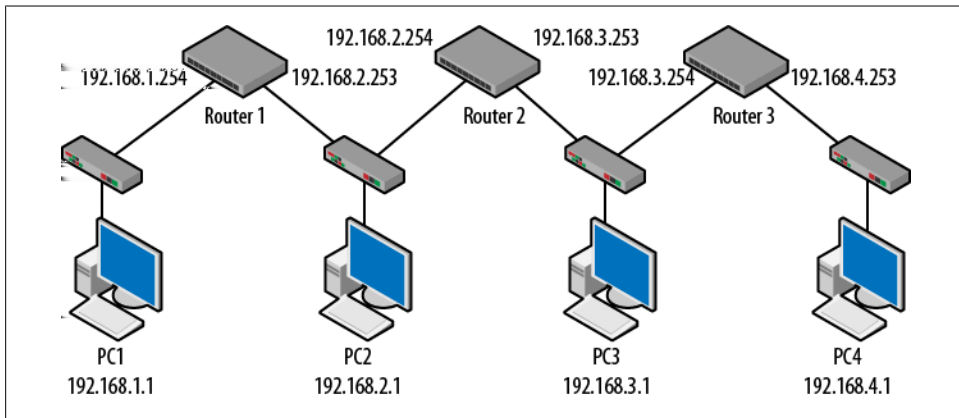


Figure 6-11. Redirect topology

PC 3 on the 192.168.3.0 network is configured with the 2621 as its default gateway. Regardless of the destination, PC 3 will send traffic to this router. However, if PC 3 attempts to connect to PC 4 on the 192.168.4.0 network, the 2621 must send this transmission right back out the same interface it came in.

When the packet arrives at the 2621, the router will consult its routing table and learn that the destination is connected back the other way, via the right 2514 router. Since the source host (PC 3) and left interface of the 2514 are both on the same network, the first condition for a redirect is satisfied.

The IP source route options are used when a host attempts to specify the path a packet will take. In that case, the IP header will expand to include as many as five additional hops or routers. However, using IP source route information is unusual (condition 2),

so the 2621 will typically generate an ICMP redirect message back to PC 3. The redirect message instructs the host to use the 2514 the next time PC 4 or the 192.168.4.0 network is the destination. An example of a redirect message is shown in [Figure 6-12](#).

```
Ethernet II, Src: Cisco_28:1b:e0 (00:05:5e:28:1b:e0), Dst: Standard_08:e0:27 (00:e0:29:08:e0:27)
Internet Protocol, Src: 192.168.3.253 (192.168.3.253), Dst: 192.168.3.1 (192.168.3.1)
Internet Control Message Protocol
  Type: 5 (Redirect)
  Code: 0 (Redirect for network)
  Checksum: 0xe0fc [correct]
  Gateway address: 192.168.3.254 (192.168.3.254)
  Internet Protocol, Src: 192.168.3.1 (192.168.3.1), Dst: 192.168.4.254 (192.168.4.254)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 60
    Identification: 0x02d0 (720)
    Flags: 0x00
    Fragment offset: 0
    Time to live: 127
    Protocol: ICMP (0x01)
    Header checksum: 0xafaf [correct]
    Source: 192.168.3.1 (192.168.3.1)
    Destination: 192.168.4.254 (192.168.4.254)
  Internet Control Message Protocol
```

Figure 6-12. ICMP redirect

There are a few more details to the process that are worth noting. First is that the 2621 router will forward the original packet to the proper destination (192.168.4.254) to avoid losing any packets. The redirect is sent to the source host. Second, once the redirect has been sent to the original source host, the host installs a new local routing table entry so that the next time, it will use the proper router.

A majority of these local routing table updates will be host- or network-specific, and the redirect messages will typically have a code of 1 or 0, as shown in [Figure 6-12](#). This is the default behavior of many routers. RFC 792 includes several different codes to go along with the type 5 message:

- 0—Redirect datagrams for the network
- 1—Redirect datagrams for the host
- 2—Redirect datagrams for the type of service and network
- 3—Redirect datagrams for the type of service and host

A host like PC 3 could have many of these dynamically created routing table entries. A large number of redirects can mean that the network design or location of resources may have to be reviewed. An example of a host routing table with a host specific entry is shown in [Figure 6-13](#).

Most routing table entries are for networks, hosts, or special addresses. In this case, there has been an entry installed for one destination in particular—192.168.4.254. This is the host-specific entry.

```

=====
Active Routes:
Network Destination        Netmask          Gateway          Interface        Metric
0.0.0.0                    0.0.0.0          192.168.3.253    192.168.3.1      1000
127.0.0.0                  255.0.0.0        127.0.0.1       127.0.0.1        1
192.168.3.0                255.255.255.0    192.168.3.1     192.168.3.1      30
192.168.3.1                255.255.255.255  127.0.0.1       127.0.0.1        30
192.168.3.255              255.255.255.255  192.168.3.1     192.168.3.1      30
192.168.4.254              255.255.255.255  192.168.3.254   192.168.3.1      1
224.0.0.0                  240.0.0.0        192.168.3.1     192.168.3.1      30
255.255.255.255            255.255.255.255  192.168.3.1     192.168.3.1      1
Default Gateway:          192.168.3.253
=====

```

Figure 6-13. Host routing table with host-specific route

Finally, an ICMP redirect has a different set of fields than the echo request/reply combination discussed earlier. In addition to the code change, the request also contains the new router address for the specified host (which also appears in the routing table entry), the original IP header, and 64 bits of the payload. The redirect not only informs the source of the new path, but also copies part of the original message. In [Figure 6-12](#), the new gateway (192.168.3.254), the original IP header, and 8 bytes of the ICMP initial echo request are all visible.

Time to Live Exceeded (Type 11)

Every single IP packet has a time to live (TTL) field. This is the number of hops or router interfaces the packet is permitted to traverse before it is removed from the network. When the TTL reaches zero, the packet is removed from the network and an ICMP “time to live exceeded” message is generated, indicating that the packet was dropped. This is sent back to the original source host as feedback. [Figure 6-14](#) shows an ICMP time exceeded message. This is a type 11 code 0 and, aside from adding a checksum, the rest is simply the original IP header and 64 bits of the dropped packet.

```

Ethernet II, Src: Cisco_28:1b:e0 (00:05:5e:28:1b:e0), Dst: Standard_08:e0:27 (00:e0:29:08:e0:27)
Internet Protocol, Src: 192.168.3.253 (192.168.3.253), Dst: 192.168.3.1 (192.168.3.1)
  Version: 4
  Header Length: 20 bytes
  Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00)
  Total Length: 56
  Identification: 0x02db (731)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 255
  Protocol: ICMP (0x01)
  Header checksum: 0x2fdb [correct]
  Source: 192.168.3.253 (192.168.3.253)
  Destination: 192.168.3.1 (192.168.3.1)
Internet Control Message Protocol
  Type: 11 (Time-to-live exceeded)
  Code: 0 (Time to live exceeded in transit)
  Checksum: 0x9fa3 [correct]
  Internet Protocol, Src: 192.168.3.1 (192.168.3.1), Dst: 192.168.1.254 (192.168.1.254)
  Internet Control Message Protocol

```

Figure 6-14. ICMP time exceeded

Time exceeded messages can occur anywhere there is a configuration problem on the network that results in routers trying to forward traffic back and forth or when topologies have loops. One common example occurs when neighboring routers are configured with each other as the forwarding router. A less obvious problem occurs when a link or route goes down, resulting in a path being removed from a routing table. [Figure 6-15](#) shows this topology. Router 2 is using Router 3 as a default route. Both Router 2 and Router 3 believe that the 192.168.1.0 network is available to the left via Router 1.

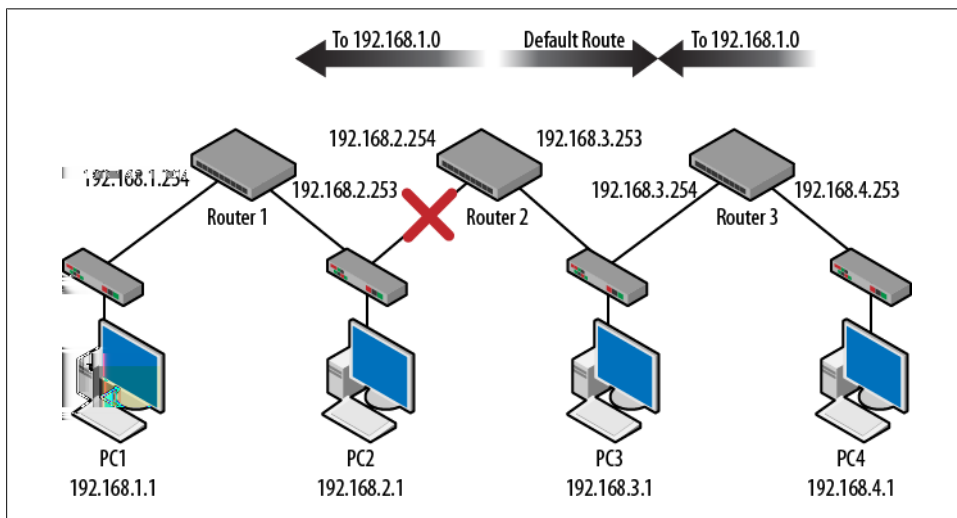


Figure 6-15. Time exceeded topology

During normal operation, the traffic for hosts on the 192.168.1.0 network will be handled by R1. However, if a network is shut down, the behavior of the routers will change dramatically. If the connection between Router 2 and Router 1 is lost, it is common to have Router 2 remove the 192.168.1.0 and the 192.168.2.0 networks from its local routing table. If a network host (192.168.3.1) attempts to contact 192.168.1.1, Router 2 will send this traffic to its default route (Router 3), because this entry no longer exists in the routing table. The problem is that Router 3 is still configured to connect to the 192.168.1.0 network via Router 2 and thus sends the traffic back to Router 2. The process begins again, eventually resulting in an ICMP time exceeded message.

Tracing a Route

You can use time exceeded messages for diagnostic purposes (for example, by using path discovery programs). Tracert is a program built into Windows (tracert for Linux and Cisco) that sends out ICMP messages with the TTL field in the IP header incremented by 1 in subsequent packets. So, the first ICMP echo request for the destination goes out with a TTL of 1. This echo request is repeated three or four times, depending on the operating system. The very first router decrements the TTL field to

0 and returns the ICMP time exceeded message. This informs the sender of the first outbound router interface.

As the transmission moves out from the source host, the IP TTL field goes up by one so that the next router returns the time exceeded message. In this way, we eventually work our way to the destination with the router interfaces that face the source host reporting their presence. An example of the Windows tracert output is shown in [Figure 6-16](#).

```
C:\Documents and Settings\Administrator>tracert 192.168.1.1

Tracing route to 192.168.1.1 over a maximum of 30 hops

  1  *             1 ms    <1 ms  192.168.3.253
  2  1 ms          1 ms    1 ms  192.168.2.253
  3  1 ms          <1 ms   <1 ms  192.168.1.1

Trace complete.
```

Figure 6-16. Windows tracert output

Referring back to the topology in [Figure 6-11](#), a host on the 192.168.4.0 network attempts to discover the path to the 192.168.1.0 network and specifically the host 192.168.1.1. The resulting output displays the router interfaces receiving the ICMP echo request trace packets.

Destination Unreachable (Type 3)

Just as the name suggests, the type 3 message tells a source host that the pathway to the destination is unknown. When a router is missing information that will allow a packet to be forwarded, the feedback provided to the source host is the ICMP destination unreachable. This is a very common occurrence when a router does not have a default route (gateway of last resort), because it will not be able to forward traffic to any network not configured via directly connected, static, or dynamic routes. Like a redirect, there are several codes for the destination unreachable message:

- 0—Net unreachable
- 1—Host unreachable
- 2—Protocol unreachable
- 3—Port unreachable
- 4—Fragmentation needed and DF (do not fragment) set
- 5—Source route failed.

The most common of these messages are the host (code 1) and port (code 3) unreachable. These are shown in [Figure 6-17](#). The messages share the following fields: type, code, checksum, the original IP header, and 64 bits of the data from the original packet. The difference between these two is that the host unreachable is generated because the router does not know the path, while the port unreachable is generated because the


```
[-] Ethernet II, Src: Cisco_23:85:68 (00:19:06:23:85:68), Dst: D-Link_c1:d2:01 (00:50:ba:c1:d2:01)
[-] Internet Protocol, Src: 192.168.10.254 (192.168.10.254), Dst: 192.168.10.11 (192.168.10.11)
[-] Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 1 (Host unreachable)
    Checksum: 0xa7a2 [correct]
[-] Internet Protocol, Src: 192.168.10.11 (192.168.10.11), Dst: 129.21.21.1 (129.21.21.1)
[-] Internet Control Message Protocol

[-] Ethernet II, Src: Cisco-Li_7f:fb:9d (00:14:bf:7f:fb:9d), Dst: HonHaiPr_90:d5:db (00:22:68:90:d5:db)
[-] Internet Protocol, Src: 10.241.128.1 (10.241.128.1), Dst: 192.168.15.103 (192.168.15.103)
[-] Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 3 (Port unreachable)
    Checksum: 0x50f5 [correct]
[-] Internet Protocol, Src: 192.168.15.103 (192.168.15.103), Dst: 10.241.128.1 (10.241.128.1)
[-] User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137)
```

Figure 6-17. ICMP destination unreachable types

service is not available via the router. Incidentally, a router or firewall can generate type 3 with a code of 13, which means that the packet has been administratively filtered or actively blocked.

Operating system vs. ICMP

There is another situation in which the phrase “destination unreachable” is used. When a bad or missing default gateway is configured on a network host, the operating system returns this text to the user. This informs the user that the host (not the router) does not know a path to the destination. This output, and therefore its value to us in troubleshooting, is quite a bit different from the ICMP message.

Pinging a destination that is unknown to the router will result in an ICMP destination unreachable and the output in the command window will look like that shown in Figure 6-18. However, if the host does not have a default gateway and we attempt to transmit to a host on another network, the output changes to that shown in Figure 6-19. In the latter case, no packets are transmitted at all—the “destination unreachable” message comes from the operating system. The value in these different messages is that they both provide vital information about the source of the problem, and the sources couldn’t be more different.

Another interesting point is that while Windows is reporting that four packets were sent, no network traffic is generated as a result of this request, because the destination is off the network and the host is missing a default gateway.

```
C:\Documents and Settings\Administrator>ping 192.168.5.1
Pinging 192.168.5.1 with 32 bytes of data:
Reply from 192.168.3.253: Destination host unreachable.
Reply from 192.168.3.253: Destination host unreachable.
```

Figure 6-18. ICMP destination unreachable command output

```

C:\Documents and Settings\Administrator>ping 192.168.5.1

Pinging 192.168.5.1 with 32 bytes of data:

Destination host unreachable.
Destination host unreachable.
Destination host unreachable.
Destination host unreachable.

Ping statistics for 192.168.5.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

Figure 6-19. Command shell output from a missing gateway

Router Solicitation (Type 10) and Router Advertisements (Type 9)

Type 10 and type 9 messages are not as common as they once were, because they have been largely supplanted by the dynamic host configuration protocol or DHCP. Their purpose is to provide or request information regarding the routers on the LAN. If a host has an IP address but does not have a default gateway, it can ask the network for an answer by sending out an ICMP router solicitation (type 10), as shown in [Figure 6-20](#). Notice that the Layer-3 destination address is the all routers multicast of 224.0.0.2, and Layer-2 addressing is also multicast.

```

Ethernet II, Src: Standard_08:e0:27 (00:e0:29:08:e0:27), Dst: IPv4mcast_00:00:02 (01:00:5e:00:00:02)
Internet Protocol, Src: 192.168.3.1 (192.168.3.1), Dst: 224.0.0.2 (224.0.0.2)
Internet Control Message Protocol
  Type: 10 (Router solicitation)
  Code: 0 ()
  Checksum: 0xf5ff [correct]

```

Figure 6-20. ICMP router solicitation

This type of ICMP message is also very small and requires extra padding at the end to reach the minimum Ethernet frame size.

Routers periodically announce themselves or answer solicitations by sending out router advertisements like the one shown in [Figure 6-21](#). In this case, the broadcast address is used. While a unicast address works well when answering a particular host, a broadcast is best for the general advertisement. This address is matched at Layer 2.

With the widespread use of DHCP, there is no reason to ask about or receive router information if the default gateway was given along with the IP address. However, router solicitations and advertisements still have use in some wireless applications. In the case of a wireless node roaming from one network to another, the MobileIP architecture still uses these messages.

A wireless node traveling from its home network to another while utilizing MobileIP does not get a new IP address via DHCP when it arrives. Instead, it contacts a device called a *foreign agent*. This foreign agent acts as a proxy for the visiting node in that it

```

Ethernet II, Src: Cisco_28:1b:e1 (00:05:5e:28:1b:e1), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol, Src: 192.168.2.254 (192.168.2.254), Dst: 255.255.255.255 (255.255.255.255)
Internet Control Message Protocol
  Type: 9 (Mobile IP Advertisement)
  Code: 0 ()
  Checksum: 0x2b4f [correct]
  Number of addresses: 1
  Address entry size: 2
  Lifetime: 30 minutes
  Router address: 192.168.2.254
  Preference level: 0

```

Figure 6-21. ICMP router advertisement

forwards all transmissions to and from the visiting host. The visiting host and the foreign agent find each other via ICMP router advertisement and solicitation. The messages are modified or extended to include the additional MobileIP-specific information. However, since most organizations do not permit roaming in and out of their networks, ICMP type 9 and 10 messages are not likely to make a big comeback. IPv6 also makes use of these message types, albeit in the ICMPv6 format.

Digging a Little Deeper—the One’s Complement

Both the checksum in the IP header and the ICMP message are calculated using the one’s complement of the one’s complement sum of the 16-bit words in either the header or the message itself. This means that the calculation called the one’s complement is completed, then the complement, or inverse, is then used as the checksum value.

For example, the following 16-bit binary streams are added together. The leading zeroes are placeholders.

0000	1001	1010	1110	1010
<u>0000</u>	<u>1011</u>	<u>0100</u>	<u>0011</u>	<u>0001</u>
0001	0100	1111	0001	1011

As you can see, a 1 is carried over to the 17th bit. With the checksum one’s complement, this carryover is added back in, as follows:

0000	1001	1010	1110	1010
<u>0000</u>	<u>1011</u>	<u>0100</u>	<u>0011</u>	<u>0001</u>
0000	0100	1111	0001	1011
				<u>0001</u>
	0100	1111	0001	1100

Now that the sum (the one's complement) has been completed, the complement or inverse is taken.

1011 0000 1110 0011

This is the value that is actually used for the checksum, as seen in the ICMP packets—the 16-bit one's complement of the one's complement sum of the ICMP datagram.

IPv6

ICMP will continue to be of use as networks migrate to IPv6. The structure and many of the message types are similar to those used in IPv4. However, ICMPv6 is governed by the rules established in RFC 4443. One change is that the IP header *next header* field changes from a value of 1 to 58. Messages will also continue to be categorized as either informational or error and, where appropriate, will contain some of the original message. The major types of ICMPv6 messages are:

- Type 1—Destination unreachable
- Type 2—Packet too big
- Type 3—Time exceeded
- Type 4—Parameter problem
- Type 128—Echo request
- Type 129—Echo reply

As you can see, many of these messages are performing similar functions to the IPv4 ICMP messages. There have also been a couple of new additions. These newer message types have more to do with changes to the operation of IPv6 than changes to ICMP.

Neighbor discovery is an example of an operational change that has significant impact on ICMP. IPv6 nodes take a very active role in learning about their local topology. During this process, the nodes seek out the Link Layer addresses of neighbors and routers willing to forward traffic, and will age out old information. There are several ICMPv6 message types involved, including router advertisement and solicitation, neighbor advertisement, and solicitation and redirects. Recall that ARP is not a part of IPv6. Some ICMPv6 examples are shown in [Figure 6-22](#), which includes an ICMPv6 router solicitation with the IPv6 header expanded. Notice the *next header* field.

The ICMPv6 echo request and neighbor solicitation messages are shown in [Figure 6-23](#) and [Figure 6-24](#). The echo request uses the standard addressing at Layer 2, but the neighbor solicitation uses an address reserved for IPv6 multicast.

```

Ethernet II, Src: HonHaiPr_12:1c:a9 (00:1f:e2:12:1c:a9), Dst: IPv6mcast_00:00:00:02 (33:33:00:00:00:02)
Internet Protocol Version 6
  0110 .... = Version: 6
  .... 0000 0000 .... = Traffic class: 0x00000000
  .... 0000 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
  Payload length: 16
  Next header: ICMPv6 (0x3a)
  Hop limit: 255
  Source: fe80::f077:1f25:cf06:ec54 (fe80::f077:1f25:cf06:ec54)
  Destination: ff02::2 (ff02::2)
Internet Control Message Protocol v6
  Type: 133 (Router solicitation)
  Code: 0
  Checksum: 0xb25a [correct]
  ICMPv6 option (Source link-layer address)
    Type: Source link-layer address (1)
    Length: 8
    Link-layer address: 00:1f:e2:12:1c:a9

```

Figure 6-22. ICMPv6 router solicitation

```

Frame 7 (94 bytes on wire, 94 bytes captured)
Ethernet II, Src: Intel_c8:b9:a0 (00:0c:f1:c8:b9:a0), Dst: Intel_c8:b4:1e (00:0c:f1:c8:b4:1e)
Internet Protocol Version 6
  Internet Control Message Protocol v6
    Type: 128 (Echo request)
    Code: 0
    Checksum: 0x8489 [correct]
    ID: 0x0000
    Sequence: 0x0005
  Data (32 bytes)
    Data: 6162636465666768696a6b6c6d6e6f7071727374757677681...
    [Length: 32]

```

Figure 6-23. ICMPv6 echo request

```

Frame 5 (86 bytes on wire, 86 bytes captured)
Ethernet II, Src: Intel_c8:b9:a0 (00:0c:f1:c8:b9:a0), Dst: IPv6mcast_ff:c8:b4:1e (33:33:ff:c8:b4:1e)
Internet Protocol Version 6
  Internet Control Message Protocol v6
    Type: 135 (Neighbor solicitation)
    Code: 0
    Checksum: 0xc8d7 [correct]
    Target: fe80::20c:f1ff:fec8:b41e (fe80::20c:f1ff:fec8:b41e)
  ICMPv6 option (Source link-layer address)
    Type: Source link-layer address (1)
    Length: 8
    Link-layer address: 00:0c:f1:c8:b9:a0

```

Figure 6-24. ICMPv6 neighbor solicitation

Summary

There are many types of ICMP messages defined in the RFCs, but only a subset of these is used with any regularity. On most networks, the ICMP echo request, echo reply, time exceeded, destination unreachable, and redirect messages are common. ICMP is broken down into informational and error messages. In the case of messages generated for feedback, a portion of the original IP packet is included. Several diagnostic programs such as ping and traceroute make use of ICMP for path discovery and troubleshooting. IPv6 extends the life of ICMP and makes use of the less-common router solicitation and advertisement in a process called neighbor discovery.

Additional Reading

RFC 792: “Internet Control Message Protocol”

RFC 1256: “ICMP Router Discovery Messages”

RFC 2461: “IPv6 Neighbor Discovery”

RFC 4443: “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification”

Review Questions

1. What is the payload of an ICMP echo request from Windows?
2. Provide two examples of ICMP messages that include 64 bits of the original message.
3. What is the “all routers multicast” address?
4. What is the IP “next header” or protocol value for ICMP?
5. Router solicitations and advertisements are not very common. What protocol is used instead?
6. What is the IPv6 process that replaces ARP and uses several different types of ICMPv6 messages?
7. What are the type and code values of the ICMP network destination unreachable message?
8. What event causes an ICMP time exceeded message to be generated?
9. Instead of a TCP or UDP port number, what value is used when forwarding ICMP packets through a NAT router?
10. On the receiving host, what is the effect of an ICMP redirect message?

Review Answers

1. The alphabet
2. Redirect, destination unreachable, time exceeded
3. 224.0.0.2
4. 01
5. DHCP
6. Neighbor discovery
7. 3 and 0
8. The IP TTL field is decremented to 0
9. The ICMP identifier
10. A route is added to the local host routing table for use in subsequent transmissions

Lab Activities

Activity 1—Ping

Materials: A computer with a connection to another network device or node

1. On the computer, start a Wireshark capture.
2. Open a command window or shell.
3. Using the `ping` command, ping the other device. For example, ping **192.168.1.1**.
4. Take a look at the ICMP packets generated as a result of this ping. What are the type, code, and payload of these messages?

Activity 2—Tracert

Materials: A computer with a connection to the Internet or routed topology

1. On the computer, start a Wireshark capture.
2. Open a command window or shell.
3. Using the `tracert` command, perform path discovery to another node or a website. For example, `tracert www.google.com`.
4. What messages are generated as a result of this command?
5. Examine the TTL field of the IP packets and determine exactly what is happening.
6. What ICMP message results from the exchange?
7. What other packets are generated as a result of using a name instead of an IP address?

Activity 3—Start Up Packet Capture

Materials: Two computers, Wireshark, and a connection to a router

1. To start, have one computer up and running and the other shut down.
2. Connect the router and both computers together. If using a home gateway, use the ports on the switch module. If not, another device such as a hub or switch may be required.
3. Start a Wireshark capture on the running machine.
4. Start up the second machine and observe the packets generated.
5. This particular exercise is concerned with the ICMP messages that are generated however the other types of traffic are very interesting as well. See if you can determine the reasons for all of the traffic present. Depending on the operating system, you will see a variety of packets including IPv4 and IPv6 ICMP messages.

Activity 4—Destination Unreachable From the OS

Materials: A computer with an active connection

1. Configure this computer with an IP address, but leave the default gateway line blank.
2. On the computer, start a Wireshark capture.
3. Open a command window or shell.
4. Ping a device or address that is not on your network.
5. What was the response? Where did this message come from?
6. Were there any packets generated as a result of this ping?

Activity 5—Destination Unreachable From the Router

Materials: A computer with an active connection to a router or home gateway

1. Configure the computer with an IP address, but this time, give the node a gateway.
2. Remove the outside connection to the home gateway.
3. On the computer, start a Wireshark capture.
4. Open a command window or shell.
5. Ping a device or address that is not on your network.
6. What was the response? Where did this message come from?
7. Were there any packets generated as a result of this ping?

Subnetting and Other Masking Acrobatics

“While these problems could be avoided by attempting to restrict the growth of the Internet, most people would prefer solutions that allow growth to continue. Fortunately, it appears that such solutions are possible, and that, in fact, our biggest problem is having too many possible solutions rather than too few.”

—RFC1380

A network can be defined in many ways. From a Layer-3 perspective, a network is a group of nodes that all share the same IP addressing scheme. The original vision for the IP-based Internet was a two-tier system in which a collection of networks were all connected to a single Internet or catenet. Confusion arises because it can be difficult to tell what the network boundaries are. The answer, and perhaps the source of the confusion, lies in the network mask. Many networking decisions are made based on the mask—host and router routing, classful and classless address space, security, QoS provisioning, and the overall design are all affected by the masks applied to the nodes.

A device operating on a network requires four numbers to ensure basic connectivity: IP address, network mask, gateway, and the DNS address. Their purpose is straightforward. IP addresses provide logical location, masks determine the network, the gateway is a router providing a pathway off of the current network, and the domain name server converts between IP addresses and more human-friendly addresses/words such as those used in web pages. The focus of this chapter will be the network mask and the corresponding network.

How Do We Use the Mask?

The early IP network address assignments were based on the three main classes—A, B, and C—that varied in size, and this size was based on the mask. Each class had a normal or natural mask, a specific number of hosts, and a range, as shown in [Table 7-1](#).

Table 7-1. IP classes and masks

Class	Address range	Mask	Number of possible networks	Number of possible hosts
A	0–127	255.0.0.0	128	16,777,216
B	128–191	255.255.0.0	16,364	65,536
C	192–223	255.255.255.0	2,097,152	256

Many of these addresses are not actually valid on the public network (e.g., 127 is used for loopback), but generally, a network can be identified by looking at its IP address and the corresponding mask. An IP address beginning with 36 in the first octet (e.g., 36.45.197.223) resides on a class A network, regardless of the rest of the address. [Table 7-1](#) shows that class A networks are few in number, but each contains millions of hosts. Class A networks also have a default mask of 255.0.0.0.

When a node is assigned an IP address, the IP address, when combined with the mask, actually includes not only the address for the host, but information about the network as well. The method used to determine the network address from the host address using the mask is a logical AND operation. This is shown in [Figure 7-1](#).

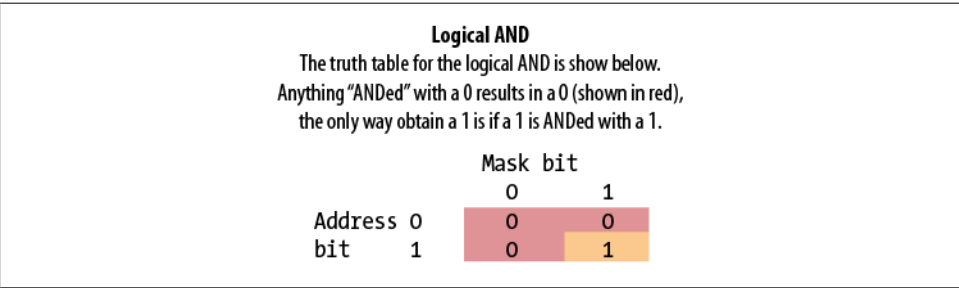


Figure 7-1. Logical AND

An organization wishing to connect to others via TCP/IP will be assigned a particular network based on this class structure. Smaller organizations will be given a class C network, while much larger ones might be given a class B or even a class A. All of the nodes within the organization will be given IP addresses within the same scheme and all of them will have the same mask. Assigning addresses based on these classes is called *classful addressing*.

For example, if a smaller organization is given a class C network address of 200.150.100.0, it will use a mask of 255.255.255.0. The 0 in the last octet of the IP address is significant. Since the possible values for any octet in an IP address is from 0 (binary 00000000) to 255 (binary 11111111), the range of possible addresses for this particular network is 200.150.100.0 to 200.150.100.255.

All of the hosts on this network will have this same network address, and this is determined by the mask. The following calculations show how this is determined (assume Host A has IP address 200.150.100.95 and mask 255.255.255.0):

1. Convert the host address to binary:
11001000.10010110.01100100.01011111
2. Convert the class mask to binary
11111111.11111111.11111111.00000000
3. Perform a bitwise AND using the host address and the mask to get the network address.
11001000.10010110.01100100.01011111
11111111.11111111.11111111.00000000
11001000.10010110.01100100.00000000
4. Convert back to base 10 numbers.
200.150.100.0

The last octet is converted to all 0s as a result of the logical AND. Any host address in the range described above will result in the same network address. *Hosts are not assigned this particular address*. When the mask octet value is 255, the IP address value is simply brought down to the result as was the case for the 200, 150, and 100. This process is required, because IP packets do not include any information regarding the network itself. [Figure 7-2](#) shows a standard IP packet—notice that the mask is not even included.

Within the network mask, there are actually two components. The binary ones (1) indicate the network portion and the zeroes (0) indicate the host portion. In this case, the network portion has been allocated three bytes. The host portion has been allocated a single byte of address space, as shown in [Figure 7-3](#).

```
Ethernet II, Src: Pentacom_5a:94:00 (00:d0:04:5a:94:00), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  Destination: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  Source: Pentacom_5a:94:00 (00:d0:04:5a:94:00)
  Type: IP (0x0800)
Internet Protocol, Src: 129.21.185.28 (129.21.185.28), Dst: 239.255.255.250 (239.255.255.250)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  Total Length: 299
  Identification: 0xf2f6 (62198)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 3
  Protocol: UDP (17)
  Header checksum: 0x999f [correct]
  Source: 129.21.185.28 (129.21.185.28)
  Destination: 239.255.255.250 (239.255.255.250)
User Datagram Protocol, Src Port: ssdp (1900), Dst Port: ssdp (1900)
  Source port: ssdp (1900)
  Destination port: ssdp (1900)
  Length: 279
  Checksum: 0x0b26 [validation disabled]
Hypertext Transfer Protocol
  NOTIFY * HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): NOTIFY * HTTP/1.1\r\n]
    Request Method: NOTIFY
    Request URI: *
    Request Version: HTTP/1.1
    LOCATION: http://129.21.185.28:8089/\r\n
    HOST: 239.255.255.250:1900\r\n
    SERVER: POSIX, UPnP/1.0, Intel MicroStack/1.0.1347\r\n
    NTS: ssdp:alive\r\n
    USN: uuid:1c852d10-b80b-1f08-98c5-02bad0d9b366::upnp:rootdevice\r\n
    CACHE-CONTROL: max-age=1800\r\n
    NT: upnp:rootdevice\r\n
    \r\n
```

Figure 7-2. Packet with all headers expanded

Binary	11111111.	11111111.	11111111.	0
Base 10	255.	255.	255.	0
Network portion				Host portion

Figure 7-3. Network mask sections

Now that the binary digits have been exposed, another way of describing the mask is to count the number of 1s. Thus, class A networks have an 8-bit mask, class B networks have a 16-bit mask, and class C networks have a 24-bit mask.

Another special address for this particular network (in addition to the network) is 200.150.100.255, which is called the *directed broadcast address*. It is used to reach all hosts on the network, so it must not be assigned to any particular host. The network address contains all 0s in the host portion of the IP address. The directed broadcast address contains all 1s in the host portion of the IP address:

Network address	11001000.10100000.01100100. 00000000 (200.150.100.0)
Directed broadcast address	11001000.10100000.01100100. 11111111 (200.150.100.255)

Summarizing for this class C network:

Network address or ID:	200.150.100.0
Network mask:	255.255.255.0
Directed broadcast address:	200.150.100.255
Possible address space:	256 (200.150.100.0–200.150.100.255)
Useable address space:	254 (200.150.100.1–200.150.100.254)

Hosts in this network, such as computers, routers, and printers will use addresses between 200.150.100.1 and 200.150.100.254. In our exercise, the organization is using all of the IP addresses within the class C network space and has not manipulated the mask, thus using classful addressing.

The nodes on a network typically use the same router to forward packets externally, and that router must also use an IP address on the network. Routers commonly use an address that is on either the low end or the high end of the range, such as 200.150.100.1 or 200.150.100.254, which cannot be given to computers or other network devices. [Figure 7-4](#) shows what this network might look like. Each host uses the router as the default gateway for packets exiting the network and the router itself has an additional address associated with the outside connection.

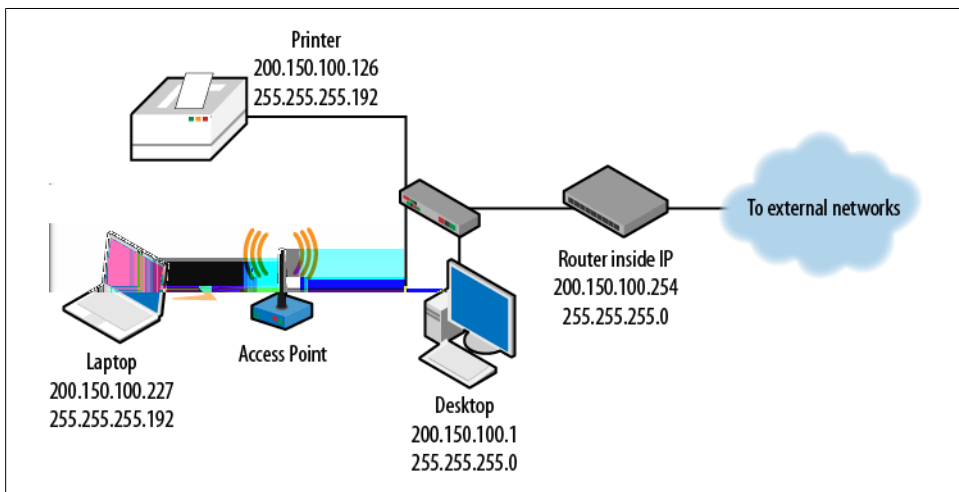


Figure 7-4. Network addressing

What Is a Subnet?

Subnets work exactly like classful networks in that they require a router to get to other networks and have a network address, a directed broadcast address, and a specific set of hosts. To quote from RFC 917:

We discuss the utility of “subnets” of Internet networks, which are logically visible sub-sections of a single Internet network. For administrative or technical reasons, many organizations have chosen to divide one Internet network into several subnets, instead of acquiring a set of Internet network numbers.

Subnets are created by manipulating the mask of the classful address space and are often utilized to create separation between departments or interconnect different LAN technologies or device types. For example, if the organization shown in [Figure 7-4](#) has a security policy requiring that the traffic from departmental nodes be isolated, a subnetting plan might be very useful. The previous address space consisting of 256 (0–255) possible addresses must be broken up into smaller networks or *subnetworks*.

When subnetting is implemented, there are changes to the mask that will now be referred to as a subnet mask (netmask). The last nonzero octet in the mask will no longer be the friendly 255, which means that the results of the ANDing process are not as easy to predict. The mask now has a subnet field in addition to the network and host portions. Subnetting slightly modifies the ANDing process outlined above.

To create the new netmask, the number of desired subnets must be determined. Assume four subnets will be created for different departments. Each subnet will be smaller than the classful address space, so the number of bits allocated to hosts in each subnet will be fewer. For this reason, bits used to describe the subnets are “stolen” from the host address space. *The number of bits stolen is determined by the number of subnets required.* These stolen bits also become the subnet field in the mask. This creates the new mask that all nodes within the subnetted classful address space will use. What follows are the binary and base 10 values for the new subnet mask achieved as a result of stealing the two bits:

Binary	11111111.	11111111.	11111111.	11000000
New mask	255.	255.	255.	192

Subnet stealing is accomplished by changing the 0 to a 1 as you move from left to right in the mask. This change impacts the ANDing process results. Instead of returning a 0, the ANDing process will not accept whatever is in the IP address for these two bits. Stated another way, the ANDing process pays attention to these two values instead of overwriting them. The change in these bits also changes the IP addresses of the networks (subnets) and inserts the subnet field. The subnet field will use the different binary patterns offered by the two bits: 00, 01, 10, and 11. These changes are shown in [Table 7-2](#).

Table 7-2. Subnet mask patterns and allocations

Class A			Class B			Class C		
Mask	Subnets	Hosts	Mask	Subnets	Hosts	Mask	Subnets	Hosts
255.0.0.0	1	16777216	255.255.0.0	1	65536	255.255.255.0	1	256
255.128.0.0	2	8388608	255.255.128.0	2	32768	255.255.255.12	2	128
255.192.0.0	4	4194304	255.255.192.0	4	16384	255.255.255.19	4	64
255.224.0.0	8	2097152	255.255.224.0	8	8192	255.255.255.22	8	32
255.240.0.0	16	1048576	255.255.240.0	16	4096	255.255.255.24	16	16
255.248.0.0	32	524288	255.255.248.0	32	2048	255.255.255.24	32	8
255.252.0.0	64	262144	255.255.252.0	64	1024	255.255.255.25	64	4
255.254.0.0	128	131072	255.255.254.0	128	512	255.255.255.25	128	2
255.255.0.0	256	65536	255.255.255.0	256	256	255.255.255.25	256	0

The first entry in the table is for the classful address space and is not really a subnet.

Subnet Patterns

Regardless of which octet has been changed, subnet masks all use the same collection of values. Familiarity with these patterns makes subnetting and the manipulation of the address space much easier. [Table 7-2](#) shows these patterns and the number of subnets and hosts that are created. The first row is for the classful address space.

Here are some items to take note of in [Table 7-2](#):

- This is a table of possible values, not useable ones. For example, it doesn't make much sense to create 256 subnets in a class C address space.
- The masks use the same values, but in different octets, so the sizes of subnets vary between the classes.
- Multiplying the maximum number of hosts in each subnet by the number of subnets created will always result in the total number of hosts in the classful address space.
- The number of subnets and the number of hosts will always be a power of 2.
- The mask values result from stealing successive bits on the right side of the mask. For example, 100000000 = 128, 110000000 = 192, 111000000 = 224, 111100000 = 240, etc.)

Due to these changes, hosts that were originally on the same classful network will now reside on different subnetworks. Subnets behave just like classful networks and have the same requirements. To determine the new addressing, the subnet field is applied to the IP addresses, but use the combinations offered by the binary patterns.

Subnet IP Addressing

As stated earlier, subnetting steals bits from the host portion of the address space and the bits stolen are described by changes in the subnet mask. Once the size and location (which octet) of the subnet field are known, all that remains is the starting point. The first subnet always has the same address as the classful address space, but it will be smaller. Using two stolen bits as an example, [Figure 7-5](#) depicts the subnets, the subnet field, and the new ranges.

Network Address		Directed Broadcast		Subnet Range
	Binary Pattern		Binary Pattern	
200.150.100.	00 000000	200.150.100.	00 111111	200.150.100.0-63
200.150.100.	01 000000	200.150.100.	01 111111	200.150.100.64-127
200.150.100.	10 000000	200.150.100.	10 111111	200.150.100.128-191
200.150.100.	11 000000	200.150.100.	11 111111	200.150.100.192-255

Subnet Field

Figure 7-5. Subnet field with binary patterns

For simplicity’s sake, the first three octets will not be converted to binary, but the last octet will be expanded to show the effect of the subnet field. Changes to the mask indicate the location of the subnet field as shown in [Figure 7-6](#). Once this is defined, we simply insert the different binary possibilities, as shown in [Figure 7-5](#). Like classful networks, the subnet address places all 0s in the host portion. The directed broadcast address places all 1s in the host portion. These two values provide the subnet range.

200.150.100.	00	000000
255.255.255.	11	000000

Figure 7-6. Masking in the subnet field

A check of the network ID using the previous process would work in the same way, but the results may be quite a bit different. The network for the same node is now 200.150.100.64, calculated as follows, assuming Host A has IP address 200.150.100.95 and mask 255.255.255.192:

1. Convert the host address to binary.
- 11001000.10010110.01100100.01011111
2. Convert the subnet mask to binary.
- 11111111.11111111.11111111.11000000

3. Perform a bitwise AND using the host address and the mask to get the network address.

11001000.10010110.01100100.01011111

11111111.11111111.11111111.11000000

11001000.10010110.01100100.01000000

4. Convert back to base 10 numbers.

200.150.100.64

Checking this value against the range for the subnet in [Figure 7-5](#) proves this to be the correct answer. [Figure 7-7](#) shows what a network topology might look like this set of subnets. The central node is a router that now has additional interfaces.

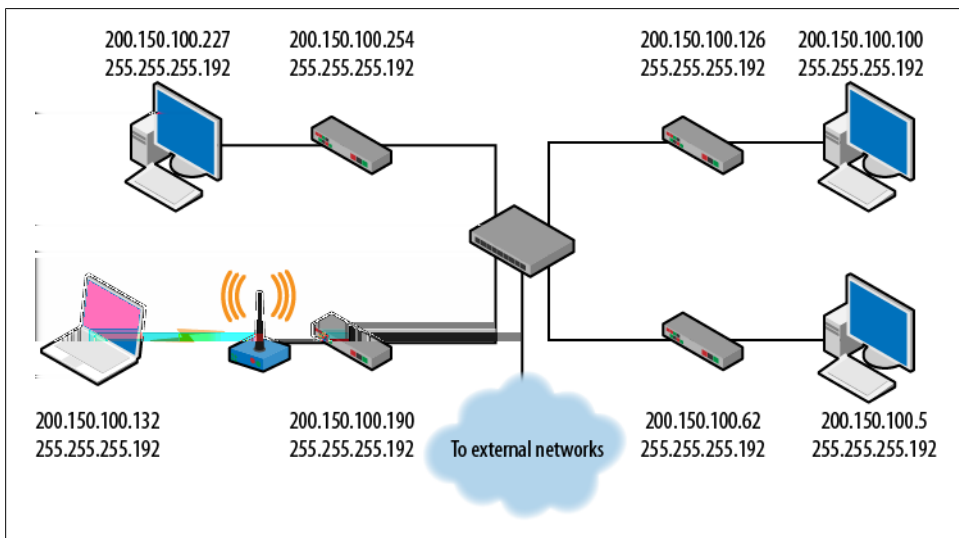


Figure 7-7. Subnetted topology

A Shorthand Technique

For a subnetting problem as straightforward as this one, there is another technique you can use to determine the addresses and ranges for each subnet. For this problem, the only given information is the classful address space (200.150.100.0 255.255.255.0) and the fact that four subnets were required. This means the 256 addresses will be divided into four equal parts of 64 ($256/4 = 64$).

This simple calculation and some basic understanding of the structure provide the basis for all of the subnets. Since the first subnet is always the same as the classful address space, the next step is simply to start counting at 0 to get the first set of 64 and then add one for the next range until the end of the address space is reached:

1. 200.150.100.0-63

This gives the first set of 64.

2. Adding 1 ($63+1=64$) gives the start of the next subnet. At this point, the range expands to a total of 64 addresses. Intuitively, counting now starts at 64, so adding 63 provides the next range ($64+63=127$).

200.150.100.64-127

3. This process continues ($127+1=128$, $127+63=191$) until the end value of 255.

200.150.100.128-191

4. Finally, $191+1=192$ and $191+63=255$.

200.150.100.192-255



Be careful when using this shorthand technique. It works well with simpler problems, but for more complex subnetting (or supernetting) schemes, it is easy to lose your way. When in doubt, go back to the binary method.

The Effect on Address Space

In the classful example, the number of addresses that are unusable by hosts is two, one for the network itself and one for the directed broadcast address. To be complete, it could be said that the router also requires an address, leaving 253 of 256 addresses for networked computers, printers, etc. in that class C network. With subnetting, the number of addresses lost this way is greater, because each subnet has the same requirements. [Figure 7-5](#) shows that creating four subnets loses eight addresses—four to the network and four to the directed broadcast. If you include the router interfaces, of the original 256, only 244 host addresses are available. Pushing this a little farther, creating 32 subnets in this address space results in a loss of more than 25 percent of the address space, without even counting the routers.

Theory vs. Reality

[Table 7-3](#) indicates what can happen if we follow the traditional letter of the law when creating subnets. Again, we'll assume that we need four subnets. Examining the classful address space of 200.150.100.0 (mask of 255.255.255.0) and the subnetted address space of 200.150.100.0 (mask of 255.255.255.192), you can see that the first subnet has the same address as the classful address space. Based on the binary values, this is sometimes called the *all 0s* subnet. In addition, the directed broadcast for the last subnet has the same address (200.150.100.255) as the classful address space, and is often called the *all 1s* subnet. Since masks are not present in packets traveling on the network, these two addresses can lead to confusion with routing tables and design of the network, because the subnet addressing is not distinct from the classful addressing. As a result,

some documentation recommends that network administrators refrain from using the subnets that include these addresses—namely, the lowest and highest subnets. In fact, RFC 950, which attempts to standardize subnetting procedures, states,

This means the values of all zeros and all ones in the SubnetField should not be assigned to actual (physical) subnets.

In the four subnet example, eliminating these two subnets will result in a 50 percent loss of address space because the addresses 200.150.100.0-63 and 200.150.100.192-255 will no longer be available. Due to the loss of the subnet and directed broadcast addresses from the remaining subnets, the IP addresses available for network hosts is actually less than 50 percent. Things can get even worse in terms of efficiency. The traditional model requires that more subnets be created in order to obtain the correct number of useable subnets. This is shown in [Table 7-3](#).

Table 7-3. Subnet address efficiency

Subnet range	Result	Addresses lost due to subnets
200.150.11.0–200.150.100.31	Not allowed	32
200.150.100.32–200.150.100.63	Used	2
200.150.100.64–200.150.100.95	Used	2
200.150.100.96–200.150.100.127	Used	2
200.150.100.128–200.150.100.159	Used	2
200.150.100.160–200.150.100.191	Unused	32
200.150.100.192–200.150.100.223	Unused	32
200.150.100.224–200.150.100.255	Not allowed	32

If, as in this example, four subnets are desired, but the guidelines prohibit the use of the all 0s and all 1s subnets, stealing two bits no longer provides four subnets, but two. To obtain four useable subnets, three bits have to be stolen, which will result in eight possible and six useable subnets. If only four are used, and with the effect of subnets on address space, less than 47 percent of the address space is used. This approach is suboptimal at best.

So, from a practical perspective, this loss of address space makes this practice unpalatable for many veterans in the field and has forced changes to vendor networking equipment and the practices. The all 0s and all 1s subnets are no longer off limits.

Supernetting

Supernetting is defined in RFC 1338 and works in an opposite manner to subnetting. This process combines chunks of address space together. A large number of nodes may be grouped together because they are not simultaneously active, network load is small, or out of a desire for route aggregation. In terms of the masks used, the process is very

similar, but instead of stealing bits from the host portion, bits are stolen from the network portion of the address. However, when stealing from the network portion, the opposite process takes place. Instead of converting the bits in the mask from 0 to 1, the bits are converted from 1 to 0. The effect is that the ANDing process no longer accepts the address information from the IP address, because more of the information will be converted to 0. The key is that in order to supernet networks together, the remaining binary patterns in the network portion of the address must be the same.

Using the same class C address space as the previous section (200.150.100.0, mask 255.255.255.0), this example will now supernet eight networks together. Eight networks will require changing three bits in the mask. [Figure 7-8](#) will help you determine the appropriate mask, the new network address and the range of hosts in the network. For simplicity, the table will only convert the octets affected by the mask change and the last octet of all 0's will be removed. Columns three and four are both part of the third octet. Like the subnet field, a supernet field will be created in the mask.

Supernet Field
Mask: 11111111 .11111111 .11111 000 .00000000

Network Address	First and Second Octets	Unchanged by Mask	Bits Stolen 1 2 3	1 2 3
200.150.96.0	200.150	011000	000	}
200.150.97.0	200.151	011000	001	
200.150.98.0	200.152	011000	010	
200.150.99.0	200.153	011000	011	
200.150.100.0	200.154	011000	100	
200.150.101.0	200.155	011000	101	
200.150.102.0	200.156	011000	110	
200.150.103.0	200.157	011000	111	
200.150.104.0	200.158	011001	000	}
200.150.105.0	200.159	011001	001	

Figure 7-8. Supernetted networks

After the conversion to binary, you can see that the patterns in the third octet start off the same, but begin to vary after moving to the right. For example, networks 200.150.96.0 and 200.150.100.97.0 are the same until the eighth bit of the third octet. If one bit (indicated by the first column from the right) was stolen from the network portion of the address, the mask would change from 255.255.255.0 to 255.255.254.0 because the bit stolen back would be changed to a zero:

```

11111111.11111111.11111111.00000000 → 11111111.11111111.11111111.0.00000000
255.      255.      255.      0      255.      255.      254.      0

```

When this is done, the last bit of the network address is ignored because, otherwise, the ANDing process would always result in a 0. Thus, the 96 and 97 networks would have the same pattern to the left of the stolen bit and effectively be in the same network. The same is true of the 98 and 99 networks, the 100 and 101 networks, and the 102 and 103 networks.

Stealing two bits changes the mask to 255.255.252.0 and causes the last two bits of the network addresses to be ignored. At this point, four networks would be supernetted together if their binary patterns to the left of the stolen bits were the same. From [Figure 7-8](#), you can see that networks 96–99 would be supernetted together, as would networks 100–103. Stealing three bits would cause 200.150.100.96–200.150.100.103 to be supernetted together.

When stealing bits from the network portion, the values are similar to those used in subnetting, but decrease as more bits are stolen or converted to 0s ([Table 7-4](#)).

Table 7-4. Supernet mask patterns

Bits stolen	Mask	This example	Class B example
0	255	255.255.255.0	255.255.0.0
1	254	255.255.254.0	255.254.0.0
2	252	255.255.252.0	255.252.0.0
3	248	255.255.248.0	255.248.0.0
4	240	255.255.240.0	255.240.0.0
5	224	255.255.224.0	255.224.0.0
6	192	255.255.192.0	255.192.0.0
7	128	255.255.128.0	255.128.0.0
8	0	255.255.0.0	255.0.0.0

The Supernetted Network

In order to determine the network address for the networks supernetted together, the lowest numbered network matching the pattern is used. Remember that supernetting actually increases the size of the network in terms of the number of hosts and will extend to the end of highest network matching the binary pattern. In this case, stealing three bits results in a mask of 255.255.248.0. The lowest numbered network matching this pattern is 200.150.96.0 and the highest is 200.150.103.0. This means that the network range is 200.150.96.0–200.150.103.255.

200.150.96.0 is the network address and 200.150.103.255 is the directed broadcast for the network. To verify this, the same ANDing process is used. Given the original host address of 200.150.100.95 and the new mask of 255.255.248.0, the ANDing process resolves as shown in [Figure 7-9](#).

					Supernet Field	
					<div>100</div> <div>000</div> <div>000</div>	
11001000	.10100000	.01100				.01011111
11111111	.11111111	.11111				.00000000
11001000	.10100000	.01100				.01000000
200.	150.	96.				0

Figure 7-9. Supernet binary

Any address in this range will have the same result after ANDing.

Classless Inter-Domain Routing

Classful addressing did not have much of a future. As the number of networks attached to the Internet passed 10,000, it became apparent that if every organization wanted its own network, it wouldn't take long to run out of possible network addresses. The information in [Table 7-1](#) shows that there are slightly more than 2 million possible networks in total, and most of these are small class Cs. A longer-term problem was the eventual exhaustion of the entire IPv4 address space because it is based on 32 bits.

Compounding the issue is the fact that the classful architecture was horribly inefficient. An organization might receive a class C networks even if it only had a dozen network nodes. Making matters worse, an organization possessing 300 nodes, or believing that it might possess 300 nodes, would receive a class B network. The ability to manipulate the mask helps, so with supernetting, this organization might receive two class Cs instead. This was an improvement, but still results in low address space utilization efficiency and there was no guarantee that the address space would be continuous. A contemporary example exists in high-speed Internet connections to the home. The traditional viewpoint might be that everyone connecting a small home network to the Internet should be granted their own network. Clearly, this is not possible.

Aside from all of this inefficiency and large of address space is routing table explosion. Routers are tasked with forwarding packets based on the destination network address. If every single organization were given a separate network, there would also be corresponding routing table entries. Routing tables across the Internet would grow until performance for routers on the interconnected networks was severely hampered. It takes time to construct and maintain a routing table and additional time to find the correct routing table entry for a particular packet by searching the entries and com-

pleting the mask-based operations. This is also referred to as *traversing the routing table*. The speed of these calculations is dependent on a number of factors, including the processing power of the router, the installed memory, and the size of the routing table. Now imagine that the router is doing this for millions of packets per second, and the scope of the problem becomes clear.

The previous section describes how supernetting works, but supernetting was actually introduced as a scheme to contend with the issues of address space conservation and controlling the size of routing tables. Specific attention was paid to the routers that do not use default routes and the class B address space. The class B address space is of a size that appears to fit no one, as midsize organizations require more addresses than a class C network can offer, but cannot make use of an entire class B network of 65536 (65634) addresses. By manipulating the mask lengths, routing can depart from the stratified class structure, in effect becoming classless. Classless Inter-domain Routing, or CIDR, was originally described in RFC 1519 and obsoletes RFC 1338 (supernetting), but the language is nearly identical.

Aggregation is a technique in which a reduced number of routing table entries can be used to forward traffic to downstream routers, because each entry encompasses several smaller networks due to variable length subnet masks. The number of routes that must be advertised via a routing protocol is reduced correspondingly. [Figure 7-10](#) shows a topology in which aggregation might be deployed.

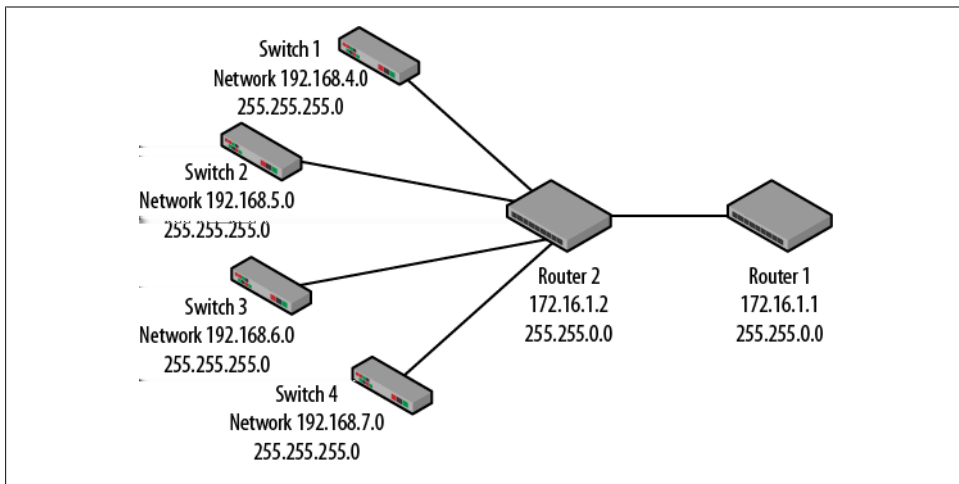


Figure 7-10. Aggregation topology

To get to all of the networks shown, the routing tables might be constructed as shown in [Table 7-5](#).

Table 7-5. Router routing tables

Router 2			Router 1		
Address	Mask	Connection	Address	Mask	Connection
192.168.4.0	255.255.255.0	Directly connected	192.168.4.0	255.255.255.0	via 172.16.1.2
192.168.5.0	255.255.255.0	Directly connected	192.168.5.0	255.255.255.0	via 172.16.1.2
192.168.6.0	255.255.255.0	Directly connected	192.168.6.0	255.255.255.0	via 172.16.1.2
192.168.7.0	255.255.255.0	Directly connected	192.168.7.0	255.255.255.0	via 172.16.1.2
172.16.0.0	255.255.0.0	Directly connected	172.16.0.0	255.255.0.0	Directly connected
Default route via 172.16.1.1					

Router 1 must have all of the networks beginning with 192 included in its routing table, and all of these networks are using a 24-bit mask. The network connecting the two routers (172.16.0.0) is a class B and uses a 16-bit mask. As demonstrated in the earlier discussion on supernetting, networks sharing a common binary pattern can be collected together to form a larger network. The same is true of routing table entries. CIDR allows the use of variable-length masks to help slow the growth of routing tables. One other important point is that the four networks beginning with 192 are all accessible via the same pathway—172.16.1.2. This means that traffic destined for the four networks must travel through the same router interface.

Examining the binary of the third octet for the networks in question demonstrates that the first six bits have the same pattern ([Table 7-6](#)).

Table 7-6. Routing table binary patterns

Network	Byte 3 in binary
192.168.4.0	00000100
192.168.5.0	00000101
192.168.6.0	00000110
192.168.7.0	00000111

Based on this information, the routing table entry can be modified as shown in [Table 7-7](#).

Table 7-7. Updated router routing table

Router 2			Router 1		
Address	Mask	Connection	Address	Mask	Connection
192.168.4.0	255.255.255.0	Directly connected	192.168.4.0	255.255.252.0	via 172.16.1.2
192.168.5.0	255.255.255.0	Directly connected	172.16.0.0	255.255.0.0	Directly connected
192.168.6.0	255.255.255.0	Directly connected			
192.168.7.0	255.255.255.0	Directly connected			
172.16.0.0	255.255.0.0	Directly connected			
Default route via 172.16.1.1					

By changing the mask, this entry now refers to a larger chunk of address space: 192.168.4.0 - 192.168.7.255. Again, it is important to note that the pathway for all of the traffic is the same as the forwarding router interface—172.16.1.2. Counting the number of binary 1s in the entry mask (255.255.252.0), it can be said that this entry has a 22-bit mask.

CIDR and Aggregation Implementation

Given the problems outlined so far, there are some limits to a plan like this one. In [Figure 7-4](#), the networks are very neatly arranged to allow the aggregation of the routing table information. Unfortunately, networks are not typically this organized. RFC 1519 mentions this particular problem, but specifies that network addresses will not be re-assigned. However, newly added addresses/networks can certainly be from aggregated address space. As an example, large ISPs, such as Time Warner, may control a block of addresses within the class A 24 network. Customers of Time Warner receive a portion of the 24 network rather than their own separate networks. In this way, the routing tables and the advertisements can be aggregated through the Time Warner paths. Both RFC 1338 and 1519 contain the following statement:

For these reasons, and in the interest of providing a consistent procedure for obtaining Internet addresses, it is recommended that most, if not all, network numbers be distributed through service providers.

A second problem is that many networks may be multihomed. A basic part of good network design is to have redundant or backup connections to the outside world. Thus, the availability of the network will be maintained by a second entry in another router and the number of advertisements required may not be as low as hoped. So, while aggregation and the variable length masks via CIDR certainly help, there are limits on the ability to slow routing table growth. At the time of its writing, the authors were concerned with routing tables growing to 10,000 routes and, in their wildest imaginations, perhaps reaching 100,000 entries. Today routing tables on some core Internet routers have actually gotten much larger. According to the CIDR report (www.cidr-

report.org) the number of entries has passed 200,000 after aggregation via CIDR and the network prefixes exceed 300,000.

Along with the manipulation of the mask length is the term *CIDR notation*. This usually refers to another way of indicating the mask length by counting the number of ones in the mask. For example, the class C network of 200.150.100.0 with a mask of 255.255.255.0 can be referred to as 200.150.100.0/24. Converting the base 10 values of the mask (255 = 11111111) returns 24 1s. CIDR notation is often used to abbreviate descriptions, providing a clear indication of different mask lengths. The number following the slash (/) indicates the network or prefix length, and is commonly used in routing tables. The following is from a router running OSPF:

```
O IA 192.168.4.0/24 [110/20] via 192.168.2.252, 00:02:48, FastEthernet0/0
O IA 192.168.5.0/24 [110/30] via 192.168.2.252, 00:02:44, FastEthernet0/0
O 192.168.1.0/24 [110/11] via 192.168.3.254, 00:09:46, FastEthernet0/1
C 192.168.2.0/24 is directly connected, FastEthernet0/0
C 192.168.3.0/24 is directly connected, FastEthernet0/1
```

RFC 4632

RFC 1519 (CIDR) and its predecessor, 1338 (supernetting), are very similar in that they both address aggregation and routing table growth. RFC 1519 adds sections to handle Class D addressing, intradomain routing, and extending CIDR to Class A networks. Recall that one of the primary problems was a reduced number of available class B networks. Much of the work done was to alleviate this stressor by manipulation of the masks specific to class B and C networks. Class A networks (and DNS) were discussed, but not to be affected upon adoption of the new addressing plan.

RFC 4632 obsoletes 1519 and includes updated discussions, clarifications, and a report on the effectiveness of the CIDR addressing scheme. It also provides the definition of the CIDR notation. The decade spanning 1994 to 2004 indicates that the CIDR effort was successful. With the possible exception of the “dot com bubble,” the growth in routing table entries and advertised routes was linear rather than exponential. Since that time, more rapid growth has been observed and, as noted earlier, the number of entries in some core routers exceeds 300,000. The rapid growth may be a factor of increasing adoption of technology or service providers not adhering to the recommendations published for CIDR deployments.

Summary

The major topics of this chapter (subnetting, supernetting, and CIDR) are all related in their manipulation of network masks. They differ in scope and application. Generally, an organization owning a small amount of address space may opt to break it up into smaller chunks via subnetting. Internet Service Providers controlling much larger sections of IPv4 address space collect customer networks together via aggregation

through supernetting or CIDR. Support of these techniques requires that the routing protocols and the equipment support classless advertisements. Aggregation is a technique employed on the Internet as whole in order to slow routing table growth and deal with the limited number of available class B networks. These techniques have been successful in both endeavors. However, in recent years, this growth has again accelerated and new solutions must be found.

There are other forces at work that may have helped slow the growth of Internet routing tables and allowed the IPv4 address space to survive this long. NAT has created an environment in which multiple private addresses can share a single public address. Adoption of IPv6, while relatively light, may have had some impact as well.

RFCs and Reading

RFC 917: “Internet Subnets”

RFC 950: “Internet Standard Subnetting Procedure”

RFC 1338: “Supernetting: an Address Assignment and Aggregation Strategy”

RFC 1519: “CIDR: an Address Assignment and Aggregation Strategy”

RFC 1817: “CIDR and Classful Routing”

RF C 4632: “CIDR: an Address Assignment and Aggregation Strategy”

The CIDR Report: www.cidr-report.org/as2.0/

Review Questions

Given an IP address of 150.125.100.1 and mask of 255.255.248.0, answer the following questions.

1. To what class does this IP address belong and what is the class mask?
2. What is the network address of this node, given the mask?
3. Is this a subnetting or a supernetting problem?
4. How many subnets/networks have been created?
5. How many possible and useable hosts exist in this subnet/supernet?
6. What is the range of host addresses for this network?
7. What are the directed broadcast addresses for the classful and network address, respectively?
8. What are possible high and low router addresses for this network?
9. What are the two problems addressed by RFCs 1338 and 1519?
10. Define aggregation in the context of routing tables and CIDR.

Review Answers

1. Class B, 255.255.0.0
2. 150.125.96.0
3. Subnetting
4. 32
5. 2048 possible, 2046 useable
6. 150.125.96.0 - 150.125.103.255
7. 150.125.255.255 and 150.125.103.255
8. 150.125.103.254 and 150.125.96.1
9. Routing table growth and a lack of class B network addresses
10. Using variable-length network masks in order to collapse smaller address chunks into larger address spaces for the purpose of reducing routing table and routing advertisement size.

Lab Activities

Activity 1—What Is My Network?

Materials: Computer with an active network connection

1. Within Windows, click on the Start button.
2. In the run box, type `cmd` and press Enter. This will open a command window.
3. Type `ipconfig /all`. This will display the IP address of your computer.
4. Perform the ANDing operation of your IP address and mask. What is your network?

Activity 2—Change Your Network

Materials: Computer with an active network connection

1. Within Windows, go to the properties of your network adapter.
2. Open up the Internet Protocol (TCP/IP) properties.
3. Give your adapter the IP address of 192.168.1.100 and a mask of 255.255.255.192.
4. Click OK to save your changes.
5. Repeat activity 1. Is there a difference? Why/why not?

Activity 3—What Is the Address Given to You by Your ISP?

1. Most of us have home gateways such as a Linksys router. The outside address of this box is provided by your ISP. See if you can determine what the IP address is and then determine the size of your network segment.
2. What would the CIDR notation be for your network?

Activity 4—Subnet Calculator

Using Excel or your favorite programming language, create a subnet calculator. Your calculator should allow the user to do the following:

- Input the IP address and mask.
- Select either the new mask or the number of subnets/supernets desired.
- Calculate the range of possible and useable host addresses per subnet/supernet.
- Provide the networked and directed broadcast address for each subnet.
- Provide possible router addresses.
- As a bonus, display the binary for the fields affected by the mask changes.

About the Author

Bruce is a faculty member in the Network, Security, and Systems Administration (NSSA) Department in the Golisano College of Computing and Information Science (GCCIS) at Rochester Institute of Technology (RIT) in Rochester, New York. He splits his time between teaching, projects, and writing.

Colophon

The animal on the cover of *Packet Guide to Core Network Protocols*, first edition, is a helmetshrike.

The cover image is from *Cassell's Natural History*. The cover font is Adobe ITC Garamond. The text font is Linotype Birka; the heading font is Adobe Myriad Condensed; and the code font is LucasFont's TheSansMonoCondensed.

