Cisco Systems

Enhanced IP Services for Cisco® Networks

A practical resource for deploying quality of service, security, IP routing, and VPN services

ciscopress.com

Donald C. Lee, CCIE™

Enhanced IP Services for Cisco Networks

Donald C. Lee

Cisco Press

Cisco Press
201 West 103rd Street
Indianapolis, IN 46290 USA

Enhanced IP Services for Cisco Networks

Donald C. Lee

Published by:
Cisco Press
201 West 103rd Street
Indianapolis, IN 46290 USA

Printed in the United States of America 3 4 5 6 7 8 9 0

Third Printing July 2001

Library of Congress Cataloging-in-Publication Number: 98-86518

ISBN: 1-57870-106-6

Warning and Disclaimer

This book is designed to provide information about enhanced IP services for Cisco networks. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The author, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc. cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

CERT® and CERT® Coordination Center are registered for Carnegie Mellon University in the U.S. Patent & Trademark Office.

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers’ feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through e-mail at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Publisher	John Wait
Executive Editor	Alicia Buckley
Cisco Systems Management	Michael Hakkert Tom Geitner William Warren
Managing Editor	Patrick Kanouse
Acquisitions Editor	Lynette Quinn
Development Editor	Katherine Trace
Project Editor	Theresa Wehrle
Copy Editor	Malinda McCain
Technical Reviewers	Erick Mar Deepak Munjal
Proofreader	Bob LaRoche
Team Coordinator	Amy Lewis
Book Designer	Gina Rexrode
Cover Designer	Aren Howell
Compositor	Wil Cruz
Indexer	Tim Wright

CISCO SYSTEMS

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems Europe
11 Rue Camille Desmoulins
92782 Issy-les-Moulineaux
Cedex 9
France
http://www-europe.cisco.com
Tel: 33 1 58 04 60 00
Fax: 33 1 58 04 61 00

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems Australia,
Pty., Ltd
Level 17, 99 Walker Street
North Sydney
NSW 2059 Australia
http://www.cisco.com
Tel: +61 2 8448 7100
Fax: +61 2 9957 4350

Cisco Systems has more than 200 offices in the following countries. Addresses, phone numbers, and fax numbers are listed on the Cisco Web site at www.cisco.com/go/offices

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China • Colombia • Costa Rica • Croatia • Czech Republic • Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong • Hungary • India • Indonesia • Ireland Israel • Italy • Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines Poland • Portugal • Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain Sweden • Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

About the Author

Donald C. Lee, CCIE #3262, is a Senior Systems Engineer at an IP networking startup. A graduate of UCLA's electrical engineering program, Donald has more than 10 years of experience in the networking industry. During his four years at Cisco Systems, he was responsible for designing, implementing, and troubleshooting network solutions for several of Cisco's largest Fortune 500 customers.

About the Technical Reviewers

Erick Mar is a Senior Systems Engineer at Cisco Systems with CCIE certification in routing and switching (CCIE #3882). As a Systems Engineer for the last seven years for various networking manufacturers, he has provided design and implementation support for large Fortune 500 companies. Erick has an M.B.A. from Santa Clara University and a B.S. in Business Administration from San Francisco State University.

Deepak Munjal (CCIE #4376) has more than 10 years of networking industry experience and a B.S. degree in Computer Science from the University of California, Berkeley. He is currently a Senior Systems Engineer at Cisco Systems and has been actively involved in the design and implementation of end-to-end networking solutions for Cisco's largest Fortune 500 customers. Prior to Cisco, Deepak was a network engineer at a leading computer manufacturing company.

Dedications

To my parents: My parents are great folks and always cheer me up. Now that I'm (mostly) grown-up, I can greatly appreciate all those times they taught, fed, encouraged, and loved me. This book is dedicated to my mom and dad who, I'm proud to say, are the Internet's newest and most distinguished "surfers."

To my wife Shirley: In the age of virtual reality, virtual pets, and virtual offices, Shirley (an incredible human being) experienced a whole new privilege as a result of my book: the *virtual widow*. Despite having to spend some 42 weekends with a keyboard-zombie for a husband, she never stopped her support for the book and she helped me edit many sentences that just wouldn't sit right. This book is also dedicated to her and her enthusiastic support for the Internet (surfing www.gap.com).

Acknowledgments

Kathy Trace deserves extreme kudos and credit for her contribution as the development editor of this book. In addition to being knowledgeable, organized, flexible, and helpful, she’s also an exceptionally nice person.

Erick Mar and Deepak Munjal, the technical reviewers, made this book much better than anything I could have written on my own. In addition to being among the best Cisco engineers I know, they are also great friends. Many thanks go to them for their expertise, advice, and editing.

Lynette Quinn did a superb job managing this project from beginning to end. Her positive attitude and receptiveness made my job easy.

Alicia Buckley provided excellent executive leadership, support, and insight. She also listened to all my random ideas for the book. Like everyone else on the Cisco Press team, she’s incredibly nice and great to work with.

Julie Fairweather and Kim Lew got me jump-started on the book, provided valuable support, and were always a pleasure to work with.

Amy Lewis is a great team coordinator whose attention to detail and timeliness were crucial to the project.

The Cisco Press team who worked behind-the-scenes is top-notch. I wish to thank everyone in illustration, editorial, layout, and the rest of the production process for their contributions.

Finally, my friends at Cisco were always very positive and supportive of this weekend project. I especially wish to thank my managers Srinivas Ketavarapu, Pasha Quadri, and Rico Sacks. Cisco is a great place to work and it is because of these folks.

Contents at a Glance

Table of Contents

Introduction

Your network should provide more than just connectivity. Successful networking means more than installing hardware and programming it to pass packets of data back and forth. Modern networks have mission-critical applications to support, more users and geographical locations, higher bandwidth requirements, and security threats from inside and outside the network. Furthermore, there's rarely enough money, time, or resources to keep up with these demands.

Requirements and resources are opposing forces and are at odds with each other. To relieve this situation, you must do whatever you can to increase the effectiveness of your network.

Effectiveness is the capability of the network to support your current and future users, applications, locations, and policies. A network that merely provides connectivity between locations might meet the requirement for basic data communication, but it won't have what it takes to deliver reliable service for mission-critical applications, scalability for a growing user population, or security for protecting information. In the end, a highly effective network enables organizations to deploy more services to more users in more locations with greater confidence and security.

Increasing effectiveness means to enhance, optimize, and extend the current capabilities of the network—that is, to make the network more useful, more efficient, and more capable of handling demand. The following are some important network capabilities covered in this book:

- **Routing**—The routing function moves data through the network efficiently and finds new paths when network outages occur. Routing also affects how large the network can grow—that is, the number of users and locations you can support, the complexity of the topology, and the stability of the network as it expands. This ability to grow is called the *scalability* of a network.
- **Intelligence and Quality of Service**—This is the capability of a network to recognize and deliver different types of data based on policies you define. An intelligent network recognizes traffic from different applications and prioritizes them into different qualities of service (also called classes of service). Your policy defines prioritized levels of service and classifies the mix of applications on your network into these levels of service. For example, you might define a high quality of service for mission-critical applications, a medium quality of service for general applications, and a low quality of service for low-priority applications. An intelligent network with quality of service ensures that high-priority traffic is delivered to its destination with the shortest possible delay. Without quality of service, all applications are treated equally. This can adversely affect the deployment and operation of applications requiring short delays and fixed levels of bandwidth.
- **Security**—Security services protect the confidentiality and integrity of information on your network. These services increase the trust users have in the network and make the network suitable for new applications. Some security services protect against attacks that aim to disable or cripple the network service itself. Security countermeasures increase the reliability of the network and are no doubt crucial for a high level of effectiveness. Security services also enable you to extend the network to new locations securely. For example, you might want to extend the network to a telecommuter's home via the public telephone system or to branch-office locations through encrypted tunnels over the public Internet.

Cisco IOS

Cisco's Internetwork Operating System (IOS) software runs in Cisco routers and switches—the devices used to build the Internet and the majority of corporate networks. IOS is packed with so many features in so many technologies that just learning the names of the features and what they do is challenging. A quick look through the documentation, which consumes a good-sized bookshelf, is all you need to realize how comprehensive and daunting the IOS feature set is. However, you do not need to learn the intricacies of every IOS command to build and maintain an effective network for your organization.

Purpose of This Book

This book focuses on *enhanced IOS services* that help you increase the effectiveness of your IP network. You might need these services to help run your network today, or you might need to understand some of these technologies to prepare for the future. This book will help you in either case, by focusing on tasks that give you the most results for your effort. In addition to showing you how to configure each service, this book also provides background on why you might need the service and how it works.

The following list is a sampling of what you will find in this book:

- Getting efficient use of network resources such as addresses and bandwidth
- Optimizing routing services
- Integrating networks with different routing protocols and different addressing architectures
- Gatekeeping the consumption of network bandwidth
- Adding intelligence and quality of service in the network to support new applications
- Setting policies on the network for users and their services
- Extending the network to new places, such as the Internet, securely
- Protecting information and network resources

Study the services and practices in this book. Then analyze the current state of your network. Finally, decide how you might take your network to the next level: to an enhanced network that is scalable, intelligent, and secure.

Audience

This book is intended for networking professionals who are responsible for designing, implementing, and managing IP services in enterprise networks. Although the focus is on Cisco IOS, the principles and strategies covered in this book can readily apply to any IP network. No major background in IOS, TCP/IP, or routing is required, but a familiarity with these topics will get you started right away. For experienced networking professionals such as Cisco Certified Internetwork Experts (CCIEs) and candidate CCIEs, this book aims to provide unique technologies and effective practices that not only deliver value on your network but also provide opportunity for professional growth. For folks completely new to Cisco router configuration, Appendix E, "A Crash Course in Cisco IOS," covers all the basics.

Organization

The eight chapters and five appendixes of this book are organized into four parts.

Part I—Managing Routing

The aim of the first part is to get the most out of IP addressing and routing. Chapter 1 progresses logically from basic addressing to more sophisticated topics such as VLSM, classless addressing, summarization, and NAT. Experienced readers may skip Chapter 2, which covers basic routing protocols and sets up a foundation for Chapter 3. Chapter 3 rounds out Part I with routing services that enhance network flexibility and scalability. These services include route filtering, redistribution, default routing, summarization, and policy routing.

Part II—Managing Quality of Service

The goal for Part II is to understand, implement, and validate quality of service (QoS) on a network. Chapter 4 identifies the driving forces behind QoS and QoS principles and covers basic services such as Priority Queuing, Custom Queuing, and Weighted Fair Queuing. Chapter 5 details IOS's advanced QoS mechanisms, including RSVP, RED, CAR, and Class-Based Weighted Fair Queuing.

Part III—Managing Security

The objective of the third part is to secure the network, protect data and users, and extend connectivity with confidence. Chapter 6 covers access lists, basic router security, AAA services, and some simple commands that enhance network security. Chapter 7 begins a survey of advanced security services and provides details about IPsec—a leading technology for building VPNs. IPsec's building blocks include IKE, transforms, security associations, modes, AH, ESP, and basic cryptography (digital certificates, digital signatures, public key cryptography, Diffie-Hellman, and the like). Finalizing the coverage of advanced security services, Chapter 8 shows you how to use IOS as a stateful firewall and an intrusion detection system. These services protect your organization from malicious attacks.

Part IV—Appendixes

Five appendixes are included in Part IV:

- Appendix A, "Obtaining IETF RFCs," provides instructions on how to obtain IETF RFCs.
- Appendix B, "Retrieving Internet Drafts," explains Internet Drafts and shows you how to get them.
- Appendix C, "Common TCP and UDP Ports," is a reference table of common TCP and UDP port numbers.
- Appendix D, "Password Recovery," is a quick reference for recovering lost or forgotten passwords on Cisco routers.
- Appendix E, "A Crash Course in Cisco IOS," is a quickstart on IOS navigation, configuration, and monitoring. It also furnishes some tips and tricks, so it's worth a skim even if you've worked with IOS for a while.

Conventions and Features

When appropriate, the services covered in this book adopt the following basic content structure:

- **What is it?** A description of the IOS service and why you might need it.
- **How does it work?** Technical information on the underlying mechanism and a look at what's going on behind the scenes.
- **How do you configure it?** Practical instructions and configuration examples.
- **How do you check it?** Some ways of validating, monitoring, and debugging your results.

Within the text, IOS commands are printed in **boldface** for readability. In some cases, boldface is also used as an aid for locating interesting text in IOS outputs. *Italic text*, when used in IOS commands, indicates arguments for which you supply values.

The listings of IOS configurations sometimes include the IOS prompt when it helps illustrate the configuration steps. Otherwise, the prompt is omitted and the relevant portion of the configuration is printed as an output of **show running-config** (or, equivalently, **write term**).

Finally, important concepts are called out from the text as notes, and sidebars offer insight into related concepts or techniques. Tips highlight information that might be helpful as you implement these enhanced services.

Support

Although every effort was made to stamp out errors, documentation bugs sometimes arise from the mass of technical details. In an effort to further customer support, the Cisco Press Web site at www.ciscopress.com is available for clarifications, corrections, and other possible errata related to this book.

PART I

Managing Routing

CHAPTER 1

Managing Your IP Address Space

The first step in achieving a scalable and effective IP network is devising a solid addressing plan. Your addressing plan lays down the foundation for the network by portioning your IP address space into smaller, manageable ranges, or *blocks*. The addressing plan also defines the deployment of these blocks into various parts of the network for supporting devices.

Unlike such protocols as IPX or AppleTalk, IP requires a respectable amount of address planning at the outset. This is true for large and small networks alike, because the growth of the Internet has made IP addresses a precious and scarce resource.

The Internet's IP address space is finite. With the growth of the Internet, the number of available addresses is diminishing and addresses are becoming more difficult to obtain. Although addressing is a rather mundane task, a solid addressing plan will save you many headaches in the future (and protect your reputation when others inherit your work). Also, IP networks can—and generally should—have a hierarchical addressing structure. This is achieved by summarizing, or *aggregating*, addresses. Summarization heightens the importance of address planning even more (see "Planning for Address Summarization," later in this chapter).

Devising your address strategy is akin to planning the layout of a house. You are going to spend a lot of time in your house, so a crucial step is spending enough time on the design and allocation of the floor space for now and in the future. Are there enough rooms? Is the size of each room adequate and appropriate? What is the most efficient use of the floor space? Although you cannot guarantee a final house design that meets all future requirements, you need to come up with a plan that makes the most sense. You want a well-thought-out design that will postpone any remodeling efforts until far off in the future. By all means, you want to avoid having to demolish the whole thing and start over with a new floor plan. Like floor plans, IP addressing plans generally do not change for long periods of time and, when they do change, overhauling them can be a major effort.

This chapter covers IP addressing concepts, design techniques, strategies for maximizing efficiency, and services for scaling network addressing.

The main topics of this chapter are

- Review of Traditional IP Addressing
- Subnetting a Classful Address Space

- Subnetting with Variable Length Subnet Masks
- Overview of Classless Addressing
- Planning for Address Summarization
- Conserving Subnets with IP Unnumbered
- Scaling the Address Space with Network Address Translation

Review of Traditional IP Addressing

Traditional IP addressing organizes the entire 32-bit IP address space into blocks called *classes* and further breaks down each class into network numbers. Early Internet standards defined five classes, outlined in Table 1-1.

Table 1-1 *The Original Organization of the 32-bit Address Space*

Class Name	Address Range	# of Addresses per Network	Purpose
A	0.1.0.0 to 126.0.0.0	16,777,216	Unicast; very large networks
B	128.0.0.0 to 191.255.0.0	65,536	Unicast; large networks
C	192.0.1.0 to 223.255.255.0	256	Unicast; small networks
D	224.0.0.0 to 239.255.255.255	N/A	Multicast
E	240.0.0.0 to 247.255.255.255	N/A	Experimental use

NOTE Network 127.0.0.0 is a special range of addresses reserved for *loopback addresses* (addresses used locally by IP hosts). Such addresses should never appear on a network.

As Table 1-1 illustrates, a 32-bit IP address is written as four *octets* (8-bit groups) separated by periods, with each octet expressed as a decimal number. This is known as *dotted decimal notation*. The following is an example IP address in its binary and dotted decimal forms:

32-bit IP address: 10101100000100000000101000010100
Same address grouped into four octets: 10101100.00010000.00001010.00010100
Same address in dotted decimal notation: 172.16.10.20
Class of the network: B
Network the address belongs to: 172.16.0.0

The class scheme served as a starting point for easy and rapid deployment of the Internet address space. Much like acquiring land for their buildings, organizations obtained network numbers from the three classes (classes A, B, and C) based on the number of IP addresses they needed. Two classes were reserved for special purposes: class D addresses for IP multicast and class E addresses for experimental use.

After an organization secured a class B network, for example, it could autonomously deploy the addresses contained in that range to its computers, or *hosts*. With the additional deployment of internetworking services (*routing*), that class B network could communicate with other class A, class B, and class C networks within the organization and throughout the Internet.

NOTE This book covers IP version 4, which is the most prevalent form of IP on private networks and the public Internet at the time of this writing. The next version of IP, version 6, has a different addressing format and intends to provide a much larger address space than IP version 4 (IPv6 increases the address space from 32 bits to 128 bits). See the bibliography for sources of IP version 6 information.

To gain more efficient use of the address space, the Internet community adopted a practice of dividing a network into subnetworks called *subnets*. When a network is divided into subnets, its original network number is called the *major network number* or *major net*. Routing is still required to interconnect subnets just as it is required to interconnect major nets.

For most organizations, subnetting is a necessary part of managing an address space—it portions a single major net of limited use into smaller subnets that can be deployed more effectively.

Still, networking professionals are faced with addressing problems that subnetting alone cannot solve. The scarce supply of major nets and pressure from an ever-growing IP population have taken the menial task of addressing to the top of the priority list. Later sections of this chapter offer solutions that will help you get more efficient use of your address space and alleviate the shortage problem.

Subnetting a Classful Address Space

As mentioned previously, the Internet's original address plan was organized into classes: classes A, B, C, D, and E. Networks deployed with this plan are said to be *classful networks* or networks with *classful addressing*. Many privately owned networks still use classful addressing, even though the public Internet has abolished classful addressing in favor of *classless addressing* (covered in "Overview of Classless Addressing" later in this chapter).

In brief, classless addressing discontinues the grouping of addresses into classes A, B, and C and treats the address space as a large, contiguous block of addresses.

NOTE The Internet community adopted classless addressing to get efficient use of the existing address space and to avoid address depletion. See "Overview of Classless Addressing" later in this chapter.

Why care about classful addressing versus classless addressing? Addresses are addresses, aren't they? The distinction between classful and classless addressing is important when it comes to routing protocols. Some routing protocols—Routing Information Protocol (RIP) and Interior Gateway Routing Protocol (IGRP), for example—were created before the practice of classless addressing and support only the rules defined by traditional classful addressing (these rules are simple, but restrictive). Classful routing protocols, such as RIP, do not support newer and more advanced features developed in classless routing protocols, such as Open Shortest Path First (OSPF) and Enhanced IGRP (EIGRP). These advanced features include variable length masking and summarization and are covered later in this chapter (see "Subnetting with Variable Length Subnet Masks," "Overview of Classless Addressing," and "Planning for Address Summarization"). Routing protocols are also covered in Chapter 2, "Deploying Interior Routing Protocols," and Chapter 3, "Managing Routing Protocols."

Although the Internet has ceased using classful addressing, many organizations need to support networks that were designed with classful networks and classful routing protocols, such as RIP and IGRP. This section covers the basics of subnetting because the technique is crucial for supporting a classful network and is a prerequisite to deploying classless networks. The section includes discussion on

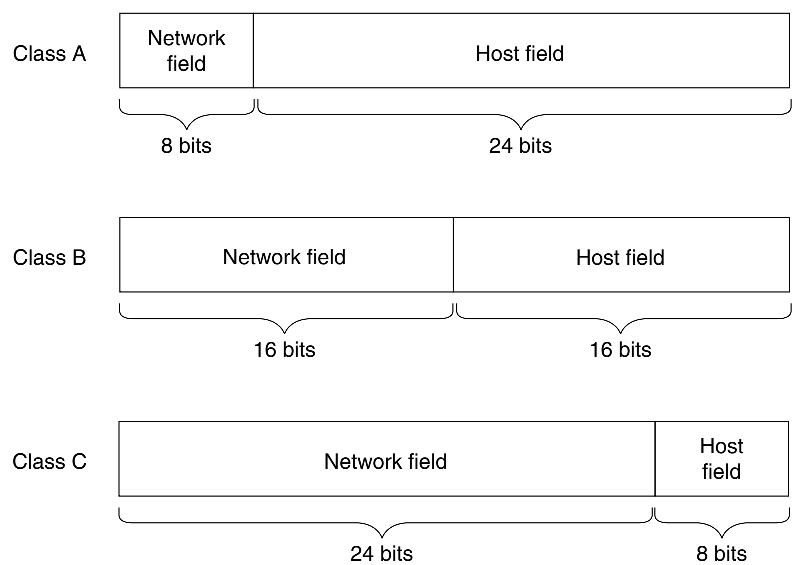
- Major Nets and Subnet Masks
- Classful Subnetting: An Example
- Calculating the Number of Host Addresses in a Subnet
- Finding Subnet Information, Given a Host Address and the Mask
- Disadvantages of Subnetting
- The Rules on Top and Bottom Subnets
- Using Subnet-Zero to Get Around the Rules

Major Nets and Subnet Masks

Every major net has two fields: the *network field*, which uniquely identifies the major net, and the *host field*, which uniquely identifies hosts within the major net. Figure 1-1 illustrates the number of bits in the network and host fields for each class.

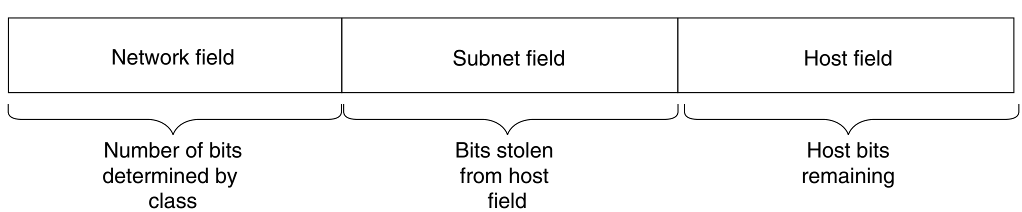
As mentioned in the previous section, subnetting is the process of dividing a major net into smaller (and generally more useful) subnets. This is accomplished by "stealing" some bits from the host field of the major net and using those bits to designate the subnet addresses. The host field varies in length, depending on the class of major net being subnetted (see Figure 1-1).

Figure 1-1 *Lengths of the Network and Host Fields by Class*



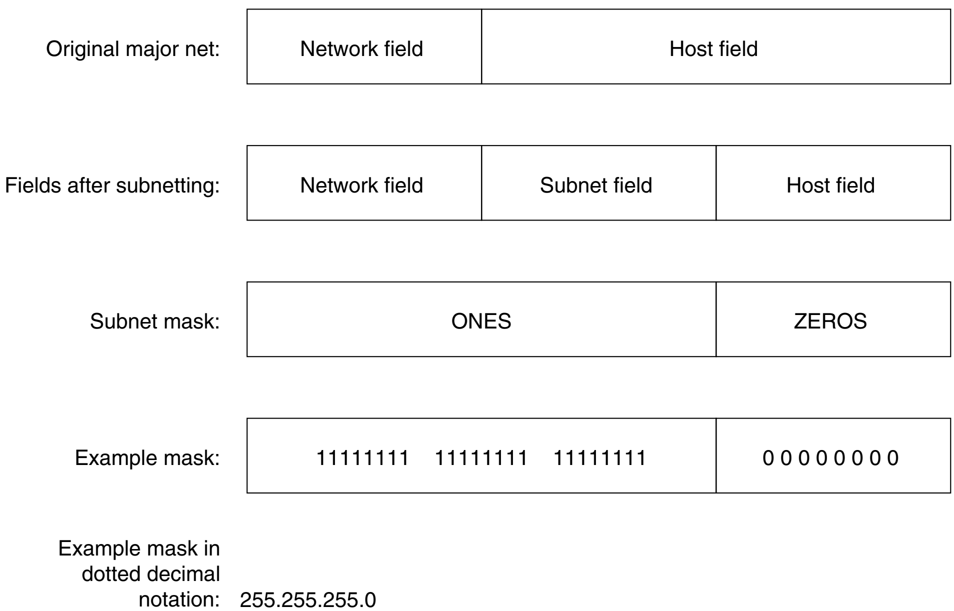
When you consume some of the bits in the host field for subnets, you are left with three fields: the original network field, a newly created subnet field, and a reduced-size host field. Figure 1-2 illustrates the three fields you get after subnetting.

Figure 1-2 *Subnetting Results in Network, Subnet, and Host Fields*



You declare the number of bits you are stealing from the host field with a 32-bit *subnet mask*. The subnet mask contains a contiguous series of ones that start from the left-most bit (also called the *most significant bit*). Where the ones end and the zeros begin is the boundary between the subnet field and the host field. Figure 1-3 describes a subnet mask and provides an example.

Figure 1-3 *Defining the Subnet and Host Fields with a Subnet Mask*



The example mask in Figure 1-3 has 24 one bits that start from the far left and 8 zero bits that fill out the remaining bits to the far right. This mask defines a host field of 8 bits because the boundary between the ones and the zeros is between the 24^{th} and 25^{th} bits (bits 25 through 32 are zero and represent the host field). The size of the subnet field depends on whether this mask is applied to a class A, class B, or class C major net. Recall from Figure 1-1 that the network field is defined by the class of the major net.

When you convert the mask from Figure 1-3 into dotted decimal notation, you get 255.255.255.0, because

- The first octet (the first group of 8 bits) is all ones (255 in decimal)
- The second octet is all ones (255 in decimal)
- The third octet is all ones (255 in decimal)
- The last octet is all zeros (0 in decimal)

The example in Figure 1-3 is a rather straightforward example because each octet is either all ones or all zeros. Things get more interesting when the boundary between the ones and zeros falls within an octet. Consider another mask:

```
11111111111111111111111111000000
```

To make this mask easier to read, separate the octets like this:

```
11111111.11111111.11111111.11000000
```

Now, convert each octet into decimal:

```
255.255.255.192
```

The preceding mask defines the subnet-host field boundary between the 26th and 27th bits, resulting in a host field of 6 bits (bits 27 through 32). Again, the size of the subnet field depends on the class of the major net to which you apply this mask. It's time for an example.

Classful Subnetting: An Example

The best way to get familiar with subnetting is to practice. Consider the following example that subnets major net 192.168.1.0 by stealing three bits from the host field to make a three-bit subnet field as shown in Example 1-1.

Example 1-1 *Subnetting a Class C Major Net with a Three-Bit Subnet Mask*

Major net: 192.168.1.0
Class: C
Length of original host field: 8 bits (from Figure 1-1)
Number of host bits to steal for subnet field: 3 bits
Number of host bits remaining after subnetting: 8-3=5 bits

Network field
192.168.1

Major net in binary: 1100-0000.1010-1000.0000-0001.0000-0000
Subnet mask in binary: 1111-1111.1111-1111.1111-1111.1110-0000

Subnet field Host field

Subnet mask in
dotted decimal notation: 255.255.255.224

The common way to write a major net together with its subnet mask is by using the shorthand notation of the major net followed by a slash (/) and the number of ones in the mask. The shorthand notation for 192.168.1.0 masked with 255.255.255.224 (see Example 1-1) is 192.168.1.0/27 (there are 27 contiguous ones in 255.255.255.224).

NOTE Both the dotted decimal and slash notations are acceptable, and both notations are used when working with Cisco routers. For example, configuring an address on a router interface requires the mask in dotted decimal notation, but the output of **show ip route** favors slash notation in most versions of IOS. Also, some people prefer one notation over the other, so a good idea is to be familiar with both.

As you can see from Example 1-1, converting from dotted-decimal notation to binary when subnetting is often convenient. A separator, such as a hyphen, makes it easier to read eight bits in a row.

Example 1-1 uses three bits for the subnet field. This yields eight unique combinations that are used to identify the subnets: 000, 001, 010, 011, 100, 101, 110, and 111. The eight subnets for Example 1-1 are listed in Table 1-2. The three bits that make up the subnet field are printed in boldface to emphasize the distinction between the subnet bits and the host bits.

Table 1-2 *The Eight Subnets for Example 1-1*

Subnet Field	Octet *x* in 192.168.1.*x* (bin)	Octet *x* in 192.168.1.*x* (dec)	Subnet Number
111	**111**0-0000	224	192.168.1.224/27
110	**110**0-0000	192	192.168.1.192/27
101	**101**0-0000	160	192.168.1.160/27
100	**100**0-0000	128	192.168.1.128/27
011	**011**0-0000	96	192.168.1.96/27
010	**010**0-0000	64	192.168.1.64/27
001	**001**0-0000	32	192.168.1.32/27
000	**000**0-0000	0	192.168.1.0/27

In traditional subnetting, you are not allowed to use the so-called *top* and *bottom* subnets. The top subnet has all ones in the subnet field and the bottom subnet contains all zeros. For the preceding example, 192.168.1.224/27 is the top subnet and 192.168.1.0/27 is the bottom subnet. This leaves the middle six subnets available for deployment, but the top and bottom subnets are wasted. The section "Using Subnet-Zero to Get Around the Rules" later in this chapter covers how you can use the bottom subnet.

Calculating the Number of Host Addresses in a Subnet

Calculating the number of hosts that can be addressed per subnet is not difficult. Each bit position can be either a one or a zero, so starting with one bit, there are two possible combinations. The number of possible combinations doubles each time you add an additional bit. Two bits yields four combinations, three bits yields eight combinations, four bits yields 16 combinations, and so on.

The formula for the number of combinations is 2^n, where n is the number of bits in the field. Example 1-1 has five bits in the host field after three bits are stolen for the subnet field. This yields 2^5=32 unique combinations for addressing hosts; however, the all-zeros and all-ones patterns are reserved for the subnet number and subnet broadcast address, respectively. After subtracting these two reserved addresses, 30 addresses per subnet remain for host addresses.

Finding Subnet Information, Given a Host Address and the Mask

Given a host address and the subnet mask, you can determine the subnet on which that host lives. This is another common exercise and is useful anytime you need to track the subnet number for a host (in a routing table, for example). Suppose you are given the following host address and subnet mask:

```
172.16.9.136/22
```

To start the process, convert the host address and mask to binary and write the mask below the host address (for clarity, the host field bits are printed in boldface here):

```
1010-1100.0001-0000.0000-1001.1000-1000 = 172.16.9.136
1111-1111.1111-1111.1111-1100.0000-0000 = /22
```

Now, focus on the boundary defined by the mask (where the ones end and the zeros begin). This is the boundary between the subnet field and the host field and tells you that the last 10 bits of the address make up the host field. An easy way to determine the subnet number is to take the host address and set all of the bits in the *host field* to zero, like this:

```
1010-1100.0001-0000.0000-1000.0000-0000 = 172.16.8.0
```

Thus, host 172.16.9.136/22 is on subnet 172.16.8.0/22.

NOTE You might notice that the subnet number is the result of a binary "AND" operation on the address and mask at each bit position. This is how computers (and routers) calculate the subnet number.

Additionally, you can easily find the IP broadcast address for the subnet. This is done by setting all of the bits in the host field (printed again in boldface) to one, like this:

```
1010-1100.0001-0000.0000-1011.1111-1111 = 172.16.11.255
```

Thus, the broadcast address of subnet 172.16.8.0/22 is 172.16.11.255. Sending a packet (a ping, for example) to 172.16.11.255 is a transmission to every host in the subnet.

Last, you can find the range of valid host addresses for this subnet. The range contains the addresses *between* the subnet number (host field of all zeros) and the broadcast address (host field of all ones), so the host address range for subnet 172.16.8.0/22 is

```
1010-1100.0001-0000.0000-1000.0000-0001 = 172.16.8.1
```

through

```
1010-1100.0001-0000.0000-1011.1111-1110 = 172.16.11.254
```

You can verify that the host address 172.16.9.136, introduced at the start of this section, indeed falls within this address range.

Disadvantages of Subnetting

Note that subnetting is restrictive because the technique forces you to commit to the number of subnets you need now and in the future. You also need to commit to the number of hosts per subnet, because every bit you steal for the subnet field means one less bit you can use for host addresses.

Making matters worse, the technique produces subnets that are all of equal size in the number of hosts that can be supported per subnet. Therefore, you often have to do the sizing based on the largest subnet needed and waste addresses when deploying the remaining subnets to areas with fewer hosts. These issues apply when you're using a routing protocol that only supports a fixed-size mask. "Subnetting with Variable Length Subnet Masks," later in this chapter, covers a method of subnetting that mitigates some of the problems with fixed-size masks.

The Rules on Top and Bottom Subnets

Arguments exist both in theory and in practice for not using the top and bottom subnets in a classful network. Theoretically, a bit field has two special patterns:

- **All-zeros pattern**—usually means "this" as in "this host" or "this network."
- **All-ones pattern**—usually means "all" as in "all hosts" or "all networks."

Early Internet documents said it was a good idea to keep these meanings and apply them to the subnet field, thus disallowing the use of the bottom subnet of all zeros and the top subnet of all ones. As a result, IP software in devices obeyed these rules and checked if users erroneously attempted to configure a device in violation of the rules.

NOTE The advent of classless addressing abolished the notion of the top and bottom subnets (and subnets in general). In a classless environment, devices can use the address space that the classful world knows as the top and bottom subnets. See "Overview of Classless Addressing" later in this chapter for information on classless addressing.

In practice, using the top or bottom subnet can be problematic, because not all devices, especially legacy devices, allow these to be configured. Although you might be successful at deploying some hosts and routers on these outer subnets, you might find that other devices forbid you to configure an address from the top or bottom subnet. You'll then have to find another subnet for those devices. To avoid problems, a good idea is to be familiar with the diversity of devices in your environment and determine the addressing allowed on those devices.

The root of the controversy lies in the ambiguity of addresses when you're using the top or bottom subnets. Take, for example, a bottom subnet field that contains all zeros (the host field also contains all zeros)—the subnet number is the same as the major net number. This is apparent in Example 1-1, where the bottom subnet 192.168.1.0/27 is the same address as the major net (see Table 1-2). This ambiguity can be a source of confusion for some devices because a reference to the subnet is indistinguishable from a reference to the major net. Similarly, an all-ones broadcast to the top subnet could be interpreted as a broadcast address to all of the major net, because the top subnet and major net broadcasts are also indistinguishable. Looking again at the example in Table 1-2, a broadcast to the upper subnet 192.168.1.224/27 is 192.168.1.255—the same address as a broadcast to the entire class C (192.168.1.0).

Using Subnet-Zero to Get Around the Rules

Keeping in mind the caveats listed in the preceding section, you can configure Cisco routers to use the bottom subnet so that you gain one more subnet out of your subnetting efforts. To enable the use of the bottom subnet, use the **ip subnet-zero** global command:

```
Router#conf t
Router(config)#ip subnet-zero
```

If you forget to configure this, the router will "complain" when it comes time to assign an address to an interface. The following is an attempt to configure an interface with an address from a bottom subnet on a router without the **ip subnet-zero** command (notice the output **Bad mask**):

```
Router(config)#int s0
Router(config-if)#ip address 192.168.1.2 255.255.255.224
Bad mask /27 for address 192.168.1.2
```

Because the broadcast address for the top subnet is the same as the broadcast address to the entire major net, deploying the top subnet with such classful routing protocols as RIP and IGRP is not recommended. This is not a problem for classless routing protocols, such as OSPF and EIGRP.

A Word on Semantics

For the remainder of this book, the term *network* defines a general service of TCP/IP communication, as in the "corporate network" or "enterprise network." This is also known as an organization's *intranet* and is usually built of campus networks and wide-area networks. The term *major net* refers to a specific IP address space that follows classful addressing, and *subnet* refers to an address space that is extracted from the major net with the subnetting procedure covered earlier in "Subnetting a Classful Address Space."

Subnetting with Variable Length Subnet Masks

With Variable Length Subnet Masks (VLSMs), you carve an address space (such as a major net) with masks of varying lengths to design subnets of different sizes. This allows you to deploy subnets that are appropriate in size to the number of hosts you need to support in a given part of the network. As a result, you can gain efficient consumption of your address space and—depending on how you deploy the addresses—flexibility in the future as you adjust the size of each subnet to handle growth.

NOTE Your routers must be running a routing protocol that supports VLSM, such as OSPF or EIGRP. RIP and IGRP are classful routing protocols and do not support VLSM. Classful routing protocols are limited to a single subnet mask per major net.

Here is the basic technique for variably subnetting a major net:

1. Subnet the space (for example, a major net) into large address blocks based on the large subnets you need in your network.
2. Deploy these large blocks of addresses to support your large subnets.
3. Take any unused large blocks and subnet them further to support smaller subnets with fewer hosts. You can think of this as a second round of subnetting.
4. Deploy the subnets from the second round of subnetting.
5. With additional rounds of subnetting, continue dividing unused blocks of addresses into multiple smaller subnets and deploying them as needed.

Some binary is involved here. Subnetting requires that you understand and visualize binary patterns and apply those patterns to masks. Consider the following example that uses a class C major net.

Using VLSM for Address Space Efficiency: An Example

Suppose Widget, Inc., asks you to subnet one of its class C major nets and tells you it needs the following:

- Two subnets that can support at least 60 hosts
- Four subnets that can support at least 10 hosts
- As many subnets as possible that can support two hosts

The subnets are needed to support some new additions to its network, as summarized in Table 1-3.

Table 1-3 *Subnets Needed by Widget, Inc.*

Subnet Size	Quantity Needed	Purpose
60+ hosts	2	Branch offices
10+ hosts	4	Server farms
2 hosts	As many as possible (use the remaining space)	Point-to-point home offices

First, you should do a quick check of the quantity of addresses needed. The branch offices require at least 120 host addresses (60 addresses times 2 branch offices), and the server farms require at least 40 host addresses (10 addresses times 4 farms). Any remaining addresses will be used for the point-to-point home offices, but this is not a hard requirement, so the basic need is for 160 (120 plus 40) addresses. This seems to be a reasonable request, because a class C has an 8-bit host field (see Figure 1-1), and an 8-bit host field with no subnetting can support up to 254 addresses (see "Calculating the Number of Host Addresses in a Subnet" earlier in this chapter). At least Widget, Inc., is not asking for the impossible; for example, it is not asking you to support 500 addresses with a single class C.

Next, tackle the largest subnets—the subnets for the branch offices. To accommodate the branch offices, you need to subnet the class C address space into chunks of at least 60 host addresses each. This is done in the following section and represents an initial round of subnetting.

Round 1 of Subnetting

To start, you create four subnets that can support 62 hosts each. You can accomplish this by applying a 26-bit subnet mask to Widget's class C. Two of the resulting subnets will be deployed for branch offices, and the other two will be subnetted further to accommodate the other requirements. The following is Widget's class C and mask (the last octet of the mask is expanded into binary to help illustrate what's happening):

Widget, Inc.'s Major Net: 192.168.1.0 (8-bit host field)
Mask for round 1: 255.255.255.**11**00-0000 (/26 mask that supports 62 hosts per subnet)

The two bits printed in boldface represent the bits that were stolen to make a 2-bit subnet field.

Table 1-4 lists the subnets created by the first round of subnetting. The two bits that make up the subnet field are printed in boldface to emphasize the distinction between the subnet bits and the host bits.

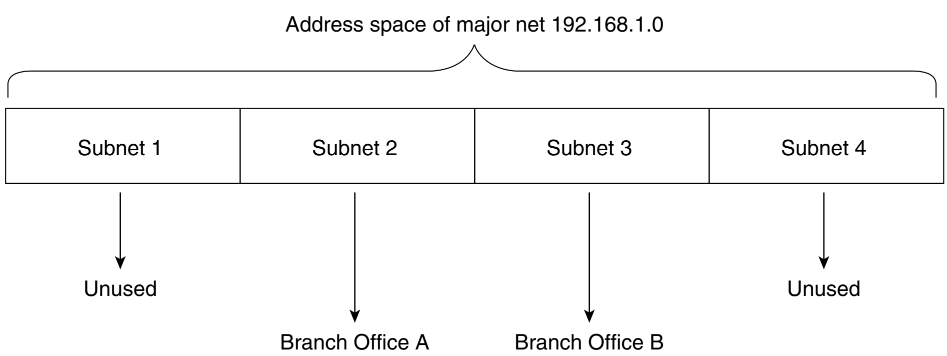
Table 1-4 *Subnets Created by the Mask for Round 1*

Name	Subnet Number in Binary (Last Octet)	Subnet Number in Decimal	Proposed Use
Subnet 1	192.168.1.**00**00-0000	192.168.1.0/26	Subnet further; see round 2
Subnet 2	192.168.1.**01**00-0000	192.168.1.64/26	Branch Office A
Subnet 3	192.168.1.**10**00-0000	192.168.1.128/26	Branch Office B
Subnet 4	192.168.1.**11**00-0000	192.168.1.192/26	Subnet further; see round 2

This first round of subnetting is nothing new—it's the same as traditional subnetting covered in "Subnetting a Classful Address Space" earlier in this chapter. Stealing two bits for the subnet field leaves six bits in the host field and yields 2^6, or 64 combinations. Subtracting the two reserved addresses for the subnet and broadcast address leaves 62 addresses for hosts. This meets Widget, Inc.'s requirement for two subnets of at least 60 hosts, so set aside Subnet 2 and Subnet 3 for the two branch offices—they are ready for deployment. Subnet 2 and Subnet 3 are selected because they are middle subnets rather than top or bottom subnets (see "The Rules on Top and Bottom Subnets" earlier in this chapter).

Figure 1-4 depicts the subnets that are set aside and unused after round 1.

Figure 1-4 *Widget, Inc.'s Address Space After Round 1 of Subnetting*



If you were doing traditional subnetting, you would now be finished, and you would have only two subnets remaining after setting aside Subnets 2 and 3. Clearly, this would not meet Widget, Inc.'s requirements, so start a second round of subnetting. This is where VLSM starts. You do not need Subnets 1 and 4 in their full size (62 host addresses), so subnet them further with a second round of subnetting and a new mask.

Round 2 of Subnetting

Perform a second round of subnetting on Subnets 1 and 4 by extending the subnet mask two bits more for a total of four bits in the mask (you are stealing two more bits from the host field and making the subnet field bigger). This further divides Subnets 1 and 4 into multiple smaller subnets.

The following is the second round of subnetting for Subnet 1. The bits printed in boldface represent the expanded subnet field (now a 4-bit field):

Subnet 1: 192.168.1.0/26 (6-bit host field)
Mask for round 2: 255.255.255.**1111**-0000 (/28 mask that supports 14 hosts per subnet)

Table 1-5 lists the new subnets created out of Subnet 1 by a second round of subnetting. For clarity, the new subnets are named Subnet 1.*x*, where *x* represents a piece of the original Subnet 1. As before, the bits that make up the subnet field are printed in boldface to emphasize the distinction between the subnet bits and the host bits. The new bits that expanded the subnet field are underlined.

Table 1-5 *Subnets Created by the Mask for Round 2 When Applied to Subnet 1*

Name	Binary (Last Octet)	Decimal	Proposed Use
Subnet 1.1	192.168.1.**00<u>00</u>**-0000	192.168.1.0/28	Subnet further; see round 3
Subnet 1.2	192.168.1.**00<u>01</u>**-0000	192.168.1.16/28	Server Farm A
Subnet 1.3	192.168.1.**00<u>10</u>**-0000	192.168.1.32/28	Server Farm B
Subnet 1.4	192.168.1.**00<u>11</u>**-0000	192.168.1.48/28	Server Farm C

NOTE Subnet 1's first two subnet bits are 00, as defined by the first round of subnetting. It is very important not to alter these two bits—any change to the 00 bits means you are no longer working with Subnet 1.

Now, perform a second round of subnetting on Subnet 4 with the same /28 mask:

Subnet 4: 192.168.1.192/26 (6-bit host field)
Mask for round 2: 255.255.255.**1111**-0000 (/28 mask that supports 14 hosts per subnet)

Table 1-6 lists the new subnets created out of Subnet 4 by a second round of subnetting. For clarity, the new subnets are named Subnet 4.*x*, where *x* represents a piece of the original Subnet 4. The new bits that expanded the subnet field are underlined.

Table 1-6 *Subnets Created by the Mask for Round 2 When Applied to Subnet 4*

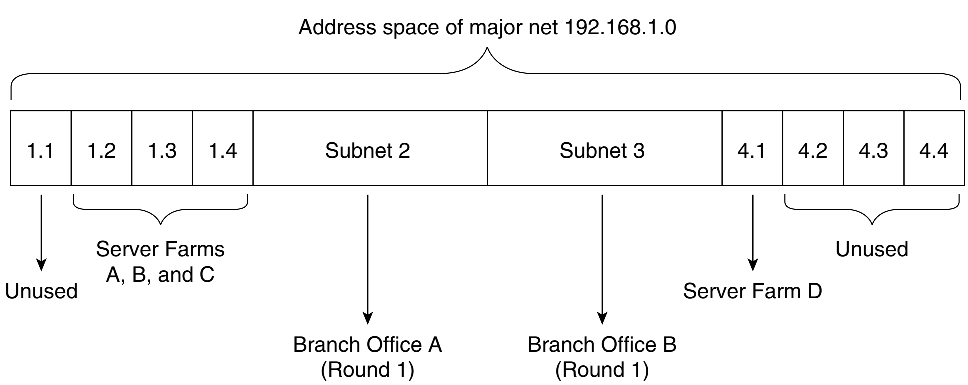
Name	Binary (Last Octet)	Decimal	Proposed Use
Subnet 4.1	192.168.1.**1100**-0000	192.168.1.192/28	Server Farm D
Subnet 4.2	192.168.1.**1101**-0000	192.168.1.208/28	Subnet further; see round 3
Subnet 4.3	192.168.1.**1110**-0000	192.168.1.224/28	Subnet further; see round 3
Subnet 4.4	192.168.1.**1111**-0000	192.168.1.240/28	Subnet further; see round 3

This second round of subnetting yields eight more subnets—eight additional subnets for Widget, Inc., out of the same address space. Each of the eight subnets (1.1 through 1.4 and 4.1 through 4.4) can support up to 14 hosts. This meets Widget, Inc.'s requirement for the server farm subnets. Widget, Inc., needs four of these subnets, so set aside Subnets 1.2, 1.3, 1.4, and 4.1 for the four server farms.

Avoid using Subnets 1.1 and 4.4, because they are the bottom and top subnets in the major net. You can deploy them if you are certain that hosts and networking devices in Widget, Inc.'s network are not affected by the caveats about using the top and bottom subnets discussed earlier.

Figure 1-5 depicts the subnets that are set aside and still unused after round 2.

Figure 1-5 *Widget, Inc.'s Address Space After Round 2 of Subnetting*



Round 3 of Subnetting

The unused subnets from round 2 can be used to satisfy Widget, Inc.'s requirement for the home office subnets (two hosts each), so now perform a third and final round of subnetting. Extend the mask from the last round by two more bits for a total of 6 bits in the mask. This further divides the unused subnets (1.1, 4.2, 4.3, and 4.4) into smaller, two-host subnets.

The following is the third round of subnetting applied to the unused Subnet 4.2 (from round 2). The bits printed in boldface represent the expanded subnet field (now a 6-bit field):

Subnet 4.2: 192.168.1.208/28 (4-bit host field)
Mask for round 3: 255.255.255.**1111-11**00 (/30 mask that supports two hosts per subnet)

Table 1-7 lists the new subnets created out of Subnet 4.2 by a third round of subnetting. For clarity, the new subnets are named Subnet 4.2.*x*, where *x* represents a piece of the Subnet 4.2. As before, the bits that make up the subnet field are printed in boldface to emphasize the distinction between the subnet bits and the host bits. The new bits that expanded the subnet field are underlined.

Table 1-7 *Subnets Created by the Mask for Round 3 When Applied to Subnet 4.2*

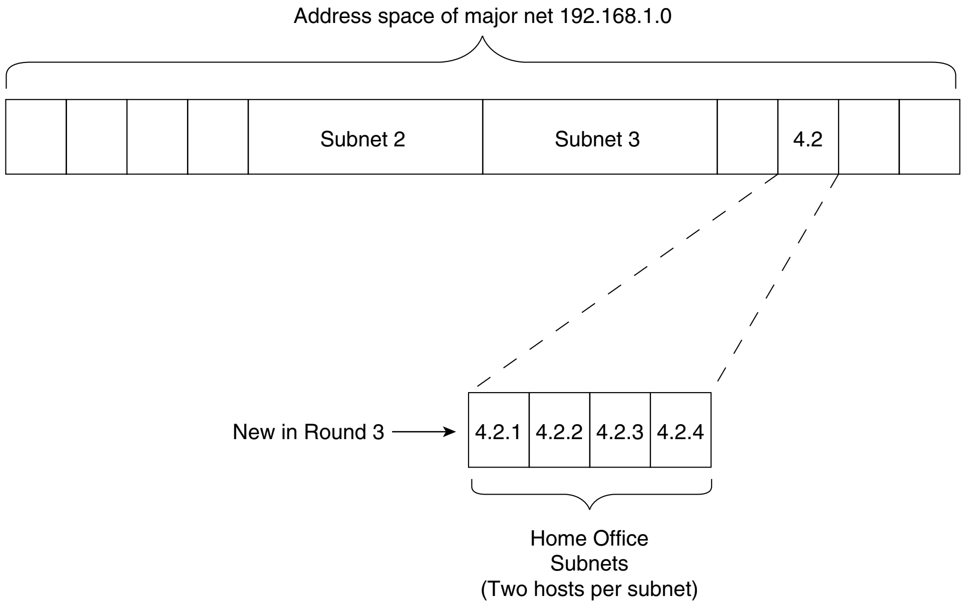
Name	Binary (Last Octet)	Decimal	Proposed Use
Subnet 4.2.1	192.168.1.**1101**-<u>**00**</u>00	192.168.1.208/30	Home Office
Subnet 4.2.2	192.168.1.**1101**-<u>**01**</u>00	192.168.1.212/30	Home Office
Subnet 4.2.3	192.168.1.**1101**-<u>**10**</u>00	192.168.1.216/30	Home Office
Subnet 4.2.4	192.168.1.**1101**-<u>**11**</u>00	192.168.1.220/30	Home Office

NOTE Subnet 4.2's first four subnet bits are 1101, as defined by the second round of subnetting. It is very important not to alter these four bits—any change to the 1101 bits means you are no longer working with Subnet 4.2.

This third round of subnetting uses a /30 mask and creates four smaller subnets out of Subnet 4.2. A subnet with a /30 mask can support only two hosts—perfect for Widget, Inc.'s home offices that connect over point-to-point links.

Figure 1-6 depicts the subnets created after subnetting Subnet 4.2 with the mask from round 3 (/30 mask).

Figure 1-6 *Subnet 4.2 After the Third Round of Subnetting*



Widget, Inc., wants to use all of the unused address space from round 2 for home offices, so with Subnet 4.2 complete (Table 1-7), simply repeat the third round of subnetting. That is, apply the same /30 mask to the other unused subnets from round 2: Subnets 1.1, 4.3, and 4.4. This results in a total of 16 two-host subnets for home offices, as summarized by Table 1-8.

Table 1-8 *A Summary of the Subnets Created by Round 3*

Name	Binary (Last Octet)	Subnet
1.1.1	192.168.1.**0000**-**00**00	192.168.1.0/30
1.1.2	192.168.1.**0000**-**01**00	192.168.1.4/30
1.1.3	192.168.1.**0000**-**10**00	192.168.1.8/30
1.1.4	192.168.1.**0000**-**11**00	192.168.1.12/30
4.2.1	192.168.1.**1101**-**00**00	192.168.1.208/30
4.2.2	192.168.1.**1101**-**01**00	192.168.1.212/30
4.2.3	192.168.1.**1101**-**10**00	192.168.1.216/30
4.2.4	192.168.1.**1101**-**11**00	192.168.1.220/30
4.3.1	192.168.1.**1110**-**00**00	192.168.1.224/30
4.3.2	192.168.1.**1110**-**01**00	192.168.1.228/30
4.3.3	192.168.1.**1110**-**10**00	192.168.1.232/30

Table 1-8 *A Summary of the Subnets Created by Round 3 (Continued)*

Name	Binary (Last Octet)	Subnet
4.3.4	192.168.1.**1110**-**11**00	192.168.1.236/30
4.4.1	192.168.1.**1111**-**00**00	192.168.1.240/30
4.4.2	192.168.1.**1111**-**01**00	192.168.1.244/30
4.4.3	192.168.1.**1111**-**10**00	192.168.1.248/30
4.4.4	192.168.1.**1111**-**11**00	192.168.1.252/30

As in the earlier rounds, you still have a top and bottom subnet after round 3; they are 192.168.1.252/30. and 192.168.1.0/30. Although these are generally not deployable, they are small two-host subnets, so you are wasting just a few addresses out of the entire major net space. The third-round VLSM process has effectively reduced the wasted address space from 128 addresses in round 1 (where Subnet 4 and Subnet 1 were the top and bottom subnets) to just 8 addresses in round 3 (where Subnets 4.4.4 and 1.1.1 are the top and bottom subnets). This represents significantly better use of the address space over fixed-length subnet masks.

Final VLSM Results for Widget, Inc.

After the third round of subnetting, you cannot use VLSM to subnet any further—a two-host subnet is the smallest you can make. The totals from all three rounds are listed in Table 1-9.

Table 1-9 *Final Results of Subnetting for Widget, Inc.*

Round	Subnets Created	Subnets Set Aside	Maximum Hosts per Subnet
1	4	2	62
2	8	4	14
3	16	14 (2 wasted)	2

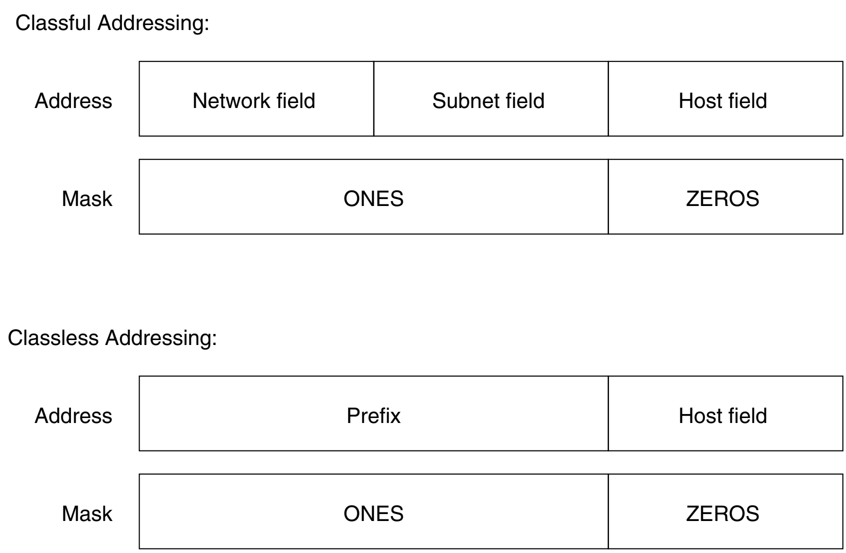
The VLSM process yields a total of 20 deployable subnets of three different sizes and meets the stated requirements of Widget, Inc.

NOTE RFC 1219 describes a VLSM subnetting strategy that allows subnets to grow in size after they are deployed and also avoids address changes. See Appendix A for information on how to retrieve RFCs.

Overview of Classless Addressing

Classless addressing (described in RFC 1519) abolishes the idea of traditional classes A, B, and C major nets and the notion of a subnet field. Subnets and major nets do not exist in a classless world; instead, there is only a network prefix and a host field. Figure 1-7 describes the difference between classful and classless addressing.

Figure 1-7 *Classful Versus Classless Addressing*



The length of the network prefix is determined by a prefix mask. The prefix mask is a contiguous series of ones that starts with the left-most bit (the most significant bit). Although the prefix mask looks like a subnet mask, it's important to realize that there is no subnet field.

An advantage of classless addressing is the capability to combine what were multiple class C addresses into a contiguous block of addresses called a *supernet* or classless interdomain routing (CIDR) block. Figure 1-8 describes an address space in two ways: as four class C major nets (classful sense) and as one supernet (classless sense).