



CCNA

200-301

Portable Command Guide

All the CCNA 200-301 commands in one compact, portable resource

Fifth Edition

ciscopress.com

Scott Empson

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



CCNA 200-301 Portable Command Guide, Fifth Edition

Scott Empson

Copyright© 2020 Cisco Systems, Inc.

Published by:

Cisco Press

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

ScoutAutomatedPrintCode

Library of Congress Control Number: 2019951511

ISBN-13: 978-0-13-593782-2

ISBN-10: 0-13-593782-5

Warning and Disclaimer

This book is designed to provide information about the Cisco Certified Network Associate (CCNA) exam (200-301). 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 authors, 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.

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided “as is” without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability, whether express, implied or statutory, fitness for a particular purpose, title and non-infringement. In no event shall Microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services.

Editor-in-Chief

Mark Taub

Alliances Manager, Cisco Press

Arezou Gol

Product Line Manager

Brett Bartow

Senior Editor

James Manly

Managing Editor

Sandra Schroeder

Development Editor

Ellie Bru

Senior Project Editor

Tonya Simpson

Copy Editor

Bill McManus

Technical Editor

Rick McDonald

Editorial Assistant

Cindy Teeters

Cover Designer

Chuti Prasertsith

Composition

codeMantra

Indexer

Lisa Stumpf

Proofreader

Abigail Bass

Contents

Introduction xix

Part I: Network Fundamentals

CHAPTER 1	IPv4 Addressing—How It Works	1
	What Are IPv4 Addresses Used For?	1
	What Does an IPv4 Address Look Like?	2
	Network and Subnetwork Masks	2
	Ways to Write a Network or Subnet Mask	3
	Network, Node, and Broadcast Addresses	3
	Classes of IPv4 Addresses	4
	Network vs. Node (Host) Bits	5
	RFC (Private) 1918 Addresses	6
	Local vs. Remote Addresses	7
	Classless Addressing	7
	Lessons Learned	9
CHAPTER 2	How to Subnet IPv4 Addresses	11
	Subnetting a Class C Network Using Binary	12
	Subnetting a Class B Network Using Binary	15
	Binary ANDing	17
	So Why AND?	19
	Shortcuts in Binary ANDing	20
CHAPTER 3	Variable Length Subnet Masking (VLSM)	23
	IP Subnet Zero	23
	VLSM Example	24
	Step 1: Determine How Many H Bits Will Be Needed to Satisfy the Largest Network	25
	Step 2: Pick a Subnet for the Largest Network to Use	25
	Step 3: Pick the Next Largest Network to Work With	26
	Step 4: Pick the Third Largest Network to Work With	28
	Step 5: Determine Network Numbers for Serial Links	30
CHAPTER 4	Route Summarization	33
	Example for Understanding Route Summarization	33
	Step 1: Summarize Winnipeg's Routes	34
	Step 2: Summarize Calgary's Routes	35

	Step 3: Summarize Edmonton's Routes	35
	Step 4: Summarize Vancouver's Routes	36
	Route Summarization and Route Flapping	38
	Requirements for Route Summarization	38
CHAPTER 5	IPv6 Addressing—How It Works	39
	IPv6: A Very Brief Introduction	39
	What Does an IPv6 Address Look Like?	40
	Reducing the Notation of an IPv6 Address	41
	Rule 1: Omit Leading 0s	41
	Rule 2: Omit All-0s Hextet	42
	Combining Rule 1 and Rule 2	42
	Prefix Length Notation	43
	IPv6 Address Types	44
	Unicast Addresses	45
	Multicast Addresses	48
	Anycast Addresses	50
CHAPTER 6	Cables and Connections	51
	Connecting a Rollover Cable to Your Router or Switch	51
	Using a USB Cable to Connect to Your Router or Switch	51
	Terminal Settings	52
	LAN Connections	53
	Serial Cable Types	53
	Which Cable to Use?	55
	ANSI/TIA Cabling Standards	56
	T568A Versus T568B Cables	57
CHAPTER 7	The Command-Line Interface	59
	Shortcuts for Entering Commands	59
	Using the <code>Tab</code> Key to Complete Commands	60
	Console Error Messages	60
	Using the Question Mark for Help	60
	enable Command	61
	exit Command	61
	end Command	61
	disable Command	61
	logout Command	62
	Setup Mode	62
	Keyboard Help	62

History Commands	63
terminal Commands	64
show Commands	64
Using the Pipe Parameter () with the show or more Commands	64
Using the no and default Forms of Commands	66

Part II: LAN Switching Technologies

CHAPTER 8 Configuring a Switch 67

Help Commands	68
Command Modes	68
Verifying Commands	68
Resetting Switch Configuration	69
Setting Host Names	69
Setting Passwords	69
Setting IP Addresses and Default Gateways	70
Setting Interface Descriptions	70
The mdix auto Command	70
Setting Duplex Operation	71
Setting Operation Speed	71
Managing the MAC Address Table	72
Configuration Example	72

CHAPTER 9 VLANs 75

Creating Static VLANs	75
Creating Static VLANs Using VLAN Configuration Mode	75
Assigning Ports to VLANs	76
Using the range Command	76
Configuring a Voice VLAN	76
Configuring Voice and Data with Trust	77
Configuring Voice and Data Without Trust	78
Verifying VLAN Information	78
Saving VLAN Configurations	79
Erasing VLAN Configurations	79
Configuration Example: VLANs	80
2960 Switch	80

CHAPTER 10 VLAN Trunking Protocol and Inter-VLAN Communication 83

Dynamic Trunking Protocol (DTP)	83
Setting the VLAN Encapsulation Type	84

VLAN Trunking Protocol (VTP)	84
Verifying VTP	86
Inter-VLAN Communication Using an External Router: Router-on-a-Stick	87
Inter-VLAN Communication on a Multilayer Switch Through a Switch Virtual Interface	88
Removing L2 Switchport Capability of an Interface on an L3 Switch	88
Configuring Inter-VLAN Communication on an L3 Switch	88
Inter-VLAN Communication Tips	88
Configuration Example: Inter-VLAN Communication	89
ISP Router	89
CORP Router	90
L2Switch2 (Catalyst 2960)	92
L3Switch1 (Catalyst 3560/3650/3750)	94
L2Switch1 (Catalyst 2960)	95
CHAPTER 11 Spanning Tree Protocol	97
Spanning Tree Protocol Definition	97
Enabling Spanning Tree Protocol	98
Changing the Spanning-Tree Mode	99
BPDU Guard (3650/9xxx Series)	99
Configuring the Root Switch	100
Configuring a Secondary Root Switch	100
Configuring Port Priority	100
Configuring the Path Cost	101
Configuring the Switch Priority of a VLAN	101
Configuring STP Timers	102
Configuring Optional Spanning-Tree Features	102
PortFast	102
BPDU Guard (2xxx/Older 3xxx Series)	103
Enabling the Extended System ID	103
Verifying STP	104
Troubleshooting Spanning Tree Protocol	104
Configuration Example: PVST+	104
Core Switch (3650)	105
Distribution 1 Switch (3650)	106
Distribution 2 Switch (3650)	106
Access 1 Switch (2960)	107
Access 2 Switch (2960)	107

Spanning-Tree Migration Example: PVST+ to Rapid-PVST+	108
Access 1 Switch (2960)	108
Access 2 Switch (2960)	108
Distribution 1 Switch (3650)	109
Distribution 2 Switch (3650)	109
Core Switch (3650)	109

CHAPTER 12 EtherChannel 111

EtherChannel	111
Interface Modes in EtherChannel	111
Default EtherChannel Configuration	112
Guidelines for Configuring EtherChannel	112
Configuring Layer 2 EtherChannel	113
Configuring Layer 3 EtherChannel	114
Configuring EtherChannel Load Balancing	114
Configuring LACP Hot-Standby Ports	115
Monitoring and Verifying EtherChannel	116
Configuration Example: EtherChannel	117
DLSwitch (3560 or 9300)	117
ALSwitch1 (2960 or 9200)	118
ALSwitch2 (2960 or 9200)	119

CHAPTER 13 Cisco Discovery Protocol (CDP) and Link Layer Discovery Protocol (LLDP) 121

Cisco Discovery Protocol	121
Configuring CDP	121
Verifying and Troubleshooting CDP	122
CDP Design Tips	122
Link Layer Discovery Protocol (802.1AB)	123
Configuring LLDP (802.1AB)	123
Verifying and Troubleshooting LLDP	124

Part III: Routing Technologies

CHAPTER 14 Configuring a Cisco Router 125

Router Modes	126
Entering Global Configuration Mode	126
Configuring a Router Name	126
Configuring Passwords	126
Password Encryption	127

Interface Names	127
Moving Between Interfaces	131
Configuring a Serial Interface	132
Assigning an IPv4 Address to a Fast Ethernet Interface	132
Assigning an IPv4 Address to a Gigabit Ethernet Interface	132
Assigning IPv6 Addresses to Interfaces	133
Creating a Message-of-the-Day Banner	133
Creating a Login Banner	134
Mapping a Local Host Name to a Remote IP Address	134
The no ip domain-lookup Command	134
Working with DNS on a Router	134
The logging synchronous Command	135
The exec-timeout Command	136
Saving Configurations	136
Erasing Configurations	136
The write Command	137
Verifying Your Configurations Using show Commands	137
EXEC Commands in Configuration Mode: The do Command	138
Configuration Example: Basic Router Configuration	138
Boston Router	138
CHAPTER 15 Static Routing	141
Configuring an IPv4 Static Route	141
Static Routes and Recursive Lookups	142
The permanent Keyword	142
Floating Static Routes in IPv4 and Administrative Distance	143
Configuring an IPv4 Default Route	144
Verifying IPv4 Static Routes	144
Configuration Example: IPv4 Static Routes	144
Ketchikan Router	145
Juneau Router	145
Sitka Router	146
Configuring an IPv6 Static Route	146
Floating Static Routes in IPv6	147
Configuring an IPv6 Default Route	147
Verifying IPv6 Static Routes	147
CHAPTER 16 Open Shortest Path First (OSPF)	149
OSPFv2 Versus OSPFv3	149
Configuring OSPF	150

- Using Wildcard Masks with OSPF Areas 150
- Loopback Interfaces 152
- Router ID 152
- DR/BDR Elections 153
- Timers 153
- Verifying OSPFv2 Configurations 153
- Troubleshooting OSPFv2 154
- Configuration Example: Single-Area OSPF 154
 - Austin Router 155
 - Houston Router 156
 - Galveston Router 157

Part IV: IP Services

CHAPTER 17 DHCP 159

- Configuring a DHCP Server on an IOS Router 159
- Using Cisco IP Phones with a DHCP Server 160
- Verifying and Troubleshooting DHCP Configuration 160
- Configuring a DHCP Helper Address 161
- Configuring a DHCP Client on a Cisco IOS Software Ethernet Interface 162
- Configuration Example: DHCP 162
 - Edmonton Router 162
 - Gibbons Router 164

CHAPTER 18 Network Address Translation (NAT) 165

- Private IP Addresses: RFC 1918 165
- Configuring Dynamic NAT: One Private to One Public Address Translation 165
- Configuring PAT: Many Private to One Public Address Translation 167
- Configuring Static NAT: One Private to One Permanent Public Address Translation 169
- Verifying NAT and PAT Configurations 170
- Troubleshooting NAT and PAT Configurations 171
- Configuration Example: PAT 171
 - ISP Router 171
 - Company Router 172

CHAPTER 19 Configuring Network Time Protocol (NTP) 175

- NTP Configuration 175
- NTP Design 176

Securing NTP	177
Enabling NTP Authentication	177
Limiting NTP Access with Access Lists	178
Verifying and Troubleshooting NTP	178
Setting the Clock on a Router	179
Using Time Stamps	182
Configuration Example: NTP	182
Core1 Router	183
Core2 Router	184
DLSwitch1	185
DLSwitch2	185
ALSwitch1	186
ALSwitch2	186

Part V: Security Fundamentals

CHAPTER 20	Layer Two Security Features	187
	Setting Passwords on a Switch	187
	Configuring Static MAC Addresses	188
	Configuring Switch Port Security	188
	Configuring Sticky MAC Addresses	189
	Verifying Switch Port Security	189
	Recovering Automatically from Error-Disabled Ports	190
	Verifying Autorecovery of Error-Disabled Ports	190
	Configuring DHCP Snooping	191
	Verifying DHCP Snooping	192
	Configuring Dynamic ARP Inspection (DAI)	193
	Verifying Dynamic ARP Inspection	193
	Configuration Example: Switch Security	194
CHAPTER 21	Managing Traffic Using Access Control Lists (ACLs)	197
	Access List Numbers	197
	Using Wildcard Masks	198
	ACL Keywords	198
	Creating Standard ACLs	198
	Applying Standard ACLs to an Interface	199
	Verifying ACLs	200
	Removing ACLs	200
	Creating Extended ACLs	200
	Applying Extended ACLs to an Interface	201

- The established Keyword 201
- The log Keyword 202
- Creating Named ACLs 203
- Using Sequence Numbers in Named ACLs 203
- Removing Specific Lines in Named ACLs Using Sequence Numbers 204
- Sequence Number Tips 204
- Including Comments About Entries in ACLs 205
- Restricting Virtual Terminal Access 206
- Tips for Configuring ACLs 206
- IPv6 ACLs 207
- Verifying IPv6 ACLs 207
- Configuration Examples: IPv4 ACLs 208
- Configuration Examples: IPv6 ACLs 210

CHAPTER 22 Device Monitoring and Hardening 213

- Device Monitoring 213
- Configuration Backups 213
- Implementing Logging 214
 - Configuring Syslog 215
 - Syslog Message Format 215
 - Syslog Severity Levels 216
 - Syslog Message Example 216
- Device Hardening 217
 - Configuring Passwords 217
 - Password Encryption 218
 - Password Encryption Algorithm Types 218
 - Configuring SSH 219
 - Verifying SSH 220
 - Restricting Virtual Terminal Access 220
 - Disabling Unneeded Services 221

Part VI: Wireless Technologies

CHAPTER 23 Configuring and Securing a WLAN AP 223

- Initial Setup of a Wireless LAN Controller (WLC) 223
- Monitoring the WLC 229
- Configuring a VLAN (Dynamic) Interface 230
- Configuring a DHCP Scope 234
- Configuring a WLAN 237

Defining a RADIUS Server	239
Exploring Management Options	242
Configuring a WLAN Using WPA2 PSK	246

Part VII: Appendices

APPENDIX A	How to Count in Decimal, Binary, and Hexadecimal	251
	How to Count in Decimal	251
	How to Count in Binary	253
	How to Count in Hexadecimal	254
	Representing Decimal, Binary, and Hexadecimal Numbers	256
APPENDIX B	How to Convert Between Number Systems	259
	How to Convert from Decimal to Binary	259
	How to Convert from Binary to Decimal	260
	How to Convert from Decimal IP Addresses to Binary and from Binary IP Addresses to Decimal	261
	A Bit of Perspective	262
	How to Convert from Hexadecimal to Binary	262
	How to Convert from Binary to Hexadecimal	263
	How to Convert from Decimal to Hexadecimal	264
	How to Convert from Hexadecimal to Decimal	265
APPENDIX C	Binary/Hex/Decimal Conversion Chart	267
APPENDIX D	Create Your Own Journal Here	275
INDEX		277

Route Summarization

This chapter provides information concerning the following topics:

- Example for understanding route summarization
- Route summarization and route flapping
- Requirements for route summarization

Route summarization, or supernetting, is needed to reduce the number of routes that a router advertises to its neighbor. Remember that for every route you advertise, the size of your update grows. It has been said that if there were no route summarization, the Internet backbone would have collapsed from the sheer size of its own routing tables back in 1997!

Routing updates, whether done with a distance-vector protocol or a link-state protocol, grow with the number of routes you need to advertise. In simple terms, a router that needs to advertise ten routes needs ten specific lines in its update packet. The more routes you have to advertise, the bigger the packet. The bigger the packet, the more bandwidth the update takes, reducing the bandwidth available to transfer data. But with route summarization, you can advertise many routes with only one line in an update packet. This reduces the size of the update, allowing you more bandwidth for data transfer.

Also, when a new data flow enters a router, the router must do a lookup in its routing table to determine which interface the traffic must be sent out. The larger the routing tables, the longer this takes, leading to more used router CPU cycles to perform the lookup. Therefore, a second reason for route summarization is that you want to minimize the amount of time and router CPU cycles that are used to route traffic.

NOTE: This example is a very simplified explanation of how routers send updates to each other. For a more in-depth description, I highly recommend you go out and read Jeff Doyle and Jennifer Carroll's book *Routing TCP/IP, Volume I*, Second Edition (Cisco Press, 2005). This book has been around for many years and is considered by most to be the authority on how the different routing protocols work. If you are considering continuing on in your certification path to try and achieve the CCIE, you need to buy Doyle's book—and memorize it; it's that good.

Example for Understanding Route Summarization

Refer to Figure 4-1 to assist you as you go through the following explanation of an example of route summarization.

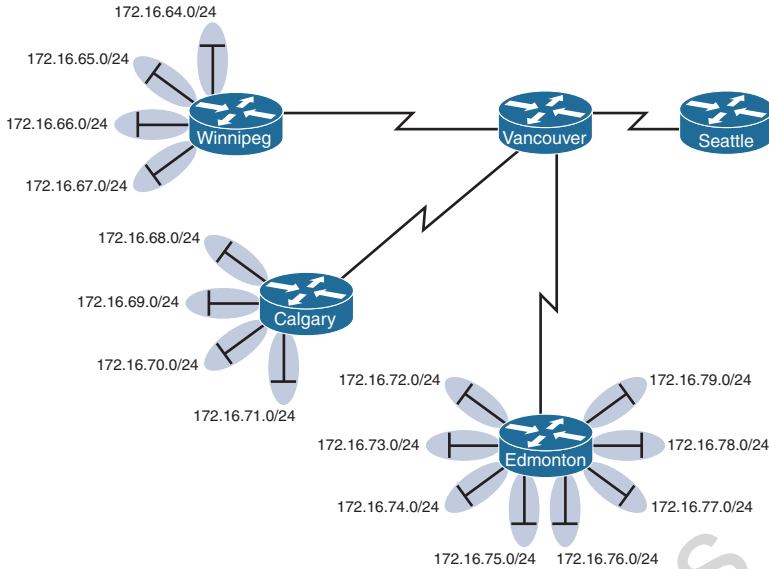


Figure 4-1 Four-City Network Without Route Summarization

As you can see from Figure 4-1, Winnipeg, Calgary, and Edmonton each have to advertise internal networks to the main router located in Vancouver. Without route summarization, Vancouver would have to advertise 16 networks to Seattle. You want to use route summarization to reduce the burden on this upstream router.

Step 1: Summarize Winnipeg’s Routes

To do this, you need to look at the routes in binary to see if there are any specific bit patterns that you can use to your advantage. What you are looking for are common bits on the network side of the addresses. Because all of these networks are /24 networks, you want to see which of the first 24 bits are common to all four networks.

172.16.64.0 = **10101100.00010000.01000000.00000000**

172.16.65.0 = **10101100.00010000.01000001.00000000**

172.16.66.0 = **10101100.00010000.01000010.00000000**

172.16.67.0 = **10101100.00010000.01000011.00000000**

Common bits: **10101100.00010000.010000xx**

You see that the first 22 bits of the four networks are common. Therefore, you can summarize the four routes by using a subnet mask that reflects that the first 22 bits are common. This is a /22 mask, or 255.255.252.0. You are left with the summarized address of

172.16.64.0/22

This address, when sent to the upstream Vancouver router, will tell Vancouver: “If you have any packets that are addressed to networks that have the first 22 bits in the pattern of 10101100.00010000.010000xx.xxxxxxxx, then send them to me here in Winnipeg.”

By sending one route to Vancouver with this supernetted subnet mask, you have advertised four routes in one line instead of using four lines. Much more efficient!

Step 2: Summarize Calgary's Routes

For Calgary, you do the same thing that you did for Winnipeg—look for common bit patterns in the routes:

172.16.68.0 = **10101100.00010000.010001**100.00000000

172.16.69.0 = **10101100.00010000.010001**101.00000000

172.16.70.0 = **10101100.00010000.010001**110.00000000

172.16.71.0 = **10101100.00010000.010001**111.00000000

Common bits: **10101100.00010000.010001**xx

Once again, the first 22 bits are common. The summarized route is therefore

172.16.68.0/22

Step 3: Summarize Edmonton's Routes

For Edmonton, you do the same thing that you did for Winnipeg and Calgary—look for common bit patterns in the routes:

172.16.72.0 = **10101100.00010000.01001**1000.00000000

172.16.73.0 = **10101100.00010000.01001**1001.00000000

172.16.74.0 = **10101100.00010000.01001**1010.00000000

172.16.75.0 = **10101100.00010000.01001**1011.00000000

172.16.76.0 = **10101100.00010000.01001**1100.00000000

172.16.77.0 = **10101100.00010000.01001**1101.00000000

172.16.78.0 = **10101100.00010000.01001**1110.00000000

172.16.79.0 = **10101100.00010000.01001**1111.00000000

Common bits: **10101100.00010000.01001**xxx

For Edmonton, the first 21 bits are common. The summarized route is therefore

172.16.72.0/21

Figure 4-2 shows what the network looks like, with Winnipeg, Calgary, and Edmonton sending their summarized routes to Vancouver.

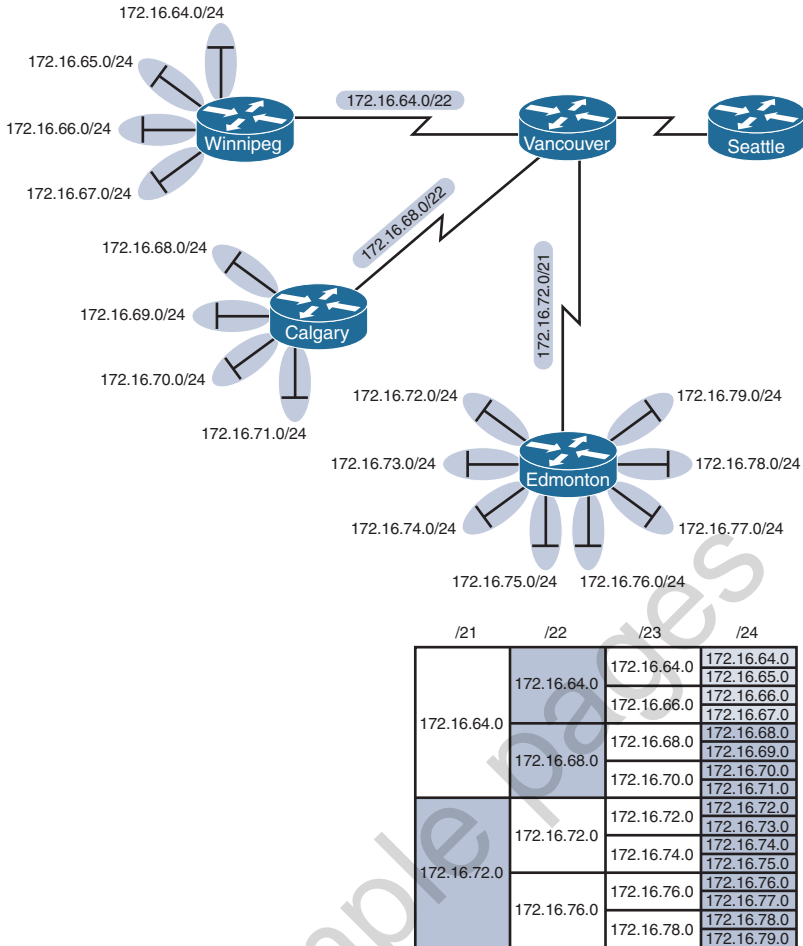


Figure 4-2 Four-City Network with Edge Cities Summarizing Routes

Step 4: Summarize Vancouver's Routes

Yes, you can summarize Vancouver's routes to Seattle. You continue in the same format as before. Take the routes that Winnipeg, Calgary, and Edmonton sent to Vancouver, and look for common bit patterns:

172.16.64.0 = 10101100.00010000.01000000.00000000

172.16.68.0 = 10101100.00010000.01000100.00000000

172.16.72.0 = 10101100.00010000.01001000.00000000

Common bits: 10101100.00010000.0100xxxx

Because there are 20 bits that are common, you can create one summary route for Vancouver to send to Seattle:

172.16.64.0/20

Vancouver has now told Seattle that in one line of a routing update, 16 different networks are being advertised. This is much more efficient than sending 16 lines in a routing update to be processed.

Figure 4-3 shows what the routing updates would look like with route summarization taking place.

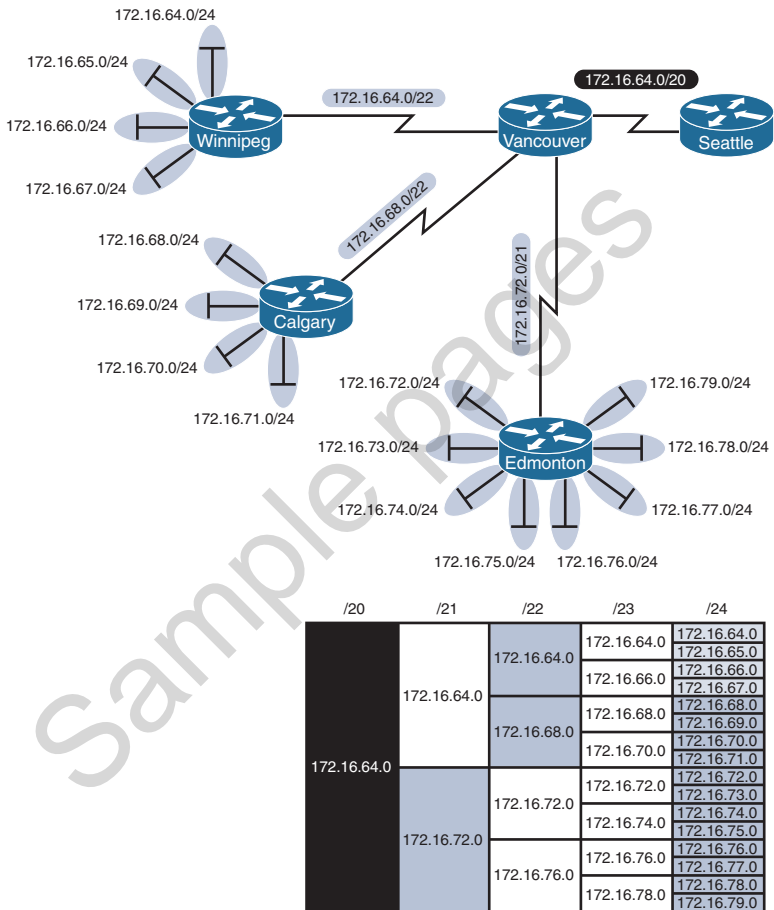


Figure 4-3 Four-City Network with Complete Route Summarization

Route Summarization and Route Flapping

Another positive aspect of route summarization has to do with route flapping. *Route flapping* is when a network, for whatever reason (such as interface hardware failure or misconfiguration), goes up and down on a router, causing that router to constantly advertise changes about that network. Route summarization can help insulate upstream neighbors from these problems.

Consider router Edmonton from Figure 4-1. Suppose that network 172.16.74.0/24 goes down. Without route summarization, Edmonton would advertise Vancouver to remove that network. Vancouver would forward that same message upstream to Calgary, Winnipeg, Seattle, and so on. Now assume the network comes back online a few seconds later. Edmonton would have to send another update informing Vancouver of the change. Each time a change needs to be advertised, the router must use CPU resources. If that route were to flap, the routers would constantly have to update their own tables, as well as advertise changes to their neighbors. In a CPU-intensive protocol such as OSPF, the constant hit on the CPU might make a noticeable change to the speed at which network traffic reaches its destination.

Route summarization enables you to avoid this problem. Even though Edmonton would still have to deal with the route constantly going up and down, no one else would notice. Edmonton advertises a single summarized route, 172.16.72.0/21, to Vancouver. Even though one of the networks is going up and down, this does not invalidate the route to the other networks that were summarized. Edmonton will deal with its own route flap, but Vancouver will be unaware of the problem downstream in Edmonton. Summarization can effectively protect or insulate other routers from route flaps.

Requirements for Route Summarization

To create route summarization, there are some necessary requirements:

- Routers need to be running a classless routing protocol, as they carry subnet mask information with them in routing updates. (Examples are RIP v2, OSPF, EIGRP, IS-IS, and BGP.)
- Addresses need to be assigned in a hierarchical fashion for the summarized address to have the same high-order bits. It does no good if Winnipeg has network 172.16.64.0 and 172.16.67.0 while 172.16.65.0 resides in Calgary and 172.16.66.0 is assigned in Edmonton. No summarization could take place from the edge routers to Vancouver.

TIP: Because most networks use NAT and the RFC 10.0.0.0/8 network internally, it is important when creating your network design that you assign network subnets in a way that they can be easily summarized. A little more planning now can save you a lot of grief later.

IPv6 Addressing—How It Works

This chapter provides information concerning the following topics:

- IPv6: A very brief introduction
- What does an IPv6 address look like?
- Reducing the notation of an IPv6 address
 - Rule 1: Omit leading 0s
 - Rule 2: Omit all-0s hextet
 - Combining rule 1 and rule 2
- Prefix length notation
- IPv6 address types
 - Unicast addresses
 - Global unicast
 - Link-local
 - Loopback
 - Unspecified
 - Unique local
 - IPv4 embedded
 - Multicast addresses
 - Well-known
 - Solicited-node
 - Anycast addresses

NOTE: This chapter is meant to be a very high-level overview of IPv6 addressing. For an excellent overview of IPv6, I strongly recommend you read Rick Graziani's book from Cisco Press: *IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6*, Second Edition. It is a brilliant read, and Rick is an amazing author. I am also very fortunate to call him a friend.

IPv6: A Very Brief Introduction

When IPv4 became a standard in 1980, its 32-bit address field created a theoretical maximum of approximately 4.29 billion addresses (2^{32}). IPv4 was originally conceived as an experiment, and not for a practical implementation, so 4.29 billion was considered to be an inexhaustible amount. But with the growth of the Internet, and the need for individuals and companies to require multiple addresses—your home PC, your cell

phone, your tablet, your PC at work/school, your Internet-aware appliances—you can see that something larger than 32-bit address fields would be required. In 1993, the Internet Engineering Task Force (IETF) formed a working group called the IP Next Generation working group. In 1994 the IETF recommended an address size of 128 bits. While many people think that IPv6 is just a way to create more addresses, there are actually many enhancements that make IPv6 a superior choice to IPv4. Again, I recommend Rick Graziani’s *IPv6 Fundamentals* as a must-have on your bookshelf for working with IPv6.

What Does an IPv6 Address Look Like?

The way that a computer or other digital device sees an IPv6 address and the way humans see an IPv6 address are different. A digital device looks at an IPv6 address as a 128-bit number. But humans have devised a way to convert this 128-bit number into something easier to look at and work with. For humans, an IPv6 address is a 128-bit number that is written as a string of hexadecimal digits. Hexadecimal is a natural fit for IPv6 addresses because any 4 bits can be represented as a single hexadecimal digit. Two hexadecimal digits represent a single byte, or octet (8 bits). The preferred form of an IPv6 address is `x:x:x:x:x:x:x:x`, where each `x` is a 16-bit section that can be represented using up to four hexadecimal digits. Each section is separated by a colon (:), as opposed to IPv4 addressing, which uses a period (.) to separate each section. The result is eight 16-bit sections (sometimes called *hexets*) for a total of 128 bits in the address.

Figure 5-1 shows this format.

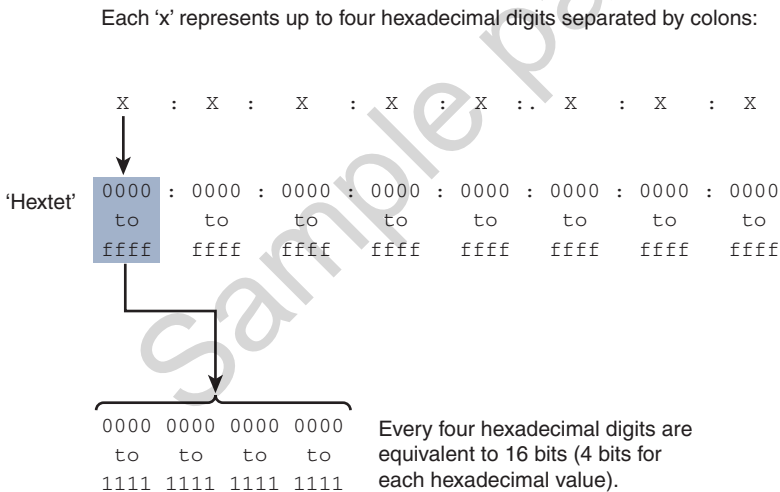


Figure 5-1 Format of an IPv6 Address

Showing all the hexadecimal digits in an IPv6 address is the longest representation of the preferred form. The next section shows you two rules for reducing the notation of an IPv6 address in the preferred format for easier use and readability.

TIP: If you need more practice working with hexadecimal and converting between hexadecimal, decimal, and binary, refer to both Appendix A, “How to Count in Decimal, Binary, and Hexadecimal,” and Appendix B, “How to Convert Between Number Systems.”

Reducing the Notation of an IPv6 Address

Looking at the longest representation of an IPv6 address can be overwhelming:

```
0000:0000:0000:0000:0000:0000:0000:0000
0000:0000:0000:0000:0000:0000:0000:0001
ff02:0000:0000:0000:0000:0000:0000:0001
fe80:0000:0000:0000:a299:9bff:fe18:50d1
2001:0db8:cafe:0001:0000:0000:0000:0200
```

There are two rules for reducing the notation.

Rule 1: Omit Leading 0s

Omit any leading 0s in any hextet (a 16-bit section). This rule applies only to leading 0s and not trailing 0s. Table 5-1 shows examples of omitting leading 0s in a hextet:

TABLE 5-1 Examples of Omitting Leading 0s in a Hextet (Leading 0s in bold; spaces retained)

Format	IPv6 Address
Preferred	0000:0000:0000:0000:0000:0000:0000:0000
Leading 0s omitted	0: 0: 0: 0: 0: 0: 0: 0 or 0:0:0:0:0:0:0:0
Preferred	0000:0000:0000:0000:0000:0000:0000:0001
Leading 0s omitted	0: 0: 0: 0: 0: 0: 0: 1 or 0:0:0:0:0:0:0:1
Preferred	ff02: 0000:0000:0000:0000:0000:0000:0001
Leading 0s omitted	ff02: 0: 0: 0: 0: 0: 0: 1 or ff02:0:0:0:0:0:0:1
Preferred	2001: 0db8:1111:000a:00b0:0000:9000:0200
Leading 0s omitted	2001: db8: 1111: a: b0: 0:9000: 200 or 2001:db8:1111:a:b0:0:9000:200

Rule 2: Omit All-0s Hextet

Use a double colon (::) to represent any single, contiguous string of two or more hextets consisting of all 0s. Table 5-2 shows examples of using the double colon.

TABLE 5-2 Examples of Omitting a Single Contiguous String of All-0s Hextets (0s in Bold Replaced By a Double Colon)

Format	IPv6 Address
Preferred	0000:0000:0000:0000:0000:0000:0000:0000
(::) All-0s segments	::
Preferred	0000:0000:0000:0000:0000:0000:0000:0001
(::) All-0s segments	::0001
Preferred	ff02: 0000:0000:0000:0000:0000:0000:0000:0001
(::) All-0s segments	ff02::0001
Preferred	2001:0db8:aaaa:0001: 0000:0000:0000:0100
(::) All-0s segments	2001:0db8:aaaa:0001::0100
Preferred	2001:0db8: 0000:0000 :abcd:0000:0000:1234
(::) All-0s segments	2001:0db8::abcd:0000:0000:1234

Only a single contiguous string of all 0s can be represented by a double colon; otherwise the address would be ambiguous. Consider the following example:

```
2001::abcd:1234
```

There are many different possible choices for the preferred address:

```
2001:0000:0000:0000:0000:abcd:0000:1234
```

```
2001:0000:0000:0000:abcd:0000:0000:1234
```

```
2001:0000:0000:abcd:0000:0000:0000:1234
```

```
2001:0000:abcd:0000:0000:0000:0000:1234
```

If two double colons are used, you cannot tell which of these addresses is correct.

If you have an address with more than one contiguous string of 0s, where should you place the double colon? RFC 5952 states that the double colon should represent

- The longest string of all-0s hextets.
- If the strings are of equal value, the first string should use the double colon notation.

Combining Rule 1 and Rule 2

You can combine the two rules to reduce an address even further. Table 5-3 shows examples of this.

TABLE 5-3 Examples of Applying Both Rule 1 and Rule 2 (Leading 0s in bold)

Format	IPv6 Address
Preferred	0000:0000:0000:0000:0000:0000:0000:0000
Leading 0s omitted	0: 0: 0: 0: 0: 0: 0: 0
(::) All-0s segments	::
Compressed	::
Preferred	0000:0000:0000:0000:0000:0000:0000:0001
Leading 0s omitted	0: 0: 0: 0: 0: 0: 0: 1
(::) All-0s segments	::1
Compressed	::1
Preferred	ff02: 0000:0000:0000:0000:0000:0000:0000:0001
Leading 0s omitted	ff02: 0: 0: 0: 0: 0: 0: 1
(::) All-0s segments	ff02::1
Compressed	ff02::1
Preferred	fe80: 0000:0000:0000:a299:9bff:fe18:50d1
Leading 0s omitted	fe80: 0: 0: 0:a299:9bff:fe18:50d1
(::) All-0s segments	fe80::a299:9bff:fe18:50d1
Compressed	fe80::a299:9bff:fe18:50d1
Preferred	2001:0db8:aaaa:0001: 0000:0000:0000:0200
Leading 0s omitted	2001: db8:aaaa: 1: 0: 0: 0: 200
(::) All-0s segments	2001: db8:aaaa: 1:: 200
Compressed	2001:db8:aaaa:1::200

Prefix Length Notation

In IPv4, the prefix of the address (the network portion) can be represented either by a dotted-decimal netmask (the subnet mask) or through CIDR notation. When we see 192.168.100.0 255.255.255.0 or 192.168.100.0/24, we know that the network portion of the address is the first 24 bits of the address (192.168.100) and that the last 8 bits (.0) are host bits. IPv6 address prefixes are represented in much the same way as IPv4 address prefixes are written in CIDR notation. IPv6 prefixes are represented using the following format:

IPv6-Address/Prefix-Length

The *prefix-length* is a decimal value showing the number of leftmost contiguous bits of the address. It identifies the prefix (the network portion) of the address. In unicast addresses, it is used to separate the prefix portion from the Interface ID. The Interface ID is equivalent to the host portion of an IPv4 address.

Looking at the address

2001:db8:aaaa:1111::100/64

we know that the leftmost 64 bits are the prefix (network portion) and the remaining bits are the Interface ID (host portion). See Figure 5-2.

Each hexadecimal digit is 4 bits; a hextet is a 16-bit segment.

2001:db8:aaaa:1111::100/64

2001 : 0db8 : aaaa : 1111 : 0000 : 0000 : 0000 : 0100

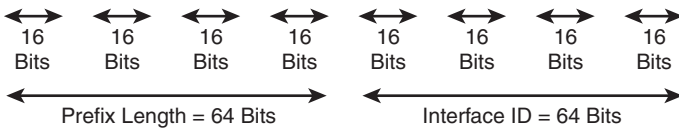


Figure 5-2 IPv6 Prefix and Prefix Length

A /64 prefix length results in an Interface ID of 64 bits. This is a common prefix length for most end-user networks. A /64 prefix length gives us 2^{64} or 18 quintillion devices on a single network (or subnet).

There are several more common prefix length examples, as shown in Figure 5-3. All of these examples fall either on a hextet boundary or on a nibble boundary (a multiple of 4 bits). Although prefix lengths do not need to fall on a nibble boundary, most usually do.

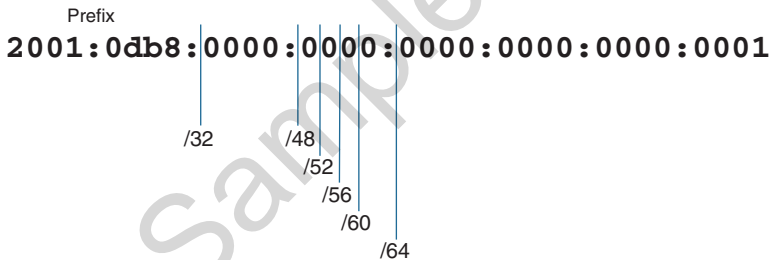


Figure 5-3 IPv6 Prefix Length Examples

IPv6 Address Types

In IPv6, there are three types of addresses: unicast, multicast, and anycast. This section gives a (very) high-level overview of these types.

NOTE: IPv6 does not have a broadcast address. There are other options that exist in IPv6 that deal with this issue, but this is beyond the scope of this book.

Figure 5-4 diagrams the three types of addresses.

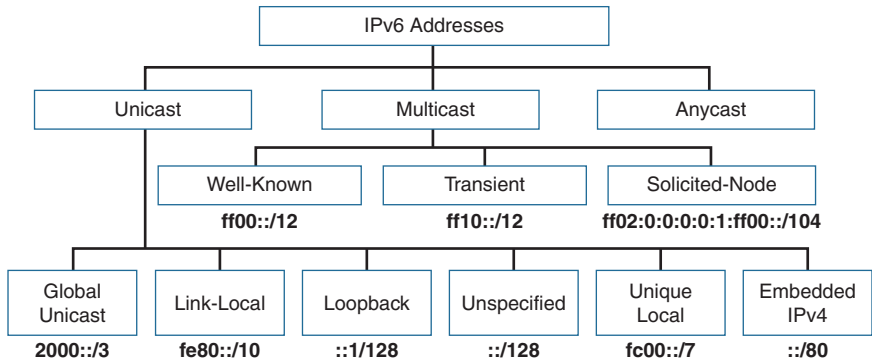


Figure 5-4 IPv6 Address Types

Unicast Addresses

A unicast address uniquely identifies an interface on an IPv6 device. A packet sent to a unicast address is received by the interface that is assigned to that address. Similar to IPv4, a source IPv6 address must be a unicast address.

As shown in Figure 5-4, there are six different types of unicast addresses:

1. **Global unicast:** A routable address in the IPv6 Internet, similar to a public IPv4 address.
2. **Link-local:** Used only to communicate with devices on the same local link.
3. **Loopback:** An address not assigned to any physical interface that can be used for a host to send an IPv6 packet to itself.
4. **Unspecified address:** Used only as a source address and indicates the absence of an IPv6 address.
5. **Unique local:** Similar to a private address in IPv4 (RFC 1918) and not intended to be routable in the IPv6 Internet. However, unlike RFC 1918 addresses, these addresses are not intended to be statefully translated to a global unicast address. Please see Rick Graziani's book *IPv6 Fundamentals* for a more detailed description of stateful translation.
6. **IPv4 embedded:** An IPv6 address that carries an IPv4 address in the low-order 32 bits of an IPv6 address.

Global Unicast Addresses

Global unicast addresses (GUAs) are globally routable and reachable in the IPv6 Internet. The generic structure of a GUA has three fields:

- **Global Routing Prefix:** The prefix or network portion of the address assigned by the provider, such as an ISP, to the customer site.

- Subnet ID:** A separate field for allocating subnets within the customer site. Unlike IPv4, it is not necessary to borrow bits from the Interface ID (host portion) to create subnets. The number of bits in the Subnet ID falls between where the Global Routing Prefix ends and the Interface ID begins.
- Interface ID:** Identifies the interface on the subnet, equivalent to the host portion of an IPv4 address. In most cases, the Interface ID is 64 bits in length.

Figure 5-5 shows the structure of a global unicast address.

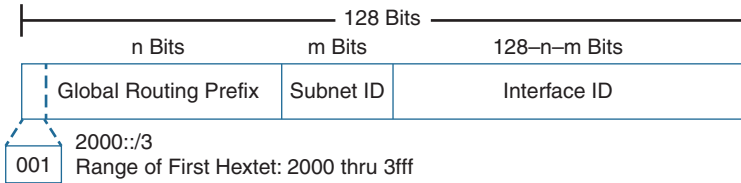


Figure 5-5 Structure of a Global Unicast Address

Link-Local Unicast Addresses

A link-local unicast address is a unicast address that is confined to a single link (a single subnet). Link-local addresses only need to be unique on the link (subnet) and do not need to be unique beyond the link. Therefore, routers do not forward packets with a link-local address.

Figure 5-6 shows the format of a link-local unicast address, which is in the range fe80::/10. Using this prefix and prefix length range results in the range of the first hexet being from fe80 to febf.

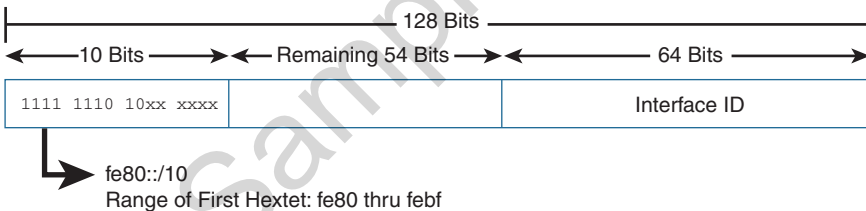


Figure 5-6 Structure of a Link-Local Unicast Address

NOTE: Using a prefix other than fe80 is permitted by RFC 4291, but the addresses should be tested prior to usage.

NOTE: To be an IPv6-enabled device, a device must have an IPv6 link-local address. You do not need to have an IPv6 global unicast address, but you must have a link-local address.

NOTE: Devices dynamically (automatically) create their own link-local IPv6 addresses upon startup. Link-local addresses can be manually configured.

NOTE: Link-local addresses only need to be unique on the link. It is very likely, and even desirable, to have the same link-local address on different interfaces that are on different links. For example, on a device named Router2, you may want all link-local interfaces to be manually configured to FE80::2, whereas all link-local interfaces on Router3 to be manually configured to FE80::3, and so on.

NOTE: There can be only one link-local address per interface. There can be multiple global unicast addresses per interface.

Loopback Addresses

An IPv6 loopback address is ::1, an all-0s address except for the last bit, which is set to 1. It is equivalent to the IPv4 address block 127.0.0.0/8, most commonly the 127.0.0.1 loopback address. The loopback address can be used by a node to send an IPv6 packet to itself, typically when testing the TCP/IP stack.

Table 5-4 shows the different formats for representing an IPv6 loopback address.

TABLE 5-4 IPv6 Loopback Address Representation

Representation	IPv6 Loopback Address
Preferred	0000:0000:0000:0000:0000:0000:0000:0001
Leading 0s omitted	0:0:0:0:0:0:0:1
Compressed	::1

NOTE: A loopback address cannot be assigned to a physical interface.

Unspecified Addresses

An unspecified unicast address is an all-0s address (see Table 5-5), used as a source address to indicate the absence of an address.

Table 5-5 shows the different formats for representing an IPv6 unspecified address.

TABLE 5-5 IPv6 Unspecified Address Representation

Representation	IPv6 Unspecified Address
Preferred	0000:0000:0000:0000:0000:0000:0000:0000
Leading 0s omitted	0:0:0:0:0:0:0:0
Compressed	::

NOTE: An unspecified address cannot be assigned to a physical interface.

Unique Local Addresses

Figure 5-7 shows the structure of the unique local address (ULA), which is the counterpart of IPv4 private addresses. ULAs are used similarly to global unicast addresses, but are for private use and cannot be routed in the global Internet. ULAs are defined in RFC 4193.

Figure 5-7 shows the different formats for representing an IPv6 unspecified address.

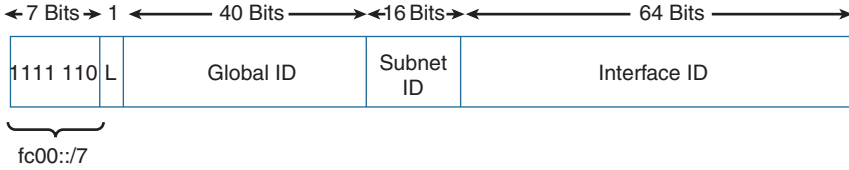


Figure 5-7 Structure of a Unique Local Unicast Address

IPv4 Embedded Addresses

Figure 5-8 shows the structure of IPv4 embedded addresses. They are used to aid in the transition from IPv4 to IPv6. IPv4 embedded addresses carry an IPv4 address in the low-order 32 bits of an IPv6 address.

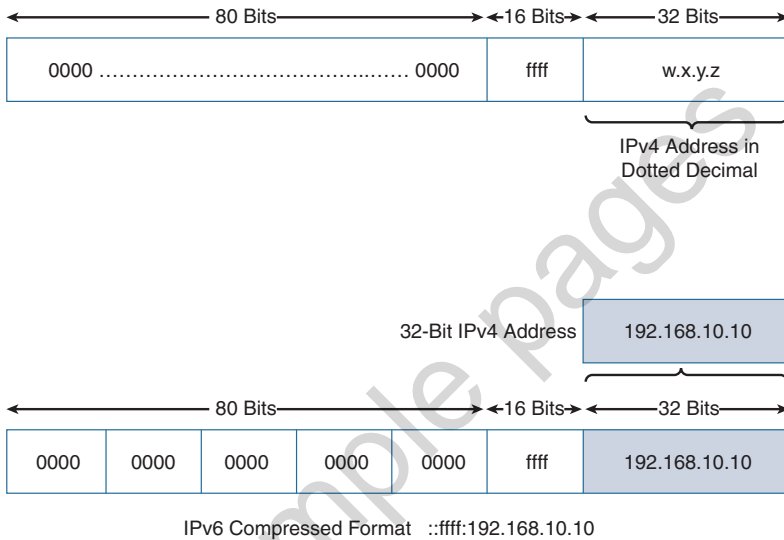


Figure 5-8 IPv4-Mapped IPv6 Address

NOTE: This is a transition technique for moving from IPv4 to IPv6 addressing. This should not be used as a permanent solution. The end goal should always be native end-to-end IPv6 connectivity.

Multicast Addresses

Multicast is a technique in which a device sends a single packet to multiple destinations simultaneously (one-to-many transmission). Multiple destinations can actually be multiple interfaces on the same device, but they are typically different devices.

An IPv6 multicast address defines a group of devices known as a multicast group. IPv6 addresses use the prefix `ff00::/8`, which is equivalent to the IPv4 multicast address `224.0.0.0/4`. A packet sent to a multicast group always has a unicast source address; a multicast address can never be the source address.

Unlike IPv4, there is no broadcast address in IPv6. Instead, IPv6 uses multicast.

Table 5-6 shows IPv6 multicast address representation.

TABLE 5-6 IPv6 Multicast Address Representation

Representation	IPv6 Multicast Address
Preferred	ff00:0000:0000:0000:0000:0000:0000/8
Leading 0s omitted	ff00:0:0:0:0:0:0/8
Compressed	ff00::/8

The structure of an IPv6 multicast is shown in Figure 5-9; the first 8 bits are 1-bits (ff) followed by 4 bits for flags and a 4-bit Scope field. The next 112 bits represent the Group ID.

8 Bits	4 Bits	4 Bits	112 Bits
1111 1111	Flags	Scope	Group ID

Figure 5-9 IPv6 Multicast Address

Although there are many different types of multicast addresses, this book defines only two of them:

- Well-known multicast addresses
- Solicited-node multicast addresses

Well-Known Multicast Addresses

Well-known multicast addresses have the prefix ff00::/12. Well-known multicast addresses are predefined or reserved multicast addresses for assigned groups of devices. These addresses are equivalent to IPv4 well-known multicast addresses in the range 224.0.0.0 to 239.255.255.255. Some examples of IPv6 well-known multicast addresses include the following:

Address	Use
ff02::1	All IPv6 devices
ff02::2	All IPv6 routers
ff02::5	All OSPFv3 routers
ff02::6	All OSPFv3 DR routers
ff02::9	All RIPng routers
ff02:a	All EIGRPv6 routers
ff02::1:2	All DHCPv6 servers and relay agents

Solicited-Node Multicast Addresses

Solicited-node multicast addresses are used as a more efficient approach to IPv4's broadcast address. A more detailed description is beyond the scope of this book.

Anycast Addresses

An IPv6 anycast address is an address that can be assigned to more than one interface (typically on different devices). In other words, multiple devices can have the same anycast address. A packet sent to an anycast address is routed to the "nearest" interface having that address, according to the router's routing table.

Figure 5-10 shows an example of anycast addressing.

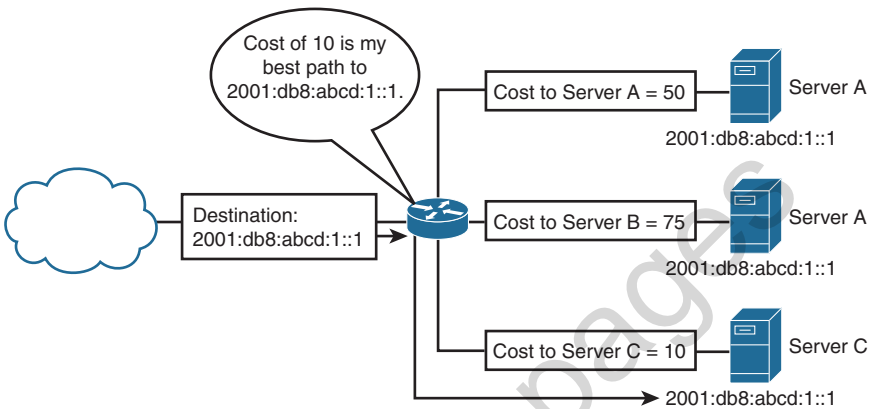


Figure 5-10 Example of Anycast Addressing

NOTE: IPv6 anycast addressing is still somewhat in the experimental stages and beyond the scope of this book.