IPv6 Fundamentals

A Straightforward Approach to Understanding IPv6

ciscopress.com

Rick Graziani

IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6

Rick Graziani

Cisco Press

800 East 96th Street

Indianapolis, IN 46240

IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6

Rick Graziani

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

First Printing October 2012

The Library of Congress Cataloging-in-Publication Data is on file.

ISBN-13: 978-1-58714-313-7

ISBN-10: 1-58714-313-5

Warning and Disclaimer

This book is designed to provide information about IPv6 (Internet Protocol version 6). 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.

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 email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

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.

Publisher: Paul Boger

Associate Publisher: Dave Dusthimer

Executive Editor: Mary Beth Ray

Managing Editor: Sandra Schroeder

Development Editor: Marianne Bartow

Editorial Assistant: Vanessa Evans

Cover Designer: Sandra Schroeder

Indexer: Tim Wright

Business Operation Manager, Cisco Press: Anand Sundaram

Cisco Press Program Manager: Sonia Torres Chavez

Technical Editors: Jim Bailey, Yenu Gobena

Copy Editor: John Edwards

Project Editor: Mandie Frank

Book Designer: Louisa Adair

Composition: Mark Shirar

Proofreader: Debbie Williams

About the Author

Rick Graziani teaches computer science and computer networking courses at Cabrillo College in Aptos, California. Prior to teaching, he worked in the information technology field for Santa Cruz Operation, Tandem Computers, and Lockheed Missiles and Space Corporation. He holds an M.A. in computer science and systems theory from California State University Monterey Bay. Rick also does consulting work for Cisco Systems. When he is not working, he is most likely surfing at one of his favorite Santa Cruz surf breaks.

Rick Graziani, the early years.

About the Technical Reviewers

Jim Bailey, CCIE #5275 (Routing & Switching; Service Provider) and CCDE #20090008, is a Services Technical Leader at Cisco Systems with more than 20 years of experience in networking. As part of the Central Engineering Network Architecture and Design Advanced Services team, he focuses on the architecture, design, and implementation of networks for Enterprise, Service Provider, and Government customers. He has focused on IPv6 integration into those networks over the last seven years.

Yenu Gobena is a Distinguished Service Engineer in a strategic role in the Borderless Network Architecture and Design Space. He is also a key player in evangelizing, consulting, and enabling multiple Cisco architectures such as BYOD, VDI/VXI, IPv6, converged Core infrastructures, and Data Centers. Yenu also focuses on consulting and providing design services to some of the leading IPv6 adopters globally. **Yenu** has been leading IPv6 customer delivery engagements and workshops for several years, and he has earned a trusted advisory role for dozens of accounts around the globe. He is a sought-after industry speaker and represents Cisco at key IPv6 initiatives around the world. Based in Research Triangle Park (Raleigh, N.C.), he has experience in a broad range of technologies, including Routing and Switching, MPLS, SP Mobility, IPv6 Architecture and Design, Enterprise, and SP NGN Architectures. Yenu holds a B.S. in telecommunications from the State University of New York. He has two children, Amara and Elena, with his wife, Molly. Yenu is CCIE certified, has published multiple white papers, and is an IPv6 Forum Certified Gold Trainer.

Dedication

This book is dedicated to my parents. Thank you for the many years of love and support.

Acknowledgments

First of all, I would like to thank all my friends and colleagues for their assistance. A special thank-you to all my friends in the Thursday evening DPS (Dropped Packet Society). Thank you for your help during and after our "meetings." There are too many of you to list, but Mark Boolootian, Dave Barnett, and Jim Warner, thanks for the many years of discussing technologies and answering questions. We've drawn a lot of topologies on a lot of napkins over the years.

Jim Bailey at Cisco Systems deserves much more credit than the brief mention as a technical reviewer for this book. He made me look a lot smarter than I am. Jim did an incredibly thorough job in making sure that this book was as accurate and up to date as possible. His expertise and background were invaluable in helping me author this book. He is definitely the unsung hero of this project. Thank you Jim for an amazing job.

Thank you Yenu Gobena at Cisco Systems for doing such a great job on the technical review. You made this book more accurate by filling in a lot of the blanks. You helped improve this book for the reader.

I owe a great deal of gratitude to Gerlinde Brady, Sue Nerton, and Jim Griffin for your friendship and support. You made sure that the CS/CIS departments at Cabrillo College continued to run smoothly while I was engaged in the writing process. I feel very fortunate to work with you and all our other friends in the CS/CIS department.

Writing this book has been one of many privileges, all of which have been because of Dennis Frezzo, Jeremy Creech, Karen Alderson, Sonya Stott, Wayne Lewis, Bob Vachon, and many others who work for the Cisco Networking Academy. Thank you all for the honor and opportunity to be part of a program that has changed the lives of thousands of students around the world. More than colleagues, you are all friends, for which I am grateful.

Thank you Pat Farley for making sure that I still got my surf sessions in. For those who surf, you know how important this is.

Special thanks to Mary Beth Ray, Executive Editor for Cisco Press and friend. Thank you for your patience and understanding through this long process. You always have that calm assurance and guidance, not to mention being a terrific dancer.

Thank you, Marianne Bartow at Cisco Press, for working with me on a daily basis—weekdays and weekends, editing, formatting, and orchestrating the entire process. You were a pleasure to work with, and I am very grateful for all the hard work you put into this book. You make me look as if I am actually a better writer than I am.

To Chris Cleveland, Mandie Frank, John Edwards, Tim Wright, Mark Shirar, Sandra Schroeder, and everyone else at Cisco Press, I am extremely grateful for everything you have done. I am constantly amazed at the level of cooperation and teamwork required to produce a technical book, and I am very thankful for all your help.

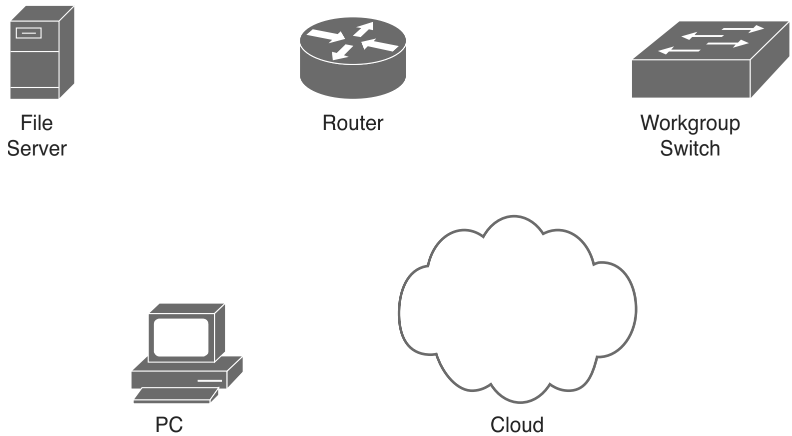
Thank you Luigi. Waking me up early every morning to go to the beach has been beneficial for both of us. Good dog!

Finally, I want to thank all my students over the many years. I am humbled by the opportunity to teach such wonderful people. You make my job fun and are the reason I love to go to work every day.

Contents at a Glance

Contents

Icons Used in This Book



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Boldface** indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a **show** command).
- *Italics* indicate arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets [ ] indicate optional elements.
- Braces { } indicate a required choice.
- Braces within brackets [{ }] indicate a required choice within an optional element.

Introduction

This book focuses on the basics of IPv6. There is a great deal to learn about IPv6. It is much more than just becoming familiar with a larger address.

My approach to writing this book was to do my best to explain each concept in a simple, step-by-step approach, as well as to include the critical details. It was a challenging balance between providing as much information as possible without overwhelming the reader. IPv6 is not difficult to learn but involves multiple protocols and processes that might be new to the reader.

Don't be overwhelmed by all the details. For example, although I have included a brief description for each field in the protocols discussed in the book, it isn't necessarily important that you understand the details of each one. I mention this throughout the book. But I did feel it necessary not to leave out or hide these details from the reader.

RFCs are cited throughout the book. It was important to include these references for two reasons. First of all, I wanted to give you the authoritative source for the material in this book so that you have a resource for more information. Second, IPv6 is currently and will continue to be a moving target for quite some time. Although it has been around for many years, there is still additional development and fine-tuning that is taking place. If you are not familiar with reading RFCs, don't be intimidated. Most of them are not difficult to read and do their best to explain the topic clearly.

At times I will introduce a technology or concept, but state that it is "beyond the scope of this book." I feel it is better to revisit some of the more advanced topics after you have a more complete understanding of the entire IPv6 topic. I do suggest resources for those who might be interested in learning more about those topics.

The objective of this book is to explain IPv6 as clearly as possible. At times it was like herding cats, trying to decide which topic to cover first.

Readers are welcome to use the resources on my website for IPv6, CCNA, or CCNP information, www.cabrillo.edu/~rgraziani. You can email me, graziani@cabrillo.edu, to obtain the username and password for all my materials.

Goals and Methods

The most important goal of this book is to provide a thorough yet easy-to-understand introduction to IPv6. This book is also intended to provide a foundation in IPv6 that will allow you to build on it. This includes explaining topics that might be a little more challenging to grasp.

Another goal of this book is to be a resource for IPv6. I have included command syntax, RFCs, and links to Cisco white papers to help guide you toward a further understanding of many of the topics.

Who Should Read This Book

This book is intended for anyone seeking a solid understanding of the fundamentals of IPv6, such as network engineers, network designers, network technicians, technical staff, and networking students, including those who are part of the Cisco Networking Academy. The reader should have a basic familiarity with IPv4 and IPv4 routing protocols equivalent to a CCNA certification or the applicable Cisco Network Academy courses.

Professionals planning to use Cisco technology to deploy IPv6 networks, provide IPv6 connectivity, and use IPv6 within their network will find this book useful. You will find examples, figures, IOS commands, and tips for configuring Cisco IOS IPv6 technology.

How This Book Is Organized

If you are new to IPv6, this book should be read from cover to cover. However, if you have some knowledge of IPv6, it is designed to be flexible and allows you to easily move between chapters and sections of chapters to cover just the material that you want to review.

Chapters 1 through 5 provide an introduction to IPv6, the protocol, addressing types, and ICMPv6 Neighbor Discovery Protocol. These chapters also include Cisco IOS commands and configuration examples. Chapters 6 through 9 use a common topology to implement the IPv6 addressing in the previous chapters and also introduce IPv6 routing protocols. DHCPv6 and other upper-layer protocols are also discussed. The last two chapters, Chapters 10 and 11, cover methods of transitioning from IPv4 to IPv6. If you do intend to read all the chapters, the order in the book is sequential.

The following list highlights the topics covered in each chapter and the book's organization:

- **Chapter 1, "Introduction to IPv6":** This chapter discusses how the Internet of today requires a new network layer protocol, IPv6, to meet the demands of its users. It also examines the limitations of IPv4 and describes how IPv6 resolves these issues while offering other advantages as well. This chapter examines the rationale of IPv6 and concerns regarding IPv4 address depletion. It presents a brief history of both IPv4 and IPv6. A review of the IPv4 migration technologies CIDR and NAT are also discussed.
- **Chapter 2, "The IPv6 Protocol":** This chapter examines the IPv6 protocol and its fields. The IPv4 protocol is first reviewed to provide a basis of comparison and to highlight the changes with IPv6. It also explores how fragmentation is handled. The IPv6 extension headers are discussed as well.
- **Chapter 3, "IPv6 Addressing":** This chapter introduces IPv6 addressing and address types. It begins with an explanation of the hexadecimal number system. Representation of IPv6 addresses is discussed along with the different formats of representing IPv6 addresses and the rules for compressing the IPv6 notation. This chapter provides an introductory look at the different types of IPv6 addresses. Subnetting IPv6 addresses is discussed, including subnetting on a nibble boundary and within the nibble boundary.

- **Chapter 4, "IPv6 Address Types":** This chapter examines the different types of IPv6 addresses in detail. Global unicast configuration methods, both manual and dynamic, are described. It explains and provides examples of enabling IPv6 on router interfaces using static, EUI-64, IPv6 unnumbered, Stateless Address Autoconfiguration (SLAAC), and DHCPv6. (DHCPv6 and SLAAC are examined in detail in later chapters.) Link-local addresses are described using static and dynamic IOS configuration examples. Loopback and unspecified unicast addresses are also discussed. Assigned and solicited node multicast addresses, along with anycast addresses, are described as well.

- **Chapter 5, "ICMPv6 and Neighbor Discovery Protocol":** This chapter examines ICMPv6. There are similarites between ICMPv6 and ICMPv4, but ICMPv6 is a much more robust protocol. ICMPv6 error messages are discussed, including Destination Unreachable, Packet Too Big, Time Exceeded, and Parameter Problem. ICMPv6 informational messages Echo Request and Echo Reply are covered along with Multicast Listener Discovery messages. Neighbor Discovery Protocol, Router Solicitation, Router Advertisement, Neighbor Solicitation, Neighbor Advertisement, and Redirect messages are examined in detail. Not only does IPv6 resolve larger address space issues, but ICMPv6 and Neighbor Discovery Protocol also present a major change in the network operatons, including link-layer address resolution (ARP in IPv4), Duplicate Address Detection (DAD), Stateless Address Autoconfiguration (SLAAC), and Neighbor Unreachability Detection (NUD). The IPv6 Neighbor Cache and Neighbor Cache States, similar to those of the IPv4 ARP Cache, are discussed.

- **Chapter 6, "IPv6 Configuration":** This chapter illustrates the configuration of IPv6 addressing using a common topology. Global unicast and link-local addresses are configured using different options. This chapter includes examples of the Neighbor Cache and modification of Router Adveristement messages for tuning the Neighbor Discovery parameters. This chapter also explores IPv6 access control lists, with configuration examples using the common topology.

- **Chapter 7, "Introduction to Routing IPv6":** This chapter examines the IPv6 routing table and changes in the configurations pertaining to IPv6. It also discusses the configuration of IPv6 static routes that are similar to static routes for IPv4. CEF for IPv6 is also covered.

- **Chapter 8, "IPv6 IGP Routing Protocols":** This chapter discusses three routing protocols: RIPng, EIGRP for IPv6, and OSPFv3. Differences between each protocol and their IPv4 counterpart are examined. A common topology is used to configure and verify each of the routing protocols.

- **Chapter 9, "DHCPv6 (Dynamic Host Configuration Protocol version 6)":** This chapter examines DHCP for IPv6, or DHCPv6. Stateful and stateless DHCPv6 services are discussed. DHCPv6 terminology and message types are covered along with the DHCPv6 process between the client and server. The Rapid Commit Option and relay agents are also explained. Other upper-layer protocols are discussed: DNS, TCP, and UDP.

- **Chapter 10, "Dual-Stack and Tunneling":** This chapter covers two of three strategies for IPv4 and IPv6 integration and coexistence: dual-stack and tunneling. Dual-stack is used when devices will implement both IPv4 and IPv6 protocols, enabling them to coexist in the same network. Tunneling is the encapsulation of one IP packet inside another IP packet. Basic tunneling terminology is discussed as well as an examination and configuration of three tunneling technologies: manual, 6to4, and ISATAP.
- **Chapter 11, "Network Address Translation IPv6 to IPv4 (NAT64)":** This chapter discusses the third technique for transition from IPv4 and IPv6: Network Address Translation, or NAT. Similar to NAT for IPv4, the use of NAT for IPv6 is a translation between the IPv4 and IPv6 protocols. NAT64 is a replacement for NAT-PT (NAT – Protocol Translation). NAT-PT is included in this chapter to provide continuity between the newer NAT64 and the previous technology that is still in use.

This page intentionally left blank

Chapter 3

IPv6 Addressing

The most recognizable difference between IPv4 and IPv6 is the address space. An IPv4 address is 32 bits and expressed in dotted-decimal notation, whereas an IPv6 address is 128 bits in length and written in hexadecimal. IPv6 addressing is defined in RFC 4291, IP Version 6 Addressing Architecture.

In this chapter, you will become familiar with reading IPv6 addresses and recognizing the different parts. You will take a brief look at the different types of IPv6 addresses and the basic structure of a global unicast address. You will configure a router's interface with an IPv6 address and verify reachability with the **ping** command. This chapter also examines how to subnet an IPv6 address, which in most cases, is much easier than subnetting in IPv4.

At first, the longer, hexadecimal IPv6 address can look intimidating. This does not have to be the case—as a matter of fact, IPv6 addresses can be easier to read and much simpler to subnet than their IPv4 counterparts. By the end of this book, you might actually prefer working with IPv6 addresses than with IPv4 addresses! I begin by making sure that you understand the hexadecimal number system and IPv6 address notation.

Hexadecimal Number System

This section is intended for people who are unfamiliar with the hexadecimal number system. If you are comfortable with hexadecimal numbers, you might want to skip this section. An IPv6 address is 128 bits in length, and you will see that hexadecimal is the ideal number system for representing long strings of bits.

If you understand the decimal, or base 10, number system, you can understand any number system, including the hexadecimal, or base 16, number system. Let's assume that you are already familiar with binary or base 2, but if you're not, you will still be able to understand base 16. The same general rules apply to all number systems.

When looking at integer-based number systems, there are three general rules:

Rule #1: Base n number systems have *n* number of digits:

- Base 10 (decimal) number system has 10 digits.
- Base 2 (binary) number system has 2 digits.
- Base 16 (hexadecimal) number system has 16 digits.

Rule #2: All number systems begin with 0.

Combining Rule #1 and Rule #2, we get:

- Base 10 has ten digits starting with 0: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Base 2 has two digits starting with 0: 0, 1.
- Base 16 has 16 digits starting with 0: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (I discuss the alphanumeric digits in a moment).

Rule #3: The first column, the rightmost column or least significant digit, is always the column of 1s (ones). Each preceding column is *n* times the previous column. (*n* is the base n number system.) Using base 10 as an example, the first column is the column of 1s; the next column is 10 times the 1s column, resulting in a column of 10s; the next column is 10 times the 10s column, resulting in a column of 100s; and so on. This makes it very easy when converting other number systems to base 10. This is illustrated in Table 3-1.

Table 3-1 *Number Systems*

Base n Number System	n^3	n^2	n^1	n^0
Base 10	1000	100	10	1
Base 2	8	4	2	1
Base 16	4096	256	16	1

- Base 10: 10,000s, 1000s, 100s, 10s, 1s
- Base 2: 128s, 64s, 32s, 16s, 8s, 4s, 2s, 1s
- Base 16: 256s, 16s, 1s

If you understand the three rules, it is time to examine the hexadecimal number system more closely. The hexadecimal number system, base 16, has 16 digits beginning with 0. Table 3-2 shows these 16 digits and their equivalence in decimal and binary.

Table 3-2 *Decimal, Hexadecimal, and Binary*

Decimal (Base 10)	Hexadecimal (Base 16)	Binary (Base 2)
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Applying the three rules to the hexadecimal number system:

- **Rule #1:** The hexadecimal number system has 16 digits.
- **Rule #2:** Table 3-2 illustrates the 16 hexadecimal digits and their decimal and binary equivalents starting with 0. Notice that you needed unique alphanumeric digits, A through F, to represent the decimal values 10 through 15.
- **Rule #3:** For representing IPv6 addresses in hexadecimal, you only need to use the first column or column of 1s.

So, why use hexadecimal numbers to represent IPv6 addresses? Hexadecimal is a very common number system used in computer science, computer networking, and other areas of computer technology. This is because any 4 bits (half of a byte or half of an octet) can be represented as a single hexadecimal digit. In other words, there are 16 unique combinations of 4 bits, and there are also 16 digits in a hexadecimal number system, so it is a perfect match. Because one hexadecimal digit can represent 4 bits, this means that two hexadecimal digits can represent a single byte or octet.

Note 4 bits is half a byte or half an octet, and is also known as a *nibble*. You will sometimes see the alternative spellings of *nybble* or *nyble*.

Representation of IPv6 Addresses

IPv6 addresses are 128 bits in length and written as a string of hexadecimal digits. Every 4 bits are represented by a single hexadecimal digit, for a total of 32 hexadecimal values (4 * 32 = 128). The alphanumeric characters used in hexadecimal are not case sensitive; therefore, uppercase and lowercase characters are equivalent.

Note RFC 5952, A Recommendation for IPv6 Address Text Representation, recommends that IPv6 addresses be represented in lowercase. Throughout this book, you will see many cases where I have used uppercase characters. I have done this to make it easier for you to visualize and differentiate the different types of addresses.

As described in RFC 4291, the preferred format is x:x:x:x:x:x:x:x. Each *x* is a 16-bit section that can be represented using up to four hexadecimal digits separated by a colon. This results in eight 16-bit sections (for a total of 128 bits) of the address, as shown in Figure 3-1.

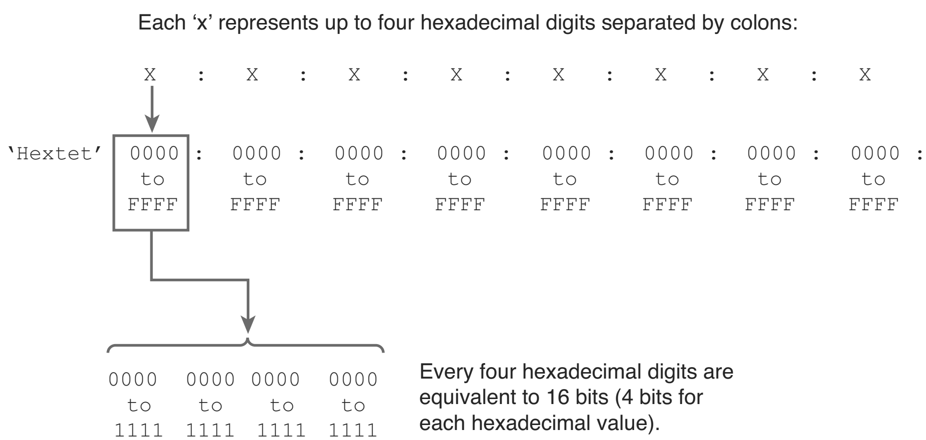


Figure 3-1 *Preferred Format of IPv6 Address*

The preferred format is the longest representation of an IPv6 address. A total of 32 hexadecimal values are used. A colon separates every group of four hexadecimal digits. Once again, each hexadecimal digit is equivalent to 4 bits.

Note The unofficial term for a section of four hexadecimal values is a *hextet*, similar to the term *octet* used in IPv4 addressing. Therefore, an IPv6 address consists of eight hextets separated by colons. As Figure 3-1 illustrates, each hextet with its four hexadecimal digits is equivalent to 16 bits. For clarity, the term *hextet* will be used throughout this book when referring to individual 16-bit segments. Table 3-3 shows several examples of IPv6 addresses using the preferred format. Notice that the last two addresses are the IPv6 addresses of PC1 and PC2 in the topology from Chapter 2, "The IPv6 Protocol."

Table 3-3 *Examples of IPv6 Addresses Using the Preferred Format*

Preferred Format of IPv6 Addresses
0000:0000:0000:0000:0000:0000:0000:0000
0000:0000:0000:0000:0000:0000:0000:0001
FF02:0000:0000:0000:0000:0000:0000:0001
FC00:0001:A000:0B00:0000:0527:0127:00AB
2001:DCBA:1111:000A:00B0:0000:9000:0200
2001:0000:0000:0000:ABCD:0000:0000:1234
2001:0DB8:AAAA:0001:0000:0000:0000:0100
2001:0DB8:AAAA:0001:0000:0000:0000:0200

At first glance, these addresses can look overwhelming. Don't worry; later in this chapter, you will be introduced to a technique to bolster your confidence in reading and using IPv6 addresses. Besides the preferred format, RFC 2373 and RFC 5952 provide two helpful rules in reducing the notation of these addresses.

Rule 1: Omission of Leading 0s

Leading 0s (zeroes) in any hextet, 16-bit section, can be omitted. This applies only to leading 0s and not to trailing 0s; otherwise, the address would be ambiguous. Using a list of preferred IPv6 addresses, Table 3-4 shows how the leading 0s can be removed.

Table 3-4 *Examples of Omitting Leading 0s in a Hextet*

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

Format	IPv6 Address
Preferred	**000**0:**000**0:**000**0:**000**0:**000**0:**000**0:**000**0:**000**1
Leading 0s omitted	0: 0: 0: 0: 0: 0: 0: 1 or 0:0:0:0:0:0:0:1
Preferred	FF02:**000**0:**000**0:**000**0:**000**0:**000**0:**000**0:**000**1
Leading 0s omitted	FF02: 0: 0: 0: 0: 0: 0: 1 or FF02:0:0:0:0:0:0:1
Preferred	FC00:**000**1:A000:**0**B00:**000**0:**0**527:**0**127:**00**AB
Leading 0s omitted	FC00: 1:A000: B00: 0: 527: 127: AB or FC00:1:A000:B00:0:527:127:AB
Preferred	2001:**0**DB8:1111:**000**A:**00**B0:**000**0:9000:**0**200
Leading 0s omitted	2001: DB8:1111: A: B0: 0:9000: 200 or 2001:DB8:1111:A:B0:0:9000:200
Preferred	2001:**0**DB8:**000**0:**000**0:ABCD:**000**0:**000**0:1234
Leading 0s omitted	2001: DB8: 0: 0:ABCD: 0: 0:1234 or 2001:DB8:0:0:ABCD:0:0:1234
Preferred	2001:**0**DB8:AAAA:**000**1:**000**0:**000**0:**000**0:**0**100
Leading 0s omitted	2001: DB8:AAAA: 1: 0: 0: 0: 100 or 2001:DB8:AAAA:1:0:0:0:100
Preferred	2001:0DB8:AAAA:**000**1:**000**0:**000**0:**000**0:**0**200
Leading 0s omitted	2001: DB8:AAAA: 1: 0: 0: 0: 200 or 2001:DB8:AAAA:1:0:0:0:200

Note In Table 3-4, the 0s to be omitted are in bold. Spaces remain to better visualize where the 0s were removed.

It is important to remember that only leading 0s can be removed; otherwise, it will make the address ambiguous. For example, if trailing 0s were also permitted, you wouldn't know what the correct address was. There can only be one correct interpretation; therefore, only leading 0s can be omitted:

■ Zeroes omitted:	`2001:1944:100:A:0:BC:ABCD:D0B`
■ Incorrect (trailing 0s):	`2001:1944:1000:A000:0000:BC00:ABCD:D0B0`
■ Correct (leading 0s):	`2001:1944:0100:000A:0000:00BC:ABCD:0D0B`

Rule 2: Omission of all-0s hextets

A double colon (::) can represent any single, contiguous string of one or more hextets (16-bit segments) consisting of all 0s. This will help further reduce the size of an IPv6 address. Table 3-5 illustrates the use of the double colon.

Table 3-5 *Example of Omitting a Single Contiguous String of All-0 Hextets*

Format	IPv6 Address
Preferred	**0000:0000:0000:0000:0000:0000:0000:0000**
(::) All-0 segments	::
Preferred	**0000:0000:0000:0000:0000:0000:0000**:0001
(::) All-0 segments	::0001
Preferred	FF02:**0000:0000:0000:0000:0000:0000**:0001
(::) All-0 segments	FF02::0001
Preferred	FC00:0001:A000:0B00:**0000**:0527:0127:00AB
(::) All-0 segments	FC00:0001:A000:0B00::0527:0127:00AB
Preferred	2001:DCBA:1111:000A:00B0:**0000**:9000:0200
(::) All-0 segments	2001:DCBA:1111:000A:00B0::9000:0200
Preferred	2001:**0000:0000:0000**:ABCD:0000:0000:1234
(::) All-0 segments	2001::ABCD:0000:0000:1234

Note: This address can also be written as `2001:0000:0000:0000:ABCD::1234`.

Format	IPv6 Address
Preferred	2001:0DB8:AAAA:0001:**0000:0000:0000**:0100
(::) All-0 segments	2001:0DB8:AAAA:0001::0100
Preferred	2001:0DB8:AAAA:0001:**0000:0000:0000**:0200
(::) All-0 segments	2001:0DB8:AAAA:0001::0200

Note In Table 3-5, the 0s in bold in the preferred address are replaced by the double colon.

Only a single contiguous string of all-0 segments can be represented with a double colon; otherwise, the address would be ambiguous:

- Incorrect address using two double colons:

  ```
  2001::ABCD::1234
  ```

- Possible ambiguous choices:

  ```
  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
  ```

As you can see, if two double colons are used, there would be multiple possibilities, and you wouldn't know which address is the correct interpretation.

Note RFC 5952 suggests that the double colon should represent the longest string of 0s.

Combining Rule 1 and Rule 2

Combining both rules can reduce the address even further. Table 3-6 illustrates all three formats. Again, spaces were left to better visualize where the 0s were removed.

Table 3-6 *Combining Rule 1 and Rule 2*

Format	IPv6 Address
Preferred	0000:0000:0000:0000:0000:0000:0000:0000
No Leading 0s	0: 0: 0: 0: 0: 0: 0: 0
":" All-0 segments	::
Preferred	0000:0000:0000:0000:0000:0000:0000:0001
No Leading 0s	0: 0: 0: 0: 0: 0: 0: 1
":" All-0 segments	::1
Preferred	FF02:0000:0000:0000:0000:0000:0000:0001
No Leading 0s	FF02: 0: 0: 0: 0: 0: 0: 1
":" All-0 segments	FF02::1
Preferred	FC00:0001:A000:0B00:0000:0527:0127:00AB
No Leading 0s	FC00: 1:A000: B00: 0: 527: 127: AB
":" All-0 segments	FC00:1:A000:B00::527:127:AB
Preferred	2001:DCBA:1111:000A:00B0:0000:9000:0200
No Leading 0s	2001:DCBA:1111: A: B0: 0:9000: 200
":" All-0 segments	2001:DCBA:1111:A:B0::9000:200
Preferred	2001:0000:0000:0000:ABCD:0000:0000:1234
No Leading 0s	2001: 0: 0: 0:ABCD: 0: 0:1234
":" All-0 segments	2001::ABCD:0:0:1234
Note: This address can also be written as 2001:0:0:0:ABCD::1234.	
Preferred	2001:0DB8:AAAA:0001:0000:0000:0000:0100
No Leading 0s	2001: DB8:AAAA: 1: 0: 0: 0: 100
":" All-0 segments	2001:DB8:AAAA:1::100
Preferred	2001:0DB8:AAAA:0001:0000:0000:0000:0200
No Leading 0s	2001: DB8:AAAA: 1: 0: 0: 0: 200
":" All-0 segments	2001:DB8:AAAA:1::200

Table 3-7 shows the preferred IPv6 address format that you began with and the final compressed format implementing both rules.

Table 3-7 *IPv6 Preferred and Compressed Format*

Preferred Format of IPv6 Addresses	Compressed Format
0000:0000:0000:0000:0000:0000:0000:0000	::
0000:0000:0000:0000:0000:0000:0000:0001	::1
FF02:0000:0000:0000:0000:0000:0000:0001	FF02::1
FC00:0001:A000:0B00:0000:0527:0127:00AB	FC00:1:A000:B00::527:127:AB
2001:DCBA:1111:000A:00B0:0000:9000:0200	2001:DCBA:1111:A:B0::9000:200
2002:0000:0000:0000:ABCD:0000:0000:1234	2002::ABCD:0:0:1234
2001:0DB8:AAAA:0001:0000:0000:0000:0100	2001:DB8:AAAA:1::100
2001:0DB8:AAAA:0001:0000:0000:0000:0200	2001:DB8:AAAA:1::200

Even with these rules to compress the format, an IPv6 address can still look unwieldy. Soon, you will look at a technique that is called the “3-1-4.” It will help you recognize the segments of an IPv6 address.

Prefix Notation

In IPv4, the prefix or network portion of the address can be identified by a dotted-decimal netmask, commonly referred to as a subnet mask. For example, 255.255.255.0 indicates that the network portion or prefix length of the IPv4 address is the leftmost 24 bits.

As defined in RFC 4291, IP Version 6 Addressing Architecture, the representation of IPv6 address prefixes is similar to the way that IPv4 address prefixes are written in classless interdomain routing (CIDR) notation. An IPv6 address prefix (network portion of the address) is represented using the following format:

ipv6-address/prefix-length

The *prefix length* is a decimal value indicating the number of leftmost contiguous bits of the address. The prefix length identifies the prefix or the network portion of the address.

Let’s look at an example using the address 2001:0DB8:AA AA:1111:0000:0000:0000:0000/64. Figure 3-2 illustrates how the /64 prefix length identifies the prefix or network portion of the address. The /64 prefix length leaves us with another 64 bits, which is the Interface ID portion of the address, known as the host portion in IPv4. I discuss the Interface ID in the next section.

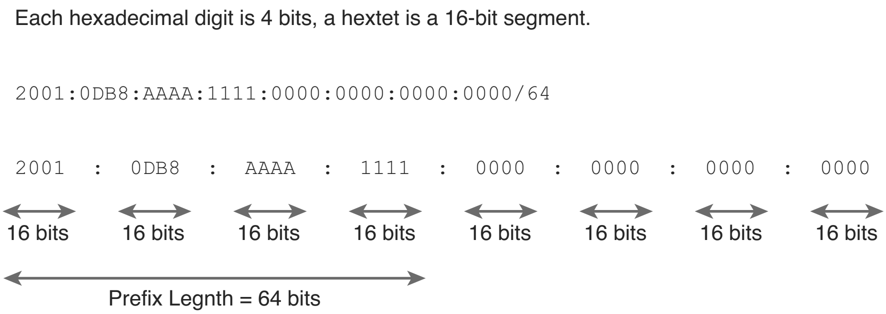


Figure 3-2 *IPv6 Prefix*

Using the rules learned for reducing the notation of an address, other valid representations of this address could also be written as

2001:0DB8:AAAA:1111:0:0:0:0/64

2001:0DB8:AAAA:1111::/64

2001:DB8:AAAA:1111::/64

Devices such as host computers would have an IPv6 address that is part of this prefix or network address, as shown in Figure 3-3. Using the topology from Chapter 2, two valid host addresses would be

2001:0DB8:AAAA:1111:0000:0000:0000:0100/64

or

2001:DB8:AAAA:1111::100/64

2001:0DB8:AAAA:1111:0000:0000:0000:0200/64

or

2001:DB8:AAAA:1111::0200/64

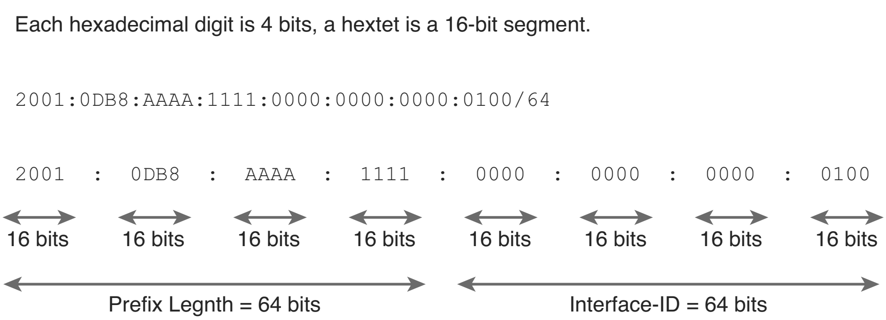


Figure 3-3 *IPv6 Prefix Length and Interface ID*

In IPv6, just as in IPv4, the number of devices you can have on your network depends on the prefix length. As discussed in Chapter 1, "Introduction to IPv6," as the Internet grew, the limited IPv4 address space was quickly becoming depleted. A customer request for an IPv4 address and a prefix length (subnet mask) to accommodate his network requirements must be justified by the customer. Most sites rely heavily on Network Address Translation (NAT) to accommodate the number of internal IPv4 hosts in their networks.

With IPv6, this is no longer the case. There is plenty of IPv6 address space. Many people have become accustomed to limiting the allocation of IPv4 addresses in networks. It's very common to use a /30 for point-to-point serial links on IPv4 networks. This is not a concern with IPv6, and it can be a difficult habit to break.

The Internet Architecture Board (IAB) and the Internet Engineering Steering Group (IESG) published a set of recommendations for IPv6 address allocations in RFC 3177, IAB/IESG Recommendations on IPv6 Address Allocations to Sites. This is a recommendation from the IAB and IESG to the five Regional Internet Registries (RIR). It is helpful to note that in their RFC, they state the following:

> "The technical principles that apply to address allocation seek to balance healthy conservation practices and wisdom with a certain ease of access. On one hand, when managing a potentially limited resource, one must conserve wisely to prevent exhaustion within an expected lifetime. On the other hand, the IPv6 address space is in no sense as limited a resource as the IPv4 address space, and unwarranted conservatism acts as a disincentive in a marketplace already dampened by other factors. So from a market development perspective, we would like to see it be very easy for a user or an ISP to obtain as many IPv6 addresses as they really need without a prospect of immediate renumbering or of scaling inefficiencies."

In RFC 3177, the IESG and the IAB recommended using specific prefix lengths for different size networks. One of their recommendations was that in general, all sites should get a /48, including home networks and small to large enterprise networks. The RIRs adopted that recommendation in 2002, but began reconsidering the policy in 2005. In 2011, RFC 6177, IPv6 Address Assignments to End Sites, obsoleted RFC 3177 and stated the following:

> "The exact choice of how much address space to assign end sites is an issue for the operational community. The IETF's role in this case is limited to providing guidance on IPv6 architectural and operational considerations. . . . Moreover, this document clarifies that a one-size-fits-all recommendation of /48 is not nuanced enough for the broad range of end sites and is no longer recommended as a single default."

What this means is that the RIR allocation of addresses to their customers, such as Internet service providers (ISP), will depend upon the RIR's own policies. American Registry for Internet Numbers (ARIN), the RIR for North America, has a current policy that it will allocate a minimum /32 to ISPs and a maximum of a /24, unless justified otherwise. End sites should get at least a /48 or a larger assignment if it can be justified. Because a /48 still seems to be the normal allocation for end sites, this will be used in these examples. RFC 6177 suggests that home sites might not need a /48, but something more like a /56. This gives ISPs more addresses to allocate, and the only difference to the home site is the number of subnets per home site network.

Note There are two types of addresses that can be assigned to an end-user organization: provider-independent (PI) and provider-aggregatable (PA). Provider-independent address space is assigned by an RIR directly to the end-user organization. Provider-independent address space allows organizations to change service providers without obtaining new address space. Provider-aggregatable address space is assigned by the RIR to the ISP. This allows the ISP to aggregate its address space for more efficient routing. These addresses belong to the ISP. Unlike PI addresses, if an end user changes providers, PA addresses cannot migrate with the end user.

As discussed in the next section, the typical host portion of an IPv6 unicast address is 64 bits, known as the Interface ID. If a site receives a /48 prefix, this allows 65,535 subnets, with 18,446,744,073,709,551,616 (18 quintillion) interface addresses (hosts) for each subnet! A /56 prefix for a home site means the same number of hosts per subnet but with only 256 subnets. This is still more than adequate for most homes.

Note A single host can have multiple interfaces, with each interface having one or more IPv6 addresses.

Brief Look at IPv6 Address Types

The following sections review the basic types of IPv6 addresses. They are examined in more detail in Chapter 4, "IPv6 Address Types." With IPv4, there are unicast, multicast, and broadcast addresses. In IPv6, there are no broadcast addresses. The three types of addresses in IPv6 are

- Unicast
- Anycast
- Multicast

Unicast Addresses

A unicast address uniquely identifies an interface on an IPv6 device. A packet sent to a unicast address is delivered to the interface identified by that address. An IPv6 address more accurately identifies an interface on a host rather than the host itself. A single interface can have multiple IPv6 addresses and an IPv4 address in as well.

There are several types of unicast addresses in IPv6, in particular

- Global unicast
- Unique local unicast (site-local was deprecated in September 2004)

- Link-local unicast
- Unspecified address
- Loopback address

There are also some special-purpose subtypes of global unicast, such as IPv6 addresses with embedded IPv4 addresses. The structure of a global unicast address is discussed later in the chapter.

Anycast Addresses

An anycast address is a unicast address assigned to several devices. A packet sent to an anycast address is delivered only to one of the devices configured with that address. The anycast packet will be routed to the nearest device.

There is an anycast address in IPv4 and, like IPv6, it is a common unicast address assigned to multiple devices. In both IPv4 and IPv6, an anycast address is syntactically indistinguishable from a unicast address. In IPv6, the devices to which the anycast address is assigned are explicitly configured to recognize that it is an anycast address. This is not necessarily the case in IPv4.

Multicast Addresses

A multicast address identifies a group of interfaces, typically belonging to different devices. A packet sent to a multicast address is delivered to all the devices identified by that address. All members of the multicast group process the packet. So, the difference between an anycast and a multicast address is that an anycast packet is only delivered to a single device, whereas multiple devices can receive a multicast packet.

There are no broadcast addresses in IPv6. In its place is an all-nodes multicast address.

Structure of a Global Unicast Address

The following sections examine the basic structure of a global unicast address. Global unicast addresses are also known as *aggregatable global unicast addresses*. These are addresses that are globally routable and reachable on the IPv6 Internet. They are equivalent to public IPv4 addresses.

Figure 3-4 shows the structure of a global unicast address for a typical site.

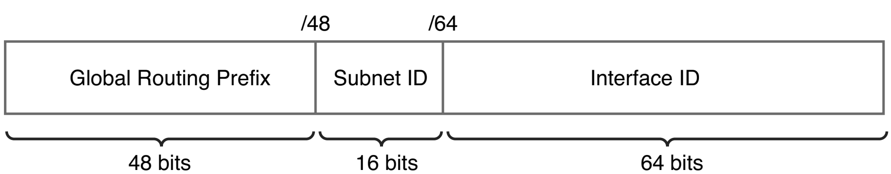


Figure 3-4 *Structure of a Global Unicast Address for a Typical Site*

Global Routing Prefix

The global routing prefix is the prefix or network portion of the address assigned by the provider, such as an ISP, to a customer or site. Although the IESG and IAB no longer recommend specific prefix lengths for different-size networks, it is still common for RIRs such as ARIN to have the policy for end sites to use a 48-bit prefix (/48). Figure 3-4 shows a typical /48 global routing prefix.

Note RFC 4291 does not specify the size of the Subnet ID. The 16-bit Subnet ID in Figure 3-4 results from a site receiving a /48 global routing prefix. With a 128-bit Interface ID, this leaves 16 bits for the Subnet ID. See RFC 3587, IPv6 Global Unicast Address Format, for more information.

Subnet ID

A big difference between IPv4 and IPv6 addresses is the location of the subnet portion of the address. In IPv4, bits are borrowed from the host portion of the address to create subnets. With IPv6, the Subnet ID is a separate field and is not part of the host portion of the address, known as the Interface ID in IPv6.

As shown in Figure 3-4, the IPv6 address has a 16-bit Subnet ID. This allows 65,536 individual subnets. Just in case you're wondering—yes, you can use the all 0s and the all 1s subnets. Subnetting is discussed in the next section.

Interface ID

The Interface ID uniquely identifies the interface on the subnet. As shown earlier, the 64-bit Interface ID allows 18,446,744,073,709,551,616 (18 quintillion) addresses for each subnet! The term *Interface ID* is used rather than *Host ID* because, as mentioned previously, a single host can have multiple interfaces, each having one or more IPv6 addresses.

Another important difference between the IPv6 and IPv4 addresses is that the all-0s and all-1s addresses are legal IPv6 interface addresses. An IPv6 Interface ID can contain all 0s or all 1s. In IPv4, all 0s in the host portion of the address are reserved for the network or subnet address. All 1s in the host portion of an IPv4 address indicate a broadcast address. Remember, there is no broadcast address in IPv6.

3-1-4 Rule

IPv6 global unicast addresses can look complicated, and it can be difficult to recognize all the parts. A simple technique that I created to quickly break it down is my 3-1-4 rule, as shown in Figure 3-5. Each number refers to the number of hextets, or 16-bit segments, of that portion of the address. Perhaps an easy way to remember these numbers is to call it the *pi rule* (pi = 3.14).

- **3:** This indicates the three hextets, or 48 bits, of the Global Routing Prefix.
- **1:** This indicates the one hextet, or 16 bits, of the Subnet ID.
- **4:** This indicates the four hextets, or 64 bits, of the Interface ID.

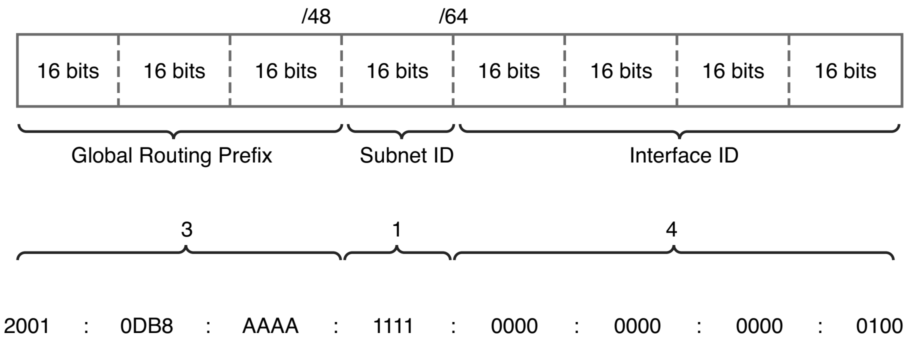


Figure 3-5 *Global Unicast Addresses and the 3-1-4 Rule*

Note This technique is useful whenever the global unicast address has a /48 global routing prefix and a 64-bit Interface ID, which is a common prefix allocation. As discussed later in this chapter and later in the book, Global Routing Prefixes and Interface IDs do not necessarily have to be 48 bits and 64 bits, respectively.

Table 3-8 shows several examples of /48 global unicast addresses using the 3-1-4 technique. Although the double colon compresses the notation of the address, it can sometimes make it more difficult to recognize the three parts of the address. Sometimes it can be easier to start from the Interface ID, or from both ends toward the middle Subnet ID.

Table 3-8 *Examples of /48 Global Unicast Addresses with the 3-1-4 Technique*

/48 Global Unicast Address	Global Routing Prefix 3	Subnet ID 1	Interface ID 4
2001:0DB8:AAAA:1234:1111:2222:3333:4444	2001:0DB8:AAAA	1234	1111:2222:3333:4444
2001:0DB8:BBBB:4321:AAAA:BBBB:CCCC:DDDD	2001:0DB8:BBBB	4321	AAAA:BBBB:CCCC:DDDD
2001:0DB8:AAAA:0001:0000:0000:0000:0100	2001:0DB8:AAAA	0001	0000:0000:0000:0100
2001:0DB8:AAAA:9:0:0:0:A	2001:0DB8:AAAA	0009	0000:0000:0000:000A
2001:0DB8:AAAA:0001::0200	2001:0DB8:AAAA	0001	0000:0000:0000:0200
2001:DB8:AAAA::200	2001:0DB8:AAAA	0000	0000:0000:0000:0200

/48 Global Unicast Address	Global Routing Prefix 3	Subnet ID 1	Interface ID 4
2001:DB8::ABC:0	2001:0DB8:0000	0000	0000:0000:0ABC:0000
2001:DB8:ABC::	2001:0DB8:0ABC	0000	0000:0000:0000:0000
2001:DB8:ABC::FFFF:FFFF:FFFF:FFFF	2001:0DB8:0ABC	0000	FFFF:FFFF:FFFF:FFFF
2001:DB8::FFFF:FFFF:FFFF:FFFF:FFFF	2001:0DB8:0000	FFFF	FFFF:FFFF:FFFF:FFFF

Notice that both of the following addresses are legal interface (host) addresses in IPv6:

- **All 0s address:** 2001:DB8:ABC:: or 2001:0DB8:0ABC:0000:0000:0000:0000:0000
- **All 1s address:** 2001:DB8::FFFF:FFFF:FFFF:FFFF:FFFF or 2001:DB8:0000:FFFF:FFFF:FFFF:FFFF:FFFF

Putting It Together

You should now have a basic understanding of IPv6 global unicast addresses. Figure 3-6 illustrates your IPv6 network.

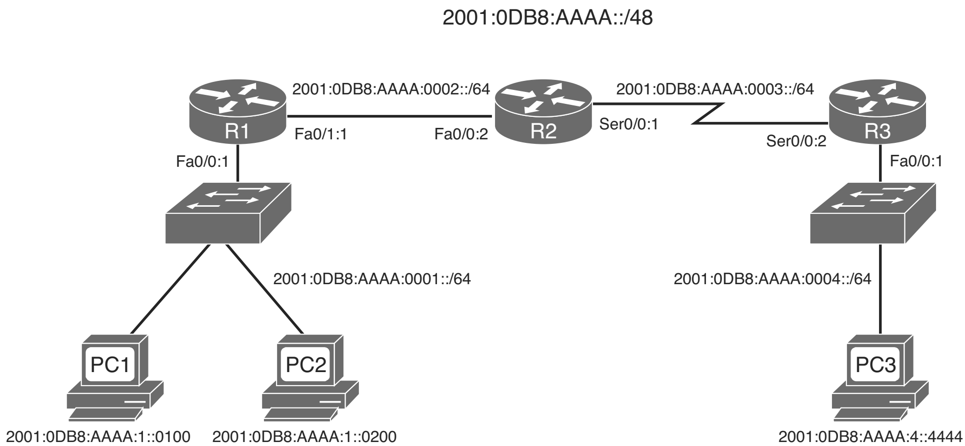


Figure 3-6 *IPv6 Topology*

Begin with using the 2001:0DB8:AAAA::/48 address. These first three hextets identify the global routing prefix or, in other words, the IPv6 network addresses that have been received from the provider. The /48 network is divided into four /64 subnets—0001, 0002, 0003, and 0004. The four subnets are

- 2001:0DB8:AAAA:0001::/64
- 2001:0DB8:AAAA:0002::/64

- 2001:0DB8:AAAA:0003::/64
- 2001:0DB8:AAAA:0004::/64

Using the 3-1-4 technique, you can quickly see that the first three hextets comprise the Global Routing Prefix (2001:0DB8:AAAA) and the fourth hextet is the Subnet ID.

Note 2001:0DB8:AAAA::/48 is part of the 2001:0DB8::/32 block of addresses reserved for examples and documentation.

Configuring an IPv6 address on a router's interface is very similar to that of IPv4. As shown throughout this book, most of the commands are identical, except the parameter **ipv6** is used in place of **ip**.

Table 3-9 illustrates the commands necessary to manually configure an IPv6 address on a router's interface. Chapter 4 explains some of the optional parameters for the **ipv6 address** command.

Note IPv6 address configuration is discussed in Chapter 4. The **ipv6 address** interface command is only included here to show the similarity between IPv6 and IPv4 commands. The global configuration command **ipv6 unicast-routing** is required to enable a router to route IPv6 packets. This command is also discussed in Chapter 4.

Table 3-9 *ipv6 address* Command

Command	Description
Router(config)# **interface** *interface-type interface-number*	Specifies the interface type and interface number.
Router(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Specifies the IPv6 address and prefix length to be assigned to the interface. To remove the address from the interface, use the **no** form of this command.

Now configure the devices for the first subnet. Using the 3-1-4 technique and the topology shown in Figure 3-6, all three components of the address are easily recognizable, as shown in Table 3-10.

Table 3-10 *IPv6 Address Chart for 2001:0DB8:AAAA:0001::/64*

Device	Global Routing Prefix 3	Subnet ID 1	Interface ID 4
Router R1	2001:0DB8:AAAA	0001	0000:0000:0000:0001
PC1	2001:0DB8:AAAA	0001	0000:0000:0000:0100
PC2	2001:0DB8:AAAA	0001	0000:0000:0000:0200

Example 3-1 shows the commands for configuring Router R1's Fast Ethernet 0/0 interface with the IPv6 address 2001:0DB8:AAAA:0001::0100 and the prefix length /64.

Example 3-1 *Cisco Router IPv6 Interface Configuration*

```
R1(config)# interface fastethernet 0/0
R1(config-if)# ipv6 address 2001:0db8:aaaa:0001::1/64
R1(config-if)# no shutdown
R1(config-if)# end
R1#
```

Note When configuring an IPv6 address in Cisco IOS, there is not a space between the IPv6 address and the prefix length.

Figure 3-7 shows PC1's IPv6 address configuration. Router R1's IPv6 address is being used for the PC's default gateway. You can verify the configuration on both Windows PCs by using the **ipconfig** command. In this example, the IPv6 address is configured manually. It is more likely that end devices will receive their IPv6 address configuration dynamically using Stateless Address Autoconfiguration or a DHCPv6 server. Both of these techniques are discussed in Chapter 4.

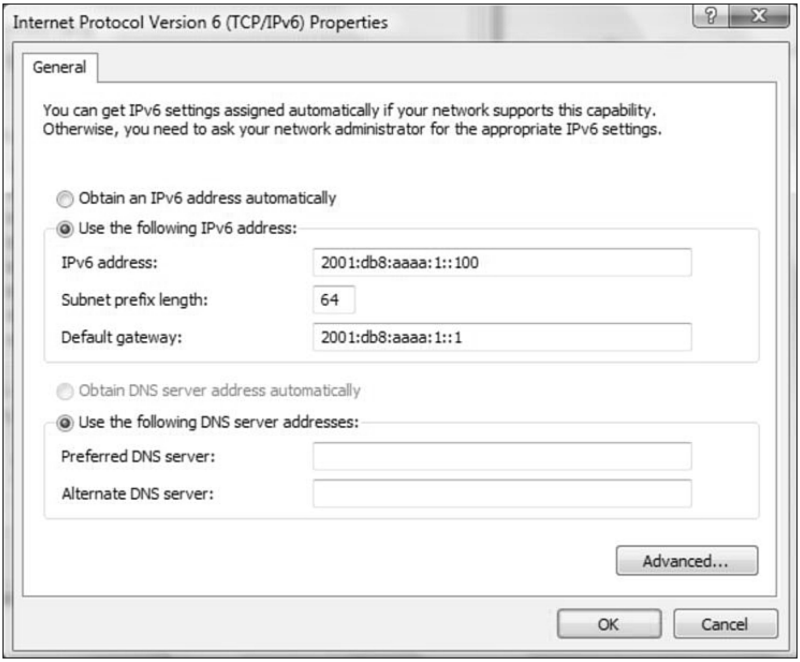


Figure 3-7 *PC1 IPv6 Interface Configuration*

In Example 3-2, IPv6 communications are verified by pinging PC1 and PC2. Notice that the same **ping** command is used as with IPv4. The only difference is that the destination is an IPv6 address. The **ping** command sends ICMPv6 Echo Request messages to the IPv6 destination address. The ! indicates that the ICMPv6 Echo Reply messages from the destination interface have been received, therefore verifying end-to-end communications. ICMPv6 is discussed in more detail in Chapter 5, "ICMPv6 and Neighbor Discovery Protocol."

Example 3-2 *Router R1 Pinging PC1 and PC2*

```
R1# ping 2001:db8:aaaa:0001::0100

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:AAAA:1::100, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R1# ping 2001:db8:aaaa:0001::0200

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:AAAA:1::200, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
R1#
```

Subnetting

Depending on the size of environment, developing an IPv6 addressing can require substantial planning. However, basic subnetting of an IPv6 address is very straightforward. In many ways, it is much simpler than subnetting an IPv4 address. Unless you are subnetting on a natural octet boundary in IPv4, the specific subnets are not always obvious.

Note RFC 5375, IPv6 Unicast Address Assignment Considerations, offers guidelines for subnet prefix considerations. Many RIRs also offer guidelines to assist in developing an IPv6 addressing plan:

- ARIN's IPv6 Addressing Plans: www.getipv6.info/index.php/IPv6_Addressing_Plans
- RIPE's Preparing an IPv6 Addressing Plan: https://labs.ripe.net/Members/steffann/preparing-an-ipv6-addressing-plan

It is important to clarify a couple of terms. As illustrated in Figure 3-8, there is both a Subnet ID and a Subnet Prefix. The term *Subnet ID* refers to the contents of the 16-bit field used to allocate individual subnets. Subnet Prefix refers to the Global Routing Prefix and the Subnet ID addressing bits.

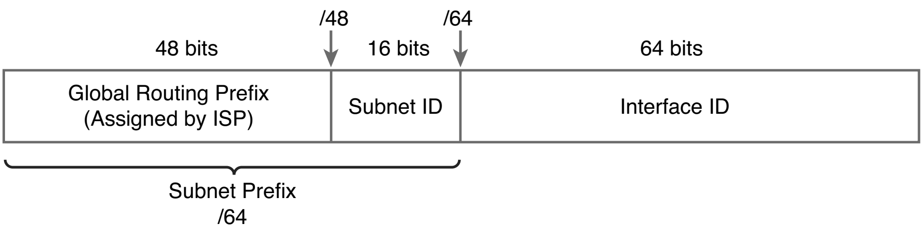


Figure 3-8 *Subnet Prefix*

A typical IPv6 site prefix will have a /48 assigned by its provider, usually from an ISP. This creates a 16-bit Subnet ID, allowing 216, or 65,536, subnets. The all-0s and all-1s subnets are valid subnets. This leaves 64 bits for the Interface ID, giving us 264, or 18 quintillion, interfaces (hosts) per subnet. Site prefixes and subnetting are examined later in Chapter 4.

Using the topology in Figure 3-6, the /48 network has been segmented into four /64 subnets, as shown in Table 3-11.

Table 3-11 *IPv6 Address Chart for 2001:0DB8:AAAA:0001::/64*

Subnet Prefix	
Global Routing Prefix	**Subnet ID**
2001:0DB8:AAAA:	0001
2001:0DB8:AAAA:	0002
2001:0DB8:AAAA:	0003
2001:0DB8:AAAA:	0004

Valid abbreviations for the four subnets are

- 2001:DB8:AAAA:1::/64
- 2001:DB8:AAAA:2::/64
- 2001:DB8:AAAA:3::/64
- 2001:DB8:AAAA:4::/64

With a 16-bit Subnet ID, the values can range from 0000 to FFFF, which provides 65,536 total subnets. Subnetting is painless because you can start with 0000 and increment by 1. Remember that this is in hexadecimal, so after 0009, the next Subnet ID would be 000A. Subnetting by using the 16-bit Subnet ID is easy to perform, as illustrated in Table 3-12.

Table 3-12 *Subnetting Using the 16-bit Subnet ID*

Range	Subnet ID
First 16 subnets	0000
	0001
	0002
	through . . .
	0009
	000A
	000B
	000C
	000D
	000E
	000F

Range	Subnet ID
Next 16 subnets	0010
	0011
	0012
	through . . .
	001F
Next 16 subnets	0020
	0021
	0022
	through . . .
	002F
And so on	0030
	0031
	Etc.

Extending the Subnet Prefix

Subnetting is not limited to a 16-bit Subnet ID. Any number of subnet bits can be chosen for the Subnet ID. Just as with IPv4, if you want to extend the number of subnets or more likely to reduce the number of hosts per subnet, you must borrow bits from the Interface ID. It is important to note that best practice dictates that this should only be done on network infrastructure links. Any segment that includes end systems should stay with a /64 prefix. A /64 prefix length is required for supporting Stateless Address Autoconfiguration.

As shown in Figure 3-9, you can use a /112 prefix length, extending the original /48 prefix by 64 bits (four hextets), giving it a prefix of /112.

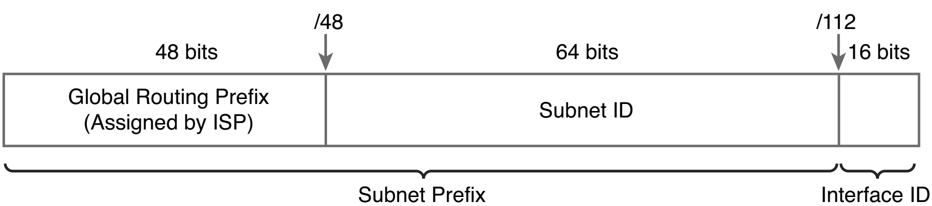


Figure 3-9 */112 Subnet Prefix*

The first four subnets would be

- 2001:0DB8:AAAA:0000:0000:0000:0000::/112
- 2001:0DB8:AAAA:0000:0000:0000:0001::/112
- 2001:0DB8:AAAA:0000:0000:0000:0002::/112
- 2001:0DB8:AAAA:0000:0000:0000:0003::/112

Figure 3-10 shows the range of subnet prefixes using a /112 prefix length.

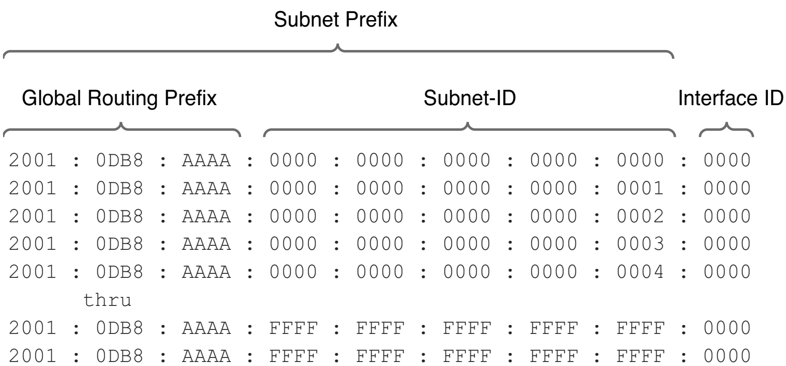


Figure 3-10 *Range of /112 Subnet Prefixes*

Note SURFnet, a nonprofit organization that forms partnerships between Dutch higher-education and research institutions in Information Technology, held a workshop with its customers on IPv6. From this workshop, an IPv6 addressing plan was developed. The document, Preparing an IPv6 Addressing Plan, can be downloaded from the Regional Internet Registry RIPE's website:

https://labs.ripe.net/Members/steffann/preparing-an-ipv6-addressing-plan

Even with extending the Subnet ID, subnetting is very straightforward as long as you subnet on a nibble boundary.

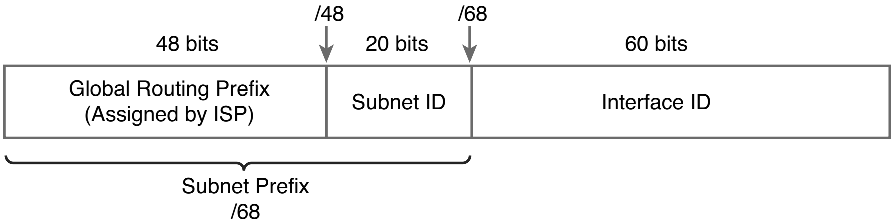
Subnetting on a Nibble Boundary

If you are going to extend the Subnet ID, which means using bits from the Interface ID, it is best practice to subnet on a nibble boundary. A nibble is 4 bits, as shown in Table 3-13.

Table 3-13 *Decimal, Hexadecimal, and Binary Chart*

Decimal	Hexadecimal	Binary (Nibble)
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

In Figure 3-11, you are extending the /64 subnet prefix by 4 bits, one nibble, to a /68. This increases the Subnet ID from 16 bits to 20 bits. By doing so, this will allow more subnets but reduce the size of the Interface ID. In this case, there isn't any practical reason for doing this except to illustrate the concept. By extending the Subnet Prefix by 4 bits, or one full nibble, you are implementing the best practice of subnetting on a nibble boundary. Using 20 bits, a factor of 4 bits makes it very easy to list the subnets. This is illustrated in Figure 3-11.



Subnetting on a nibble (4 bit) boundary makes it easier to list the subnets.

```
2001:0DB8:AAAA:0000:0000::/68
2001:0DB8:AAAA:0000:1000::/68
2001:0DB8:AAAA:1111:2000::/68
     thru
2001:0DB8:AAAA:FFFF:F000::/68
```

Figure 3-11 *Subnetting on a Nibble Boundary*

Subnetting Within a Nibble

For most customer networks, subnetting within a nibble is not recommended. It provides little if any benefits and only makes implementation and troubleshooting more difficult. However, there can be cases when subnetting on a nibble is potentially wasteful and it is beneficial to subnet within the 4-bit nibble. The addressing plans discussed previously help address some of those instances.

When you subnet within a nibble, life becomes a little more problematic. In Figure 3-12, you are using a /70 subnet prefix, extending the simple /68 to a more difficult /70. Because it is extended by only 2 bits instead of a nibble (4 bits), it makes the conversion a little more troublesome. Of course it is perfectly valid; it just makes it more cumbersome to convert.

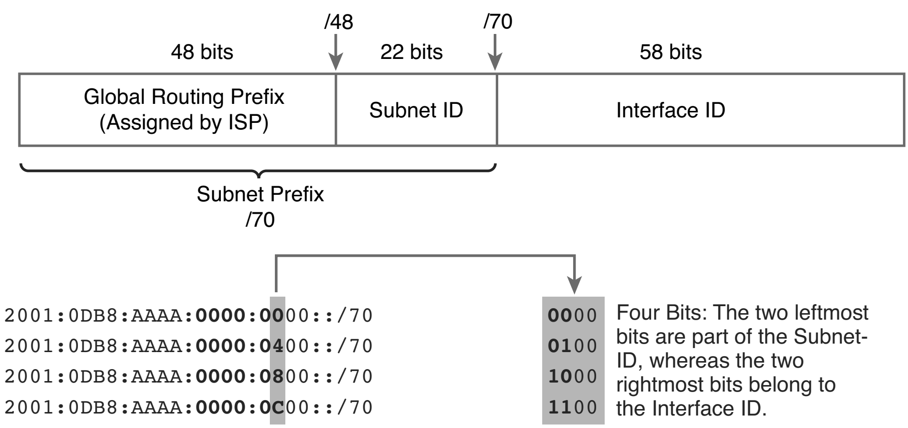


Figure 3-12 *Subnetting Within a Nibble*

Looking at the first four subnets, you get

- 2001:0DB8:AAAA:**0000:00**00::/70
- 2001:0DB8:AAAA:**0000:04**00::/70
- 2001:0DB8:AAAA:**0000:08**00::/70
- 2001:0DB8:AAAA:**0000:0C**00::/70

The first subnet is easy enough to figure out, but right away, you can see that the second subnet requires a little more thinking. IPv6 addresses use hexadecimal values to represent each 4 bits. Because a /70 subnet prefix was chosen, the first half of the last hexadecimal digit belongs to the Subnet ID and the other half belongs to the Interface ID. So, only the first 2 bits of the last digit of the Subnet ID are modified, as shown in Table 3-14.

Table 3-14 *Subnetting Within a Nibble*

Subnetting Within a Nibble	Last Digit of Subnet ID Binary to Hexadecimal
2001:0DB8:AAAA:0**000:00**00::/70	0000 = 0
2001:0DB8:AAAA:**0000:04**00::/70	0100 = 4
2001:0DB8:AAAA:**0000:08**00::/70	1000 = 8
2001:0DB8:AAAA:**0000:0C**00::/70	1100 = C

Limiting the Interface ID Space

Although IPv6 address space is plentiful, there can be reasons for limiting the size of the Interface ID within a network infrastructure. There is much debate on this issue. This section includes a brief explanation for a basic understanding of the topic.

RFC 6164, Using 127-Bit IPv6 Prefixes on Inter-Router Links, recommends employing 127-bit IPv6 prefixes (/127) on inter-router point-to-point links, for security and other reasons. This is similar to the use of 31-bit (/31) prefixes in IPv4. An issue of considerable debate centers around what is known as an NDP (Neighbor Discovery Protocol) exhaustion attack.

A 64-bit Interface ID allows an abundance of interface addresses, more than 18 quintillion interfaces (hosts) per subnet. IPv4 hosts have Address Resolution Protocol (ARP) caches to maintain a list of IPv4 addresses and their relative Layer 2 MAC addresses. In IPv6, there is something similar known as a Neighbor Cache, which is also explained in Chapter 5.

There is concern that with such a large IPv6 Interface ID space, an attacker could send a continuous stream of packets to routers or other devices on the subnet from hundreds of millions of fake source IPv6 addresses. This would cause the recipient to create a Neighbor Cache for each address, consuming large amounts of memory. Depending upon the type of packet sent, it might cause the recipient to respond with a large number of Neighbor Solicitation packets that will never receive replies, thereby consuming large amounts of memory and processing. (Neighbor Solicitation messages are part of NDP and will also be discussed in Chapter 5.) This is the equivalent to filling a device's ARP cache in IPv4.

RFC 3756, IPv6 Neighbor Discovery (ND) Trust Models and Threats, propose some techniques to address the NDP exhaustion attack issue such as Secure Neighbor Discover (SeND). The ultimate resolution is the source of much debate and beyond the scope of an IPv6 Fundamentals book. For more information on this topic, read "IPv6 NDP Table Exhaustion Attack," by Jeff S Wheeler. His presentation can be downloaded from http://inconcepts.biz/~jsw/IPv6_NDP_Exhaustion.pdf.

There are also reasons for using the larger 64-bit Interface ID. As shown in Chapter 4, Stateless Address Autoconfiguration (SLAAC) used on segments with end systems requires a /64 prefix and a 64-bit Interface ID.

Summary

This chapter explained the basics of IPv6 addressing. The preferred format of an IPv6 128-bit address is written as eight 16-bit segments (hextets) and separated by colons. The notation of the address can be reduced by omitting leading 0s and by using the double colon to replace contiguous hextets of 0s.

The IPv6 prefix length was discussed as well as the recommended size of the global routing prefix for different organizations. There was an introduction to the different IPv6 address types: unicast, anycast, and multicast.

The basic structure of a global unicast address has three parts: the Global Routing Prefix, Subnet ID, and Interface ID. Using the 3-1-4 technique can help to recognize the hextets in a /48 global unicast address.

The simplicity of subnetting an IPv6 address using the Subnet ID or extending the Subnet ID on a nibble boundary was also acknowledged. Subnetting within a nibble can be done, but it is a little more difficult and not recommended.

References

RFCs:

RFC 2374, *An IPv6 Aggregatable Global Unicast Address Format*, R. Hinden, Nokia, IETF, www.ietf.org/rfc/rfc2374.txt, July 1998

RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*, S. Deering, Cisco Systems, IETF, www.ietf.org/rfc/rfc2460.txt, December 1998

RFC 3177, *IAB/IESG Recommendations on IPv6 Addresses*, IAB, IESG, www.ietf.org/rfc/rfc3177.txt, September 2001

RFC 3587, *IPv6 Global Unicast Address Format*, R. Hinden, Nokia, IETF, www.ietf.org/rfc/rfc3587.txt, August 2003

RFC 3756, *IPv6 Neighbor Discovery (ND) Trust Models and Threats*, P. Nikander, Ericsson Research Nomadic Lab, IETF, www.ietf.org/rfc/rfc3756.txt, May 2004

RFC 4291, *IP Version 6 Addressing Architecture*, R. Hinden, Nokia, IETF, www.ietf.org/rfc/rfc4291.txt, February 2006

RFC 5453, *Reserved IPv6 Interface Identifiers*, S. Krishnan, www.ietf.org/rfc/rfc5453.txt, February 2009

RFC 6164, *Using 127-Bit IPv6 Prefixes on Inter-Router Links*, M. Kohno, Juniper Networks, www.ietf.org/rfc/rfc6164.txt, April 2011

RFC 6177, *IPv6 Address Assignment to End Sites*, IAB, IESG, www.ietf.org/rfc/rfc6167.txt, March 2011

Websites:

IPv6 NDP Table Exhaustion Attack, Jeff S Wheeler, http://inconcepts.biz/~jsw/IPv6_NDP_Exhaustion.pdf

AfriNIC IPv6 Resource website: www2.afrinic.net/IPv6/index.htm

ARIN IPv6 Resource website: http://ipv6.com/articles/general/ipv6-the-next-generation-internet.htm

APNIC IPv6 Resource website: http://icons.apnic.net/display/IPv6/home

LACNIC IPv6 Resource website: http://portalipv6.lacnic.net

RIPE NCC IPv6 Resource website: www.ripe.net/internet-coordination/ipv6

This page intentionally left blank

Index

Numerics

A

B

C

D

E

F

G

H

I

J-K

L

M

N

O

P

Q-R

S

T

U

V

W-X-Y-Z