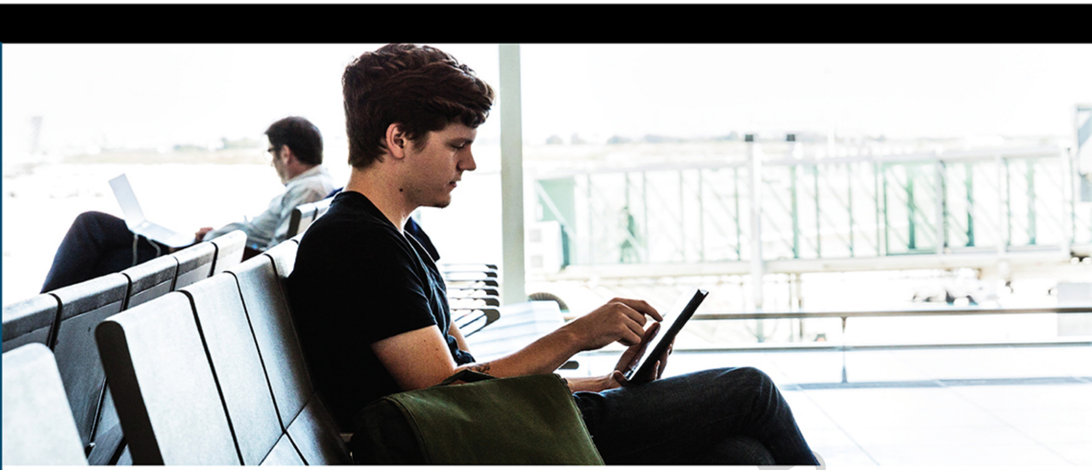


CCNA 200-301

Portable Command Guide

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

Fifth Edition

ciscopress.com

Scott Empson

CCNA 200-301 Portable Command Guide, Fifth Edition

Scott Empson

Published by:

Cisco Press

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

Part II: LAN Switching Technologies

Part III: Routing Technologies

Part IV: IP Services

Part V: Security Fundamentals

Part VII: Appendices

CHAPTER 4

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.

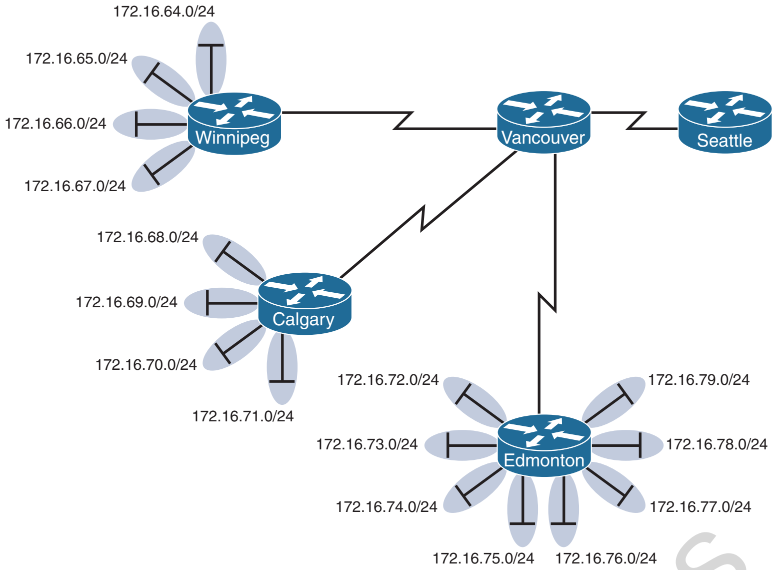


Figure 4-1 Four-City Network Without Route Summarization

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 = **10101100.00010000.010000**00.00000000

172.16.65.0 = **10101100.00010000.010000**01.00000000

172.16.66.0 = **10101100.00010000.010000**10.00000000

172.16.67.0 = **10101100.00010000.010000**11.00000000

Common bits: **10101100.00010000.010000**xx

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**00.00000000

172.16.69.0 = **10101100.00010000.010001**01.00000000

172.16.70.0 = **10101100.00010000.010001**10.00000000

172.16.71.0 = **10101100.00010000.010001**11.00000000

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**000.00000000

172.16.73.0 = **10101100.00010000.01001**001.00000000

172.16.74.0 = **10101100.00010000 01001**010.00000000

172.16.75.0 = **10101100.00010000 01001**011.00000000

172.16.76.0 = **10101100.00010000.01001**100.00000000

172.16.77.0 = **10101100.00010000.01001**101.00000000

172.16.78.0 = **10101100.00010000.01001**110.00000000

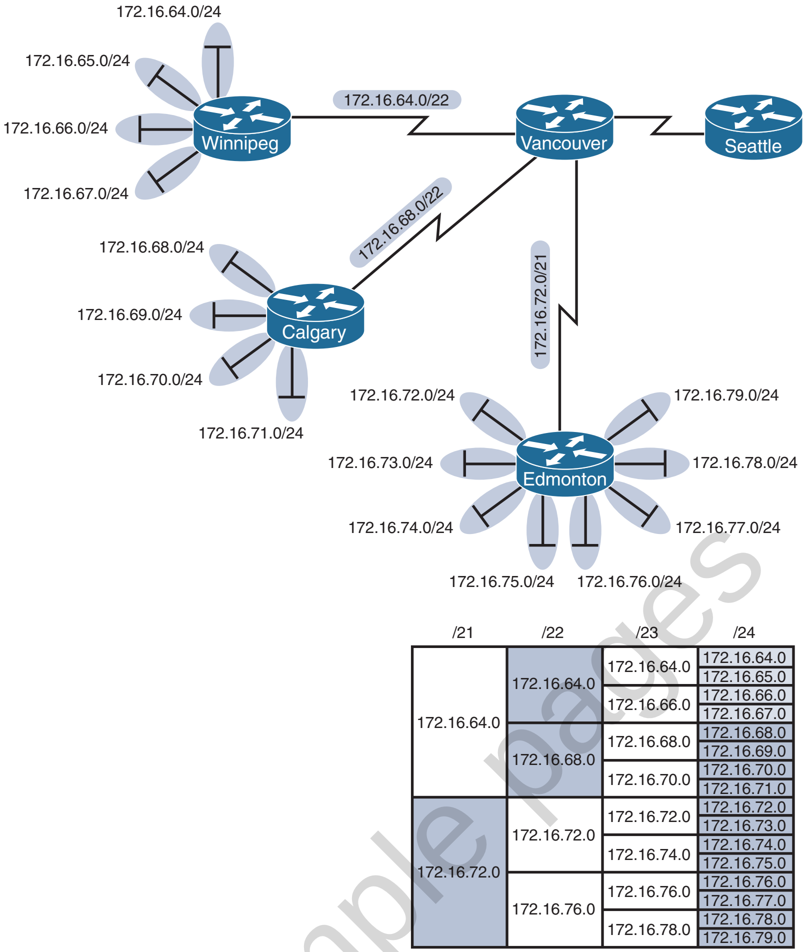
172.16.79.0 = **10101100.00010000.01001**111.00000000

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.



/21	/22	/23	/24
172.16.64.0	172.16.64.0	172.16.64.0	172.16.64.0
			172.16.65.0
		172.16.66.0	172.16.66.0
			172.16.67.0
	172.16.68.0	172.16.68.0	172.16.68.0
			172.16.69.0
		172.16.70.0	172.16.70.0
			172.16.71.0
172.16.72.0	172.16.72.0	172.16.72.0	172.16.72.0
			172.16.73.0
		172.16.74.0	172.16.74.0
			172.16.75.0
	172.16.76.0	172.16.76.0	172.16.76.0
			172.16.77.0
		172.16.78.0	172.16.78.0
			172.16.79.0

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 = **10101100.00010000.01000**000.00000000

172.16.68.0 = **10101100.00010000.01000**100.00000000

172.16.72.0 = **10101100.00010000.01001**000.00000000

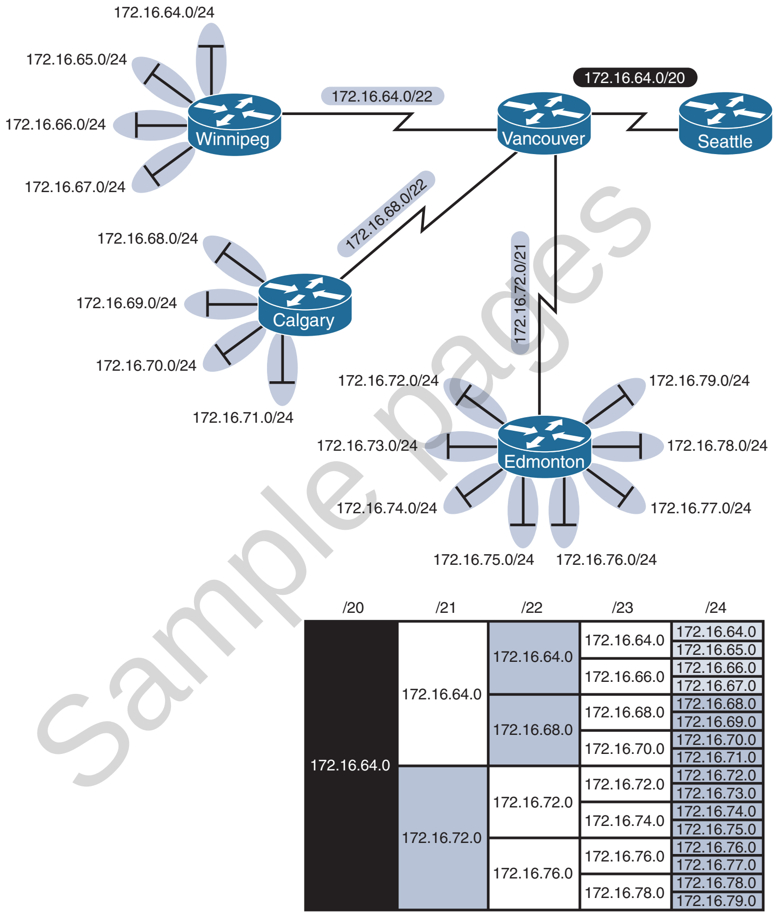
Common bits: **10101100.00010000.0100**xxxx

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.



/20	/21	/22	/23	/24
172.16.64.0	172.16.64.0	172.16.64.0	172.16.64.0	172.16.64.0
				172.16.65.0
			172.16.66.0	172.16.66.0
				172.16.67.0
		172.16.68.0	172.16.68.0	172.16.68.0
				172.16.69.0
			172.16.70.0	172.16.70.0
				172.16.71.0
	172.16.72.0	172.16.72.0	172.16.72.0	172.16.72.0
				172.16.73.0
			172.16.74.0	172.16.74.0
				172.16.75.0
		172.16.76.0	172.16.76.0	172.16.76.0
				172.16.77.0
			172.16.78.0	172.16.78.0
				172.16.79.0

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.

CHAPTER 5

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

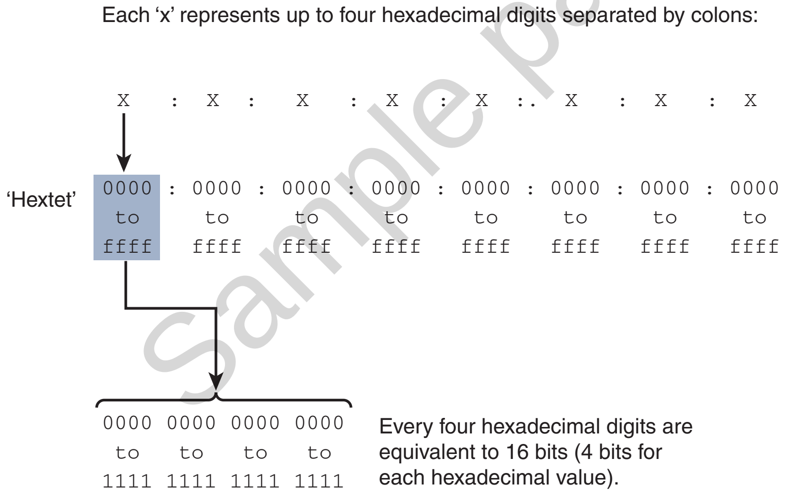
NOTE: This chapter is meant to be a very high-level overview of IPv6 addressing. For an excellent overview of IPv6, I strongly recommend you read Rick Graziani's book from Cisco Press: *IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6*, Second Edition. It is a brilliant read, and Rick 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 *hextets*) for a total of 128 bits in the address. Figure 5-1 shows this format.


Each 'x' represents up to four hexadecimal digits separated by colons:

Every four hexadecimal digits are equivalent to 16 bits (4 bits for each hexadecimal value).

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 hexadecimals 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	**000**0:**000**0:**000**0:**000**0:**000**0:**000**0:**000**0:**000**0
Leading 0s omitted	0: 0: 0: 0: 0: 0: 0: 0 or 0:0:0:0:0:0:0:0
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	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

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**: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	**000**0:**000**0:**000**0:**000**0:**000**0:**000**0:**000**0:**000**0
Leading 0s omitted	0: 0: 0: 0: 0: 0: 0: 0
(::) All-0s segments	::
Compressed	::
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
(::) All-0s segments	::1
Compressed	::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
(::) All-0s segments	ff02::1
Compressed	ff02::1
Preferred	fe80:**000**0:**000**0:**000**0: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:**0**db8:aaaa:0001:**000**0:**000**0:**000**0:**0**200
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.

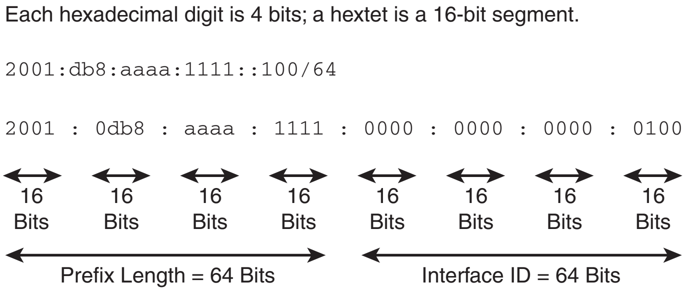

Each hexadecimal digit is 4 bits; a hextet is a 16-bit segment.
2001:db8:aaaa:1111::100/64
2001 : 0db8 : aaaa : 1111 : 0000 : 0000 : 0000 : 0100
16 Bits
16 Bits
16 Bits
16 Bits
16 Bits
16 Bits
16 Bits
16 Bits
Prefix Length = 64 Bits
Interface ID = 64 Bits

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.

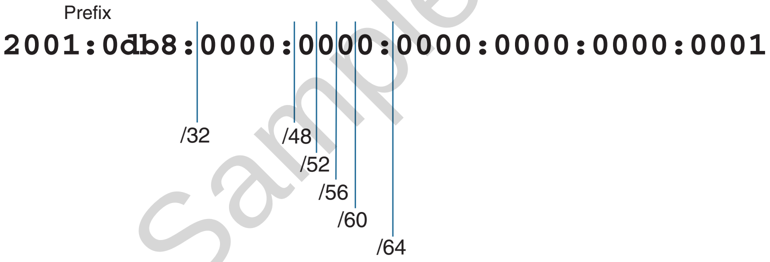


Figure 5-3 IPv6 Prefix Length Examples

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.

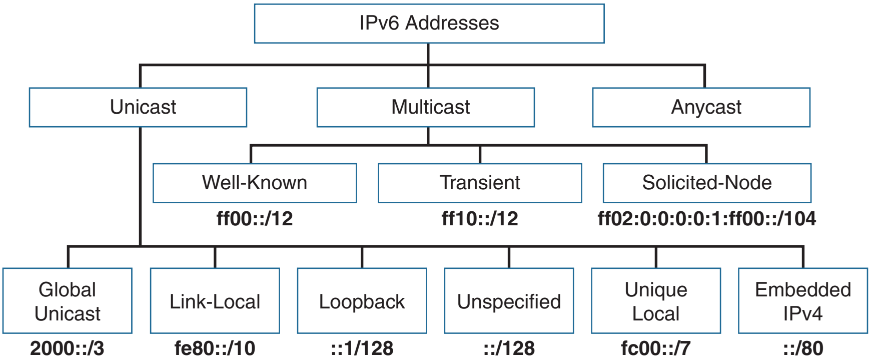


Figure 5-4 IPv6 Address Types

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.

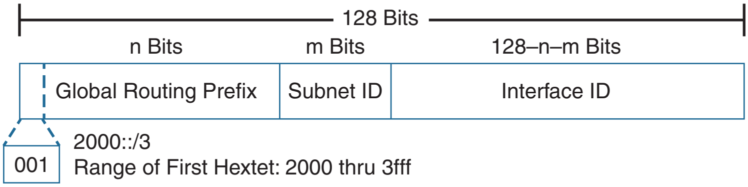


Figure 5-5 Structure of a Global Unicast Address

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 hextet being from fe80 to febf.

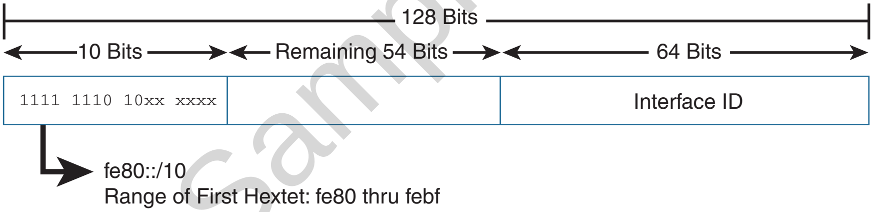


Figure 5-6 Structure of a Link-Local Unicast Address

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.

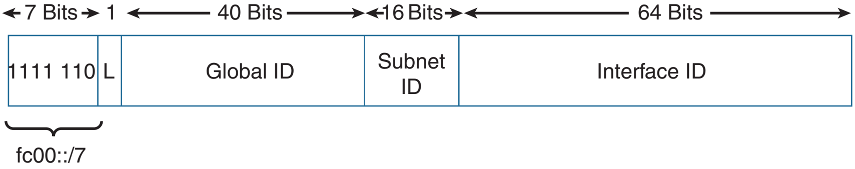


Figure 5-7 Structure of a Unique Local Unicast Address

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.

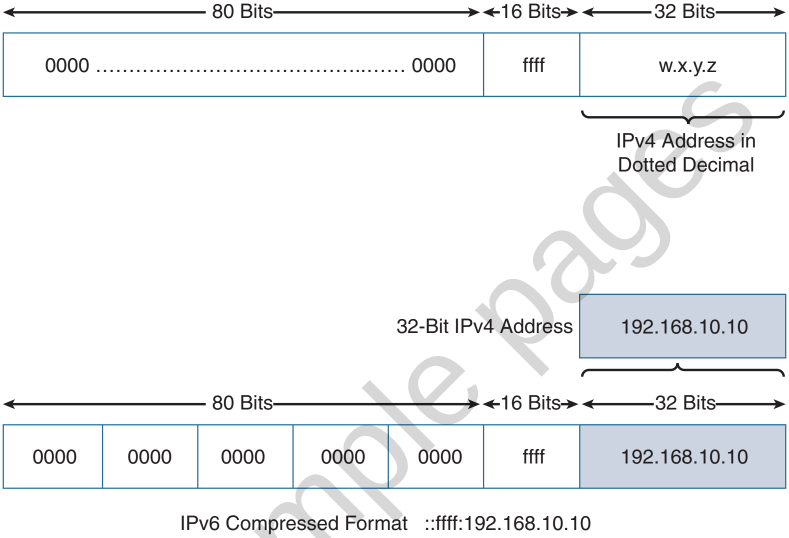


Figure 5-8 IPv4-Mapped IPv6 Address

> **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.

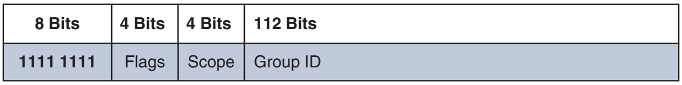
Unlike IPv4, there is no broadcast address in IPv6. Instead, IPv6 uses multicast.

Table 5-6 shows IPv6 multicast address representation.

TABLE 5-6 IPv6 Multicast Address Representation

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

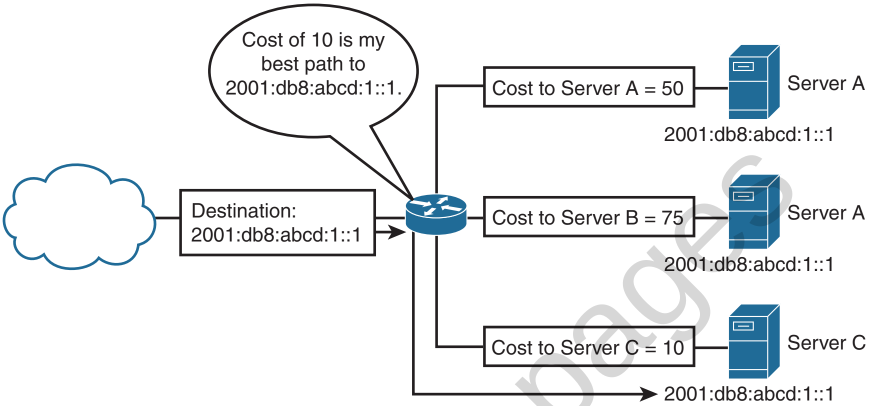


Figure 5-10 Example of Anycast Addressing

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