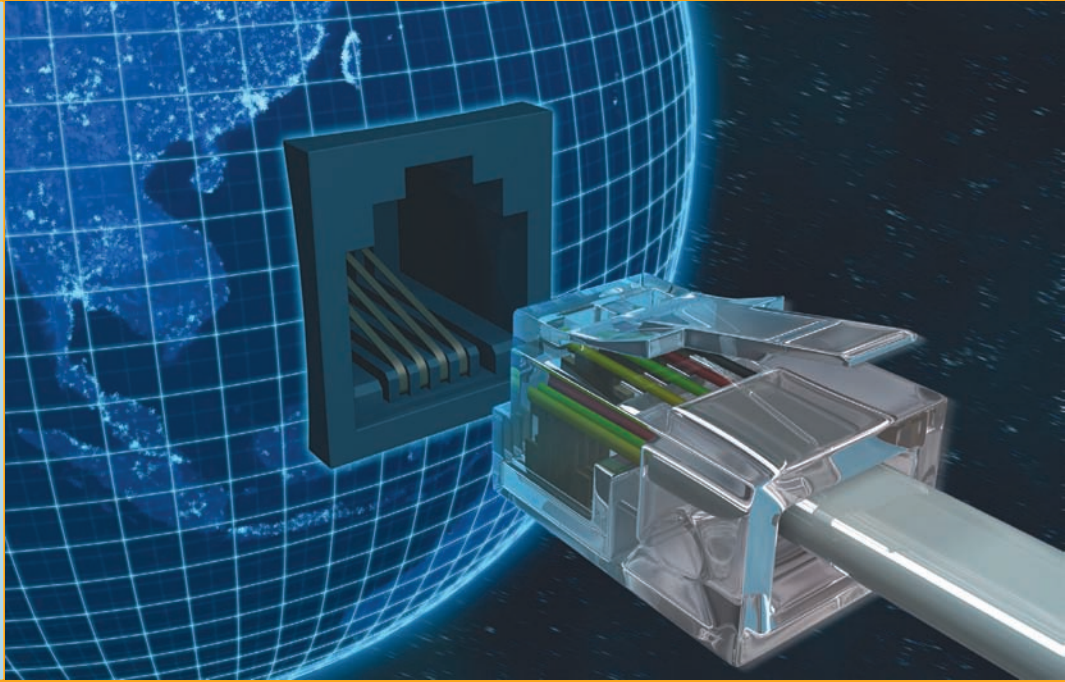


*“Give a man a fish and he will eat for a day. Teach a man to fish and he will eat for a lifetime. Teach a man to create an artificial shortage of fish and he will eat steak.”*

—JAY LENO



## In this chapter, you will learn how to

- Discuss the fundamental concepts of IPv6
- Describe IPv6 practices
- Implement IPv6 in a TCP/IP network

The Internet developers wanted to make a networking protocol that had serious longevity, so they had to define a large enough IP address space to last well beyond the foreseeable future. They had to determine how many computers might exist in the future, then make the IP address space even bigger. But how many computers would exist in the future? Keep in mind that TCP/IP development took place back in the early 1970s. There were less than 1000 computers in the entire world at the time, but that didn't keep the IP framers from thinking big! They decided to go absolutely crazy (as many people considered at the time) and around 1979 created the **Internet Protocol version 4 (IPv4)** 32-bit IP address space, creating up to about 4 billion IP addresses. That should hold us for the foreseeable future!

It hasn't. First, the TCP/IP folks wasted huge chunks of IP addresses due to classful addressing and a generally easygoing, wasteful method of parceling out IP addresses. Second, the Internet reached a level of popularity way beyond the original framers' imagination. By the mid-1980s the rate of consumption for IP addresses started to worry the Internet people and the writing was on the wall for IPv4's 32-bit addressing. As a result, the Internet Engineering Task Force (IETF) developed a new IP addressing scheme, called **Internet Protocol version 6 (IPv6)**, that is slowly replacing IPv4. IPv6 extends the 32-bit IP address space to 128 bits, allowing up to  $2^{128}$  (that's close to  $3.4 \times 10^{38}$ ) addresses! That should hold us for the foreseeable future!  $3.4 \times 10^{38}$  addresses is something like all the grains of sand on Earth or 1/8 of all the molecules in the atmosphere.

But IPv6 isn't just about expanding the IP address space. IPv6 also improves security by making the Internet Protocol Security (IPSec) protocol support a standard part of every IPv6 stack. That doesn't mean you actually have to use IPSec, just that manufacturers must support it. If you use IPSec, every packet sent from your system is encrypted, opening the possibility that IPv6 would eliminate most (but not all) of the many encryption methods currently in use today.

IPv6 also provides a more efficient routing scheme. Taking advantage of aggregation (see the section "Aggregation" later in this chapter), routing tables should shrink dramatically, enabling fast routing.

It's taking a while, but IPv6 is finally gaining traction. You must learn and use IPv6, both for the CompTIA Network+ exam and for the real world. This chapter breaks the process into three parts for you. First, you need the basic concepts, such as how the numbers work. Second, you need to learn how to enable or apply IPv6 in a variety of technologies, such as NAT and DHCP. Finally, you need answers on how to deploy IPv6 in an IPv4 world.



If you really want to know how many IP addresses IPv6 provides, here's your number: 340,282,366,920,938,463,463,374,607,431,768,211,456.

## ■ IPv6 Basics

Although they achieve the same function—enabling computers on IP networks to send packets to each other—IPv6 and IPv4 differ a lot when it comes to implementation. The addressing numbers work differently, for example, and don't look alike. IPv6 uses link-local addressing, a concept not present in IPv4. Subnetting works differently as well. You also need to understand the concepts of global addresses and aggregation, both topics uniquely IPv6. Let's look at all five topics.

## Test Specific

### IPv6 Address Notation

The 32-bit IPv4 addresses are written as 197.169.94.82, using four octets. Well, IPv6 has 128 bits, so octets are gone. IPv6 addresses are written like this:

```
2001:0000:0000:3210:0800:200C:00CF:1234
```



For those who don't play with hex regularly, one hexadecimal character (for example, *F*) represents 4 bits, so four hexadecimal characters make a 16-bit group.

IPv6 uses a colon as a separator, instead of the period used in IPv4's dotted decimal format. Each group is a hexadecimal number between 0000 and FFFF.

A complete IPv6 address always has eight groups of four hexadecimal characters. If this sounds like you're going to type in really long IP addresses, don't worry, IPv6 offers a number of shortcuts.

First, leading zeroes can be dropped from any group, so 00CF becomes CF and 0000 becomes 0. Let's rewrite that IPv6 address using this shortcut:

```
2001:0:0:3210:800:200C:CF:1234
```

To write IPv6 addresses containing strings of zeroes, you can use a pair of colons (::) to represent a string of consecutive groups with a value of zero. For example, using the :: rule you can write the IPv6 address:

```
2001:0:0:3210:800:200C:CF:1234
```

as

```
2001::3210:800:200C:CF:1234
```

Double colons are very handy but you have to be careful when you use them. Take a look at this IPv6 address:

```
FEDC:0000:0000:0000:00CF:0000:BA98:1234
```

If I convert it to

```
FEDC::CF:0:BA98:1234
```

I may not use a second :: to represent the third-to-last group of four zeroes—only one :: is allowed per address! There's a good reason for this rule. If more than one :: was used, how could you tell how many sets of zeroes were in each group? Answer: you couldn't.

Here's an example of a very special IPv6 address that takes full advantage of the double colon, the IPv6 loopback address:

```
::1
```

Without using the double-colon nomenclature, the IPv6 address would look like this:

```
0000:0000:0000:0000:0000:0000:0000:0001
```



The unspecified address (all zeroes) can never be used, and neither can an address that contains all ones (all *F*s in IPv6 notation).



## Cross Check

### Loopback

You learned about the IPv4 loopback address in Chapter 7, "TCP/IP Basics," so check your memory as you read about the IPv6 loopback address here. What IP address or addresses could you use for a loopback address? When might you PING the loopback address? How would this differ from loopback testing discussed in Chapter 6, "Working with Physical Networks"?

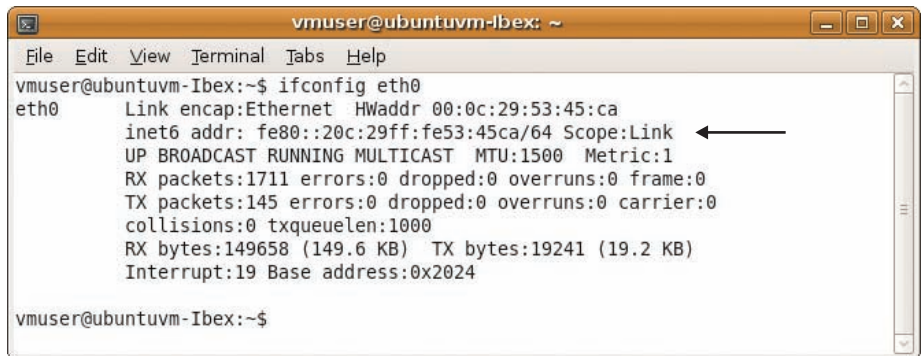
IPv6 still uses subnets but you won't find a place to type in 255s anywhere. IPv6 uses the “/X” Classless Inter-Domain Routing (CIDR) nomenclature. Here's how to write an IP address and subnet for a typical IPv6 host:

FEDC::CF:0:BA98:1234/64

## Link-Local Address

The folks who created IPv6 worked hard to make it powerful and easy to use, but you pretty much have to forget all the rules you learned about IPv4 addressing. The biggest item to wrap your mind around is that you no longer have a single IP address unless your network isn't connected to a router. When a computer running IPv6 first boots up, it gives itself a **link-local address**. Think of a link-local address as IPv6's equivalent to IPv4's APIPA address. The first 64-bits of a link-local address is always FE80::/64. That means every address always begins with FE80:0000:0000:0000. If your operating system supports IPv6 and IPv6 is enabled, you can see this address. Figure 13.1 is a screenshot of Linux's IFCONFIG command, showing the link-local address.

Take a close look at the IP address and compare it to the MAC address. The last 64 bits of the IPv6 address, collectively called the **Extended Unique Identifier, 64-bit (EUI-64)**, are taken from the MAC address. MAC addresses only contain 48 bits, so your system creates the EUI-64 as follows:



```
vmuser@ubuntuvm-Ibex:~$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0c:29:53:45:ca
          inet6 addr: fe80::20c:29ff:fe53:45ca/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1711 errors:0 dropped:0 overruns:0 frame:0
          TX packets:145 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:149658 (149.6 KB) TX bytes:19241 (19.2 KB)
          Interrupt:19 Base address:0x2024

vmuser@ubuntuvm-Ibex:~$
```

• **Figure 13.1** Link-local address

1. Remove the dashes from the MAC address and split it in half:

000C29    5345CA

2. Add “FFFE” in the middle:

000C29FFFE5345CA

3. This step requires a little binary knowledge. You convert the second hexadecimal digit, in this example the second 0, into binary: 0 in hex = 0000 in binary. You take the third binary digit and complement it, which means that if it's a 0, as in this example, you make it a 1, and if it's a 1, you make it a 0. Then convert it back to hexadecimal: 0010 = 2 in hexadecimal.

4. Put that value back in the second position:

020C29FFFE5345CA

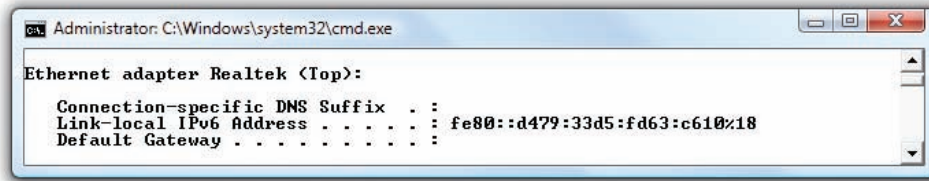
5. Break it into standard IPv6 format:

020C:29FF:FE53:45CA



You can reconfigure Vista to use EUI-64 for link-local addresses by typing this at a command prompt:

```
netsh interface ipv6 set
global
randomizeidentifiers=
disabled
```



• **Figure 13.2** Link-local address in Windows Vista

6. Add it to the first 64 bits and drop leading zeroes:

FE80::20C:29FF:FE53:45CA

Don't worry about exactly how to make your own EUI-64: just understand how your system comes up with this value. Every operating system, with the exception of Windows Vista and Windows 7, creates link-local addresses using EUI-64 by default. Microsoft adds a lot of extra steps in Vista but the big difference is that the last 64 bits of the link-local addresses are generated randomly. I think Microsoft does this as a privacy issue. If someone's link-local address ties directly to a real MAC address, in theory some

bad guy might use this against them. No other operating system, not even Windows XP or Windows Server, does this randomized link-local numbering (Figure 13.2).

The link-local address does all the hard work in IPv6 and, as long as you don't need an Internet connection, it's all you need.

The old concepts of static and DHCP addressing don't really make much sense in IPv6 unless you have dedicated servers (even in IPv6 servers usually still have static IP addresses). Link-local addressing takes care of all your local network needs!

## IPv6 Subnet Masks

IPv6 subnets function the same as IPv4 subnets in that systems use them to determine whether to ARP for a MAC address or send packets to a default gateway. But there are two new rules you need to know:

- The last 64 bits of an IPv6 address are generated by the NIC, leaving a maximum of 64 bits for the subnet. Therefore, no subnet is ever longer than /64.
- The IANA passes out /32 subnets to big ISPs and end users who need large allotments. ISPs and others may pass out /48 and /64 subnets to end users. 3. Therefore, the vast majority of IPv6 subnets are between /48 and /64.

You will never type in a subnet mask. With link-local addressing the subnet mask is defined as /64. Other types of IPv6 addresses get the subnet information automatically from their routers (described next).

## The End of Broadcast

A system's IPv6 link-local address is a **unicast address**, a unique address that is exclusive to that system. IPv4 also relies on unicast addresses. But IPv6 completely drops the idea of broadcast addressing and replaces it with the idea of *multicast*. An IPv6 **multicast address** is a set of reserved addresses designed to go



only to certain systems. For example, if a system sends out a multicast to the address FF02::2, only routers read the message, while everyone else ignores it (Figure 13.3).

Technically, a multicast is a broadcast in that every computer gets the packet, but while every computer reads an IPv4 broadcast (like 255.255.255.255), only the computers defined by the IPv6 address read an IPv6 multicast. The difference is subtle but important. Table 13.1 shows some of the more useful IPv6 multicast addresses. You’ve just seen FF02::2; you’ll see the rest explained later in this chapter.

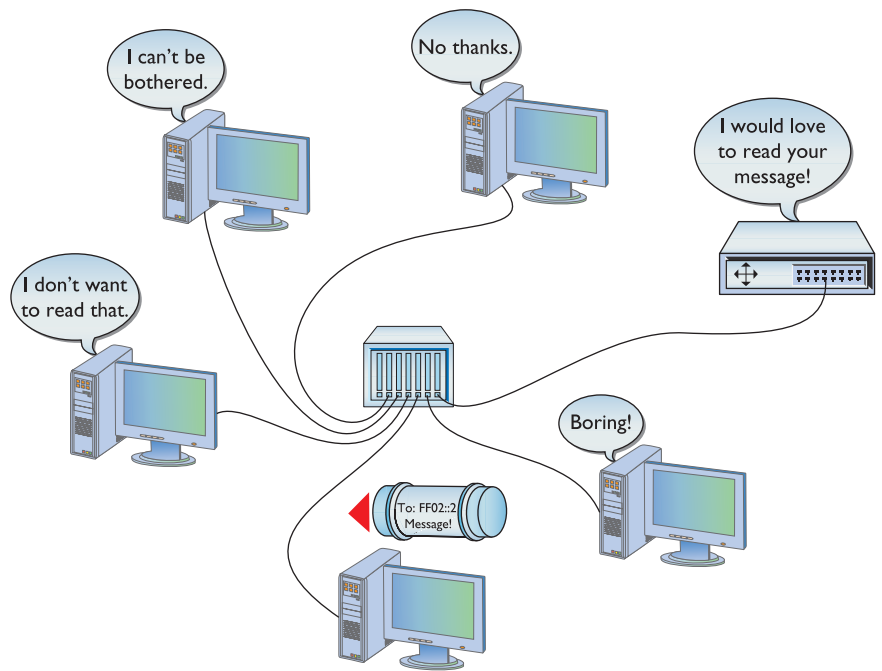
Looking at the first listing, FF02::1, you might ask: “How is that different from a broadcast?” The answer lies more in the definition of multicast than in what really takes place. A computer must be a member of a particular group to read a particular multicast. In this case if a computer is a member of “All Nodes” then it will read the message. Granted, it looks and acts exactly like an IPv4 broadcast but it is a multicast.

Beyond unicast and multicast, IPv6 uses a third type of addressing called **anycast**. An anycast address is a bit of a strange animal, so it’s very helpful to see why you need an anycast address before you try to understand what one is. The best place to look to understand how anycast works and why it is needed is the one place where its use is very common: DNS.

You learned in Chapter 10, “Network Naming,” that the top of the DNS root structure consists of a number of root DNS servers. Every DNS server on the Internet keeps the IP addresses of the root servers in a file called root hints. Here’s one part of the root hints file from my own DNS server:

```
.                NS      F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.  A      192.5.5.241
F.ROOT-SERVERS.NET.  AAAA   2001:500:2f::f
```

At first glance, you might think that this root server is a single physical box because it only has a single IPv4 address and a single IPv6 address. It’s not. It is roughly 20 groups of server clusters strategically placed all over the world. Back in Chapter 12 you saw how DNS can make a cluster of



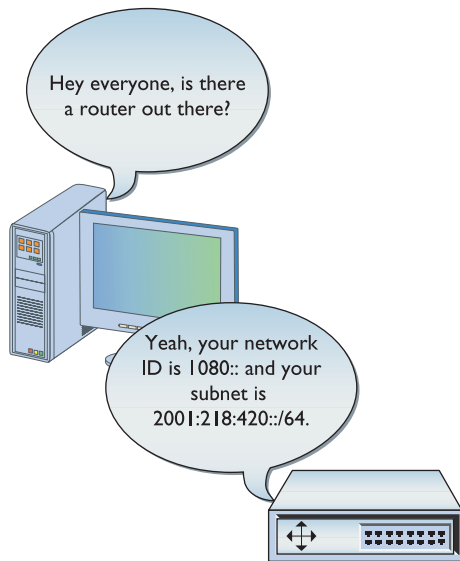
• Figure 13.3 Multicast to routers

Table 13.1 IPv6 Multicast Addresses	
Address	Function
FF02::1	All Nodes Address
FF02::2	All Routers Address
FF02::1:FFXX:XXXX	Solicited-Node Address

computers (by *cluster* I mean the computers are all in the same building, behind a common box of some sort) act as a single server, but none of those solutions can make a bunch of clusters all over the world act as a single server. To do this we need anycasting.

Anycasting starts by giving a number of computers (or clusters of computers) the same IP address. We then give routers (in the case of DNS only the biggest, tier-one Internet routers) the smarts to determine which of the many computers with the same IP address are closest. When that router gets a packet addressed to that IP address, it sends it only to the closest root DNS server, even though it may know where others are located. That is an anycast address.

An anycast address looks like a unicast address and, in most cases, the computer sending the packet doesn't know or care to know that the address is anycast and not unicast. The only device that knows (and cares) is the top-tier router that has the smarts to send the packet only to the closest root DNS server.



• **Figure 13.4** Getting a global address

## Global Address

To get on the Internet, your system needs a second IPv6 address called a **global unicast address**, usually shortened to “global address.” The only way to get a global address is from your default gateway router, which must be configured to pass out global IPv6 addresses. When your computer boots up, it sends out a router solicitation message on multicast address FF02::2 looking for a router. Your router hears this message and tells your computer your network ID and subnet (together called the prefix). See Figure 13.4.

Once you have your prefix, your computer generates the rest of the global address. It uses the MAC address to create the last 64 bits just like it does to create the last 64 bits of a link-local address. You now have a legitimate, public IP address as well as your link-local address. Figure 13.5 shows the IPv6 information on a Macintosh running OS X 10.5.

Let's look at this process in detail with an example:

1. An IPv6-capable computer boots up. As it boots it sends out a router solicitation message (FF02::2).
2. An IPv6-configured router hears the request and then sends the prefix to the computer. In this example let's say it is 2001:470:b8f9:1/64.
3. The computer takes the prefix and adds the EUI-64 address to the end of the prefix. If the MAC address is 00-0C-29-53-45-CA, then the EUI-64 address is 20C:29FF:FE53:45CA.
4. Putting the prefix with the EUI-64 address, you get the global address: 2001:470:b8f9:1: 20C:29FF:FE53:45CA.

A global address is a true Internet address. If another computer is running IPv6 and also has a global address, it can access your system unless you have some form of firewall.



At the moment, IANA only passes out global addresses that begin with the number 2 (for example, 2001::, 2002::, and so on). As demand increases this will certainly change, but for now it sure makes it easy to know a global address when you see one.



Computers using IPv6 need a global address to access the Internet.

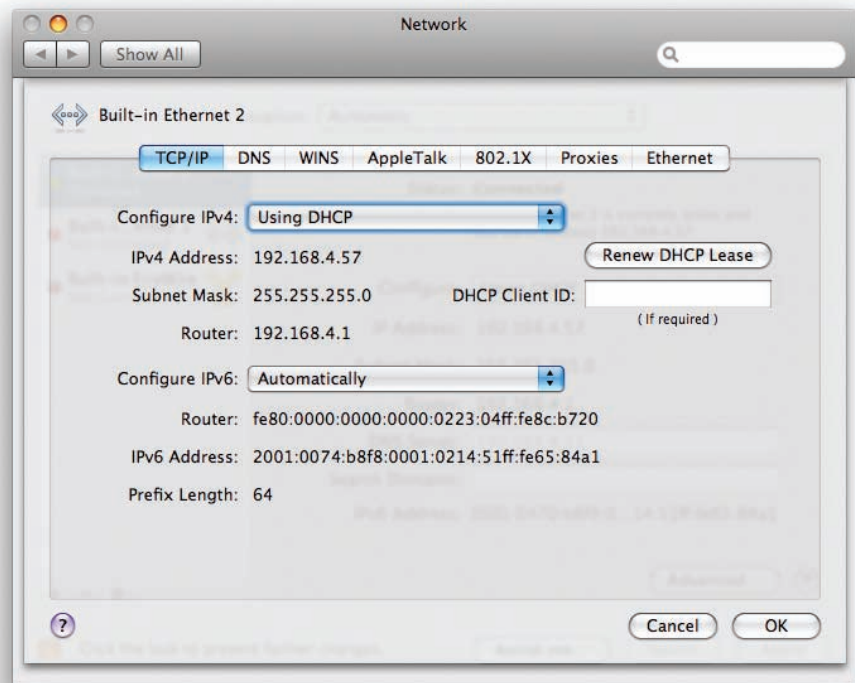
## Aggregation

Routers need to know where to send every packet they encounter. Most routers have a default path on which they send packets that aren't specifically defined to go on any other route. As you get to the top of the Internet, the tier-one routers that connect to the other tier-one routers can't have any default route (Figure 13.6). We call these the no-default routers.

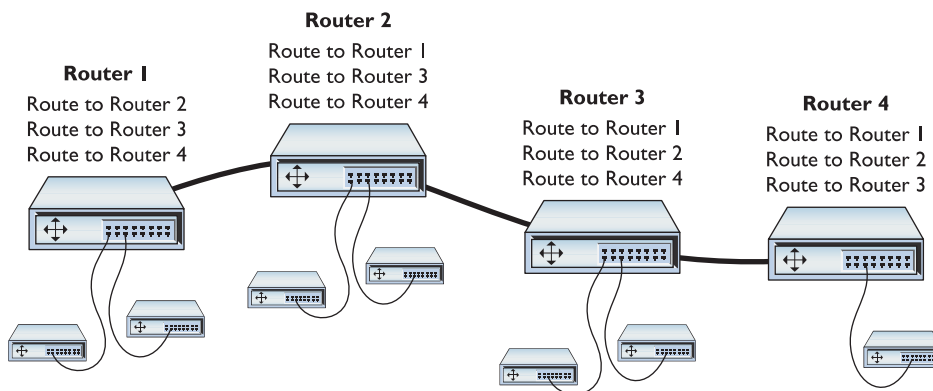
The current state of the upper tiers of the Internet is rather messy. A typical nondefault router has somewhere around 30,000 to 50,000 routes in its routing table, requiring a router with massive firepower. But what would happen if the Internet was organized as shown in Figure 13.7? Note how every router underneath one router always uses a subnet of that router's existing routes. This is called **aggregation**.

Aggregation would drastically reduce the size and complexity of routing tables and make the Internet faster. Aggregation would also give a more detailed, geographic picture of how the Internet is organized—you could get a strong idea of where a person is physically located just by looking at their IP address.

It's way too late for IPv4 to use aggregation. Many organizations who received class licenses 20 to 30 years ago simply will not relinquish them, and

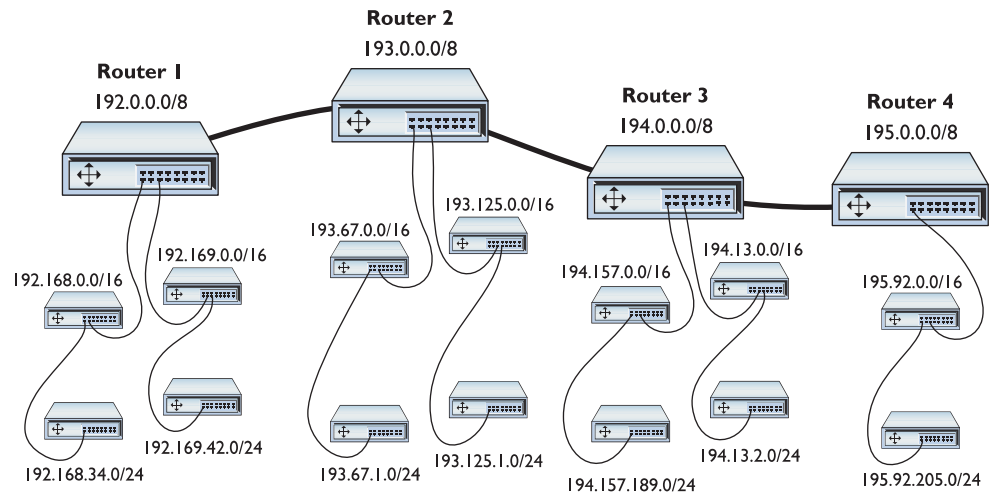


• Figure 13.5 IPv6 configuration on Macintosh OS X



• Figure 13.6 No-default routers





• **Figure 13.7** Aggregation



A 48-bit prefix from upstream router + 16-bit subnet from gateway router + 64-bit unique number = 128-bit IPv6 address.

the amount of work necessary to make aggregation work would require a level of synchronization that would bring the entire Internet to its knees for days if not weeks.

But aggregation is part and parcel with IPv6. Remember that your computer gets the first 64 bits of its Internet address from your gateway router. The router in turn gets a (usually) 48-bit prefix from its upstream router and adds its own 16-bit subnet.

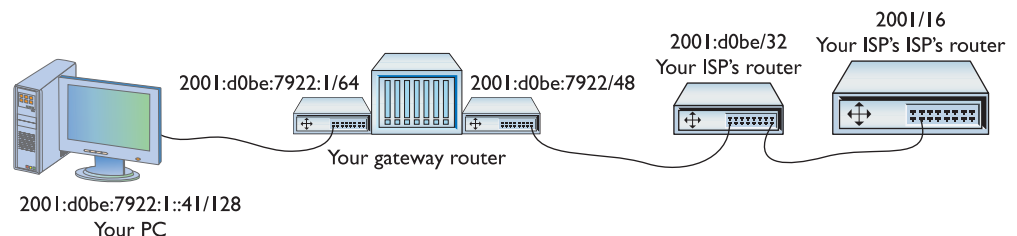
This method allows the entire IPv6 network to change IP addresses on-the-fly to keep aggregation working. Imagine you have your gateway router connected to an upstream router from your ISP as shown in Figure 13.8.

Your PC's IPv6 address is: 2001:d0be:7922:1:fc2d:aeb2:99d2:e2b4. Let's cut out the last 64 bits and look at the prefix and see where this comes from:

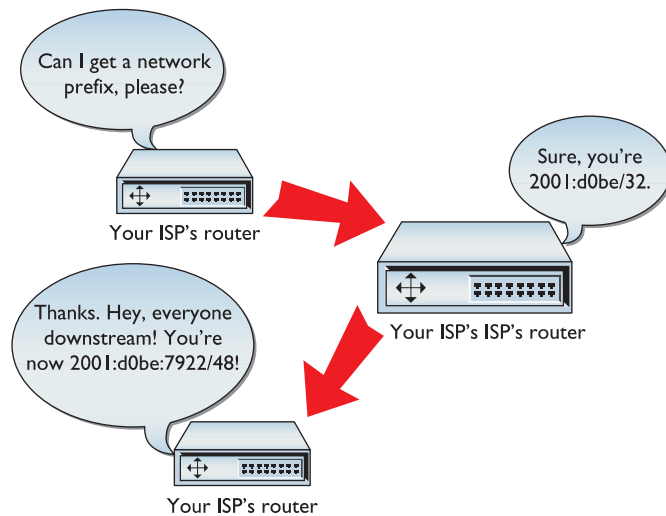
Your network's prefix: 2001:d0be:7922:1/64

IPv6 addresses begin at the very top of the Internet from the no-default servers. We'll assume your ISP's ISP is one of those routers. Your ISP gets (usually) a 32-bit prefix from IANA or from its ISP if it is small.

In this case the prefix is 2001:d0be/32. This prefix comes from the upstream router, and your ISP has no control over it. However, the person setting up the ISP's router will add a 16-bit subnet to the prefix as shown in Figure 13.9.



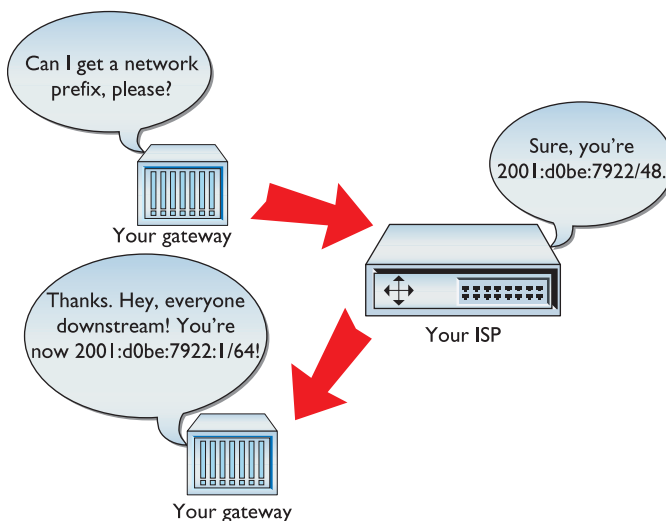
• **Figure 13.8** An IPv6 group of routers



• **Figure 13.9** Adding the first prefix

Your gateway router receives a 48-bit prefix (in this case 2001:d0be:7922/48) from your ISP's router. Your gateway router has no control over that prefix. However, the person setting up your gateway router adds your own 16-bit subnet (in this case :0001 or :1) to the 48-bit prefix to make the 64-bit prefix for you network (Figure 13.10).

What makes all this particularly interesting is that any router upstream of anyone else may change the prefix they send downstream, keeping aggregation intact. To see this in action, let's watch what happens if your ISP decides to change to another upstream ISP (Figure 13.11). In this case, your ISP moves from the old ISP (ISP1) to a new ISP (ISP2). When your ISP makes the new connection, the new ISP passes out a different 32-bit prefix (in this example 2AB0:3C05/32). As quickly as this change takes place, all of the



• **Figure 13.10** Adding the second prefix

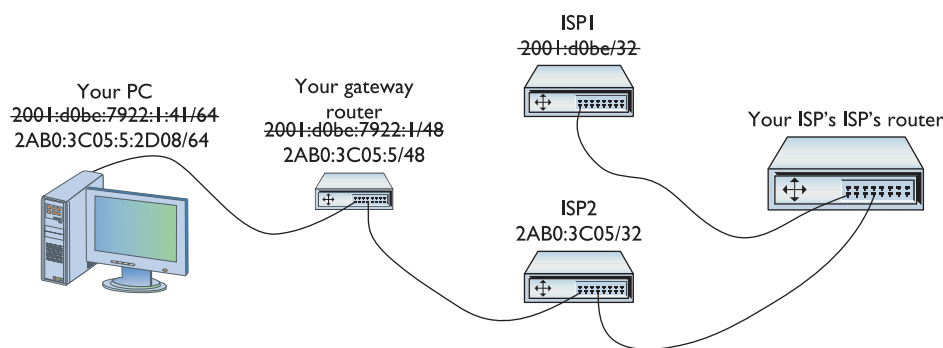


## Tech Tip

### Regional Internet Registries

The IANA doesn't actually pass out IPv6 prefixes. This job is delegated to the five Regional Internet Registries (RIRs):

- American Registry for Internet Numbers (ARIN) supports North America.
- RIPE Network Coordination Centre (RIPE NCC) supports Europe, the Middle East, and Central Asia.
- Asia-Pacific Network Information Centre (APNIC) supports Asia and the Pacific region.
- Latin American and Caribbean Internet Addresses Registry (LACNIC) supports Central and South America and parts of the Caribbean.
- African Network Information Centre (AfriNIC) supports Africa.



• **Figure 13.11** New IP address updated downstream

downstream routers make an “all nodes” multicast and all clients get new IP addresses.

Aggregation is an intrinsic but for the most part completely transparent part of IPv6. Know that our IPv6 Internet addresses may suddenly change from time to time and that the address changes are a fairly rare but normal aspect of using IPv6.

## ■ Using IPv6

Once IPv6 fully replaces IPv4 we will find ourselves in a very different world from the one we left in terms of configuration. In this section you will see what it takes to turn on IPv6 for your network. This section also assumes you’ve turned off IPv4—which isn’t a realistic option right now because IPv4 is prevalent, but it makes understanding some aspects of using IPv6 much easier. You’ll also learn how IPv6 works (or doesn’t work, as the case may be) with NAT, DHCP, and DNS. We’ll cover the idea of running IPv6 and IPv4 at the same time in the next section.

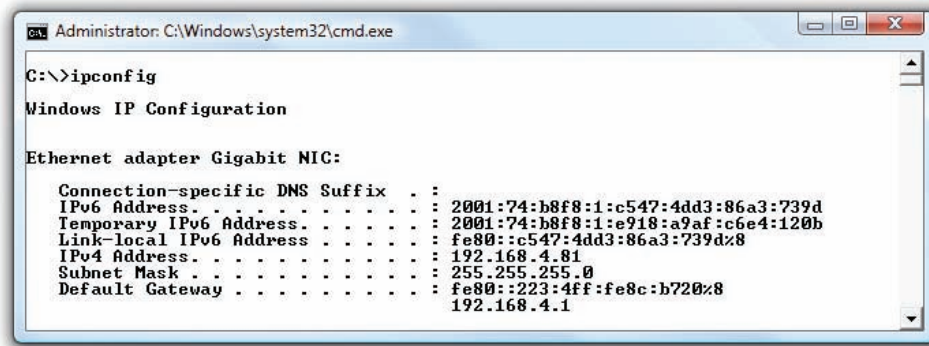
## Enabling IPv6

Enabling IPv6 is very easy because in most cases it is already running on your operating system. Table 13.2 lists the popular operating systems and their IPv6 status.

The fastest way to verify if your system runs IPv6 is to check the IP status for your OS. In Windows, go to a command prompt and type `ipconfig` (Figure 13.12). In Linux or Macintosh OS X, go to a terminal and type `ifconfig` (Figure 13.13). Remember that you will have a link-local address only if your router isn’t configured for IPv6.

**Table 13.2** IPv6 Adoption by OS

Operating System	IPv6 Status
Windows 2000	Windows 2000 came with “developmental” IPv6 support. Microsoft does not recommend using Windows 2000 for IPv6.
Windows XP	Original Windows XP came with a rudimentary but fully functional IPv6 stack that had to be installed from the command prompt. SP1 added the ability to add the same IPv6 stack under the <b>Install   Protocols</b> menu.
Windows Vista/Windows 7	Complete IPv6 support. IPv6 is active on default installs.
Windows Server 2003	Complete IPv6 support. IPv6 is not installed by default but is easily installed via the <b>Install   Protocols</b> menu.
Windows Server 2008	Complete IPv6 support. IPv6 is active on default installs.
Linux	Complete IPv6 support from kernel 2.6. IPv6 is active on most default installs.
Macintosh OS X	Complete IPv6 support on all versions. IPv6 is active on default installs.



• **Figure 13.12** IPv6 enabled in Windows Vista

## NAT in IPv6

The folks pushing IPv6 forward are a vocal group with some pretty strong feelings about how IPv6 should work. If you want to get some really nasty e-mail, just go to one of the many IPv6 sites and ask this question: “How do I set up NAT on IPv6?” I know they will get mad because I asked that question. Here are some of the answers as I saw them:

“NAT was developed a long time ago as a way to avoid running out of IP addresses. It was never meant to be a type of firewall.”

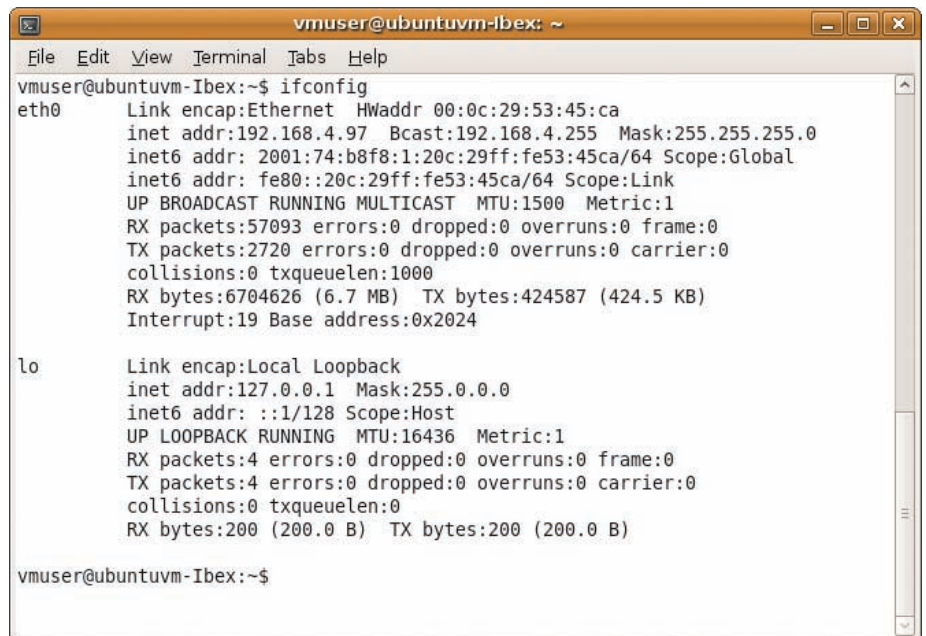
“NAT messes up IPSec.”

“Only jerks use NAT!”

If you’re going to use IPv6, you’re not going to use NAT. That means every one of your IP addresses will be exposed to the Internet, and that’s not good. The answer is: count on IPv6 to make life hard on hackers and use a good firewall.

One big problem with IPv4 is how easy it is to sniff networks. Using tools like Anton Keks’ popular Angry IP Scanner ([www.angryziber.com/w/Home](http://www.angryziber.com/w/Home)), you can scan an entire subnet looking for active IP addresses, as shown in Figure 13.14.

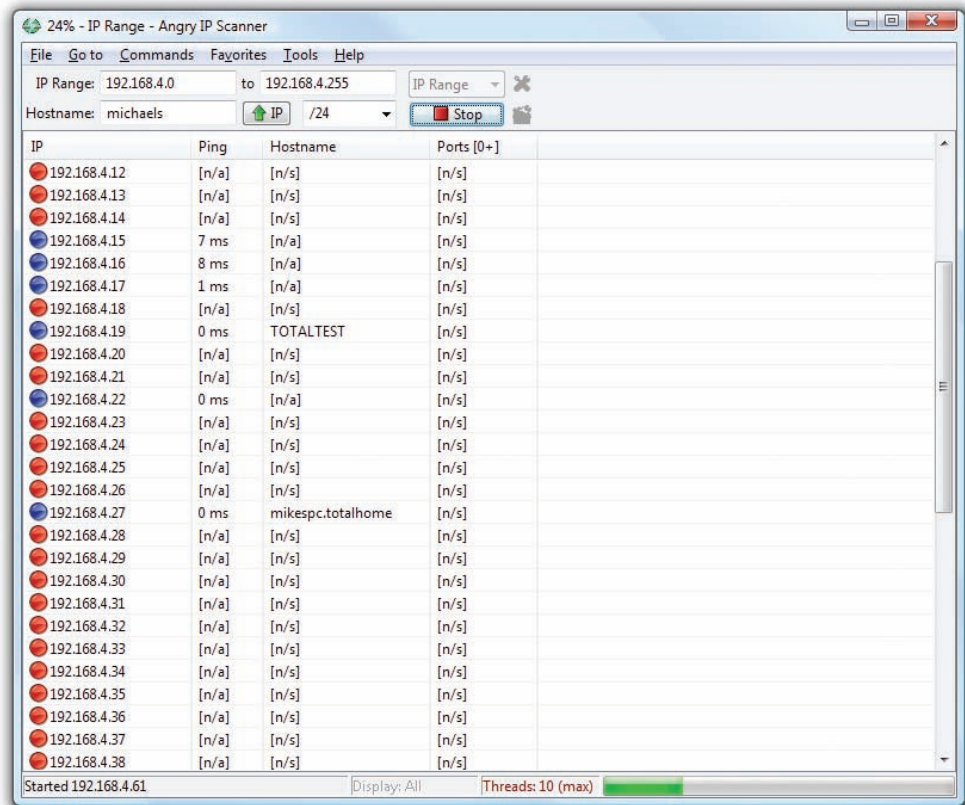
IPv6’s huge address space makes such scanning programs obsolete. Let’s say you knew my subnet was 2001:d0be:7922:1/64. There are  $2^{64}$



• **Figure 13.13** IPv6 enabled in Ubuntu 8.10



There is a version of NAT for IPv6 called NAPT-PT (an earlier version was called NAT-PT). This is a solution to get an IPv6 network to use a single IPv4 address and it does not work well.



• **Figure 13.14** Angry IP Scanner at work

different possible IP addresses on this subnet. Assuming a scanner could check one million addresses/second, it would take something like 580,000 years to check them all.

If a bad guy knows your address, the hope is that you'll be using IPSec to prevent other people from reading your information. Otherwise there are a number of other security options that aren't specific to IPv6 that you can use. Some of these, like encryption, you've seen in earlier chapters. Others, like good firewalling, you'll see in Chapter 17, "Protecting Your Network."

## DHCP in IPv6

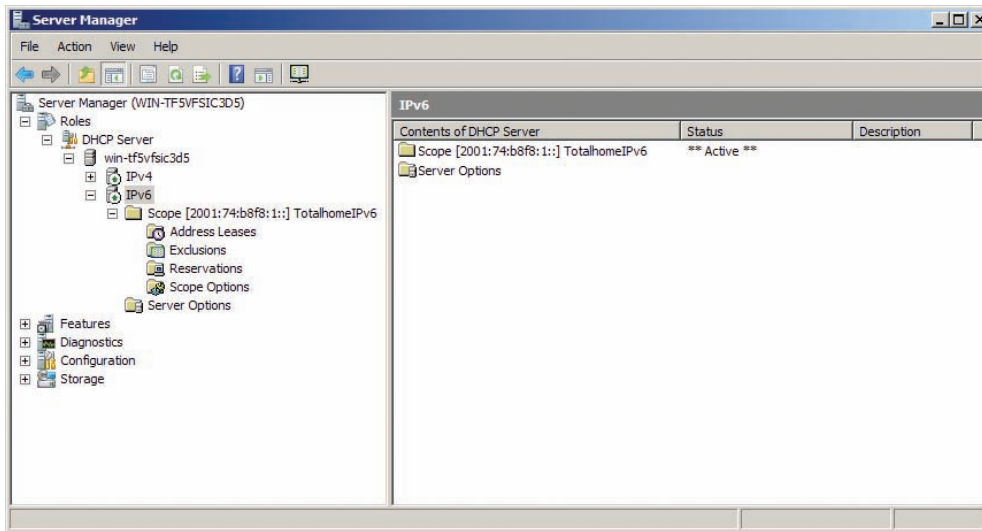
DHCP is alive and well in the IPv6 world but works very differently than IPv4's DHCP. At first glance you'd think you wouldn't need DHCP anymore. IPv6 clients get their IP address and subnet from their gateway router's advertisements (so they also know the default gateway). This is true, but IPv6 router advertisements do not pass out a number of other very important bits of information that clients need, such as DNS server information, giving DHCP a very important place in IPv6.

A fully functional DHCPv6 server works in one of two modes: stateless or stateful. A **stateful** DHCPv6 server works very similarly to an IPv4 DHCP



We call IPv6's DHCP servers DHCPv6 to separate them from IPv4 DHCP servers.





• **Figure 13.15** DHCPv6 server in action

server, passing out IPv6 addresses, subnet masks, and default gateways as well as optional items like DNS server addresses. A **stateless** DHCPv6 server only passes out optional information. Figure 13.15 shows the DHCPv6 server on Windows Server 2008.

Most IPv6 installations should take advantage of IPv6's autoconfiguration and only run stateless DHCPv6 servers. But if you really want to skip aggregation you may certainly go stateful. Be careful going stateful: as long as you're using an Intranet or your upstream router knows what to do with your non-aggregated network ID, you're okay. Stateful DHCPv6 might be helpful for internal networks that do strange things like try to use subnets greater than /64, but for the most part expect stateless to be the norm.

### Cross Check

#### DHCP with IPv4

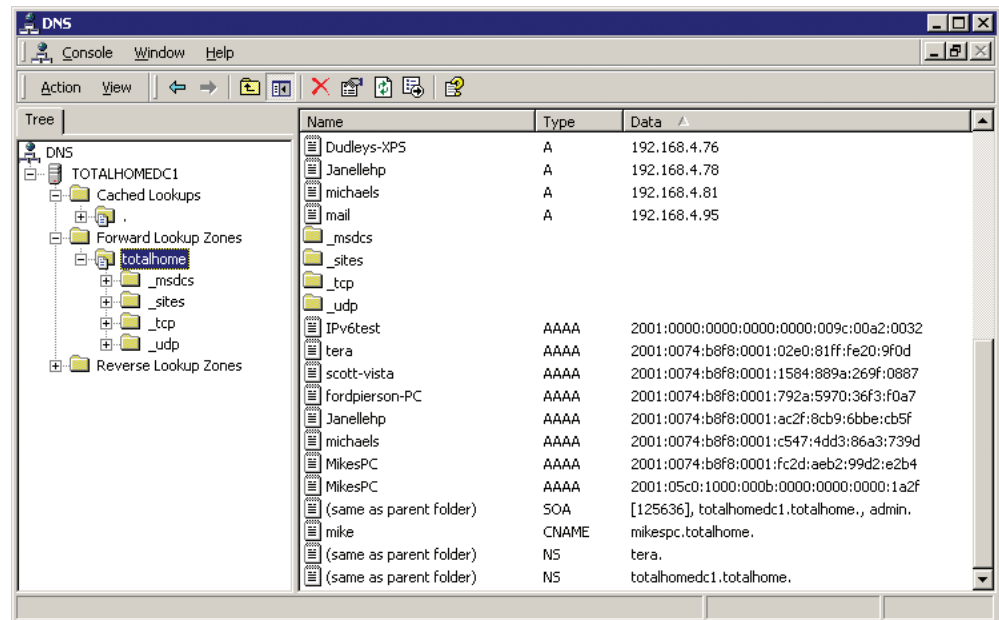
You read about the IPv4 version of DHCP way back in Chapter 7, "TCP/IP Basics," so check your memory now. How does DHCP work? What does a DHCP lease do for you? What happens if your computer can't get to a DHCP server but is configured for DHCP?

There's a push to get DNS server information added to IPv6 router advertisements. If this happens the need for DHCPv6 might fall dramatically.

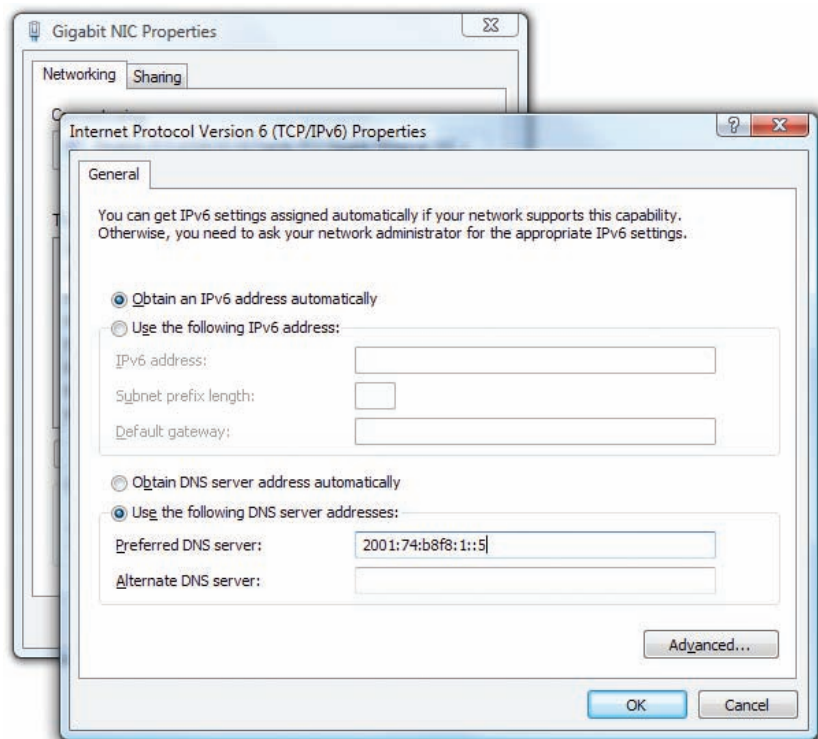
## DNS in IPv6

DNS with IPv6 is trivial. Just about every DNS server made in the last five to six years or so supports IPv6 addresses. All IPv6 addresses use an AAAA nomenclature. Figure 13.16 shows some IPv6 addresses in Windows Server 2000.

A common trick to get around using DHCPv6 is to manually add DNS server information to IPv6 clients. You do this exactly the same as you do with IPv4, as shown in Figure 13.17. This isn't the best long-term solution but, until all the IPv6 DNS details are worked out, it works well.



• **Figure 13.16** IPv6 addresses on DNS server



• **Figure 13.17** Manually adding an IPv6 DNS server in Vista

## ■ Moving to IPv6

There's no reason for you *not* to try running IPv6 today—like right now! At the very least there's a whole world of IPv6-only Web sites for you to explore. At the most you may very well become the IPv6 expert in your organization. You almost certainly have an operating system ready to do IPv6; the only trick is to get you connected to the rest of us fun-loving IPv6-capable folks.

This section is designed to help you get connected. If you can, grab an IPv6-capable system, fire up IPv6 as shown earlier, and make sure you're connected to the Internet, because we are going someplace you've never been before, the IPv6 Internet.

### IPv4 and IPv6

The first and most important point to make right now is that you can run both IPv4 and IPv6 on your computers and routers at the same time, just as my computer does, as shown in Figure 13.18. This ability is a critical part of the process enabling the world to slowly migrate from IPv4 to IPv6.

Almost all operating systems support IPv6 and almost all serious routers support IPv6 but very few of the small home routers support IPv6. Plus not all routers on the Internet have their IPv6 support turned on.

In order for IPv6 to work we need every router and every computer on the Internet to support IPv6, and we are not yet there. However, two critical parts of the Internet are ready:

- All of the root DNS servers now support IPv6 resolution.
- Almost all of the tier-one ISP routers properly forward IPv6 packets.

The problem is that the routers and DNS servers between your IPv6-capable computer and the other IPv6-capable computers to which you would like to connect are not yet IPv6-ready. How do we get past this IPv6 gap (Figure 13.19)?

### Tunnels

To get on the IPv6 network you need to leap over this gap. The folks who developed IPv6 have a number of ways for you to do this using one of many IPv4-to-IPv6 tunneling standards. An IPv4-to-IPv6 tunnel works like any other tunnel, encapsulating one type of data into another. In this case, you will be encapsulating your IPv6 traffic into an IPv4 tunnel to get to an IPv6-capable router, as shown in Figure 13.20.

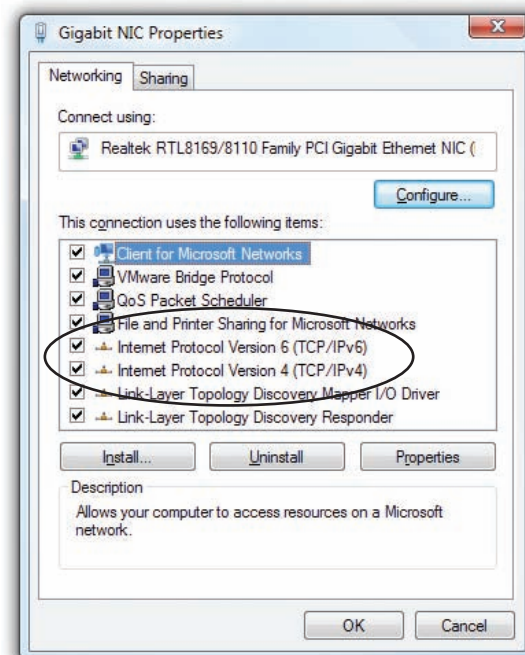
To make this tunnel, you are going to download a tunneling client and install it on your computer. You will then fire up the client and make the tunnel connection—it's very easy to do. Before we create this tunnel, you need to appreciate that this is only one way to make an IPv6 connection—I'll show you other ways in a moment. There are four popular IPv4-to-IPv6 tunneling standards, described next.



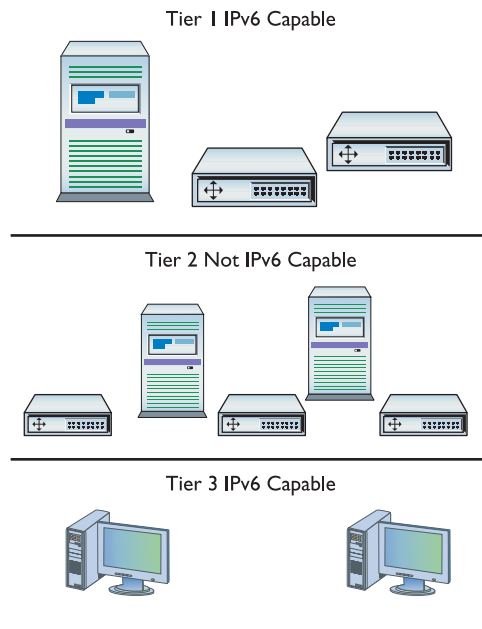
#### Tech Tip

##### IPv6 Security

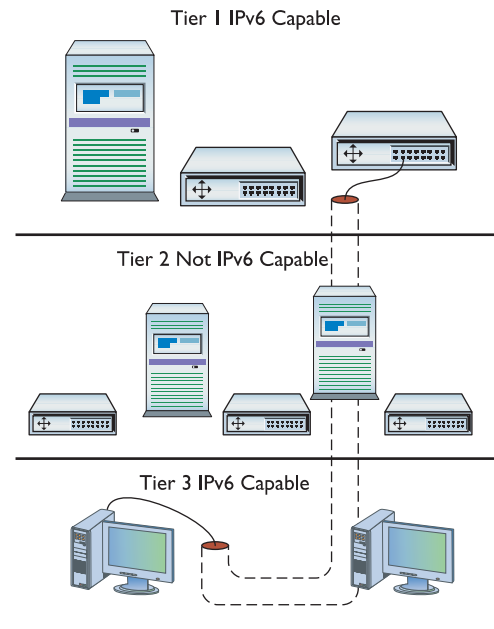
*IPv6 is just now gaining wide support so there are issues in connecting to the IPv6 world. There are potential security risks as well as less than perfect IPv6 support with operating systems. Don't connect to the IPv6 Internet on a critical computer.*



• Figure 13.18 IPv4 and IPv6 on one computer



• Figure 13.19 The IPv6 gap



• Figure 13.20 The IPv4-to-IPv6 tunnel

## 6to4

**6to4** is the dominant tunneling protocol because it is the only IPv6 tunnel that doesn't require a tunnel broker (see "Tunnel Brokers" below). It is usually used to directly connect two routers because it normally requires a public IPv4 address. 6to4 addresses always start with 2002::/16. If you have an IPv6-capable router, or if you have a computer directly connected to the Internet, you can set up a 6to4 tunnel. 6to4 uses public relay routers all around the world. Search the Web for "**public 6to4 relays**" to find one close to you. One IPv4 address, 192.88.99.1, is called the *6to4 anycast address* and works everywhere.

Setting up a 6to4 tunnel can be more challenging than setting up the tunnels that use tunnel brokers. If you're feeling adventurous, just do a Web search on "**6to4 setup**" and the name of your operating system. There are hundreds of Web sites to show you how to set up a 6to4 tunnel.

## 6in4

**6in4** (also called IPv6-in-IPv4) is one of the most popular of all the IPv6 tunneling standards and the one we'll be using in our tunneling example. 6in4 is one of only two IPv6 tunneling protocols that can go through a NAT (called NAT traversal).

## Teredo

**Teredo** is the second NAT-traversal IPv6 tunneling protocol. Teredo is built into Microsoft Windows and as a result sees some adoption. Most people prefer to skip Windows built-in support and instead get a third-party tool that supports 6to4 or 6in4. Teredo addresses start with 2001:0000::/32. If you want to get on the IPv6 Internet very quickly, just try the following Try This!

## ISATAP

**Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)** is designed to work within an IPv4 network by actually adding the IPv4 address to an IPv6 prefix to create a rather interesting but non-standard address for the endpoints. One example of an ISATAP address is 2001:DB8::98CA:200:131.107.28.9. ISATAP has a strong following but other tunneling standards are gaining ground because they use a more common IPv6 addressing structure.

## Tunnel Brokers

Setting up an IPv6 tunnel can be a chore. You have to find someone willing to act as the far endpoint, you have to connect to them somehow, and then you have to know the tunneling standard they use. To make life easier, those who provide the endpoints have created the idea of the **tunnel broker**. Tunnel brokers create the actual tunnel and (usually) offer a custom-made endpoint client for you to use, although more advanced users can often make a manual connection. Many tunnel brokers take advantage of one of two automatic configuration protocols, called **Tunnel Setup Protocol (TSP)** and **Tunnel Information Control protocol (TIC)**. These protocols set up the tunnel and handle configuration as well as login. If it wasn't for TSP and TIC, there would be no such thing as automatic third-party tunnel endpoint clients for you to use. Here's a short list of the most popular IPv6 tunnel brokers around the world. For a more complete list, go to [www.sixxs.net/tools/aiccu/brokers](http://www.sixxs.net/tools/aiccu/brokers).

Tunnel Broker	URL
Hexago/Freenet/Go6	<a href="http://www.go6.net">www.go6.net</a>
SixXS	<a href="http://www.sixxs.net">www.sixxs.net</a>
Hurricane Electric (no TSP/TIC)	<a href="http://www.tunnelbroker.net">www.tunnelbroker.net</a>
AARNet	<a href="http://broker.aarnet.net.au">http://broker.aarnet.net.au</a>



## Try This!

### Using Teredo

If you're using Windows XP (with Service Pack 1 or later) or Windows Vista you have nothing to lose but your chains, so try this! You can use Teredo to access the IPv6 Internet as long as you have access to the Internet normally and your computer is not part of a Windows domain. This can be done on a domain, but the process gets a little ugly in my opinion. Beware! Some home routers can't handle Teredo and many high-end routers are specifically designed to prevent this traffic (it's a great way to get around many network defenses) so if it doesn't work, blame the router. Here are the steps in Windows Vista:

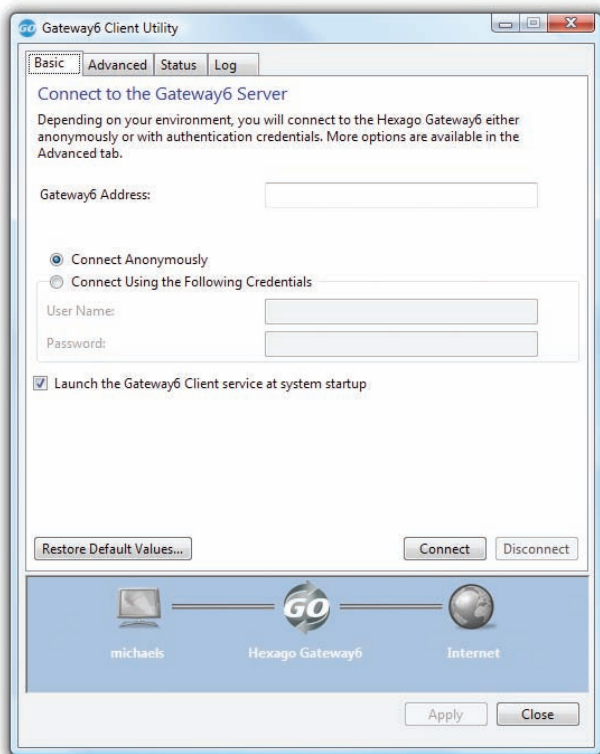
1. Make sure the Windows Firewall is enabled. If you have a third-party firewall, turn it off.
2. Go to **Start** and type **cmd** in the Start Search box, but don't press ENTER yet. Instead, right-click the command prompt option above and select **Run as administrator**.
3. From the command prompt, type these commands, followed by ENTER each time:

```
netsh
interface
teredo
set state client
exit
```
4. Test by typing `ipconfig /all`. You should see an adapter called "Tunnel adapter Teredo tunneling pseudo-interface" (or something close to that) with an IP address starting with 2001.
5. Then type `ping ipv6.google.com` to make sure you can reach the Internet.
6. Open a Web browser and go to an IPv6 Web site, like [www.sixxs.com](http://www.sixxs.com) or [ipv6.google.com](http://ipv6.google.com).
7. Remember, Microsoft loves to change things. If this isn't working, search for Teredo on the Microsoft Web site.



You rarely have a choice of tunneling protocol. The tunneling protocol you use is the one your tunnel broker provides and is usually invisible to you.





• Figure 13.21 Gateway6 Client Utility



• Figure 13.22 Gateway6 Client Utility Status tab

## Setting Up a Tunnel

Every tunnel broker has its own setup, so you should always read the instructions carefully. In this example we will install the Hexago client (called the Go6 Gateway6 client) onto Windows Vista. Go to [www.go6.net](http://www.go6.net) and register for an account (there is currently a “Join go6” link on the page). You’ll then be led to a download page where you then download the Gateway6 client. Go6 is always updating this client so be sure to download the latest version. Install the client to see the screen shown in Figure 13.21. Enter the Gateway6 address and your username and password. You can log on anonymously but I think it works more reliably if you log in.

Click **Connect** and, assuming you have a good connection, you should be on the IPv6 Internet. Go to the **Status** tab to see your IP information (Figure 13.22).

Excellent! Now let’s go check out the Internet. Try these IPv6-only Web pages:

- [www.ipv6.sixxs.net](http://www.ipv6.sixxs.net) (click the **Enter website** hyperlink to see your IPv6 address on the bottom left of the page)
- <http://ipv6.google.com>
- <http://ipv6.sunny.ch> (shows your IPv6 address)

## IPv6 Is Here, Really!

Depending on who you speak to, IPv6 is going to happen, and when it does happen it’s going to happen very quickly. Again depending on who you speak to, IPv4 addresses will run out anywhere between 2011 and 2019. The people who know IPv6 will be in a unique place to leap into the insanity of what will invariably be called “The Big Switchover” and will find themselves making a lot of money. Take some time to learn IPv6. You’ll be glad you did.

# Chapter 13 Review

## ■ Chapter Summary

After reading this chapter and completing the exercises, you should understand the following about IPv6.

### Discuss the fundamental concepts of IPv6

- IPv4 supports only about 4 billion addresses, which is no longer enough for the future. IPv6 supports  $2^{128}$  (or  $\sim 3.4 \times 10^{38}$ ) IP addresses.
- In addition to an expanded address space, IPv6 improves security by supporting IPSec out of the box.
- IPv6 provides a more efficient routing scheme because it uses aggregation.
- IPv6 addresses are composed of 128 bits written in hexadecimal notation. Every 4 bits are separated by a colon. 2001:0000:0000:3210:0800:200C:00CF:1234 is a valid IPv6 address.
- Leading zeroes can be dropped and double colons may be used to represent consecutive groups of zeroes in order to write an IPv6 address with fewer characters. 2001::3210:800:200C:CF:1234 is a valid IPv6 address.
- IPv6 subnet masks are represented with the /X CIDR naming convention. FEDC::CF:0:BA98:1234/64 translates to a 64-bit subnet mask.
- Computers using IPv6 that are on the Internet have two IPv6 addresses: a link-local address and a global address.
- A link-local address is similar to an IPv4 APIPA address in that it is self-generated. The link-local address is guaranteed to be unique because it is partially based on the MAC address of the NIC.
- A link-local address always starts with FE80::. The last 64 bits of the link-local address are generated from the NIC's MAC address.
- Microsoft Windows Vista and Microsoft Windows 7 generate the last 64 bits of a link-local address randomly so as to not reveal the MAC address. This adds security against hackers.
- An IPv6 computer not on the Internet needs only the self-generated link-local address to participate on a local network. However, a server on such a network still needs a static IP address, not a random self-generated link-local address.
- Link-local addresses always use /64 as the subnet mask.
- IPv6 link-local addresses are unicast, or unique to a specific computer or network node.
- IPv6 replaced broadcasts with multicasts. A multicast is a set of reserved addresses designed to go to only certain systems. Packets sent to addresses beginning with FF02::2 are only sent to routers.
- Multicasts, like broadcasts, are still sent to every computer on the network. Unlike broadcasts, though, only the destined systems read the multicast packet.
- An IPv6 global unicast address is required for Internet access.
- Global unicast addresses are distributed by the default gateway router, provided the router is configured to pass out global IPv6 addresses.
- The first half of a global unicast address is called the prefix and consists of the network ID and subnet mask. The prefix is passed out by the default gateway router. The last half of the global address is self-generated by the computer.
- Aggregation reduces the size and complexity of routing tables by allowing downstream routers to use a subset of an upstream router's routes to populate its routing table rather than tens of thousands of disjointed routes.

### Describe IPv6 practices

- Not all versions of Windows support IPv6, and some that do enable it by default whereas others require manual installation.
- IPv6 is active by default on Macintosh OS X, and active by default on most Linux installs using at least kernel 2.6.
- IPv6 does not support NAT. Every IPv6 address is exposed to the Internet, so use a good firewall!

- While IPv6 global addresses are passed out by the default gateway router (with a portion self-generated), DHCP servers are still important because they pass out DNS server information.
- Stateful DHCPv6 servers pass out IPv6 addresses, subnet masks, default gateway addresses, and DNS server addresses, as well as other, optional information.
- Stateless DHCPv6 servers pass out only optional information. Stateless DHCPv6 servers are preferred to stateful servers because stateless servers support aggregation.
- DHCPv6 servers may be bypassed by manually entering DNS server information into the IP settings of an IPv6 client.
- An IPv4-to-IPv6 tunnel can be used to bridge the gap created by non-IPv6 routers, allowing you access to the root and tier-one routers that do support IPv6.
- There are four popular tunneling standards: 6to4, 6in4, Teredo, and ISATAP.
- 6to4 is the dominant tunneling protocol and is the only one that doesn't require a tunnel broker. However, it is the most challenging to set up. 6to4 addresses start with 2002:/16.
- Only 6in4 and Teredo can go through NAT.
- Teredo is built into Microsoft Windows. Teredo addresses always start with 2001:0000:/32.
- ISATAP adds an IPv4 address to an IPv6 prefix. For example, 2001:DB8:98CA:200:131.107.28.9.

### Implement IPv6 in a TCP/IP network

- Do not connect to the IPv6 Internet on a critical computer! Limited IPv6 support means potential security risks.
- Currently, all root DNS servers support IPv6 resolution and almost all tier-one ISP routers properly forward IPv6 packets. However, the routers between you and these root and tier-one servers may not support IPv6 at the moment.

- A tunnel broker is a service provider that creates the tunnel, acts as the far endpoint, and often provides a tunneling client for easier setup.
- TSP and TIC are two automatic configuration protocols for setting up IPv4-to-IPv6 tunnels.
- It is estimated that IPv4 addresses will run out sometime between the year 2011 and 2019.

## Key Terms

**6in4** (354)

**6to4** (354)

**aggregation** (345)

**anycast** (343)

**Extended Unique Identifier, 64-bit (EUI-64)** (341)

**global unicast address** (344)

**Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)** (355)

**Internet Protocol version 4 (IPv4)** (338)

**Internet Protocol version 6 (IPv6)** (339)

**link-local address** (341)

**multicast address** (342)

**stateful** (350)

**stateless** (351)

**Teredo** (354)

**tunnel broker** (355)

**Tunnel Information Control protocol (TIC)** (355)

**Tunnel Setup Protocol (TSP)** (355)

**unicast address** (342)

## Key Term Quiz

Use the Key Terms list to complete the sentences that follow. Not all the terms will be used.

1. A(n) \_\_\_\_\_ DHCPv6 server passes out only optional information.
2. The \_\_\_\_\_ tunneling protocol is the only one that doesn't use a tunnel broker.
3. It is the practice of \_\_\_\_\_ that greatly reduces the size of IPv6 routing tables by reducing them to a subnet of an upstream router.
4. You must have a(n) \_\_\_\_\_ to connect to the IPv6 Internet.

5. A(n) \_\_\_\_\_ address contains a total of 32 bits.
6. A packet sent to a(n) \_\_\_\_\_ is broadcast to all network nodes, but only the target nodes read the packet.
7. The MAC address is used to generate the complete \_\_\_\_\_, except in Windows Vista and Windows 7.
8. The \_\_\_\_\_ appends an IPv4 address to the end of the IPv6 prefix.
9. Computers involved in a local network that has no Internet connectivity require only a(n) \_\_\_\_\_.
10. Employing the services of a(n) \_\_\_\_\_ automates the process of setting up an IPv6 tunnel.

## ■ Multiple-Choice Quiz

1. How many bits comprise an IPv6 address?
  - A. 32
  - B. 48
  - C. 64
  - D. 128
2. Which of the following is a valid IPv6 address?
  - A. 192.168.0.1
  - B. 2001:376:BDS:0:3378:BAAF:QR9:223
  - C. 2541:FDC::ACDF:2770:23
  - D. 0000:0000:0000:0000:0000:0000:0000:0000
3. Which of the following IPv6 addresses are equivalent to ACCB:0876:0000:0000:FD87:0000:0000:0064? (Select two.)
  - A. ACCB:876::FD87:0:0:64
  - B. ACCB:876::FD87::64
  - C. ACCB:876:0:0:FD87::64
  - D. ACCB:876:0:FD87:0:64
4. What is the only type of IPv6 address required to communicate with other computers on a local network?
  - A. Link-local
  - B. Global unicast
  - C. EUI-64
  - D. Multicast
5. Which of the following is a valid link-local address?
  - A. 2001:2323:CCE:34FF:19:DE3:2DBA:52
  - B. FE80::1994:33DD:22CE:769B
  - C. FEFE:0:0:0:FEFE:0:0:0
  - D. FFFF:FFFF:FFFF:FFFF:232D:0:DE44:CB2
6. What is true of link-local addresses?
  - A. They are passed out by the default gateway router.
  - B. They are completely randomly generated by each computer.
  - C. The last 64 bits are always generated from the MAC address, except on Windows Vista and Windows 7.
  - D. They always start with 169.254.
7. What is a valid IPv6 subnet mask?
  - A. /64
  - B. /72
  - C. /255
  - D. 255.255.255.0
8. How do IPv6 multicasts differ from broadcasts?
  - A. Broadcasts are sent to all network nodes. Multicasts are sent only to specific network nodes.
  - B. Both broadcasts and multicasts are sent to all network nodes, but only the destination nodes read the incoming packets.
  - C. Broadcasts can cross over a router, whereas multicasts cannot.
  - D. Broadcasts are used on local networks; multicasts are used on the Internet.
9. What type of address applies to a single unique network node?
  - A. Unicast
  - B. Unilateral

- C. Multicast
  - D. Omnicast
10. A packet has been sent to the address FF02:0000:0000:0002:0BCD:23DD:3456:0001. What will read the sent packet?
- A. The single computer with the address FF02:0000:0000:0002:0BCD:23DD:3456:0001.
  - B. Every network node.
  - C. Every router on the network.
  - D. Nothing will read the packet because it is an invalid address.
11. What must your computer have to access the IPv6 Internet?
- A. An IPv4 address
  - B. A global multicast address
  - C. A link-local address
  - D. A global unicast address
12. What is true of current global addresses?
- A. They always begin with 2001::, 2002::, 2003::, and so on.
  - B. They always begin with FF02::1, FF02::2, FF03::3, and so on.
  - C. They are only 64 bits long.
  - D. They are only used by root and tier-one routers.
13. What is the main benefit of IPv6 aggregation?
- A. It allows users to combine multiple IPv6 addresses to increase their bandwidth and overall Internet speed exponentially.
  - B. It is backward-compatible and can be directly applied to IPv4 networks.
  - C. It reduces the size and complexity of routing tables, allowing routers to work more efficiently.
  - D. Signals are increased with each router the packet travels through, allowing for greater distances over wireless networks.
14. Which operating systems fully support IPv6? (Select three.)
- A. Windows 2000
  - B. Windows XP
  - C. Windows Vista
  - D. Macintosh OS X
15. As IPv6 clients can get a portion of their IP address from the default gateway server, what purpose does a DHCPv6 server serve?
- A. DHCPv6 servers can still distribute DNS server information.
  - B. DHCPv6 servers provide link-local addresses.
  - C. DHCPv6 servers provide the other half of the IPv6 address.
  - D. There is no such thing as a DHCPv6 server.

## ■ Essay Quiz

---

1. Explain to a colleague the difference between link-local and global IPv6 addresses. Be sure to include when each one is necessary.
2. Explain how aggregation reduces the size and complexity of routing tables.
3. NAT is not supported in IPv6, meaning that every computer with a global IPv6 address is exposed to the Internet. Why is this not a big concern?



## Lab Projects

### • Lab Project 13.1

Any decent network tech can work effectively with binary and hexadecimal notation. Get some practice

00-14-22-46-8A-77

BC-23-44-AB-A7-21

12-00-CF-C2-44-1A

Now use IPCONFIG /ALL to find your own MAC address and calculate the EUI-64.

by taking the following MAC addresses and calculating the EUI-64:

### • Lab Project 13.2

Choose one of the IPv4-to-IPv6 tunneling methods (6to4, 6in4, Teredo, or ISATAP) and configure a lab computer to connect to the Internet using IPv6.

Write down which method you used and the steps you took to get it working. Make your steps clear so

that someone else can follow them. Swap your steps with a classmate who used a different tunneling method and follow his or her steps to connect using the alternate method. Which method did you find more difficult? Why?