chapter 2

How Computers Find Each Other on Networks

After reading this chapter and completing the exercises, you will be able to:

- Describe how computers and other devices are addressed on a network
- Explain how host names and domain names work
- Identify how ports and sockets work at the OSI Transport layer
- Demonstrate how IP addresses are assigned and formatted at the OSI Network layer
- Use command-line tools to troubleshoot problems with network addresses

On the Job

While I was working as a junior project manager in the Technology Solutions Department for a large corporation, I was assigned to work on a network infrastructure project. At the time, I had no training as a network engineer, and was instead responsible for small- to medium-sized technology projects as they related to a business unit that spanned five states. For this new project, our goal was to change the network's topology in a way that would allow the network to grow over time for the least amount of money, and to keep the network up to date with the latest trends within the industry.

As with most projects, a budget was set at the beginning. This budget allowed us to hire a professional vendor to complete the wiring and cabling installations. The network engineers who worked for the vendor were experts on everything related to wiring and cabling. However, before they could get very far, our budget was aggressively reduced. Suddenly, we could no longer afford the cabling experts. Instead, senior managers decided that work would be completed by our company's own junior IT technicians, people who were better suited to printer paper jam resolution than recabling an entire network. They knew nothing about hierarchical cable structure, maximum cable distances, or endpoint terminations.

This ignorance of basic networking standards had dire consequences on our project's budget and timeline. But the problem wasn't just that the IT people doing the work lacked the proper knowledge. As the project manager, with no systematic knowledge of networking standards, I was also hampered in my ability to keep things on track.

Part of a successful project manager's job is recognizing the need for subject matter experts, or at least being able to understand where to find key pieces of information related to the project and then interpreting that information as it relates to the project. In my case, a simple understanding of a set of telecommunications standards, or TIA/EIA-568, would have been indispensable in completing the network topology change project.

Our in-house team began the project on a vacant floor that was to become new employee office space. We unknowingly exceeded cable runs, terminated wall outlet connection points incorrectly, and generally did a poor installation job. Only after new client computers were installed and exhibited a variety of connection issues did we realize our installation was most likely the culprit. We soon understood that our lack of prior planning and our ignorance of industry standards were to blame. Through painful trial and error, we gained an in-depth knowledge of telecommunications structured cabling and the tools needed to implement a network topology change, but with the cost of this knowledge was a lot of time on a ladder removing ceiling tiles and working late into the night to ensure clients were able to effectively run their applications at the start of the next workday.

Tom Johnson
Segment Account Manager, Defense Industry

In Chapter 1, you learned that the OSI model can be used to describe just about every aspect of networking. You saw firsthand the usefulness of working your way up or down the seven layers of the OSI model to troubleshoot networking problems. In this chapter, you learn the several methods used to address and find software, computers, and other devices on a network. We'll take a top-down approach to the OSI model as we explore these topics, starting at the Application layer and working our way down to the Data Link layer. (The lowest OSI layer, the Physical layer, does not require a network address.) At the end of this chapter, you learn how to troubleshoot addressing problems by using an OSI top-down or bottom-up approach to solving the problem.

In Chapter 3, you'll take your networking skills to the next level by learning how data is transported over a network. Now let's begin this chapter with an overview of the several ways software and devices are addressed on a network.

Overview of Addressing on Networks

Network+ 1.3 1.8 2.6 4.2

In Chapter 1, you learned that addressing methods operate at the Application, Transport, Network, and Data Link layers of the OSI model so that one host or node can find another on a network.

The organization responsible for tracking the assignments of port numbers, domain names, and IP addresses is the **Internet Assigned Numbers Authority (IANA)** (pronounced "I–anna"). IANA is a department of the **Internet Corporation for Assigned Names and Numbers (ICANN)**. ICANN is a nonprofit organization charged with setting many policies that guide how the Internet works. For more information, see iana.org and icann.org. At icann.org, you can download helpful white papers that explain how the Internet works.

Here's a quick overview of the four addressing methods, starting at the top of the OSI model:

- *Application layer FQDNs, computer names, and host names*—Every host on a network is assigned a unique character-based name called the **fully qualified host name** or the **fully qualified domain name (FQDN)**, for example, john.mycompany.com, ftp.mycompany.com, and www.mycompany.com. Collectively, the last two parts of a host name (for example, mycompany.com) are called the **domain name**, which matches the name of the organization's domain or network. The first part (for example, john, ftp, and www) is the **host name**, which identifies the individual computer on the network. Ftp is the host name usually given to an FTP server, and www is typically the host name assigned to a computer running a Web server. The FQDN is sometimes called the **computer name** and, more loosely, it is simply called the host name.

When a techie refers to a host name, you can assume she's actually referring to the FQDN unless stated otherwise.

- *Transport layer port numbers*—Recall that a port number identifies one application among several applications that might be running on a host and is used by the Transport layer to find an application. For example, a Web server application is usually configured to listen for incoming requests at port 80.
- *Network layer IP address*—An IP address is assigned to every **interface**, which is a network connection made by a node or host on a network. The IP address can be used to find hosts on any computer on the globe if the IP address is public on the Internet. Two types of IP addresses are used on the Internet:
  - *IPv4*—**Internet Protocol version 4 (IPv4)** addresses have 32 bits and are written as four decimal numbers called **octets**, for example, 92.106.50.200.
  - *IPv6*—**Internet Protocol version 6 (IPv6)** addresses have 128 bits and are written as eight blocks of hexadecimal numbers, for example 2001:0DB8:0B80:0000:0000:00D3:9C5A:00CC.
- *Data Link layer MAC address*—The MAC address, also called the physical address, is embedded on every NIC on the globe and is assumed to be unique to that NIC. Nodes on a LAN find each other using their MAC addresses. However, MAC addresses are not used to find nodes on networks other than the local network.

A **hexadecimal number** (also called a **hex number**) is a number written in the base 16 number system, which uses the 16 numerals 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. The CompTIA Network+ exam expects you to be familiar with the hex number system. To learn how this number system works and how to convert hex numbers to other number systems, see Appendix B.

IP addresses may be stored, retrieved, and tracked in an application, such as when you enter an IP address into your browser. But for routing purposes, an IP address is used only at the Network layer.

Network+ 1.8

MAC Addresses

You can find a network adapter's MAC address (physical address) by examining the NIC. It will be stamped directly onto the NIC's circuit board or on a sticker attached to some part of the NIC, as shown in Figure 2-1. Later in this chapter, you'll learn to use TCP/IP utilities to report the MAC address.

Traditional MAC addresses contain two parts, are 48 bits long, and are written as hexadecimal numbers separated by colons—for example, 00:60:8C:00:54:99. The first 24 bits (six hex characters, such as 00:60:8C in our example) are known as the **OUI (Organizationally Unique Identifier)** or **block ID** or **company-ID**, and identifies the NIC's manufacturer. A manufacturer's OUI is assigned by the Institute of Electrical and Electronics Engineers (IEEE). If you know a computer's MAC address, you can determine which company manufactured its NIC by looking up its block ID. The IEEE maintains a database of block IDs and their manufacturers, which is accessible via the Web. At the time of this writing, the database search page could be found at *http://standards.ieee.org/regauth/oui/index.shtml*.

Courtesy of D-Link North America

Figure 2-1 NIC with MAC address

Links to Web sites given in this book might become outdated as Web sites change. If a given link doesn't work, try a Google search on the item to find the new link.

The last 24 bits make up the **extension identifier** or **device ID** and identify the device. Manufacturers assign each NIC a unique extension identifier, based on the NIC's model and manufacture date, so that no two NICs share the same MAC address.

Network+ 1.3 1.8 2.6 4.2

Applying Concepts

Explore Addresses on Your Computer

You can permanently assign a **static IP address** to a computer or device, or you can configure the computer or device to receive (or lease) a **dynamic IP address** from a **DHCP (Dynamic Host Configuration Protocol)** server each time it connects to the network and requests an IP address. For Windows 7, follow these steps to configure these TCP/IP settings:

1. In Control Panel, open the **Network and Sharing Center**. Then click **Change adapter settings**. Right-click the network connection and click **Properties**.
2. Using the TCP/IPv4 properties box (see Figure 2-2), you can select **Obtain an IP address automatically** for dynamic IP addressing to be assigned by a DHCP server, or you can manually assign a static IP address, subnet mask, and default gateway. Notice you can also configure TCP/IP to obtain DNS server addresses from the DHCP server, or you can manually assign DNS server addresses.

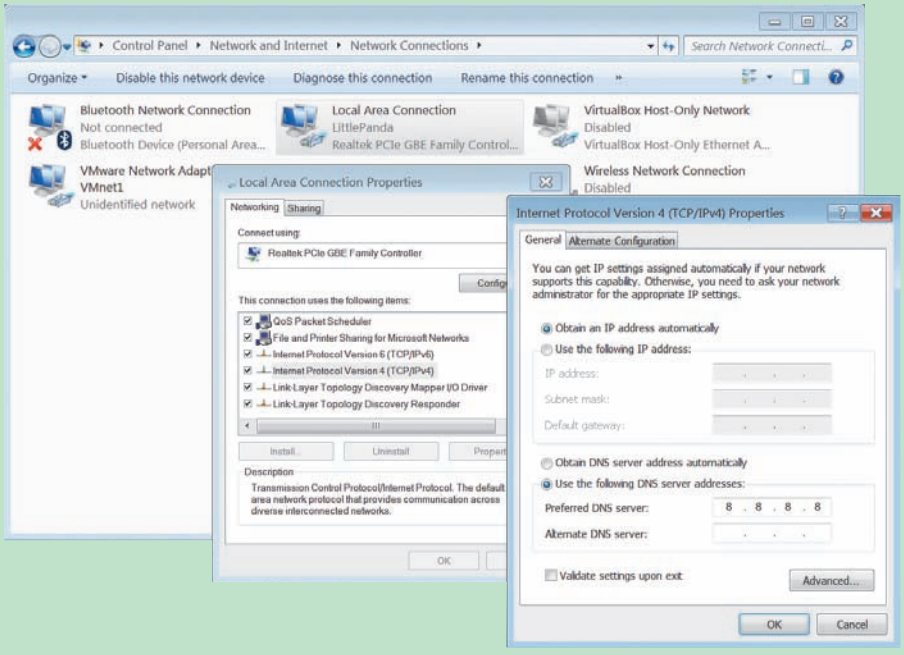


Figure 2-2 Configure TCP/IP for a network interface by using static or dynamic IP addressing

Here's a brief explanation of these settings:

- A **gateway** is a computer, router, or other device that a host uses to access another network. The **default gateway** is the gateway device that nodes on the network turn to first for access to the outside world.
- A **subnet mask** is a 32-bit number that helps one computer find another. The 32 bits are used to indicate what portion of an IP address is the network portion and what part is the host portion. Using this information, a computer can know if a remote computer with a given IP address is on its own or a different network.
- **DNS servers** are responsible for tracking computer names and their IP addresses. When you enter a computer name, such as *www.cengage.com*, in your browser address box, a DNS server is needed to find the IP address of that host.

After the connection to the network is made, you can use the **ipconfig** utility in a Command Prompt window to find out the current TCP/IP settings.

In Windows, commands can be entered at a command-line interface (CLI) that does not provide the Windows graphics normally provided by the graphical user interface (GUI). Network technicians need to be comfortable with the CLI because it is quicker and often more powerful and flexible than a GUI. In Windows, the CLI is provided by a Command Prompt window.

To open a regular Command Prompt window in Windows 7, click Start, type cmd in the Search programs and files box, and press Enter. To open a Command Prompt window with administrative privileges (called an elevated command prompt window), click Start, type cmd, right-click cmd.exe, and click Run as administrator.

To open a regular Command Prompt window in Windows 8.1, right-click Start and click Command Prompt in the Quick Link menu. To open an elevated Command Prompt window, click Command Prompt (Admin) in the Quick Link menu.

Here are two ways to use the `ipconfig` command. You'll learn more about this command later in this chapter:

1. Open a Command Prompt window and enter the **`ipconfig`** command to view IP configuration information (see Figure 2-3). Which Local Area Connections are available on your computer? Which ones are currently connected? Also locate your connection's IPv4 or IPv6 address, subnet mask, and default gateway.

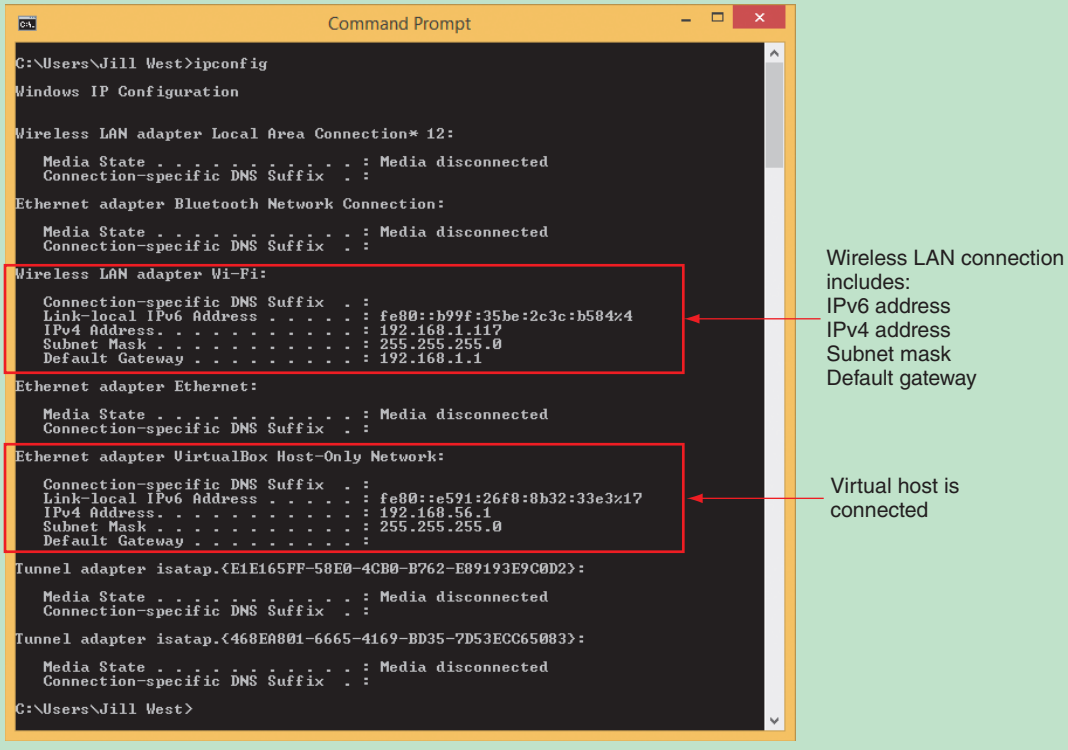


Figure 2-3 This computer is connected to two different network interfaces, one of which is a virtual network inside VirtualBox

Source: Microsoft LLC

2. The `ipconfig` command shows an abbreviated summary of configuration information. To see a more complete summary, use the command **`ipconfig /all`**. See Figure 2-4 for an example.

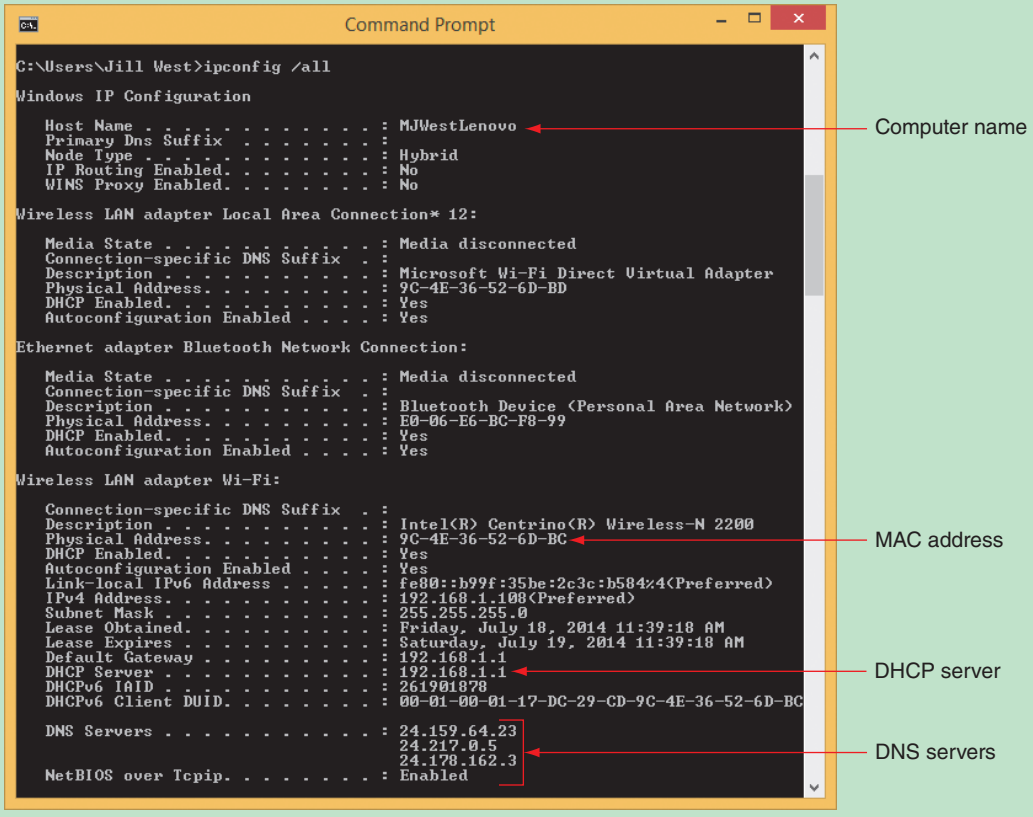


Figure 2-4 `ipconfig /all` gives a great deal more information than `ipconfig` by itself

Source: Microsoft LLC

Now that you have the big picture of how addressing happens at each layer of the OSI model, let's dig into the details of how it all works, beginning with host names and domain names at the top of the model.

How Host Names and Domain Names Work

Network+ 1.3

7 APPLICATION
6 PRESENTATION
5 SESSION
4 TRANSPORT
3 NETWORK
2 DATA LINK
1 PHYSICAL

Host names and domain names were created because character-based names are easier to remember than numeric IP addresses. Recall that an FQDN is a host name and a domain name together, such as *www.cengage.com*. The last part of an FQDN (*com* in our example) is called the top-level domain (TLD).

Host names and domain names can include letters, hyphens, and underscores, but no other special characters.

Domain names must be registered with an Internet naming authority that works on behalf of ICANN. Table 2-1 lists some well-known ICANN-approved TLDs. The first eight TLDs listed in this table were established in the mid-1980s. Of these, no restrictions exist on the use of the .com, .org, and .net TLDs, but ICANN does restrict what type of hosts can be associated with the .arpa, .mil, .int, .edu, and .gov TLDs. A complete list of current TLDs can be found at *iana.org/domains/root/db/*.

Table 2-1 Some well-known top-level domains

Domain suffix	Type of organization
ARPA	Reverse lookup domain (special Internet function)
COM	Commercial
EDU	Educational
GOV	Government
ORG	Noncommercial organization (such as a nonprofit agency)
NET	Network (such as an ISP)
INT	International Treaty Organization
MIL	United States military organization
BIZ	Businesses
INFO	Unrestricted use
AERO	Air-transport industry
COOP	Cooperatives

Registries and registrars of domain names are organizations with unique functions. A domain name registry operator, also known as a registry, is an organization or country responsible for one or more TLDs and maintains a database or registry of TLD information. A domain name registrar such as *godaddy.com* is an organization accredited by registries and ICANN to lease domain names to companies or individuals, following the guidelines of the TLD registry operators.

In 2011, ICANN decided to loosen its restrictions on TLD names and allow organizations and countries to apply for a new TLD composed of almost any alphanumeric string, including one that uses characters not found in the English language. Applying for a new TLD costs $185,000, and each application undergoes a rigorous evaluation.

You're now ready to learn about **name resolution**, which is the process of discovering the IP address of a host when you know its fully qualified domain name. Before we study name resolution on the Internet, let's see how name resolution can work on a local network.

Network+ 1.3

Legacy Networking

Hosts Files

The first incarnation of the Internet, which was called ARPANET, had fewer than 1000 hosts. The entire network relied on one ASCII text file called HOSTS.TXT to associate computer names with IP addresses. This file was generically known as a **hosts file** or **host table**. Growth of the Internet soon made this simple arrangement impossible to maintain. However, when using a peer-to-peer network that doesn't have its own DNS server, you may still encounter this older system of using a text file to associate internal host names with their IP addresses on the local network. For UNIX, Linux, and Windows systems, the filename of a hosts file is hosts with no file extension. In UNIX or Linux, the hosts file is stored in the /etc directory, and on a Windows computer, the hosts file is located in the \Windows\System32\drivers\etc folder.

UNIX or Linux filenames and commands are case sensitive: a Hosts file and a hosts file are considered two different files. Windows, on the other hand, is not case sensitive. The Hosts and hosts filenames in command lines refer to the same file.

Also know that UNIX and Linux use the forward slash in paths to filenames and Windows uses the backslash for this purpose.

Figure 2-5 shows an example of a hosts file. Notice that each host (for example, *www.cengage.com*) is matched by one line identifying the host's name and IP address. In addition, a third field, called an **alias**, provides a nickname for the host (for example, Web). A line that begins with a hashtag (pound symbol) is called a comment line. Comments are used to document the contents of a file and are not interpreted by a program accessing the file.

```
# Host database
#
# This file contains the mappings of IP addresses to host names and the
# aliases for each host name. In the presence of the domain name service,
# this file may not be consulted.
#
# Comments (such as these) may be inserted on individual lines or
# following the machine name denoted by a '#' symbol.
#
#
# Address   Host name                          Alias

::1         localhost.cengage.com              localhost
127.0.0.1   localhost.cengage.com              localhost


69.32.133.79      www.cengage.com              Web
69.32.134.163     ftp.cengage.com              FTP
69.32.146.63      gale.cengage.com             Gale
69.32.132.117     poweron.cengage.com          TechSupport
```

Figure 2-5 Sample hosts file

Source: The Linux Foundation

To use a hosts file, suppose a user on the cengage.com domain wants to access the Web site provided by the corporate network. He enters the address *www.cengage.com* in his browser address box, and the OS resolves the name to its IP address by searching for *www.cengage.com* in its hosts file. It then can send the browser's request to the correct IP address on the local network.

To set up a hosts file, a network administrator must store the hosts file in the correct directory on all computers on the network and update as necessary. This method might work for a small organization that has a few servers made available for its users. However, it's not sufficient for large organizations, much less for the Internet. Instead, an automated solution is mandatory.

Web site developers sometimes use hosts files to assign a host name to a new Web site so that the site can be tested on the local network before it's deployed to the Internet.

Network+ 1.3

DNS (Domain Name System)

In the mid-1980s, **DNS (Domain Name System** or **Domain Name Service**) was designed to associate computer names with IP addresses. DNS is an Application layer client-server system of computers and databases made up of these elements:

- *namespace*—The DNS **namespace** is the entire collection of computer names and their associated IP addresses stored in databases on DNS name servers around the globe.
- *name servers*—DNS **name servers**, also called DNS servers, hold these databases, which are organized in a hierarchical structure.
- *resolvers*—A **resolver** is a DNS client that requests information from DNS name servers.

How Name Servers Are Organized DNS name servers are organized in the hierarchical structure shown in Figure 2-6. At the root level, 13 clusters of **root servers** hold information used to locate the top-level domain (TLD) servers. These TLD servers hold information about the **authoritative servers**, which are the authority on computer names and their IP addresses for computers in their domains.

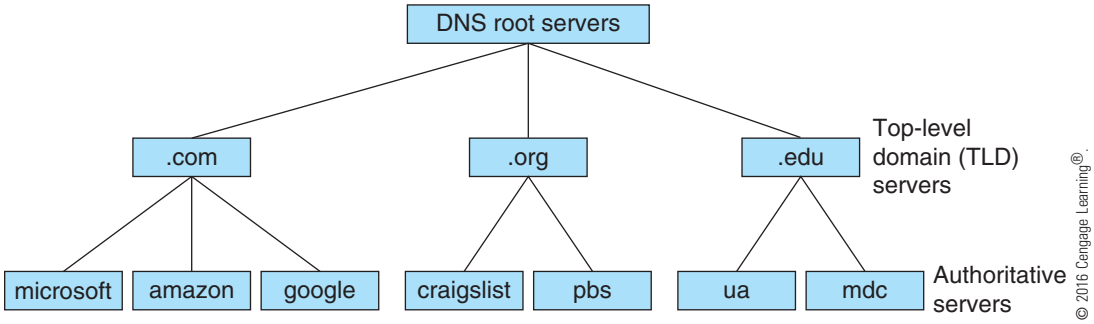


Figure 2-6 Hierarchy of name servers

To understand how the three levels of servers work, let's look at an example. Suppose an employee at Cengage, using a computer in the cengage.com domain, enters `www.mdc.edu` in

her Web browser address box. The browser makes an API call to the DNS resolver, a TCP/IP component in the OS, for the IP address of the *www.mdc.edu* host.

Recall that an application uses an API call to request the operating system perform a service or task.

Here are the steps to resolve the name, which are also illustrated in Figure 2-7:

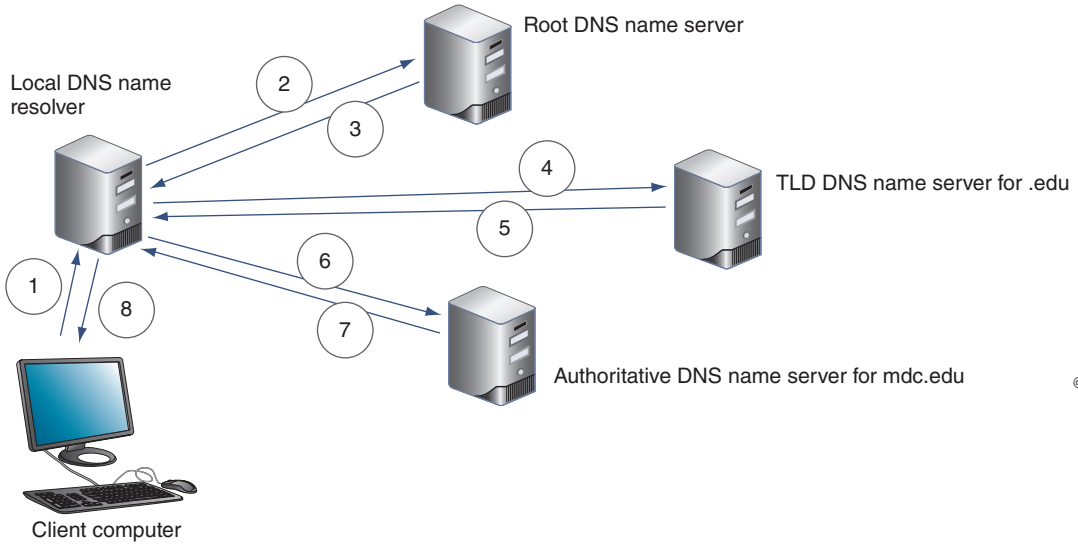


Figure 2-7 Queries for name resolution for *www.mdc.edu*

Step 1—The resolver on the client computer first searches its **DNS cache**, a database stored on the local computer, for the match. If it can't find the information there, the resolver sends a DNS message or query to its local DNS name server. This name server is the authoritative name server for the cengage.com domain. In this example, let's assume it doesn't yet know the IP address of the *www.mdc.edu* host.

DNS messages are Application layer messages that use UDP at the Transport layer. Communication with DNS servers occur on port 53.

Steps 2 and 3—The local name server queries a root server with the request. The root server responds to the local name server with a list of IP addresses of TLD name servers responsible for the .edu suffix.

Steps 4 and 5—The local name server makes the same request to one of the TLD name servers responsible for the .edu suffix. The TLD name server responds with the IP address of the mdc.edu authoritative server.

Steps 6 and 7—The local name server makes the request to the DNS name server at Miami Dade Community College, which responds to the Cengage name server with the IP address of the *www.mdc.edu* host.

Step 8—The local name server responds to the client resolver with the requested IP address. Both the Cengage name server and the Cengage client computer store the information in their DNS caches, and, therefore, don't need to ask again.

Requests sometimes involve additional name servers. Following are a few ways the process can get more complex:

- The local name server might not be an authoritative name server for its organization. Instead, it might exist merely to resolve names for clients, in which case it is called a **caching-only server**. In that situation, when it receives a request for information that is not stored in its DNS cache, it will first query the company's authoritative name server.
- Name servers within a company might not have access to root servers. The local name server might query the name server at the company's Internet service provider (ISP), which might query a name server elsewhere on the Internet that acts as the ISP's naming authority. This name server might query a root server; however, if any name server in the process has the requested information, it responds without the involvement of a root server, TLD name server, or authoritarian name server.
- A TLD name server might be aware of an intermediate name server rather than the authoritative name server. When the local name server queries this intermediate name server, it might respond with the IP address of the authoritative name server.

Recursive and Iterative Queries There are two types of DNS requests: recursive queries and iterative queries. A **recursive query** is a query that demands a resolution or the answer "It can't be found." For example, the initial request the resolver makes to the local server is a recursive query. In other words, the local server must provide the information requested by the resolver, as in "The buck stops here." When the local server issues queries to other servers, these queries are called **iterative queries**, which means the other servers only provide information if they have it; iterative queries do not demand a resolution. Although it's possible for a name server to make a recursive query of another server, generally, it doesn't happen.

DNS Zones and Zone Transfers The records for host names and IP addresses are stored on thousands of servers around the globe, rather than being centralized on a single server or group of servers. In other words, DNS doesn't follow a centralized database model, but rather a **distributed database model**. Because data is distributed over thousands of servers, DNS will not fail catastrophically if one or a handful of servers experience errors.

Each organization that provides host services (for example, Web sites or email) on the public Internet is responsible for providing and maintaining DNS authoritative servers for public access. An organization will have an authoritative name server (called the primary DNS server) and a backup authoritative name server (called the secondary DNS server), and possibly several caching-only name servers. The domains (for example, cengage.com and course.com) that the organization is responsible for managing are called collectively a **DNS zone**. A large organization can keep all its domains in a single zone, or it can subdivide its domains into multiple zones to make each zone easier to manage.

The primary DNS server holds the authoritative DNS database for the organization. When a secondary DNS server needs to update its database, it makes the request to the primary server for the update; the process is called a **zone transfer**. Caching-only DNS servers do not participate in zone transfers, which helps reduce network traffic on slow links in intranets where these servers are often used.

DNS Server Software What software can you run to provide a DNS name server and DNS database? By far, the most popular DNS server software is **BIND (Berkeley Internet Name Domain)**, which is free, open source software that runs on Linux, UNIX, and Windows platforms. **Open source** is the term for software whose code is publicly available for use and modification. You can download the software from *isc.org*. However, most Linux and UNIX distributions include BIND in the distribution.

Many other DNS server software products exist. For example, the Windows Server operating system has a built-in DNS service called Microsoft DNS Server, which partners closely with Active Directory (AD) services. A wise network administrator knows that DNS authoritative records must be accessible to Internet users, but Active Directory must be highly secured. The solution is to use a **split DNS** design, also called a **split-horizon DNS**, in which internal and external DNS queries are handled by different DNS servers or by a single DNS server that is specially configured to keep internal and external DNS zones separate.

In Figure 2-8, you can see two firewalls, one protecting the external DNS server and another one in front of the internal DNS server. A **firewall** is a device, either a router or a computer running special software, that selectively filters or blocks traffic between networks. All firewalls are porous to some degree in that they always let *some* traffic through; the question is what kind of traffic they let through. The external DNS server is behind a more porous firewall, which allows greater exposure to the Internet so that certain permissible traffic gets through. The internal DNS server is better protected behind the second, more hardened firewall, which is stricter about the types of traffic allowed through. The area between the two firewalls is called a **DMZ** or **demilitarized zone**. All DNS requests from the inside network that require external resolution are forwarded to the external DNS server, which also handles incoming queries from the Internet. Internal DNS requests are handled by AD's DNS server, which is kept secure from the Internet.

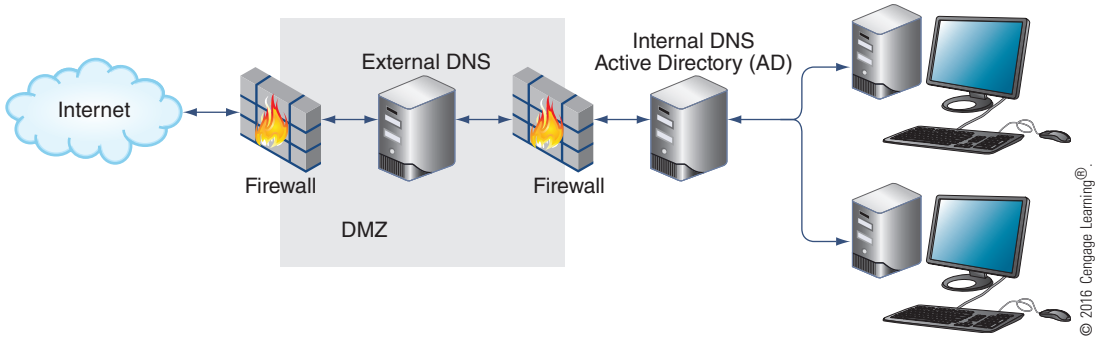
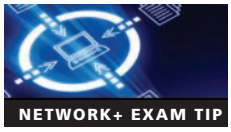


Figure 2-8 DNS services handled by two different servers so that the internal network remains protected

How a Namespace Database Is Organized Now let's see how records in a DNS database are organized. Several types of records, called **resource records**, are kept in a DNS database:

- An **A (Address) record** stores the name-to-address mapping for a host. This resource record provides the primary function of DNS—to match host names to IP addresses, using IPv4 addresses.
- An **AAAA (Address) record** (called a "quad-A record") also holds the name-to-address mapping, but the IP address is an IPv6 type IP address.
- A **CNAME (Canonical Name) record** holds alternative names for a host.

- A **PTR (Pointer) record** is used for reverse lookups, to provide a host name when you know its IP address.
- An **MX (Mail Exchanger) record** identifies a mail server and is used for email traffic.



The CompTIA Network+ exam expects you to know about the five types of DNS resource records listed above.

Each resource record includes a **Time to Live field** that identifies how long the record should be saved in a cache on a server, and this Time to Live is included in zone transfers. Administrators can set the time to live based on how volatile is the DNS data (in other words, how often the administrator expects the IP addresses to change).

Network+ 1.3

Applying Concepts

Configure a DNS Server

The steps for configuring a DNS server vary, depending on the software. For example, you configure the popular BIND software by creating or editing specific text files called **zone files**. When the BIND service first starts, it reads the data in the zone files and, as with all DNS servers, it listens for DNS requests at port 53. Table 2-2 lists a few zone file entries. Each line, or record, contains the text *IN* which indicates the record can be used by DNS servers on the Internet.

Table 2-2 Zone file records used to configure a DNS server

Record	Description
`www.example.com IN A 92.100.80.40`	This A record maps the server named www in the example.com domain to the IP address 92.100.80.40. The name www.example.com is called the **canonical name** or true name of the server.
`40.80.100.92.in-addr-arpa. IN PTR www.example.com`	This PTR record is used for reverse lookup—that is, to find the name when you know the IP address. Note the IP address is reversed and *in-addr-arpa* is appended to it.

`example.com IN MX 1 panda.horse.com`
`example.com IN MX 2 jack.sally.com`
`example.com IN MX 3 susie.horse.com`
	These MX records tell mailers the preferred routes to take, ordered by best route, when sending mail to the example.com domain.

`ns1.example.com IN CNAME www .example.com`	This CNAME (canonical name) record says that the www .example.com host can also be addressed by its alias name ns1.example.com.
`www.example.com IN AAAA 2001:db8:cafe:f9::d3`	This AAAA record maps a name to an IPv6 address.

DDNS (Dynamic DNS) Suppose you want to maintain a Web server and Web site in your home office, but you don't maintain a DNS name server and you don't lease a static IP address from your ISP. How can name resolution to your Web site work without your having a DNS server and a static IP address? The solution is to sign up with a Dynamic DNS provider, such as dynDNS.org or TZO.com, to manage dynamic updates to its DNS

records for your domain name. The provider uses monitoring software and the **DDNS (Dynamic DNS)** protocol to monitor the IP addresses dynamically assigned to your home network by your ISP. The monitoring software reports IP address changes to the DDNS service, which automatically updates DNS records. Home routers sometimes provide the monitoring software embedded in the router firmware (see Figure 2-9).

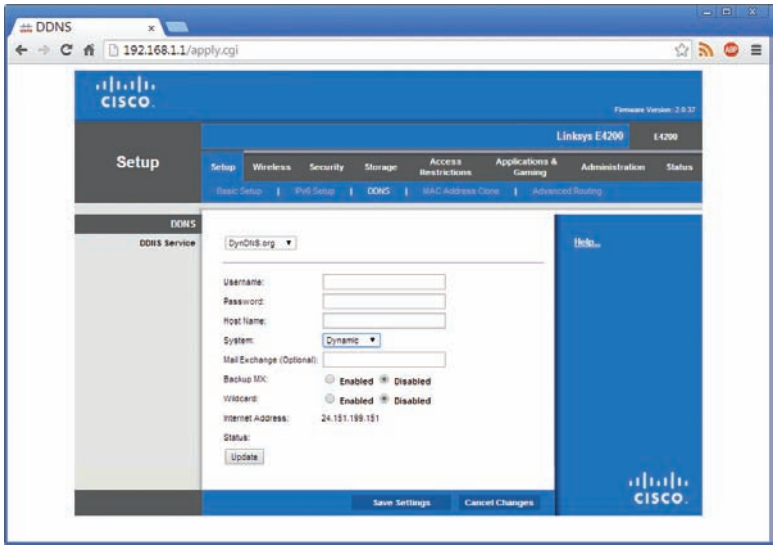


Figure 2-9 A Cisco home router can enable monitoring the IP address and report changes to dynDNS.org or TZO.com

Source: Cisco Systems, Inc.

Although a DNS record update on an authoritative name server becomes effective throughout the Internet in a matter of hours, the delay is still seen as a negative impact on those accessing your site. For this reason, most organizations are willing to pay more for a statically assigned IP address.

Now we move down to Layer 4 of the OSI model to see how port numbers are used to identify an application when it receives communication from a remote host.

How Ports and Sockets Work

Network+ 2.1 5.9 5.10

7 APPLICATION
6 PRESENTATION
5 SESSION
4 TRANSPORT
3 NETWORK
2 DATA LINK
1 PHYSICAL

Port numbers ensure that data is transmitted to the correct application among other applications running on a computer. If you compare network addressing with the addressing system used by the postal service, and you equate a host's IP address to the address of a building, then a port number is similar to an apartment number within that building.

A **socket** consists of a host's IP address and the port number of an application running on the host, with a colon separating the two values. For example, the standard port number for the Telnet service is 23. If a host has an IP address of 10.43.3.87, the socket address for Telnet running on that host is 10.43.3.87:23.

When the host receives a request to communicate on port 23, it establishes or opens a session for communication with the Telnet service and the socket is said to be open. When the TCP session is complete, the socket is closed or dissolved. You can think of a socket as a virtual circuit between a server and client. (See Figure 2-10.)

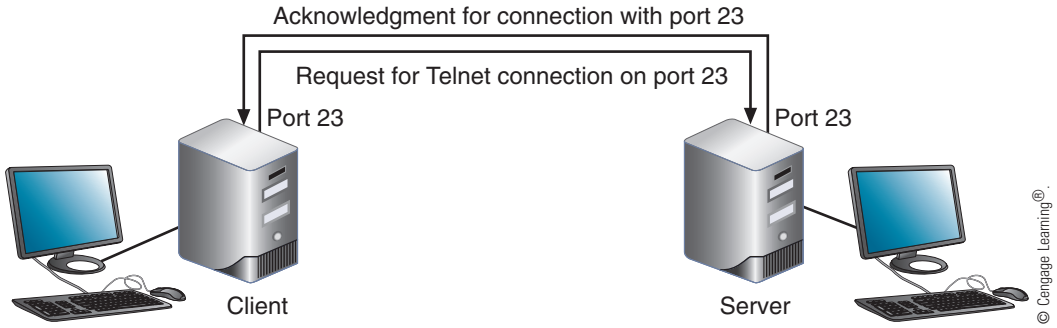
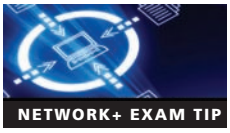


Figure 2-10 A virtual connection for the Telnet service

Port numbers range from 0 to 65535 and are divided by IANA into three types:

- *well-known ports*—These ports range from 0 to 1023 and are assigned by IANA to widely used and well-known protocols and programs, such as Telnet, FTP, and HTTP. Table 2-3 lists some of these **well-known ports** used by TCP and/or UDP.
- *registered ports*—These ports range from 1024 to 49151 and can be used by network users and processes that are not considered standard processes. Default assignments of these **registered ports** must be registered with IANA. For example, port 1109 is registered to the Kerberos authentication protocol, and port 1293 is registered to the IPsec encryption protocol. (Later in this book, you'll learn more about these two protocols.)
- *dynamic and private ports*—These ports range from 49152 to 65535 and are open for use without restriction. A **dynamic port** is a port number that can be assigned by a client or server as the need arises. For example, if a client program has several open sockets with multiple servers, it can use a different dynamic port number for each socket. A **private port** number is one assigned by a network administrator that is different from the well-known port number for that service. For example, the administrator might assign a private port number other than the standard port 80 to a Web server on the Internet so that several people can test the site before it's made available to the public. To reach the Web server, a tester must enter the private port number in the browser address box.



To prepare for the CompTIA Network+ exam, you need to memorize all the well-known port numbers listed in Table 2-3. Several of these protocols are discussed in detail in later chapters. We've put them all together in this table for easy reference.

In Chapter 1, you learned about most of the protocols listed in Table 2-3. Here's a brief description of the ones not yet covered:

- **SNMP (Simple Network Management Protocol)** is used to monitor and manage network traffic. You'll learn much more about it in Chapter 9.
- **TFTP (Trivial File Transfer Protocol)** is seldom used by humans. Computers commonly use it as they are booting up to request configuration files from another computer on the local network. TFTP uses the UDP transport protocol, whereas normal FTP uses the TCP transport protocol.

Table 2-3 Well-known TCP and UDP ports

Port number	Process name	Protocol used	Description
20	FTP-DATA	TCP	File transfer—data
21	FTP	TCP	File transfer—control (An FTP server listens at port 21 and sends/receives data at port 20)
22	SSH	TCP	Secure Shell
23	TELNET	TCP	Telnet
25	SMTP	TCP	Simple Mail Transfer Protocol
53	DNS	TCP and UDP	Domain Name System
67	DHCPv4	UDP	Dynamic Host Configuration Protocol for IPv4—client to server
68	DHCPv4	UDP	Dynamic Host Configuration Protocol for IPv4—server to client
69	TFTP	UDP	Trivial File Transfer Protocol
80	HTTP	TCP and UDP	Hypertext Transfer Protocol
110	POP3	TCP	Post Office Protocol, version 3
123	NTP	UDP	Network Time Protocol
137-139	NetBIOS	TCP and UDP	TCP/IP legacy support for the outdated NetBIOS protocols
143	IMAP	TCP	Internet Message Access Protocol
161	SNMP	TCP and UDP	Simple Network Management Protocol
443	HTTPS	TCP	Secure implementation of HTTP
445	SMB	TCP	Server Message Block
546	DHCPv6	UDP	Dynamic Host Configuration Protocol for IPv6—client to server
547	DHCPv6	UDP	Dynamic Host Configuration Protocol for IPv6—server to client
1720	H.323	TCP	Packet-Based Multimedia Communications Systems
2427/2727	MGCP	TCP and UDP	Media Gateway Control Protocol
3389	RDP	TCP	Remote Desktop Protocol
5004	RTP	UDP	Real-time Transport Protocol
5005	RTCP	UDP	Real-time Transport Control Protocol
5060	SIP	UDP	Session Initiation Protocol or SIP, not encrypted
5061	SIP	UDP	Encrypted SIP

- **NTP (Network Time Protocol)** is a simple protocol used to synchronize clocks on computers on a network.
- **SMB (Server Message Block)** was first used by earlier Windows OSs for file sharing on a network. UNIX uses a version of SMB in its **Samba** software, which is used to share files with other operating systems, including Windows systems. The cross-platform version of SMB used between Windows, UNIX, and other operating systems is called the **CIFS (Common Internet File System)** protocol.
- **SIP (Session Initiation Protocol)** is used to make an initial connection between hosts for transferring multimedia data. After the connection is established, another protocol is typically used—for example, VoIP in a video conference. SIP is a type of **signaling protocol**,

which is a protocol that makes an initial connection between hosts but that does not actually participate in data exchange. You'll learn more about SIP in Chapter 9.

- **H.323** is another signaling protocol used to make a connection between hosts prior to communicating multimedia data. H.323 has largely been replaced by SIP, which is easier to use.
- **MGCP (Media Gateway Control Protocol)** is yet another signaling protocol used to communicate multimedia data. You'll learn more about it in Chapter 9.
- **NetBIOS over TCP/IP**, also called **NetBT** or simply **NetBIOS**, is a protocol that allows old applications designed for out-of-date NetBIOS networks to work on TCP/IP networks. NetBIOS has its own name resolution service, Windows Internet Name Service (WINS), which uses port 137; a datagram service for connectionless communication, which uses port 138; and connection-oriented communication, which uses port 139. If you find NetBIOS applications on your network, you'll need to enable NetBIOS over TCP/IP to make them work. However, you'll want to replace these out-of-date applications as soon as you can.

You can use a **packet analyzer**, also called a **protocol analyzer**, to collect and examine network messages that use all of these various protocols. You'll install and use the Wireshark protocol analyzer in a project at the end of this chapter.

How IP Addresses Are Formatted and Assigned

Network+ 1.3 1.8 1.9

Recall that networks may use two types of IP addresses: IPv4 addresses, which have 32 bits, and IPv6 addresses, which have 128 bits. In this part of the chapter, you learn how IPv4 addresses are formatted and assigned. Then you learn how IPv6 addresses are formatted and assigned.

How IPv4 Addresses Are Formatted and Assigned

Network+ 1.3 1.8 1.9

7 APPLICATION
6 PRESENTATION
5 SESSION
4 TRANSPORT
3 NETWORK
2 DATA LINK
1 PHYSICAL

Recall that a 32-bit IP address is organized into four groups of 8 bits each, which are presented as four decimal numbers separated by periods, such as 72.56.105.12. Each of these four groups is called an octet. The largest possible 8-bit number is 11111111, which is equal to 255 in decimal, so the largest possible IP address in decimal is 255.255.255.255, which in binary is 11111111.11111111.11111111.11111111. Each of the four octets can be any number from 0 to 255, making a total of about 4.3 billion IPv4 addresses (256×256×256×256). Some IP addresses are reserved, so these numbers are approximations.

Computers rely on the **binary number system**, also called the **base 2 number system**, which uses only two numerals called **bits**: 0 and 1. The **octal number system** or **base 8 number system** has eight numerals (0 through 7). The CompTIA Network+ exam expects you to be familiar with the binary and octal number systems. To learn how the binary and octal number systems work and how to convert between number systems, see Appendix B.

Let's begin our discussion of IPv4 addresses by looking at how they are classified.

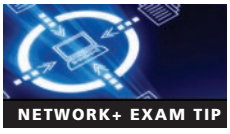
Classes of IP Addresses IPv4 addresses are divided into five classes: Class A, Class B, Class C, Class D, and Class E. When IPv4 addresses were available from IANA, a company could lease a Class A, Class B, or Class C license, acquiring multiple IP addresses in a class

license. As shown in Table 2-4, the first part of an IP address identifies the network, and the last part identifies the host.

Table 2-4 IP address classes

Class	Network octets*	Approximate number of possible networks or licenses	Approximate number of possible IP addresses in each network
A	1.x.y.z to 126.x.y.z	126	16 million
B	128.0.x.y to 191.255.x.y	16,000	65,000
C	192.0.0.x to 223.255.255.x	2 million	254

*An x, y, or z in the IP address stands for an octet that is used to identify hosts.



The CompTIA Network+ exam expects you to be able to identify the class of any IP address. For the exam, memorize the second column in Table 2-4.

Figure 2-11 shows how each class of IP addresses is divided into the network and host portions.

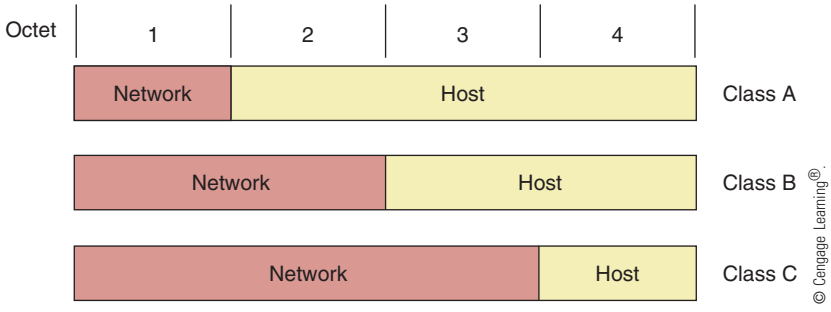


Figure 2-11 The network portion and host portion for each class of IP addresses

When class licenses were available from IANA, here's how Class A, B, and C licenses were leased:

- A **Class A** license was for a single octet. For example, a company that leased the Class A license 119 acquired 119.0.0.0 through 119.255.255.255 IP addresses.
- A **Class B** license was for the first two octets. For example, a company that leased the Class B 150.100 license acquired 150.100.0.0 through 150.100.255.255 IP addresses.
- A **Class C** license was for the first three octets. For example, a company that leased the Class C 200.80.15 license acquired 200.80.15.0 through 200.80.15.255 IP addresses.

Class D and Class E IP addresses are not available for general use. Class D addresses begin with octets 224 through 239 and are used for **multicasting**, in which one host sends messages to multiple hosts, such as when a host transmits a videoconference over the Internet. Class E addresses, which begin with 240 through 254, are reserved for research. Also, the block of addresses that begin with 127 are reserved for research and loopback addresses.

The IP addresses listed in Table 2-5 are reserved for special use by TCP/IP and should not be assigned to a device on a network.

Table 2-5 Reserved IP addresses

IP address	How it is used
255.255.255.255	Used for broadcast messages by TCP/IP background processes; a broadcast message is read by every node on the network
0.0.0.0	Currently unassigned
127.0.0.1	Indicates your own computer and is called the **loopback address**

Later in the chapter, you learn to use the loopback address to verify that TCP/IP is configured correctly on a computer. The computer actually "talks to itself" using the TCP/IP **loopback interface**, which is the computer's connection with itself. When the computer can "hear itself," you know that TCP/IP is configured correctly.

How a DHCP Server Assigns IP Addresses Recall that static IP addresses are manually assigned by the network administrator, whereas dynamic IP addresses are automatically assigned by a DHCP server each time a computer connects to the network. Because it's unmanageable to keep up with static IP address assignments, most network administrators choose to use dynamic IP addressing.

If a computer configured to use DHCP first connects to the network and is unable to lease an IPv4 address from the DHCP server, it uses an **Automatic Private IP Addressing (APIPA)** address in the address range 169.254.0.1 through 169.254.255.254.

Network+ 1.3

Applying Concepts

Configure a DHCP Server

Each type of DHCP server software is configured differently. Generally, you define a range of IP addresses, called a **DHCP scope**, to be assigned to clients when they request an address. For example, Figure 2-12 shows a screen provided by the firmware utility for a home router, which is also a DHCP server. Using this screen, you set the starting IP address (192.168.1.100 in the figure) and the maximum number of IP addresses that can be assigned (50 in the figure). Therefore, the scope or range of IP addresses this DHCP server assigns is 192.168.1.100 to 192.168.1.149.

When other nodes on the network need to know the IP address of a particular client, you can have DHCP assign the client a static IP address. A static IP address assigned by DHCP is called a DHCP **reservation**. For example, suppose a network printer needs a static IP address so that computers on the network can consistently find it. To make a reservation for the DHCP server shown in Figure 2-12, click DHCP Reservation. This displays the screen shown in Figure 2-13, where you can view the currently assigned IP addresses and reserve addresses. In Figure 2-13, the Canon1719B0 network printer has a reserved IP address of 192.168.1.100.

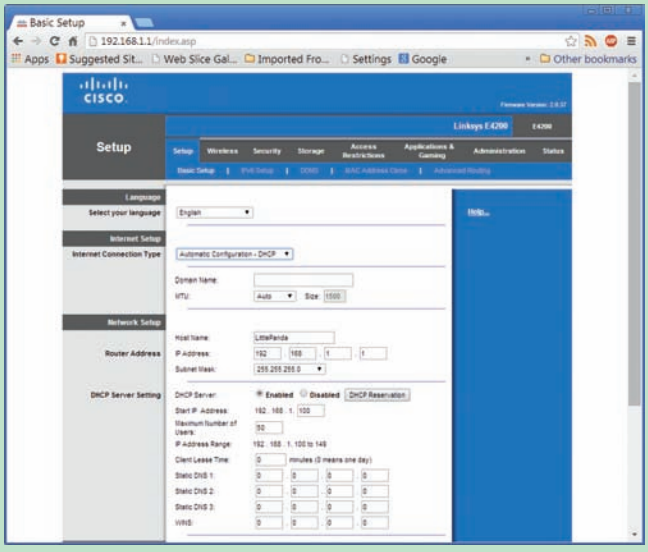


Figure 2-12 Set a range of IP addresses on a DHCP server

Source: Cisco Systems, Inc.

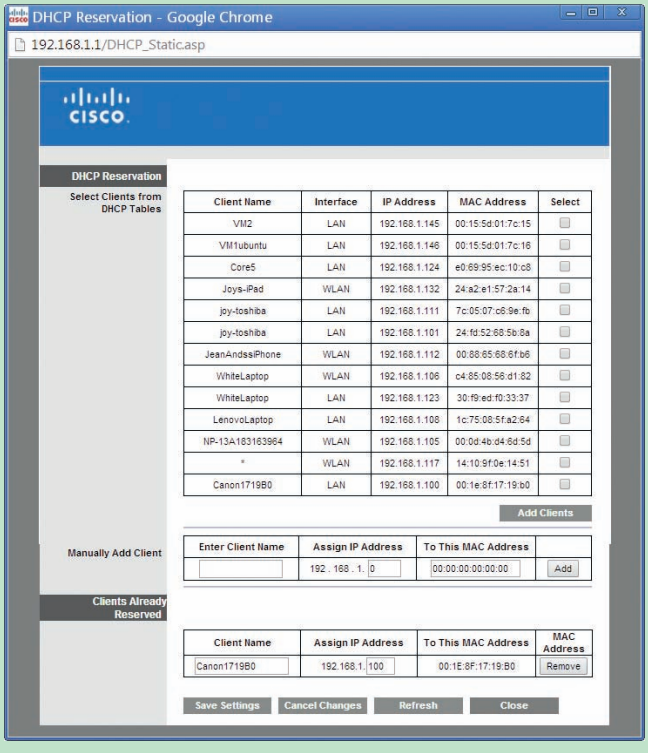
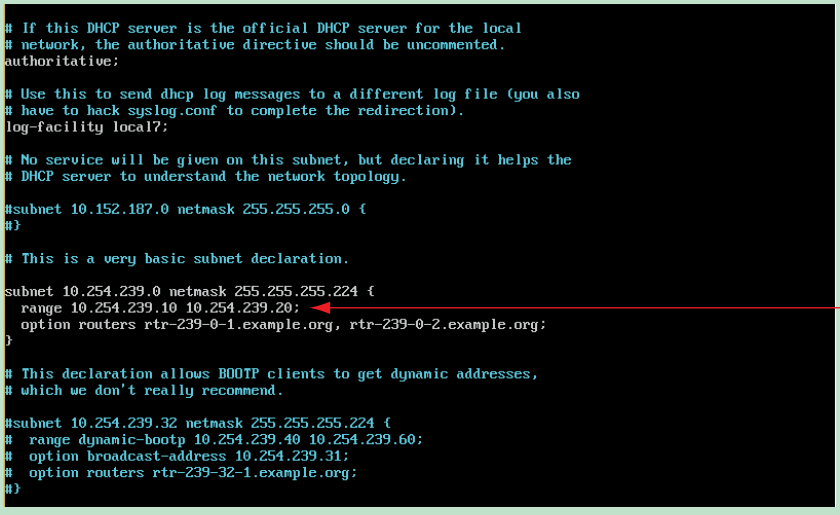


Figure 2-13 Clients on the network can reserve an IP address to be assigned by the DHCP server

Source: Cisco Systems, Inc.

In Linux systems, you configure the DHCP software by editing a text file. For example, the text file for one Linux DHCP server is dhcpd.conf, which is stored in the /etc/dhcp directory. Figure 2-14 shows the text file as it appears in the Linux **vim text editor**. A # at the beginning of a line identifies the line as a comment line (a line that is not executed). The range of IP addresses that will be assigned to clients in Figure 2-14 is 10.254.239.10 to 10.254.239.20, which is 11 IP addresses.




```
# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
#}

# This is a very basic subnet declaration.

subnet 10.254.239.0 netmask 255.255.255.224 {
  range 10.254.239.10 10.254.239.20;
  option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
}

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

#subnet 10.254.239.32 netmask 255.255.255.224 {
#  range dynamic-bootp 10.254.239.40 10.254.239.60;
#  option broadcast-address 10.254.239.31;
#  option routers rtr-239-32-1.example.org;
#}
```

Figure 2-14 Edit a text file in Linux to set an IP address range for a DHCP server

Source: Canonical Ltd.

DHCP for IPv4 servers listen at port 67 and DHCPv4 clients receive responses at port 68. When using DHCP for IPv6, DHCP servers listen at port 546 and clients receive responses at port 547.

Public and Private IP Addresses The Class A, B, and C licensed IP addresses are available for use on the Internet and are therefore called **public IP addresses**. To conserve its public IP addresses, a company can use **private IP addresses** on its private networks, which are not allowed on the Internet. IEEE recommends that the following IP addresses be used for private networks:

- 10.0.0.0 through 10.255.255.255
- 172.16.0.0 through 172.31.255.255
- 192.168.0.0 through 192.168.255.255

IEEE, a nonprofit organization, is responsible for many Internet standards. Standards are proposed to the networking community in the form of an RFC (Request for Comment). RFC 1918 outlines recommendations for private IP addresses. To view an RFC, visit the Web site *rfc-editor.org*.

Address Translation, NAT, and PAT Network Address Translation (NAT) is a technique designed to conserve the number of public IP addresses needed by a network.

A gateway device or router that stands between a private network and other networks substitutes the private IP addresses used by computers on the private network with its own public IP address when these computers need access to other networks or the Internet. The process is called **address translation**. Besides requiring only a single public IP address for the entire private network, another advantage of NAT is security; the gateway hides the entire private network behind this one address.

What happens when a host on the Internet needs to respond to the local host that sent it a request? You might wonder how the gateway knows which local host is the intended recipient, when several local hosts might have made the request. The gateway uses **Port Address Translation (PAT)** to assign a separate TCP port number to each ongoing conversation, or session, between a local host and an Internet host. See Figure 2-15. When the Internet host responds to the local host, the gateway uses PAT to determine which local host is the intended recipient.

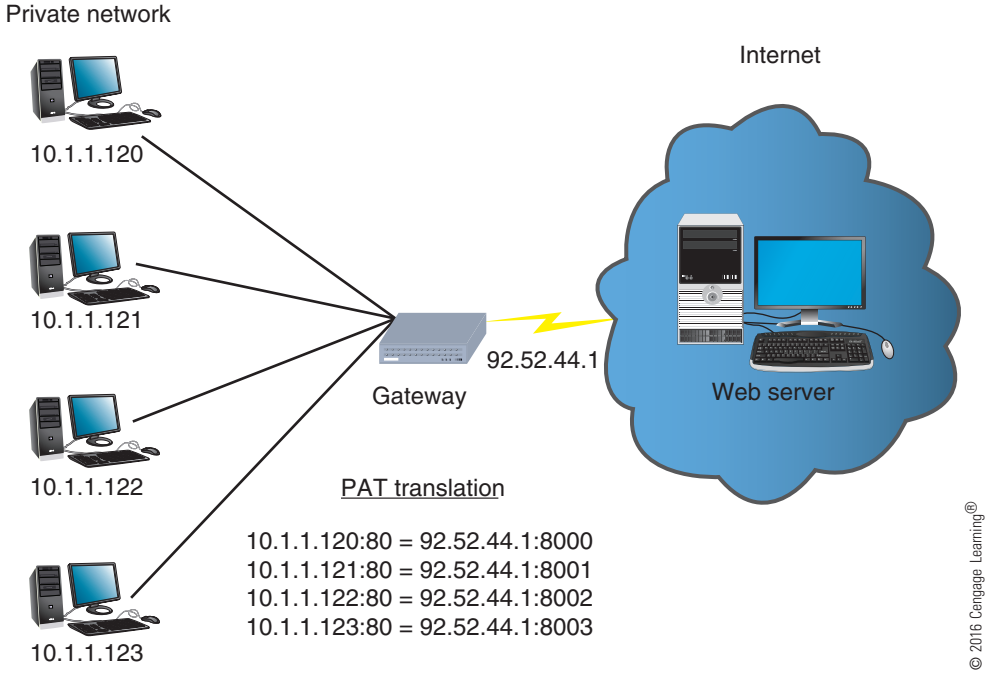


Figure 2-15 PAT (Port Address Translation)

Two variations of NAT you need to be aware of are:

- *SNAT*—Using **Static Network Address Translation (SNAT)**, the gateway assigns the same public IP address to a host each time it makes a request to access the Internet. This method works well when a local host is running a server that is accessed from the Internet. It's used on home networks that have only a single public IP address provided by an ISP.
- *DNAT or Dynamic NAT*—Using **Dynamic Network Address Translation (DNAT)**, the gateway has a pool of public IP addresses that it is free to assign to a local host whenever the local host makes a request to access the Internet. Large organizations that lease many public IP addresses use DNAT.

Network+ 1.3

Applying Concepts

Configure Address Translation Using NAT

2

For simple default gateways such as a home router, configuring address translation means making sure NAT is turned on. That's about all you can do. However, for more advanced gateways, such as an industrial-grade Cisco router or Linux server, you configure the NAT software by editing NAT translation tables stored on the device.

For example, suppose your network supports a Web server available to the Internet, as shown in Figure 2-16. On the Web, the Web site is known by the public IP address 69.32.208.74. Figure 2-17 shows the sample text file required to set up the translation tables for SNAT to direct traffic to the Web server at private IP address 192.168.10.7. (The lines that begin with ! or exclamation marks are comment lines.) The first group of lines defines the router's outside interface, which connects with the outside network, and is called the serial interface. The second group defines the router's inside Ethernet interface. The last line that is not a comment line says that when clients from the Internet send a request to IP address 69.32.208.74, the request is translated to the IP address 192.168.10.7.

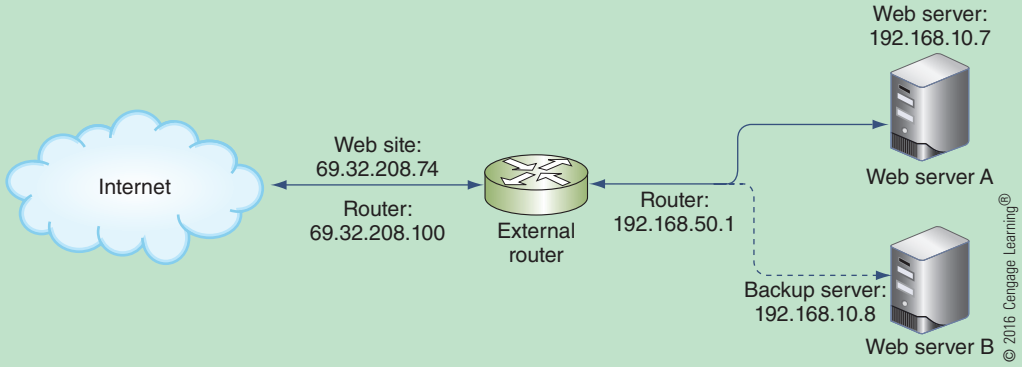


Figure 2-16 Messages to the Web site are being routed to Web server A

```
interface serial 0/0
  ip address 69.32.208.100 255.255.255.0
  ip nat outside

!--- Defines the serial 0/0 interface as the router's NAT outside interface
!--- with an IP address of 69.32.208.100

interface ethernet 1/1
  ip address 192.168.50.1 255.255.255.0
  ip nat inside

!--- Defines the Ethernet 1/1 interface as the router's NAT inside interface
!--- with an IP address of 192.168.50.1


ip nat inside source static 192.168.10.7 69.32.208.74

!--- States that source information about the inside host will be translated
!--- so the host's private IP address (192.168.10.7) will appear as the
!--- public IP address (69.32.208.74). Both ingoing and outgoing traffic
!--- exchanged with the public IP address will be routed to the host at the
!--- private IP address.
```


Figure 2-17 NAT translation table entry in Linux

At the end of this chapter, you'll create your own NAT translation table entry using this example as a template. To help you better understand where the IP address in a translation table entry comes from, answer the following questions about the information in Figures 2-16 and 2-17:

1. What is the router's outside interface IP address?
2. What is the router's inside interface IP address?
3. What is the Web site's public IP address?
4. What is the private IP address of the active Web server?

Network+ 1.8

How IPv6 Addresses Are Formatted and Assigned

The IPv6 standards were developed to improve routing capabilities and speed of communication over the established IPv4 standards and to allow for more public IP addresses on the Internet. Let's begin our discussion of IPv6 by looking at how IPv6 addresses are written and displayed:

- Recall that an IPv6 address has 128 bits that are written as eight blocks (also called quartets) of hexadecimal numbers separated by colons, like this: 2001:0000:0B80:0000:0000:00D3:9C5A:00CC
- Each block is 16 bits. For example, the first block in the preceding IP address is the hexadecimal number 2001, which can be written as 0010 0000 0000 0001 in binary.
- Leading zeroes in a four-character hex block can be eliminated. This means our sample IP address can be written as 2001:0000:B80:0000:0000:D3:9C5A:CC.
- If blocks contain all zeroes, they can be written as double colons (::). This means our sample IP address can be written two ways:
  - 2001::B80:0000:0000:D3:9C5A:CC
  - 2001:0000:B80::D3:9C5A:CC

  To avoid confusion, only one set of double colons is used in an IP address. In this example, the preferred method is the second one 2001:0000:B80::D3:9C5A:CC because the address contains the fewest zeroes.

The way computers communicate using IPv6 has changed the terminology used to describe TCP/IP communication. Here are a few terms used in the IPv6 standards:

- A **link**, sometimes called the **local link**, is any local area network (LAN) bounded by routers.
- An interface is a node's attachment to a link. The attachment can be a physical attachment using a network adapter or wireless connection or a logical attachment. For example, a logical attachment can be used for tunneling. **Tunneling** is a method used by IPv6 to transport IPv6 packets through or over an IPv4 network.
- The last 64 bits or four blocks of an IPv6 address identify the interface and are called the **interface ID** or interface identifier. These 64 bits uniquely identify an interface on the local link.
- **Neighbors** are two or more nodes on the same link.

Types of IP Addresses IPv6 classifies IP addresses differently than IPv4. IPv6 supports these three types of IP addresses, classified as to how the address is used:

- ***unicast address***—This type of address specifies a single node on a network. Two types of unicast addresses are global and link local addresses.
- ***multicast address***—Packets are delivered to all nodes in the targeted, multicast group.
- ***anycast address***—This type of address can identify multiple destinations, with packets delivered to the closest destination. For example, a DNS name server might send a DNS request to a group of DNS servers that have all been assigned the same anycast address. A router handling the request examines routes to all the DNS servers in the group and routes the request to the closest server.

Recall that with IPv4 broadcasting, messages are sent to every node on a network. However, IPv6 reduces network traffic by eliminating broadcasting. The concepts of broadcasting, multicasting, anycasting, and unicasting are depicted in Figure 2-18.

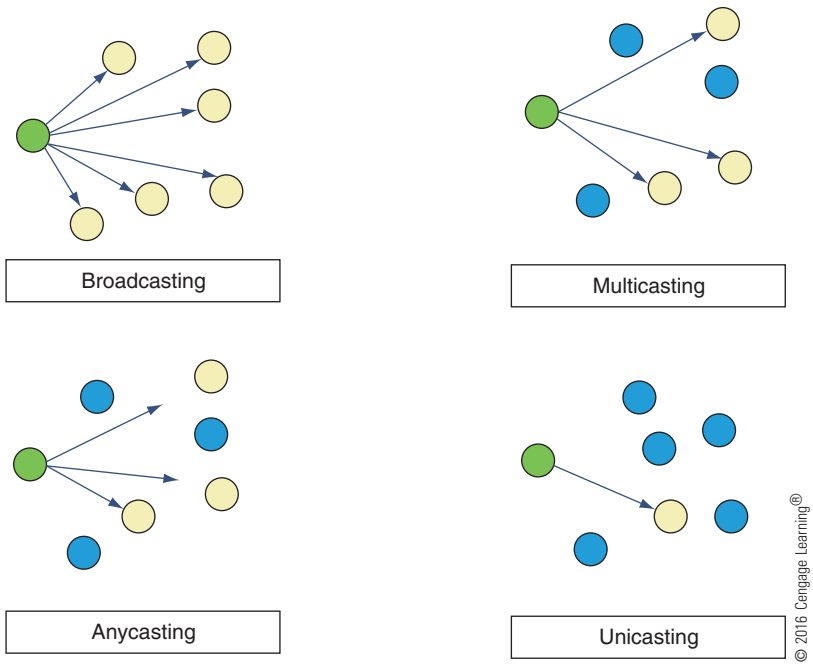


Figure 2-18 Concepts of broadcasting, multicasting, anycasting, and unicasting

Two types of unicast addresses are diagrammed in Figure 2-19 and described next:

- *global addresses*—A **global unicast address**, also called a **global address**, can be routed on the Internet. These addresses are similar to public IPv4 addresses. Most global addresses begin with the prefix 2000::/3, although other prefixes are being released. The /3 indicates that the first three bits are fixed and are always 001. Looking at Figure 2-19, notice the 16 bits or one block called the **subnet ID**, which can be used to identify a subnet on a large corporate network. A **subnet** is a smaller network within a larger network.

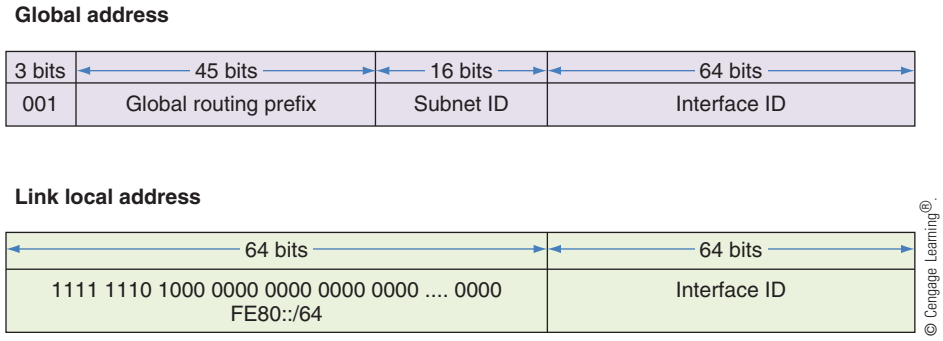


Figure 2-19 Two types of IPv6 addresses

- *link local addresses*—A **link local unicast address**, also called a **link local address** or local address, can be used for communicating with nodes in the same link. These addresses are similar to IPv4's autoconfigured APIPA addresses and begin with FE80::/10. This prefix notation means the address begins with FE80, but the first 10 bits of the reserved prefix must be followed by another 54 zeroes to make 64 bits for the network portion of the address. Link local addresses are not allowed on the Internet.

A third type of unicast address is a site local unicast address, which is a hybrid between a global and local unicast address. It was not put to popular use and was deprecated (omitted) from the latest IPv6 standards.

Table 2-6 lists some currently used address prefixes for IPv6 addresses. Notice in the table the unique local unicast addresses, which work on local links and are similar to IPv4 private IP addresses. You can expect more prefixes to be assigned as they are needed.

Table 2-6 Address prefixes for types of IPv6 addresses

IP address type	Address prefix	Notes
Global unicast	2000::/3	First 3 bits are always 001
Link local unicast	FE80::/64	First 64 bits are always 1111 1110 1000 0000 0000 0000 0000
Unique local unicast	FC00::/7	First 7 bits are always 1111 110
	FD00::/8	First 8 bits are always 1111 1101
Multicast	FF00::/8	First 8 bits are always 1111 1111

An excellent resource for learning more about IPv6 and how it works is the e-book *TCP/IP Fundamentals for Microsoft Windows*. To download the free PDF, search for it at *microsoft.com/download*.

You can use the `ipconfig` command to view IPv4 and IPv6 addresses assigned to all network connections. For example, in Figure 2-20, four IP addresses have been assigned to the physical connections on a laptop, as follows:

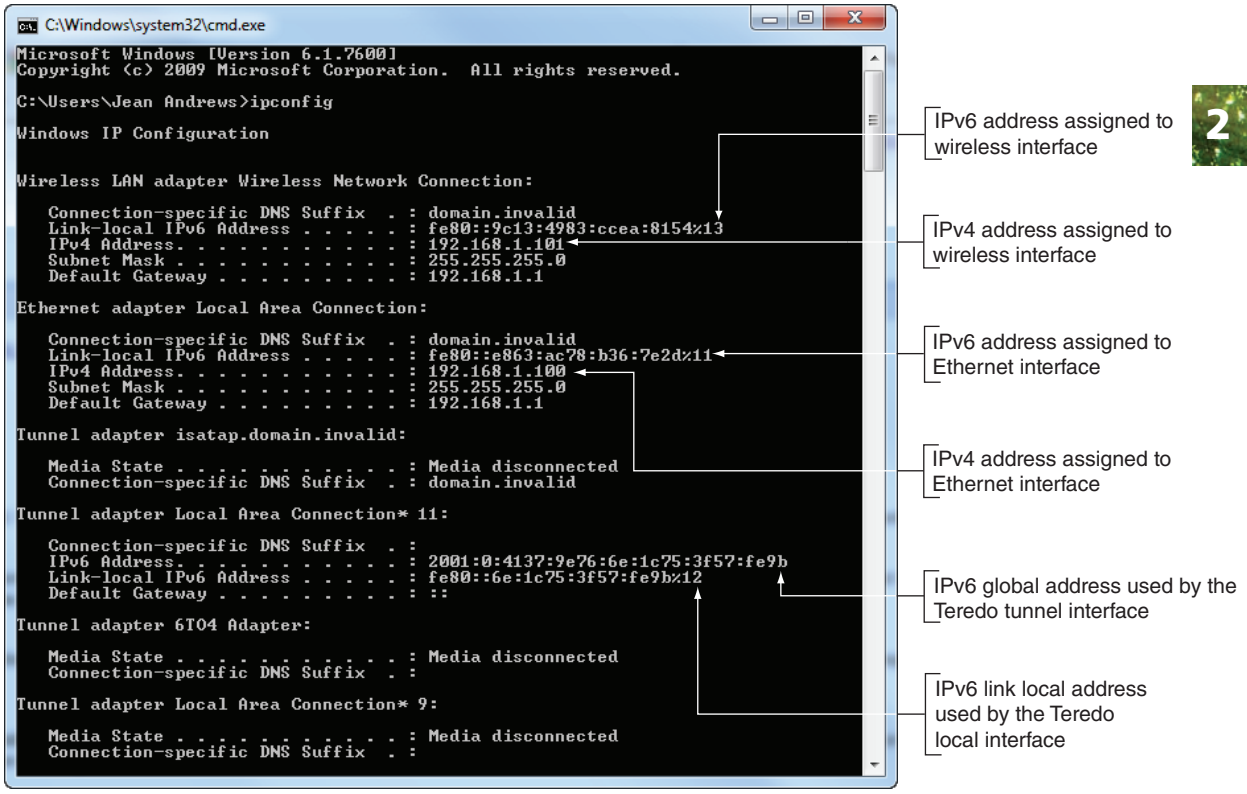


Figure 2-20 The `ipconfig` command shows IPv4 and IPv6 addresses assigned to this computer

Source: Microsoft LLC

- Windows has assigned to the wireless connection two IP addresses, one using IPv4 and one using IPv6.
- The Ethernet LAN connection has also been assigned an IPv4 address and an IPv6 address.

IPv6 addresses are followed by a % sign and a number. For example, %13 follows the first IP address in Figure 2-20. This number, which is called the **zone ID** or **scope ID**, is used to identify the link the computer belongs to.

IPv6 Autoconfiguration IPv6 addressing is designed so that a computer can autoconfigure its own link local IP address without the help of a DHCPv6 server. This is similar to how IPv4 uses an APIPA address. Here's what happens with autoconfiguration when a computer using IPv6 first makes a network connection:

Step 1—The computer creates its IPv6 address. It uses FE80::/64 as the first 64 bits or the prefix. Depending on how the OS is configured, the last 64 bits (called the interface ID) can be generated in two ways:

- *The 64 bits are randomly generated*—In this case, the IP address is called a temporary address and is never registered in DNS or used to generate global addresses for use on the Internet. The IP address changes often to help prevent hackers from discovering the computer. This is the default method used by Windows 7 and Windows 8.

- *The 64 bits are generated from the network adapter's MAC address*—MAC addresses consist of 48 bits and must be converted to the 64-bit standard, called the **EUI-64 (Extended Unique Identifier-64)** standard. To generate the interface ID, the OS takes the 48 bits of the device's MAC address, inserts a fixed 16-bit value in the middle of the 48 bits between the OUI and NIC portions, and inverts the value of the seventh bit.

The seventh bit of the 48-bit MAC address is always set to 0 if the address was assigned by the IEEE, as verification that the MAC address is, indeed, globally unique. If the seventh bit is set to 1, you know that the MAC address was assigned locally, perhaps by a virtual machine or manually by an administrator, and is therefore not necessarily unique. Either way, the seventh bit's value is reversed when the MAC address is used to generate an IPv6 address.

Step 2—The computer checks to make sure its IP address is unique on the network.

Step 3—The computer asks if a router on the network can provide configuration information. If a router responds with DHCP information, the computer uses whatever information this might be, such as the IP addresses of DNS servers or the network prefix, which will become the first 64 bits of its own IP address. The process is called prefix discovery and the computer uses the prefix to generate its own link local or global IPv6 address by appending its interface ID to the prefix.

Because a computer can generate its own link local or global IP address, a **DHCPv6** server, also called a **DHCP6** server, usually serves up only global IPv6 addresses to hosts that require static IP addresses. For example, Web servers and DNS name servers can receive their static IPv6 addresses from a DHCP6 server.

Tunneling When a network is configured to use both IPv4 and IPv6 protocols, the network is said to be **dual stacked**. However, if packets on this network must traverse other networks where dual stacking is not used, the solution is to use tunneling. Because the Internet is not completely dual stacked, tunneling is always used for IPv6 transmission on the Internet. Three tunneling protocols developed for IPv6 packets to travel over or through an IPv4 network are:

- **6to4** is the most common tunneling protocol. IPv6 addresses intended to be used by this protocol always begin with the same 16-bit prefix (called fixed bits), which is 2002 and the prefix is written as 2002::/16. The next 32 bits of the IPv6 address are the 32 bits of the IPv4 address of the sending host.
- **ISATAP** (pronounced "eye-sa-tap") stands for **Intra-Site Automatic Tunnel Addressing Protocol**. This protocol works only on a single organization's intranet. By default, ISATAP is enabled in Windows 7 and Windows 8.1.
- **Teredo** (pronounced "ter-EE-do") is named after the Teredo worm, which bores holes in wood. IPv6 addresses intended to be used by this protocol always begin with 2001 and the prefix is written as 2001::/32. Teredo is enabled by default in Windows 7, but not

Windows 8.1. On UNIX and Linux systems that don't have Teredo installed by default, you can install third-party software such as Miredo to provide the Teredo service.

To run a tunneling protocol, you simply need to enable it. The service is managed by software called the tunnel broker using automatic tunneling. The 6to4 tunneling protocol doesn't work with a tunnel broker and must be manually configured.

One more tunneling example is needed. Suppose you have a rather futuristic network that is set up to use *only* IPv6 protocols and not IPv4. In that case, IPv4 packets could only traverse the network via the **4to6** tunneling protocol. This network would also require static IPv4 routes configured on routers and Layer 3 switches so that 4to6 encapsulated IPv4 packets could arrive, pass through, and leave the network. An example is illustrated in Figure 2-21 where routers A, C, D, and Z are configured to use the 4to6 protocol to tunnel IPv4 packets that arrive from outside the intranet at gateway routers A or Z and are routed through the 4to6 tunnel to the other side of the intranet.

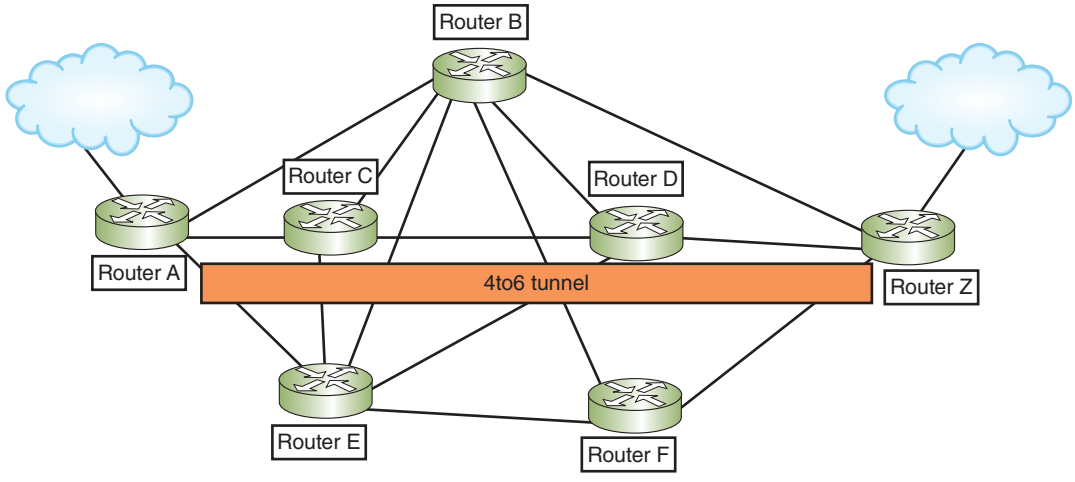


Figure 2-21 A 4to6 tunnel is used to move IPv4 packets through a futuristic IPv6 network that is configured to not use IPv4

Tools for Troubleshooting IP Address Problems

Network+ 4.2 4.6

Now that you are familiar with the basics of IP addressing, you can learn how to solve problems with IP addresses. Event Viewer is one of the first places to start looking for clues when something goes wrong with a computer. It can provide a lot of valuable information about the problems the computer is experiencing, and may even make suggestions for what to do next. For example, consider the NetBT error shown in Figure 2-22.

When Event Viewer doesn't give the information you need, you might try the command prompt instead. We used the `ipconfig` command in the Command Prompt window earlier

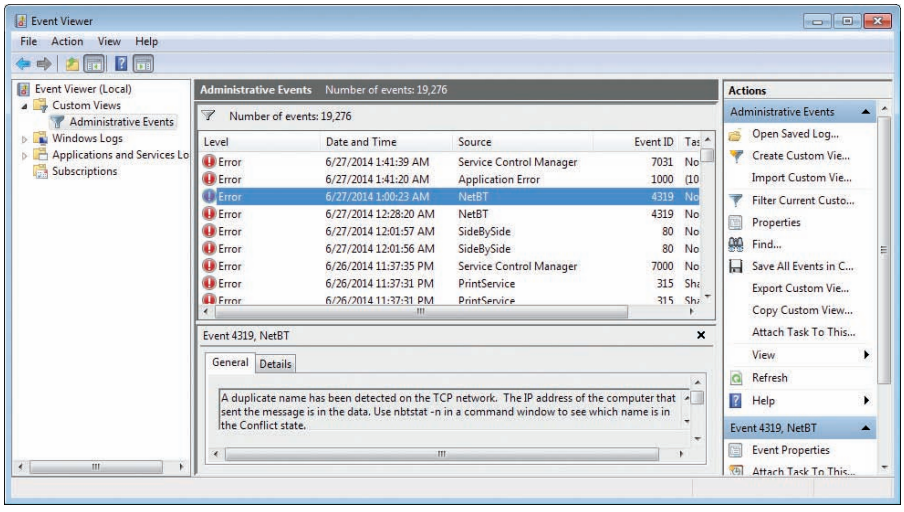


Figure 2-22 Event Viewer provided the diagnosis of a problem and recommended steps to fix the problem

Source: Microsoft LLC

in the chapter. Let's look at a few other tools here, and, in the next chapter, we'll explore a few more. These command-line tools all require a CLI.

Earlier in the chapter, you learned to access the CLI in Windows. On a Linux system, you'll need to open a shell prompt. The steps for accessing a shell prompt vary depending on the Linux distribution that you're using. For Ubuntu Desktop, click the Dashboard icon at the top of the left sidebar, and, in the Applications group, select Terminal. To close the shell prompt, use the exit command.

Network+ 4.2 4.6

ping

The utility **ping (Packet Internet Groper)** is used to verify that TCP/IP is installed, bound to the NIC, configured correctly, and communicating with the network. Think about how a whale sends out a signal and listens for the echo. The nature of the echo can tell the whale a lot of information about the object the original signal bumped into. The ping utility starts by sending out a signal called an echo request to another computer, which is simply a request for a response. The other computer then responds to the request in the form of an echo reply. The process of sending this signal back and forth is known as pinging. The protocol used by the echo request and echo reply is ICMP, a light-weight protocol used to carry error messages and information about the network.

The first tool you should use to test basic connectivity to the network, Internet, and specific hosts is ping. The `ping` command has several options or parameters, and a few of them are listed here:

```
ping [-a] [-t] [-n] [-?] [IP address] [host name] [/?]
```

Table 2-7 gives some examples of how these options can be used.

Table 2-7 Options for the `ping` command

Sample `ping` commands	Description
`ping www.google.com`	You can ping a host using its host name to verify you have Internet access and name resolution. Google.com is a reliable site to use for testing. See the results in Figure 2-23.
`ping 8.8.8.8`	Ping an IP address on the Internet to verify you have Internet access. The address 8.8.8.8, which is easy to remember, points to Google's public DNS servers.
`ping -a 8.8.8.8`	Use the `-a` parameter in the command line to test for name resolution and to display the host name to verify DNS is working.
`ping 92.10.11.200`	In this example, `92.10.11.200` is the address of a host on another subnet in your corporate network. This ping shows if you can reach that subnet.
`ping 192.168.1.1`	In this example, `192.168.1.1` is the address of your default gateway. This ping shows if you can reach it.
`ping 127.0.0.1`	Ping the loopback address, 127.0.0.1, to determine whether your workstation's TCP/IP services are running.
`ping localhost`	This is another way of pinging your loopback address.
`ping -?` or `ping /?`	These two commands display the help text for the `ping` command, including its syntax and a full list of parameters.
`ping -t 192.168.1.1`	The `-t` parameter causes pinging to continue until interrupted. To display statistics, press CTRL+Break. To stop pinging, press CTRL+C.
`ping -n 2 192.168.1.1`	The `-n` parameter defines a number of echo requests to send. By default, ping sends four echo requests. In this example, we've limited it to two.

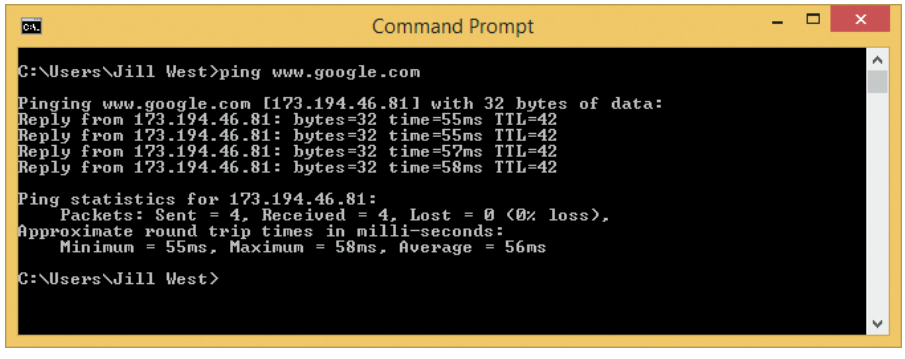


Figure 2-23 Results of a successful `ping`

Source: Microsoft LLC

IPv6 networks use a version of ICMP called **ICMPv6**. Here are two variations of ping for different operating systems, which can be used with IPv6 addresses:

- **ping6**—On Linux computers running IPv6, use `ping6` to verify whether an IPv6 host is available. When you ping a multicast address with `ping6`, you get responses from all IPv6 hosts on that subnet.
- **ping -6**—On Windows computers, use `ping` with the `-6` switch. The `ping -6` command verifies connectivity on IPv6 networks.

In Windows, the `-6` parameter is not necessary when pinging an IPv6 address (as opposed to pinging a host name) because the format of the address itself specifies that an IPv6 host is being pinged.

For the `ping6` and `ping -6` commands to work over the Internet, you must have access to the IPv6 Internet. Your ISP may provide native IPv6 connectivity, or you may be able to use an IPv6 tunnel provided by an IPv6 tunnel broker service, such as Tunnelbroker.net, offered by Hurricane Electric, or SixXS.net. Let's review some sample IPv6 pings. The command `ping -6 fe80::20c5:6548:7ba0:b92c` pings an IPv6 host on the local subnet using that host's IPv6 address. The results of this ping are shown in Figure 2-24.

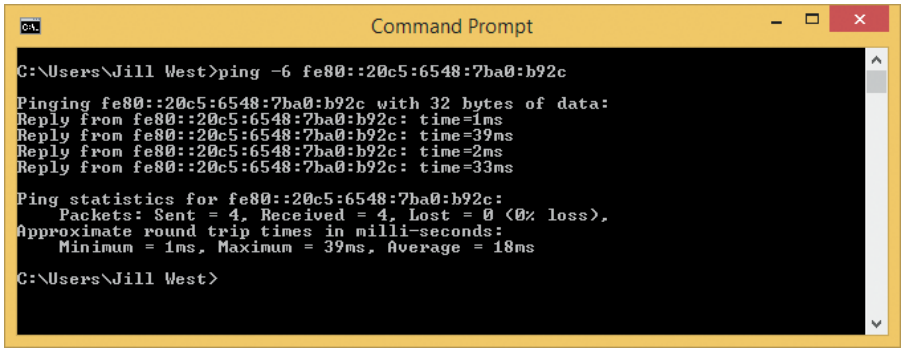


Figure 2-24 An IPv6 `ping` sent to a neighboring computer at the IPv6 address fe80::20c5:6548:7ba0:b92c

Source: Microsoft LLC

In this case, a successful connection shows that the computer issuing the `ping` command does have IPv6 capability, and that the local network supports IPv6 connectivity. Now on your own computer, try pinging Google's IPv6 DNS server, as follows: **`ping -6 2001:4860:4860::8888`**.

Figure 2-25 shows the results on a computer with an ISP that does not provide access to the IPv6 Internet; the IPv6 ping was unsuccessful.

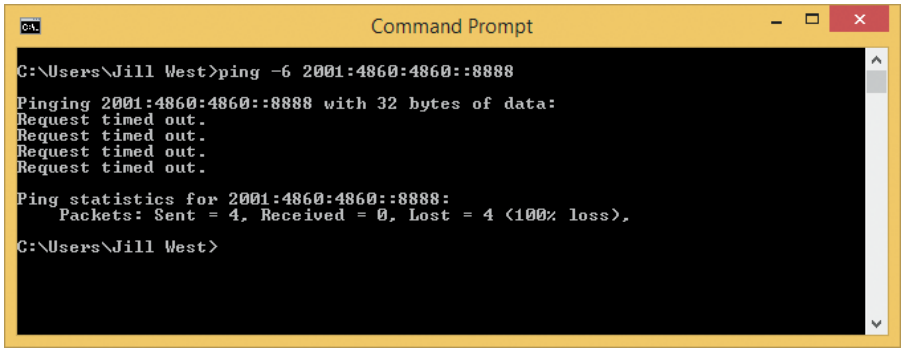


Figure 2-25 This `ping` failed because the ISP does not provide IPv6 connectivity

Source: Microsoft LLC

As IPv6 connectivity becomes more prevalent, the likelihood of needing to ping an IPv6 host on the open Web increases. The IPv6 address for Google's public DNS servers is relatively easy to remember, if you can remember a couple of simple tips.

The address is **2001:4860:4860::8888**. The IPv6 prefix 2001 is very common—recall that Teredo IP addresses begin with 2001. The next two sections can be typed out on your number pad by rotating clockwise, starting at 4 and ending at 0 (type the 4, 8, and 6 with your first three fingers and the 0 with your thumb), and do this twice. Don't forget the double colon before the next section, which replaces several sections of zeroes, and end with the same 8888 that you memorized for Google's IPv4 address.

Network+ 4.2 4.6

ipconfig

You learned about the Windows utility ipconfig earlier in this chapter. Table 2-8 describes some popular parameters for the `ipconfig` command. Notice that, with the `ipconfig` command, you need to type a forward slash (/) before a parameter, rather than a hyphen, as you do with the `ping` command.

Table 2-8 Examples of the `ipconfig` command

`ipconfig` command	Description
`ipconfig /? or ipconfig -?`	Displays the help text for the `ipconfig` command, including its syntax and a full list of parameters.
`ipconfig /all`	Displays TCP/IP configuration information for each network adapter.
`ipconfig /release`	Releases the IP address when dynamic IP addressing is being used. Releasing the IP address effectively disables the computer's communications with the network until a new IP address is assigned.
`ipconfig /release6`	Releases an IPv6 IP address.

`ipconfig /renew`	Leases a new IP address (often the same one you just released) from a DHCP server. To solve problems with duplicate IP addresses, misconfigured DHCP, or misconfigured DNS, reset the TCP/IP connection by using these two commands:
	`ipconfig /release`
	`ipconfig /renew`

`ipconfig /renew6`	Leases a new IPv6 IP address from a DHCP IPv6 server.
`ipconfig /displaydns`	Displays information about name resolutions that Windows currently holds in the DNS resolver cache.
`ipconfig /flushdns`	Flushes—or clears—the name resolver cache, which might solve a problem when the browser cannot find a host on the Internet or when a misconfigured DNS server has sent wrong information to the resolver cache.

Network+ 4.2 4.6

ifconfig

On UNIX and Linux systems, use the **ifconfig** utility to view and manage TCP/IP settings. As with ipconfig on Windows systems, you can use ifconfig to view and modify TCP/IP settings and to release and renew the DHCP configuration.

Remember that Linux and UNIX commands are case sensitive. Be sure to use the `ifconfig` command and not `Ifconfig`.

If your Linux or UNIX system provides a GUI (graphical user interface), first open a shell prompt. At the shell prompt, you can use the `ifconfig` commands listed in Table 2-9.

Table 2-9 Some `ifconfig` commands

`ifconfig` command	Description
`ifconfig`	Displays basic TCP/IP information and network information, including the MAC address of the NIC.
`ifconfig -a`	Displays TCP/IP information associated with every interface on a Linux device; can be used with other parameters. See Figure 2-26.
`ifconfig down`	Marks the interface, or network connection, as unavailable to the network.
`ifconfig up`	Reinitializes the interface after it has been taken down (via the `ifconfig down` command), so that it is once again available to the network.
`man ifconfig`	Displays the manual pages, called man pages, for the `ifconfig` command, which tells you how to use the command and about command parameters (similar to the `ipconfig /?` command in Windows).

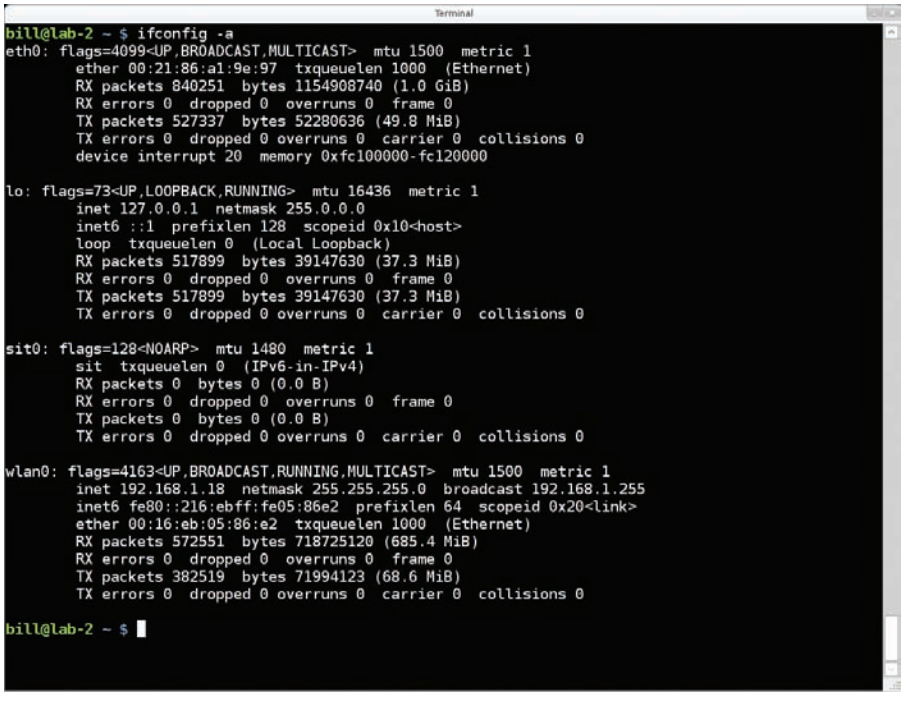


Figure 2-26 Detailed information available through `ifconfig`

Source: The Linux Foundation

Other `ifconfig` parameters, such as those that apply to DHCP settings, vary according to the type and version of the UNIX or Linux system you use.

Network+ 4.2 4.6

nslookup

The **nslookup (name space lookup)** utility allows you to query the DNS database from any computer on the network and find the host name of a device by specifying its IP address, or vice versa. This is useful for verifying that a host is configured correctly or for troubleshooting DNS resolution problems. For example, if you want to find out whether the host named *www.cengage.com* is operational, enter the command `nslookup www.cengage.com`.

Figure 2-27 shows the result of running a simple `nslookup` command.

Figure 2-27 **nslookup** shows server and host information

Source: Microsoft LLC

Notice that the command provides the target host's IP address as well as the name and address of the primary DNS server for the local network that provided the information.

To find the host name of a device whose IP address you know, you need to perform a **reverse DNS lookup**: `nslookup 69.32.208.74`. In this case, the response would include the FQDN for the target host and the name and address of the primary DNS server that made the response.

The `nslookup` command is primarily used for troubleshooting DNS servers. If you think your DNS server at IP address 24.159.64.23 is down, you can perform this test: `nslookup 127.0.0.1 24.159.64.23`. This command directs your DNS server at 24.159.64.23 to resolve the identity of the host at 127.0.0.1, which of course is the local host. (See Figure 2-28.)

Figure 2-28 DNS server returns the identity of the local host

Source: Microsoft LLC

The nslookup utility is available in two modes: interactive and noninteractive. So far you've used nslookup in noninteractive mode, which gives a response for a single `nslookup` command. This is fine when you're investigating only one server, or when you're retrieving single items of information at a time. However, to test multiple DNS servers at one time, you'll want to use the nslookup utility in interactive mode, which makes available more of the utility's options. To launch interactive mode, type the `nslookup` command without any parameters and press Enter.

As shown in Figure 2-29, after you enter this command, the command prompt changes to a greater-than symbol (>). You can then use additional commands to find out more about the contents of the DNS database. For example, on a computer running UNIX, you could view a list of all the host name and IP address correlations on a particular DNS server by entering the command `ls`. Or you could specify 5 seconds as the period to wait for a response instead of the default of 10 seconds by entering `timeout=5`.

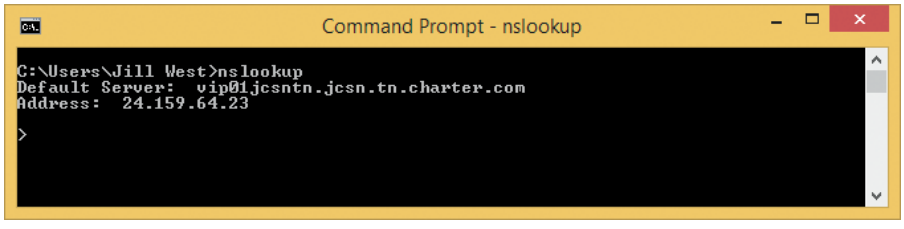


Figure 2-29 Interactive mode of the nslookup utility

Source: Microsoft LLC

You can change DNS servers from within interactive mode with the `server` subcommand and specifying the IP address of the new DNS server. Before using the `server` subcommand, it's helpful to use the `ipconfig /all` command in noninteractive mode to determine what other DNS servers are available. After you pick a DNS server, you can enter nslookup's interactive mode, and then assign a new DNS server with the command: `server 24.217.0.5`. This example, shown in Figure 2-30, assigns a DNS server with the address 24.217.0.5.

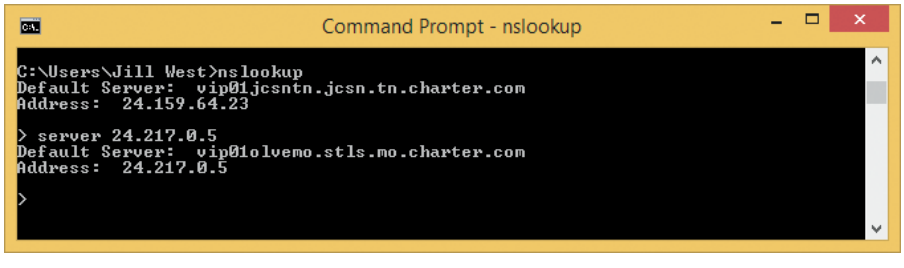


Figure 2-30 The `server` subcommand can be used to change DNS servers

Source: Microsoft LLC

To exit nslookup's interactive mode and return to the normal command prompt, enter `exit`.

Many other nslookup options exist. To see these options on a UNIX or Linux system, use the `man nslookup` command. On a Windows-based system, use the `nslookup ?` command.

Chapter Summary

Overview of Addressing on Networks

- Hosts on a network are assigned host names, which include the organization's domain name. DNS keeps track of which host name belongs to each IP address.
- Applications are assigned one or more port numbers to communicate with other applications.
- IPv4 addresses have 32 bits and are written as four decimal numbers called octets. IPv6 addresses have 128 bits and are written as eight blocks of hexadecimal numbers.
- Every NIC on the globe is assigned a unique 48-bit MAC address, which is frequently written as 12 hexadecimal numerals separated by colons. The first part of the MAC address is the 24-bit OUI, which identifies the NIC's manufacturer. The second part is the 24-bit device ID, which is based on the NIC's model and manufacture date.
- You can assign a static IP address to a computer or device, or you can configure a device to receive a dynamic IP address from a DHCP server each time the device connects to the network.
- Use the `ipconfig` command in the Command Prompt window to view IP configuration information. `ipconfig /all` shows more complete configuration information.

How Host Names and Domain Names Work

- A fully qualified domain name (FQDN) includes both a host name portion and a domain name portion. The last part of a host name is the top-level domain (TLD).
- Name resolution is the process of matching an FQDN to its IP address.
- The hosts file is a text file that contains a list of IP addresses and associated host names. In UNIX or Linux, the hosts file is stored in the /etc directory. On a Windows computer, the hosts file is located in the \Windows\System32\drivers\etc folder.
- DNS is an automated name resolution service that operates at the Application layer. The DNS namespace is the entire collection of computer names and their associated IP addresses stored in databases around the world. Hierarchical name servers hold these databases, and resolvers request the information from the name servers.
- DNS root servers hold information used to locate TLD servers. TLD servers hold information about authoritative servers, which maintain authoritative records of computer names and IP addresses in their domains.
- A resolver on the client computer sends a recursive query, which demands resolution, to a local DNS server; the local server takes responsibility for ensuring that a recursive query is resolved. The local server sends iterative queries to other servers—that is, it sends queries that the other servers respond to with whatever information they have. Iterative queries do not require a resolution.
- DNS data is spread throughout the globe in a distributed database model. Each hosting organization is responsible for providing DNS authoritative servers for public access to the DNS zone it manages.

- Several DNS server software options are available, the most popular being BIND (Berkeley Internet Name Domain).
- Windows Server includes its own Microsoft DNS Server, which can be configured as an integral part of Active Directory. A split DNS design places a separate DNS server in the DMZ for public access in order to protect Active Directory and the rest of an organization's internal network from the outer Web.
- Five common types of DNS resource records are A (Address) records, AAAA (Address) records, CNAME (Canonical Name) records, PTR (Pointer) records, and MX (Mail Exchanger) records.
- Small organizations may choose to use DDNS (Dynamic DNS) to report on IP address assignment changes to their Web server and Web sites when they don't want to pay for a static IP address.

How Ports and Sockets Work

- An IP address and a port number written together, for example 10.43.3.87:23, is called a socket. During a communication session, the socket is open. When the session is complete, the socket is closed.
- Well-known ports range from 0 to 1023 and are assigned by IANA. Registered ports range from 1024 to 49151; only default assignments of these ports are registered with IANA. Dynamic ports and private ports range from 49152 to 65535 and are open for use without restriction.
- You can use a packet analyzer to collect and examine network messages that use several protocols, including SNMP, SSH, TFTP, NTP, SMB, CIFS, SIP, H.323, and MGCP.

How IP Addresses Are Formatted and Assigned

- Each of the four octets in an IPv4 address can be any number from 0 to 255, making a total of about 4.3 billion possible IPv4 addresses.
- Class A addresses range from 1.x.y.z to 126.x.y.z. Class B addresses range from 128.0.x.y to 191.255.x.y. Class C addresses range from 192.0.0.x to 223.255.255.x. Reserved IPv4 addresses include 255.255.255.255, 0.0.0.0, and 127.0.0.1.
- You can define a range of available IP addresses in DHCP, or assign a static IP address as a DHCP reservation, such as for a network printer.
- Classes A, B, and C IP addresses are available as both public and private addresses. Class A private addresses are 10.0.0.0 through 10.255.255.255. Class B private addresses are 172.16.0.0 through 172.31.255.255. Class C private addresses are 192.168.0.0 through 192.168.255.255.
- NAT (Network Address Translation) is used to allow devices that have private IP addresses access to the Internet and to protect these devices on a private network from direct exposure to the Internet. Translation table entries can be used to configure static NAT assignments.
- According to IPv6 standards, a link is any local area network bounded by routers. An interface is a node's attachment to a link. The last 64 bits of an IPv6 address are the interface identifier. Neighbors are two or more nodes on the same link.
- Tunneling protocols are used to allow IPv6 packets to travel over or through an IPv4 network: ISATAP, Teredo, Miredo, and 6to4. The 4to6 protocol is a futuristic

protocol intended to be used to tunnel IPv4 packets over an IPv6 network that does not support IPv4 traffic.

- The three types of IPv6 addresses are unicast, multicast, and anycast addresses. Two types of unicast addresses are global and link local addresses.
- IPv6 addressing is designed so a computer can configure its own link local or global IP address without the help of a DHCPv6 server. The settings for this autoconfiguration feature can be adjusted to generate a random number, or to use the NIC's MAC address to define the last 64 bits of the address.

Tools for Troubleshooting IP Address Problems

- The ping utility uses ICMP to verify that TCP/IP is installed, bound to the NIC, configured correctly, and communicating with the network.
- The ipconfig utility is useful for viewing and adjusting a Windows computer's TCP/IP settings.
- On UNIX and Linux systems, the ifconfig utility is used to view and manage TCP/IP settings, including DHCP configuration.
- The nslookup utility allows you to query the DNS database from any computer on the network. You can use nslookup in interactive mode to test multiple DNS servers at a time and change the DNS server selected for the device.

Key Terms

For definitions of key terms, see the Glossary near the end of the book.

4to6
6to4
A (Address) record
AAAA (Address) record
address translation
alias
anycast address
authoritative server
Automatic Private IP Addressing (APIPA)
base 2 number system
base 8 number system
binary number system
BIND (Berkeley Internet Name Domain)
bit
block ID
caching-only server
canonical name
CIFS (Common Internet File System)
Class A
Class B
Class C
CNAME (Canonical Name) record
command-line interface (CLI)
company-ID
computer name
DDNS (Dynamic DNS)
default gateway
device ID
DHCP (Dynamic Host Configuration Protocol)
DHCP6
DHCPv6
DHCP scope
distributed database model
DMZ (demilitarized zone)
DNS (Domain Name System or Domain Name Service)
DNS cache
DNS server
DNS zone
domain name
dual stacked
dynamic IP address
Dynamic Network Address Translation (DNAT)
dynamic port
elevated command prompt window
EUI-64 (Extended Unique Identifier-64)
extension identifier
firewall
fully qualified domain name (FQDN)

fully qualified host name
gateway
global address
global unicast address
H.323
hex number
hexadecimal number
host name
host table
hosts file
ICMPv6
ifconfig
interface
interface ID
Internet Corporation for Assigned Names and Numbers (ICANN)
Internet Protocol version 4 (IPv4)
Internet Protocol version 6 (IPv6)
ipconfig
ISATAP (Intra-Site Automatic Tunnel Addressing Protocol)
iterative query
link
link local address
link local unicast address
local link
loopback address
loopback interface
MGCP (Media Gateway Control Protocol)
Miredo
multicast address
multicasting
MX (Mail Exchanger) record
name resolution
name server
namespace
neighbor
NetBIOS
NetBT (NetBIOS over TCP/IP)
Network Address Translation (NAT)
nslookup (name space lookup)
NTP (Network Time Protocol)
octal number system
octet
open source
OUI (Organizationally Unique Identifier)
packet analyzer
ping (Packet Internet Groper)
ping -6
ping6
Port Address Translation (PAT)
private IP address
private port
protocol analyzer
PTR (Pointer) record
public IP address
recursive query
registered port
reservation
resolver
resource record
reverse DNS lookup
root server
Samba
scope ID
shell prompt
signaling protocol
SIP (Session Initiation Protocol)
SMB (Server Message Block)
SNMP (Simple Network Management Protocol)
socket
split DNS
split-horizon DNS
static IP address
Static Network Address Translation (SNAT)
subnet
subnet ID
subnet mask
Teredo
TFTP (Trivial File Transfer Protocol)
Time to Live field
top-level domain (TLD)
tunneling
unicast address
vim text editor
well-known port
zone file
zone ID
zone transfer

Review Questions

1. Which part of a MAC address is unique to each manufacturer?
   a. The network identifier
   b. The OUI
   c. The device identifier
   d. The physical address

2. What decimal number corresponds to the binary number 11111111?
   a. 255
   b. 256
   c. 127
   d. 11111111
3. What type of device does a computer turn to first when attempting to make contact with a host on another network?
   a. Default gateway
   b. DNS server
   c. Root server
   d. DHCP server
4. Which statement describes SMTP?
   a. SMTP is a connectionless protocol that uses UDP
   b. SMTP is a connection-based protocol that uses UDP
   c. SMTP is a connectionless protocol that uses TCP
   d. SMTP is a connection-based protocol that uses TCP
5. When your computer first joins an IPv6 network, what is the prefix of the IP address the computer first configures for itself?
   a. FE80::/10
   b. FF00::/8
   c. 2001::/64
   d. 2001::/3
6. You have just brought online a new secondary DNS server and notice your monitoring software reports a significant increase in network traffic. Which two hosts on your network are likely to be causing the increased traffic and why?
   a. The caching and primary DNS servers, because the caching server is requesting zone transfers from the primary server
   b. The secondary and primary DNS servers, because the secondary server is requesting zone transfers from the primary server
   c. The root and primary DNS servers, because the primary server is requesting zone transfers from the root server.
   d. The Web server and primary DNS server, because the Web server is requesting zone transfers from the primary DNS server.
7. Suppose you send data to the 11111111 11111111 11111111 11111111 IP address on an IPv4 network. To which device(s) are you transmitting?
   a. All devices on the Internet
   b. All devices on your local network

c. The one device with this given IP address

d. Because no device can have this given IP address, no devices receive the transmission

8. If you are connected to a network that uses DHCP, and you need to terminate your Windows workstation's DHCP lease, which command would you use?

   a. `ipconfig/release`

   b. `ipconfig/renew`

   c. `ifconfig/release`

   d. `ifconfig/new`

9. What computers are the highest authorities in the Domain Name System hierarchy?

   a. Authoritative name servers

   b. Root servers

   c. Top-level domain servers

   d. Primary DNS server

10. What version of SMB can be used across Windows, UNIX, and other operating systems?

    a. SIP (Session Initiation Protocol)

    b. RDP (Remote Desktop Protocol)

    c. CIFS (Common Internet File System)

    d. MGCP (Media Gateway Control Protocol)

11. Suppose you want to change the default port for RDP as a security precaution. What port does RDP use by default, and from what range of numbers should you select a private port number?

12. Which type of DNS record identifies a mail server?

13. How many bits does an IPv6 address contain?

14. On what port is an IPv6 client listening for DHCP messages?

15. The second 64 bits of an autoconfigured IPv6 address may either be random or generated from the computer's MAC address, which contains 48 bits. What standard defines the conversion of the MAC address to the IPv6 64-bit device ID?

16. You issue a transmission from your workstation to the following socket on your LAN: 10.1.1.145:110. Assuming your network uses standard port designations, what Application layer protocol are you using?

17. What protocol does a network gateway use to keep track of which internal client is talking to which external Web server?

18. You are the network manager for a computer training center that allows clients to bring their own laptops to class for learning and taking notes. Clients need access to the Internet, so you have configured your network's DHCP server to issue IP addresses automatically. What DHCP option should you modify to make sure you are not wasting addresses that were used by clients who have left for the day?

19. What is the range of IP addresses that might be assigned by APIPA?

20. While troubleshooting a network connection problem for a coworker, you discover the computer is querying a nonexistent DNS server. What command-line utility can you use to assign the correct DNS server IP address?
21. FTP sometimes uses a random port for data transfer, but an FTP server always, unless programmed otherwise, listens to the same port for session requests from clients. What port is the FTP server listening on?
22. While troubleshooting a network connection problem for a coworker, you discover that the computer has a static IP address and is giving a duplicate IP address error. What command-line utility can you use to find out what other device may already be using that IP address?
23. What is the IPv4 loopback address? What is the IPv6 loopback address?
24. You have just set up a new wireless network in your house, and you want to determine whether your Linux laptop has connected to it and obtained a valid IP address. What command will give you the information you need?
25. You have decided to use SNAT and PAT on your small office network. At minimum, how many IP addresses must you obtain from your ISP for all five clients in your office to be able to access servers on the Internet?
26. If you know that your colleague's TCP/IP host name is JSMITH, and you need to find out his IP address, what command should you type at your shell prompt or command prompt?
27. When determining whether a local network has any NetBIOS traffic, do you use the nslookup utility in interactive mode or a packet analyzer such as Wireshark?
28. List three signaling protocols discussed in the chapter that are used for communicating multimedia data.
29. What version of the `ping` command do you use in Windows with IPv6 addresses? What version do you use on a Linux system?
30. When running a scan on your computer, you find that a session has been established with a host at the address 208.85.40.44:80. Which protocol is in use for this session? What command-line utility might you use to find out who the host is?

Hands-On Projects

Project 2-1: Create a NAT Translation Table Entry

Your corporation hosts a Web site at the static public IP address 92.110.30.123. A router directs this traffic to a Web server at the private IP address 192.168.11.100. However, the Web server needs a hardware upgrade and will be down for two days. Your network administrator has asked you to configure the router so that requests to the IP address 92.110.30.123 are redirected to the backup server for the Web site, which has the private IP address 192.168.11.110. The router's inside Ethernet interface uses IP address 192.168.11.254 and its outside interface uses

the IP address 92.110.30.65. Answer the following questions about the new static route you'll be creating:

1. What is the router's outside interface IP address?
2. What is the router's inside interface IP address?
3. What is the Web site's public IP address?
4. What is the private IP address of the backup Web server?

Use the example given earlier in the chapter as a template to create the NAT translation table entries for the address translation. For the subnet masks, use the default subnet mask for a Class C IP address license. Include appropriate comment lines in your table.

Project 2-2: View and Change IPv6 Autoconfiguration

By default, when configuring an IPv6 address, Windows 8 generates a random number to fill out the bits needed for the NIC portion of the IPv6 address. This security measure helps conceal your device's MAC address, and further protects your privacy by generating a new number every so often. There may be times, however, when you need your system to maintain a static IPv6 address. To do this, you can disable the temporary IPv6 address feature using the Netsh utility in an elevated command prompt window. Do the following:

1. Open an elevated command prompt window.
2. Find your computer's current IPv6 address and MAC address. Carefully compare the two addresses. Are they in any way numerically related?
3. To disable the random IP address generation feature, enter the command:

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

4. To instruct Windows to use the EUI-64 standard instead of the default settings, use this command:

   ```
   netsh interface ipv6 set privacy state=disabled
   ```

5. What is your computer's new IPv6 address? Notice that the fixed value FF FE has been inserted halfway through the MAC address values in the second half of the IPv6 address. Note that the host portion of the IPv6 address may look slightly different because of the way the values are converted for use by IPv6. Recall that the seventh bit of the MAC address is inverted, resulting in a slightly different value.
6. Reenable random IPv6 address generation with these commands:

   ```
   netsh interface ipv6 set global randomizeidentifiers=enabled
   netsh interface ipv6 set privacy state=enabled
   ```

Project 2-3: Manage DNS

You have learned that clients as well as name servers store DNS information to associate names with IP addresses. In this project, you view the contents of a local DNS cache, clear it, and view it again after performing some DNS lookups. Then you change DNS servers and view the DNS cache once again.

1. To view the DNS cache, open a command prompt and enter the following command: **ipconfig /displaydns**
2. If this computer has been used to resolve host names with IP addresses—for example, if it has been used to retrieve mail or browse the Web—a list of locally cached resource records appears. Read the file to see what kinds of records have been saved, using the scroll bar if necessary. How many are A records and how many are a different type, such as CNAME?
3. Next clear the DNS cache with this command: **ipconfig /flushdns**

The operating system confirms that the DNS resolver cache has been flushed. One circumstance in which you might want to empty a client's DNS cache is if the client needs to reach a host whose IP address has changed (for example, a Web site whose server was moved to a different hosting company). As long as the DNS information is locally cached, the client will continue to look for the host at the old location. Clearing the cache allows the client to retrieve the new IP address for the host.

4. View the DNS cache again with the command: **ipconfig /displaydns**

Because you just emptied the DNS cache, you will receive a message that indicates that Windows could not display the DNS resolver cache. (See Figure 2-31.)

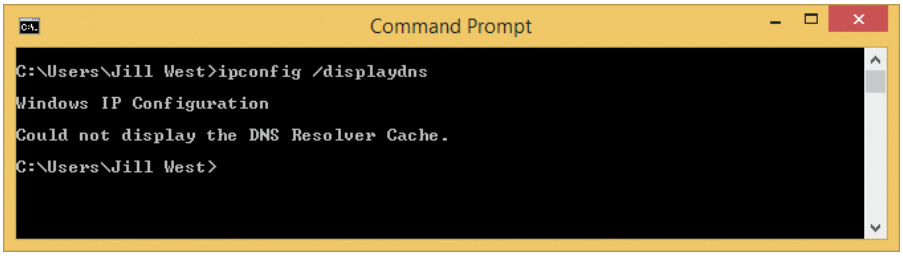


Figure 2-31 This DNS cache is empty

Source: Microsoft LLC

5. Switch to your browser window and go to **www.cengage.com**. Next go to **www.google.com**. Finally, go to **www.loc.gov**.
6. Return to the Command Prompt window and view the DNS cache once more to see a new list of resource records using this command: **ipconfig /displaydns**
7. Scroll up through the list of resource records and note how many associations were saved in your local DNS cache after visiting just three Web sites. How many hosts are identified for each site you visited? What type of record is most common? Can you think of any situations, other than wanting to reach a host that has moved to a different address, in which you might want to clear your DNS cache?

By default, DHCP supplies the IP addresses of DNS servers when you first connect to a network. When traveling, you can still use your organization's DNS servers, even when they are far away from your laptop. Doing so means you don't have to rely on DNS servers provided by a public hot spot, which might be controlled by hackers.

Follow these steps to view or change the name server information on a Windows 8.1 workstation:

8. Open the Network and Sharing Center and click **Change adapter settings**.

9. Right-click the connection you want to configure, and click **Properties** on the shortcut menu. Respond to the UAC box as necessary.
10. On the Networking tab under "This connection uses the following items," select **Internet Protocol Version 4 (TCP/IPv4)**, and click **Properties**. The Internet Protocol Version 4 (TCP/ IPv4) Properties dialog box opens.
11. To change the default settings and specify the DNS server for your workstation, rather than allowing DHCP to supply the DNS server address, on the General tab, click **Use the following DNS server addresses**.
12. Enter the IP address for your primary DNS server in the Preferred DNS server space and the address for your secondary DNS server in the Alternate DNS server space. For the purposes of this project, if your instructor has not specified another DNS server, you can point to Google's public DNS servers. Use 8.8.8.8 as the Preferred DNS server and 8.8.4.4 as the Alternate DNS server.
13. Now that you have changed your DNS servers, will you still have DNS data stored in your DNS cache? To find out, return to the command prompt and view the DNS cache to see what records are still there. Then close all windows, saving your changes.

Project 2-4: Set Up an FTP Server

In this project, using the small network you created in Chapter 1 in Project 1-1, you install and use FTP, which is a client-server application. Designate one computer as computer A, the server, and the other computer as computer B, the client. Do the following using computer A:

1. Create a folder named **Normal Users** and create a file in the folder named **Normal Users.txt**. Later, any files or folders you want on your FTP site can be stored in this folder.
2. Go to **filezilla-project.org** and download the free FTP FileZilla Server software to your desktop and install the software. As you do so, be sure to not accept other free software the site offers. You might need to restart the installation as you reject other software.
3. When the FileZilla server installs, accept all default settings, which places a shortcut on your desktop and sets the FTP service to start automatically.
4. During the installation, the Connect to Server dialog box appears (see Figure 2-32). Enter an administration password and be sure to write down this password. Because you're running only one FTP server on computer A, check the **Always connect to this server** check box. Also note the Server Address is 127.0.0.1, which is your loopback IP address. When you click OK, the FileZilla Server admin window opens. You can also open the admin window by using the shortcut on your desktop.
5. You're now ready to configure your FTP server. To set up a user group, click **Edit, Groups**. In the right pane under Groups, click **Add**. In the Add user group dialog box, type **Normal Users** and click **OK**.
6. Under Directories, click **Add**. Point to the **Normal Users** folder and click **OK**. The folder is listed under Directories. Under Directories, select the **Normal Users** directory and then click **Set as home dir.** Click **OK**.
7. Next, click **Edit, Users**, and create a new user named **User1**. Put the user in the **Normal Users** group.

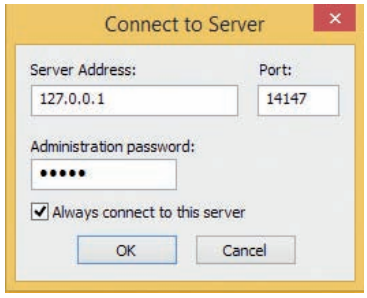


Figure 2-32 Enter the admin password used to log on and manage the FileZilla FTP server

Source: Microsoft LLC

8. In the Account settings pane, check **Password** and assign the password **password**. Click **OK**.
9. To verify the service is working, let's use the FTP client commands embedded in Windows. As you work, watch the dialogue recorded in the FileZilla Server window (see Figure 2-33). Open a Command Prompt window and enter the following:

Command to connect to the FTP service:	`ftp 127.0.0.1`
Enter your user ID:	`User1`
Enter the password:	`password`
Command to list the contents of the shared folder:	`dir`
Command to close the FTP session:	`quit`

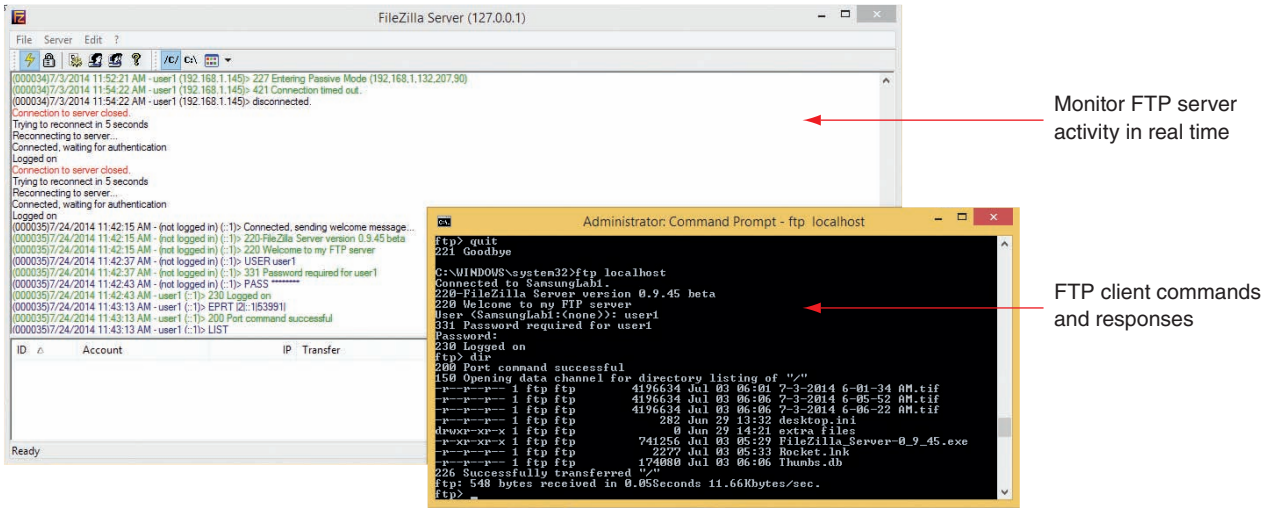


Figure 2-33 Use the FileZilla Server window to monitor real-time activity on the FTP server

Source: Microsoft LLC

10. In the FileZilla Server window, click **Edit**, **Settings**. Under General settings, note that the server is listening at port 21. You can now close the window.

11. The server software is still running as a background service, listening at port 21 for clients to initiate a session. To see the service running, open the Windows Services console. To do so, right-click **Start**, click **Run**, type **services.msc**, and press **Enter**. In the Services console, verify that the FileZilla service is running and set to start automatically each time the computer starts. Close the Services console.

12. To find out the IP address of computer A, in the Command Prompt window, enter **`ipconfig`**. What is the IP address?

Using computer B, you're now ready to test the FTP client. Do the following:

13. Open a Command Prompt window and ping computer A. The ping should give replies from computer A, indicating connectivity.

14. Now try the same commands as in step 8 above, using the IP address of computer A in the first command line. Most likely, you will not be able to connect because the firewall on computer A blocks incoming connections on port 21 by default.

On computer A, do the following to open port 21:

15. In the Network and Sharing Center, click **Windows Firewall**. In the Windows Firewall window, click **Advanced settings**. In the left pane, click **Inbound Rules** and then click **New Rule** in the right pane. Create a new rule that opens the TCP local port 21.

On computer B, you should now be able to open an FTP session with computer A. Do the following:

16. Using the entries listed in step 8 and the IP address of computer A, open the session and verify you can see the contents of the shared folder. Quit the session and close the Command Prompt window.

Case Projects

Case Project 2-1: Create a VM and Install Ubuntu Desktop

In the case projects of Chapter 1, you created a virtual machine using Windows 8.1 Client Hyper-V or Oracle VirtualBox. In this case study, you create a second VM in your virtual network and install Ubuntu Desktop in the VM. In the Chapter 3, you'll install Ubuntu Server in your network.

Using the same computer that you used in Case Project 1-1 or 1-2 that has Client Hyper-V or Oracle VirtualBox installed, follow these steps:

1. Go to **ubuntu.com** and download the Ubuntu Desktop OS to your hard drive. The file that downloads is an ISO file. Ubuntu is a well-known version of Linux and offers both desktop and server editions.

2. Open the Oracle VM VirtualBox Manager or Hyper-V Manager. Following the directions in Chapter 1 case projects, create a new VM named VM2. Mount the ISO file that contains the Ubuntu Desktop download to a virtual DVD in your VM.
3. Start up the VM and install Ubuntu Desktop, accepting all default settings (see Figure 2-34). When given the option, don't install any extra software bundled with the OS. You'll need to restart the VM when the installation is done.

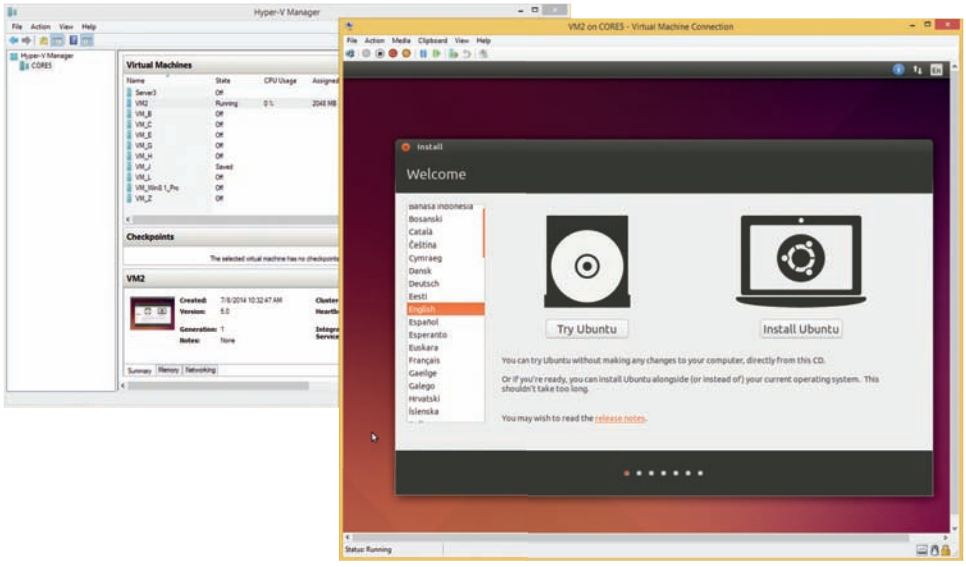


Figure 2-34 Ubuntu Desktop is installed in a VM in Windows 8.1 Client Hyper-V

Source: Microsoft LLC

4. To verify you have an Internet connection, open the Mozilla Firefox browser and surf the Web.

Good network technicians must know how to use many operating systems. Poke around in the Ubuntu Desktop interface and get familiar with it. What can you do with the Dashboard icon at the top of the left sidebar? You can also search the Web for tutorials and YouTube videos on how to use Ubuntu Desktop. When you're ready to shut down your VM, click the gear icon in the upper-right corner of the Ubuntu Desktop screen and click **Shut Down** in the menu that appears.

Case Project 2-2: Install and Use Wireshark

Wireshark is a free, open source network protocol analyzer that can help demystify network messages and help make the OSI model a little more tangible. Using Wireshark for the first time can be an epiphany experience for you. You can study the OSI layers, all of the information that is added to every message, and all of the messages that have to go back and forth just to bring up a Web page or even just to connect to the network. It all becomes much more real when you see how many packets Wireshark collects during even a short capture.

We'll install Wireshark in this project and take a first look at how it works. In later chapters, we'll dig deeper into Wireshark's capabilities.

1. To begin, go to the Web site at **wireshark.org**. Download and install the appropriate version for your OS.

You may also need to install WinPcap during the Wireshark installation process. WinPcap is a Windows service that does not come standard in Windows, but is required to capture live network data. You can keep the default setting presented in the Wireshark installer to start WinPcap at boot time, but consider unchecking this option if other, nonadministrative users of the computer should not have access to live network data.

2. To start our first capture, in the Wireshark Network Analyzer window, look in the Capture pane under the Start group and select your network interface. Then click **Start**. While the capture is running, challenge your network a bit by opening a couple of Web pages, sending an email with a local email client, or pinging other hosts on the network.
3. You can adjust the pane sizes by grabbing a border between them and dragging. Expand the top pane so you can see more of the captured packets at one time.
4. Let the capture run for a couple of minutes, and then click **Stop** on the command ribbon.

Take a look at some of the items you might have captured, and start to decode this blur of numbers and letters.

The color highlighting can help you begin to make sense of what's on the screen. Notice in Figure 2-35 that TCP messages are a light gray color, SMB2 packets are a yellowish color, and pnrp packets are a light bluish color. You can see the protocol names in the Protocol column.

No.	Time	Source	Destination	Protocol	Length	Info
387	115.279062	fe80::b99f:35be:2c3fe80::20c5:6548:7ba		SMB2	342	Create Response File: Jill West\Documents\2015 Net+
388	115.279349	fe80::20c5:6548:7bafe80::b99f:35be:2c3		SMB2	174	Notify Request File: Jill West\Documents
389	115.281175	fe80::20c5:6548:7bafe80::b99f:35be:2c3		SMB2	280	Find Request File: Jill West\Documents\2015 Net+ SMB2_FIND_ID_E
390	115.281329	fe80::b99f:35be:2c3fe80::20c5:6548:7ba		TCP	74	microsoft-ds > 51715 [ACK] Seq=1912 Ack=2302 Win=258 Len=0
391	115.283423	fe80::b99f:35be:2c3fe80::20c5:6548:7ba		TCP	1514	[TCP segment of a reassembled PDU]
392	115.283461	fe80::b99f:35be:2c3fe80::20c5:6548:7ba		TCP	1514	[TCP segment of a reassembled PDU]
393	115.283491	fe80::b99f:35be:2c3fe80::20c5:6548:7ba		SMB2	358	Find Response;Find Response, Error: STATUS_NO_MORE_FILES
394	115.287874	fe80::20c5:6548:7bafe80::b99f:35be:2c3		TCP	74	51715 > microsoft-ds [ACK] Seq=2302 Ack=4792 Win=258 Len=0
395	115.288260	fe80::20c5:6548:7bafe80::b99f:35be:2c3		SMB2	166	Close Request File: Jill West\Documents\2015 Net+
396	115.288611	fe80::b99f:35be:2c3fe80::20c5:6548:7ba		SMB2	202	Close Response
397	115.289396	fe80::b99f:35be:2c3fe80::20c5:6548:7ba		SMB2	151	Notify Response, Error: STATUS_PENDING
398	115.291304	fe80::20c5:6548:7bafe80::b99f:35be:2c3		TCP	74	51715 > microsoft-ds [ACK] Seq=2394 Ack=5281 Win=256 Len=0
399	115.351788	fe80::b99f:35be:2c3fe80::20c5:6548:7ba		pnrp	284	PNRP LOOKUP Message
400	115.353860	fe80::20c5:6548:7bafe80::b99f:35be:2c3		pnrp	96	PNRP AUTHORITY Message
401	115.354064	fe80::b99f:35be:2c3fe80::20c5:6548:7ba		pnrp	138	PNRP INQUIRE Message
402	115.358557	fe80::20c5:6548:7bafe80::b99f:35be:2c3		pnrp	1278	PNRP AUTHORITY Message [Malformed Packet]
403	115.359264	fe80::20c5:6548:7bafe80::b99f:35be:2c3		pnrp	1278	PNRP AUTHORITY Message
404	115.359336	fe80::20c5:6548:7bafe80::b99f:35be:2c3		pnrp	1278	PNRP AUTHORITY Message

Figure 2-35 Different highlight colors correspond to different protocols

5. To see a list of all colors used for highlighting that are currently assigned and to adjust these assignments, click the **Edit coloring rules** button. Here, you can change the priority for matching protocols to colors (because often more than one protocol is used in a single message), and you can assign colors that are easier to spot. In Figure 2-36, the assigned color for TCP is a bright green.

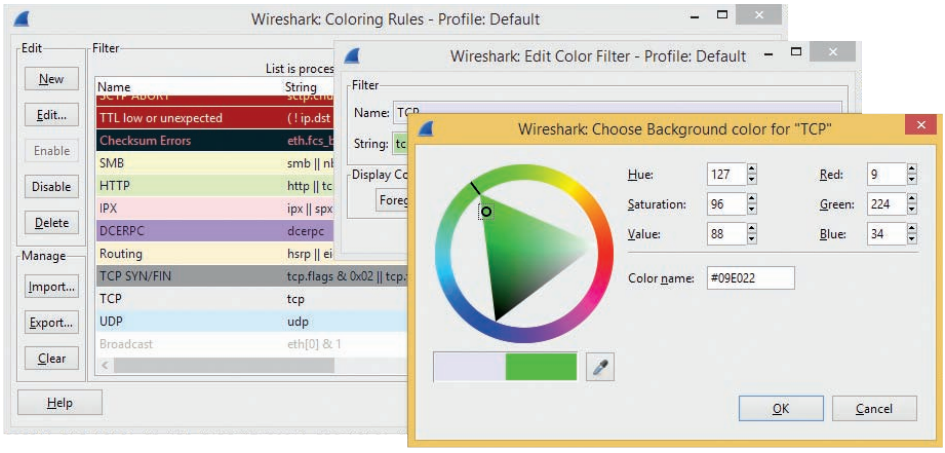


Figure 2-36 Choose colors that are easier to spot

Source: The Wireshark Foundation

6. To filter for a particular kind of packet, type the name of the protocol in the Filter box. Figure 2-37 shows Wireshark filtered for ICMPv6 packets. Try filtering for other protocols discussed earlier in this chapter and see how many different types you can find in your capture. Click **Clear** between searches to return to the complete capture data.

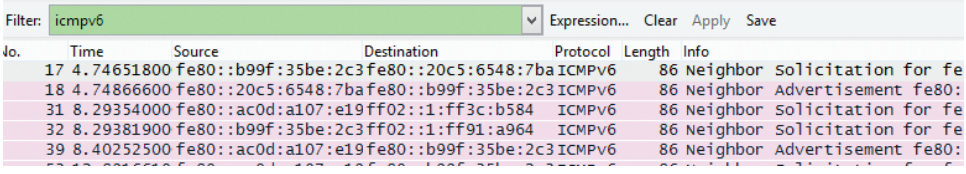


Figure 2-37 Use the filter to narrow your search

Source: The Wireshark Foundation

7. To compare OSI layers represented by each of these protocols, do a slightly more complicated filter where you can see both HTTP packets and ICMPv6 packets in the same search. Enter the following fields into the Filter box: **`http or icmpv6`**.
8. Look at an ICMPv6 packet and count how many sections of information are available in the middle pane. In Figure 2-38, there are four sections of information, which correspond to Layer 2 (Frame and Ethernet II) and Layer 3 (Internet Protocol Version 6 and Internet Control Message Protocol v6).
9. Examine an HTTP packet (in Figure 2-39, the labeled protocol is SSDP). In Figure 2-39, there are now five sections of information. This time, Layer 7 (Hypertext Transfer Protocol) and Layer 4 (User Datagram Protocol) are represented, in addition to Layer 3 (Internet Protocol Version 4) and Layer 2 (Ethernet II and Frame).

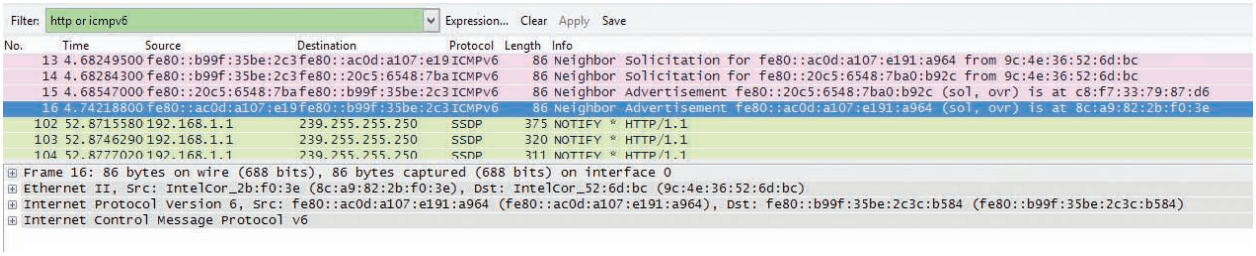


Figure 2-38 Use the middle pane to dig into each layer's headers

Source: The Wireshark Foundation

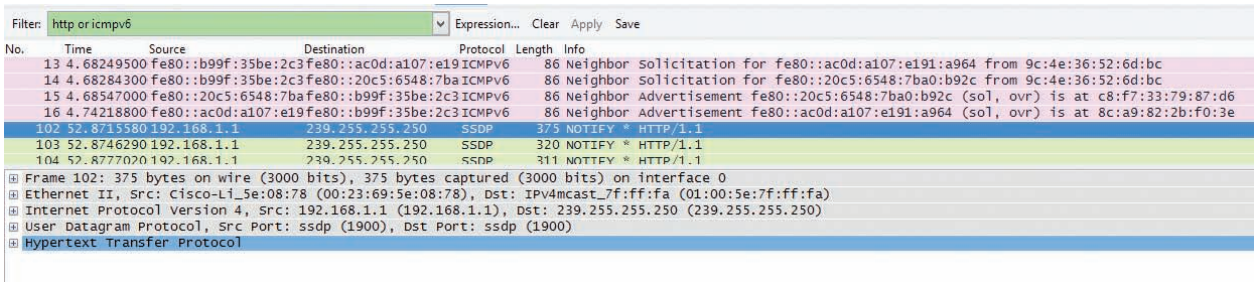


Figure 2-39 This HTTP message is using UDP

Source: The Wireshark Foundation

10. Recall that TCP is a connection-oriented protocol. You can filter a capture to follow a TCP stream so you can see how these messages go back and forth for a single session. Find a TCP packet, right-click it, and select **Follow TCP Stream**. Next, close the Follow TCP Stream window and note that Wireshark has filtered the capture for this stream's packets.
11. Select a TCP message from this filtered data, and explore the middle pane. Click to open each section in that pane. In Figure 2-40, Frame 229 is opened, and the list for the Flags bits is expanded. Notice that the Acknowledgment bit is set, which corresponds to the (ACK) flag on the packet Info in the top pane. You'll learn about these flags in the next chapter.

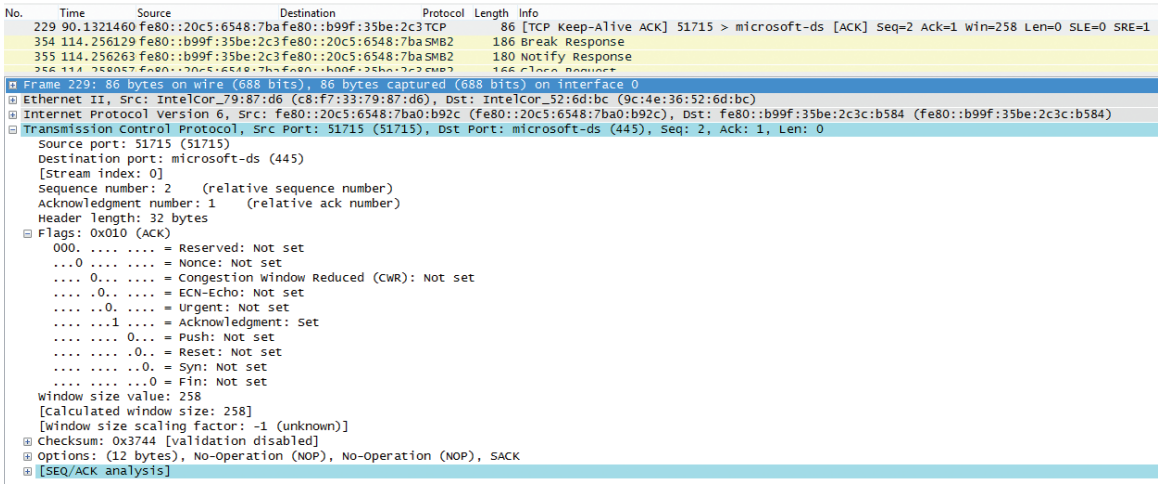


Figure 2-40 Other TCP segments might have other bits set

Source: The Wireshark Foundation

12. Click **Close this capture file** without saving the file. This returns you to the Wireshark home page, where you can open saved capture files, or you can look through sample captures. Click **Sample Captures** to go to the Wireshark wiki site where you can find samples of many different types of captures. Browse through some of these to become familiar with what to look for when examining different types of messages.