CHAPTER 1

Scaling IP Addresses

Objectives

Upon completion of this chapter, you should be able to answer the following questions:

- What methods are currently available to overcome the depletion of IPv4 address space?
- How can RFC 1918 address space be used to help conserve IP addresses?
- What types of NAT are currently available, and in which situations should they be deployed?
- What are the relative advantages of each type of NAT?
- What commands are used to configure, verify, and troubleshoot NAT operation?
- What is DHCP, and how does it differ from its predecessor, BOOTP?
- What commands are used to configure, verify, and troubleshoot DHCP operation?
- What is a helper address, and why is it required?

Key Terms

This chapter uses the following key terms. You can find the definitions in the glossary at the end of the book.

IPv4  page 3
ARPANET  page 3
IPv6  page 3
classful  page 3
classless  page 3
Variable-Length Subnet Mask (VLSM)  page 3
Network Address Translation (NAT)  page 4
inside network  page 5
outside network  page 5
local address  page 5
global address  page 5
inside local address  page 5
inside global address  page 5
translation  page 5
translation table  page 5
static NAT  page 6
static map  page 6
inside interface  page 7
outside interface  page 7
dynamic NAT  page 9
active translation  page 9
NAT pool  page 10
interesting traffic  page 10
Access Control List (ACL)  page 10
overloaded NAT  page 11
Port Address Translation (PAT)  page 11

continues
continued

source port  page 11

timeout value  page 14

process-switched  page 15

fast-switched  page 15

Dynamic Host Configuration Protocol (DHCP)  page 16

BOOTP  page 16

lease  page 17

DHCPDISCOVER  page 18

DHCPOFFER  page 18

DHCPREQUEST  page 18

DHCPACK  page 19

DHCPDECLINE  page 19

DHCPNAK  page 19

DHCPRELEASE  page 19

DHCP pool  page 19

binding  page 21

server farm  page 22

DHCP Relay  page 22

helper address  page 22

directed broadcast  page 24

logical broadcast  page 24

physical broadcast  page 24
When the current **IPv4** addressing scheme was developed, nobody anticipated the explosive growth that networking would undergo and the impact that this would have on the available IP address space. IPv4 was first standardized in 1981 by **ARPANET**. The decision to use 32-bit addresses and to assign them in large blocks resulted in a rapidly diminishing supply of available addresses. In addition, the recent incorporation of IP addresses into many aspects of everyday life, such as cell phones, has greatly contributed to this address space exhaustion.

Original expectations were that the IPv4 address space would be completely exhausted during the 1990s. This prompted the development of many different techniques to help alleviate the problem. The most ambitious endeavor to date has been the development of an entirely new addressing scheme known as **IPv6**. The development of IPv6 started in 1992. It increased the number of bits used to assign an address from the 32 used by IPv4 to 128. This change greatly increased the number of addresses available for allocation. The downside of IPv6 is that it requires additional equipment and configurations to fully implement. To date it has not been widely deployed.

Recent announcements by the U.S. government will compel all U.S. federal agencies to upgrade their network backbones by 2008. With these agencies forced to upgrade, government contractors, hardware and software vendors, and service providers will need to make certain that their offerings are also updated. This is expected to be the catalyst that will spark adoption of IPv6 in the commercial world.

The slow acceptance of IPv6 is partly due to the development of techniques to conserve the current IPv4 address space that have reduced the necessity for a rapid migration to the new system. One technique that has contributed greatly to the conservation of IPv4 addresses is the change from the **classful** system of addressing to the **classless** system. The wasteful assignment of equal-sized subnets encountered in the classful system is eliminated in the classless system, where the assignment of addresses can be more closely structured to the requirements of the individual network segment. This is accomplished through the use of a **Variable-Length Subnet Mask (VLSM)**, which essentially allows a subnet to be subnetted even further. The migration to the classless system has allowed many large pools of addresses to be returned for reallocation and has decreased the urgency of migrating to IPv6.

Other techniques for address conservation have also been developed:

- RFC 1918 address space
- Network Address Translation (NAT)
- Dynamic Host Configuration Protocol (DHCP)

These techniques can be deployed either independently or in combination and are discussed in this chapter.
Scaling Networks with NAT and PAT

In most current network installations, Network Address Translation (NAT) is deployed along with private address space to allow the network administrator to efficiently and securely manage the corporate IP resources. This combination not only offers flexibility in the distribution of internal address space, but also contributes to the preservation of the current IPv4 address space.

Private Address Space (RFC 1918)

IETF RFC 1918 sets aside three blocks of IP addresses that are reserved for private use only and will not be assigned to any one individual or organization. Table 1-1 summarizes the RFC 1918 address space. Because these address blocks will remain unregistered, any organization is free to use them on its internal networks. Properly configured routers will not advertise this address space to the Internet, and packets that contain these addresses will be dropped as soon as they hit the Internet. The use of this address space on an organization’s internal network eliminates the waste and expense of maintaining large pools of routable addresses and allows the network administrator to design an IP addressing scheme that reflects the organization’s structure.

Table 1-1   RFC 1918 Address Space

<table>
<thead>
<tr>
<th>Class</th>
<th>Range</th>
<th>CIDR Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10.0.0.0 to 10.255.255.255</td>
<td>10.0.0.0/8</td>
</tr>
<tr>
<td>B</td>
<td>172.16.0.0 to 172.31.255.255</td>
<td>172.16.0.0/12</td>
</tr>
<tr>
<td>C</td>
<td>192.168.0.0 to 192.168.255.255</td>
<td>192.168.0.0/16</td>
</tr>
</tbody>
</table>

Network Address Translation (NAT)

Deploying RFC 1918 addresses within an organization presents an interesting challenge. Because these addresses cannot be routed on the Internet, any packets that leave the internal network must have their source address replaced by one that is routable. Replies addressed to this routable address must then have the destination address translated back into the original private address before being placed on the internal network. This process of rewriting the Layer 3 information as a packet moves between the internal and external networks is known as Network Address Translation (NAT). NAT, as discussed in IETF RFC 1631, allows an organization to present itself to the Internet with fewer IP addresses than there are nodes on the internal network. NAT usually maps an unregistered address used on an internal network to a registered address for use on the Internet, but it can also map between networks deploying routable or nonroutable addresses for certain applications. This chapter covers three forms of NAT:

- Static NAT
- Dynamic NAT
- Overloaded NAT or Port Address Translation (PAT)
NAT Terminology

Understanding the process of NAT and being able to troubleshoot it as issues arise requires a detailed understanding of the terminology used. NAT divides the network into two areas: inside and outside. The inside network is usually an organization’s internal LAN, and the outside network is usually the Internet, although it could be any other network. Local addresses refer to a node on the network as seen by another node on the same network. Global addresses are how a node on one network is seen by a node on another network. For example, how a machine on the LAN appears to another machine on the same LAN is an inside local address, whereas that same machine would be seen by the inside global address from the outside world. A translation is the rewriting of information in the packet header from one IP address to another as the packet passes between the internal and external networks. The IOS maintains a translation table in RAM that tracks current translations. NAT simply changes the source IP address or port numbers on outbound traffic and the destination IP address or port number on inbound traffic. Figure 1-1 shows the inside local address being replaced with the inside global address on packets leaving the internal network. Then the inside global address is translated back to the inside local address in the replies before they are placed on the internal network.

Figure 1-1   NAT Translation
Understanding and Configuring Static NAT

One form of NAT that is commonly implemented is known as static NAT. In this scenario, the network administrator manually configures a predefined one-to-one mapping of addresses between the internal and external networks. Static NAT has no conservation of IP address space, because each internal IP address must be mapped to a unique, routable external address. This mapping of addresses, illustrated in Figure 1-2, ensures that no packets are dropped due to lack of available address space. It also minimizes the delay introduced by building a dynamic translation.

Figure 1-2  Static NAT

<table>
<thead>
<tr>
<th>Source Address</th>
<th>Inside Local</th>
<th>Inside Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.20.20.2</td>
<td>10.20.20.2</td>
<td>209.165.200.227</td>
</tr>
<tr>
<td>10.20.20.3</td>
<td>10.20.20.3</td>
<td>209.165.200.228</td>
</tr>
<tr>
<td>10.20.20.4</td>
<td>10.20.20.4</td>
<td>209.165.200.229</td>
</tr>
</tbody>
</table>

Static NAT is often deployed together with dynamic NAT, which is discussed in the next section. Static NAT is usually deployed to allow access to internal servers from the outside world. Servers that must be accessed from the outside are assigned an IP address consistent with that deployed on the internal network. Then a static map is created that maps a routable IP address to the internal address. This allows the server to be accessed using the inside local address from within the organization and also from outside the LAN using the inside global address. If the location of the internal server is changed, a new static map is created, making the change transparent to external users. In all forms of NAT, the hiding of internal address space from the external world provides a limited level of security in that no direct access to the internal network from the outside world is possible.
To configure static NAT, the inside and outside interfaces must be identified. Usually the Ethernet or LAN port is the *inside interface* and the serial or WAN port is the *outside interface*. This must be clearly stated, because all translations are done in the specified direction only.

Figure 1-3 shows a typical assignment of inside and outside interfaces, and Example 1-1 provides the corresponding configuration. Inside and outside interfaces are defined using the `ip nat inside` and `ip nat outside` commands at the appropriate interface configuration prompt.

**Figure 1-3**   Multiple Inside Interfaces with NAT
As soon as the inside and outside interfaces are established, the direction of translation and the actual translation are defined. The translation is an inside local address to an inside global address, as shown in the following syntax:

```
Router(config)# ip nat inside source static inside_local inside_global
```

For example, if you wanted to create three static translations to internal servers, as illustrated in Figure 1-2, you would issue the following commands:

```
Toronto(config)# ip nat inside source static 10.20.20.2 209.165.200.227
Toronto(config)# ip nat inside source static 10.20.20.3 209.165.200.228
Toronto(config)# ip nat inside source static 10.20.20.4 209.165.200.229
```

The NAT table in Example 1-2 clearly shows the mapping of the inside local address to the inside global address.

```
Pro Inside global     Inside local      Outside local       Outside global
-- 209.165.200.227    10.20.20.2         --                --
-- 209.165.200.228    10.20.20.3         --                --
-- 209.165.200.229    10.20.20.4         --                --
```
Depending on the deployment scenario, it may be desirable to NAT the inside source, outside source, or inside destination addresses. This may be accomplished by modifying the commands shown in Table 1-2.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip nat inside source</td>
<td>Enables NAT of the inside source addresses</td>
</tr>
<tr>
<td>ip nat outside source</td>
<td>Enables NAT of the outside source addresses</td>
</tr>
<tr>
<td>ip nat inside destination</td>
<td>Enables NAT of the inside destination addresses</td>
</tr>
</tbody>
</table>

Lab 1-3 Configuring Static NAT Addresses (1.4.1c)

In this lab, you configure a router to use NAT to convert internal IP addresses, typically private addresses, into outside public addresses. You configure static mapping to allow outside access to an internal host.

Understanding and Configuring Dynamic NAT

Dynamic NAT eliminates the requirement for equal numbers of internal and external addresses by creating a pool of IP addresses that can be used for translation. Dynamic NAT does not allow for the assignment of a predefined address between the internal and external networks. Therefore, it is of little use for mapping internal servers to the outside world. This is why it is often deployed together with static NAT. Dynamic NAT does not guarantee that a translation will be successful if the pool of addresses is exhausted. Therefore, you must take care to provide an address pool of sufficient size to handle all possible simultaneous translations.

The most common use of dynamic NAT is to provide Internet connectivity to the employees in an organization that deploys RFC 1918 address space on the internal network. Because not all employees will require Internet access simultaneously, a pool of registered addresses smaller than the number of internal addresses can be configured. This reduces the expense of maintaining registered addresses for all employees, as would be required in a purely static NAT deployment.

Addresses are translated as packets leave the internal network. Translations are active only for the duration of the conversation or a configurable time period, whichever occurs first, as illustrated in Figure 1-4. After the reply is received, the translation is dissolved, and the address is returned to the NAT pool for reuse. The router maintains a record of active translations in the form of a translation table. Each active translation consumes approximately 160 bytes of DRAM. This translates to approximately 1.6 MB of DRAM for 10,000 active translations. Modern routers have more than enough memory to support thousands of NAT translations.
To create the pool of addresses used in dynamic NAT, use the following global configuration command:

```
ip nat pool name start-ip-address end-ip-address {netmask netmask | prefix-length prefix-length} {type rotary}
```

Giving the *NAT pool* a name that describes the location or task makes troubleshooting easier if multiple pools are to be configured.

In the example pictured in Figure 1-4, if you wanted to use the 209.165.202.129/27 subnet as a pool of addresses for dynamic NAT on the Toronto router, you would issue the following command:

```
```

It is neither necessary nor advisable to translate all traffic found on the network, because this would waste valuable router resources. Traffic that should be translated is termed *interesting traffic* and is selected with the aid of an access control list (ACL). NAT does not translate traffic not considered interesting as defined by the ACL.

If you wanted to allow only traffic originating on the 10.20.20.0/24 subnet of the Toronto network, you could use the following ACL to select the interesting traffic:

```
Toronto(config)#access-list 7 permit 10.20.20.0 0.0.0.255
```
The ACL is then tied to the NAT pool. The direction of translation is defined by the following global configuration command:

```
ip nat inside source {list {access-list-number | name} pool name [overload] | static local-ip-address global-ip-address}
```

Continuing with our example, you would tie the ACL to the NAT pool using the following command:

```
Toronto(config)#ip nat inside source list 7 pool border
```

This command checks the source address on packets entering an interface that has been defined as an inside interface. If the source address matches the specified ACL, the traffic is identified as interesting, and the address is translated to an address found in the NAT pool before the traffic is moved out through the outside interface. If the source address does not match the ACL, the packet is not translated. A record of the translation is maintained in the NAT table. The router uses it to translate the IP address of replies before they are placed on the internal network.

### Lab 1-1 Configuring NAT (1.1.4a)

In this lab, you configure a router to use NAT to convert internal IP addresses into outside public addresses.

### Understanding and Configuring Overloaded NAT

A modified form of dynamic NAT is *overloaded NAT*, which is also known as *Port Address Translation (PAT)*. With overloaded NAT, many addresses can be mapped to a single IP address using port numbers. This further reduces the number of addresses required by an organization to provide Internet connectivity to its users. Most large organizations now deploy PAT with a pool of IP addresses.

PAT divides the available ports per global IP address into three ranges of 0–511, 512–1023, and 1024–65535. When a translation is requested, PAT tries to preserve the *source port* on the global IP address. If the source port is unavailable, PAT begins searching from the beginning of the particular port range to find the first available port. If no ports are available in the selected range, the packet is dropped.

PAT can also be supplied with a pool of addresses to use. If PAT is requested and a pool of IP addresses has been configured, PAT first attempts to maintain the original source port on the first available global IP address. If this port is unavailable, PAT starts searching from the beginning of the relevant port range for an available port. If no ports are available in the relevant port range, PAT checks the second global IP address, trying to maintain the original source port. If the original port is unavailable, PAT again searches from the beginning of the relevant port range for an available port. This process is repeated until a new source port-IP address can be assigned or until PAT runs out of IP addresses to search. If PAT cannot find an available IP address-source port, it drops the packet. Figure 1-5 shows how PAT works.
The configuration of PAT is nearly identical to the configuration of dynamic NAT, except that the keyword `overload` must be specified when setting up the translation. This keyword allows NAT to reassign ports on the source address while preserving the original IP address. It is also possible to overload an interface’s IP address, commonly the WAN interface, instead of supplying a separate pool of addresses. Because the WAN port of most border routers is configured with an IP address that has been supplied by the ISP, using this address eliminates the requirement of obtaining a new IP address. The problem with this approach is that only one IP address is available, so if many translations must be made, one address may not suffice. To overload either a pool of addresses or the address assigned to an interface, use one of the following global commands:

```
ip nat inside source list acl_list_number pool pool_name overload
nip nat inside source list acl_list_number interface interface_name overload
```

Assume that you want to use the single address 209.165.201.2 to provide connectivity to the Internet. To accomplish this, you would need to define both the inside and outside interfaces, create an ACL to select interesting traffic, create a pool of addresses, and then apply the translation. It is possible to create an address pool that contains only one address by specifying the same address for both the beginning and ending address in the pool. Here are the final two steps:

```
Toronto(config)#ip nat pool border2 209.165.201.2 209.165.201.2 netmask 255.255.255.224
Toronto(config)#ip nat inside source list 7 pool border2 overload
```

If you decided to overload the IP address associated with the outside interface instead of creating a separate pool, the command would simply be as follows:

```
Toronto(config)#ip nat inside source list 7 interface serial 0 overload
```
In this lab, you configure a router to use PAT to convert internal IP addresses into outside public addresses.

Verifying NAT/PAT Functionality

A good place to start to verify NAT functionality or to troubleshoot a problematic implementation is with the `show ip nat statistics` command. This command provides information such as the number of packets that matched the ACL and were translated versus those that did not match the ACL, the location of the inside and outside interfaces, and the number of currently active translations. Quite often, this points to a problem with either the ACL or the assignment of interfaces. Example 1-3 provides sample output from the `show ip nat statistics` command.

<table>
<thead>
<tr>
<th>Example 1-3 Viewing NAT Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toronto# <code>show ip nat statistics</code></td>
</tr>
<tr>
<td>Total active translations: 0 (0 static, 0 dynamic; 0 extended)</td>
</tr>
<tr>
<td>Outside interfaces:</td>
</tr>
<tr>
<td>Serial 0</td>
</tr>
<tr>
<td>Inside interfaces:</td>
</tr>
<tr>
<td>FastEthernet0</td>
</tr>
<tr>
<td>Hits: 47  Misses: 3</td>
</tr>
<tr>
<td>Expired translations: 50</td>
</tr>
<tr>
<td>Dynamic mappings:</td>
</tr>
<tr>
<td>-- Inside Source</td>
</tr>
<tr>
<td>[Id: 1] access-list 7 pool toronto2 refcount 0</td>
</tr>
<tr>
<td>pool toronto2: netmask 255.255.255.224</td>
</tr>
<tr>
<td>start 209.165.201.2 end 209.165.201.2</td>
</tr>
<tr>
<td>type generic, total addresses 1, allocated 0 (0%), misses 0</td>
</tr>
</tbody>
</table>

To view translations that are currently active, use the `show ip nat translations` command. This displays the contents of the translation table, allowing you to determine the correctness of the translations that are occurring. An improperly configured address pool could be handing out addresses outside of the desired range and impacting network accessibility. The addition of the keyword `verbose` to this command provides additional information such as how long the translation has been active. This can help diagnose issues related to network connectivity and improperly configured timeout values. Remember that NAT relies on an ACL to select traffic that is to be translated. By checking to see which packets matched the ACL, you can determine if any problems encountered are with the NAT translation process or merely the selection of interesting traffic by an improperly configured ACL. Example 1-4 provides sample output from the `show ip nat translations` command.
You can configure the length of time a translation remains active and stays in the translation table. Normally the translation is released when a reply is received, at which time the IP address is returned to the NAT pool. If a reply is not received, the entry stays in the table until it is removed by the timeout value or is manually removed by the network administrator. By default this timeout is set to 24 hours for TCP and 5 minutes for UDP. During the time that a translation remains in the NAT table, no other translation can use the IP address. If network delays or other issues arise that prevent the NAT process from receiving the reply, it is quite possible to run out of IP addresses that can be used for translations. This prevents users from accessing the Internet. Careful consideration of the timeout value is most important in dynamic NAT, which has a limited number of addresses available. In static NAT the mapping is predetermined, and PAT usually has large numbers of potential translations available.

You can configure a number of different timeout values in global mode to affect how NAT functions. Table 1-3 provides the commands necessary to set these timeout values along with the default values.

### Table 1-3 Setting NAT Timeout Values

<table>
<thead>
<tr>
<th>Command</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip nat translation timeout time</td>
<td>24-hours</td>
</tr>
<tr>
<td>ip nat translation tcp-timeout time</td>
<td>24-hours</td>
</tr>
<tr>
<td>ip nat translation udp-timeout time</td>
<td>5-minutes</td>
</tr>
<tr>
<td>ip nat translation finrst-timeout time</td>
<td>60-seconds</td>
</tr>
<tr>
<td>ip nat translation dns-timeout time</td>
<td>60-seconds</td>
</tr>
</tbody>
</table>

To remove a translation that is currently active, you must clear it from the NAT table. The command used to remove an active translation from the NAT table depends on the type of translation that has occurred:

- **clear ip nat translation** * clears all dynamic NAT translations.
- **clear ip nat translation inside global_ip local_ip** (outside local_ip global_ip) removes a single dynamic translation entry.
clear ip nat translation protocol inside global_ip global_port local_ip local_port
{outside local_ip local_port global_ip global_port} removes the extended dynamic entry.

To verify that the correct translations are occurring, use the debug ip nat [detailed] command. The output from this command, as shown in Example 1-5, provides information detailing the translation of packets as they occur. In this output the source address is specified by an s, and d identifies the destination IP address. An arrow (->) indicates a translation that has occurred.

Example 1-5  Debugging NAT Translations
Toronto#debug ip nat
*Mar 1 14:01:01.620: NAT*: s=209.165.202.130, d=209.165.201.2->10.20.20.1  [3926]
*Mar 1 14:01:01.624: NAT*: s=209.165.202.130, d=209.165.201.2->10.20.20.1  [3927]
*Mar 1 14:01:01.628: NAT*: s=209.165.202.130, d=209.165.201.2->10.20.20.1  [3928]
*Mar 1 14:01:01.632: NAT*: s=209.165.202.130, d=209.165.201.2->10.20.20.1  [3929]
*Mar 1 14:01:01.632: NAT*: s=209.165.202.130, d=209.165.201.2->10.20.20.1  [3930]

The first packet in any conversation must be process-switched, which establishes the translation entry. As soon as an entry exists in cache, any remaining packets in the conversation may take the fast-switched path. Packets that have taken the fast-switched path are flagged with an asterisk in the debug output.

NAT has contributed greatly to the conservation of existing address space by allowing the privatization of internal networks. It also allows an organization’s internal network configuration to remain unchanged when moving from one service provider to another. Furthermore, it provides limited security by hiding the internal address space from the outside world.

Unfortunately, applications that rely on end-to-end IP addressing for their functionality do not work across a NAT router. If the remote IP address is contained in a packet’s data portion instead of the header, NAT cannot occur. This is because NAT/PAT examines only the header. Any application that uses physical addresses instead of a qualified domain name will not reach destinations across a NAT router. This problem can sometimes be overcome using static mappings. The following applications do not always function across a NAT router, although methods exist to allow them to function:

- Routing table updates
- DNS zone transfers
- BOOTP
- Talk, ntalk

Caution
Exercise caution when clearing NAT translations on a production network. Clearing an active NAT translation forces the session to be restarted.
Simple Network Management Protocol (SNMP)

VoIP

Lab 1-4 Verifying NAT and PAT Configuration (1.1.5)

In this lab, you configure a router for NAT and PAT, test the configuration, and verify NAT/PAT statistics.

Lab 1-5 Troubleshooting NAT and PAT (1.1.6)

In this lab, you configure and troubleshoot a router for NAT and PAT.

Dynamic Host Configuration Protocol (DHCP)

Before the advent of *Dynamic Host Configuration Protocol (DHCP)*, all hosts on a network had to be manually configured with a separate, unique IP address before they could use network resources. Because of the static nature of the assignments, it was not possible to reuse any addresses, and any changes in network configuration required the network administrator to manually reconfigure each device. DHCP allows the automatic assignment of network configuration information to hosts on a network. With DHCP, addresses are assigned when required and are returned to a common pool when not in use. The DHCP protocol is described in RFC 2131 and is a successor to BOOTP.

*BOOTP*, as explained in RFC 951 and RFC 1542, was designed to enable diskless workstations to boot and request a minimal network configuration based on their MAC address. A BOOTP server is configured with a table of MAC addresses and the corresponding IP address that the administrator wants to assign. The server matches the Layer 2 address in the request to the Layer 3 address, and then this is sent back to the client in a reply. With BOOTP the same IP address is always handed to a workstation. This permanent assignment of IP addresses, static mappings, and a limit of four configuration parameters restricted BOOTP’s usefulness and paved the way for a more versatile technique to dynamically configure network parameters.

DHCP reduces Internet access costs when NAT/PAT is not deployed by allowing address space to be dynamically assigned and reused. It also minimizes the time and expense of client configuration and provides a means for centralized management of IP information. There are many reasons why a network administrator may decide to run DHCP on a router rather than on a server-based system. Running DHCP on a router eliminates the need for extra hardware and allows DHCP to be run locally even if WAN connectivity is lost. On the downside, this decentralized method of running DHCP can complicate management and troubleshooting. Running a
DHCP server on a router forces a network administrator to use the Cisco IOS commands instead of a graphical configuration tool such as that provided by Windows. It also adds another service on a device that may at times already be overburdened.

DHCP provides a mechanism by which a client can be assigned an IP address for a limited amount of time known as the *lease* period. This allows a client to be assigned another address when it moves to a different subnet. It also lets the DHCP server assign the address to another client when the first client no longer requires it. This address reuse is a major factor in the conservation of IP addresses. If, at the end of the lease period, the client still requires the use of the same networking information, it may request a renewal of the lease from the DHCP server.

DHCP additionally provides a method by which a client can gather other IP configuration information, such as the location of WINS and DNS servers, along with a domain name. The Cisco IOS implementation of DHCP supports more than 30 configuration parameters.

DHCP is not always possible or desirable. Servers on a network often require a static address, and some network administrators still prefer to statically assign network addresses to simplify management and security. In addition, the DHCP server must be available whenever an address is required, and end users must be prevented from manually configuring their network settings and potentially generating a conflict with dynamically assigned addresses.

DHCP supports three mechanisms for address allocation:

- **Automatic allocation** assigns a permanent address to the client.

- **Dynamic allocation** assigns an IP address for a fixed period of time or until the client relinquishes the address.

- With **manual allocation**, the administrator assigns an IP address to the client, and DHCP is used to convey the assigned address to the client.

DHCP, like its predecessor, BOOTP, uses UDP port 67 for requests and UDP port 68 for replies. The DHCP request is sent out as both a Layer 2 and Layer 3 broadcast for the destination but contains the client’s MAC address as the source. The request asks the server to supply a client IP address and mask. The server responds to the request with a unicast message supplying the requested information. If the client is on the same segment as the DHCP server, the gateway IP address field is set to 0, and the IP address of the receiving interface is used to determine the pool of addresses that will be used. If the gateway IP address field is not 0, it is used to determine the subnet of the pool of addresses that should be distributed. In the reply, the destination MAC is the client’s MAC address, and the destination IP address is one that has been supplied by the DHCP server. The Layer 2 and Layer 3 source address for the reply is that of the DHCP server. The client accepts this information and configures its network settings accordingly, as shown in Figure 1-6.
Properly troubleshooting the DHCP process requires a deeper understanding of how DHCP works. Figure 1-7 shows the complete process by which a DHCP configuration is requested:

1. The first step in the process is when the client machine issues a broadcast **DHCPDISCOVER** message to locate a DHCP server on the network.

2. Any DHCP servers that hear the message generate a unicast **DHCPOFFER** that contains configuration information such as an IP address, lease period, and domain name. If multiple DHCP servers are available on the network, the client may receive more than one DHCPOFFER. The client evaluates the offer and decides which one to accept. Usually the client accepts the first offer it receives. This is only an offer of configuration information at this point; the server has not assigned the information to the client.

3. The server usually reserves the information for the client until it has a chance to receive a **DHCPREQUEST** broadcast message from the client or it learns that the client has accepted configuration information from another server. The DHCPREQUEST is sent as a broadcast so that other DHCP servers on the network will not wait to hear from the client and will reclaim the addresses they had offered.
4. The server confirms that the offered information is still available and has been allocated to the client by issuing a **DHCPACK** unicast message to the client.

**Figure 1-7** DHCP Process

If the DHCPOFFER is seen as invalid, the client refuses the offer by issuing a **DHCPDECLINE** broadcast message. If the client is slow in responding to the server, or an error occurs in the negotiation of parameters, the server issues a broadcast **DHCPNAK**. When the client no longer requires the IP address, it issues a **DHCPRELEASE** message telling the server that it may return the address to the pool for reassignment.

**Configuring DHCP**

By default the IOS DHCP service is enabled on routers. To enable the service if it has been disabled, use the global configuration command **service dhcp**. On larger networks, DHCP services are usually handled by one or more dedicated servers that provide a more robust solution than can be implemented on a single router. For security reasons, it’s a good idea to disable all services on a network device except those required for network operability. Use the global command **no service dhcp** to disable the DHCP service on a router.

After the service is enabled on a router, you must create a pool or pools of addresses using the following command:

```
Router(config)#ip dhcp pool name
```

Troubleshooting is much easier if the name chosen for the **DHCP pool** is meaningful within the context of the network or organization.

After the address pool is defined, you must specify the addresses it contains. You can specify the network or subnet range available for the address pool by a network subnet address followed by either a valid subnet mask or the prefix length, as shown here:

```
Router(dhcp-config)#network network-number [mask | prefix-length]
```

From this point, you can specify a number of optional configuration parameters, including the domain name, the location of the DNS and NetBIOS name servers, the default gateway (default-router), and the lease’s duration. For the location of the DNS server, NetBIOS name
server, and default gateway, you can specify up to eight addresses, but only one is required. The default lease period is 24 hours. These parameters are configured in DHCP configuration mode, as shown here:

```
Router(dhcp-config)#domain-name domain
Router(dhcp-config)#dns-server address [address2 ... address8]
Router(dhcp-config)#netbios-name-server address [address2 ... address8]
Router(dhcp-config)#netbios-node-type type
Router(dhcp-config)#default-router address [address2 ... address8]
Router(dhcp-config)#lease {days [hours][minutes] | infinite}
```

One last step in DHCP configuration is the removal of addresses from the DHCP pool that have been used on the network or that should be reserved for other uses. This includes addresses assigned to servers and the default gateway, along with any static pools that are required. You can remove IP addresses from the DHCP pool using the following global configuration command:

```
Router(config)#ip dhcp excluded-address low-address [high-address]
```

Example 1-6 provides a sample DHCP configuration for the network illustrated in Figure 1-8. The configuration gives the DHCP clients connected to the LAN segment of the border router an IP address from the 10.20.30.0/24 subnet. It also sets the domain name to cisco.com and the default gateway to 10.20.30.1. The first ten addresses of this subnet are excluded and are not assigned to any clients.

---

**Example 1-6    Sample DHCP Configuration**

```
Border (config)#ip dhcp excluded-address 10.20.30.1 10.20.30.10
Border (config)#interface FastEthernet 0/0
Border (config-if)#ip address 10.20.30.1 255.255.255.0
<commands omitted>
Border (config)#ip dhcp pool border
Border (config)#network 10.20.30.0 255.255.255.0
Border (config)#default-router 10.20.30.1
Border (config)#domain-name cisco.com
```

---

**Figure 1-8    DHCP Implementation**

DHCP Client

DHCP Client

DHCP Client

DHCP Server

Border

Internet
Lab 1-6 Configuring DHCP (1.2.6)

In this lab, you configure a router for DHCP to dynamically assign addresses to attached hosts.

DHCP Verification and Troubleshooting

DHCP binds an IP address to a MAC address. You can obtain information on the binding using the following command:

Router#show ip dhcp binding [address]

Example 1-7 shows a sample binding of MAC to IP addresses. This output also shows how the addresses were assigned and when the lease expires.

<table>
<thead>
<tr>
<th>IP address</th>
<th>Hardware address</th>
<th>Lease expiration</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.20.20.7</td>
<td>0100.e098.97d9.e7</td>
<td>Mar 02 1993 07:09 PM</td>
<td>Automatic</td>
</tr>
<tr>
<td>10.20.20.8</td>
<td>0100.10a4.e411.6f</td>
<td>Mar 02 1993 07:09 PM</td>
<td>Automatic</td>
</tr>
</tbody>
</table>

Bindings normally are removed at the end of the lease period unless the client renews the lease. To remove bindings before the lease expires, use the following command:

Router#clear ip dhcp binding {address | *}

To view the binding process in real time, use the following command:

Router#debug ip dhcp server {events | packets | linkage}

Example 1-8 shows sample output for a debugging session in which two hosts have been assigned IP addresses.

<table>
<thead>
<tr>
<th>Time</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Mar 1 19:14:36.815</td>
<td>DHCPD: assigned IP address 10.20.20.8 to client 0100.10a4.e411.6f.</td>
</tr>
</tbody>
</table>
Detailed information on the DHCP messages that the DHCP server has seen and replied to can point to improperly configured clients or loss of network connectivity. If the server is not seeing the DHCPREQUEST message, it cannot respond. If DHCPREQUEST messages are being seen, but the server is not responding, this indicates that the service is not properly configured. This information can be obtained from the command `show ip dhcp server statistics`, as shown in Example 1-9.

### Example 1-9  Viewing DHCP Server Statistics

```
Toronto#show ip dhcp server statistics
Memory usage         14468
Address pools        1
Database agents      0
Automatic bindings   2
Manual bindings      0
Expired bindings     0
Malformed messages   0

Message             Received
BOOTREQUEST          0
DHCPDISCOVER         18
DHCPREQUEST          23
DHCPDECLINE          0
DHCPRELEASE          16
DHCPINFORM           6

Message             Sent
BOOTREPLY            0
DHCPOFFER            18
DHCPACK              26
DHCPNAK              0
```

### DHCP Relay

In an enterprise environment, it is not uncommon to place all servers on one segment of the network and have users on other segments. Figure 1-9 shows an enterprise network that has collected all servers on one segment to form a **server farm**. Even if DHCP is running on a router, the network administrator may choose to use only one router in the organization as a DHCP server to simplify management. Because routers normally block broadcasts, they must be told to allow the DHCP request to come through and reach the DHCP server. This **DHCP Relay** is accomplished by implementing a **helper address**.
A helper address accepts the broadcast on one interface and turns it into a unicast. The `ip helper-address` command must be applied on the interface on which the broadcast would be received. The IP address specified is that of the server that the broadcast should be able to reach. The syntax of this command is as follows:

```
ip helper-address address
```

Specifically, in Figure 1-9, the command would be applied to the FastEthernet 0/0 interface of the Toronto router as follows:

```
Toronto(config-if)#ip helper-address 10.20.30.11
```

This forwards the DHCP broadcasts originating on the FastEthernet 0/0 segment of the network to the DHCP server located at 10.20.30.11.

By default, this command forwards eight UDP services. If this is not the desired result, these services must be disabled individually. Table 1-4 lists the services forwarded by helper addresses.

**Table 1-4**  Default UDP Services Forwarded by a Helper Address

<table>
<thead>
<tr>
<th>Service</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>37</td>
</tr>
<tr>
<td>TACACS</td>
<td>49</td>
</tr>
<tr>
<td>DNS</td>
<td>53</td>
</tr>
<tr>
<td>BOOTP/DHCP server</td>
<td>67</td>
</tr>
<tr>
<td>BOOTP/DHCP client</td>
<td>68</td>
</tr>
<tr>
<td>TFTP</td>
<td>69</td>
</tr>
<tr>
<td>NetBIOS name service</td>
<td>137</td>
</tr>
<tr>
<td>NetBIOS datagram service</td>
<td>138</td>
</tr>
</tbody>
</table>
If it is desirable to forward additional ports, you can use the global `ip forward-protocol` command. To minimize the impact of unnecessary traffic on a segment, it might be desirable to disable forwarding of certain ports. This can be accomplished using the global `no ip forward-protocol` command. The following shows the syntax for both commands:

```
ip forward-protocol {udp [port] | nd | sdns}
no ip forward-protocol {udp [port] | nd | sdns}
```

The `ip forward-protocol` command allows the forwarding of UDP ports, the Network Disk (nd) protocol used by some older diskless Sun workstations, and Secure Data Network Service (sdns).

Although security best practices do not recommend doing so, it is possible to use a logical broadcast address as the destination address in the `helper-address` command. To do so, you must enable `directed broadcasts` on the exit interface using the `ip directed-broadcast` command:

```
ip directed-broadcast [access-list-number] | [extended access-list-number]
```

When this command is used, the router converts the Layer 3 `logical broadcast` address into a Layer 2 `physical broadcast` with an address of FF-FF-FF-FF-FF-FF. This combination of `helper-address` and `directed-broadcast` commands lets the router accept a broadcast on one segment and send it to another segment as a directed broadcast. This allows all machines on the destination segment to hear and respond to the request as appropriate.

DHCP is an extremely valuable tool in a network environment that may be run on either an IOS router or a server, depending on the organization’s needs. It is easy to implement and troubleshoot and greatly simplifies the job of maintaining up-to-date network configurations on networked machines.

**Lab 1-7 Configuring DHCP Relay (1.2.8)**

In this lab, you configure a router for DHCP, add the ability for workstations to remotely obtain DHCP addresses, and dynamically assign addresses to the attached hosts.
Summary

Many techniques have been developed to help conserve the diminishing supply of IPv4 addresses. The most notable change has been the adoption of the classless system of IP addressing and the implementation of VLSM to minimize waste in the allocation of address space that was generated by the classful system.

Within individual organizations, the implementation of private address space on the internal network and the use of NAT to allow packets to travel to the Internet have reduced the requirements of maintaining large pools of registered addresses. Of the many different forms of NAT available, most enterprise network managers deploy PAT to give internal users access to the Internet in combination with static NAT to give outside users access to internal resources. In addition, the implementation of DHCP on internal networks allows the dynamic allocation of network resources and minimizes the amount of time required to configure hosts. These techniques allow network administrators to make more efficient use of the available IP address space. Their implementation has extended the lifetime of the current IPv4 addressing scheme and has reduced the necessity for a rapid migration to IPv6.

The next chapter examines the equipment and technologies used in a WAN environment. We consider both circuit-switched and packet-switched technologies, looking briefly at such things as dialup, ISDN, Frame Relay, cable, and DSL. WAN design and current standards also are introduced.
Check Your Understanding

Complete all the review questions listed here to test your understanding of the topics and concepts in this chapter. Answers are listed in Appendix A, “Answers to Check Your Understanding and Challenge Questions.”

1. Which type of NAT is usually configured to provide Internet connectivity to corporate users in a large organization?
   A. Dynamic
   B. Static
   C. Overloaded
   D. Manual

2. Through which kind of address does a machine on the external network see another machine on the same external network?
   A. Inside local
   B. Inside global
   C. Outside local
   D. Outside global

3. What does PAT do first if the source port is in use on the first available inside global address and a pool of addresses has been configured?
   A. PAT moves to the next available address and checks for source port availability.
   B. PAT checks for an available port in the same range as the source port on the first available global address.
   C. PAT randomly picks a source port on the first global address for the translation.
   D. PAT randomly picks a source port-global address combination from the available resources.
   E. PAT does not translate the packet.

4. Which port does DHCP use for a DHCPREQUEST?
   A. TCP 67
   B. TCP 68
   C. UDP 67
   D. UDP 68

5. Which message does the DHCP server issue if the client takes too long to accept the DHCPOFFER?
   A. DHCPACK
6. Which DHCP message informs the client that it can use the supplied network configuration?
   A. DHCPREQUEST
   B. DHCPOFFER
   C. DHCPDISCOVER
   D. DHCPACK
   E. DHCPNAK

7. Which DHCP messages are sent as a broadcast?
   A. DHCPDISCOVER
   B. DHCPOFFER
   C. DHCPREQUEST
   D. DHCPACK
   E. DHCPDECLINE
   F. DHCPNAK

8. Which command provides detailed information about the DHCP messages seen by and replied to by the IOS DHCP service?
   A. `show ip dhcp services`
   B. `debug ip dhcp events`
   C. `show ip dhcp server statistics`
   D. `debug ip dhcp`

9. Which command allows the network administrator to exclude the first ten addresses in the 10.20.30.0/24 DHCP subnet pool for use by devices requiring static IP addresses?
   A. `Router(config-dhcp)#exclude 10.20.30.1 10.20.30.10`
   B. `Router(config-dhcp)#exclude first 10`
   C. `Router(config-dhcp)#ip dhcp excluded-address 10.20.30.1 10.20.30.10`
   D. `Router(config)#ip dhcp excluded-address 10.20.20.1 10.20.30.10`
10. Why is a helper address required in an environment where the DHCP server is on a different segment of the network than the hosts?
   A. The helper address forwards all DHCP messages from the client to the server.
   B. The helper address forwards all DHCP messages from the server to the client.
   C. The helper address forwards all broadcast DHCP messages from the client to the server.
   D. The helper address forwards all broadcast DHCP messages from the server to the client.

11. What technology is described by each of the following IETF RFCs?
   RFC 951
   RFC 1542
   RFC 1631
   RFC 1918
   RFC 2131

12. What are the three mechanisms by which DHCP can assign an IP address?

13. How does the classless system help conserve the existing IPv4 address space?

14. Why might a network administrator want to run DHCP on a router instead of a server?

15. Why is it important to adjust the timeout value for dynamic NAT?

16. By default, which UDP ports are forwarded by a helper address?

Challenge Questions and Activities

Complete the following questions as well. These questions are purposefully designed to be similar to the more complex styles of questions you might expect to see on the CCNA exam. This section may also list activities that will help you prepare for the exams.

Activity 1

Using Packet Tracer, load the NAT_preconfigs.pkt configuration file that can be downloaded from www.ciscopress.com/1587131722. This configuration matches Figure 1-10. Complete all exercises using this as a starting file.
ABC Inc. has decided to reduce its network expenses by implementing NAT on the border router. Different scenarios are currently being investigated, and you have been asked to implement appropriate solutions for each scenario. Figure 1-10 diagrams the network. The starting configurations can be found in the file NAT_preconfigs.pkt.

After much discussion, it was decided to evaluate three different scenarios for possible implementation. The first scenario evaluates static NAT, the second dynamic NAT, and the third PAT. Each of these is to be evaluated independently for possible inclusion in the final solution.

**Scenario A**

Two machines belong to the network administration team and require static addresses to be used for translation. The machines have the internal addresses 10.20.30.3 and 10.20.30.4. These must be translated to the addresses 209.165.200.225 and 209.165.200.226, respectively, before leaving the internal network.

**Scenario B**

All IP traffic originating on the 10.20.30.0/24 network must be assigned an address from the address pool 209.165.200.225 to 209.165.200.230 before leaving the internal network.

**Scenario C**

IP traffic originating from the 10.20.30.0/24 network should use the address of the exit interface (serial 2/0) when leaving the internal network.
Final Implementation

After careful consideration of the benefits of each of the different types of NAT that were evaluated, it was decided that a solution combining the technologies was the most appropriate for the company. Implement the solution documented next using the NAT_preconfigs.pkt file as a starting point.

Static maps must be created for the machines in the IT department. These machines have IP addresses ranging from 10.20.30.11 to 10.20.30.15 and must be mapped to the addresses 209.165.200.231 to 209.165.200.235.

All other IP traffic originating from the 10.20.30.0/25 subnet should use the addresses 209.165.200.225/27 to 209.165.200.229/27 for translation.

IP traffic originating from the 10.20.30.128/25 subnet must use the address 209.165.200.230/27 when exiting the network.

Activity 2

As a network administrator, you have decided to rearrange the network to create a server farm that will house all servers, including those that support DHCP, NetBIOS, and TFTP. Figure 1-11 shows both the old and new networks. Part (a) of Figure 1-11 diagrams the initial network, and part (b) diagrams the new network. What changes will have to be made in the router configuration to give clients on the fastethernet 0/0 and fastethernet 0/1 segments access to the server farm resources?

Figure 1-11  DHCP Network Redesign

![Diagram of DHCP Network Redesign](image)