CCNA™
CISCO® CERTIFIED NETWORK ASSOCIATE

14

Advanced IP Features

CERTIFICATION OBJECTIVES

The preceding chapter introduced you to ACLs, one of the advanced features of the router's IOS. This chapter covers two more advanced features: address translation and the Dynamic Host Configuration Protocol (DHCP). Address translation allows you to change the source or destination address inside the IP packet. This is typically done if you are using private IP addresses inside your network, or have overlapping addresses. The first half of this chapter provides an overview of address translation, including the many terms used and the different types of address translation and its configuration. The second half of this book has a brief overview of DHCP, which allows you to assign and acquire IP addressing information dynamically, and its configuration.

CERTIFICATION OBJECTIVE 14.01

Address Translation Overview

Address translation was originally developed to solve two problems: handling a shortage of IP addresses and hiding network addressing schemes. Most people think that address translation is used primarily to solve the first problem. However, as the first half of this chapter illustrates, address translation provides solutions for many problems and has many advantages.

Running Out of Addresses

Because of the huge Internet explosion during the early 1990s, it was foreseen that the current IP addressing scheme would not accommodate the number of devices that would need public addresses. A long-term solution was conceived to address this; it called for the enhancement of the TCP/IP protocol stack, including the addressing format. This new addressing format was called IPv6. Whereas the current IP addressing scheme (IPv4) uses 32 bits to represent addresses, IPv6 uses 128 bits for addressing, creating billions of extra addresses.

Private Addresses

It took a while for IPv6 to become a standard, and on top of this, not many companies have implemented it, even ISPs on the Internet backbone. The main reason that this standard hasn't been embraced is the success of the two short-term solutions to the address shortage problem: schemes to create additional addresses, called private addresses, and to translate these addresses to public addresses using address translation.

exam Watch

Remember the private addresses listed in Table 14-1.

RFC 1918, by the Internet Engineering Task Force (IETF), is a document that was created to address the shortage of addresses. When devices want to communicate, each device needs a unique IP address. RFC 1918 has created a private address space that any company can use internally. Table 14-1 shows the range of private addresses that RFC 1918 set aside. As you can see from this table, you have 1 Class A, 16 Class B, and 256 Class C addresses at your disposal. Just the single Class A address of 10.0.0.0 has over 17 million IP addresses, more than enough to accommodate your company's needs.

One of the main issues of RFC 1918 addresses is that they can be used only internally within a company and cannot be used to communicate to a public network, such as the Internet. For this reason, they are commonly referred to as *private addresses*. If you send packets with RFC 1918 addresses in them to your ISP, for instance, your ISP will either filter them or not be able to route this traffic back to your devices. Obviously, this creates a connectivity problem, since many of your devices with private addresses need to send and receive traffic from public networks.

Address Translation

A second standard, RFC 1631, was created to solve this problem. It defines a process called Network Address Translation (NAT), which allows you to change an IP address in a packet to a different address. When communicating to devices in a public network, your device needs to use a source address that is a public address. Address translation allows you to translate your internal private addresses to public addresses before these packets leave your network.

exam Watch

Remember the reasons you might want to use address translation in your network.

Actually, RFC 1631 doesn't specify that the address you are changing has to be a private address—it can be any address. This is useful if you randomly chose someone else's public address space but still want to connect to the Internet. Obviously, you don't own this address space, but address translation allows you to keep

TABLE 14-1 RFC 1918 Private Addresses

Class	Range of Addresses
A	10.0.0.0–10.255.255.255
B	172.16.0.0–172.31.255.255
C	192.168.0.0–192.168.255.255

your current addressing scheme but translate these source addresses to the ones your ISP assigned to you before your packets enter the Internet.

Here are some common reasons that you might need to employ address translation:

- You have to use private addressing because your ISP didn't assign you enough public addresses.
- You are using public addresses but have changed ISPs, and your new ISP won't support these public addresses.
- You are merging two companies together and they are using the same address space, for instance, 10.0.0.0, which creates routing and reachability issues.
- You want to assign the same IP address to multiple machines so that users on the Internet see this offered service as a single logical computer.

Types of Address Translation

Address translation comes in a variety of types, like Network Address Translation (NAT), Port Address Translation (PAT), dynamic address translation, and static address translation. Because of the many terms used, the concept of address translation can be confusing, especially since many people use the address translation terms incorrectly. The following sections cover the different types of address translation.

exam Watch

Remember the terms in Tables 14-2 and 14-3.

Terms and Definitions

Table 14-2 shows some common terms used in address translation, and Table 14-3 shows some terms used for types of address translation.

Network Address Translation

Network Address Translation (NAT) translates one IP address to another. This can be a source address or a destination address. There are two basic implementations of NAT: static and dynamic. The following two sections cover the mechanics of these implementations.

Static NAT With static NAT, a manual translation is performed by an address translation device, translating one IP address to a different one. Typically, static

TABLE 14-2 Common Address Translation Terms

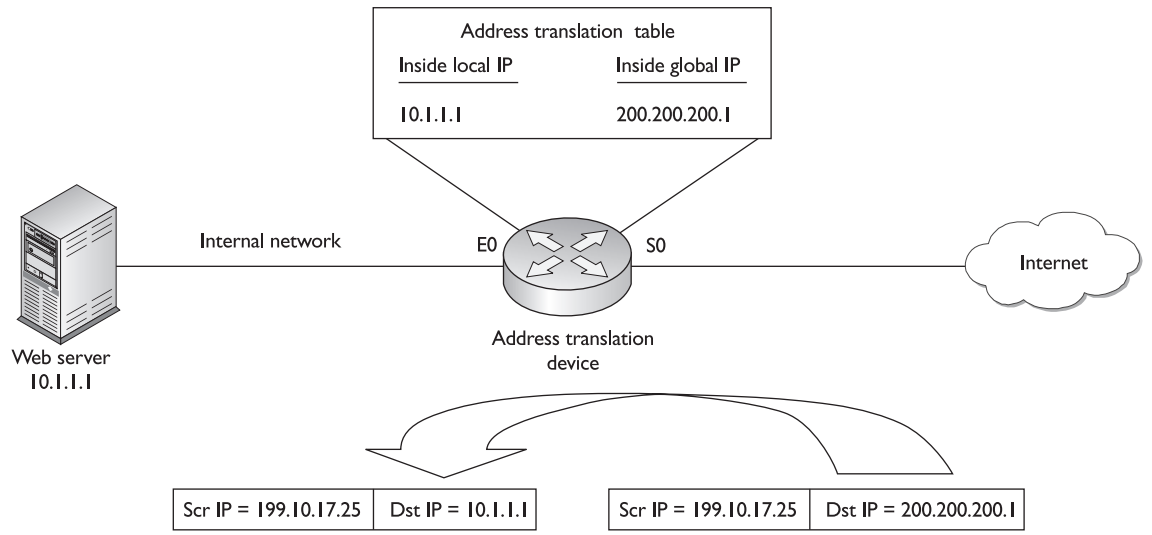
Term	Definition
Inside	Networks located on the inside of your network
Outside	Networks located outside of your network
Local	The IP address physically assigned to a device
Global	The public IP address physically or logically assigned to a device
Inside local IP address	An inside device with an assigned private IP address
Inside global IP address	An inside device with a registered public IP address
Outside global IP address	An outside device with a registered public IP address
Outside local IP address	An outside device with an assigned private IP address

NAT is used to translate destination IP addresses in packets as they come into your network, but you can translate source addresses also. Figure 14-1 shows a simple example of outside users trying to access an inside web server. In this example, you want Internet users to access an internal web server, but this server is using a private address (10.1.1.1). This creates a problem, since if an outside user would put a private address in the destination IP address field, their ISP would drop this. Therefore, the web server needs to be presented as a having a public address. This is defined in the address translation device (in our case, this is a Cisco router).

TABLE 14-3 Common Address Translation Types

Translation Type	Explanation
Simple	One IP address is translated to a different IP address.
Extended	One IP address and one TCP/UDP port number are mapped to a different IP address and, possibly, port number.
Static	A manual address translation is performed between two addresses, and possibly port numbers.
Dynamic	An address translation device automatically performs address translation between two addresses, and possibly port numbers.
Network Address Translation (NAT)	Only IP addresses are translated (not port numbers).
Port Address Translation (PAT)	Many inside IP addresses are translated to a single IP address, where each inside address is given a different port number for uniqueness.

FIGURE 14-1 Static NAT example



The web server is assigned an inside global IP address of 200.200.200.1 on the router, and your DNS server advertises this address to the outside users. When outside users send packets to the 200.200.200.1 address, the router examines its translation table for a matching entry. In this case, it sees that 200.200.200.1 maps to 10.1.1.1. The router then changes the destination IP address to 10.1.1.1 and forwards it to the inside web server. Note that if the router didn't do the translation to 10.1.1.1, the web server wouldn't know this information was meant for itself, since the outside user sent the traffic originally to 200.200.200.1. Likewise, when the web server sends traffic out to the public network, the router compares the source IP address to entries in its translation table, and if it finds a match, it changes the inside local IP address (private source address--10.1.1.1) to the inside global IP address (public source address--200.200.200.1).

Dynamic NAT With static address translation, you need to manually build the translations. If you have 1,000 devices, you need to create 1,000 static entries in the address translation table, which is a lot of work. Typically, static translation is done for inside resources that outside people want to access. When inside users access outside resources, dynamic NAT is typically used. In this situation, the address assigned to the internal user isn't that important, since outside devices don't directly access your internal users—they just return traffic to them that the inside user requested.

With dynamic NAT, you must manually define two sets of addresses on your address translation device. One set defines which inside addresses are allowed to be translated, and the other defines what these addresses are to be translated to. When an inside user sends traffic through the address translation device, say a router, it examines the source IP address and compares it to the internal local address pool. If it finds a match, then it determines which inside global address pool it should use for the translation. It then dynamically picks an address in the global address pool that is not currently assigned to an inside device. The router adds this entry in its address translation table, and the packet is then sent to the outside world. If no entry is found in the local address pool, then the address is not translated and forwarded to the outside world in its original state.

When returning traffic comes back into your network, the address translation device examines the destination IP addresses and checks them against the address translation table. Upon finding a matching entry, it converts the global inside address to the local inside address in the destination IP address field of the packet header and forwards the packet to the inside network.

Port Address Translation

One problem with static or dynamic NAT is that it provides only a one-to-one address translation. Therefore, if you have 5,000 internal devices with private addresses, and all 5,000 devices try to reach the Internet simultaneously, you need 5,000 public addresses in your inside global address pool. If you have only 1,000 public addresses, only the first 1,000 devices are translated and the remaining 4,000 won't be able to reach outside destinations.

To overcome this problem, you can use a process called *address overloading*. There are actually many terms used to describe this process, including Port Address Translation (PAT) and Network Address Port Translation (NAPT).

Using the Same IP Address With PAT, all machines that go through the address translation device have the same IP address assigned to them, and so the source port numbers are used to differentiate the different connections. If two devices have the same source port number, the translation device changes one of them to ensure uniqueness. When you look at the translation table in the address translation device, you'll see the following items:

- Inside local IP address (original source private IP)
- Inside local port number (original source port number)
- Inside global IP address (translated public source IP)

- Inside global port number (new source port number)
- Outside global IP address (destination public address)
- Outside global port number (destination port number)

One main advantage of NAT over PAT is that NAT will basically work with most types of IP connections. Since PAT relies on port numbers to differentiate connections, PAT works only with the TCP and UDP protocols; however, many vendors, including Cisco, also support ICMP with PAT using a proprietary translation method.

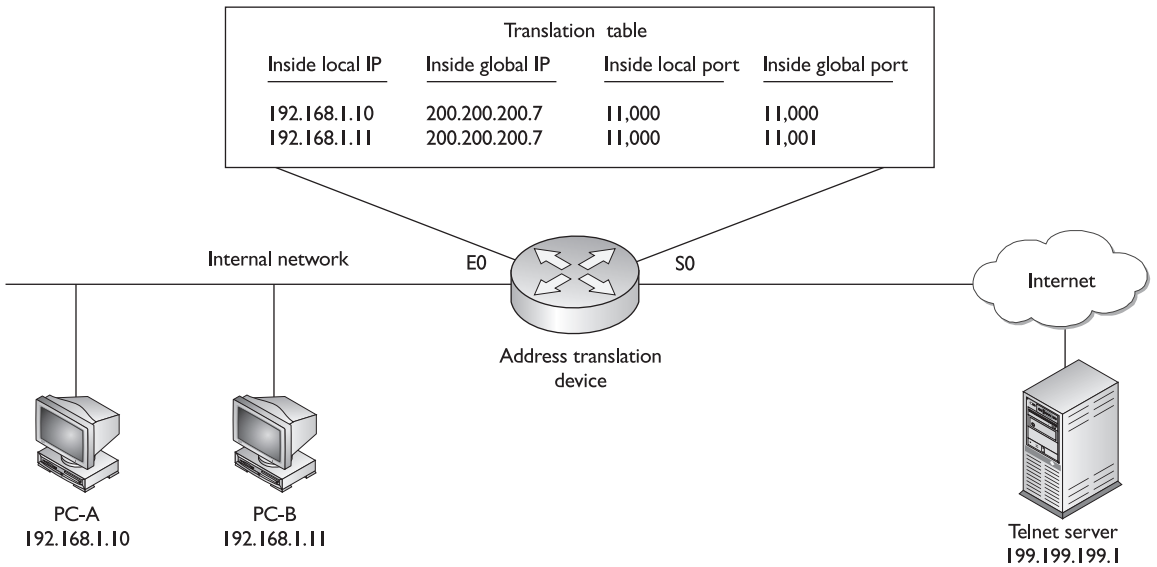
Example Using PAT Let's take a look at an example, shown in Figure 14-2, using PAT. In this example, both PCs execute a telnet to 199.199.199.1, and both of these connections use a source port number of 11,000. When these connections reach the address translation device, the translation device performs its PAT translation. For the first connection, say PC-A, the source IP address is changed to 200.200.200.7. Since this is the first connection, the source port number is left as is. When PC-B makes a telnet connection to the remote device, since it is using a source port number already in the table for a connection to the telnet server, the address translation device changes it from 11,000 to 11,001. Therefore, when traffic is sent from the telnet server to the inside PCs, the address translation device will be able to differentiate the two connections and undo the translation correctly by examining both the destination IP address and port number.

Since the port number in the TCP and UDP header is 16 bits in length, you can theoretically represent 65,536 internal machines with a single public IP address. However, in reality, this number is about 4,000 devices per public address. Note that you don't have to restrict yourself to one type of address translation process. For instance, you can use PAT for inside-to-outside connections and static NAT for outside-to-inside connections.

exam Watch

PAT, or address overloading, allows you to use the same global IP address for all internal devices, where the source port is used (possibly changed), to differentiate among the different translated connections.

FIGURE 14-2 PAT example



Port Address Redirection The last example showed PAT being carried out dynamically by the address translation device. There are situations, however, where this will not work. For instance, your ISP might assign you a single public IP address. You need to use this with PAT to allow inside users to access outside resources. However, you have a problem if you want outside users to access an internal service, such as a web server. Dynamic PAT, unfortunately, won't work in this situation.

However, there is another solution: static PAT. Static PAT is often called port address redirection (PAR). Let's look at a simple example to illustrate how PAR works. Assume that your ISP has assigned you a single public IP address: 199.199.199.1. You need to use this address for inside users to access the outside world, but you still need the outside world to access an internal web server. With static PAT, you set up your address translation device to look at not only the destination IP address (199.199.199.1), but also the destination port number (80 for a web server). You create a static PAT entry such that when the address translation device sees this combination of address and port

exam Watch

Port address redirection allows you to redirect application traffic directed to one address to a different address.

number, the device translates it to the inside local IP address and, possibly, the port number used for the service on this inside device.

Advantages of Address Translation

As mentioned at the beginning of this part of the chapter, address translation devices are typically used to give you an almost inexhaustible number of addresses as well as to hide your internal network addressing scheme. Another advantage of address translation is that if you change ISPs or merge with another company, you can keep your current scheme and make any necessary changes on your address translation device or devices, making your address management easier.

Another big advantage that address translation provides is that it gives you tighter control over traffic entering and leaving your network. For example, if you are using private addresses internally, all traffic entering and leaving must pass through an address translation device. Because of this restriction, it is much easier to implement your security and business policies.

Disadvantages of Address Translation

Even though address translation solves many problems and has many advantages, it also has its share of disadvantages. Here are the three main issues with address translation:

- Each connection has an added delay.
- Troubleshooting is more difficult.
- Not all applications work with address translation.

exam Watch

Remember the disadvantages and limitations of address translation.

Since address translation changes the contents of packets and, possibly, segment headers, as well as computing any necessary new checksum values, extra processing is required on each packet. This extra processing, obviously, will affect the throughput and speed of your connections. The more packets that pass through your address translation device needing translation, the more likely your users are to notice the delay. Therefore, choosing the appropriate product for address translation becomes very important.

Also, whenever problems arise with connections involving address translation, it is more difficult to troubleshoot them. When troubleshooting, it becomes more difficult to track down the real source and destination of a connection—you have

to log into your address translation device and look at your translation tables. And if the packet is going through multiple layers of translation, possibly at both the source and destination sites, this can be a hair-pulling experience. Also, even though one of the advantages of address translation is that it hides your internal addressing scheme, it also creates security issues—an external hacker can more easily hide their identity by having their packets go through a translation device or multiple translation devices, trying to hide their true IP address.

Probably the most difficult issue with address translation is that not all applications will work with it. For instance, some applications embed IP addressing or port information in the actual data payload, expecting the destination device to use this addressing information in the payload instead of what is in the packet and segment headers. This can pose a problem with address translation, since address translation, by default, doesn't translate data payload information, only header information. Multimedia and NetBIOS applications are notorious for embedding addressing information in data payloads.

In some instances, certain vendors' address translation devices have the ability to detect this process for certain applications. For instance, Cisco routers and PIX firewalls support a fix-up process that addresses many NetBIOS and multimedia issues, including embedded addressing information. However, if your product doesn't support this feature, you'll need to disable address translation for the affected devices.

CERTIFICATION OBJECTIVE 14.02

Address Translation Configuration

The configuration of the different types of address translation, like NAT and PAT, is very similar. The following sections cover the configuration and verification of some of the types of address translation discussed so far.

NAT Configuration

As mentioned earlier, there are two types of NAT: static and dynamic. The configuration process is similar for the two types. Probably the most difficult process of configuring address translation is understanding the difference between the terms *inside* and *outside*. These terms refer to where your devices are located (inside) and where the external network (the Internet, for instance) is (outside). This is important when it comes to the configuration

of address translation. In the IOS, there are two basic configuration steps that you must perform:

- Define the address translation type (*Global Configuration* mode commands).
- Define the location of devices (*Interface Subconfiguration* mode commands).

The following sections cover the configuration of both static and dynamic NAT.

Static NAT

As mentioned earlier in this chapter, static NAT is typically used when devices on the outside of your network want to access resources, such as web, DNS, and email servers, on the inside. Here are the two commands to define the static translations for NAT:

```
Router(config)# ip nat inside source static
                        inside_local_source_IP_address
                        inside_global_source_IP_address
Router(config)# ip nat outside source static
                        outside_global_destination_IP_address
outside_local_destination_IP_address
```

exam Watch

Remember how to create a static translation with the `ip nat inside|outside source static` command.

The **inside** and **outside** parameters specify the direction in which translation will occur. For instance, the **inside** keyword specifies that the inside *source* local IP addresses are translated to an inside global IP address. The **outside** keyword changes the outside *destination* global IP address to an outside local address.

After you configure your translations, you must specify which interfaces on your router are considered to be on the inside and which ones are on the outside. This is done with the following configuration:

```
Router(config)# interface type [slot_#/]port_#
Router(config-if)# ip nat inside|outside
```

exam Watch

Use the `ip nat inside|outside` Interface command to specify which interfaces are considered "`inside`" and which are "`outside`."

Specify **inside** for interfaces connected to the inside of your network and **outside** for interfaces connected to external networks.

Let's take a look a simple static NAT example. I'll use the network shown in

Figure 14-3 for this example. In this example, an internal web server (10.1.1.1) will be assigned a global IP address of 200.200.200.1.

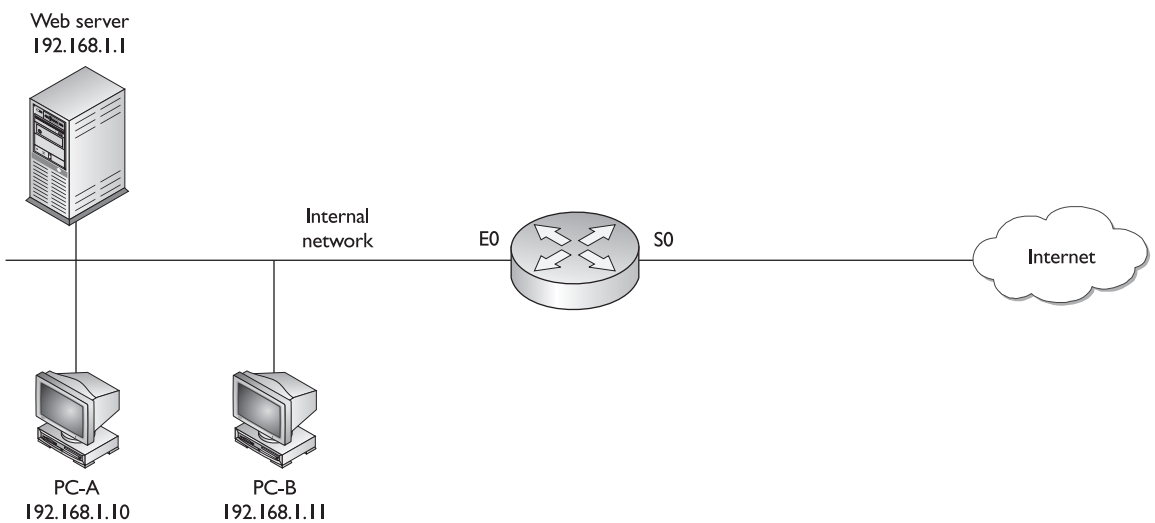
Here's the configuration:

```
Router(config)# ip nat inside source static
                        192.168.1.1 200.200.200.1
Router(config)# interface ethernet 0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# interface serial 0
 Router(config-if)# ip nat outside
```

The **ip nat inside source static** command defines the translation. The **ip nat inside** and **outside** commands specify what interfaces are on the inside (E0) and what interfaces are on the outside (S0). Note that any packets that don't match the address translation rule will pass between these two interfaces untranslated. If you want only translated packets to pass between these interfaces, you'll need to configure an appropriate ACL or ACLs.

14.01. The CD contains a multimedia demonstration of configuring static NAT on a router.

FIGURE 14-3 Network translation example



Dynamic NAT

When you are configuring dynamic NAT, you'll need to configure three things: what inside addresses are to be translated, what global addresses will be used for the dynamic translation, and what interfaces are involved in the translation. To specify what internal devices will have their source address translated, use the following command:

```
Router(config)# ip nat inside source
                        list standard_IP_ACL_#
pool NAT_pool_name
```

The **ip nat inside source list** command requires you to configure a standard IP ACL that has a list of the inside source addresses that will be translated—any addresses listed with a **permit** statement will be translated, and any addresses listed with a **deny**, or the implicit deny, statement will not be translated. Following this is the name of the address pool. This ties together the address pool you'll use that contains your global source IP addresses.

To create the pool of source inside global IP addresses, use this command:

```
Router(config)# ip nat pool NAT_pool_name
beginning_inside_global_IP_address
ending_inside_global_IP_address
                        netmask subnet_mask_of_addresses
```

The pool name that you specify references the inside addresses that will be translated from the **ip nat inside source list** command. Next, list the beginning and ending IP addresses in the pool, followed by the subnet mask for the addresses.

Once you have done this, the last thing you need to configure is which interfaces are considered to be on the inside and outside of your network. Use the **ip nat**

exam Watch

The `ip nat inside source list` command specifies which internal addresses will be dynamically translated. The `ip nat pool` command specifies the global addresses to use when performing dynamic translation of local addresses.

inside and **ip nat outside** *Interface Subconfiguration* mode commands discussed earlier.

I'll use the network shown in Figure 14-3 to illustrate how dynamic NAT is configured. In this example, the two PCs will have dynamic NAT performed on them.

```
Router(config)# ip nat inside source list 1 pool nat-pool
Router(config)# access-list 1 permit 192.168.1.10 0.0.0.0
Router(config)# access-list 1 permit 192.168.1.11 0.0.0.0
Router(config)# ip nat pool nat-pool 200.200.200.2
                      200.200.200.3 netmask 255.255.255.0
Router(config)# interface ethernet 0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# interface serial 0
Router(config-if)# ip nat outside
```

The **ip nat inside source list** command specifies the inside source IP addresses that will be translated. Notice that these are addresses in ACL 1—192.168.1.10 and 192.168.1.11. They are associated with the global address pool called *nat-pool*. The **ip nat pool** command specifies the global addresses that the inside source addresses will be translated to. And finally, `ethernet0` is specified as being on the inside and `serial0` is on the outside.

14.02. The CD contains a multimedia demonstration of configuring dynamic NAT on a router.

PAT Configuration

The last example showed an example of dynamic NAT. This section covers how to configure PAT on your router. This configuration, which is very similar to configuring dynamic NAT, requires three basic translation commands. The first thing you specify is which inside devices will have their source addresses translated. You'll use the same command as you used in dynamic NAT, but you'll add the **overload** parameter to specify that PAT is to be performed:

```
Router(config)# ip nat inside source
                      list standard_IP_ACL_#
                      pool NAT_pool_name overload
```

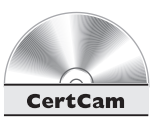
Next, you specify the global pool to use. Again, you'll use the same command as you used in dynamic NAT:

```
Router(config)# ip nat pool NAT_pool_name
beginning_inside_global_IP_address
ending_inside_global_IP_address
                        netmask subnet_mask_of_addresses
```

You can specify more than one address to use in PAT, or you can specify a single IP address (use the same address for the beginning and ending addresses). And last, you have to tell the IOS which interfaces are inside and outside, respectively, in terms of the **ip nat inside** and **ip nat outside** commands.

Let's use Figure 14-3 to illustrate how PAT is configured. In this example, only a single IP address is placed in the address pool (200.200.200.2):

```
Router(config)# ip nat inside source list 1 pool
                        nat-pool overload
Router(config)# access-list 1 permit 192.168.1.10 0.0.0.0
Router(config)# access-list 1 permit 192.168.1.11 0.0.0.0
Router(config)# ip nat pool nat-pool 200.200.200.2
                        200.200.200.2
                        netmask 255.255.255.0
Router(config)# interface ethernet 0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# interface serial 0
Router(config-if)# ip nat outside
```



14.03. The CD contains a multimedia demonstration of configuring PAT on a router.

Load Distribution Configuration

Cisco routers support a process called *load*, or *traffic*, *distribution*. Load distribution allows you to distribute connection requests destined to a single IP address to multiple machines. For instance, you might have two web servers with the same content and want to split the incoming connections across both of these machines. Since both machines have *different* IP addresses, this creates a problem. Normally, a DNS server would send back just one address for a requested name resolution. You could solve this by using an enhanced DNS product that varies its replies among a group of addresses. The problem with this approach is that devices typically cache this information, and thus more traffic might be sent to one server than another.

A better choice is to use the load distribution feature in NAT. Set up your DNS server to send back a single IP address. Within your NAT configuration, you'll tell the router to round-robin between a range of internal addresses where this service is located. One problem with this feature is that it doesn't keep tabs on which services are available or not available, nor does it keep track of actual traffic loads on each of these internal devices—it load-balances on a connection-by-connection basis. Therefore, if you are concerned about these limitations, you'll want to purchase a true load balancing product.

The configuration of load distribution involves three steps. In the first step, you specify the internal IP addresses that are configured on the devices offering the service. This is done with the **ip nat pool** command:

```
Router(config)# ip nat pool pool_name
                        beginning_inside_local_IP_address
ending_inside_local_IP_address
                        prefix-length subnet_mask_bits
                        type rotary
```

First, you need to give the internal addresses a unique pool name. Following this are the beginning and ending internal addresses of the devices offering the same service. Following this, you need to configure the length, in bits, of the subnet mask of the location of these devices. And last, you have to specify the **type rotary** parameter to tell the IOS that you want to round-robin the assignment of connections to these internal devices. This causes the IOS to send the first connection request to the first address, the second request to the second address, and so on.

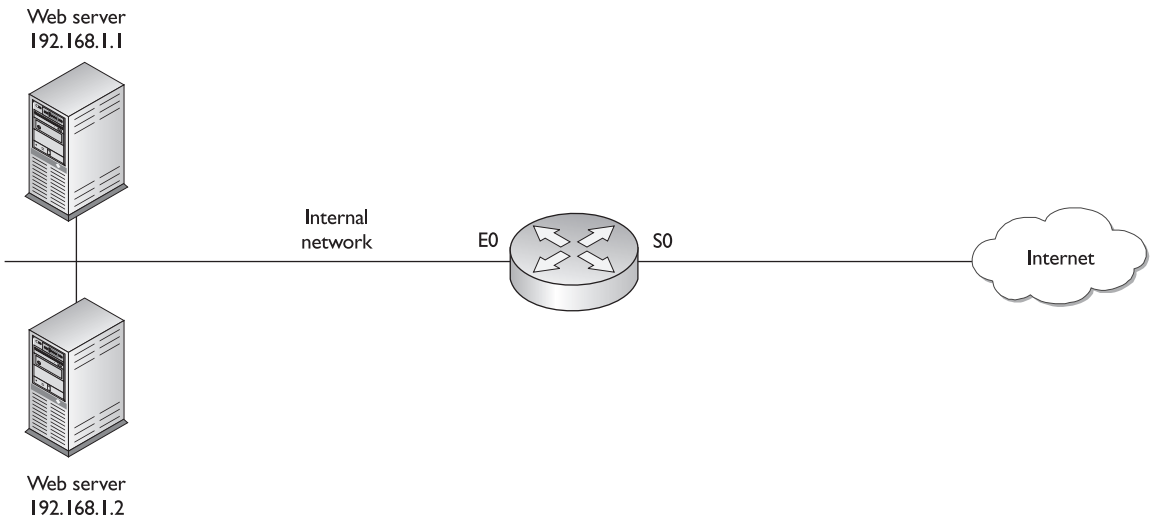
Next, you need to specify the global IP address that outside devices are using to reach the inside resource:

```
Router(config)# ip nat inside destination
                        list standard_ACL_# pool pool_name
```

This command requires you to specify a standard ACL number, which references the global IP address or addresses that will be redirected to internal machines. Second, you need to specify the pool name that needs to match the **ip nat pool** command. And last, you have to tell the IOS which interfaces are inside and outside, respectively, in terms of the **ip nat inside** and **ip nat outside** commands.

Let's take a look at an example configuration that uses load distribution. I'll use the network shown in Figure 14-4 to illustrate the configuration. In this example, there are two web servers with the same information on them: 192.168.1.1 and 192.168.1.2. They are represented as a single device with a global address of 200.200.200.1.

FIGURE 14-4 Load distribution example



Here's the configuration:

```
Router(config)# ip nat pool inside-hosts
                        192.168.1.1 192.168.1.2
                        prefix-length 24 type rotary
Router(config)# ip nat inside destination list 1
                        pool inside-hosts
Router(config)# access-list 1 permit 200.200.200.1
Router(config)# interface ethernet 0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# interface serial 0
Router(config-if)# ip nat outside
```

14.04. The CD contains a multimedia demonstration of configuring load distribution on a router.

Address Translation Verification

Once you have configured address translation, there are many commands you can use to verify and troubleshoot the operation of address translation on your router. For instance, if you want to see the address translation on your router, use the **show ip nat translations** command:

```
Router# show ip nat translations
Pro  Inside global  Inside local  Outside local  Outside global
---  200.200.200.1  192.168.1.1   ---            ---
---  200.200.200.2  192.168.1.2   ---            ---
```

In this example, two addresses are being translated: 192.168.1.1 to 200.200.200.1 and 192.168.1.2 to 200.200.200.2. Notice that no protocol is listed (`Pro`) or port numbers, indicating that this is NAT, not PAT.

Here's an example using PAT:

```
Router# show ip nat translations
Pro Inside global      Inside local     Outside local Outside global
tcp 200.200.200.1:1080 192.168.1.1:1080 201.1.1.1:23  201.1.1.1:23
tcp 200.200.200.1:1081 192.168.1.2:1080 201.1.1.1:23  201.1.1.1:23
```

In this example, both 192.168.1.1 and 192.168.1.2 are accessing the same outside device (201.1.1.1) using telnet. Notice that both also use the same source port number (1080 under the `Inside local` column). The IOS has noticed this and changed the second connection's source port number from 1080 to 1081.

14.05. The CD contains a multimedia demonstration of the `show ip nat translations` command on a router.

You can even see address translations statistics on your router with this command:

```
Router# show ip nat statistics
Total translations: 2 (0 static, 2 dynamic; 0 extended)
Outside interfaces: Serial0
Inside interfaces: Ethernet0
Hits: 98 Misses: 4
Expired translations: 1
Dynamic mappings:
-- Inside Source
access-list 1 pool nat-pool refcount 2
pool nat-pool: netmask 255.255.255.255
start 200.200.200.10 end 200.200.200.254
type generic, total addresses 12, allocated 1 (9%), misses 0
```

In this example, *hits* refers to the number of times the IOS looked into the translation table and found a match, while *misses* indicates the number of times the IOS looked in the table for a translation, didn't find one, and had to create one.

14.06. The CD contains a multimedia demonstration of the `show ip nat statistics` command on a router.

For dynamic entries in the translation table, you can clear all of entries, or specific entries, using the following commands:

```
Router# clear ip nat translation *
Router# clear ip nat translation inside
                        global_IP_address local_IP_address
Router# clear ip nat translation outside
                        global_IP_address local_IP_address
Router# clear ip nat translation protocol inside
                        global_IP_address global_port
local_IP_address local_port
```

The first command clears all dynamic entries in the table. Note that to clear static entries, you need to delete your static NAT configuration commands from within *Configuration* mode.

14.07. The CD contains a multimedia demonstration of the `clear ip nat translation` command on a router.

Besides using **show** commands, you can also use **debug** commands for troubleshooting. The **debug** `ip nat` command, for instance, will show the translations the IOS is doing on every translated packet. This is useful in determining if the IOS is translating your packet and segment header addressing information correctly. Please note that on a busy network, this command will chew up a lot of CPU cycles on your router. Here's an example of this command:

```
Router# debug ip nat
05:32:23: NAT: s=192.168.1.10->200.200.200.2, d=201.1.1.1 [70]
05:32:23: NAT*: s=201.1.1.1, d=200.200.200.2->192.168.1.10 [70]
```

exam Watch

Use the `show ip nat translations` command to display the router's translations. Use the `clear ip nat translations` command to clear dynamic translations from the translation table. The `debug ip nat` command shows the router performing address translation in a real-time fashion.

In the first line of this example, an internal machine (192.168.1.10) is having its address translated to 200.200.200.2 where the packet is being sent to 201.1.1.1. The second line shows the returning traffic from 201.1.1.1 and the translation from the global to the local inside address.

14.08. The CD contains a multimedia demonstration of the `debug ip nat` command on a router.

EXERCISE 14-1

Configuring Address Translation

These last few sections dealt with the configuration of address translation on IOS routers. This exercise will help you reinforce this material by configuring a simple static NAT translation. You'll perform this lab using Boson's NetSim™ simulator. This exercise has you first set static routes two routers (2600 and 2500) and verify network connectivity. Following this, you'll configure your ACL. After starting up the simulator, click on the *LabNavigator* button. Next, double-click on *Exercise 14-1* and click on the *Load Lab* button. This will load the lab configuration based on Chapter 5's and 7's exercises.

1. On the 2500, configure a static route to 192.168.1.0/24, which is off of the 2600. View the routing table.

   At the top of the simulator in the menu bar, click on the *eRouters* icon and choose *2500*. Configure the static route: **`configure terminal`**, **`ip route 192.168.1.0 255.255.255.0 192.168.2.1`**, and **`end`**. View the static route: **`show ip route`**. Make sure that 192.168.1.0/24 shows up in the routing table as a static route (`S`).

2. On the 2600, configure a static route to 192.168.3.0/24, which is off of the 2500. View the routing table.

   At the top of the simulator in the menu bar, click on the *eRouters* icon and choose *2600*. Configure the static route: **`configure terminal`**, **`ip route 192.168.3.0 255.255.255.0 192.168.2.2`**, and **`end`**. View the static route: **`show ip route`**. Make sure that 192.168.3.0/24 shows up in the routing table as a static route (`S`).

3. From Host3, ping the `fa0/0` interface of the 2600. From Host3, ping Host1.

At the top of the simulator in the menu bar, click on the *eStations* icon and choose *Host3*. Ping the `serial0` and `fa0/0` interface of the 2600 router: **`ping 192.168.2.1`** and **`ping 192.168.1.1`**. The pings should be successful. Ping Host1: **`ping 192.168.1.10`**. The ping should be successful.

4. Check network connectivity between the two 2950 switches and the Host3.

   At the top of the simulator in the menu bar, click on the *eSwitches* icon and choose *2950-1*. From the 2950-1 switch, ping Host3: **`ping 192.168.3.2`**. At the top of the simulator in the menu bar, click on the *eSwitches* icon and choose *2950-2*. From the 2950-2 switch, ping Host3: **`ping 192.168.3.2`**. At the top of the simulator in the menu bar, click on the *eStations* icon and choose *Host3*. From Host3, ping the 2950-1 switch: **`ping 192.168.1.4`**. From Host3, ping the 2950-2 switch: **`ping 192.168.1.3`**.

5. Set up a static route on the 2500 to reach 10.0.0.0/8, which are the global addresses behind the 2600. Remove the 192.168.1.0/24 static route.

   At the top of the simulator in the menu bar, click on the *eRouters* icon and choose *2500*. On the 2500, set up a static route: **`configure terminal`** and **`ip route 10.0.0.0 255.0.0.0 192.168.2.1`**. Remove the old static route: **`no ip route 192.168.1.0 255.255.255.0 192.168.2.1`**. Exit *Configuration* mode: **`end`**. View the routing table: **`show ip route`**.

6. On the 2600 router, set up a static NAT translation for 2950-1 (10.0.0.1) and the 2950-2 (10.0.0.2). Configure `fa0/0` as the inside and `s0` as the outside for NAT. View your static translations.

   At the top of the simulator in the menu bar, click on the *eRouters* icon and choose *2600*. Access *Configuration* mode: **`configure terminal`**. Set up static NAT statement on the 2600 router for 2950-1: **`ip nat inside source static 192.168.1.4 10.0.0.1`**. Set up static NAT statement on the 2600 router for 2950-2: **`ip nat inside source static 192.168.1.3 10.0.0.2`**. Specify **`fa0/0`** as the inside: **`interface fa0/0`**, **`ip nat inside`**, and **`exit`**. Specify s0 as the outside: **`interface s0`**, **`ip nat outside`**, and **`end`**. View the static translations: **`show ip nat translations`**.

7. Test the translation from Host3 by pinging the two switches with their global and local addresses.

   At the top of the simulator in the menu bar, click on the *eStations* icon and choose *Host3*. From Host3, ping 2950-1's global address: **`ping 10.0.0.1`**. The ping should be successful. From Host3, ping 2950-1's local address: **`ping`**

192.168.1.4. The ping should fail (no route). From Host3, ping 2950-2's global address: **ping 10.0.0.2**. The ping should be successful. From Host3, ping 2950-2's local address: **ping 192.168.1.3**. The ping should fail (no route).

Now you should be more comfortable with configuring address translation on a router. You do not need to save this simulator configuration.

CERTIFICATION OBJECTIVE 14.03

Dynamic Host Configuration Protocol

The Dynamic Host Configuration Protocol (DHCP) allows devices to dynamically acquire their addressing information. DHCP, defined in RFC 2131, is actually based on BOOTP. It is built on a client/server model and defines two components:

- **Server** Delivering host configuration information
- **Client** Requesting and acquiring host configuration information

DHCP provides the following advantages:

- It reduces the amount of configuration on devices.
- It reduces likelihood of configuration errors.
- It gives you more control by centralizing IP addressing information.

exam Watch

Remember the advantages that DHCP provides.

Most networks today employ DHCP because it is easy to implement and manage. Imagine you work for a company that is bought by another company and you must re-address your network, which contains 2,000 devices. If you have manually configured the IP addresses on these machines, then you must manually change each device's configuration. However, if you were using DHCP, you only have to change the configuration on the DHCP servers, and when the clients either reboot or must renew their addressing information, they'll acquire the addressing information from the new scheme.

DHCP Devices and Operation

As mentioned in the last section, DHCP contains two types of devices: servers and clients. IOS routers support both functions. Servers are responsible for assigning addressing information to clients, and clients request addressing information from servers. This section covers the interaction between these two types of devices.

A DHCP server can use three mechanisms, which are described in Table 14-4, when allocating address information. Most DHCP implementations use the dynamic allocation type.

When acquiring addressing information, a DHCP client goes through four steps:

1. A client generates a DHCPDISCOVER broadcast to discover who the DHCP servers are on the LAN segment.
2. All DHCP servers on the segment can respond to the client with a DHCPOFFER unicast message, which offers IP addressing information to the client. If a client receives messages from multiple servers, it chooses one (typically the first one).
3. Upon choosing one of the offers, the client responds to the corresponding server with a DHCPREQUEST message, telling the server that it wants to use the addressing information the server sent. If there is only one server and the server's information conflicts with the client's configuration, the client will respond with a DHCPDECLINE message.
4. The DHCP server responds with a DHCPACK, which is an acknowledgment to the client indicating that it received the DHCPREQUEST message and that the client accepted the addressing information. The server can also respond with a DHCPNACK, which tells the client the offer is no longer valid and the client should request addressing information again. This can happen if the client is tardy in responding with a DHCPREQUEST message after the server generated the DHCPOFFER message.

TABLE 14-4

DHCP Address Allocation Types

Allocation Type	Explanation
Automatic	Server assigns a permanent IP address to the client
Dynamic	Server assigns an IP address to a client for a period of time
Manual	IP address manually configured on the client, and DHCP is used to convey additional addressing information and verification

exam Watch

Remember the four steps that DHCP goes through when a client requests addressing information.

exam Watch

Remember that a DCHP server can assign an IP address; a subnet mask; a default gateway; DNS server, TFTP server, and WINS server addresses; and a domain name.

When a client shuts down gracefully, it can generate a DHCPRELEASE message, telling the server it no longer needs its assigned IP address. Most DHCP configurations involve a lease time, which specifies a time period that the client is allowed to use the address. Upon reaching this time limit, the client must renew its lease or get a new IP address.

DHCPOFFER server messages include the following information: IP address of the client, subnet mask of the segment, IP address of the default gateway, DNS domain name, DNS server address or addresses, WINS server address or addresses, and the TFTP server address or addresses. Please note that this is not an all-encompassing list.

DHCP Server Configuration

Cisco IOS routers can be DHCP servers. Please note, though, that this is not a full-functioning DHCP product and is typically used in small networking environments, such as SOHO or branch offices. Use the following configuration to set up a DHCP server:

```
Router(config)# [no] service dhcp
Router(config)# ip dhcp pool pool_name
Router(config-dhcp)# network network_number
                      [subnet_mask | /prefix_length]
Router(config-dhcp)# domain-name domain_name
Router(config-dhcp)# dns-server IP_address
[IP_address_2...IP_address_8]
Router(config-dhcp)# netbios-name-server IP_address
[IP_address_2...IP_address_8]
Router(config-dhcp)# netbios-node-type node_type
Router(config-dhcp)# default-router IP_address
[IP_address_2...IP_address_8]
Router(config-dhcp)# lease days [hours][minutes] | infinite
Router(config-dhcp)# exit
Router(config)# ip dhcp ping timeout milliseconds
Router(config)# ip dhcp excluded-address beginning_IP_address
[ending_IP_address]
```

The **service dhcp** command enables and disables the DHCP server feature on your router. By default, this is enabled on your router. Precede the command with the **no** parameter to disable it. The **ip dhcp pool** command creates an addressing pool. The name you give this pool must be unique. Notice that when you execute this command, you are placed in *DHCP Subconfiguration* mode.

The **network** command specifies the range of IP addresses to be assigned to clients. You specify a network number followed by either a subnet mask or a slash and the number of networking bits in the network. If you omit the subnet mask value, it defaults to the subnet mask of the Class A, B, or C network.

The **domain-name** command assigns the domain name to the client. The **dns-server** command allows you to assign up to eight DNS server addresses to the client. Separate each address from the next with a space. The **netbios-name-server** command allows you to assign up to eight WINS server addresses to the client. The **netbios-name-type** command assigns the node type to a Microsoft client. This identifies how Microsoft clients perform resolution. These types can be **b** (broadcast only), **p** (WINS only), **m** (broadcast, then WINS), or **h** (WINS, then broadcast). The **default-router** command allows you to assign up to eight default gateway addresses to the client for this range of addresses. The **lease** command specifies the duration of the lease. If you omit this, it defaults to one day. If you specify the **infinite** parameter, the IP address assigned to the client is assigned permanently.

The second to the last command in the preceding code listing is not done within *DHCP Subconfiguration* mode. The **ip dhcp ping timeout** command is used by the DHCP server to test if an available address the server has in its pool is or is not being used. Before a server will send an address in a DHCPOFFER message, it pings the address. This command is used to define how long the server should wait for a reply. By default, this is 500 milliseconds. If the server doesn't receive a reply in this time period, the server will assume the address is not being used and offer this to the client.

The **ip dhcp excluded-address** command excludes addresses from your network pool—these addresses are addresses that are already statically assigned to devices, perhaps servers, on the same segment as the client.

Let's take a look at a simple example of a DHCP server configuration. I'll use the network shown in Figure 14-5. The router in this example is the DHCP server, providing addresses for the 192.168.1.0/24 network.

Here's the configuration:

```
Router(config)# ip dhcp pool dhcppool
Router(config-dhcp)# network 192.168.1.0 255.255.255.0
```

```
Router(config-dhcp)# domain-name thisnetwork.com
Router(config-dhcp)# dns-server 192.168.1.2
Router(config-dhcp)# default-router 192.168.1.1
Router(config-dhcp)# lease 5
Router(config-dhcp)# exit
Router(config)# ip dhcp excluded-address 192.168.1.1
                                         192.168.1.2
```

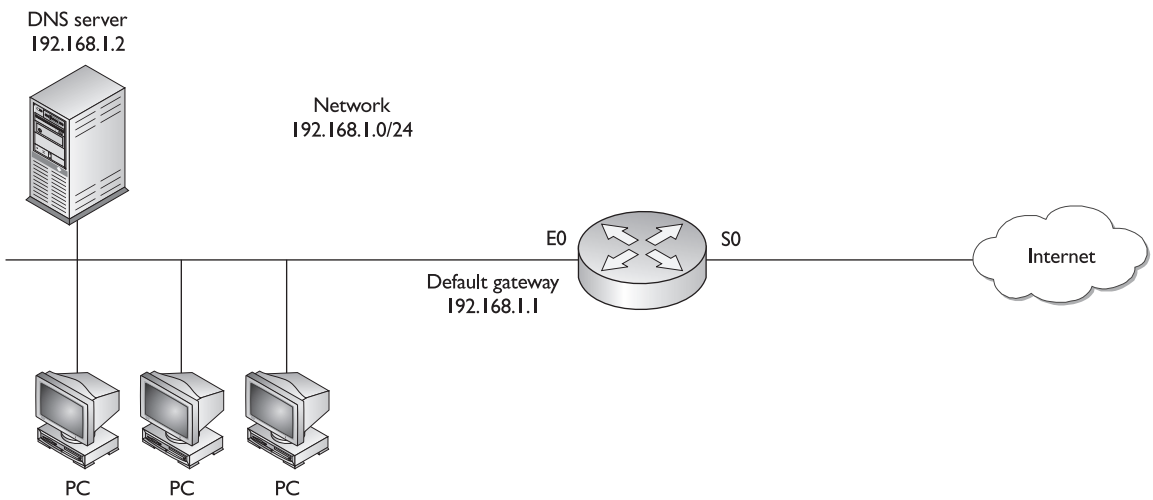
The pool, named *dhcppool*, includes all addresses in network 192.168.1.0/24 with the exception of the two excluded addresses: 192.168.1.1 and 192.168.1.2. The lease length was changed to five days.

exam Watch

The `service dhcp` command enables the DHCP server function on a router.

14.09. The CD contains a multimedia demonstration of a DHCP server configuration on a router.

FIGURE 14-5 DHCP server example



DHCP Client Configuration

You can configure an interface on your IOS router to use DHCP to *acquire* its IP address. This is commonly done if your router is directly connected to your ISP via a dialup, PPPoE, or cable modem connection and your ISP is assigning the addressing information to your router via DHCP. Here is the command to set up a DHCP client on your router:

exam Watch

Use the `ip address dhcp` command on a router's interface in order to use DHCP to acquire the router's IP address on the interface.

```
Router(config)# interface type [slot_#/]port_#
Router(config-if)# ip address dhcp
```

As you can see, this configuration is very easy.

14.10. The CD contains a multimedia demonstration of a DHCP client configuration on a router.

DHCP Verification

Once you have your DHCP server up and running, you can view the addresses assigned to clients with the following command:

```
Router# show ip dhcp binding [client_address]
```

On a DHCP server, you can clear an assigned client address with the following command:

```
Router# clear ip dhcp binding client_address | *
```

Entering an asterisk will clear all of the bounded addresses for clients.

If you are a client, use the **show ip interface brief** or **show interfaces** command to see your dynamically assigned address. For more detailed troubleshooting, you can use the following **debug** command:

```
Router# debug ip dhcp server events|packet|linkage
```

14.11. The CD contains a multimedia demonstration of verifying a DHCP server configuration on a router.

CERTIFICATION SUMMARY

Private addresses are defined in RFC 1918: 10.0.0.0/8, 172.16.0.0/16–172.31.0.0/16, and 192.168.0.0/24–192.168.255.0/24. If you use private addresses, you must have these translated to a public address before these packets reach a public network. Address translation is used when you don't have enough public addresses, you change ISPs but keep you existing addresses, you are merging companies with overlapping address spaces, or you want to assign the same IP address to multiple machines.

The term *inside local IP address* refers to packets with a private, or original IP address. The term *inside global IP address* refers to packets with a public, or translated, address. NAT translates one IP address to another where PAT (address overloading) translates many IP addresses to the same global address, where the source port numbers are changed to ensure the translation device can differentiate the connections. PAR redirects traffic destined to a port on device to a different device.

Address translation allows access to an almost inexhaustible group of addresses and enables you to hide your internal network design from outsiders. It also gives you tighter control over traffic entering and leaving your network. However, address translation adds delay to your traffic, makes troubleshooting more difficult, and won't work with all applications, especially multimedia applications.

The **`ip nat inside source static`** command sets up static NAT. The **`ip nat inside source list`** and **`ip nat pool`** (add **`overload`** to do PAT) commands set up dynamic NAT or PAT. The **`ip nat inside|outside`** *Interface* commands define which interfaces are considered internal and external for address translation.

Use the **`show ip nat translations`** command to view the router's address translation table. The **`clear ip nat translation`** * command clears all dynamic address translation entries in the router's translation table. The **`debug ip nat`** command will show the translations the IOS is doing on every translated packet.

DHCP reduces the amount of configuration on devices, reduces the likelihood of configuration errors, and gives you more control of addressing by centralizing your addressing policies. A client goes through four steps when acquiring addressing information: DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, and DHCPACK.

To enable the DHCP server function on a router, use the **`service`** `dhcp` command. Create your addressing policies with the **`ip dhcp pool`** command and create your address pool with the **`network`** command. To have the router act as a DHCP client, use the **`ip address dhcp`** *Interface* command.

TWO-MINUTE DRILL

Address Translation Overview

- RFC 1918 private addresses include 10.0.0.0/8, 172.16.0.0/16–172.31.0.0/16, and 192.168.0.0/24–192.168.255.0/24
- Reasons to use address translation include not having enough public addresses, changing ISPs, merging networks with overlapping addresses, and representing multiple devices as a single logical device. Disadvantages of address translation include connection delays, difficult troubleshooting, and that it doesn't work with all applications.
- An inside local IP address is a private address assigned to an inside device. An inside global IP address is a public address associated with an inside device.
- NAT does a one-to-one address translation. PAT translates multiple IP addresses to a single address, using the source port number to differentiate connections. Port address redirection is a form of static PAT, where traffic sent to a specific address and port is redirected to another machine (and possibly a different port). PAT can support up to 4,000 translations.

Address Translation Configuration

- To define inside and outside, use the **`ip nat inside|outside`** *Interface Subconfiguration* mode command.
- To configure static NAT, use the **`ip nat inside|outside source static`** command.
- To set up dynamic NAT, use the **`ip nat inside source list`** command, with a standard ACL specifying the inside local addresses. Add **`overload`** to this command to do PAT. Use the **`ip nat pool`** command to specify the global addresses.
- Load distribution allows you to distribute traffic sent to one IP address to multiple IP addresses.
- Use the **`show ip nat translations`** command to view the static and dynamic address translations. Use the **`clear ip nat translation *`** command to clear the dynamic translations from the address translation table. Use **`debug ip nat`** to see the actual translation process.

Dynamic Host Configuration Protocol

- Four steps occur when a client requests addressing information: a client generates a DHCPDISCOVER; servers respond with a DHCPOFFER; the client chooses one reply and responds to the server with a DHCPREQUEST; and the server acknowledges the request with a DHCPACK.
- To create a DHCP server pool, use the **ip** `dhcp pool` command. To specify the range of addresses, use the **network** command. Other commands include **domain-name**, **dns-server**, **netbios-name-server**, **default-router**, and **lease**. The default lease period is 30 days. The default ping timeout is 500 milliseconds.
- To acquire a DHCP address on a router's interface, use the **ip address** `dhcp` command.
- To view the assigned DHCP addresses, use the **show ip dhcp binding** command. To clear DHCP addresses on the server, use the **clear ip dhcp binding** * command.

SELF TEST

The following Self Test questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully, as there may be more than one correct answer. Choose all correct answers for each question.

Address Translation Overview

1. Which of the following is a private address?
   - A. 192.169.7.17
   - B. 172.32.28.39
   - C. 10.1.256.8
   - D. 172.16.255.89
2. Which of the following reasons might you need to use address translation?
   - A. You have to use public addressing because your ISP didn't assign you enough private addresses.
   - B. You are using private addresses but have changed ISPs, and your new ISP won't support these private addresses.
   - C. You want to assign the same IP address to multiple machines so that users on the Internet see this offered service as a single logical computer.
   - D. You are merging two companies that use different address spaces.
3. Private addressing is defined in RFC _____ (enter a number).
4. _____ only translates one (and only one) IP address to another.
   - A. NAT
   - B. PAT
   - C. PAR
   - D. NAT and PAT
5. An _____ is a public IP address associated with an inside device.
   - A. Inside global IP address
   - B. Inside local IP address
   - C. Outside global IP address
   - D. Outside local IP address

Address Translation Configuration

6. Which command is used to define the local addresses that are statically translated to global addresses?
   A. **`ip nat inside source static`**
   B. **`ip nat inside`**
   C. **`ip nat inside source list`**
   D. **`ip nat pool`**
7. Enter the router command to specify an interface to be considered as *outside* for address translation: _____.
8. Enter the router command to view the address translation table: _____.
9. When configuring the **`ip`** `nat inside source` command, which parameter must you specify to perform PAT?
   A. **`pat`**
   B. **`overload`**
   C. **`load`**
   D. **`port`**
10. Enter the router command to view, in real-time, the address translations the IOS is performing: _____.
11. Which router command clears all of the static translations in the address translation table?
   A. **`erase ip nat translation`** *
   B. **`clear ip nat translation`** *
   C. **`clear ip nat translation all`**
   D. None of these commands
12. _____ allows you to distribute connection requests destined to a single IP address to multiple machines.
   A. Traffic load
   B. Traffic configuration
   C. PAT
   D. Load distribution

Dynamic Host Configuration Protocol

13. Which of the following is not an advantage of DHCP?

A. It allows for the use of private addressing.
B. It reduces the amount of configuration on devices.
C. It reduces likelihood of configuration errors.
D. It gives you more control by centralizing IP addressing information.

14. What command creates a DHCP address pool on a router?

A. **`ip dhcp`**
B. **`ip pool dhcp`**
C. **`ip dhcp pool network`**
D. **`ip dhcp pool`**

15. Enter the router command to have a router's interface acquire its IP address via DHCP: _____.

SELF TEST ANSWERS

Address Translation Overview

1. ☑ **D.** The address 172.16.255.89 is a private address.
   ☒ **A** and **B** are public addresses. **C** is an invalid address (256 is an invalid value).
2. ☑ **C.** You will need to use address translation (load distribution) if you want to assign the same IP address to multiple machines so that users on the Internet see this offered service as a single logical computer.
   ☒ **A** is not true, because it reverses the word public and private. **B** is not true, because it refers to private, not public addresses. **D** is not true, because it should say the same, not different, address spaces.
3. ☑ Private addressing is defined in RFC 1918.
4. ☑ **A.** NAT only translates one IP address to another.
   ☒ **B** and **D** are incorrect because PAT translates many addresses to one address. **C** is incorrect because PAR can translate a port number to another port number.
5. ☑ **A.** An inside global IP address is a public IP address assigned to an inside device.
   ☒ **B** refers to an inside private address. **C** refers to an outside public address. **D** refers to an outside private address.

Address Translation Configuration

6. ☑ **A.** The **ip** `nat inside source static` command configures static NAT translations
   ☒ **B** specifies an interface as being inside. **C** and **D** are used to configure dynamic NAT.
7. **SYMBOL 254 \f "Wingdings" \s 11 ip nat outside.**
8. **SYMBOL 254 \f "Wingdings" \s 11 show ip nat translations.**
9. ☑ **B**. Use the **overload** parameter with the **ip nat inside source**e command to set up PAT.
   ☒ **A**, **C**, and **D** are invalid parameters.
10. **SYMBOL 254 \f "Wingdings" \s 11 debug ip nat**.
11. ☑ **D**. The only way to remove static entries is to remove your static NAT commands from the router's configuration with the **no** parameter.
    ☒ **A** and **C** are invalid commands. **B** removes the dynamic entries from the address translation table.

12. ☑ **D**. Load distribution allows you to distribute connection requests destined to a single IP address to multiple machines.
☒ **A** and **B** are invalid terms. **D** refers to translating many addresses to a single address, changing the source port number to ensure uniqueness among connections.

Dynamic Host Configuration Protocol

13. ☑ **A.** Actually, DHCP allows for both private and public addressing—this is not an advantage of DHCP, just a function of it.
☒ **B**, **C**, and **D** are advantages of using DHCP.

14. ☑ **D**. The **`ip dhcp pool`** command creates a DHCP address pool.
☒ **A**, **B**, and **C** are invalid commands.

15. **`SYMBOL 254 \f "Wingdings" \s 11`** ip **`address dhcp`**.