Chapter 11

NAT

Introduction

There is a limited supply of Internet Protocol (IP) version 4 addresses. In the early 1990s, many experts believed that we would run out of IP addresses (if the Internet didn't collapse under the weight of too many IP networks first). Today, IPv4 is no longer in immediate danger of failing, thanks to new technologies and enhancements. One of the technologies that helped IPv4 stave off address depletion is *Network Address Translation* (*NAT*).

NAT, as defined in RFC 1631, is the process of swapping one address for another in the IP packet header. In practice, NAT is used to allow hosts that are privately addressed to access the Internet.

NAT is particularly effective when connecting a small office or home office (SOHO) to the corporate network. By using NAT, a company does not have to allocate a real IP address for each of its remote users.

This chapter provides an overview of NAT, and describes how to configure NAT functions, including static NAT, dynamic NAT, NAT overload, and TCP distribution. Finally, this chapter discusses the drawbacks of NAT and how its operation can be monitored using the Cisco IOS.

NAT Overview

Strictly speaking, NAT is the process of altering the IP header of a packet so that the destination address, the source address, or both addresses are replaced in the header by different addresses. This swapping process is performed by a router with specialized NAT software or hardware. Such a NAT-enabled device is often called a NAT *box* because it can be a Cisco router, a UNIX system, a Windows host, or any number of other systems.

A NAT-enabled device typically operates at the border of a *stub domain*. A stub domain is a network that has a single connection to the outside world. Figure 11-1 presents a simple example of a stub domain. When a host inside the stub domain, such as 10.1.1.6, wants to transmit to a host on the outside, it forwards the packet to its default gateway. In this case, the host's default gateway is also the NAT box.

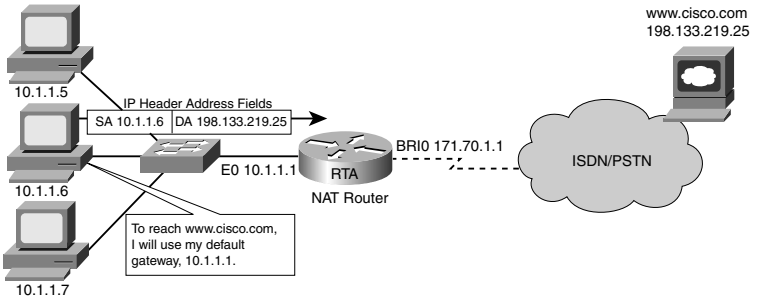


Figure 11-1
A host inside the stub domain sends packets to the Internet via a NAT router.

The NAT process then looks inside the IP header and, if appropriate, replaces the local IP address with a globally unique IP address. Figure 11-2 illustrates this address translation. RTA, the NAT router, determines that the source IP address of the packet (10.1.1.6) should be swapped. In this case, RTA replaces the private address with a global (real) address, 171.70.2.1. RTA also keeps a record of this translation in a NAT translation table.

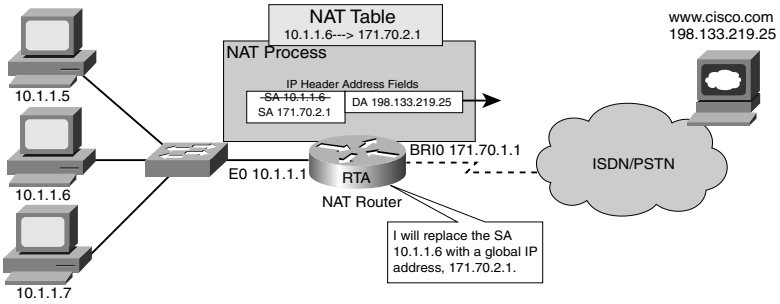


Figure 11-2
RTA keeps a table that contains an entry for each translation.

When an outside host sends a response (see Figure 11-3), the NAT router receives it, checks the current table of network address translations, and replaces the destination address with the original inside source address (see Figure 11-4).

NAT translations can occur dynamically or statically and can be used for a variety of purposes. The following sections describe these key NAT concepts and configurations:

- Private addressing
- NAT terminology
- Dynamic NAT
- Static NAT
- NAT overload
- TCP load distribution

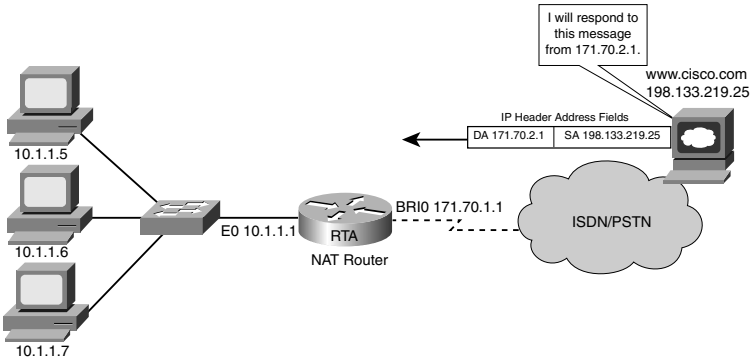


Figure 11-3
An outside host replies to translated address 171.70.2.1.

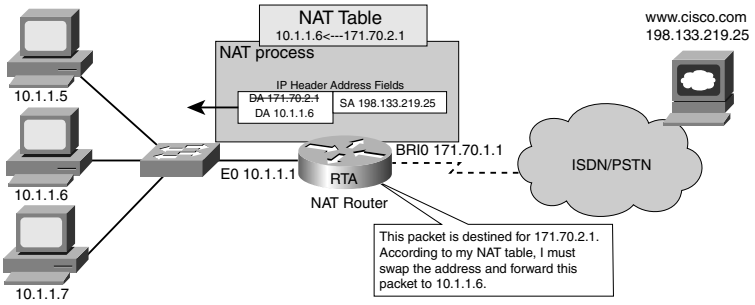


Figure 11-4
RTA checks incoming packets on BRI0 against its translation table and swaps the destination address, if necessary.

Private IP Addresses

RFC 1918 sets aside three blocks of IP addresses—a Class A, a Class B, and a Class C range—for private, internal use (see Table 11-1). These three ranges provide more than 17 million addresses for internal use.

Table 11-1 RFC 1918 Addresses

Class	RFC 1918 Internal Address Range	CIDR Prefix
A	10.0.0.0–10.255.255.255	10.0.0.0/8
B	172.16.0.0–172.31.255.255	172.16.0.0/12
C	192.168.0.0–192.168.255.255	192.168.0.0/16

Public addresses must be registered by an organization or leased from a provider (almost always for a fee). On the other hand, private IP addresses are set aside to be used by anyone. That means two networks, or two million networks, can each use the same private address. The restriction is that private addresses cannot be used on the public Internet. A private address cannot be used on the Internet because ISPs typically configure their routers to prevent privately addressed customer traffic from being forwarded.

NAT provides tremendous benefits to individual companies and the Internet, as well. Before NAT, a host with a private address could not access the Internet. With NAT, individual companies can address some, or all, of their hosts with private addresses and then use NAT to access the public Internet. At the same time, these hosts connect to the Internet without necessarily depleting its address space.

NAT Terminology

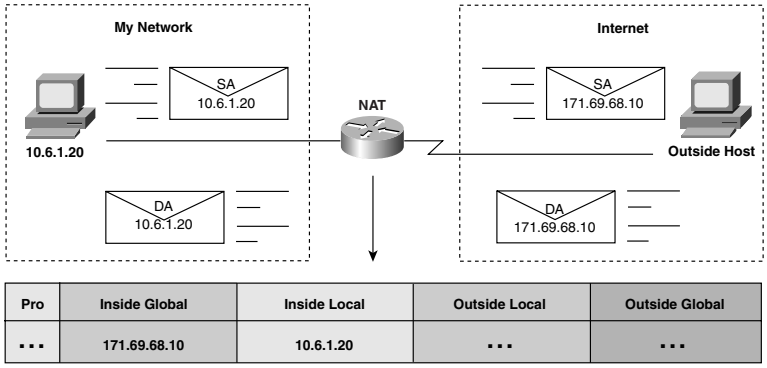
When configuring NAT using the Cisco IOS, it's critical that you understand NAT terminology. In particular, you must have a strong grasp of the following terms:

- **Inside addresses**—The set of networks that are subject to translation. Inside addresses are typically RFC 1918 addresses, but they can be any valid IP address.
- **Outside addresses**—All other addresses. Usually, these are valid addresses located on the Internet.

Inside addresses are associated with hosts inside the NAT boundary, regardless of whether they are private (RFC 1918) or public addresses. Inside addresses are part of your network. Outside addresses are typically associated with all Internet addresses. However, in some cases, outside addresses can be associated with hosts on your own network, beyond the NAT boundary. Two different kinds of inside addresses and two different types of outside addresses exist:

- *Inside local address*—Configured IP address assigned to a host on the inside network. Address might be globally unique, allocated out of the private address space defined in RFC 1918, or officially allocated to another organization (see Figure 11-5).
- *Inside global address*—The IP address of an inside host as it appears to the outside network (see Figure 11-5). The inside global address is the translated address. These addresses are typically allocated from a globally unique address space, typically provided by the Internet Service Provider (ISP) (if the enterprise is connected to the global Internet).
- *Outside local address*—The IP address of an outside host as it appears to the inside network. These addresses can be allocated from the RFC 1918 space if desired (see Figure 11-6).
- *Outside global address*—The configured IP address assigned to a host in the outside network (see Figure 11-6).

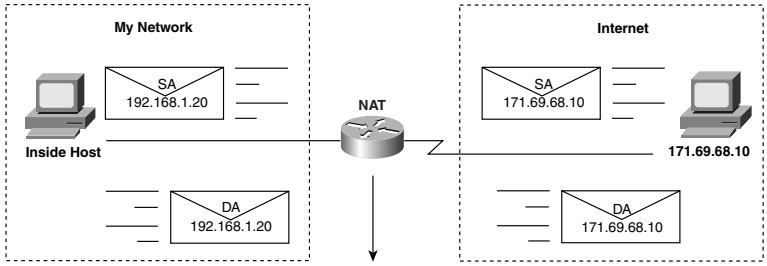
Figure 11-5
Typically, inside local addresses are private addresses that are translated into real inside global addresses.



Pro	Inside Global	Inside Local	Outside Local	Outside Global
...	171.69.68.10	10.6.1.20	...	...

10.6.1.20 is Inside Local Address
171.69.68.10 is Inside Global Address

Figure 11-6
NAT translates outside local addresses, which are addresses that your inside hosts use to reach outside hosts.



Pro	Inside Global	Inside Local	Outside Local	Outside Global
...	...	...	192.168.1.20	171.69.68.10

192.168.1.20 is Outside Local Address
171.69.10 is Outside Global Address

Dynamic NAT

With *dynamic NAT*, translations don't exist in the NAT translation table until the router receives traffic that requires translation (such traffic is defined by an administrator). Dynamic translations are temporary and eventually time out.

For example, host 10.4.1.1 transmits a packet to an Internet host, as shown in Figure 11-7. Because a private address can't be routed on the Internet, this host uses the services of a router configured for NAT.

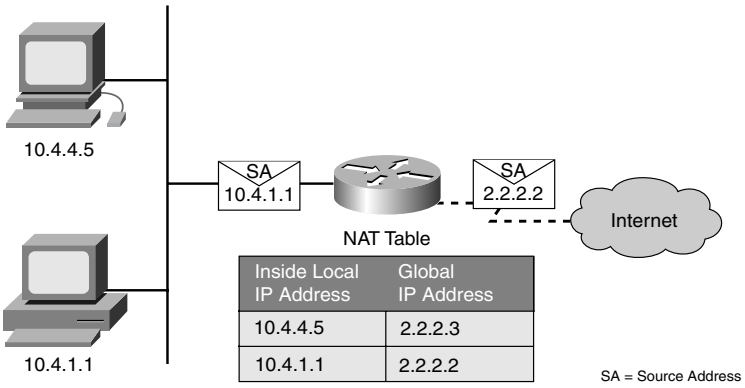


Figure 11-7 NAT routers keep a table that maps global IP addresses to private, internal addresses.

The NAT router alters the IP packet by removing the original source address, 10.4.1.1, and replacing it with a globally unique address from a pool defined by an administrator.

As shown in Figure 11-7, the inside host is dynamically assigned 2.2.2.2 from the address pool. The NAT box keeps a record of this address translation in its NAT table. When an Internet host's reply packet is sent to 2.2.2.2, it arrives at the NAT router, which checks its NAT table for the mapping to the local inside address. The NAT router then replaces the destination address with the original local address, 10.4.1.1. The translation mapping is not permanent; it ages out after a configurable period of time.

When configuring dynamic NAT, you typically create a pool of global addresses to be allocated as needed. Use the **ip nat pool** command to configure the address pool, as shown here:

```
Router(config)#ip nat pool name start-ip end-ip {netmask netmask | prefix-length
  prefix-length}
```

You must also specify which packets should be translated. Typically, you specify packets matching a certain range of source addresses to be translated. Use the **access-list** global configuration command to create an access list to match addresses that the router should translate:

```
Router(config)#access-list access-list-number permit source [source-wildcard]
```

To establish a dynamic translation based on source address, use the **ip nat inside source list** command:

```
Router(config)#ip nat inside source list access-list-number pool name
```

This command must specify the number of the access list.

Finally, you must configure at least one interface on the router as the inside interface by using the following interface configuration command:

```
Router(config-if)#ip nat inside
```

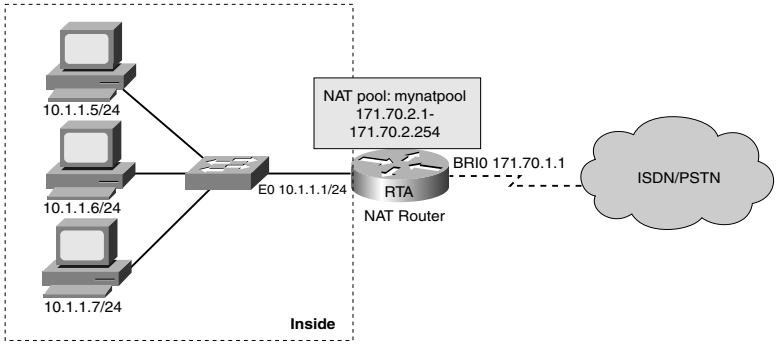
The router only creates dynamic entries in the translation table for packets arriving on interfaces configured with the **ip nat inside** command.

Use the **ip nat outside** command to mark an interface as an outside interface:

```
Router(config-if)#ip nat outside
```

Following these steps, you can configure RTA; the NAT router is shown in Figure 11-8.

Figure 11-8
RTA is configured for dynamic NAT.



NOTE

When using the **ip nat pool** command, you have the option of specifying the subnet mask or the prefix length. The **netmask** keyword uses a dotted-decimal argument, such as 255.255.255.0. A 24-bit mask can also be specified using the **prefix-length 24** command.

First, define the NAT pool, as shown in Example 11-1.

Example 11-1 Defining a NAT Pool

```
RTA(config)#ip nat pool mynatpool 171.70.2.1 171.70.2.254 netmask 255.255.255.0
```

This command creates a pool of global addresses called **mynatpool** that can be used by inside local hosts. But which local hosts are allowed to use this pool? Example 11-2 uses an access list to match the source addresses to be translated.

Example 11-2 Using Access Lists with NAT Pools

```
RTA(config)#access-list 24 permit 10.1.1.0 0.0.0.255
RTA(config)#ip nat inside source list 24 pool mynatpool
```

The last command configures the router to use **access-list 24** to decide whether to translate the IP source address using **mynatpool**.

As the final configuration steps on the NAT router, Example 11-3 configures the appropriate interfaces to take on the role of **outside** and **inside**.

Example 11-3 Configuring NAT Interfaces

```
RTA(config)#interface bri0
RTA(config-if)#ip nat outside
RTA(config-if)#interface e0
RTA(config-if)#ip nat inside
```

If the host at 10.1.1.6 sends an IP packet to an outside host, such as 4.4.4.1, RTA translates the source address and creates a NAT table entry. Use the **show ip nat translations** command, as shown in Example 11-4, to view the translation table.

Example 11-4 Using the show ip nat translations Command

```
RTA#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 171.70.2.1         10.1.1.6           ---                ---
```

Example 11-4 shows that the inside local address 10.1.1.6 has been translated to the inside global address 171.70.2.1. Although this table entry exists, outside hosts can use the global IP address 171.70.2.1 to reach the host at 10.1.1.6.

On a Cisco router, dynamic NAT table entries remain in the table for 24 hours by default. After the entry ages out, outside hosts can no longer reach 10.1.1.6 until a new table entry is created. The table entry can be created only from the inside.

A 24-hour timeout is relatively long. You can adjust the translation timeout using the following command:

```
Router(config)#ip nat translation timeout seconds
```

One of the primary advantages to dynamic NAT is the ability to serve a large number of hosts with a smaller number of globally routable IP addresses. It is important for translation table entries to timeout so that addresses in the pool become available again for other hosts.

For example, you could configure a pool of 30 inside global addresses for 250 inside local hosts. Only 30 of the inside hosts can use a global address at any one time. This configuration might work well in an environment where outside (Internet) connectivity is infrequent and short-lived. If the inside hosts are using outside connections for occasional Web surfing or e-mail, this configuration might be appropriate. However, if translation table entries don't age out fast enough, the entire pool of addresses could be in use, and additional hosts would be unable to access the Internet. To serve a large number of hosts with just a handful of addresses, you might have to use address overloading (see the section, "NAT Overload").

Although NAT is not a security firewall, it can prevent outsiders from initiating connections with inside hosts, unless a permanent global address mapping exists in the NAT table (static NAT). Because outside hosts never see the "pretranslated" inside addresses, NAT has the effect of hiding the inside structure of a network.

Static NAT

Static translation occurs when you specifically configure addresses in a lookup table. A specific inside local address maps to a prespecified outside global address. The inside and outside addresses are statically mapped one for one. This means that for every inside local address, static NAT requires an inside global address. If you used static NAT exclusively, you would not conserve real IP addresses.

For this reason, static NAT is typically used in conjunction with dynamic NAT, in cases where you have overlapping networks (see the section, "Overlapping Networks" later in this chapter), or in cases when you have changed from one numbering scheme to another (if you change providers, etc.).

Consider how static NAT can be used in conjunction with dynamic NAT. Company XYZ uses dynamic NAT to allow inside hosts to access the Internet. But what if the company wants outside users to access an internally addressed Web server? Without a permanent global address, outside hosts are unable to consistently access the server.

Company XYZ can statically map a global address (170.70.2.10) to an inside address (10.1.1.7). Static mappings exist in the NAT table until an administrator removes them. Internet hosts, and Domain Name System (DNS), can use the global address (170.70.2.10) to access the privately addressed Web server.

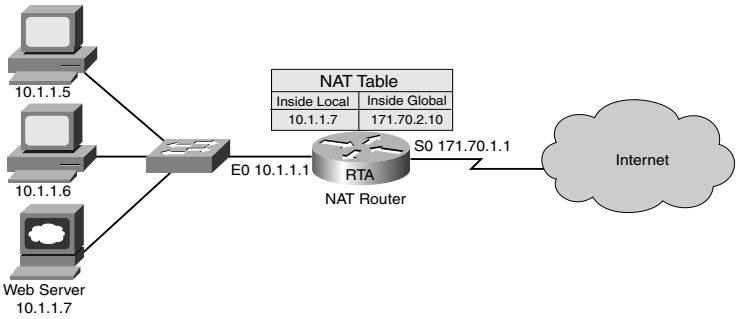


Figure 11-9
A static NAT configuration allows Internet hosts to access the Web server (10.1.1.7) by using the inside global address 171.70.2.10.

Use the following command to configure a static translation between an inside local address and an inside global address:

```
Router(config)#ip nat inside source static local-ip global-ip
```

After you configure the static mapping(s), you must specify an inside and outside interface, as shown in Example 11-5.

Example 11-5 Configuring Static NAT

```
RTA(config)#ip nat inside source static 10.1.1.7 172.70.2.10
RTA(config)#interface bri0
RTA(config-if)#ip nat outside
RTA(config-if)#interface e0
RTA(config-if)#ip nat inside
```

The ability to create static mappings makes NAT a useful tool if Company XYZ were ever to change providers. If the company moves from one ISP to another, it might have to completely readdress its systems. Instead of readdressing, NAT can be deployed to temporarily translate the old addresses to new ones, with static mappings in place to keep Web and other public services available to the outside.

NAT Overload

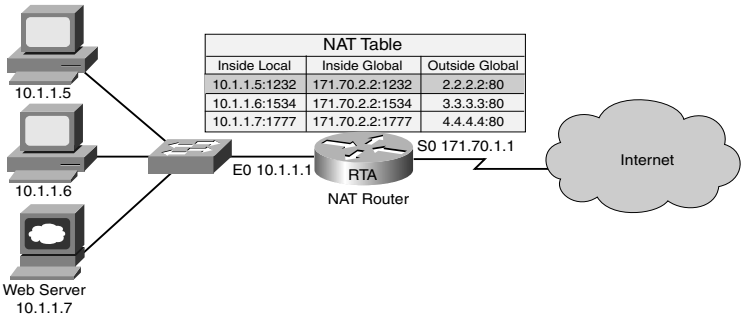
One of the most powerful features of NAT routers is their capability to use *Port Address Translation* (*PAT*), which allows multiple inside addresses to map to the same global address. This is sometimes called a many-to-one NAT, or *address overloading*. With address overloading, literally hundreds of privately addressed nodes can access the Internet using a single global address. The NAT router keeps track of the different conversations by mapping TCP and UDP port numbers in the translation table. A

translation entry that maps one IP address and port pair to another is called an *extended table entry*.

For example, Figure 11-10 shows three inside nodes using the same translated global address of 170.70.2.2. Each of these hosts can communicate with different Internet hosts, or even with the same outside host.

According to the NAT table shown in Figure 11-10, RTA translates the packet from the inside local address, 10.1.1.5, TCP port 1232. The translated inside global address is 171.70.2.2, also on port 1232.

Figure 11-10
With NAT overload, TCP and UDP port numbers can be used to keep track of address translations in the NAT table.



With the outside host at 2.2.2.2, TCP port 80 replies to the address 171.70.2.2, on port 1232. When RTA (the NAT router) receives this reply, it uses the destination port number to determine whether the destination IP address should be translated to 10.1.1.5, 10.1.1.6, or 10.1.1.7.

As long as the inside global port numbers are unique for each inside local host, NAT overload works. For example, if the host at 10.1.1.5 and 10.1.1.6 both use TCP port 1234, the NAT router can create the extended table entries mapping 10.1.1.5:1234 to 171.70.2.2:1234 and 10.1.1.6:1234 to 171.70.2.2:1235. NAT implementations don't necessarily try to preserve the original port number.

NOTE

TCP and UDP use a 16-bit field to represent port numbers. Port numbers range from 1 to 65,535. Port numbers are used by applications to send data and listen for connections. Client applications, such as Web browsers, typically use port numbers in the range 1024 to 65,535.

NAT FOR THE HOME USER

The advent of broadband Internet connections in the home has had an interesting side effect: Home users are networking their own computers with the goal of sharing a single high-speed Internet connection. Recognizing this new market, vendors quickly packaged affordable NAT solutions for the home. Some products are stand-alone solutions, or appliances, that provide many-to-one NAT along with Dynamic Host Configuration Protocol (DHCP) services—all with little or no configuration. Other products are software solutions that run on a computer with multiple logical addresses. Notable among these software solutions is Internet Connection Sharing, which ships with Microsoft's latest versions of Windows. With NAT now built into the world's most ubiquitous operating system, you can bet its popularity will only increase.

NAT overload can go a long way to alleviate address depletion, but its capabilities are limited. Over 65,000 inside addresses can theoretically map to a single outside address. However, the actual number of translations supported by a Cisco router varies.

NAT overload can be used in conjunction with dynamic mappings to a NAT pool. A NAT device, such as a Cisco PIX Firewall, can use a one-to-one dynamic mapping until the available addresses are almost depleted, at which time NAT can overload the remaining address or addresses. However, on a Cisco IOS router, NAT overloads the first address in the pool until it's maxed out and then moves on to the second address, and so on.

Configure NAT overload by using the keyword **overload**:

```
Router(config)#ip nat inside source list access-list-number pool name overload
```

RTA (Figure 11-10) is configured as shown in Example 11-6.

Example 11-6 Configuring NAT Overload

```
RTA(config)#ip nat pool mypatpool 171.70.2.1 171.70.2.30 netmask 255.255.255.0
RTA(config)#access-list 24 permit 10.1.1.0 0.0.0.255
RTA(config)#ip nat inside source list 24 pool mypatpool overload
RTA(config)#interface serial 0
RTA(config-if)#ip nat outside
RTA(config-if)#interface ethernet 0
RTA(config-if)#ip nat inside
```

The **ip nat pool** command creates the pool of addresses that are used for overloading. This pool, **mypatpool**, contains only 30 addresses. Using NAT overload, these 30 addresses can comfortably serve hundreds, or even thousands, of inside hosts. The **access-list** command creates the access list that is used to match addresses that are to be translated. The **ip nat inside source list 24** command configures the router to translate addresses that match access list 24 using inside global addresses from **mypatpool**.

You do not necessarily have to configure an address pool in order for NAT overload to work. If you don't have any available IP addresses, you can overload the address of the outside interface, as shown here:

```
Router(config)#ip nat inside source list access-list-number interface interface-
  name overload
```

Typically, home users receive only one IP address from their provider. Figure 11-11 shows how NAT overload can be configured using the outside interface.

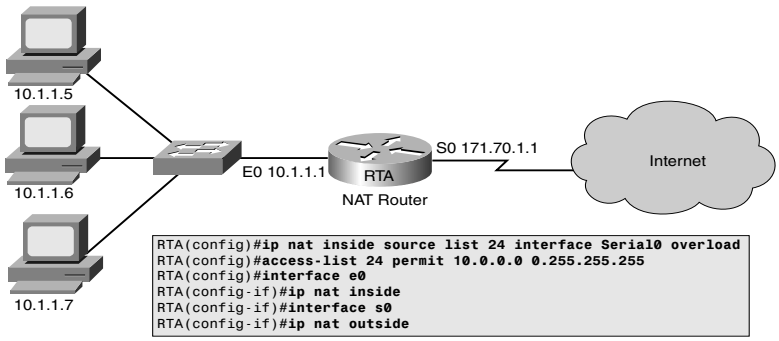


Figure 11-11
You can configure NAT to overload the outside interface's address.

TCP Load Distribution

As an extension to static mapping, Cisco routers support *TCP load distribution*, a powerful NAT feature that allows you to map one global address to multiple inside addresses for the purpose of distributing conversations among multiple (usually mirrored) hosts. In Figure 11-12, the NAT router rotates conversations between two inside Web servers at 10.1.1.6 and 10.1.1.7 when an outside host requests Web services at 171.70.2.10.

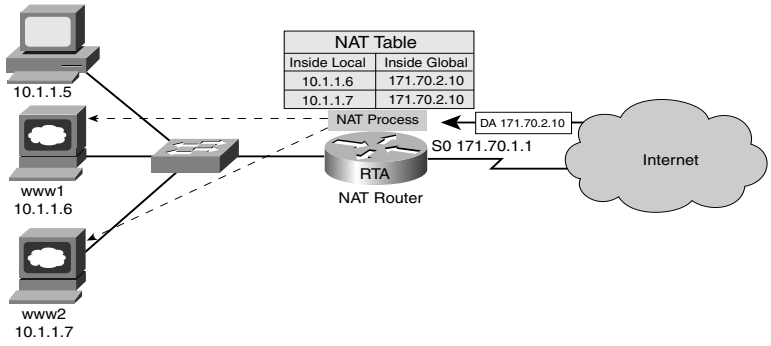


Figure 11-12
TCP load distribution assigns multiple local addresses to a single global address.

TCP load distribution can be used even if you are not translating between private addresses and public addresses. The scenario depicted in Figure 11-13 shows that RTA is configured to map both www1 (171.70.2.3/24) and www2 (171.70.2.4/24) to the same inside global IP address (171.70.2.10/24). All three of these IP addresses are public addresses on the same subnet. In such configurations, the address 171.70.2.10 is referred to as a *virtual host*.

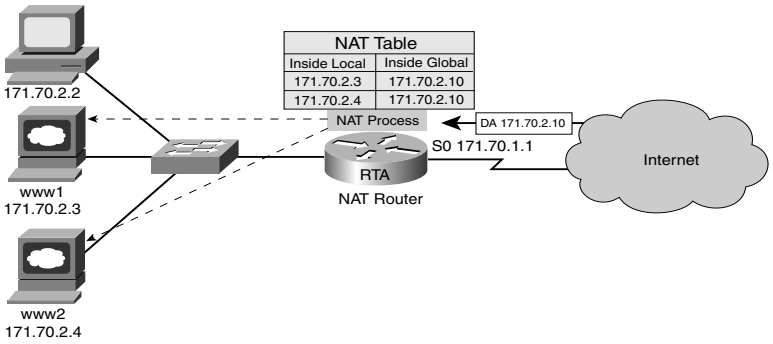


Figure 11-13
Traffic to the virtual host, 171.70.2.10, is distributed between www1 (171.70.2.3) and www2 (171.70.2.4).

The following are the steps for configuring a TCP load distribution:

1. Define a pool of addresses containing the addresses of the real hosts:
   ```
   Router(config)#ip nat pool name start-ip end-ip {netmask netmask |
     prefix-length prefix-length} type rotary
   ```
2. Define an access list permitting the address of the virtual host:
   ```
   Router(config)#access-list access-list-number permit source [source-wildcard]
   ```
3. Establish dynamic inside destination translation, identifying the access list defined in Step 2:
   ```
   Router(config)#ip nat inside destination list access-list-number pool name
   ```
4. Specify the inside interface:
   ```
   Router(config)#interface type number
   ```
5. Mark the interface as connected to the inside:
   ```
   Router(config-if)#ip nat inside
   ```
6. Specify the outside interface:
   ```
   Router(config-if)#interface type number
   ```
7. Mark the interface as connected to the outside:
   ```
   Router(config-if)#ip nat outside
   ```

Using the commands in these steps, RTA in Figure 11-13 is configured as shown in Example 11-7.

Example 11-7 Configuring TCP Load Distribution

```
RTA(config)#ip nat pool webservers 171.70.2.3 171.70.2.4 netmask 255.255.255.0
  type rotary
RTA(config)#access-list 46 permit host 171.70.2.10
RTA(config)#ip nat inside destination list 46 pool webservers
RTA(config)#interface e0
```

Example 11-7 Configuring TCP Load Distribution (Continued)

```
RTA(config-if)#ip nat inside
RTA(config-if)#interface s0
RTA(config)#ip nat outside
```

The keyword **rotary** is used so that the router rotates through the **webservers** pool when translating. Access list 46 is used to define the virtual host address.

RTA is configured to translate destination addresses that match 171.70.2.10 (access list 46), using the **webservers** pool. Because the **webservers** pool was defined using the **rotary** keyword, the first translation is to 171.70.2.3, the second is to 171.70.2.4, the third back to 171.70.2.3, and so on. In this way, the load is distributed among the Web servers.

Overlapping Networks

Overlapping networks result when you assign an IP address to a device on your network that is already legally owned and assigned to a different device on the Internet or outside network. Overlapping networks also result when two companies, both of whom use RFC 1918 IP addresses in their networks, merge. These two networks need to communicate, preferably without having to readdress all their devices.

Figure 11-14 illustrates an overlapping network scenario. The inside device, HostA, is addressed using the same IP subnet as the outside device, HostZ. HostA can't reach HostZ by using HostZ's IP address. If HostA pings 10.1.1.6, it will be pinging its local neighbor and not HostZ.

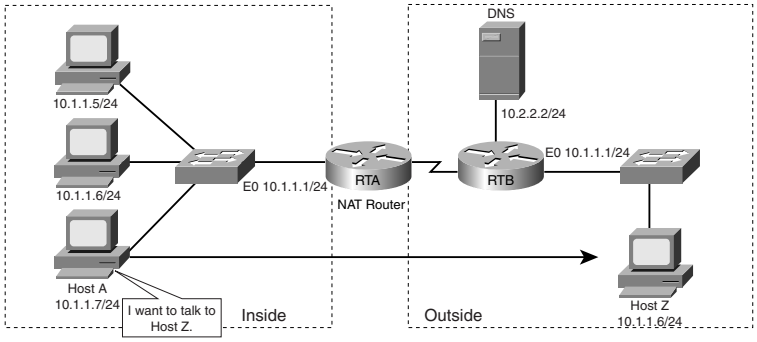
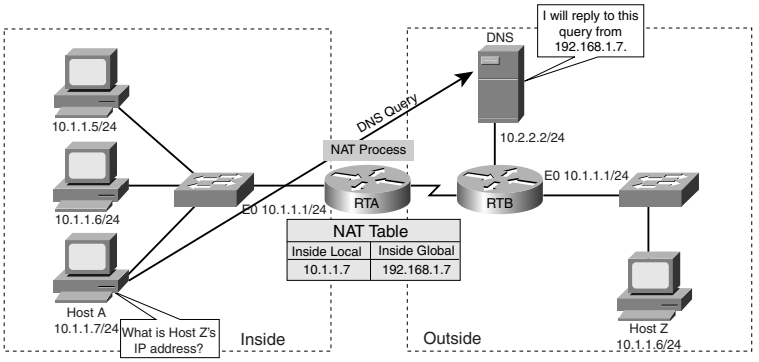


Figure 11-14 HostA wants to establish an IP connection with HostZ but cannot use HostZ's actual IP address (10.1.1.6).

One way to allow HostA to communicate with HostZ is to use DNS and NAT. Instead of using HostZ's actual IP address, HostA can use HostZ's host name. For example, a

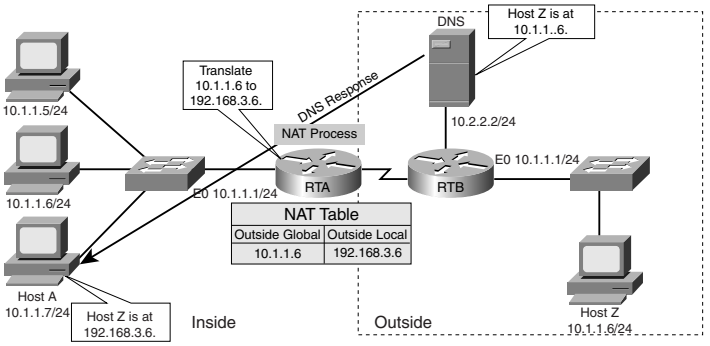
user on HostA could issue the command **ping HostZ**, which would result in a name-to-address lookup using DNS (see Figure 11-15).

Figure 11-15
HostA's DNS request is translated by NAT.



A NAT translation is for the DNS query sourced from 10.1.1.7. The query from 10.1.1.7 is translated by RTA so that it appears to be from the inside global address 192.168.1.7. The DNS server responds to this query, as shown in Figure 11-16.

Figure 11-16
In an overlapping networks scenario, DNS responses are translated by NAT.



NOTE

NAT doesn't look at the payload of the DNS reply unless translation occurs on the IP header of the reply packet.

This DNS response is the key to making overlapping networks coexist. The DNS server responds with HostZ's actual IP address, 10.1.1.6; however, RTA translates the payload of the DNS response. Thus, Cisco's implementation of NAT actually alters the contents of a DNS packet, creating a simple table entry and mapping the outside global address, 10.1.1.6, to an outside local address, 192.168.3.6. In this way, HostA believes that HostZ is at 192.168.3.6 (presumably, a reachable IP network).

HostA can then begin a conversation with HostZ. When HostA sends a packet to HostZ, RTA creates an extended table entry, as shown in Figure 11-17. From HostA's point of view, this conversation is between 10.1.1.7 (HostA) and 192.168.3.6 (HostZ).

However, both the source and destination addresses are translated by RTA so that HostZ believes this same conversation is between 192.168.1.7 (HostA) and 10.1.1.6 (HostZ).

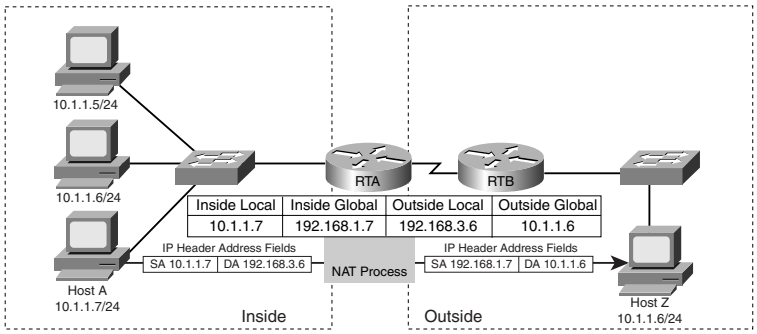


Figure 11-17
From HostA's point of view, the packet shown is sourced from 10.1.1.7, but after the translation, HostZ sees the same packet as sourced from 192.168.1.7.

The configuration for RTA is shown in Example 11-8.

Example 11-8 Configuring NAT for Overlapping Networks

```
RTA(config)#ip nat pool inGlobal 192.168.1.1 192.168.1.254 prefix-length 24
RTA(config)#ip nat pool outLocal 192.168.3.1 192.168.3.254 prefix-length 24
RTA(config)#ip nat inside source list 2 pool inGlobal
RTA(config)#ip nat outside source list 2 pool outLocal
RTA(config)#access-list 2 permit 10.1.1.0 0.0.0.255
RTA(config)#interface e0
RTA(config-if)#ip nat inside
RTA(config-if)#interface s0
RTA(config)#ip nat outside
```

RTA uses the **inGlobal** address pool to translate HostA's address so that outside hosts can reach HostA. RTA uses the **outLocal** pool to translate outside hosts in the overlapping network so that HostA can reach those hosts. Example 11-9 provides the output of the **show ip nat translations** command after HostA has sent HostZ an IP packet.

Example 11-9 Output of show ip nat translations in an Overlapping Network Scenario

```
RTA#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 192.168.1.7        10.1.1.7           ---                ---
--- ---                ---                192.168.3.6        10.1.1.6
--- 192.168.1.7        10.1.1.7           192.168.3.6        10.1.1.6
```

The first entry was created when HostA sent a DNS query. The second entry was created when RTA translated the payload of the DNS reply. The third entry was created when the packet was exchanged between HostA and HostZ. The third entry is a summary of the first two entries and is used for more efficient translations.

Verifying NAT Configurations

You can display translation information and clear address translation entries from the NAT translation with the commands covered in this section.

The **show ip nat translations** [**verbose**] command can be used to verify the active translations, as shown in Examples 11-4 and 11-9. The **verbose** keyword can be used with this command to display more information, including the time remaining for a dynamic entry, as shown in Example 11-10.

Example 11-10 Using the show ip nat translations verbose Command

```
RTX#show ip nat translation verbose
Pro Inside global      Inside local       Outside local      Outside global
icmp 42.0.0.55:1536    192.168.0.21:1536  10.0.0.5:1536      10.0.0.5:1536
    create 00:00:09, use 00:00:06, left 00:00:53,
    flags:
extended, use_count: 0
```

You can use the **show ip nat statistics** command to see NAT statistics, as shown in Example 11-11.

Example 11-11 Using the show ip nat statistics Command

```
SanJose1#show ip nat statistics
Total active translations: 3 (3 static, 0 dynamic; 0 extended)
Outside interfaces:
  Serial0/0
Inside interfaces:
  FastEthernet0/0
Hits: 4  Misses: 0
Expired translations: 0
Dynamic mappings:
```

If you need to use a trace on NAT operation, you can use the **debug ip nat** command to display a line of output for each packet that gets translated. You can include the **detailed** keyword to output even more information. The output shown in Example 11-12 is a sample of a debug of address translation inside to outside.

Example 11-12 Using the debug ip nat Command

```
RTX#debug ip nat
IP NAT debugging is on
06:37:40: NAT:  s=192.168.0.21->42.0.0.55, d=10.0.0.5 [63]
06:37:40: NAT*: s=10.0.0.5, d=42.0.0.55->192.168.0.21 [63]
06:37:41: NAT*: s=192.168.0.21->42.0.0.55, d=10.0.0.5 [64]
06:37:41: NAT*: s=10.0.0.5, d=42.0.0.55->192.168.0.21 [64]
06:37:42: NAT*: s=192.168.0.21->42.0.0.55, d=10.0.0.5 [65]
06:37:42: NAT*: s=10.0.0.5, d=42.0.0.55->192.168.0.21 [65]
06:37:43: NAT*: s=192.168.0.21->42.0.0.55, d=10.0.0.5 [66]
06:37:43: NAT*: s=10.0.0.5, d=42.0.0.55->192.168.0.21 [66]
06:38:43: NAT: expiring 42.0.0.55 (192.168.0.21) icmp 1536 (1536)
```

You can decode the debug output in Example 11-12 by using the following key points:

- The asterisk next to **NAT** indicates that the translation is occurring in the fast path. The first packet in a conversation always goes through the slow path (process-switched). The remaining packets go through the fast path if a cache entry exists.
- **s = a.b.c.d** is the source address.
- **d = a.b.c.d** is the destination address.
- **a.b.c.d -> w.x.y.z** indicates that the address was translated.
- The value in brackets is the IP identification number. This information could be useful for debugging because it enables you to correlate with other packet traces from sniffers.

To clear all translated entries, use the **clear ip nat translation** *command**.**

You can clear a simple translation entry containing an inside translation, or both an inside and outside translation, by using the **clear ip nat translation inside** *global-ip local-ip* [**outside** *local-ip global-ip*] command.

You can clear a simple translation entry that contains an outside translation by using the **clear ip nat translation outside** *local-ip global-ip* command.

If you want to clear an extended entry (in its various forms), use the **clear ip nat translation** *protocol* **inside** *global-ip global-port local-ip local-port* [**outside** *local-ip local-port global-ip global-port*] command. Example 11-13 shows the use of this command.

Example 11-13 Using the clear ip nat translations Command

```
RTX#clear ip nat translations udp inside 192.168.2.2 1220 10.1.1.2 1220 outside
  171.69.2.132 53 171.69.2.132 53
```

If NAT is properly configured but translations are not occurring, clear the NAT translations and check to see if the translations occur.

NAT Considerations

NAT has several advantages, including the following:

- NAT conserves the legally registered addressing scheme by allowing the privatization of intranets, yet it allows legal addressing scheme pools to be set up to gain access to the Internet.
- NAT also reduces the instances in which addressing schemes overlap. If a scheme was originally set up within a private network, the network was connected to the public network (which might use the same addressing scheme). Without address translation, the potential for overlap exists globally.
- NAT increases the flexibility of connection to the public network. Multiple pools, backup pools, and load sharing/balancing pools can be implemented to help ensure reliable public network connections. Network design is also simplified because planners have more flexibility when creating an address plan.
- Deprivatization of a network requires the renumbering of the existing network; the costs can be associated with the number of hosts that require conversion to the new addressing scheme. NAT allows the existing scheme to remain, and it still supports the new assigned addressing scheme outside the private network.

NAT is not without drawbacks. The tradeoff for address translation is a loss of functionality, particularly with any protocol or application that involves sending IP address information outside the IP header. NAT disadvantages include the following:

- NAT increases delay. Switching path delays are introduced because of the translation of each IP address within the packet headers. Performance might be a consideration because NAT is currently accomplished by using process switching. The CPU must look at every packet to decide whether it has to translate it, and then alter the IP header—and possibly the TCP header. It is not likely that this process will be easily cacheable.

- One significant disadvantage when implementing and using NAT is the loss of end-to-end IP traceability. It becomes much more difficult to trace packets that undergo numerous packet-address changes over multiple NAT hops. This scenario does, however, lead to more secure links because hackers who want to determine the source of a packet will find it difficult, if not impossible, to trace or obtain the original source or destination address.
- NAT also forces some applications that use IP addressing to stop functioning because it hides end-to-end IP addresses. Applications that use physical addresses instead of a qualified domain name will not reach destinations that are translated across the NAT router. Sometimes, this problem can be avoided by implementing static NAT mappings.

The following traffic types are supported by Cisco IOS NAT:

- Any TCP/UDP traffic that does not carry source or destination IP addresses in the application data stream
- Hypertext Transfer Protocol (HTTP)
- Trivial File Transfer Protocol (TFTP)
- Telnet
- Archie
- Finger
- Network Timing Protocol (NTP)
- Network File System (NFS)
- rlogin, rsh, rcp

Although the following traffic types carry IP addresses in the application data stream, they are supported by Cisco IOS NAT:

- ICMP
- File Transfer Protocol (FTP) (including PORT and PASV commands)
- NetBIOS over TCP/IP (datagram, name, and session services)
- Progressive Networks' RealAudio
- White Pines' CuSeeMe
- Xing Technologies' Streamworks
- DNS "A" and "PTR" queries
- H.323/NetMeeting [12.0(1)/12.0(1)T and later]
- VDOLive [11.3(4)11.3(4)T and later]
- Vxtreme [11.3(4)11.3(4)T and later]
- IP multicast [12.0(1)T] (source address translation only)

The following traffic types are not supported by Cisco IOS NAT:

- Routing table updates
- DNS zone transfers
- BOOTP
- talk, ntalk
- Simple Network Management Protocol (SNMP)
- NetShow

Summary

In this chapter, you learned that NAT allows your network to scale without depleting your limited supply of global IP addresses. You learned how to configure static NAT, and you learned about dynamic NAT and NAT overload (PAT). You also saw how NAT can be used to provide connectivity in overlapping IP networks.

Review Questions

Use the following review questions to test your understanding of the concepts covered in this chapter. In some cases, there is more than one correct answer, so choose all that apply. Answers are listed in Appendix A, "Answers to Review Questions."

1. Which of the following are not RFC 1918 addresses?
   - **A.** 1.1.1.1
   - **B.** 172.31.255.221
   - **C.** 192.168.192.192
   - **D.** 172.168.0.1
2. Which of the following is true about static NAT?
   - **A.** Static NAT translations timeout after 24 hours by default.
   - **B.** Static NAT translations are assigned from an address pool.
   - **C.** Static NAT maps one local address to one global address.
   - **D.** Cisco routers use static NAT by default.
3. Which command correctly configures S0 as an outside interface?
   - **A.** (config)**#ip nat outside s0**
   - **B.** (config)**#ip nat s0**
   - **C.** (config-if)**#ip nat outside**
   - **D.** (config-if)**#ip nat s0 out**
4. Which of the following commands correctly configures NAT overload?
   - **A.** (config)**#ip nat inside source list 24 interface s0 overload**
   - **B.** (config)**#ip nat inside source 24 pool mypool overload**
   - **C.** (config)**#ip nat source 24 pool mypool overload**
   - **D.** (config)**#ip nat source list 24 pool overload**
5. For overlapping networks, which of the following is true about NAT:
   - **A.** NAT can be used only in overlapping networks that use the same major network number, but not the same subnets.
   - **B.** NAT translates the payload of DNS requests to facilitate name lookups between overlapping networks.

C. NAT cannot be used with overlapping networks.

D. NAT can modify only the IP header of a DNS query and response.

6. Which command modifies the timeout for dynamic NAT translations?

A. (config-if)**#ip nat timeout 1200**

B. (config)**#ip nat timeout 120000**

C. (config-if)**#ip nat translation timeout 1200000**

D. (config)**#ip nat translation timeout 12000000**

7. Which command can be used to clear all NAT translations?

A. **clear nat translations all**

B. **clear ip nat translations all**

C. **clear ip nat translations ***

D. **clear nat translations ***

8. What is one disadvantage of using NAT?

A. Using NAT accelerates IP address depletion.

B. NAT is not compatible with ICMP.

C. NAT increases latency.

D. NAT cannot support RFC 1918 outside global addresses.

9. Which of the following commands will output the NAT translations table?

A. **show ip nat translations table**

B. **show ip nat translations**

C. **show ip nat translations ***

D. **show ip nat database**

10. Which of the following commands configures TCP load distribution?

A. RTA(config)**#ip nat load-distribution**

B. RTA(config)**#ip nat pool webservers 171.70.2.3 171.70.2.4 netmask 255.255.255.0 type overload**

C. RTA(config)**#ip nat pool webservers overload**

D. RTA(config)**#ip nat pool webservers 171.70.2.3 171.70.2.4 netmask 255.255.255.0 type rotary**

Key Terms

Inside global address In NAT, the IP address of an inside host as it appears to the outside network. The inside global address is the translated address. These addresses are typically allocated from a globally unique address space, typically provided by the Internet Service Provider (ISP) (if the enterprise is connected to the global Internet).

Inside local address In NAT, the configured IP address assigned to a host on the inside network. The address might be globally unique, allocated out of the private address space defined in RFC 1918, or officially allocated to another organization.

NAT (Network Address Translation) Mechanism for reducing the need for globally unique IP addresses. NAT allows an organization with addresses that are not globally unique to connect to the Internet by translating those addresses into globally routable address space.

Outside global address In NAT, the configured IP address assigned to a host in the outside network.

Outside local address In NAT, the IP address of an outside host as it appears to the inside network. These addresses can be allocated from the RFC 1918 space if desired.

Overlapping networks Two or more networks using addresses from the same public or private IP address space.

Stub domain A network that has a single connection to the outside world.

PAT (Port Address Translation) A NAT process that maps multiple inside addresses to the same global address by using port numbers to keep track of the translations. This is sometimes called a many-to-one NAT, or *address overloading*.