

Configuring Network Address Translation

Two of the key problems facing the Internet are depletion of IP address space and scaling in routing. Network Address Translation (NAT) is a feature that allows an organization's IP network to appear from the outside to use a different IP address space than what it is actually using. Thus, NAT allows an organization with nonglobally routable addresses to connect to the Internet by translating those addresses into a globally routable address space. NAT also allows a more graceful renumbering strategy for organizations that are changing service providers or voluntarily renumbering into classless interdomain routing (CIDR) blocks. NAT is also described in RFC 1631.

For a complete description of the NAT commands in this chapter, refer to the "Network Address Translation Commands" chapter of the *Dial Solutions Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

NAT Business Applications

NAT has several applications. Use it for the following purposes:

- You want to connect to the Internet, but not all your hosts have globally unique IP addresses. NAT enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT is configured on the router at the border of a stub domain (referred to as the *inside network*) and a public network such as the Internet (referred to as the *outside network*). NAT translates the internal local addresses to globally unique IP addresses before sending packets to the outside network.
- You must change your internal addresses. Instead of changing them, which can be a considerable amount of work, you can translate them by using NAT.
- You want to do basic load sharing of TCP traffic. You can map a single global IP address to many local IP addresses by using the TCP load distribution feature.

As a solution to the connectivity problem, NAT is practical only when relatively few hosts in a stub domain communicate outside of the domain at the same time. When this is the case, only a small subset of the IP addresses in the domain must be translated into globally unique IP addresses when outside communication is necessary, and these addresses can be reused when no longer in use.

Benefits of NAT

A significant advantage of NAT is that it can be configured without requiring changes to hosts or routers other than those few routers on which NAT will be configured. As discussed previously, NAT may not be practical if large numbers of hosts in the stub domain communicate outside of the domain. Furthermore, some applications use embedded IP addresses in such a way that it is impractical for a NAT device to translate. These applications may not work transparently or at all through a NAT device. NAT also hides the identity of hosts, which may be an advantage or a disadvantage.

A router configured with NAT will have at least one interface to the inside and one to the outside. In a typical environment, NAT is configured at the exit router between a stub domain and backbone. When a packet is leaving the domain, NAT translates the locally significant source address into a globally unique address. When a packet is entering the domain, NAT translates the globally unique destination address into a local address. If more than one exit point exists, each NAT must have the same translation table. If the software cannot allocate an address because it has run out of addresses, it drops the packet and sends an ICMP Host Unreachable packet.

A router configured with NAT must not advertise the local networks to the outside. However, routing information that NAT receives from the outside can be advertised in the stub domain as usual.

NAT Terminology

As mentioned previously, the term *inside* refers to those networks that are owned by an organization and that must be translated. Inside this domain, hosts will have address in the one address space, while on the outside, they will appear to have addresses in a another address space when NAT is configured. The first address space is referred to as the *local* address space while the second is referred to as the *global* address space.

Similarly, *outside* refers to those networks to which the stub network connects, and which are generally not under the organization's control. As will be described later, hosts in outside networks can be subject to translation also, and can, thus, have local and global addresses.

To summarize, NAT uses the following definitions:

- **inside local address**—The IP address that is assigned to a host on the inside network. The address is probably not a legitimate IP address assigned by the Network Information Center (NIC) or service provider.
- **inside global address**—A legitimate IP address (assigned by the NIC or service provider) that represents one or more inside local IP addresses to the outside world.
- **outside local address**—The IP address of an outside host as it appears to the inside network. Not necessarily a legitimate address, it was allocated from an address space routable on the inside.
- **outside global address**—The IP address assigned to a host on the outside network by the host's owner. The address was allocated from globally routable address or network space.

NAT Configuration Task List

Before configuring any NAT translation, you must know your inside local addresses and inside global addresses. The following sections discuss how you can use NAT to perform optional tasks:

- Translate Inside Source Addresses
- Overload an Inside Global Address
- Translate Overlapping Addresses
- Provide TCP Load Distribution
- Change Translation Timeouts
- Monitor and Maintain NAT

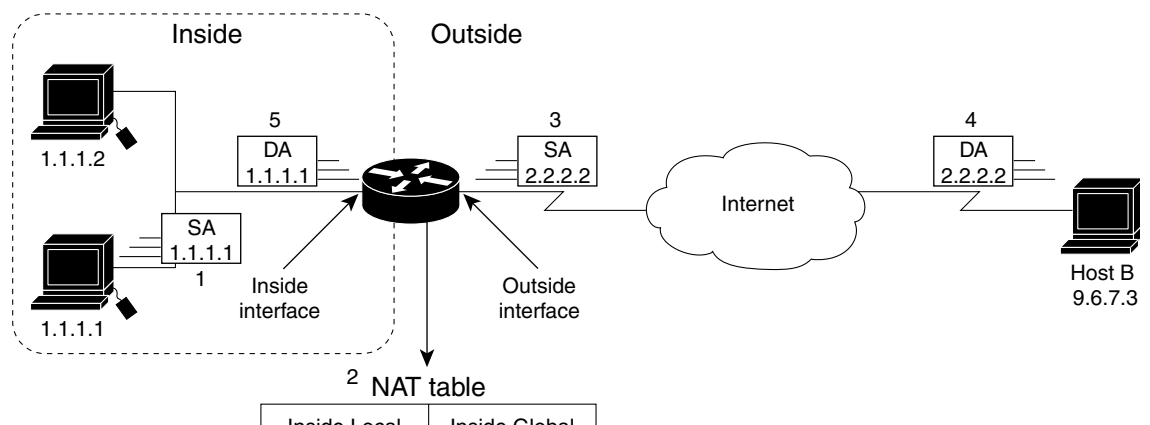
Translate Inside Source Addresses

Use this feature to translate your own IP addresses into globally unique IP addresses when communicating outside of your network. You can configure static or dynamic inside source translation as follows:

- *Static translation* establishes a one-to-one mapping between your inside local address and an inside global address. Static translation is useful when a host on the inside must be accessible by a fixed address from the outside.
- *Dynamic translation* establishes a mapping between an inside local address and a pool of global addresses.

Figure 130 illustrates a router that is translating a source address inside a network to a source address outside the network.

Figure 130 NAT Inside Source Translation



The following process describes inside source address translation, as shown in Figure 130:

- 1 The user at Host 1.1.1.1 opens a connection to Host B.
- 2 The first packet that the router receives from Host 1.1.1.1 causes the router to check its NAT table.
 - If a static translation entry was configured, the router goes to Step 3.
 - If no translation entry exists, the router determines that source address (SA) 1.1.1.1 must be translated dynamically, selects a legal, global address from the dynamic address pool, and creates a translation entry. This type of entry is called a *simple entry*.
- 3 The router replaces the inside local source address of Host 1.1.1.1 with the translation entry's global address, and forwards the packet.
- 4 Host B receives the packet and responds to Host 1.1.1.1 by using the inside global IP destination address (DA) 2.2.2.2.
- 5 When the router receives the packet with the inside global IP address, it performs a NAT table lookup by using the inside global address as a key. It then translates the address to the inside local address of Host 1.1.1.1 and forwards the packet to Host 1.1.1.1.
- 6 Host 1.1.1.1 receives the packet and continues the conversation. The router performs Steps 2 through 5 for each packet.

Configure Static Translation

To configure static inside source address translation, perform the following tasks beginning in global configuration mode:

Task	Command
Establish static translation between an inside local address and an inside global address.	ip nat inside source static <i>local-ip global-ip</i>
Specify the inside interface.	interface <i>type number</i>
Mark the interface as connected to the inside.	ip nat inside
Specify the outside interface.	interface <i>type number</i>
Mark the interface as connected to the outside.	ip nat outside

The previous steps are the minimum you must configure. You could configure multiple inside and outside interfaces.

Configure Dynamic Translation

To configure dynamic inside source address translation, perform the following tasks beginning in global configuration mode:

Task	Command
Define a pool of global addresses to be allocated as needed.	ip nat pool <i>name start-ip end-ip {netmask netmask prefix-length prefix-length}</i>
Define a standard access list permitting those addresses that are to be translated.	access-list <i>access-list-number permit source [source-wildcard]</i>
Establish dynamic source translation, specifying the access list defined in the prior step.	ip nat inside source list <i>access-list-number pool name</i>
Specify the inside interface.	interface <i>type number</i>

Task	Command
Mark the interface as connected to the inside.	ip nat inside
Specify the outside interface.	interface <i>type number</i>
Mark the interface as connected to the outside.	ip nat outside

Note The access list must permit only those addresses that are to be translated. (Remember that there is an implicit “deny all” at the end of each access-list.) An access list that is too permissive can lead to unpredictable results.

The following example translates all source addresses passing access list 1 (having a source address from 192.168.1.0/24) to an address from the pool named net-208. The pool contains addresses from 171.69.233.208 to 171.69.233.233.

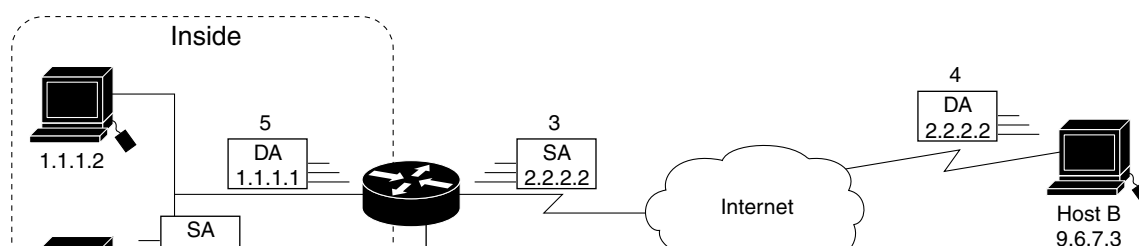
```
ip nat pool net-208 171.69.233.208 171.69.233.233 netmask 255.255.255.240
ip nat inside source list 1 pool net-208
!
interface serial 0
 ip address 171.69.232.182 255.255.255.240
 ip nat outside
!
interface ethernet 0
 ip address 192.168.1.94 255.255.255.0
 ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255
```

Overload an Inside Global Address

You can conserve addresses in the inside global address pool by allowing the router to use one global address for many local addresses. When this overloading is configured, the router maintains enough information from higher-level protocols (for example, TCP or UDP port numbers) to translate the global address back to the correct local address. When multiple local addresses map to one global address, each the TCP or UDP port numbers of each inside host distinguish between the local addresses.

Figure 131 illustrates NAT operation when one inside global address represents multiple inside local addresses. The TCP port numbers act as differentiators.

Figure 131 NAT Overloading Inside Global Addresses



The router performs the following process in overloading inside global addresses, as shown in Figure 131. Both Host B and Host C think they are talking to a single host at address 2.2.2.2. They are actually talking to different hosts; the port number is the differentiator. In fact, many inside hosts could share the inside global IP address by using many port numbers.

- 1 The user at Host 1.1.1.1 opens a connection to Host B.
- 2 The first packet that the router receives from Host 1.1.1.1 causes the router to check its NAT table. If no translation entry exists, the router determines that address 1.1.1.1 must be translated, and sets up a translation of inside local address 1.1.1.1 to a legal global address. If overloading is enabled, and another translation is active, the router reuses the global address from that translation and saves enough information to be able to translate back. This type of entry is called an *extended entry*.
- 3 The router replaces the inside local source address 1.1.1.1 with the selected global address and forwards the packet.
- 4 Host B receives the packet and responds to Host 1.1.1.1 by using the inside global IP address 2.2.2.2.
- 5 When the router receives the packet with the inside global IP address, it performs a NAT table lookup, using the protocol, inside global address and port, and outside address and port as a key, translates the address to inside local address 1.1.1.1, and forwards the packet to Host 1.1.1.1.
- 6 Host 1.1.1.1 receives the packet and continues the conversation. The router performs Steps 2 through 5 for each packet.

To configure overloading of inside global addresses, perform the following tasks beginning in global configuration mode:

Task	Command
Define a pool of global addresses to be allocated as needed.	ip nat pool <i>name start-ip end-ip {netmask netmask prefix-length prefix-length}</i>
Define a standard access list.	access-list <i>access-list-number permit source [source-wildcard]</i>
Establish dynamic source translation, identifying the access list defined in the prior step.	ip nat inside source list <i>access-list-number pool name overload</i>
Specify the inside interface.	interface <i>type number</i>
Mark the interface as connected to the inside.	ip nat inside
Specify the outside interface.	interface <i>type number</i>
Mark the interface as connected to the outside.	ip nat outside

Note The access list must permit only those addresses that are to be translated. (Remember that there is an implicit “deny all” at the end of each access list.) An access list that is too permissive can lead to unpredictable results.

The following example creates a pool of addresses named net-208. The pool contains addresses from 171.69.233.208 to 171.69.233.233. Access list 1 allows packets having the source address from 192.168.1.0 to 192.168.1.255. If no translation exists, packets matching access list 1 are translated to an address from the pool. The router allows multiple local addresses (192.168.1.0 to 192.168.1.255) to use the same global address. The router retains port numbers to differentiate the connections.

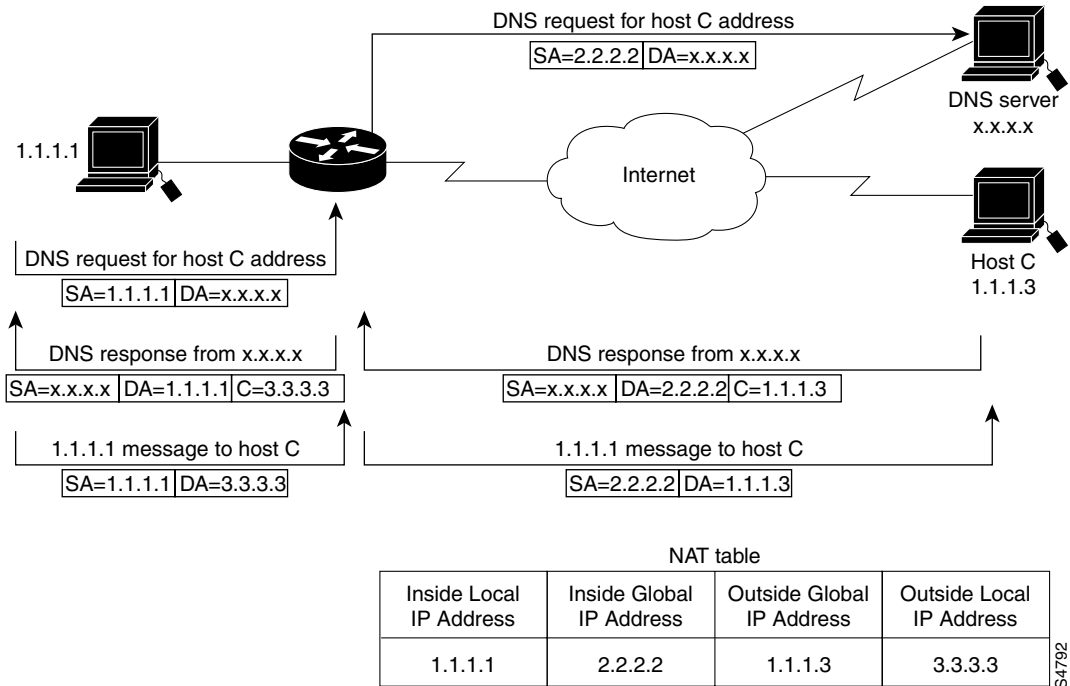
```
ip nat pool net-208 171.69.233.208 171.69.233.233 netmask 255.255.255.240
ip nat inside source list 1 pool net-208 overload
!
interface serial0
 ip address 171.69.232.182 255.255.255.240
 ip nat outside
!
interface ethernet0
 ip address 192.168.1.94 255.255.255.0
 ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255
```

Translate Overlapping Addresses

The NAT overview discusses translating IP addresses, perhaps because your IP addresses are not legal, officially assigned IP addresses. Perhaps you chose IP addresses that officially belong to another network. The case of an address used both illegally and legally is called *overlapping*. You can use NAT to translate inside addresses that overlap with outside addresses. Use this feature if your IP addresses in the stub network are legitimate IP addresses belonging to another network, and you want to communicate with those hosts or routers.

Figure 132 shows how NAT translates overlapping networks.

Figure 132 NAT Translating Overlapping Addresses



The router performs the following process when translating overlapping addresses:

- 1 The user at Host 1.1.1.1 opens a connection to Host C by name, requesting a name-to-address lookup from a DNS server.
- 2 The router intercepts the DNS reply and translates the returned address if there is an overlap (that is, the resulting legal address resides illegally in the inside network). To translate the return address, the router creates a simple translation entry mapping the overlapping address 1.1.1.3 to an address from a separately configured, outside local address pool.
The router examines every DNS reply from everywhere, ensuring that the IP address is not in the stub network. If it is, the router translates the address.
- 3 Host 1.1.1.1 opens a connection to 3.3.3.3.
- 4 The router sets up translations mapping inside local and global addresses to each other, and outside global and local addresses to each other.
- 5 The router replaces the source address with the inside global address and replaces the destination address with the outside global address.
- 6 Host C receives the packet and continues the conversation.
- 7 The router does a lookup, replaces the destination address with the inside local address, and replaces the source address with the outside local address.
- 8 Host 1.1.1.1 receives the packet and the conversation continues, using this translation process.

Configure Static Translation

To configure static outside source address translation, perform the following tasks beginning in global configuration mode:

Task	Command
Establish static translation between an outside local address and an outside global address.	ip nat outside source static <i>global-ip local-ip</i>
Specify the inside interface.	interface <i>type number</i>
Mark the interface as connected to the inside.	ip nat inside
Specify the outside interface.	interface <i>type number</i>
Mark the interface as connected to the outside.	ip nat outside

Configure Dynamic Translation

To configure dynamic outside source address translation, perform the following tasks beginning in global configuration mode.

Task	Command
Define a pool of local addresses to be allocated as needed.	ip nat pool <i>name start-ip end-ip {netmask netmask prefix-length prefix-length}</i>
Define a standard access list.	access-list <i>access-list-number permit source [source-wildcard]</i>
Establish dynamic outside source translation, specifying the access list defined in the prior step.	ip nat outside source list <i>access-list-number pool name</i>
Specify the inside interface.	interface <i>type number</i>
Mark the interface as connected to the inside.	ip nat inside
Specify the outside interface.	interface <i>type number</i>
Mark the interface as connected to the outside.	ip nat outside

Note The access list must permit only those addresses that are to be translated. (Remember that there is an implicit “deny all” at the end of each access list.) An access list that is too permissive can lead to unpredictable results.

In the following example, the addresses in the local network are being used legitimately by someone else on the Internet. An extra translation is required to access that external network. Pool net-10 is a pool of outside local IP addresses. The statement `ip nat outside source list 1 pool net-10` translates the addresses of hosts from the outside overlapping network to addresses in that pool.

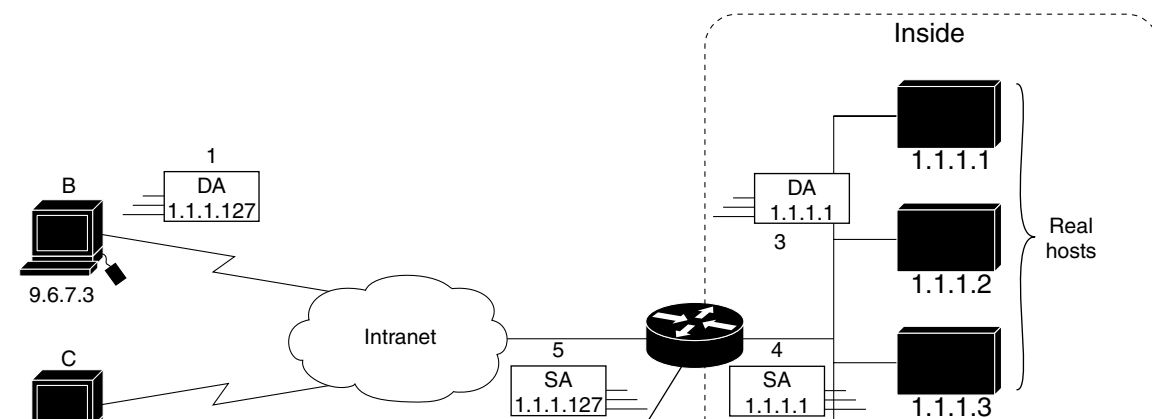
```
ip nat pool net-208 171.69.233.208 171.69.233.223 prefix-length 28
ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24
ip nat inside source list 1 pool net-208
ip nat outside source list 1 pool net-10
!
interface serial 0
 ip address 171.69.232.192 255.255.255.240
 ip nat outside
!
```

```
interface ethernet0
 ip address 192.168.1.94 255.255.255.0
 ip nat inside
 access-list 1 permit 192.168.1.0 0.0.0.255
```

Provide TCP Load Distribution

Another use of NAT is unrelated to Internet addresses. Your organization may have multiple hosts that must communicate with a heavily used host. Using NAT, you can establish a virtual host on the inside network that coordinates load sharing among real hosts. Destination addresses that match an access list are replaced with addresses from a rotary pool. Allocation is done in a round-robin basis, and only when a new connection is opened from the outside to the inside. Non-TCP traffic is passed untranslated (unless other translations are in effect). Figure 133 illustrates this feature.

Figure 133 NAT TCP Load Distribution



The router performs the following process when translating rotary addresses:

- 1 The user on Host B (9.6.7.3) opens a connection to virtual host at 1.1.1.127.
- 2 The router receives the connection request and creates a new translation, allocating the next real host (1.1.1.1) for the inside local IP address.
- 3 The router replaces the destination address with the selected real host address and forwards the packet.
- 4 Host 1.1.1.1 receives the packet and responds.
- 5 The router receives the packet, performs a NAT table lookup using the inside local address and port number, and the outside address and port number as the key. The router then translates the source address to the address of the virtual host and forwards the packet.

The next connection request will cause the router to allocate 1.1.1.2 for the inside local address.

To configure destination address rotary translation, perform the following tasks beginning in global configuration mode. This allows you to map one virtual host to many real hosts. Each new TCP session opened with the virtual host will be translated into a session with a different real host.

Task	Command
Define a pool of addresses containing the addresses of the real hosts.	ip nat pool <i>name start-ip end-ip</i> { netmask <i>netmask</i> prefix-length <i>prefix-length</i> } type rotary
Define an access list permitting the address of the virtual host.	access-list <i>access-list-number</i> permit <i>source</i> [<i>source-wildcard</i>]
Establish dynamic inside destination translation, identifying the access list defined in the prior step.	ip nat inside destination list <i>access-list-number</i> pool <i>name</i>
Specify the inside interface.	interface <i>type number</i>
Mark the interface as connected to the inside.	ip nat inside
Specify the outside interface.	interface <i>type number</i>
Mark the interface as connected to the outside.	ip nat outside

Note The access list must permit only those addresses that are to be translated. (Remember that there is an implicit “deny all” at the end of each access list.) An access list that is too permissive can lead to unpredictable results.

In the following example, the goal is to define a virtual address, connections to which are distributed among a set of real hosts. The pool defines the addresses of the real hosts. The access list defines the virtual address. If a translation does not already exist, TCP packets from serial 0 (the outside interface) whose destination matches the access list are translated to an address from the pool.

```
ip nat pool real-hosts 192.168.15.2 192.168.15.15 prefix-length 28 type rotary
ip nat inside destination list 2 pool real-hosts
!
interface serial 0
 ip address 192.168.15.129 255.255.255.240
 ip nat outside
!
interface ethernet 0
 ip address 192.168.15.17 255.255.255.240
 ip nat inside
!
access-list 2 permit 192.168.15.1
```

Change Translation Timeouts

By default, dynamic address translations time out after some period of non-use. You can change the default values on timeouts, if necessary. When overloading is not configured, simple translation entries time out after 24 hours. To change this value, perform the following task in global configuration mode:

Task	Command
Change the timeout value for dynamic address translations that do not use overloading.	ip nat translation timeout <i>seconds</i>

If you have configured overloading, you have finer control over translation entry timeout because each entry contains more context about the traffic that is using it. To change timeouts on extended entries, perform one or more of the following tasks in global configuration mode:

Task	Command
Change the UDP timeout value from 5 minutes.	ip nat translation udp-timeout <i>seconds</i>
Change the DNS timeout value from 1 minute.	ip nat translation dns-timeout <i>seconds</i>
Change the TCP timeout value from 24 hours.	ip nat translation tcp-timeout <i>seconds</i>
Change the Finish and Reset timeout value from 1 minute.	ip nat translation finrst-timeout <i>seconds</i>

Monitor and Maintain NAT

By default, dynamic address translations will time out from the NAT translation table at some point. You can clear the entries before the timeout by performing one of the following tasks in EXEC mode:

Task	Command
Clear all dynamic address translation entries from the NAT translation table.	clear ip nat translation *
Clear a simple dynamic translation entry containing an inside translation, or both inside and outside translation.	clear ip nat translation inside <i>global-ip local-ip</i> [outside <i>local-ip global-ip</i>]
Clear a simple dynamic translation entry containing an outside translation.	clear ip nat translation outside <i>local-ip global-ip</i>
Clear an extended dynamic translation entry.	clear ip nat translation protocol inside <i>global-ip</i> <i>global-port local-ip local-port</i> [outside <i>local-ip</i> <i>local-port global-ip global-port</i>]

You can display translation information by performing one of the following tasks in EXEC mode:

Task	Command
Display active translations.	show ip nat translations [verbose]
Display translation statistics.	show ip nat statistics

NAT Configuration Examples

The following are NAT configuration examples.

Dynamic Inside Source Translation Example

The following example translates all source addresses passing access list 1 (having a source address from 192.168.1.0/24) to an address from the pool named net-208. The pool contains addresses from 171.69.233.208 to 171.69.233.233.

```
ip nat pool net-208 171.69.233.208 171.69.233.233 netmask 255.255.255.240
ip nat inside source list 1 pool net-208
!
interface serial 0
 ip address 171.69.232.182 255.255.255.240
 ip nat outside
!
```

```

interface ethernet 0
  ip address 192.168.1.94 255.255.255.0
  ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255

```

Overloading Inside Global Addresses Example

The following example creates a pool of addresses named net-208. The pool contains addresses from 171.69.233.208 to 171.69.233.233. Access list 1 allows packets having the source address from 192.168.1.0 to 192.168.1.255. If no translation exists, packets matching access list 1 are translated to an address from the pool. The router allows multiple local addresses (192.168.1.0 to 192.168.1.255) to use the same global address. The router retains port numbers to differentiate the connections.

```

ip nat pool net-208 171.69.233.208 171.69.233.233 netmask 255.255.255.240
ip nat inside source list 1 pool net-208 overload
!
interface serial0
  ip address 171.69.232.182 255.255.255.240
  ip nat outside
!
interface ethernet0
  ip address 192.168.1.94 255.255.255.0
  ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255

```

Translating Overlapping Address Example

In the following example, the addresses in the local network are being used legitimately by someone else on the Internet. An extra translation is required to access that external network. Pool net-10 is a pool of outside local IP addresses. The statement `ip nat outside source list 1 pool net-10` translates the addresses of hosts from the outside overlapping network to addresses in that pool.

```

ip nat pool net-208 171.69.233.208 171.69.233.223 prefix-length 28
ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24
ip nat inside source list 1 pool net-208
ip nat outside source list 1 pool net-10
!
interface serial 0
  ip address 171.69.232.192 255.255.255.240
  ip nat outside
!
interface ethernet0
  ip address 192.168.1.94 255.255.255.0
  ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255

```

TCP Load Distribution Example

In the following example, the goal is to define a virtual address, connections to which are distributed among a set of real hosts. The pool defines the addresses of the real hosts. The access list defines the virtual address. If a translation does not already exist, TCP packets from serial 0 (the outside interface) whose destination matches the access list are translated to an address from the pool.

```

ip nat pool real-hosts 192.168.15.2 192.168.15.15 prefix-length 28 type rotary
ip nat inside destination list 2 pool real-hosts
!

```

NAT Configuration Examples

```
interface serial 0
 ip address 192.168.15.129 255.255.255.240
 ip nat outside
!
interface ethernet 0
 ip address 192.168.15.17 255.255.255.240
 ip nat inside
!
access-list 2 permit 192.168.15.1
```