

Configuring NAT for IP Address Conservation

This module describes how to configure Network Address Translation (NAT) for IP address conservation and how to configure inside and outside source addresses. This module also provides information about the benefits of configuring NAT for IP address conservation.

NAT enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT operates on a device, usually connecting two networks. Before packets are forwarded onto another network, NAT translates the private (not globally unique) addresses in the internal network into legal addresses. NAT can be configured to advertise to the outside world only one address for the entire network. This ability provides more security by effectively hiding the entire internal network behind that one address.

NAT is also used at the enterprise edge to allow internal users access to the Internet. It allows Internet access to internal devices such as mail servers.

- Prerequisites for Configuring NAT for IP Address Conservation, on page 1
- Restrictions for Configuring Network Address Translation for IP Address Conservation, on page 2
- Information About Configuring NAT for IP Address Conservation, on page 5
- How to Configure NAT for IP Address Conservation, on page 15
- Configuration Examples for Configuring NAT for IP Address Conservation, on page 40
- Where to Go Next, on page 45
- Additional References for Configuring NAT for IP Address Conservation, on page 45

Prerequisites for Configuring NAT for IP Address Conservation

Access Lists

All access lists that are required for use with the configuration tasks that are described in this module must be configured before initiating a configuration task. For information about how to configure an access list, see the *IP Access List EntrySequence Numbering* document.

Note If you specify an access list with a NAT command, NAT will not support the **permit ip any any** command. This command is commonly used in an access list.

NAT Requirements

Before configuring NAT in your network, ensure that you know the interfaces on which NAT is configured and for what purposes. The following requirements help you decide how to configure and use NAT:

- Define the NAT inside and outside interfaces if:
  - Users exist off multiple interfaces.
  - Multiple interfaces connect to the internet.
- Define what you need NAT to accomplish:
  - Allow internal users to access the internet.
  - Allow the internet to access internal devices such as a mail server.
  - Allow overlapping networks to communicate.
  - Allow networks with different address schemes to communicate.
  - Allow networks with different address schemes to communicate.
  - Redirect TCP traffic to another TCP port or address.
  - Use NAT during a network transition.

From Cisco IOS XE Denali 16.3 release, NAT support is introduced on Bridge Domain Interface (BDI) for enabling NAT configuration on the BDI interface.

Restrictions for Configuring Network Address Translation for IP Address Conservation

- When you configure Network Address Translation (NAT) on an interface, that interface becomes optimized for NAT packet flow. Any nontranslated packet that flows through the NAT interface goes through a series of checks to determine whether the packet must be translated or not. These checks result in increased latency for nontranslated packet flows and thus negatively impact the packet processing latency of all packet flows through the NAT interface. We highly recommend that a NAT interface must be used only for NAT-only traffic. Any non-NAT packets must be separated and these packets must go through an interface that does not have NAT configured on it. You can use Policy-Based Routing (PBR) for separating non-NAT traffic.
- NAT Virtual Interfaces (NVIs) are not supported in the Cisco IOS XE software.
- In Cisco IOS XE software, NAT outside interfaces show up in the translations tables, by default. This view of NAT outside interfaces causes the connection that originates from the outside interface of the device to fail. To restore connectivity, you must explicitly deny the outside Interface within the NAT ACL using the **deny** command. After using the **deny** command, no translation is observed for the outside interface.
- NAT is not practical if large numbers of hosts in the stub domain communicate outside of the domain.
- Some applications use embedded IP addresses in such a way that translation by a NAT device is impractical. These applications may not work transparently or at all through a NAT device.

- In a NAT configuration, addresses configured for any inside mapping must not be configured for any outside mapping.
- Do not configure the interface IP address as part of the IP address NAT pool.
- By default, support for the Session Initiation Protocol (SIP) is enabled on port 5060. Therefore, NAT-enabled devices interpret all packets on this port as SIP call messages. If other applications in the system use port 5060 to send packets, the NAT service may corrupt the packet. This packet corruption is due to its attempt to interpret the packet as a SIP call message.
- NAT hides the identity of hosts, which may be an advantage or a disadvantage depending on the needed result.
- Devices that are configured with NAT must not advertise the local networks to outside the network. However, routing information that NAT receives from the outside can be advertised in the stub domain as usual.
- NAT outside interface is not supported on a VRF. However, NAT outside interface is supported in iWAN and is part of the Cisco Validated Design.
- For VRF-aware NAT, remove the NAT configuration before you remove the VRF configuration.
- If you specify an access list to use with a NAT command, NAT does not support the **permit ip any any** command. This NAT command is commonly used in the access list.
- This platform does not support an access list with a port range.
- NAT configuration is not supported on the access side of the Intelligent Services Gateway (ISG).
- Using any IP address that is configured of a device as an address pool or in a NAT static rule is not supported. NAT can share the physical interface address (not any other IP address) of a device only by using the NAT interface overload configuration. A device uses the ports of its physical interface and NAT must receive communication about the ports that it can safely use for translation. This communication happens only when the NAT interface overload is configured.
- The output of the **show ip nat statistics** command displays information about all IP address pools and NAT mappings that you have configured. If your NAT configuration has a high number of IP address pools and NAT mappings, the update rate of the pool and mapping statistics in **show ip nat statistics** is slow. For example, NAT configuration output with 1000 to 4000 NAT mappings.
- Static and dynamic NAT with generic routing encapsulation (generic GRE) and dynamic NAT with Layer 2 do not work when used along with hardware-based Cisco AppNav appliances such as, Wide Area Application Services (WAAS). In the context of WAAS, generic GRE is an out of path deployment mechanism. It helps to return packets from the WAAS Wide-Area Application Engine (WAE) through the GRE tunnel to the same device from which they were originally redirected after completing optimization.
- Port Address Translation (also called NAT overload) only supports protocols whose port numbers are known; these protocols are Internet Control Message Protocol (ICMP), TCP, and UDP. Other protocols do not work with PAT because they consume the entire address in an address pool. Configure your access control list to only permit ICMP, TCP, and UDP protocols, so that all other protocol traffic is prevented from entering the network.
- NAT, Zone-Based Policy Firewall, and Web Cache Communication Protocol (WCCP) cannot coexist in a network.

- Non-Pattable traffic, is traffic for a protocol where there are no ports. PAT/Overload can only be done on protocols where the ports are known, that is, UDP, TCP, and ICMP.

  When NAT overload (PAT) is configured and Non-Pattable traffic hits the router, Non-Pattable BIND entry gets created for this traffic. Following is a bind entry in the NAT table:

  ```
  ---  213.252.7.132         172.16.254.242        ---
  ```

  This bind entry consumes an entire address from the pool. In this example, 213.252.7.132 is an address from an overloaded pool.

  That means an inside local IP Address gets bound to the outside global IP which is similar to static NAT. Because of this binding action, new inside local IP Addresses cannot use this global IP Address until the current entry gets timed out. All the translation that is created off this BIND is 1-to-1 translations instead of overload.

  To avoid consumption of an entire address from the pool, make sure that there are not any entries for the Non-Pattable traffic across the router.

- Using the **ip nat settings nonpatdrop** command, the router drops packets that cause Non-Pattable NAT translation attempts, which are traffic types without port numbers, instead of creating one-to-one bindings. This helps conserve NAT pool addresses and prevents address exhaustion from such traffic.
- When configuring NAT with ACLs or route maps, the ACLs or route maps must not overlap. If the ACLs or route maps overlap, NAT cannot map to the required transition.
- Port Address Translation (PAT) over 64K Endpoint Dependant Mapping (EDM) is only supported in the classic NAT mode and not the in Carrier Grade Network Address Translation (CGN) mode and SD-WAN mode.
- Virtual Routing and Forwarding (VRF) is not supported in PAT over 64K.
- Password Authentication Protocol (PAP) mode is not supported in PAT over 64K.
- Application Level Gateway (ALGs) are not supported in PAT over 64K.
- You can only use 1024-65535 ports and not 0-1023 ports in PAT over 64K.
- When configuring a static NAT entry while dynamic NAT is already in use, first ensure the IP address for static mapping doesn't overlap with dynamic NAT addresses. If conflicts exist, update the ACL to deny those IPs and clear any conflicting dynamic NAT entries from the translation table. Once resolved, add the static NAT configuration.



Note The control plane manages PAT interface overload port allocation. To detect allocation failures, monitor the drop counters in the show ip nat statistics command instead of relying on resource-intensive syslogs. The effective port range is typically 53,000 to 58,000.

Information About Configuring NAT for IP Address Conservation

Benefits of Configuring NAT for IP Address Conservation

NAT allows organizations to resolve the problem of IP address depletion when they have existing networks and must access the Internet. Sites that do not yet possess Network Information Center (NIC)-registered IP addresses must acquire them. If more than 254 clients are present or planned, the scarcity of Class B addresses becomes a serious issue. Cisco IOS XE NAT addresses these issues by mapping thousands of hidden internal addresses to a range of easy-to-get Class C addresses.

Sites that already have registered IP addresses for clients on an internal network may want to hide those addresses from the Internet. This action disable hacker to directly attack the clients. With clients addresses hidden, an extent of security is established. Cisco IOS XE NAT gives LAN administrators complete freedom to expand Class A addressing. The Class A addressing expansion is drawn from the reserve pool of the Internet Assigned Numbers Authority (RFC 1597). This expansion occurs within the organization without concern for addressing changes at the LAN/Internet interface.

The Cisco IOS XE software can selectively or dynamically perform NAT. This flexibility allows the network administrator to use a mix of RFC 1597 and RFC 1918 addresses or registered addresses. NAT is designed for use on various devices for IP address simplification and conservation. In addition, Cisco IOS XE NAT allows the selection of internal hosts that are available for NAT.

A significant advantage of NAT is that it can be configured without requiring changes to hosts or devices in the network. However, changes are required on few other devices where NAT is configured.

In Cisco IOS XE Denali 16.3 release, Multi-Tenant support for NAT feature was introduced. With Multi-Tenant support, the configuration changes of a Virtual Routing and Forwarding (VRF) instance does not interrupt the traffic flow of other VRFs in the network.

NAT is a feature that allows the IP network of an organization to appear, from the outside, to be using a different IP address space than the one that 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 graceful renumbering strategy for organizations that are changing service providers or voluntarily renumbering into classless interdomain routing (CIDR) blocks. NAT is described in RFC 1631.

How NAT Works

A device that is configured with NAT has at least one interface to the inside network and one to the outside network. In a typical environment, NAT is configured at the exit device between a stub domain and the backbone. When a packet exits the domain, NAT translates the locally significant source address into a globally unique address. When a packet enters 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 NAT cannot allocate an address because it has run out of addresses, it drops the packet. Then, NAT sends an Internet Control Message Protocol (ICMP) host unreachable packet to the destination.

Uses of NAT

NAT can be used for the following scenarios:

- Connect to the internet when all your hosts do not have globally unique IP addresses. Network Address Translation (NAT) enables private IP networks that use nonregistered IP addresses to connect to the Internet. NAT is configured on a device at the border of a stub domain (mentioned as the *inside network*) and a public network such as the Internet (mentioned as the *outside network*). NAT translates internal local addresses to globally unique IP addresses before sending packets to the outside network. As a solution to the connectivity problem, NAT is practical only when relatively few hosts in a stub domain communicate simultaneously outside the domain. When outside communication is necessary, only a small subset of the IP addresses in the domain must be translated into globally unique IP addresses. Also, these addresses can be reused when they are no longer in use.
- Change your internal addresses. Instead of changing the internal addresses, which can be a considerable amount of work, you can translate them by using NAT.
- For basic load-sharing of TCP traffic. You can map a single global IP address with many local IP addresses by using the TCP Load Distribution feature.

Types of NAT

NAT operates on a router—generally connecting only two networks. Before any packets are forwarded to another network, NAT translates the private (inside local) addresses within the internal network into public (inside global) addresses. This functionality gives you the option to configure NAT so that it advertises only a single address for your entire network to the outside world. Doing this translation, NAT effectively hides the internal network from the world, giving you some additional security.

The types of NAT include:

- Static address translation (static NAT)—Allows one-to-one mapping between local and global addresses.
- Dynamic address translation (dynamic NAT)—Maps unregistered IP addresses to registered IP addresses from a pool of registered IP addresses.
- Overloading—Maps multiple unregistered IP addresses to a single registered IP address (many to one) by using different ports. This method is also known as Port Address Translation (PAT). Thousands of users can be connected to the Internet by using only one real global IP address through overloading.

NAT Inside and Outside Addresses

The term *inside* in a Network Address Translation (NAT) context refers to networks owned by an organization that must be translated. When NAT is configured, hosts within this network have addresses in one space (known as the *local* address space). These hosts appear to those users outside the network as being in another space (known as the *global* address space).

Similarly, the term *outside* refers to those networks to which the stub network connects, and which are not under the control of an organization. Also, hosts in outside networks can be subject to translation, and can thus have local and global addresses. NAT uses the following definitions:

- Inside local address—An IP address that is assigned to a host on the inside network. The address that the Network Information Center (NIC) or service provider assigns is probably not a legitimate IP address.
- 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 is allocated from the address space that is routable on the inside.
- Outside global address—The IP address that is assigned to a host on the outside network by the owner of the host. The address is allocated from a globally routable address or network space.

NAT supports the following VRFs:

Table 1: VRF NAT Support

NAT Inside Interface	NAT Outside Interface	Condition
Global VRF (also referred to as a non-VRF interface)	Global VRF (also referred to as a non-VRF interface)	Normal
VRF X	Global VRF (also referred to as a non-VRF interface)	When NAT is not configured for Match-in-VRF support. For more details, see the *Match-in-VRF Support for NAT* chapter.
VRF X	VRF X	When both inside and outside interfaces are in the same VRF, and NAT is configured with Match-in-VRF support.

This section describes the following topics:

- Inside Source Address Translation, on page 7
- Overloading of Inside Global Addresses, on page 8

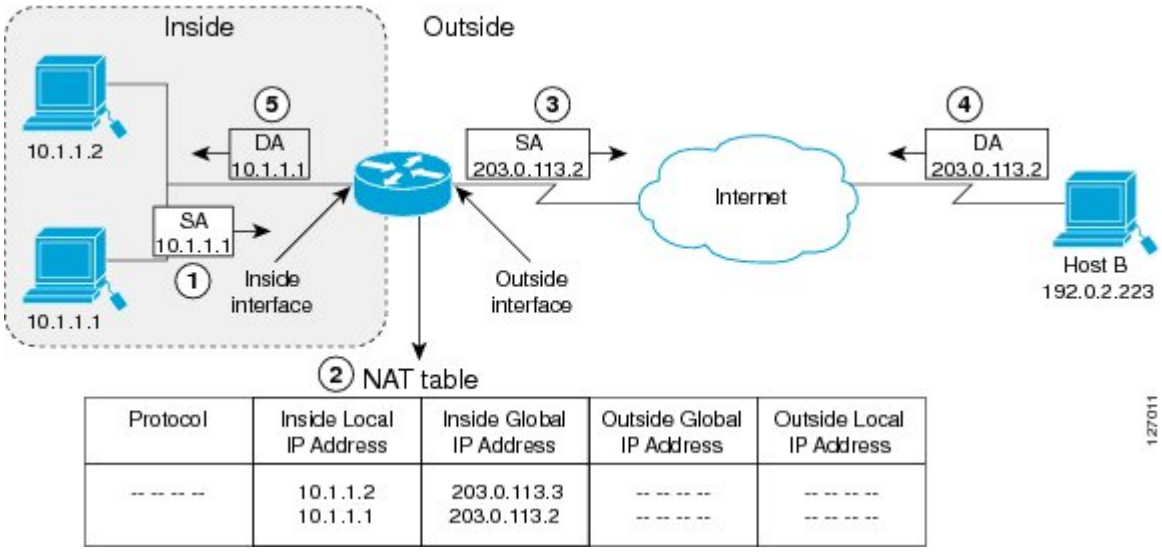
Inside Source Address Translation

You can translate IP addresses into globally unique IP addresses when communicating outside of your network. You can configure inside source address translation of static or dynamic NAT as follows:

- *Static translation* establishes a one-to-one mapping between the 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.

The following figure illustrates a device that is translating a source address inside a network to a source address outside the network.

Figure 1: NAT Inside Source Translation



The following process describes the inside source address translation, as shown in the preceding figure:

1. The user at host 10.1.1.1 opens a connection to Host B in the outside network.
2. The first packet that the device receives from host 10.1.1.1 causes the device to check its Network Address Translation (NAT) table. Based on the NAT configuration, the following scenarios are possible:
   - If a static translation entry is configured, the device goes to Step 3.
   - If no translation entry exists, the device determines that the source address (SA) 10.1.1.1 must be translated dynamically. The device selects a legal, global address from the dynamic address pool, and creates a translation entry in the NAT table. This kind of translation entry is called a *simple entry*.
3. The device replaces the inside local source address of host 10.1.1.1 with the global address of the translation entry and forwards the packet.
4. Host B receives the packet and responds to host 10.1.1.1 by using the inside global IP destination address (DA) 203.0.113.2.
5. When the device 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 10.1.1.1 and forwards the packet to host 10.1.1.1.

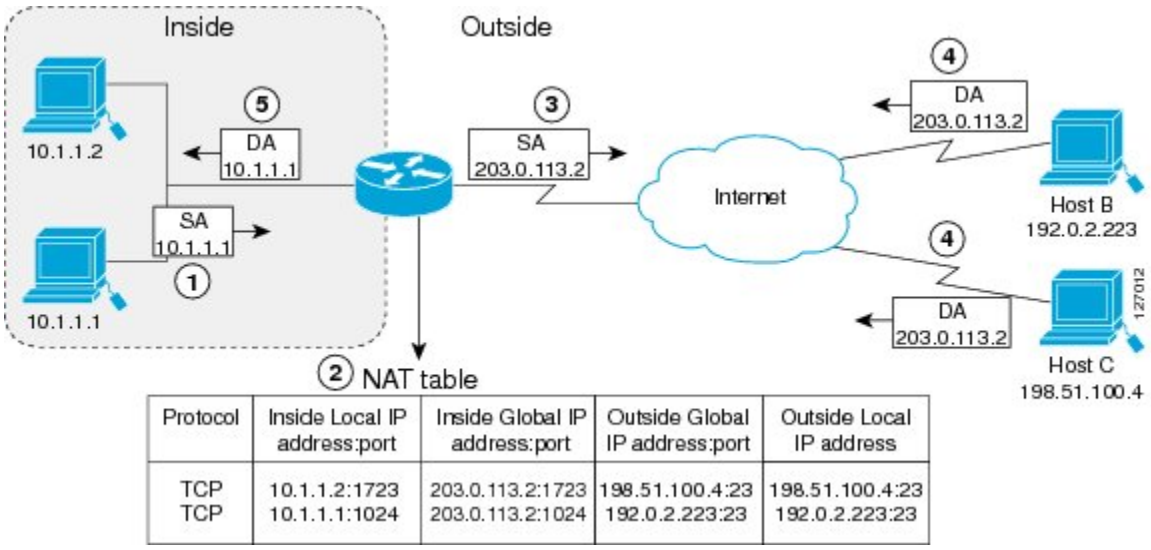
Host 10.1.1.1 receives the packet and continues the conversation. The device performs Steps 2 to 5 for each packet that it receives.

Overloading of Inside Global Addresses

You can conserve addresses in the inside global address pool by allowing a device to use one global address for many local addresses. This type of Network Address Translation (NAT) configuration is called overloading. When overloading is configured, the device maintains enough information from higher-level protocols (for example, TCP or UDP port numbers). This action translates the global address back to the correct local address. When multiple local addresses map to one global address, the TCP or UDP port numbers of each inside host distinguish between local addresses.

The following figure illustrates a NAT operation when an inside global address represents multiple inside local addresses. The TCP port numbers act as differentiators.

Figure 2: NAT Overloading Inside Global Addresses



Protocol	Inside Local IP address:port	Inside Global IP address:port	Outside Global IP address:port	Outside Local IP address
TCP	10.1.1.2:1723	203.0.113.2:1723	198.51.100.4:23	198.51.100.4:23
TCP	10.1.1.1:1024	203.0.113.2:1024	192.0.2.223:23	192.0.2.223:23

The device performs the following process in the overloading of inside global addresses, as shown in the preceding figure. Both Host B and Host C believe that they are communicating with a single host at address 203.0.113.2. Whereas, they are actually communicating with different hosts; the port number is the differentiator. In fact, many inside hosts can share the inside global IP address by using many port numbers.

1. The user at host 10.1.1.1 opens a connection to Host B.
2. The first packet that the device receives from host 10.1.1.1 causes the device to check its NAT table. Based on your NAT configuration the following scenarios are possible:
   - If no translation entry exists, the device determines that IP address 10.1.1.1 must be translated, and translates inside local address 10.1.1.1 to a legal global address.
   - If overloading is enabled and another translation is active, the device reuses the global address from that translation and saves enough information. This saved information can be used to translate the global address back, as an entry in the NAT table. This type of translation entry is called an *extended entry*.
3. The device replaces inside local source address 10.1.1.1 with the selected global address and forwards the packet.
4. Host B receives the packet and responds to host 10.1.1.1 by using the inside global IP address 203.0.113.2.
5. When the device receives the packet with the inside global IP address, it performs a NAT table lookup by using a protocol, the inside global address and port, and the outside address and port as keys. It translates the address to the inside local address 10.1.1.1 and forwards the packet to host 10.1.1.1.

Host 10.1.1.1 receives the packet and continues the conversation. The device performs Steps 2 to 5 for each packet it receives.

Port Address Translation over 64K

With normal Port Address Translation (PAT), a single global IP address can only support up to 64K UDP and 64K TCP sessions. With Port over 64K, the port manager must keep track of the destination address of the flow per global address. This allows the port manager to support 64K UDP and 64K TCP sessions for a single global IP address for each outside destination address. This allows to scale the translations over 64K per protocol per public address.

The port translations also depend on the destination (end point), hence this type of NAT is also called Endpoint Dependent Mapping (EDM). It is likely that PAT over 64 requires significantly more memory than traditional port manager because port information needs to be tracked per destination IP address. Its performance is expected to be slightly lower than traditional Network Address Translation (NAT) due to additional complexity.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ipnatinsidesourcelist***access-list-number***pool***name***overloadextended***
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**
13. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:**	Defines a pool of global addresses to be allocated as needed.

	Command or Action	Purpose
	`Device(config)# ip nat pool nat_pool_edm 192.168.202.129 192.168.202.158 netmask 255.255.255.224`	
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 192.168.201.30 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated. • 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.) Use of an access list that is too permissive can lead to unpredictable results.
Step 5	**ipnatinsidesourcelist***access-list-number***pool***name***overloadextended*** **Example:** `Device(config)# ip nat inside source list 1 pool nat_pool_edm overload extended`	Establishes dynamic source translation with overloading translations over 64K per address for protocol. **Note** You can also configure the interface to **ipnatinsidesourcelist***access-list-numbe*
Step 6	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.201.1 255.255.255.240`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters the interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.201.29 255.255.255.240`	Sets a primary IP address for the interface.

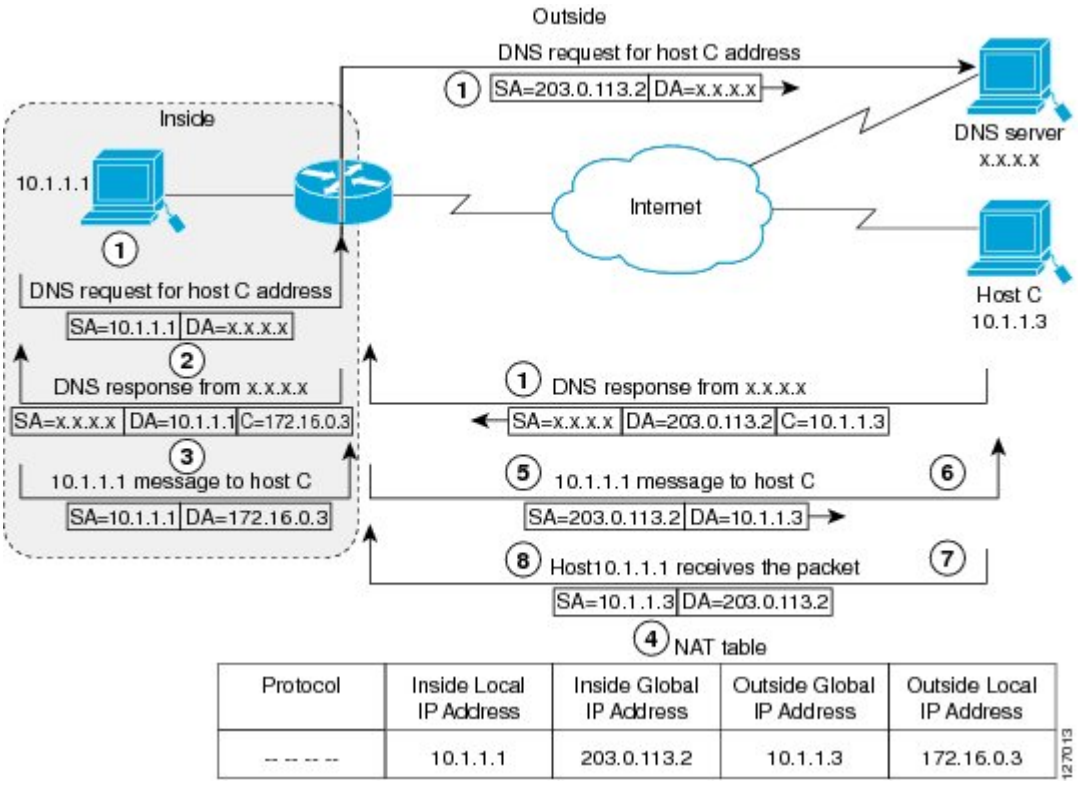
	Command or Action	Purpose
Step 12	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 13	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Address Translation of Overlapping Networks

Use Network Address Translation (NAT) to translate IP addresses if the IP addresses that you use are not legal or officially assigned. Overlapping networks result when you assign an IP address to a device on your network. This device is already legally owned and assigned to a different device on the Internet or outside the network.

The following figure shows how NAT translates overlapping networks.

Figure 3: NAT Translating Overlapping Addresses



The following steps describe how a device translates overlapping addresses:

1. Host 10.1.1.1 opens a connection to Host C using a name, requesting a name-to-address lookup from a Domain Name System (DNS) server.

2. The device 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 device creates a simple translation entry. This entry maps the overlapping address, 10.1.1.3 to an address from a separately configured, outside the local address pool.

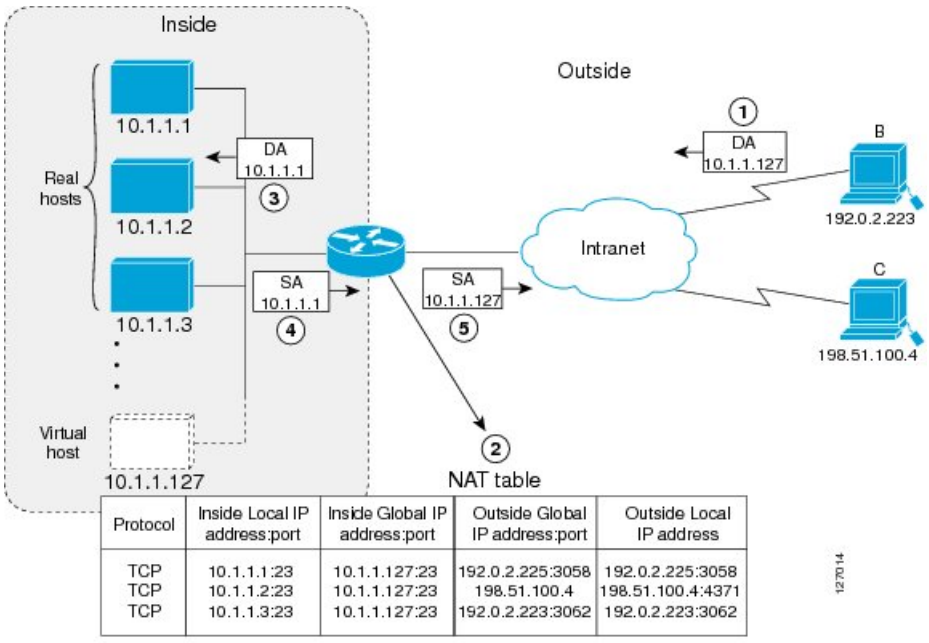
The device examines every DNS reply to ensure that the IP address is not in a stub network. If it is, the device translates the address as described in the following steps:

1. Host 10.1.1.1 opens a connection to 172.16.0.3.
2. The device sets up the translation mapping of the inside local and global addresses to each other. It also sets up the translation mapping of the outside global and local addresses to each other.
3. The device replaces the SA with the inside global address and replaces the DA with the outside global address.
4. Host C receives the packet and continues the conversation.
5. The device does a lookup, replaces the DA with the inside local address, and replaces the SA with the outside local address.
6. Host 10.1.1.1 receives the packet and the conversation continues using this translation process.

TCP Load Distribution for NAT

Your organization may have multiple hosts that must communicate with a heavily used host. By using Network Address Translation (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 on a round-robin basis and only when a new connection is opened from the outside to inside the network. Non-TCP traffic is passed untranslated (unless other translations are configured). The following figure illustrates how TCP load distribution works.

Figure 4: NAT TCP Load Distribution



A device performs the following process when translating rotary addresses:

1. Host B (192.0.2.223) opens a connection to a virtual host at 10.1.1.127.
2. The device receives the connection request and creates a new translation, allocating the next real host (10.1.1.1) for the inside local IP address.
3. The device replaces the destination address with the selected real host address and forwards the packet.
4. Host 10.1.1.1 receives the packet and responds.
5. The device receives the packet and performs a NAT table lookup by using the inside local address and port number. It also does a NAT table lookup by using the outside address and port number as keys. The device then translates the source address to the address of the virtual host and forwards the packet.
6. The device will allocate IP address 10.1.1.2 as the inside local address for the next connection request.

Static IP Address Support

A public wireless LAN provides users of mobile computing devices with wireless connections to a public network, such as the Internet.

To support users who are configured with a static IP address, the NAT Static IP Address Support feature extends the capabilities of public wireless LAN providers. By configuring a device to support users with a static IP address, public wireless LAN providers extend their services to a greater number of users.

Users with static IP addresses can use services of the public wireless LAN provider without changing their IP address. NAT entries are created for static IP clients and a routable address is provided.

RADIUS

RADIUS is a distributed client/server system that secures networks against unauthorized access. Communication between a network access server (NAS) and a RADIUS server is based on UDP. Generally, the RADIUS protocol is considered a connectionless service. RADIUS-enabled devices handle issues that are related to a server availability, retransmission, and timeouts rather than the transmission protocol.

The RADIUS client is typically a NAS, and the RADIUS server is usually a daemon process running on a UNIX or Windows NT machine. The client passes user information to designated RADIUS servers and acts on the response that is returned. To deliver service to the user, RADIUS servers receive a user connection request, authenticate the user, and then return the configuration information necessary for the client. A RADIUS server can act as a proxy client to other RADIUS servers or other kinds of authentication servers.

Denial-of-Service Attacks

A denial-of-service (DoS) attack typically involves misuse of standard protocols or connection processes. The intent of DoS attack is to overload and disable a target, such as a device or web server. DoS attacks can come from a malicious user or from a computer that is infected with a virus or worm. Distributed DoS attack is an attack that comes from many different sources at once. This attack can be when a virus or worm has infected many computers. Such distributed DoS attacks can spread rapidly and involve thousands of systems.

Viruses and Worms That Target NAT

Viruses and worms are malicious programs that are designed to attack computers and networking equipment. Although viruses are typically embedded in discrete applications and run only when executed, worms self-propagate and can quickly spread by their own. Although a specific virus or worm may not expressly target NAT, it may use NAT resources to propagate itself. The Rate Limiting NAT Translation feature can be used to limit the impact of viruses and worms. These viruses and worms originate from specific hosts, access control lists, and VPN routing and forwarding (VRF) instances.

How to Configure NAT for IP Address Conservation

The tasks that are described in this section configure NAT for IP address conservation. Ensure that you configure at least one of the tasks that are described in this section. Based on your configuration, you may need to configure more than one task.

Configuring Inside Source Addresses

Inside source addresses, can be configured for static or dynamic translations. Based on your requirements, you can configure either static or dynamic translations.

Configuring Static Translation of Inside Source Addresses

Configure static translation of the inside source addresses to allow one-to-one mapping between an 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.

Note Configure different IP addresses for an interface on which NAT is configured and for inside addresses that are configured by using the **ip nat inside source static** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip*
4. **interface** *type number*
5. **ip address** *ip-address mask* [**secondary**]
6. **ip nat inside**
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask* [**secondary**]
10. **ip nat outside**
11. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Device(config)# ip nat inside source static 10.10.10.1 172.16.131.1`	Establishes static translation between an inside local address and an inside global address.
Step 4	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for an interface.

	Command or Action	Purpose
Step 6	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies a different interface and enters the interface configuration mode.
Step 9	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 10	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 11	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode. **Note** Conditional translation is not supported

Configuring Dynamic Translation of Inside Source Addresses

Dynamic translation establishes a mapping between an inside local address and a pool of global addresses. Dynamic translation is useful when multiple users on a private network must access the Internet. The dynamically configured pool IP address may be used as needed. It is released for use by other users when access to the Internet is no longer required.

Note When inside global or outside local addresses belong to a directly connected subnet on a NAT device, the device adds IP aliases for them. This action enables it to answer Address Resolution Protocol (ARP) requests. However, a situation can arise where the device answers packets that are not destined for it, possibly causing a security issue. This security issue can happen when an incoming Internet Control Message Protocol (ICMP) packet or a UDP packet that is destined for one of the aliased addresses does not have a corresponding NAT translation in the NAT table. Also, the device itself runs a corresponding service, for example, Network Time Protocol (NTP). Such a situation can cause minor security risks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ip nat inside source list** *access-list-number* **pool** *name*
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**
13. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool net-208 172.16.233.208 172.16.233.223 prefix-length 28`	Defines a pool of global addresses to be allocated as needed.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 192.168.34.0 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated.
Step 5	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Device(config)# ip nat inside source list 1 pool net-208`	Establishes dynamic source translation, specifying the access list defined in Step 4.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters an interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters an interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 13	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the Same Global Address for Static NAT and PAT

You can configure the same global address for the static NAT and PAT. Static translation is useful when a host on the inside must be accessible by a fixed address from the outside.

Note This is not supported with **ip nat inside source static** configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat outside source static** *outside global-ip outside local-ip*
4. **ip nat outside source static** {**tcp | udp**} *outside global-ip global-port outside local-ip local-port* **extendable**
5. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat outside source static** *outside global-ip outside local-ip* **Example:** `Device(config)# ip nat outside source static 10.21.0.202 12.182.174.202`	Establishes static translation between an outside local address and an outside global address.
Step 4	**ip nat outside source static** {**tcp	udp**} *outside global-ip global-port outside local-ip local-port* **extendable** **Example:** `Router(config)# ip nat outside source static tcp 10.21.14.49 22512 12.182.174.202 5009 extendable`	• Establishes static translation between an outside global address and inside local address.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Using NAT to Allow Internal Users Access to the Internet

Perform this task to allow your internal users access to the Internet and conserve addresses in the inside global address pool using overloading of global addresses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ip nat inside source list** *access-list-number* **pool** *name* **overload**
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**
13. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool net-208 192.168.202.129 192.168.202.158 netmask 255.255.255.224`	Defines a pool of global addresses to be allocated as needed.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 192.168.201.30 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated. • 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.) Use of an access list that is too permissive can lead to unpredictable results.
Step 5	**ip nat inside source list** *access-list-number* **pool** *name* **overload** **Example:** `Device(config)# ip nat inside source list 1 pool net-208 overload`	Establishes dynamic source translation with overloading, specifying the access list defined in Step 4.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.201.1 255.255.255.240`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters the interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.201.29 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 13	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Address Translation Timeouts

You can configure address translation timeouts that is based on your NAT configuration.

By default, dynamic address translations time out after a period of remaining idle. You can change the default values on timeouts, if necessary. When overloading is not configured, simple translation entries time out after 24 hours. Use the **ip nat translation timeout** command to change the timeout value for dynamic address translations.

You can use the **ip nat translation max-entries** command to change the default global NAT translation limit.



Note On Catalyst 6500 Series Switches, when the NAT translation is done in the hardware, timers are reset every 100 seconds or once the set timeout value is reached.

Changing the Translation Timeout

By default, dynamic address translations time out after some period of remaining idle. You can change the default values on timeouts, if necessary. When overloading is not configured, simple translation entries time out after 24 hours. Configure the **ip nat translation timeout** *seconds* command to change the timeout value for dynamic address translations that do not use overloading.

Changing the Timeouts When Overloading Is Configured

If you have configured overloading, you can control the translation entry timeout, because each translation entry contains more context about the traffic using it.

Based on your configuration, you can change the timeouts that are described in this section. If you must quickly free your global IP address for a dynamic configuration, configure a shorter timeout than the default timeout. You can do it by using the **ip nat translation timeout** command. However, the configured timeout is longer than the other timeouts configured using commands specified in the following task. If a finish (FIN) packet does not close a TCP session properly from both sides or during a reset, change the default TCP timeout. You can do it by using the **ip nat translation tcp-timeout** command.

When you change the default timeout using the **ip nat translation** *timeout* command, the timeout that you configure overrides the default TCP and UDP timeout values, unless you explicitly configure the TCP timeout value (using the **ip nat translation tcp-timeout** *seconds* command) or the UDP timeout value (using the **ip nat translation udp-timeout** *seconds* command).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat translation** *seconds*
4. **ip nat translation udp-timeout** *seconds*
5. **ip nat translation dns-timeout** *seconds*
6. **ip nat translation tcp-timeout** *seconds*
7. **ip nat translation finrst-timeout** *seconds*
8. **ip nat translation icmp-timeout** *seconds*
9. **ip nat translation syn-timeout** *seconds*
10. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat translation** *seconds* **Example:** `Device(config)# ip nat translation 300`	(Optional) Changes the amount of time after which NAT translations time out. • The default timeout is 24 hours, and it applies to the aging time for half-entries. • The timeout configured using this command overrides the default TCP and UDP timeout values, unless explicitly configured.
Step 4	**ip nat translation udp-timeout** *seconds* **Example:** `Device(config)# ip nat translation udp-timeout 300`	(Optional) Changes the UDP timeout value. • The default is 300 seconds. This default value only applies if the general IP NAT translation timeout value (using the **ip nat translation** *seconds* command) is not configured.
Step 5	**ip nat translation dns-timeout** *seconds* **Example:** `Device(config)# ip nat translation dns-timeout 45`	(Optional) Changes the Domain Name System (DNS) timeout value.
Step 6	**ip nat translation tcp-timeout** *seconds* **Example:** `Device(config)# ip nat translation tcp-timeout 2500`	(Optional) Changes the TCP timeout value. • The default is 7440 seconds. This default value only applies if the general IP NAT translation timeout value (using the **ip nat translation** *seconds* command) is not configured.
Step 7	**ip nat translation finrst-timeout** *seconds* **Example:** `Device(config)# ip nat translation finrst-timeout 45`	(Optional) Changes the finish and reset timeout value. • **finrst-timeout**—The aging time after a TCP session receives both finish-in (FIN-IN) and finish-out (FIN-OUT) requests or after the reset of a TCP session.
Step 8	**ip nat translation icmp-timeout** *seconds* **Example:** `Device(config)# ip nat translation icmp-timeout 45`	(Optional) Changes the ICMP timeout value.
Step 9	**ip nat translation syn-timeout** *seconds* **Example:**	(Optional) Changes the synchronous (SYN) timeout value.

	Command or Action	Purpose
	`Device(config)# ip nat translation syn-timeout 45`	• The synchronous timeout or the aging time is used only when a SYN request is received on a TCP session. When a synchronous acknowledgment (SYNACK) request is received, the timeout changes to TCP timeout.
Step 10	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Allowing Overlapping Networks to Communicate Using NAT

Tasks in this section are grouped because they perform the same action. However, the tasks are executed differently depending on the type of translation that is implemented—static or dynamic. Perform the task that applies to the translation type that you have implemented.

This section contains the following tasks:

- Configuring Static Translation of Overlapping Networks
- Configuring Dynamic Translation of Overlapping Networks
- What to Do Next

Configuring Static Translation of Overlapping Networks

Configure static translation of overlapping networks that are based on the following requirements:

- If your IP addresses in the stub network are legitimate IP addresses belonging to another network.
- If you want to communicate with those hosts or routers by using static translation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip*
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **ip nat inside**
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask*
10. **ip nat outside**
11. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Device(config)# ip nat inside source static 192.168.121.33 10.2.2.1`	Establishes static translation between an inside local address and an inside global address.
Step 4	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 6	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters the interface configuration mode.
Step 9	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 10	**ip nat outside** **Example:**	Marks the interface as connected to the outside.

	Command or Action	Purpose
	`Device(config-if)# ip nat outside`	
Step 11	**end** **Example:** `Device(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

What to Do Next

When you have completed the required configuration, go to the "Monitoring and Maintaining NAT" module.

Configuring Server TCP Load Balancing

Perform this task to configure a server TCP load balancing by way of destination address rotary translation. The commands that are specified in the task allow you to map one virtual host with many real hosts. Each new TCP session opened with the virtual host is translated into a session with a different real host.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **type rotary**
4. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
5. **ip nat inside destination-list** *access-list-number* **pool** *name*
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**
13. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **type rotary** **Example:** `Device(config)# ip nat pool real-hosts 192.168.201.2 192.168.201.5 prefix-length 28 type rotary`	Defines a pool of addresses containing the addresses of the real hosts.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 192.168.201.30 0.0.0.255`	Defines an access list permitting the address of the virtual host.
Step 5	**ip nat inside destination-list** *access-list-number* **pool** *name* **Example:** `Device(config)# ip nat inside destination-list 2 pool real-hosts`	Establishes dynamic inside destination translation, specifying the access list defined in the prior step.
Step 6	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters the interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.201.1 255.255.255.240`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface serial 0`	Specifies a different interface and enters the interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.15.129 255.255.255.240`	Sets a primary IP address for the interface.

	Command or Action	Purpose
Step 12	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 13	**end** **Example:** `Device(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Enabling Route Maps on Inside Interfaces

Before you begin

All route maps required for use with this task must be configured before you begin the configuration task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**]| **static** *local-ip global-ip* [**route-map** *map-name*]}
4. **exit**
5. **show ip nat translations** [**verbose**]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**]| **static** *local-ip global-ip* [**route-map** *map-name*]} **Example:** `Device(config)# ip nat inside source static 192.168.201.6 192.168.201.21 route-map isp2`	Enables route mapping with static NAT configured on the NAT inside interface.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations`	(Optional) Displays active NAT.

Enabling NAT Route Maps Outside-to-Inside Support

The NAT Route Maps Outside-to-Inside Support feature enables you to configure a Network Address Translation (NAT) route map configuration. It allows IP sessions to be initiated from the outside to the inside. Perform this task to enable the NAT Route Maps Outside-to-Inside Support feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* **netmask** *netmask*
4. **ip nat pool** *name start-ip end-ip* **netmask** *netmask*
5. **ip nat inside source route-map** *name* **pool** *name* [**reversible**]
6. **ip nat inside source route-map** *name* **pool** *name* [**reversible**]
7. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Device(config)# ip nat pool POOL-A 192.168.201.4 192.168.201.6 netmask 255.255.255.128`	Defines a pool of network addresses for NAT.
Step 4	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:**	Defines a pool of network addresses for NAT.

	Command or Action	Purpose
	`Device(config)# ip nat pool POOL-B 192.168.201.7 192.168.201.9 netmask 255.255.255.128`	
Step 5	**ip nat inside source route-map** *name* **pool** *name* [**reversible**] **Example:** `Device(config)# ip nat inside source route-map MAP-A pool POOL-A reversible`	Enables outside-to-inside initiated sessions to use route maps for destination-based NAT.
Step 6	**ip nat inside source route-map** *name* **pool** *name* [**reversible**] **Example:** `Device(config)# ip nat inside source route-map MAP-B pool POOL-B reversible`	Enables outside-to-inside initiated sessions to use route maps for destination-based NAT.
Step 7	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Configuring NAT of External IP Addresses Only

When you configure NAT of external IP addresses, NAT can be configured to ignore all embedded IP addresses for any application and traffic type. Traffic between a host and the traffic outside an enterprise's network flows through the internal network. A device that is configured for NAT translates the packet to an address that can be routed inside the internal network. If the intended destination is outside an enterprise's network, the packet gets translated back to an external address and is sent out.

Note When you configure the **ip nat outside source static** command to add static routes for outside local addresses, there is a delay in the translation of packets and packets are dropped. Packets are dropped because a shortcut is not created for the initial synchronization (SYN) packet when NAT is configured for static translation. To avoid dropped packets, configure either the **ip nat outside source static add-route** command or the **ip route** command.

Benefits of configuring NAT of external IP addresses only are:

- Allows an enterprise to use the Internet as its enterprise backbone network.
- Allows the use of network architecture that requires only the header translation.
- Gives the end client a usable IP address at the starting point. This address is the address that is used for IPsec connections and for traffic flows.
- Supports public and private network architecture with no specific route updates.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static network** *local-ip global-ip* [**no-payload**]}
4. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**no-payload**]}
5. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** [**network**] *local-network-mask global-network-mask* [**no-payload**]}
6. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** *local-ip global-ip* [**no-payload**]}
7. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**no-payload**]}
8. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** [**network**] *local-network-mask global-network-mask* [**no-payload**]}
9. **exit**
10. **show ip nat translations** [**verbose**]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static network** *local-ip global-ip* [**no-payload**]} **Example:** `Device(config)# ip nat inside source static network 10.1.1.1 192.168.251.0/24 no-payload`	Disables the network packet translation on the inside host device.
Step 4	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**no-payload**]} **Example:** `Device(config)# ip nat inside source static tcp 10.1.1.1 2000 192.168.1.1 2000 no-payload`	Disables port packet translation on the inside host device.
Step 5	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static**	Disables packet translation on the inside host device.

	Command or Action	Purpose
	[**network**] *local-network-mask global-network-mask* [**no-payload**]} **Example:** `Device(config)# ip nat inside source static 10.1.1.1 192.168.1.1 no-payload`	
Step 6	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** *local-ip global-ip* [**no-payload**]} **Example:** `Device(config)# ip nat outside source static 10.1.1.1 192.168.1.1 no-payload`	Disables packet translation on the outside host device.
Step 7	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**no-payload**]} **Example:** `Device(config)# ip nat outside source static tcp 10.1.1.1 20000 192.168.1.1 20000 no-payload`	Disables port packet translation on the outside host device.
Step 8	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** [**network**] *local-network-mask global-network-mask* [**no-payload**]} **Example:** `Device(config)# ip nat outside source static network 10.1.1.1 192.168.251.0/24 no-payload`	Disables network packet translation on the outside host device.
Step 9	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 10	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations`	Displays active NAT.

Configuring the NAT Default Inside Server Feature

The NAT Default Inside Server feature helps forward packets from the outside to a specified inside local address. Traffic that does not match any existing dynamic translations or static port translations are redirected, and packets are not dropped.

Dynamic mapping and interface overload can be configured for gaming devices. For online games, outside traffic comes on a different UDP port. If a packet is destined for an interface from outside an enterprise's network, and there is no match in the NAT table for fully extended entry or static port entry, the packet is forwarded to the gaming device using a simple static entry.

Note

- You can use this feature to configure gaming devices with an IP address different from the IP address of the PC. To avoid unwanted traffic or DoS attacks, use access lists.
- For traffic going from the PC to the outside, it is better to use a route map so that extended entries are created.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip* **interface** *type number*
4. **ip nat inside source static tcp** *local-ip local-port* **interface** *global-port*
5. **exit**
6. **show ip nat translations** [**verbose**]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip* **interface** *type number* **Example:** `Device(config)# ip nat inside source static 10.1.1.1 interface Ethernet 1/1`	Enables static NAT on the interface.
Step 4	**ip nat inside source static tcp** *local-ip local-port* **interface** *global-port* **Example:** `Device(config)# ip nat inside source static tcp 10.1.1.1 23 interface 23`	(Optional) Enables the use of telnet to the device from the outside.
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations`	(Optional) Displays active NAT.

Reenabling RTSP on a NAT Router

The Real Time Streaming Protocol (RTSP) is a client/server multimedia presentation control protocol that supports multimedia application delivery. Some of the applications that use RTSP include Windows Media Services (WMS) by Microsoft, QuickTime by Apple Computer, and RealSystem G2 by RealNetworks.

When the RTSP protocol passes through a NAT router, the embedded address and port must be translated for the connection to be successful. NAT uses Network Based Application Recognition (NBAR) architecture to parse the payload and translate the embedded information in the RTSP payload.

RTSP is enabled by default. Use the **ip nat service rtsp port** *port-number* command to reenable RTSP on a NAT router if this configuration has been disabled.

Configuring Support for Users with Static IP Addresses

Configuring support for users with static IP addresses enables those users to establish an IP session in a public wireless LAN environment.

Before you begin

Before configuring support for users with static IP addresses, you must first enable NAT on your router and configure a RADIUS server host.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat inside**
5. **exit**
6. **ip nat allow-static-host**
7. **ip nat pool** *name start-ip end-ip* **netmask** *netmask* **accounting** *list-name*
8. **ip nat inside source list** *access-list-number* **pool** *name*
9. **access-list** *access-list-number* **deny ip** *source*
10. **end**
11. **show ip nat translations verbose**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Configures an interface and enters an interface configuration mode.
Step 4	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**ip nat allow-static-host** **Example:** `Device(config)# ip nat allow-static-host`	Enables static IP address support. • Dynamic Address Resolution Protocol (ARP) learning will be disabled on this interface, and NAT will control the creation and deletion of ARP entries for the static IP host.
Step 7	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **accounting** *list-name* **Example:** `Device(config)# ip nat pool pool1 172.16.0.0 172.16.0.254 netmask 255.255.255.0 accounting WLAN-ACCT`	Specifies an existing RADIUS profile name to be used for authentication of the static IP host.
Step 8	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Device(config)# ip nat inside source list 1 pool net-208`	Specifies the access list and pool to be used for static IP support. • The specified access list must permit all traffic.
Step 9	**access-list** *access-list-number* **deny ip** *source* **Example:**	Removes the traffic of the device from NAT. • The *source* argument is the IP address of the device that supports the NAT Static IP Support feature.

	Command or Action	Purpose
	`Device(config)# access-list 1 deny ip 192.168.196.51`	
Step 10	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 11	**show ip nat translations verbose** **Example:** `Device# show ip nat translations verbose`	(Optional) Displays active NAT translations and additional information for each translation table entry, including how long ago the entry was created and used.

Examples

The following is sample output from the **show ip nat translations verbose** command:

```
Device# show ip nat translations verbose

--- 172.16.0.0 10.1.1.1           ---                ---
create 00:05:59, use 00:03:39, left 23:56:20, Map-Id(In): 1, flags: none wlan-flags: Secure
 ARP added, Accounting Start sent Mac-Address:0010.7bc2.9ff6 Input-IDB:Ethernet1/2, use_count:
 0, entry-id:7, lc_entries: 0
```

Configuring the Rate Limiting NAT Translation Feature

SUMMARY STEPS

1. **enable**
2. **show ip nat translations**
3. **configure terminal**
4. **ip nat translation max-entries** {*number* | **all-vrf** *number* | **host** *ip-address number* | **list** *listname number* | **vrf** *name number*}
5. **end**
6. **show ip nat statistics**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip nat translations** **Example:**	(Optional) Displays active NAT.

	Command or Action	Purpose
	`Device# show ip nat translations`	• A specific host, access control list, or VRF instance generating an unexpectedly high number of NAT requests may be the source of a malicious virus or worm attack.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**ip nat translation max-entries** {*number* | **all-vrf** *number* | **host** *ip-address number* | **list** *listname number* | **vrf** *name number*} **Example:** `Device(config)# ip nat translation max-entries 300`	Configures the maximum number of NAT entries that are allowed from the specified source. • The maximum number of allowed NAT entries is 2147483647, although a typical range for a NAT rate limit is 100 to 300 entries. • When you configure a NAT rate limit for all VRF instances, each VRF instance is limited to the maximum number of NAT entries that you specify. • When you configure a NAT rate limit for a specific VRF instance, you can specify a maximum number of NAT entries for the named VRF instance that is greater than or less than that allowed for all VRF instances.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip nat statistics** **Example:** `Device# show ip nat statistics`	(Optional) Displays current NAT usage information, including NAT rate limit settings. • After setting a NAT rate limit, use the **show ip nat statistics** command to verify the current NAT rate limit settings.

	Command or Action	Purpose
		Note The CEF counters associated with the ou packets that are translated and forwarded SW plane in the absence of the correspon out the translation and program the corre translation for subsequent packets that m A route-map based NAT rule does not m flow that matches the given rule is direct undergo translation in the SW plane. This This is an expected behavior when you e these packets that undergo translation in programmed in the HW. This is to facilit flow.

Configuring Bypass NAT Functionality

The Bypass NAT functionality feature reduces the TCAM size by resolving the deny jump issue. To enable the Bypass NAT functionality feature, you must:

- Create a NAT bypass pool by using a reserved loopback address (127.0.0.1).
- Create a new NAT mapping containing a new ACL with all existing deny statements that are converted to permit statements.

You can enable the Bypass NAT functionality by creating new NAT mapping with new ACL mapped to a bypass pool.

To configure the bypass-pool with 127.0.0.1 as reserved loopback address:

```
enable
configure terminal
access-list 60 permit 25.33.0.0 0.0.255.255
ip nat pool bypass-pool 127.0.0.1 127.0.0.1 prefix-length 24
ip nat inside source list 60 pool bypass-pool
end
```

To convert existing configuration with deny statements:

```
enable
configure terminal
ip access list extended nat-acl
deny ip host 10.10.10.10 host 10.77.64.17
permit ip any 10.77.64.0 0.0.15.255
ip nat inside source list nat-acl pool nat-pool
end
```

New converted configuration using bypass pool with permit statements:

```
enable
configure terminal
ip nat pool bypass-pool 127.0.0.1 127.0.0.1 prefix-length 24
ip access list extended nat-bypass-acl
permit ip host 10.10.10.10 host 10.77.64.17
```

```
ip nat inside source list nat-bypass-acl pool bypass-pool
ip access list extended nat-acl
permit ip any 10.77.64.0 0.0.15.255
ip nat inside source list nat-acl pool nat-pool
end
```

Configuration Examples for Configuring NAT for IP Address Conservation

Example: Configuring Static Translation of Inside Source Addresses

The following example shows how inside hosts addressed from the 10.114.11.0 network are translated to the globally unique 172.31.233.208/28 network. Further, packets from outside hosts that are addressed from the 10.114.11.0 network (the true 10.114.11.0 network) are translated to appear from the 10.0.1.0/24 network.

```
ip nat pool net-208 172.31.233.208 172.31.233.223 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 gigabitethernet 0/0/0
 ip address 172.31.232.182 255.255.255.240
 ip nat outside
!
interface gigabitethernet 1/1/1
 ip address 10.114.11.39 255.255.255.0
 ip nat inside
!
access-list 1 permit 10.114.11.0 0.0.0.255
```

The following example shows NAT configured on the provider edge (PE) device with a static route to the shared service for the vrf1 and vrf2 VPNs. NAT is configured as inside source static one-to-one translation.

```
ip nat pool outside 10.4.4.1 10.4.4.254 netmask 255.255.255.0
ip nat outside source list 1 pool mypool
access-list 1 permit 172.16.18.0 0.0.0.255
ip nat inside source static 192.168.121.33 10.2.2.1 vrf vrf1
ip nat inside source static 192.168.121.33.10.2.2.2 vrf vrf2
```

Example: Configuring Dynamic Translation of Inside Source Addresses

The following example shows how inside hosts addressed from either the 192.168.1.0 or the 192.168.2.0 network are translated to the globally unique 172.31.233.208/28 network:

```
ip nat pool net-208 172.31.233.208 172.31.233.223 prefix-length 9
ip nat inside source list 1 pool net-208
!
interface gigabitethernet 0/0/0
 ip address 172.31.232.182 255.255.255.240
 ip nat outside
!
```

```
interface gigabitethernet 1/1/1
 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
access-list 1 permit 192.168.2.0 0.0.0.255
!
```

The following example shows how only traffic local to the provider edge (PE) device running NAT is translated:

```
ip nat inside source list 1 interface gigabitethernet 0/0/0 vrf vrf1 overload
ip nat inside source list 1 interface gigabitethernet 0/0/0 vrf vrf2 overload
!
ip route vrf vrf1 0.0.0.0 0.0.0.0 192.168.1.1
ip route vrf vrf2 0.0.0.0 0.0.0.0 192.168.1.1
!
access-list 1 permit 10.1.1.1.0 0.0.0.255
!
ip nat inside source list 1 interface gigabitethernet 1/1/1 vrf vrf1 overload
ip nat inside source list 1 interface gigabitethernet 1/1/1 vrf vrf2 overload
!
ip route vrf vrf1 0.0.0.0 0.0.0.0 172.16.1.1 global
ip route vrf vrf2 0.0.0.0 0.0.0.0 172.16.1.1 global
access-list 1 permit 10.1.1.0 0.0.0.255
!
```

Example: Using NAT to Allow Internal Users Access to the Internet

The following example shows how to create a pool of addresses that is named net-208. The pool contains addresses from 172.31.233.208 to 172.31.233.233. Access list 1 allows packets with SA from 192.168.1.0 to 192.168.1.255. If no translation exists, packets matching access list 1 is 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-208 172.31.233.208 172.31.233.233 netmask 255.255.255.240
access-list 1 permit 192.168.1.0 0.0.0.255
ip nat inside source list 1 pool net-208 overload
interface gigabitethernet 1/1/1
 ip address 192.168.201.1 255.255.255.240
 ip nat inside
!
interface gigabitethernet 0/0/0
 ip address 192.168.201.29 255.255.255.240
 ip nat outside
!
```

Example: Allowing Overlapping Networks to Communicate Using NAT

Example: Configuring Static Translation of Overlapping Networks

```
ip nat inside source static 192.168.121.33 10.2.2.1
!
interface gigabitethernet 1/1/1
 ip address 10.114.11.39 255.255.255.0
 ip nat inside
```

```
!
interface gigabitethernet 0/0/0
 ip address 172.16.232.182 255.255.255.240
 ip nat outside
!
```

Example: Configuring Dynamic Translation of Overlapping Networks

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 the external network. The pool net-10 is a pool of outside local IP addresses. The **ip nat outside source list 1 pool** *net-10* command translates the addresses of hosts from the outside overlapping network to addresses in that pool.

```
ip nat pool net-208 172.31.233.208 172.31.233.223 prefix-length 28
ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24
access-list 1 permit 10.114.11.0 0.0.0.255
ip nat inside source list 1 pool net-208
ip nat outside source list 1 pool net-10
!
interface gigabitethernet 1/1/1
 ip address 10.114.11.39 255.255.255.0
 ip nat inside
!
interface gigabitethernet 0/0/0
 ip address 172.16.232.182 255.255.255.240
 ip nat outside
!
```

Example: Configuring Server TCP Load Balancing

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 addresses of real hosts. The access list defines the virtual address. If a translation does not exist, TCP packets from serial interface 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
access-list 2 permit 192.168.15.1
ip nat inside destination list 2 pool real-hosts
interface gigabitethernet 0/0/0
 ip address 192.168.15.129 255.255.255.240
 ip nat inside
!
interface serial 0
 ip address 192.168.15.17 255.255.255.240
 ip nat outside
!
```

Example: Enabling Route Maps on Inside Interfaces

```
ip nat inside source static 192.168.201.6 192.168.201.21
!
```

Example: Enabling NAT Route Maps Outside-to-Inside Support

The following example shows how to configure a route map A and route map B to allow outside-to-inside translation for a destination-based Network Address Translation (NAT):

```
ip nat pool POOL-A 192.168.201.4 192.168.201.6 netmask 255.255.255.128
ip nat pool POOL-B 192.168.201.7 192.168.201.9 netmask 255.255.255.128
ip nat inside source route-map MAP-A pool POOL-A reversible
ip nat inside source route-map MAP-B pool POOL-B reversible
```

Example: Configuring NAT of External IP Addresses Only

```
ip nat inside source static network 10.1.1.1 192.168.2510/24 no-payload
ip nat inside source static tcp 10.1.1.1 2000 192.168.1.1 2000 no-payload
ip nat inside source static 10.1.1.1 192.168.1.1 no-payload
ip nat outside source static 10.1.1. 192.168.1.1 no-payload
ip nat outside source static tcp 10.1.1.1 20000 192.168.1.1 20000 no-payload
ip nat outside source static network 10.1.1.1 192.168.251.0/24 no-payload
```

Example: Configuring Support for Users with Static IP Addresses

```
interface gigabitethernet 1/1/1
 ip nat inside
!
ip nat allow-static-host
ip nat pool pool1 172.16.0.0 172.16.0.254 netmask 255.255.255.0 accounting WLAN-ACCT
ip nat inside source list 1 pool net-208
access-list 1 deny ip 192.168.196.51
```

Example: Configuring NAT Static IP Support

The following example shows how to enable static IP address support for the device at 192.168.196.51:

```
interface gigabitethernet 1/1/1
 ip nat inside
!
ip nat allow-static-host
ip nat pool net-208 172.16.1.1 172.16.1.10 netmask 255.255.255.0 accounting WLAN-ACCT
ip nat inside source list 1 pool net-208
access-list 1 deny ip 192.168.196.51
```

Example: Creating a RADIUS Profile for NAT Static IP Support

The following example shows how to create a RADIUS profile for use with the NAT Static IP Support feature:

```
aaa new-model
!
aaa group server radius WLAN-RADIUS
 server 172.16.88.1 auth-port 1645 acct-port 1645
 server 172.16.88.1 auth-port 1645 acct-port 1646
!
aaa accounting network WLAN-ACCT start-stop group WLAN-RADIUS
aaa session-id common
ip radius source-interface gigabitethernet3/0
radius-server host 172.31.88.1 auth-port 1645 acct-port 1646
radius-server key cisco
```

Example: Configuring the Rate Limiting NAT Translation Feature

The following example shows how to limit the maximum number of allowed NAT entries to 300:

```
ip nat translation max-entries 300
```

The following example shows how to limit the VRF instance named “vrf1” to 150 NAT entries:

```
ip nat translation max-entries vrf vrf1 150
```

The following example shows how to limit each VRF instance to 200 NAT entries:

```
ip nat translation max-entries all-vrf 200
```

The following example shows how to limit the VRF instance, “vrf2” to 225 NAT entries, but limit all other VRF instances to 100 NAT entries each:

```
ip nat translation max-entries all-vrf 100
ip nat translation max-entries vrf vrf2 225
```

The following example shows how to limit the access control list named “vrf3” to 100 NAT entries:

```
ip nat translation max-entries list vrf3 100
```

The following example shows how to limit the host at IP address 10.0.0.1 to 300 NAT entries:

```
ip nat translation max-entries host 10.0.0.1 300
```

Example: Setting a Global NAT Rate Limit

The following example shows how to limit the maximum number of allowed NAT entries to 300:

```
ip nat translation max-entries 300
```

Example: Setting NAT Rate Limits for a Specific VRF Instance

The following example shows how to limit the VRF instance named “vrf1” to 150 NAT entries:

```
ip nat translation max-entries vrf vrf1 150
```

Example: Setting NAT Rate Limits for All VRF Instances

The following example shows how to limit each VRF instance to 200 NAT entries:

```
ip nat translation max-entries all-vrf 200
```

The following example shows how to limit the VRF instance, “vrf2” to 225 NAT entries, but limit all other VRF instances to 100 NAT entries each:

```
ip nat translation max-entries all-vrf 100
ip nat translation max-entries vrf vrf2 225
```

Example: Setting NAT Rate Limits for Access Control Lists

The following example shows how to limit the access control list named “vrf3” to 100 NAT entries:

```
ip nat translation max-entries list vrf3 100
```

Example: Setting NAT Rate Limits for an IP Address

The following example shows how to limit the host at IP address 10.0.0.1 to 300 NAT entries:

```
ip nat translation max-entries host 10.0.0.1 300
```

Where to Go Next

- To configure NAT for use with application-level gateways, see the "Using Application Level Gateways with NAT" module.
- To verify, monitor, and maintain NAT, see the "Monitoring and Maintaining NAT" module.
- To integrate NAT with Multiprotocol Label Switching (MPLS) VPNs, see the "Integrating NAT with MPLS VPNs" module.
- To configure NAT for high availability, see the "Configuring NAT for High Availability" module.

Additional References for Configuring NAT for IP Address Conservation

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands: complete command syntax, command mode command history, defaults, usage guidelines, and examples	Cisco IOS IP Addressing Services Command Reference
Application-level gateways	*Using Application Level Gateways with NAT* module
IP access list sequence numbering	IP Access List Entry Sequence Numbering document
RADIUS attributes overview	*RADIUS Attributes Overview and RADIUS IETF Attributes* module

Standards and RFCs

Standard/RFC	Title
IETF Behave Draft NAT MIB	Definitions of Managed Objects for Network Address Translators (NAT) draft-ietf-behave-nat-mib-11
RFC 1597	Internet Assigned Numbers Authority
RFC 1631	The IP Network Address Translation (NAT)

Standard/RFC	Title
RFC 1918	Address Allocation for Private Internets
RFC 2663	IP Network Address Translation (NAT) Terminology and Considerations
RFC 3022	Traditional IP Network Address Translation (Traditional NAT)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services. These services are the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support