IP Addressing: NAT Configuration Guide, Cisco IOS XE Release 2

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Configuring NAT for IP Address Conservation

This module describes how to configure Network Address Translation (NAT) for IP address conservation and 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 router, usually connecting two networks, and translates the private (not globally unique) addresses in the internal network into legal addresses before packets are forwarded onto another network. NAT can be configured to advertise only one address for the entire network to the outside world. This ability provides additional security, 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 and to allow Internet access to internal devices such as mail servers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring NAT for IP Address Conservation

Access Lists

All access lists required for use with the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, refer to the *IP Access List Sequence Numbering* document.

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

NAT Requirements, Objectives, and Interfaces

Before configuring NAT in your network, you should understand on which interfaces NAT will be configured and for what purposes. The following requirements will help you to decide how to configure and use NAT:

1. Define NAT inside and outside interfaces if:
    - Users exist off multiple interfaces.
    - Multiple interfaces connect to the Internet.
2. 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 the use of an application-level gateway (ALG).
    - Redirect TCP traffic to another TCP port or address.
    - Use NAT during a network transition.

Restrictions for Configuring NAT for IP Address Conservation

- NAT Virtual Interfaces (NVIs) are not supported in the Cisco IOS XE software.
- Network Address Translation (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.
- 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 as it attempts 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 desired result.
- A device 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.
- If you specify an access list to use with a NAT command, NAT does not support the **permit ip any any** command that is commonly used in the access list.
- An access list with a port range is not supported on the Cisco ASR 1000 Series Aggregation Services Routers.

- NAT configuration is not supported on the access side of the Intelligent Services Gateway (ISG).
- Using the physical interface address of a device as an address pool is not supported. NAT can share the physical interface 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 **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 (for example 1000 to 4000), the update rate of the pool and mapping statistics in the **show ip nat statistics** is very slow.

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 need to access the Internet. Sites that do not yet possess Network Information Center (NIC)-registered IP addresses must acquire them, and 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 issued 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 so that hackers cannot directly attack the clients. With client addresses hidden, a degree of security is established. Cisco IOS XE NAT gives LAN administrators complete freedom to expand Class A addressing, which 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 a variety of routers for IP address simplification and conservation. In addition, Cisco IOS XE NAT allows the selection of which internal hosts are available for 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.

Purpose of NAT

Two key problems facing the Internet are depletion of IP address space and scaling in routing. NAT is a feature that allows the IP network of an organization to appear from the outside to use 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 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 router configured with NAT will have at least one interface to the inside network and one to the outside network. In a typical environment, NAT is configured at the exit router between a stub domain and a backbone. When a packet leaves 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 and sends an Internet Control Message Protocol (ICMP) host unreachable packet.

Uses of NAT

NAT can be used for the following applications:

- When 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. 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.
- When 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.
- When 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.

NAT Inside and Outside Addresses

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

Similarly, *outside* refers to those networks to which the stub network connects, and which are generally not under the control of the organization. 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--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 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. The address is not necessarily legitimate; it was allocated from the address space 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 was allocated from a globally routable address or network space.

- Inside Source Address Translation, page 5
- Inside Global Addresses Overloading, page 6

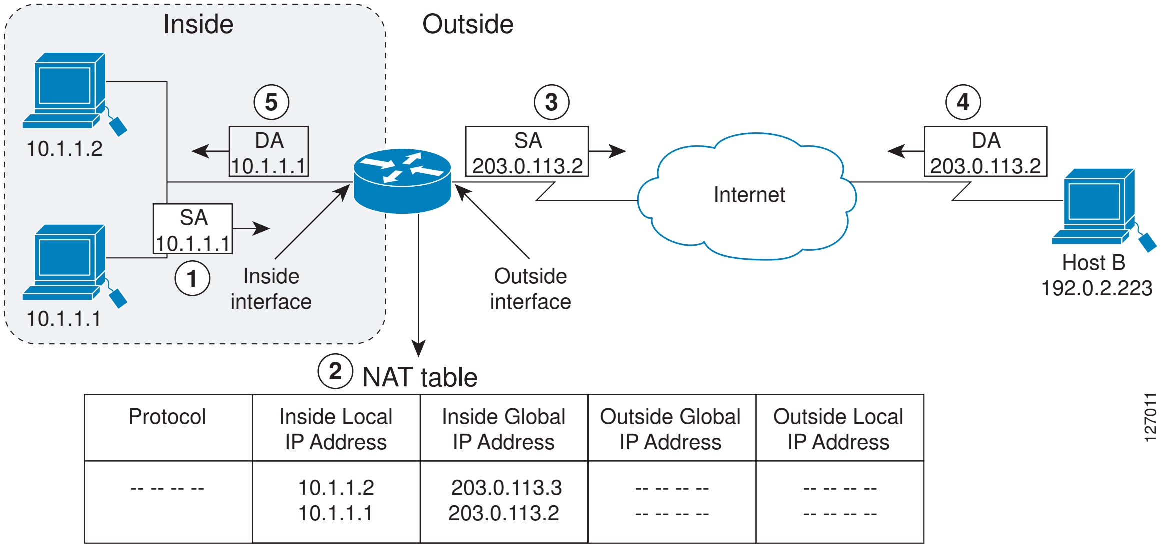
Inside Source Address Translation

You can 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.

The figure below illustrates a router that is translating a source address inside a network to a source address outside the network.

Figure 1 NAT Inside Source Translation



Protocol	Inside Local IP Address	Inside Global IP Address	Outside Global IP Address	Outside Local IP Address
-- -- -- --	10.1.1.2 10.1.1.1	203.0.113.3 203.0.113.2	-- -- -- -- -- -- -- --	-- -- -- -- -- -- -- --

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

1. The user at host 10.1.1.1 opens a connection to host B.
2. The first packet that the router receives from host 10.1.1.1 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) 10.1.1.1 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 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 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 10.1.1.1 and forwards the packet to host 10.1.1.1.

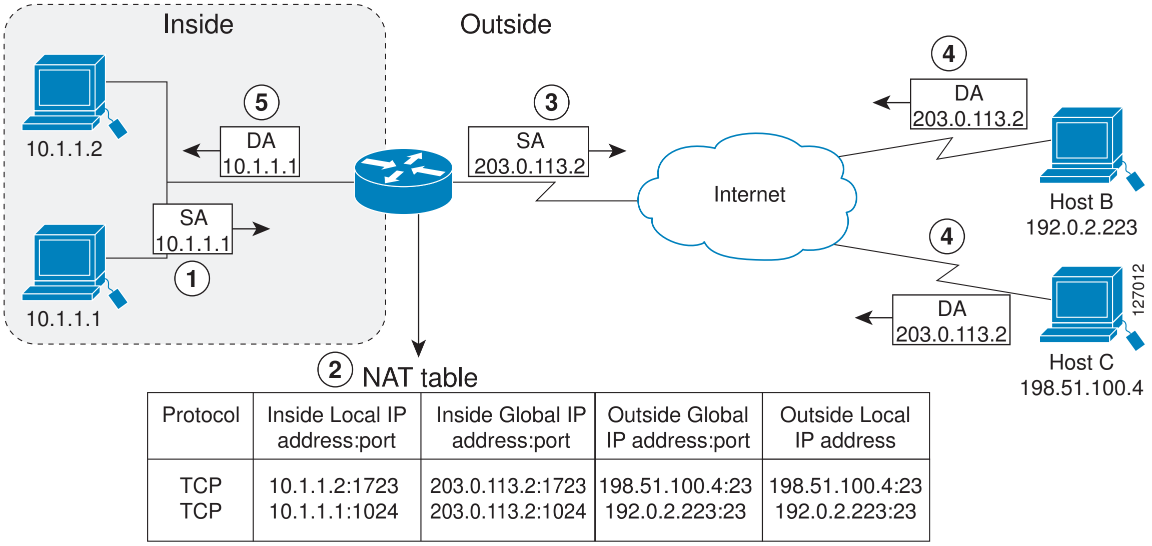
Host 10.1.1.1 receives the packet and continues the conversation. The router performs Steps 2 through 5 for each packet.

Inside Global Addresses Overloading

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, the TCP or UDP port numbers of each inside host distinguish between the local addresses.

The figure below illustrates a NAT operation when one 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 router performs the following process in overloading inside global addresses, as shown in the figure above. Both host B and host C believe that they are communicating with a single host at address 203.0.113.2. They are actually communicating with 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 10.1.1.1 opens a connection to host B.
2. The first packet that the router receives from host 10.1.1.1 causes the router to check its NAT table:

- If no translation entry exists, the router determines that address 10.1.1.1 must be translated, and sets up a translation of inside local address 10.1.1.1 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 the global address back. This type of entry is called an *extended entry*.

3 The router replaces the 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 router receives the packet with the inside global IP address, it performs a NAT table lookup, using the protocol, the inside global address and port, and the outside address and port as a key; translates the address to inside local address 10.1.1.1; and forwards the packet to host 10.1.1.1.

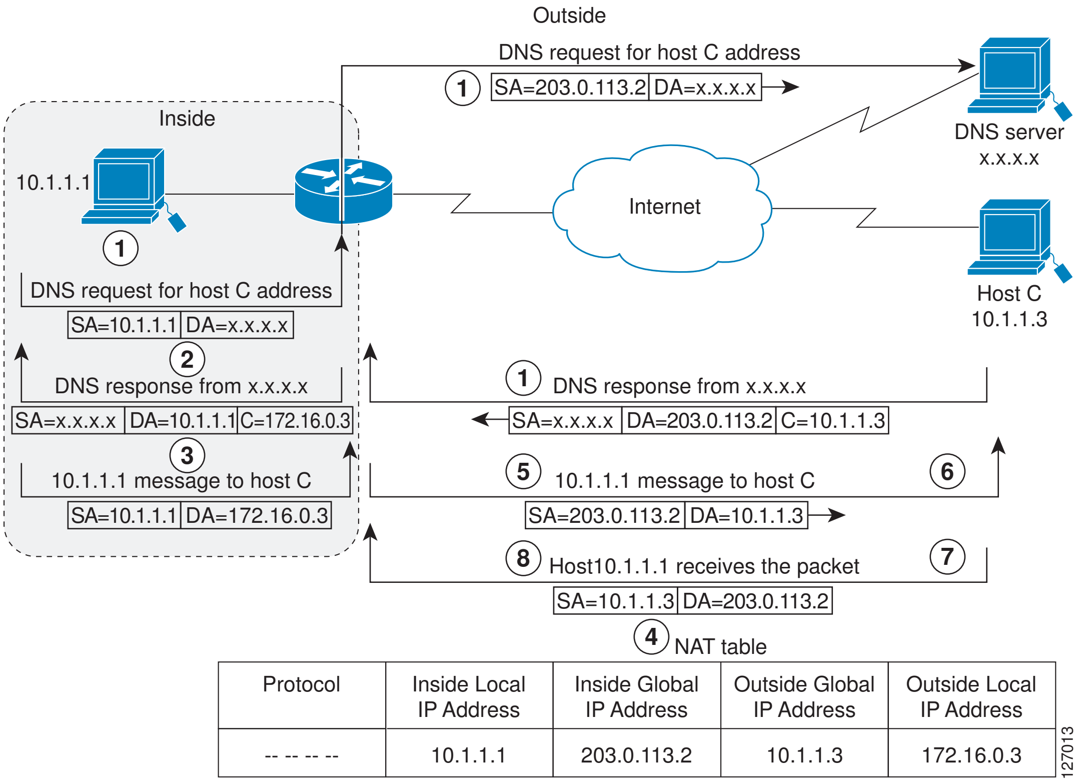
Host 10.1.1.1 receives the packet and continues the conversation. The router performs Steps 2 through 5 for each packet.

Address Translation of Overlapping Networks

NAT is used to translate your IP addresses, which could occur 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 *index overlapping*. You can use NAT to translate inside addresses that overlap with outside addresses.

The figure below shows how NAT translates overlapping networks.

Figure 3 NAT Translating Overlapping Addresses


Outside
DNS request for host C address
1
SA=203.0.113.2 DA=x.x.x.x
DNS server
x.x.x.x
Inside
10.1.1.1
Internet
Host C
10.1.1.3
1
DNS request for host C address
SA=10.1.1.1 DA=x.x.x.x
2
DNS response from x.x.x.x
SA=x.x.x.x DA=10.1.1.1 C=172.16.0.3
3
10.1.1.1 message to host C
SA=10.1.1.1 DA=172.16.0.3
1
DNS response from x.x.x.x
SA=x.x.x.x DA=203.0.113.2 C=10.1.1.3
5
10.1.1.1 message to host C
6
SA=203.0.113.2 DA=10.1.1.3
8
Host10.1.1.1 receives the packet
7
SA=10.1.1.3 DA=203.0.113.2
4
NAT table
127013

Protocol	Inside Local IP Address	Inside Global IP Address	Outside Global IP Address	Outside Local IP Address
-- -- -- --	10.1.1.1	203.0.113.2	10.1.1.3	172.16.0.3

The router performs the following process when translating overlapping addresses:

1. The user at host 10.1.1.1 opens a connection to host C by name, requesting a name-to-address lookup from a Domain Name System (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 10.1.1.3 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.

1. Host 10.1.1.1 opens a connection to 172.16.0.3.
2. The router sets up translations mapping inside local and global addresses to each other, and outside global and local addresses to each other.
3. The router 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 router 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.

Types of NAT

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

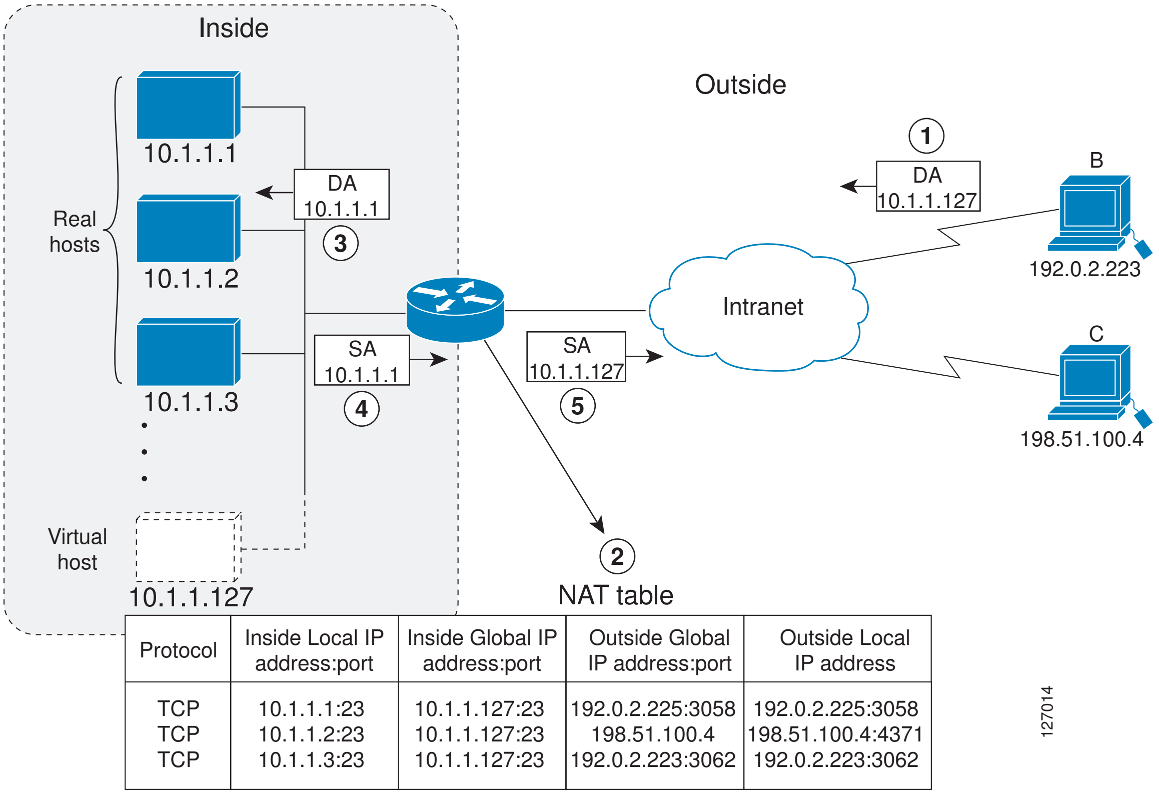
NAT types include:

- 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 of out of a pool of registered IP addresses.
- Overloading--a form of dynamic NAT that maps multiple unregistered IP addresses to a single registered IP address (many to one) using different ports. This method is also known as Port Address Translation (PAT). By using PAT (NAT Overload), thousands of users can be connected to the Internet using only one real global IP address.

TCP Load Distribution for NAT

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. DAs that match an access list are replaced with addresses from a rotary pool. Allocation is done on a roundrobin 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). The figure below illustrates this feature.

Figure 4 NAT TCP Load Distribution



Protocol	Inside Local IP address:port	Inside Global IP address:port	Outside Global IP address:port	Outside Local IP address
TCP	10.1.1.1:23	10.1.1.127:23	192.0.2.225:3058	192.0.2.225:3058
TCP	10.1.2:23	10.1.1.127:23	198.51.100.4	198.51.100.4:4371
TCP	10.1.1.3:23	10.1.1.127:23	192.0.2.223:3062	192.0.2.223:3062

The router performs the following process when translating rotary addresses:

1 The user on host B (192.0.2.223) opens a connection to the virtual host at 10.1.1.127.
2 The router 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 router 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 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.
6 The next connection request will cause the router to allocate 10.1.1.2 for the inside local address.

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.

The NAT Static IP Address Support feature extends the capabilities of public wireless LAN providers to support users configured with a static IP address. By configuring a router to support users with a static IP address, public wireless LAN providers extend their services to a greater number of potential users, which can lead to greater user satisfaction and additional revenue.

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

Remote Authentication Dial-In User Service (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 the User Datagram Protocol (UDP). Generally, the RADIUS protocol is considered a connectionless service. Issues related to server availability, retransmission, and timeouts are handled by RADIUS-enabled devices rather than the transmission protocol.

RADIUS is a client/server 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. RADIUS servers receive user connection requests, authenticate the user, and then return the configuration information necessary for the client to deliver service to the user. A RADIUS server can act as a proxy client to other RADIUS servers or other kinds of authentication servers.

Viruses and Worms that Target NAT

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

Denial-of-Service Attacks

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

Creating NAT Half Entries

How to Configure NAT for IP Address Conservation

The tasks described in this section configure NAT for IP address conservation. No single task in this section is required; however, at least one of the tasks must be performed. More than one of the tasks may need to be performed.

- Configuring Inside Source Addresses, page 11
- Using NAT to Allow Internal Users Access to the Internet, page 16
- Configuring Address Translation Timeouts, page 19
- Allowing Overlapping Networks to Communicate Using NAT, page 21

- Configuring Sever TCP Load Balancing, page 26
- Enabling Route Maps on Inside Interfaces, page 29
- Enabling NAT Route Maps Outside-to-Inside Support, page 30
- Configuring NAT of External IP Addresses Only, page 32
- Configuring Support for Users with Static IP Addresses, page 34
- Configuring the Rate Limiting NAT Translation Feature, page 37

Configuring Inside Source Addresses

Inside source address can be configured for static or dynamic translation. Perform one of the following tasks depending on your requirements:

- Configuring Static Translation of Inside Source Addresses, page 11
- Configuring Dynamic Translation of Inside Source Addresses, page 13
- Troubleshooting Tips, page 16

Configuring Static Translation of Inside Source Addresses

Configure static translation of inside source addresses when you want to allow 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.

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*
10. **ip nat outside**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Router(config)# ip nat inside source static 10.10.10.1 172.16.131.1`	Establishes static translation between an inside local address and inside global address.
Step 4	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for an interface.
Step 6	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/1`	Specifies a different interface and returns to interface configuration mode.

	Command or Action	Purpose
Step 9	**ip address** *ip-address mask* **Example:** `Router(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:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.

Step 11	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and enters privileged EXEC mode.

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 need to access the Internet. The dynamically configured pool IP address may be used as needed and 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 router, the router adds IP aliases for them so that it can answer Address Resolution Protocol (ARP) requests. However, a situation can arise where the router answers packets that are not destined for it, possibly causing a security issue. This can happen when an incoming Internet Control Message Protocol (ICMP) or UDP packet that is destined for one of the aliased addresses does not have a corresponding NAT translation in the NAT table, and the router itself runs a corresponding service, for example, Network Time Protocol (NTP). Such a situation might 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*} [**type** {**match-host** | **rotary**}]
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

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} [**type** {**match-host** | **rotary**}] **Example:** `Router(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:** `Router(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.

	Command or Action	Purpose
Step 5	**ip nat inside source list** *access-list -number* **pool** *name* **Example:** `Router(config)# ip nat inside source list 1 pool net-208`	Establishes dynamic source translation, specifying the access list defined in the prior step.
Step 6	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Router(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:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config-if)# interface GigabitEthernet 0/0/1`	Specifies a different interface and returns to interface configuration mode.

Step 11	**ip address** *ip-address mask*
	Example:
	`Router(config-if)# ip address 172.16.232.182`
	`255.255.255.240`
		Sets a primary IP address for the interface.

	Command or Action	Purpose
Step 12	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.

Step 13	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and enters privileged EXEC mode.

Troubleshooting Tips

Before removing or changing a mapping or NAT pool of global addresses, you must remove the associated access list or remove NAT from the interface. Then, you must use the **clear ip nat translation *** command option to clear all dynamic translations from the translation table.

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 NAT 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

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

Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
	Example:
	`Router(config)# ip nat pool net-208 192.168.202.129 192.168.202.158 netmask 255.255.255.240`
		Defines a pool of global addresses to be allocated as needed.

Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Router(config)# access-list 1 permit 192.168.201.30 0 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:** `Router(config)# ip nat inside source list 1 pool net-208 overload`	Establishes dynamic source translation with overloading, specifying the access list defined in the prior step.
Step 6	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 7	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.201.1 255.255.255.0`	Sets a primary IP address for the interface.
Step 8	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/1`	Specifies a different interface and returns to interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Router(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:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.

Step 13	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and enters privileged EXEC mode.

Configuring Address Translation Timeouts

You can configure address translation timeouts based on your specific configuration of NAT.

By default, dynamic address translations time out after some period of nonuse. 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 that do not use overloading.

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

- Changing the Default Timeouts for Protocol-Based Translations, page 19

Changing the Default Timeouts for Protocol-Based Translations

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 described in this section. If you need to quickly free your global IP address for a dynamic configuration, you should configure a shorter timeout than the default by using the **ip nat translation timeout** command. However, the configured timeout should be longer than the other timeouts configured by using the commands specified in the following task. If a TCP session is not properly closed by a finish (FIN) packet from both sides or during a reset, you should change the default TCP timeout by using the **ip nat translation tcp-timeout** 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*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
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.
Step 4	**ip nat translation udp-timeout** *seconds* **Example:** `Device(config)# ip nat translation udp-timeout 300`	(Optional) Changes the UDP timeout value.
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.
Step 7	**ip nat translation finrst-timeout** *seconds* **Example:** `Device(config)# ip nat translation finrst-timeout 45`	(Optional) Changes the Finish and Reset (FINRST) timeout value. • **finrst-timeout**—The aging time after a TCP session receives both finish-in (FIN-IN) and finish-out (FIN-OUT) messages 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.

	Command or Action	Purpose
Step 9	**ip nat translation syn-timeout** *seconds* **Example:** `Device(config)# ip nat translation syn-timeout 45`	(Optional) Changes the synchronous (SYN) timeout value. • The synchronous timeout or the aging time is used only when a SYN is received on a TCP session. When a synchronous acknowledgment (SYNACK) is received, the timeout changes to TCP timeout.

Allowing Overlapping Networks to Communicate Using NAT

The tasks in this section are grouped because they perform the same action but 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 is implemented:

- Configuring Static Translation of Overlapping Networks, page 21
- Configuring Dynamic Translation of Overlapping Networks, page 23

Configuring Static Translation of Overlapping Networks

Configure static translation of overlapping networks 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 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

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Router(config)# ip nat inside source static 192.168.121.33 10.2.2.1`	Establishes static translation between an inside local address and inside global address.
Step 4	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/1`	Specifies an interface and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Router(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:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 8	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Specifies a different interface and returns to interface configuration mode.
Step 9	**ip address** *ip-address mask* **Example:** `Router(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:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.

Step 11	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and enters privileged EXEC mode.

Configuring Dynamic Translation of Overlapping Networks

Configure dynamic translation of overlapping networks 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 using dynamic translation.

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 outside 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

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

Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
	Example:
	`Router(config)# ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24`
		Defines a pool of global addresses to be allocated as needed.

	Command or Action	Purpose
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Router(config)# access-list 1 permit 10.114.11.0 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 outside source list** *access-list-number* **pool** *name* **Example:** `Router(config)# ip nat outside source list 1 pool net-10`	Establishes dynamic outside source translation, specifying the access list defined in Step 4.
Step 6	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/1`	Specifies an interface and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Router(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:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/2/0`	Specifies a different interface and returns to interface configuration mode.

	Command or Action	Purpose
Step 11	**ip address** *ip-address mask* **Example:** `Router(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:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.

Step 13	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Sever TCP Load Balancing

Perform this task to configure server TCP load balancing by way of destination address rotary translation. These commands allow 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.

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

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **type rotary** **Example:** `Router(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:** `Router(config)# access-list 1 permit 192.168.201.30 0 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:** `Router(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:** `Router(config)# interface GigabitEthernet 0/0/1`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 7	**ip address** *ip-address mask* **Example:** `Router(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:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface Serial 0/0/0`	Specifies a different interface and returns to interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.15.129 255.255.255.240`	Sets a primary IP address for the interface.
Step 12	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside.

Step 13	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and enters privileged EXEC mode.

Enabling Route Maps on Inside Interfaces

For NAT, a route map can be processed instead of an access list. A route map allows you to match any combination of access-list, next-hop IP address, and output interface to determine which pool to use. The ability to use route maps with static translations enables NAT multihoming capability with static address translations. Multihomed internal networks can now host common services such as the Internet and DNS, which are accessed from different outside networks.

Benefits of Using Route Maps on Inside Interfaces

The benefits of using router maps are as follows:

- The ability to configure route map statements provides the option of using IPSec with NAT.
- Translation decisions can be made based on the destination IP address when static translation entries are used.

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

Note Cisco IOS XE software supports only the following commands for using route maps with NAT:

- **match ip address** (with an ACL)
- **match ip next-hop**
- **match interface**

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

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# 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:** `Router(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.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**show ip nat translations** [**verbose**] **Example:** `Router# 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 the deployment of a NAT route map configuration that will allow 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.

- Route Maps Outside-to-Inside Support Design, page 30

Route Maps Outside-to-Inside Support Design

An initial session from the inside to the outside host is required to trigger a NAT. New translation sessions can then be initiated from outside to the inside host that triggered the initial translation.

When route maps are used to allocate global addresses, the global address can allow return traffic, and the return traffic is allowed only if the return traffic matches the defined route map in the reverse direction. The outside-to-inside functionality remains unchanged (by not creating additional entries to allow the return traffic for a route-map-based dynamic entry) unless you configure the **reversible** keyword with the **ip nat inside source** command.

Note

- Access lists with reversible route maps must be configured to match the inside-to-outside traffic.
- Only IP hosts that are part of the route-map configuration will allow outside sessions.
- Outside-to-inside support is not available with PAT.
- Outside sessions must use an access list.
- The **match interface** and **match ip next-hop** commands are not supported for reversible route maps.
- Reversible route maps are not supported for static NAT.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router(config)# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Router(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 inside source route-map** *name* **pool** *name* **reversible** **Example:** `Router(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.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

Configuring NAT of External IP Addresses Only

When you configure NAT of external IP addresses only, 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 router 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 sent out.

- Benefits of Configuring NAT of External IP Addresses Only, page 32

Benefits of Configuring NAT of External IP Addresses Only

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

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** | **upd**} *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* [**overload**] | **static** *local-ip global-ip* **no-payload**}
7. **ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **upd**} *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* [**overload**] | **static** [**network**] *local-network-mask global-network-mask* **no-payload**}
9. **exit**
10. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# 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:** `Router(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 router.
Step 4	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **upd**} *local-ip local-port global-ip global-port* **no-payload**} **Example:** `Router(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 router.
Step 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**} **Example:** `Router(config)# ip nat inside source static 10.1.1.1 192.168.1.1 no-payload`	Disables the packet translation on the inside host router.
Step 6	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** *local-ip global-ip* **no-payload**} **Example:** `Router(config)# ip nat outside source static 10.1.1.1 192.168.1.1 no-payload`	Disables packet translation on the outside host router.

	Command or Action	Purpose
Step 7	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **upd**} *local-ip local-port global-ip global-port* **no-payload**} **Example:** `Router(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 router.
Step 8	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** [**network**] *local-network-mask global-network-mask* **no-payload**} **Example:** `Router(config)# ip nat outside source static network 10.1.1.0 192.168.251.0/24 no-payload`	Disables network packet translation on the outside host router.
Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 10	**show ip nat translations** [**verbose**] **Example:** `Router# show ip nat translations`	Displays active NAT.

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.

The NAT Static IP Support feature extends the capabilities of Public Wireless LAN providers to support users configured with a static IP address. By configuring a router to support users with a static IP address, Public Wireless LAN providers extend their services to a greater number of potential users, which can lead to greater user satisfaction and additional revenue.

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.

This section contains the following procedures:

- Prerequisites, page 35
- Configuring Static IP Support, page 35

Prerequisites

Before configuring support for users with static IP addresses for NAT, you must first enable NAT on your router and configure a RADIUS server host. For additional information on NAT and RADIUS configuration, see the "Additional References, page 41" section.

Configuring Static IP Support

Perform this task to configure the NAT Static IP Support feature.

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. **exit**
11. **show ip nat translations verbose**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Specifies the interface to be configured and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**ip nat allow-static-host** **Example:** `Router(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:** `Router(config)# ip nat pool pool1 172.16.1.1 172.16.255.255 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:** `Router(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:** `Router(config)# access-list 1 deny ip 192.168.196.51`	Removes the router’s own traffic from NAT. • The *source* argument is the IP address of the router that supports the NAT Static IP Support feature.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 11	**show ip nat translations verbose** **Example:** `Router# 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.

Configuring the Rate Limiting NAT Translation Feature

Limiting the number of concurrent NAT operations using the Rate Limiting NAT Translation feature provides users more control over how NAT addresses are used. The Rate Limiting NAT Translation feature can be used to limit the effects of viruses, worms, and DoS attacks.

Because NAT is a CPU-intensive process, router performance can be adversely affected by DoS attacks, viruses, and worms that target NAT. The Rate Limiting NAT Translation feature allows you to limit the maximum number of concurrent NAT requests on a router.

Prerequisites for configuring the Rate Limiting NAT Translation feature

- Classify current NAT usage and determine the sources of requests for NAT. A specific host or access control list generating an unexpectedly high number of NAT requests may be the source of a malicious virus or worm attack.
- Once you have identified the source of excess NAT requests, you can set a NAT rate limit that contains a specific host or access control list, or you can set a general limit for the maximum number of NAT requests allowed regardless of their source.

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

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show ip nat translations** **Example:** `Router# show ip nat translations`	(Optional) Displays active NAT. • A specific host or access control list generating an unexpectedly high number of NAT requests may be the source of a malicious virus or worm attack.
Step 3	**configure terminal** **Example:** `Router# 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:** `Router(config)# ip nat translation max-entries 300`	Configures the maximum number of NAT entries allowed from the specified source. • The maximum number of allowed NAT entries is 2147483647, although a typical range for a NAT rate limit is from 100 to 300 entries.

Step 5	**end**
	Example:
	`Router(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Step 6	**show ip nat statistics** **Example:** `Router# 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 current NAT rate limit settings.

Configuration Examples for Configuring NAT for IP Address Conservation

Example: Configuring Static Translation of Inside Source Addresses

The following example shows how the 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 addressed from the 10.114.11.0 network (the true 10.114.11.0 network) are translated to appear to be 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 0/0/1
 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 router with a static route. NAT is configured as inside source static one-to-one translations.

```
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 2.2.2.1
ip nat inside source static 192.169.121.33.2.2.2.2
```

Example: Configuring Dynamic Translation of Inside Source Addresses

The following example shows how inside hosts addressed from either the 192.168.1.0 or 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 28
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 0/0/1
 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
```

Example: Overloading Inside Global Addresses

The following example creates a pool of addresses named net-208. The pool contains addresses from 172.31.233.208 to 172.31.233.233. Access list 1 allows packets having the SA 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-208 172.31.233.208 172.31.233.233 netmask 255.255.255.240
ip nat inside source list 1 pool net-208 overload
!
interface serial 0/0/0
 ip address 172.31.232.182 255.255.255.240
 ip nat inside
!
interface GigabitEthernet 0/0/0
 ip address 192.168.1.94 255.255.255.0
 ip nat outside
!
access-list 1 permit 192.168.1.0 0.0.0.255
```

Example: Using NAT to Allow Overlapping Networks to Communicate

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 **ip nat outside source list 1 pool net-10** statement 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 0/0/0
 ip address 171.69.232.192 255.255.255.240
 ip nat outside
!
interface GigabitEthernet 0/0/0
 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
```

Example: Configuring 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 the addresses of the real hosts. The access list defines the virtual address. If a translation does not already 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
ip nat inside destination list 2 pool real-hosts
!
interface serial 0/0/0
 ip address 192.168.15.129 255.255.255.240
 ip nat inside
!
interface GigabitEthernet 0/0/1
 ip address 192.168.15.17 255.255.255.240
 ip nat outside
!
access-list 2 permit 192.168.15.1
```

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

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

```
ip nat pool POOL-A 10.1.10.1 10.1.10.126 netmask 255.255.255.128
ip nat pool POOL-B 10.1.20.1 10.1.20.126 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
```

```
!
ip access-list extended ACL-A
 permit ip any 10.1.10.128 0.0.0.127
ip access-list extended ACL-B
 permit ip any 10.1.20.128 0.0.0.127
!
route-map MAP-A permit 10
 match ip address ACL-A
!
route-map MAP-B permit 10
 match ip address ACL-B
```

Example: Configuring NAT of External IP Addresses Only

The following example shows how to translate the packet to an address that can be routed inside the internal network:

```
configure terminal
 ip nat inside source static network 10.1.1.1 192.168.251.0/24 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 10.1.1.1 192.168.1.1 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 4.1.1.0 192.168.251.0/24 no-payload
```

Example: Configuring Support for Users with Static IP Addresses

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

```
interface GigabitEthernet 0/0/1
 ip nat inside
ip nat allow-static-host
ip nat pool pool1 172.16.1.1 171.1.1.10 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
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
NAT commands: complete command syntax, command mode command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
NAT concepts, configuration tasks, and examples	*Cisco IOS XE IP Addressing Services Configuration Guide*
IP access list sequence numbering	"IP Access List Entry Sequence Numbering" module in the *Securing the Data Plane Configuration Guide*

Related Topic	Document Title
RADIUS attributes overview	"RADIUS Attributes Overview and RADIUS IETF Attributes" module in the *Securing User Services Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 1597	*Internet Assigned Numbers Authorit* y
RFC 1631	*The IP Network Address Translation (NAT)*
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)*

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring NAT for IP Address Conservation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for Configuring NAT for IP Address Conservation

Feature Name	Releases	Feature Information
Destination-Based NAT Using Route Maps	Cisco IOS XE Release 2.1	The Destination-Based NAT Using Route Maps feature adds support for destination-based NAT using route maps.
NAT Duplicate Inside Global Address	Cisco IOS XE Release 2.1	The Cisco IOS XE software supports the NAT Duplicate Inside Global Addresses feature.
NAT Host Number Preservation	Cisco IOS XE Release 2.1	For ease of network management, some sites prefer to translate prefixes rather than addresses. These sites want the translated address to have the same host number as the original address. The two prefixes must be of the same length. The NAT Host Number Preservation feature can be enabled by configuring dynamic translation with the address pool of the type, match-host.
NAT Performance Enhancement--Translation Table Optimization	Cisco IOS XE Release 2.1	The NAT Performance Enhancement--Translation Table Optimization feature provides greater structure for storing translation table entries and an optimized lookup in the table for associating table entries to IP connections.

Feature Name	Releases	Feature Information
NAT Route Maps Outside-to-Inside Support	Cisco IOS XE Release 2.2	The NAT Route Maps Outside-to-Inside Support feature enables the deployment of a NAT route map configuration that will allow IP sessions to be initiated from the outside to the inside.
NAT Static IP Support	Cisco IOS XE Release 2.1	The NAT Static IP Support feature provides support for users with static IP addresses, enabling those users to establish an IP session in a public wireless LAN environment.
NAT Timers	Cisco IOS XE Release 2.1	The NAT Timers feature allows you to change the amount of time after which NAT translations time out.
NAT Translation of External IP Addresses Only	Cisco IOS XE Release 2.1	You can use the NAT Translation of External IP Address Only feature to configure NAT to ignore all embedded IP addresses for any application and traffic type.
Rate Limiting NAT Translation	Cisco IOS XE Release 2.1	The Rate Limiting NAT Translation feature provides the ability to limit the maximum number of concurrent NAT operations on a router. In addition to giving users more control over how NAT addresses are used, the Rate Limiting NAT Translation feature can be used to limit the effects of viruses, worms, and DoS attacks.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Using Application-Level Gateways with NAT

This module describes the basic tasks to configure an application-level gateway (ALG) with Network Address Translation (NAT). This module also provides information about the protocols that use ALG for IP header translation.

NAT performs translation services on any TCP/UDP traffic that does not carry source and destination IP addresses in the application data stream. The protocols that do not carry the source and destination IP addresses include HTTP, TFTP, telnet, archie, finger, Network Time Protocol (NTP), Network File System (NFS), and remote copy (rcp).

Specific protocols that embed IP address information within the payload require support of an ALG. NAT requires a variety of ALGs to handle application data stream (Layer 7) protocol-specific services such as translating embedded IP addresses and port numbers in the packet payload and extracting new connection/session information from control channels.

NAT supports virtual routing and forwarding (VRF) for protocols that have a supported ALG.

The Support for IPsec ESP Through NAT feature provides the ability to support multiple concurrent IPsec Encapsulating Security Payload (ESP) tunnels or connections through a NAT device configured in Overload or Port Address Translation (PAT) mode.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Using Application-Level Gateways with NAT

- Before performing the tasks in this module, you should be familiar with the concepts described in the “Configuring NAT for IP Address Conservation” module.
- You should have already configured all access lists required for use with the tasks in this module.
- You should verify that Session Initiation Protocol (SIP) and H.323 are not disabled. SIP and H.323 are enabled by default.

Information About Configuring Application-Level Gateways with NAT

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

IPsec

IPsec is a set of extensions to the IP family in a framework of open standards for ensuring secure private communications over the Internet. Based on standards developed by the IETF, IPsec ensures confidentiality, integrity, and authenticity of data communications across the public network and provides cryptographic security services.

Secure tunnels between two peers, such as two routers, are provided and decisions are made as to which packets are considered sensitive and should be sent through these secure tunnels and which parameters should be used to protect these sensitive packets by specifying the characteristics of these tunnels. When the IPsec peer receives a sensitive packet, the peer sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer.

IPsec using Encapsulating Security Payload (ESP) can pass through a router running NAT without any specific support from it as long as Network Address Port Translation (NAPT) or address overloading is not configured.

There are a number of factors to consider when attempting an IPsec VPN connection that traverses a NAPT device that represents multiple private internal IP addresses as a single public external IP address. These factors include capabilities of the VPN server and client, capabilities of the NAPT device, and whether more than one simultaneous connection is attempted across the NAPT device.

There are two possible methods for configuring IPsec on a device with NAPT:

- Encapsulate IPsec in a Layer 4 protocol such as TCP or UDP. In this case, IPsec is *sneaking* through NAT. The NAT device is unaware of the encapsulation.
- Add IPsec-specific support to NAPT. IPsec works with NAT in this case as opposed to *sneaking* through NAT. The NAT Support for IPsec ESP—Phase II feature provides support for Internet Key Exchange (IKE) and ESP without encapsulation in tunnel mode through a device configured with NAPT.

We recommend that TCP and UDP be used when conducting IPsec sessions that traverse a NAPT device. However, not all VPN servers or clients support TCP or UDP.

- Benefits of Configuring NAT IPsec, page 49

Benefits of Configuring NAT IPsec

- NAT enables customers to deploy private IP addresses within their network and perform translation to public IP addresses when connecting to the Internet or interconnecting with another corporate network.
- ESP entries in the translation table are normally delayed from being transmitted until a reply is received from the destination. With predictable security parameter indexes (SPIs) and SPI matching, the delay can be eliminated because the SPI entries are matched. Some third-party concentrators require both the source and incoming ports to use port 500. Use of the **preserve-port** keyword with the **ip nat service** command preserves the ports rather than changing them, which is required with regular NAT.

SPI Matching

SPI matching is used to establish VPN connections between multiple pairs of destinations. NAT entries will immediately be placed in the translation table for endpoints matching the configured access list.

NAT Support for Application-Level Gateways

The following section provides information on NAT support for ALGs.

The features described in the following subsections are enabled by default unless otherwise noted; no configuration is necessary:

- NAT Support of Skinny Client Control Protocol, page 50
- NAT SCCP Video Support, page 50
- NAT vTCP ALG Support, page 50
- NAT NetBIOS ALG Support, page 51

NAT Support of Skinny Client Control Protocol

Cisco IP phones use the Skinny Client Control Protocol (SCCP) to connect with and register to Cisco Unified CallManager.

To deploy NAT between the IP phone and the Cisco Unified CallManager in a scalable environment, NAT must detect SCCP and understand the information that is passed within these messages. Messages that flow back and forth include the IP address and the port information to identify other IP phone users with whom calls can be placed.

The SCCP client to the Cisco Unified CallManager communication typically flows from inside to outside. The Domain Name System (DNS) is used to resolve the Cisco Unified CallManager IP address connection when the Cisco Unified CallManager is configured on the inside (behind the NAT device), or when static NAT is configured to reach the Cisco Unified CallManager on the inside.

When an IP phone attempts to connect to the Cisco Unified CallManager and matches the configured NAT rules, NAT translates the original source IP address and replaces it with one from the configured pool. This new IP address is reflected in the Cisco Unified CallManager and is visible to other IP phone users.

NAT SCCP Video Support

NAT provides SCCP video message translation support.

NAT vTCP ALG Support

NAT provides virtual TCP (vTCP) support to handle TCP segmentation and reassembling for ALG. When a Layer 7 protocol uses TCP for transportation, the payload can be segmented due to various reasons, such as Maximum Segment Size (MSS), application design, and TCP window size. Proper recognition of these TCP segments is required to perform parsing. Therefore, a generic framework called vTCP is used by various ALGs to perform TCP segmentation.

Some applications such as SIP and NAT require the entire payload to rewrite embedded data. In addition, ALGs are not developed to consider data splitting between the packets, which is required for the firewall. Therefore, vTCP is also required for the firewall without any changes to current ALGs. NAT and the firewall ALG configuration activate the vTCP configuration.

vTCP does not support data channel traffic. To protect system resources, vTCP does not support reassembled messages larger than 8 KB.

- NAT ALG--vTCP for SIP, page 51

NAT ALG--vTCP for SIP

Cisco IOS XE Release 3.2S supports the NAT ALG—vTCP for SIP feature. With the introduction of vTCP support for SIP, individual TCP segments will be chained together to form a complete SIP message and passed to the SIP parser. vTCP also supports acknowledgement (ACK) and reliable transmission of buffered data. ACK is a SIP method that is used to acknowledge that the received message is valid and accepted.

The NAT ALG—vTCP for SIP feature does not support:

- Data channel traffic.
- Reassembled Layer 7 messages that are larger than 8 KB.
- TCP segments that are larger than 8 KB.
- vTCP SIP trunk calls.

NAT NetBIOS ALG Support

NAT application awareness includes support for Network Basic Input Output System (NetBIOS) applications. A NetBIOS ALG translates IP addresses and port numbers embedded in NetBIOS packets when a NAT mapping is processed. The NAT NetBIOS ALG Support feature introduced the **show platform hardware qfp** [**active** | **standby**] **feature alg statistics netbios** command to display NetBIOS-specific information for a device and the **match protocol netbios** command to configure network-based application recognition (NBAR) to match the NetBIOS traffic.

NAT RCMD ALG Support

NAT application awareness includes support for remote command (RCMD) execution service applications, remote login (rlogin), remote shell (rsh) protocol, and remote execution (rexec). An RCMD ALG translates IP addresses and port numbers embedded in RCMD application packets when a NAT mapping is processed. The NAT RCMD ALG Support feature introduced the **show platform software trace message** *process* **qfp active** command to display RCMD-specific information for a device.

NAT RTSP ALG Support

NAT application awareness includes support for Real-Time Streaming Protocol (RTSP) applications. An RTSP ALG translates IP addresses and port numbers embedded in RTSP packets when a NAT mapping is processed.

NAT Support for SIP—Voice and Multimedia over IP Networks

SIP is a protocol developed by the IETF Multiparty Multimedia Session Control (MMUSIC) Working Group. The Cisco Session Initiation Protocol (SIP) functionality equips Cisco devices to signal the setup of voice and multimedia calls over IP networks. SIP provides an alternative to H.323 within VoIP internetworking software.

Session Description Protocol (SDP) is a protocol that describes multimedia sessions. SDP may be used in SIP message bodies to describe multimedia sessions used for creating and controlling multimedia sessions with two or more participants.

The NAT Support for SIP feature allows SIP embedded messages passing through a device that is configured with NAT to be translated and encoded back to the packet. An ALG is used with NAT to translate SIP messages.

Note By default, support for 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 as it attempts to interpret the packet as a SIP call message.

NAT ALG--SIP Multiple Media Line Support

The NAT ALG—SIP Multiple Media Line Support feature supports a maximum of five media lines in SDP. These media lines can be a combination of audio, video, and data.

SDP describes multimedia sessions. The description includes the media type, the transport port to which the media stream is sent, the transport protocol, and the media format. All media descriptions start with the media line attribute “m=” and terminate at the end of the session description. There can be multiple media lines depending on the services supported by SIP peers.

The NAT ALG—SIP Multiple Media Line Support feature uses the transport port information in the media description to create a door for NAT. Doors are transient structures that allow incoming traffic that matches a specific criterion. A door is created when there is not enough information to create a complete NAT session entry. A door contains information about the source IP address and destination IP address and the destination port. However, it does not have information about the source port. When media data arrives, the source port information is known and the door is promoted to a real NAT session.

When a door receives information about the source IP address, destination IP address, source port, destination port, and protocol from the incoming packet, it will change itself from a door to a full NAT session. A door and a full NAT session are saved in different databases. When a door becomes a full NAT session, the door entry is removed from the door database and a new NAT entry is added to the NAT session database.

NAT ALG--SIP REFER Method

The NAT ALG—SIP REFER Method feature is used for call transfers. A REFER message is used to refer to a peer. The REFER method indicates that the recipient of a call, identified by a request Uniform Resource Identifier (URI), must contact a third party using the contact information provided in the request.

The NAT ALG—SIP REFER Method feature supports two types of call transfers, unattended (blind) transfer and attended (consultative) transfer.

NAT ALG--SIP Trunking Support

A SIP trunk is a direct connection of an IP PBX to a service provider over an IP network using SIP. There can be numerous concurrent calls in a SIP trunk. During the call setup process, all calls use the same control channel for call establishment. More than one call uses the same control channel for call setup. When the same control channel is used by more than one call, the stateful information stored in the control channel sessions becomes unreliable. SIP stateful information consists of media channel information such as the IP address and port number used by client/server endpoints to send media data. The media channel information is used to create a door for the data channel in NAT. Because multiple calls use the same control channel for call setup, there will be multiple sets of media data. The NAT ALG—SIP Trunking

Support feature uses a local database to store all media-related information within a SIP trunk. Call IDs of each call are used to index this local database.

TCP segmentation in a SIP trunk can cause unexpected behavior that includes packet drops, TCP reset, and slow response.

NAT SIP Extended Methods

NAT supports extended methods for SIP.

ALG--SCCP Version 17 Support

The ALG—SCCP Version 17 Support feature enables the SCCP ALG to parse SCCP Version 17 packets. Cisco Unified Communications Manager 7.0 and the IP phones that use Cisco Unified Communications Manager 7.0 support only SCCP Version 17 messages. The format of SCCP changed from Version 17 to support IPv6. The SCCP ALG checks for the SCCP version in the prefix of a message before parsing it according to the version. The SCCP message version is extracted from the message header and if it is greater than Version 17, the message is parsed by using the Version 17 format and the IPv4 address and port information is extracted. The SCCP ALG supports the inspection and translation of IPv4 address information in SCCP messages.

Note IPv6 address inspection and translation are not supported.

The IP address format of the following SCCP ALG-handled messages changed in Version 17:

- StationOpenMultiMediaReceiveChannelAck
- StationOpenReceiveChannelAckMessage
- StationRegisterMessage
- StationStartMediaTransmissionAckMessage
- StationStartMultiMediaTransmissionAckMessage
- StationStartMediaTransmissionMessage
- StationStartMultiMediaTransmissionMessage

Basic H.323 ALG Support

H.323 is a recommendation published by the ITU-T defining a series of network elements and protocols for multimedia transmission through packet-based networks. H.323 defines a number of network elements used in multimedia transmission.

Although most H.323 implementations today utilize TCP as the transport mechanism for signaling, H.323 Version 2 enables basic UDP transport.

- H.323 Terminal—This element is an endpoint in the network, providing two-way communication with another H.323 terminal or gateway.
- H.323 Gateway—This element provides protocol conversion between H.323 terminals and other terminals that do not support H.323.
- H.323 Gatekeeper—This element provides services like address translation, network access control, and bandwidth management and account for H.323 terminals and gateways.

The following core protocols are described by the H.323 specification:

- H.225—This protocol describes call signaling methods used between any two H.323 entities to establish communication.
- H.225 Registration, Admission, and Status (RAS)—This protocol is used by the H.323 endpoint and gateway for address resolution and admission control services.
- H.245—This protocol is used for exchanging the capabilities of multimedia communication and for the opening and closing of logical channels for audio, video, and data.

In addition to the protocols listed, the H.323 specification describes the use of various IETF protocols like the Real Time Transport (RTP) protocol and audio (G.711, G.729, and so on) and video (H.261, H.263, and H.264) codecs.

NAT requires a variety of ALGs to handle Layer 7 protocol-specific services such as translating embedded IP addresses and port numbers in the packet payload and extracting new connection/session information from control channels. The H.323 ALG performs these specific services for H.323 messages.

NAT Support of H.323 v2 RAS

NAT supports all H.225 and H.245 message types, including those sent in the Remote Access Service (RAS) protocol. RAS provides a number of messages that are used by software clients and VoIP devices to register their location, request assistance in call setup, and control bandwidth. RAS messages are directed toward an H.323 gatekeeper.

Some RAS messages include IP addressing information in the payload, typically meant to register a user with the gatekeeper or to learn about another user already registered. If these messages are not known to NAT, they cannot be translated to an IP address that is visible to the public.

Embedded IP addresses can be inspected for potential address translation.

ALG—H.323 v6 Support

ALG—H.323 v6 supports the parsing of H.323 v6 packets and the inspection and translation of IPv4 address information in H.323 messages. The basic H.323 ALG supports only the parsing of H.323 v4 messages. H.323 v6 extends the basic H.323 ALG support to recognize the new message format and to handle new fields that contain the IPv4 address information.

H.323 v6 consists of core protocols, H.225.0 v6, and H.245 v13 and uses an assigned gatekeeper for transmission.

ALG—H.323 v6 does not support:

- Stream Control Transmission Protocol (SCTP)—This protocol provides similar services like TCP or UDP.
- Configuring of the H225 port number.

NAT NetMeeting Directory (LDAP)

NAT provides ALG support for NetMeeting directory Lightweight Directory Access Protocol (LDAP) Version 2 and Version 3 messages.

Users can establish calls/connections among each other directly or through a NetMeeting directory. NetMeeting implements a series of LDAP messages for users to register themselves and perform lookups of other NetMeeting users against the directory. These messages include IP address information.

Before a NAT device can use a NetMeeting directory, NAT needs to understand the LDAP messages and perform standard NAT processing against the IP address information within these messages.

NAT DNS ALG Support

NAT application awareness includes support for the Domain Name System (DNS). An application-level gateway (ALG) translates IP addresses and port numbers embedded in the DNS payload when a NAT mapping is processed.

With CSCuc05660, for DNS payloads that are address-translated, the DNS time to live (TTL) value in CNAME entries is passed through. Before CSCuc05660 and before support for the **ip nat service dns-reset-ttl** command was added, the TTL value in the CNAME entries was reset by default.

NAT ICMP ALG Support

NAT application awareness includes translation support for the Internet Control Message Protocol (ICMP). An ALG translates data embedded in the ICMP payload when a NAT mapping is processed.

NAT TFTP ALG Support

NAT application awareness includes support for TFTP. A TFTP ALG creates a path for the TFTP data to traverse the NAT-enabled device.

NAT FTP ALG Support

NAT application awareness includes support for FTP. An FTP ALG performs translation for the IP addresses and TCP port information embedded in the payload of an FTP control session.

How to Configure Application-Level Gateways with NAT

Configuring IPsec ESP Through NAT

The IPsec ESP Through NAT feature provides the ability to support multiple concurrent IPsec Encapsulating Security Payload (ESP) tunnels or connections through a Network Address Translation (NAT) device configured in Overload or Port Address Translation (PAT) mode.

Note IPsec can be configured for any type of NAT configuration, not just static NAT configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static esp** *local-ip* **interface** *type number*
4. **exit**
5. **show ip nat translations**

DETAILED STEPS

	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 esp** *local-ip* **interface** *type number* **Example:** `Device(config)# ip nat inside source static esp 192.0.2.23 interface gigabitethernet 0/0/0`	Establishes the IPsec Encapsulating Security Payload (ESP) (tunnel mode) support.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 5	**show ip nat translations** **Example:** `Device# show ip nat translations`	(Optional) Displays active NATs.

- Restrictions, page 56

Restrictions

- Network Address Translation (NAT) will translate only embedded IPv4 addresses.
- The multicast gatekeeper discovery mechanism is not supported.

Enabling the Preserve Port

This task is used for IPsec traffic using port 500 for the source port and incoming port. Some third-party concentrators require both the source and incoming ports to use port 500. Use of the **preserve-port** keyword with the **ip nat service** command preserves the ports rather than changing one, which is required with regular Network Address Translation (NAT).

Note This task is required by certain VPN concentrators, but will cause problems with other concentrators. Cisco VPN devices generally do not use this feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service list** *access-list-number* **IKE preserve-port**
4. **end**

DETAILED STEPS

	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 service list** *access-list-number* **IKE preserve-port** **Example:** `Device(config)# ip nat service list 10 IKE preserve-port`	Preserves the UDP port in IKE packets.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Disabling SPI Matching on the NAT Device or Changing the Default Port

SPI matching is used to establish VPN connections between multiple pairs of destinations. NAT entries are immediately placed in the translation table for endpoints that match the configured access list.

The generation of SPIs that are predictable and symmetric is enabled. SPI matching should be used in conjunction with NAT devices when multiple Encapsulating Security Payload (ESP) connections across a NAT device are desired.

SPI matching is enabled by default for listening on port 2000. You can use this task to either change the default port or to disable SPI matching.

Cisco software must be running on both the source device and the remote gateway, enabling parallel processing.

Note Security parameter index (SPI) matching must be configured on the Network Address Translation (NAT) device and on both endpoint devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service list** *access-list-number* **esp spi-match**
4. **no ip nat service list** *access-list-number* **esp spi-match**
5. **end**

DETAILED STEPS

	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 service list** *access-list-number* **esp spi-match** **Example:** `Device(config)# ip nat service list 10 esp spi-match`	Specifies a port other than the default port. • This example shows how to enter ESP traffic matching list 10 into the NAT table, based on the assumption that both devices are Cisco devices and are configured to provide matchable SPIs.
Step 4	**no ip nat service list** *access-list-number* **esp spi-match** **Example:** `Device(config)# no ip nat service list 10 esp spi-match`	Disables SPI matching.

	Command or Action	Purpose

Step 5	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Enabling SPI Matching on Endpoints

Cisco software must be running on both the source device and the remote gateway, enabling parallel processing.

Note Security parameter index (SPI) matching must be configured on the Network Address Translation (NAT) device and on both endpoint devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec nat-transparency spi-matching**
4. **end**

DETAILED STEPS

	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	**crypto ipsec nat-transparency spi-matching** **Example:** `Device(config)# crypto ipsec nat-transparency spi-matching`	Enables SPI matching on both endpoints.

	Command or Action	Purpose

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Specifying a Port for NAT Translation

The following task describes how to configure Skinny Client Control Protocol (SCCP) for a Cisco IP phone to Cisco Unified CallManager communication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service skinny tcp port** *number*
4. **end**

DETAILED STEPS

	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 service skinny tcp port** *number* **Example:** `Device(config)# ip nat service skinny tcp port 20002`	Configures the Skinny protocol on the specified TCP port.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for Using Application-Level Gateways with NAT

Example: Configuring IPsec ESP Through NAT

The following example shows NAT configured on a device with a static route. NAT is configured as inside source static one-to-one translations.

```
ip nat pool outside 192.0.2.1 192.0.2.14 netmask 255.255.255.0
ip nat outside source list 1 pool mypool
access-list 1 permit 192.0.2.3 0.0.0.255
ip nat inside source static esp 192.0.2.23 interface gigabitethernet 0/0/0
ip nat inside source static esp 192.0.2.21 interface gigabitethernet 0/0/1
```

Example: Enabling the Preserve Port

The following example shows how to configure TCP port 500 of the third-party concentrator:

```
ip nat service list 10 IKE preserve-port
```

Example: Disabling SPI Matching on the NAT Device or Changing the Default Port

```
ip nat service list 10 esp spi-match
no ip nat service list 10 esp spimatch
```

Example: Enabling SPI Matching on Endpoints

```
crypto ipsec nat-transparency spi-matching
```

Example: Specifying a port for NAT Translation

```
ip nat service skinny tcp port 20002
```

Additional References for Using Application-Level Gateways with NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
NAT commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
Configuring NAT for IP Address Conservation	"Configuring NAT for IP Address Conservation" module
IP Addressing Services configuration tasks	*Cisco IOS XE IP Addressing Services Configuration Guide*
NAT and Firewall ALG support	*NAT and Firewall ALG Support on Cisco ASR 1000 Series Routers* matrix
SIP Call Flows	*"SIP Call Flows"* document

Standards and RFCs

Standard/RFC	Title
RFC 3515	*The Session Initiation Protocol (SIP) Refer Method*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Using Application-Level Gateways with NAT

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2 Feature Information for Using Application-Level Gateways with NAT

Feature Name	Releases	Feature Information
ALG--H.323 v6 Support	Cisco IOS XE Release 3.6S	The ALG-H.323 v6 supports the parsing of H.323 v6 packets and the inspection and translation of IPv4 address information in H.323 messages.
ALG--SCCP Version 17 Support	Cisco IOS XE Release 3.5S	The ALG-SCCP Version 17 Support feature enables the SCCP ALG to parse SCCP Version 17 packets. Cisco Unified Communications Manager 7.0 and IP phones that use Cisco Unified Communications Manager 7.0 support only SCCP Version 17 messages. The SCCP Version 17 packets support IPv6 packets. The SCCP ALG supports the inspection and translation of IPv4 address information in SCCP messages.
NAT ALG--SIP REFER Method	Cisco IOS XE Release 3.2S	The NAT ALG--SIP REFER method feature supports two types of call transfers, unattended (blind) transfer and attended (consultative) transfer.
NAT ALG--SIP Trunking Support	Cisco IOS XE Release 3.2S	The NAT ALG--SIP Trunking Support feature uses a local database to store all media-related information within a SIP trunk. Call IDs of each call are used to index this local database.
NAT Basic H.323 ALG Support	Cisco IOS XE Release 2.1	NAT requires a variety of ALGs to handle Layer 7 protocol-specific services such as translating embedded IP addresses and port numbers in the packet payload and extracting new connection/session information from control channels. The NAT Basic H.323 ALG support feature provides these specific services for H.323 messages.

Feature Name	Releases	Feature Information
NAT DNS ALG Support	Cisco IOS XE Release 2.1	The NAT DNS ALG Support feature supports translation of DNS packets.
NAT FTP ALG Support	Cisco IOS XE Release 2.1	The NAT FTP ALG Support feature supports translation of FTP packets.
NAT H.323 RAS	Cisco IOS XE Release 2.4	NAT supports all H.225 and H.245 message types, including those sent in the Registration, Admission, and Status (RAS) protocol. RAS provides a number of messages that are used by software clients and VoIP devices to register their location, request assistance in call setup, and control bandwidth. The RAS messages are directed toward an H.323 gatekeeper.
NAT ICMP ALG Support	Cisco IOS XE Release 2.1	The NAT ICMP ALG Support feature supports translation of ICMP packets.
NAT NetBIOS ALG Support	Cisco IOS XE Release 3.1S	NAT provides Network Basic Input Output System (NetBIOS) message translation support. The NAT NetBIOS ALG Support feature introduced the following command to display NetBIOS-specific information for a device: **show platform hardware qfp** [**active** | **standby**] **feature alg statistics netbios**.
NAT NetMeeting Directory (LDAP)	Cisco IOS XE Release 2.4	The NAT NetMeeting Directory (LDAP) feature provides ALG support for NetMeeting directory LDAP messages.

Feature Name	Releases	Feature Information
NAT RCMD ALG Support	Cisco IOS XE Release 3.1S	NAT provides remote command execution service (RCMD) message translation support. The NAT RCMD ALG Support feature introduced the following command to display RCMD-specific information for a device: **show platform software trace message** *process* **qfp active**.
NAT RTSP ALG Support	Cisco IOS XE Release 3.1S	The NAT RTSP ALG Support feature provides RTSP message translation support.
NAT--SCCP for Video	Cisco IOS XE Release 2.4	The NAT--SCCP for Video feature provides SCCP video message translation support.
NAT--SIP ALG Enhancement for T.38 Fax Relay	Cisco IOS XE Release 2.4.1	The NAT--SIP ALG Enhancement for T.38 Fax Relay feature provides translation support for SIP ALG support of T.38 Fax Relay over IP.
NAT--SIP Extended Methods	Cisco IOS XE Release 2.4	The NAT--SIP Extended Methods feature supports extended methods for SIP.
NAT Support of IP Phone to Cisco CallManager	Cisco IOS XE Release 2.1	The NAT Support of IP Phone to Cisco CallManager feature adds NAT support for configuring Cisco SCCP for a Cisco IP phone-to-Cisco CallManager communication.
NAT Support for IPsec ESP--Phase II	Cisco IOS XE Release 2.1	The NAT Support for IPsec ESP--Phase II feature provides support for Internet Key Exchange (IKE) and ESP without encapsulation in tunnel mode through a device configured with NAPT.
NAT Support for SIP	Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.2S	The NAT Support for SIP feature adds the ability to deploy NAT between VoIP solutions based on SIP.
NAT TFTP ALG Support	Cisco IOS XE Release 2.1	The NAT TFTP ALG Support feature supports translation of TFTP packets.

Feature Name	Releases	Feature Information
NAT VRF-Aware ALG Support	Cisco IOS XE Release 2.5	The NAT VRF-Aware ALG Support feature supports VPN routing and forwarding (VRF) for protocols that have a supported ALG.
NAT vTCP ALG Support	Cisco IOS XE Release 3.1S Cisco IOS XE Release 3.2S	The NAT vTCP ALG Support feature provides vTCP support to handle TCP segmentation and reassembling for ALG.
Support for IPsec ESP Through NAT	Cisco IOS XE Release 2.1	The Support for IPsec ESP Through NAT feature provides the ability to support multiple, concurrent IPsec ESP tunnels or connections through a NAT device configured in Overload or PAT mode.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

MSRPC ALG Support for Firewall and NAT

The MSRPC ALG Support for Firewall and NAT feature provides support for the Microsoft (MS) Remote Procedure Call (RPC) application-level gateway (ALG) on the firewall and Network Address Translation (NAT). The MSRPC ALG provides deep packet inspection (DPI) of the MSRPC protocol. The MSRPC ALG works in conjunction with a provisioning system to allow the network administrator to configure match filters to define match criteria that can be searched in an MSRPC packet.

Prerequisites for MSRPC ALG Support for Firewall and NAT

- You must enable the Cisco IOS XE firewall and NAT before applying the MSRPC ALG on packets.

Restrictions for MSRPC AIC Support for Firewall and NAT

- Only TCP-based MSRPC is supported.
- You cannot configure the **allow** and **reset** commands together.
- You must configure the **match protocol msrpc** command for DPI.

Information About MSRPC ALG Support for Firewall and NAT

Application-Level Gateways

An application-level gateway (ALG), also known as an application-layer gateway, is an application that translates the IP address information inside the payload of an application packet. An ALG is used to interpret the application-layer protocol and perform firewall and Network Address Translation (NAT) actions. These actions can be one or more of the following depending on your configuration of the firewall and NAT:

- Allow client applications to use dynamic TCP or UDP ports to communicate with the server application.
- Recognize application-specific commands and offer granular security control over them.
- Synchronize multiple streams or sessions of data between two hosts that are exchanging data.
- Translate the network-layer address information that is available in the application payload.

The firewall opens a pinhole, and NAT performs translation service on any TCP or UDP traffic that does not carry the source and destination IP addresses in the application-layer data stream. Specific protocols or applications that embed IP address information require the support of an ALG.

MSRPC

MSRPC is a framework that developers use to publish a set of applications and services for servers and enterprises. RPC is an interprocess communication technique that allows the client and server software to communicate over the network. MSRPC is an application-layer protocol that is used by a wide array of Microsoft applications. MSRPC supports both connection-oriented (CO) and connectionless (CL) Distributed Computing Environment (DCE) RPC modes over a wide variety of transport protocols. All services of MSRPC establish an initial session that is referred to as the primary connection. A secondary session over a port range between 1024 to 65535 as the destination port is established by some services of MSRPC.

For MSRPC to work when firewall and NAT are enabled, in addition to inspecting MSRPC packets, the ALG is required to handle MSRPC specific issues like establishing dynamic firewall sessions and fixing the packet content after the NAT.

By applying MSRPC protocol inspection, most MSRPC services are supported, eliminating the need for Layer 7 policy filters.

MSRPC ALG on Firewall

After you configure the firewall to inspect the MSRPC protocol, the MSRPC ALG starts parsing MSRPC messages. The following table describes the types of Protocol Data Units (PDU) supported by the MSRPC ALG Support on Firewall and NAT feature:

Table 3 Supported PDU Types

PDU	Number	Type	Description
REQUEST	0	call	Initiates a call request.
RESPONSE	2	call	Responds to a call request.
FAULT	3	call	Indicates an RPC runtime, RPC stub, or RPC-specific exception.

PDU	Number	Type	Description
BIND	11	association	Initiates the presentation negotiation for the body data.
BIND_ACK	12	association	Accepts a bind request.
BIND_NAK	13	association	Rejects an association request.
ALTER_CONTEXT	14	association	Requests additional presentation negotiation for another interface and/or version, or to negotiate a new security context, or both.
ALTER_CONTEXT_RESP	15	association	Responds to the ALTER_CONTEXT PDU. Valid values are accept or deny.
SHUTDOWN	17	call	Requests a client to terminate the connection and free the related resources.
CO_CANCEL	18	call	Cancels or orphans a connection. This message is sent when a client encounters a cancel fault.
ORPHANED	19	call	Aborts a request that in progress and that has not been entirely transmitted yet, or aborts a (possibly lengthy) response that is in progress.

MSRPC ALG on NAT

When NAT receives an MSRPC packet, it invokes the MSRPC ALG that parses the packet payload and forms a token to translate any embedded IP addresses. This token is passed to NAT, which translates addresses or ports as per your NAT configuration. The translated addresses are then written back into the packet payload by the MSRPC ALG.

If you have configured both the firewall and NAT, NAT calls the ALG first.

MSRPC Stateful Parser

The MSRPC state machine or the parser is the brain of the MSRPC ALG. The MSRPC stateful parser keeps all stateful information within the firewall or NAT depending on which feature invokes the parser first. The parser provides DPI of MSRPC protocol packets. It checks for protocol conformance and detects out-of-sequence commands and malformed packets. As the packet is parsed, the state machine records various data and fills in the correct token information for NAT and firewall inspection.

How to Configure MSRPC ALG Support for Firewall and NAT

Note By default, MSRPC ALG is automatically enabled when NAT is enabled. There is no need to explicitly enable MSRPC ALG in the NAT-only configuration. You can use the **no ip nat service msrpc** command to disable MSRPC ALG on NAT.

- Configuring a Layer 4 MSRPC Class Map and Policy Map, page 70
- Configuring a Zone Pair and Attaching an MSRPC Policy Map, page 72

Configuring a Layer 4 MSRPC Class Map and Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map type inspect match-any** *class-map-name*
4. **match protocol** *protocol-name*
5. **exit**
6. **policy-map type inspect** *policy-map-name*
7. **class type inspect** *class-map-name*
8. **inspect**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map type inspect match-any** *class-map-name* **Example:** `Router(config)# class-map type inspect match-any msrpc-cmap`	Creates an inspect type class map for the traffic class and enters QoS class-map configuration mode.

	Command or Action	Purpose
Step 4	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol msrpc`	Configures the match criteria for a class map on the basis of a specified protocol. • Only Cisco IOS XE stateful packet inspection-supported protocols can be used as match criteria in inspect type class maps.
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Exits QoS class-map configuration mode and enters global configuration mode.
Step 6	**policy-map type inspect** *policy-map-name* **Example:** `Router(config)# policy-map type inspect msrpc-pmap`	Creates a Layer 3 or Layer 4 inspect type policy map and enters QoS policy-map configuration mode.

Step 7	**class type inspect** *class-map-name*
	Example:
	`Router(config-pmap)# class type inspect msrpc-class-map`
		Specifies the traffic (class) on which an action is to be performed and enters QoS policy-map class configuration mode.

Step 8	**inspect** **Example:** `Router(config-pmap-c)# inspect`	Enables Cisco IOS XE stateful packet inspection.

Step 9	**end**
	Example:
	`Router(config-pmap-c)# end`
		Exits QoS policy-map class configuration mode and enters privileged EXEC mode.

Configuring a Zone Pair and Attaching an MSRPC Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **zone security** *security-zone-name*
4. **exit**
5. **zone security** *security-zone-name*
6. **exit**
7. **zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** [*destination-zone*]]
8. **service-policy type inspect** *policy-map-name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Rotuer# configure terminal`	Enters global configuration mode.
Step 3	**zone security** *security-zone-name* **Example:** `Router(config)# zone security in-zone`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 4	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 5	**zone security** *security-zone-name* **Example:** `Router(config)# zone security out-zone`	Creates a security zone to which interfaces can be assigned and enters security zone configuration mode.
Step 6	**exit** **Example:** `Router(config-sec-zone)# exit`	Exits security zone configuration mode and enters global configuration mode.
Step 7	**zone-pair security** *zone-pair-name* [**source** *source-zone* **destination** [*destination-zone*]] **Example:** `Router(config)# zone-pair security in-out source in-zone destination out-zone`	Creates a zone pair and enters security zone pair configuration mode. **Note** To apply a policy, you must configure a zone pair.
Step 8	**service-policy type inspect** *policy-map-name* **Example:** `Router(config-sec-zone-pair)# service-policy type inspect msrpc-pmap`	Attaches a firewall policy map to the destination zone pair. **Note** If a policy is not configured between a pair of zones, traffic is dropped by default.

Step 9	**end**
	Example:
	`Router(config-sec-zone-pair)# end`
		Exits security zone pair configuration mode and enters privileged EXEC mode.

Configuration Examples for MSRPC ALG Support for Firewall and NAT

Example: Configuring a Layer 4 MSRPC Class Map and Policy Map

```
Router# configure terminal
Router(config)# class-map type inspect match-any msrpc-cmap
Router(config-cmap)# match protocol msrpc
Router(config-cmap)# exit
Router(config)# policy-map type inspect msrpc-pmap
Router(config-pmap)# class type inspect msrpc-cmap
Router(config-pmap-c)# inspect
Router(config-pmap-c)# end
```

Example: Configuring a Zone Pair and Attaching an MSRPC Policy Map

```
Router# configure terminal
Router(config)# zone security in-zone
Router(config-sec-zone)# exit
Router(config)# zone security out-zone
Router(config-sec-zone)# exit
Router(config)# zone-pair security in-out source in-zone destination out-zone
Router(config-sec-zone-pair)# service-policy type inspect msrpc-pmap
Router(config-sec-zone-pair)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
NAT commands	*Cisco IOS IP Addressing Services Command Reference*
Security commands	• *Cisco IOS Security Command Reference: Commands A to C* • *Cisco IOS Security Command Reference: Commands D to L* • *Cisco IOS Security Command Reference: Commands M to R* • *Cisco IOS Security Command Reference: Commands S to Z*
NAT ALGs	"Using Application-Level Gateways with NAT" module
ALG support	*NAT and Firewall ALG Support on Cisco ASR 1000 Series Routers*

Standards and RFCs

Standard/RFC	Title
No new or modified standards or RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MSRPC ALG Support for Firewall and NAT

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4 *Feature Information for MSRPC ALG Support for Firewall and NAT*

Feature Name	Releases	Feature Information
MSRPC ALG Support for Firewall and NAT	Cisco IOS XE Release 3.5S	The MSRPC ALG Support for Firewall and NAT feature provides support for the MSRPC ALG on the firewall and NAT. The MSRPC ALG provides deep packet inspection of the MSRPC protocol. The MSRPC ALG works in conjunction with a provisioning system to allow the network administrator to configure match filters that define match criteria that can be searched in an MSRPC packet. The following commands were introduced or modified: **ip nat service msrpc**, **match protocol msrpc**.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Configuring NAT for High Availability

This module contains procedures for configuring Network Address Translation (NAT) to support the increasing need for highly resilient IP networks. This network resiliency is required where application connectivity needs to continue unaffected by failures to links and routers at the NAT border.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring NAT for High Availability

To understand how High Availability (HA) is implemented on the Cisco ASR 1000 Series Aggregation Services Routers, see the "High Availability Overview" chapter in the *Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide*.

Information About Configuring NAT for High Availability

Static Mapping Support with HSRP for High Availability Feature Overview

When an Address Resolution Protocol (ARP) query is triggered for an address that is configured with NAT static mapping and owned by the router, NAT responds with the burned in MAC (BIA MAC) address on the interface to which the ARP is pointing. Two routers act as the Hot Standby Router Protocol (HSRP) active and standby. You must enable and configure the NAT inside interfaces of the active and standby routers to belong to a group.

Address Resolution with ARP

A device can have both a local address (which uniquely identifies the device on its local segment or LAN) and a network address (which identifies the network to which the device belongs). The local address is known as the data link address because it is contained in the data link layer of the packet header and is read by data-link devices (bridges and all device interfaces). The local address is also referred to as the MAC address, because the MAC sublayer within the data link layer processes addresses for the layer.

To communicate with a device on an Ethernet port your software must first determine the 48-bit MAC or local data link address of that device; for example, the Cisco IOS XE software first must determine the 48-bit MAC or local data link address of the Ethernet port for communication. The process of determining the local data link address from an IP address is called address resolution. The process of determining the IP address from a local data-link address is called reverse address resolution.

You can use Address Resolution Protocol (ARP), proxy ARP, and Probe (similar to ARP), and Reverse Address Resolution Protocol (RARP) for address resolution. ARP, proxy ARP, and RARP are defined in RFCs 826, 1027, and 903, respectively. Probe is a protocol developed by the Hewlett-Packard Company (HP) for use on IEEE-802.3 networks.

ARP is used to associate IP addresses with media or MAC addresses. Taking an IP address as input, ARP determines the associated media address. Once a media or MAC address is determined, the IP address or the media address association is stored in an ARP cache for rapid retrieval. The IP datagram is encapsulated in a link layer frame and sent over the network. Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet is specified by the Subnetwork Access Protocol (SNAP).

How to Configure NAT for High Availability

- Configuring NAT Static Mapping Support for HSRP, page 78

Configuring NAT Static Mapping Support for HSRP

Both of the following tasks are required and must be performed on both the active and standby routers to configure NAT static mapping support for HSRP:

- Restrictions for Configuring Static Mapping Support for HSRP, page 79
- Benefits of Configuring Static Mapping Support for HSRP, page 79
- Enabling HSRP on the NAT Interface, page 79
- Enabling Static NAT in an HSRP Environment, page 81

Restrictions for Configuring Static Mapping Support for HSRP

- Static NAT mappings must be mirrored on two or more HSRP routers, because the NAT state will not be exchanged between routers running NAT in an HSRP group.
- If you configure both HSRP routers with the same static NAT and the **hsrp** keyword to link the routers to the same HSRP group is not configured, the behavior of the routers will be unpredictable.

Benefits of Configuring Static Mapping Support for HSRP

- When you configure static mapping for HSRP and the HSRP routers have an identical NAT configuration for redundancy, the failover happens without timing out and repopulating upstream ARP caches.
- Static mapping support for HSRP allows an HSRP active router to respond to an incoming ARP request for a router that is configured with a NAT address.

Enabling HSRP on the NAT Interface

Perform this task on both active and standby routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **no ip redirects**
6. **ip nat** {**inside** | **outside**}
7. **standby** [*group-number*] **priority** *priority*
8. **standby** [*group-number*] **preempt**
9. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
10. **standby** [*group-number*] **name** [*group-name*]
11. **standby** [*group-number*] **track** *interface number*
12. **end**
13. **show standby**
14. **show ip nat translations** [**verbose**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.1.27 255.255.255.0`	Sets the primary IP address on the interface.
Step 5	**no ip redirects** **Example:** `Router(config-if)# no ip redirects`	Disables the sending of redirect messages

Step 6	**ip nat** {**inside** | **outside**}
	Example:
	`Router(config-if)# ip nat inside`
		Marks the interface as connected to the inside.

Step 7	**standby** [*group-number*] **priority** *priority* **Example:** `Router(config-if)# standby 10 priority 105`	Configures HSRP priority.
Step 8	**standby** [*group-number*] **preempt** **Example:** `Router(config-if)# standby 10 preempt`	Configures HSRP preemption.
Step 9	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Router(config-if)# standby 10 ip 192.168.5.30`	Enables the HSRP protocol.

	Command or Action	Purpose
Step 10	**standby** [*group-number*] **name** [*group-name*] **Example:** `Router(config-if)# standby 10 name HSRP1`	Sets the HSRP group name.
Step 11	**standby** [*group-number*] **track** *interface number* **Example:** `Router(config-if)# standby 10 track GigabitEthernet0/0/1`	Configures HSRP to track an object and to change the hot standby priority on the basis of the state of the object.

Step 12	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and enters privileged EXEC mode.

Step 13	**show standby** **Example:** `Router# show standby`	(Optional) Displays HSRP information
Step 14	**show ip nat translations** [**verbose**] **Example:** `Router# show ip nat translations verbose`	(Optional) Displays active NAT translations.

Enabling Static NAT in an HSRP Environment

To enable static mapping support with HRSP for high availability, perform this task on both the active and standby routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *local-ip global-ip* **redundancy** *group-name*
4. **ip classless**
5. **ip route** *prefix mask interface-type interface-number*
6. **no ip http server**
7. **end**
8. **show ip nat translations** [**verbose**]

DETAILED STEPS

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

Step 3	**ip nat inside source static** *local-ip global-ip* **redundancy** *group-name*
	Example:
	`Router(config)# ip nat inside source static`
	`192.168.5.33 10.10.10.5 redundancy HSRP1`
		Enables the router to respond to ARP queries using BIA MAC, if HSRP is configured on the NAT inside interface.

Step 4	**ip classless** **Example:** `Router(config)# ip classless`	Enables a router to forward packets that are destined for a subnet of a network that has no network default route, to the best supernet route possible.

Step 5	**ip route** *prefix mask interface-type interface-number*
	Example:
	`Router(config)# ip route 10.10.10.0 255.255.255.0`
	`GigabitEthernet0/0/0`
		Establishes static routes.

Step 6	**no ip http server** **Example:** `Router(config)# no ip http server`	Enables the HTTP server on your IP system.

Step 7	**end**
	Example:
	`Router(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Step 8	**show ip nat translations** [**verbose**] **Example:** `Router# show ip nat translations verbose`	(Optional) Displays active NAT translations.

Configuration Examples for NAT for High Availability

Configuring Static NAT in an HSRP Environment Examples

The following example shows support for NAT with a static configuration in an HSRP environment. Two routers are acting as HSRP active and standby, and the NAT inside interfaces are HSRP enabled and configured to belong to the group HSRP1.

Active Router Configuration

```
interface GigabitEthernet 0/1/1
 ip address 192.168.5.54 255.255.255.255.0
 no ip redirects
 ip nat inside
 standby 10 priority 105
 standby 10 preempt
 standby 10 ip 192.168.5.30
 standby 10 name HSRP1
 standby 10 track GigabitEthernet0/0/0
!
 ip default-gateway 10.0.18.126
 ip nat inside source static 192.168.5.33 10.10.10.5 redundancy HSRP1
 ip classless
 ip route 10.10.10.0 255.255.255.0 GigabitEthernet0/0/0
 ip route 172.22.33.0 255.255.255.0 GigabitEthernet0/0/0
 no ip http server
```

Standby Router Configuration

```
interface GigabitEthernet 0/1/1
 ip address 192.168.5.56 255.255.255.255.0
 no ip redirects
 ip nat inside
 standby 10 priority 100
 standby 10 preempt
 standby 10 ip 192.168.5.30
 standby 10 name HSRP1
 standby 10 track GigabitEthernet0/0/1
!
 ip default-gateway 10.0.18.126
 ip nat inside source static 192.168.5.33 3.3.3.5 redundancy HSRP1
 ip classless
 ip route 10.0.32.231 255.255.255 GigabitEthernet0/0/1
 ip route 10.10.10.0 255.255.255.0 GigabitEthernet0/0/1
 no ip http server
```

Additional References

The following sections provide references related to NAT for high availability.

Related Documents

Related Topic	Document Title
High Availability on the Cisco ASR 1000 Series Aggregation Services Routers	"High Availability Overview" chapter in the *Cisco ASR Series 1000 Aggregation Services Routers Software Configuration Guide*
Cisco IOS XE ISSU NAT	"Cisco IOS XE In Service Software Upgrade Process" module
NAT configuration tasks	"Configuring NAT for IP Address Conservation" module
NAT commands: complete command syntax, command mode, command history, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
IP addressing configuration tasks and concepts.	*Cisco IOS XE IP Addressing Services Configuration Guide*

Standards

Standards	Title
None	

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 826	*Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware*
RFC 903	*Reverse Address Resolution Protocol*
RFC 1027	*Using ARP to implement transparent subnet gateways*

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, such as 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/techsupport

Feature Information for Configuring NAT for High Availability

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5 Feature Information for Configuring NAT for High Availability

Feature Name	Releases	Feature Configuration Information
NAT--Static Mapping Support with HSRP for High Availability	Cisco IOS XE Release 2.1	Static mapping support for HSRP allows the option of having only the HSRP active router respond to an incoming ARP for a router configured with a NAT address.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Integrating NAT with MPLS VPNs

The NAT Integration with MPLS VPNs feature allows multiple Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) to be configured on a single device to work together. NAT can differentiate which MPLS VPN it receives IP traffic from even if the MPLS VPNs are all using the same IP addressing scheme. This enhancement enables multiple MPLS VPN customers to share services while ensuring that each MPLS VPN is completely separate from the other.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Integrating NAT with MPLS VPNs

- Before performing the tasks in this module, you should be familiar with the concepts described in the “Configuring NAT for IP Address Conservation” module.
- All access lists required for use with the tasks in this module should be configured prior to beginning the configuration task. For information about how to configure an access list, see the “IP Access List Sequence Numbering” document at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s14/fsaclseq.htm

Note If you specify an access list to use with a NAT command, NAT does not support the commonly used **permit ip any any** command in the access list.

Restrictions for Integrating NAT with MPLS VPNs

This feature was introduced in Cisco IOS XE 2.5. For a list of restrictions, see the Cisco IOS XE 2 Release Notes .

Information About Integrating NAT with MPLS VPNs

Benefits of NAT Integration with MPLS VPNs

For MPLS service providers to provide value-added services such as Internet connectivity, domain name servers (DNS), and VoIP service to their customers, their customers' IP addresses be unique when reaching the services. Because MPLS VPN allows customers to use overlapped IP addresses in their networks, NAT must be implemented to make the services possible.

Implementation Options for Integrating NAT with MPLS VPNs

There are two approaches to implementing NAT in the MPLS VPN network. NAT can be implemented on the customer edge (CE) router, which is already supported by NAT, or it can be implemented on a provider edge (PE) router. The NAT Integration with MPLS VPNs feature enables the implementation of NAT on a PE router in an MPLS cloud.

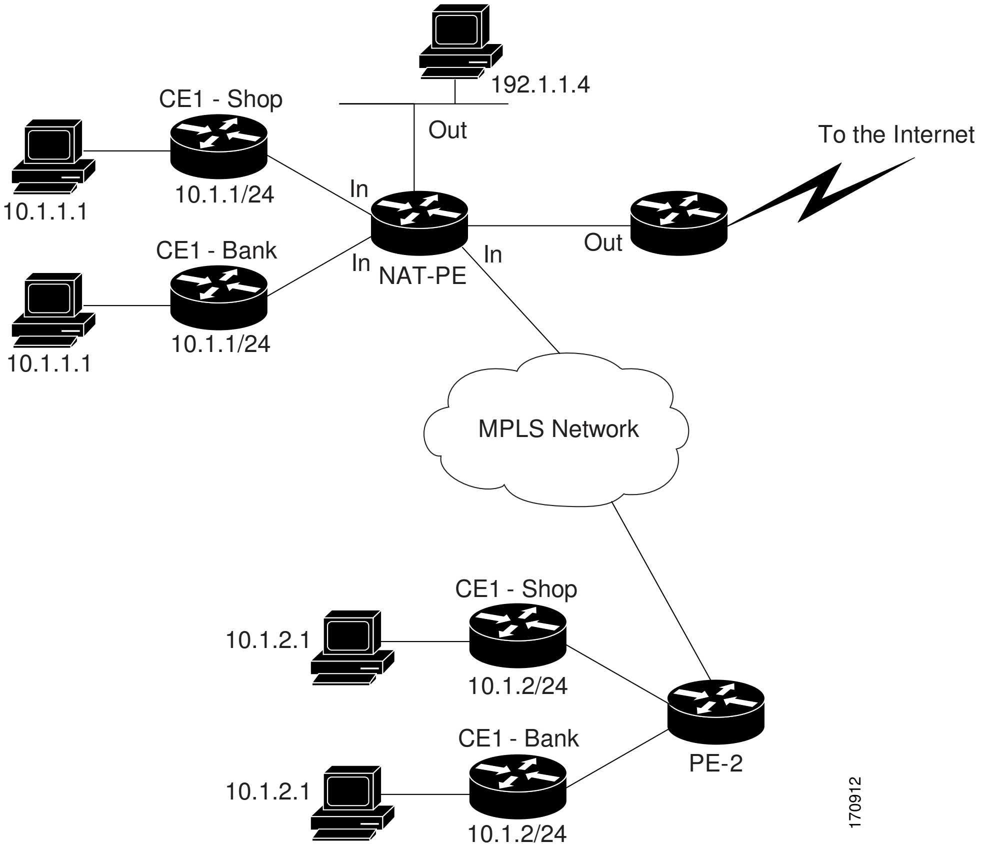
Scenarios for Implementing NAT on the PE Router

NAT can be implemented on the PE router in the following scenarios:

- Service point--Shared access can be from a generic interface or from a VPN interface.
- NAT point--NAT can be configured on the PE router that is directly connected to the shared access gateway, or on the PE router that is not directly connected to the shared access gateway.
- NAT interface--The shared access gateway interface most often is configured as the outside interface of NAT. The inside interface of NAT can be either the PE-CE interface of a VPN or the interface to the MPLS backbone, or both. The shared access gateway interface can also be configured as the inside interface.
- Routing type--Common service can be Internet connectivity or a common server. For Internet connectivity, a default route should be propagated to all the VPN customers that use the service. For common server access, a static or dynamically learned route should be propagated to the VPN customers.
- NAT configuration--NAT can have different configurations: static, dynamic, pool/interface overloading, and route map.

The figure below shows a typical NAT integration with MPLS VPNs. The PE router is connected to the Internet and centralized mail service is employed to do the address translation.

Figure 5 Typical NAT Integration with MPLS VPNs



How to Integrate NAT with MPLS VPNs

Perform one or more of the following tasks depending on the type of translation you want to configure for your network:

- Configuring Inside Dynamic NAT with MPLS VPNs, page 89
- Configuring Inside Static NAT with MPLS VPNs, page 91
- Configuring Outside Dynamic NAT with MPLS VPNs, page 93
- Configuring Outside Static NAT with MPLS VPNs, page 94

Configuring Inside Dynamic NAT with MPLS VPNs

Perform this task to configure your NAT PE router for dynamic translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* **netmask** *netmask*
4. **ip nat** [**inside** | **outside**] **source** [**list** {*access-list-number*| *access-list-name*} | **route-map** *name*] [**interface** *type number* | **pool** *pool-name*] **vrf** *vrf-name*[**overload**]
5. Repeat Step 4 for each VPN being configured
6. **ip route vrf** *vrf-name prefix mask interface-type interface-number next-hop-address*
7. Repeat Step 6 for each VPN being configured.
8. **exit**
9. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Router(config)# ip nat pool inside 10.2.2.10 10.2.2.10 netmask 255.0.0.0`	Defines a pool of IP addresses for NAT.
Step 4	**ip nat** [**inside** | **outside**] **source** [**list** {*access-list-number*| *access-list-name*} | **route-map** *name*] [**interface** *type number* | **pool** *pool-name*] **vrf** *vrf-name*[**overload**] **Example:** `Router(config)# ip nat inside source list 1 pool mypool vrf shop overload`	Allows NAT to be configured on a particular VPN. • For a list of restrictions, see the Cisco IOS XE 2 Release Notes .
Step 5	Repeat Step 4 for each VPN being configured	Allows NAT to be configured on additional VPNs.

	Command or Action	Purpose

Step 6	**ip route vrf** *vrf-name prefix mask interface-type interface-number next-hop-address*
	Example:
	`Router(config)#`
	`ip route vrf shop 0.0.0.0 0.0.0.0 fastethernet 0`
	`192.168.88.2`
		Allows the route to be shared by customers using the specified VPN.

Step 7	Repeat Step 6 for each VPN being configured.	Allows the route to be shared by customers using additional specified VPNs.
Step 8	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 9	**show ip nat translations vrf** *vrf-name* **Example:** `Router# show ip nat translations vrf shop`	(Optional) Displays the settings used by VRF table translations.

Configuring Inside Static NAT with MPLS VPNs

Perform this task to configure your NAT PE router for static translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** {**esp** *local-ip* **interface** *type number* | *local-ip global-ip*} [**extendable** | **mapping-id** *map-id*| **no-alias** | **no-payload** | **redundancy** *group-name* | **route-map** | **vrf** *name*]
4. Repeat Step 3 for each VPN being configured.
5. **ip route vrf** *vrf-name* **prefix** *prefix mask next-hop-address* **global**
6. Repeat Step 5 for each VPN being configured.
7. **exit**
8. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**ip nat inside source static** {**esp** *local-ip* **interface** *type number* | *local-ip global-ip*} [**extendable** | **mapping-id** *map-id*| **no-alias** | **no-payload** | **redundancy** *group-name* | **route-map** | **vrf** *name*]
	Example:
	`Router(config)#`
	`ip nat inside source static 192.168.121.113 10.2.2.1 vrf`
	`shop`
		Enables inside static translation on the specified VRF.
		• For a list of restrictions, see the Cisco IOS XE 2 Release Notes .

Step 4	Repeat Step 3 for each VPN being configured.	Enables inside static translation on additional VRFs.

Step 5	**ip route vrf** *vrf-name* **prefix** *prefix mask next-hop-address* **global**
	Example:
	`Router(config)#`
	`ip route vrf shop prefix 0.0.0.0 0.0.0.0 192.168.88.2`
	`global`
		Allows the route to be shared by customers using the specified VPN.

Step 6	Repeat Step 5 for each VPN being configured.	Allows the route to be shared by customers using additional specified VPNs.
Step 7	**exit** **Example:** `Router config)# exit`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 8	**show ip nat translations vrf** *vrf-name* **Example:** `Router# show ip nat translations vrf shop`	(Optional) Displays the settings used by VRF translations.

Configuring Outside Dynamic NAT with MPLS VPNs

Perform this step to configure your NAT PE router for dynamic outside translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name global-ip local-ip* **netmask** *netmask*
4. **ip nat inside source static** *local-ip global-ip* **vrf** *vrf-name*
5. Repeat Step 4 for each VRF being configured.
6. **ip nat outside source static** *global-ip local-ip* **vrf** *vrf-name*
7. **exit**
8. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**ip nat pool** *name global-ip local-ip* **netmask** *netmask*
	Example:
	`Router(config)#`
	`ip nat pool out_pool 10.4.4.1 10.4.4.254 netmask`
	`255.0.0.00`
		Allows the configured VRF to be associated with the NAT translation rule.

	Command or Action	Purpose

Step 4	**ip nat inside source static** *local-ip global-ip* **vrf** *vrf-name*
	Example:
	`Router(config)#`
	`ip nat inside source static 192.168.121.113 10.2.2.1`
	`vrf shop`
		Allows the route to be shared by customers using the specified VPN.
		• For a list of restrictions, see the Cisco IOS XE 2 Release Notes .

Step 5	Repeat Step 4 for each VRF being configured.	Allows the route to be shared by customers using additional specified VPNs.

Step 6	**ip nat outside source static** *global-ip local-ip* **vrf** *vrf-name*
	Example:
	`Router(config)# i`
	`p nat outside source static 192.168.88.2 10.4.4.1 vrf`
	`shop`
		Enables NAT translation of the outside source address.

Step 7	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 8	**show ip nat translations vrf** *vrf-name* **Example:** `Router# show ip nat translations vrf shop`	(Optional) Displays the settings used by VRF translations.

Configuring Outside Static NAT with MPLS VPNs

Perform this task to configure your NAT PE router for static outside translations to integrate with MPLS VPNs.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **ip nat pool** *name global-ip local-ip* **netmask** *netmask*
4. Repeat Step 3 for each pool being configured.
5. **ip nat inside source list** *access-list-number* **pool** *pool-name* **vrf** *vrf-name*
6. Repeat Step 5 for each pool being configured.
7. **ip nat outside source static** *global-ip local-ip* **vrf** *vrf-name*
8. Repeat Step 7 for all VPNs being configured.
9. **exit**
10. **show ip nat translations vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.

Step 2	**configure** {**terminal** | **memory** | **network**}
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**ip nat pool** *name global-ip local-ip* **netmask** *netmask*
	Example:
	`Router(config)# i`
	`p nat pool in_pool 10.2.1.1 10.2.1.254 netmask`
	`255.0.0.0`
		Allows the configured VRF to be associated with a NAT translation rule.

Step 4	Repeat Step 3 for each pool being configured.	Allows the configured VRF to be associated with additional NAT translation rules.

Step 5	**ip nat inside source list** *access-list-number* **pool** *pool-name* **vrf** *vrf-name*
	Example:
	`Router(config)#`
	`ip nat inside source list 1 pool in_pool vrf shop`
		Allows the route to be shared by several customers.
		• For a list of restrictions, see the Cisco IOS XE 2 Release Notes .

	Command or Action	Purpose
Step 6	Repeat Step 5 for each pool being configured.	Defines the access list.

Step 7	**ip nat outside source static** *global-ip local-ip* **vrf** *vrf-name*
	Example:
	`Router(config)#`
	`ip nat outside source static 192.168.88.2 10.4.4.1 vrf shop`
		Allows the route to be shared by customers using the specified VPN.

Step 8	Repeat Step 7 for all VPNs being configured.	Allows the route to be shared by customers using additional specified VPNs.
Step 9	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 10	**show ip nat translations vrf** *vrf-name* **Example:** `Router# show ip nat translations vrf shop`	(Optional) Displays the settings used by VRF translations.

Configuration Examples for Integrating NAT with MPLS VPNs

- Configuring Inside Dynamic NAT with MPLS VPNs Example, page 96
- Configuring Inside Static NAT with MPLS VPNs Example, page 97
- Configuring Outside Dynamic NAT with MPLS VPNs Example, page 97
- Configuring Outside Static NAT with MPLS VPNs Example, page 97

Configuring Inside Dynamic NAT with MPLS VPNs Example

The following example shows how to configure inside Dynamic NAT with MPLS VPNs:

```
!
ip nat pool inside 10.2.2.10 10.2.2.10 netmask 255.0.0.0
ip nat inside source list 1 pool inside vrf bank overload
ip nat inside source list 1 pool inside vrf park overload
ip nat inside source list 1 pool inside vrf shop overload
!
ip route vrf shop 0.0.0.0 0.0.0.0 fastethernet1/3 192.168.88.2
ip route vrf bank 0.0.0.0 0.0.0.0 fastethernet1/3 192.168.88.2
ip route vrf park 0.0.0.0 0.0.0.0 fastethernet1/3 192.168.88.2
!
access-list 1 permit 192.168.0.0 0.0.255.255
```

Configuring Inside Static NAT with MPLS VPNs Example

The following example shows how to configure inside static NAT with MPLS VPNs:

```
!
ip nat inside source static 192.168.121.113 10.2.2.1 vrf shop
ip nat inside source static 192.168.122.49 10.2.2.2 vrf shop
ip nat inside source static 192.168.121.113 10.2.2.3 vrf bank
ip nat inside source static 192.168.22.49 10.2.2.4 vrf bank
ip nat inside source static 192.168.121.113 10.2.2.5 vrf park
ip nat inside source static 192.168.22.49 10.2.2.6 vrf park
ip nat inside source static 192.168.11.1 10.2.2.11 vrf shop
ip nat inside source static 192.168.11.3 10.2.2.12 vrf shop
ip nat inside source static 140.48.5.20 10.2.2.13 vrf shop
!
ip route 10.2.2.1 255.255.255.255 fastethernet1/0 192.168.121.113
ip route 10.2.2.2 255.255.255.255 fastethernet1/0 192.168.121.113
ip route 10.2.2.3 255.255.255.255 Serial2/1.1 192.168.121.113
ip route 10.2.2.4 255.255.255.255 Serial2/1.1 192.168.121.113
ip route 10.2.2.5 255.255.255.255 fastethernet0/0 192.168.121.113
ip route 10.2.2.6 255.255.255.255 fastethernet0/0 192.168.121.113
ip route 10.2.2.11 255.255.255.255 fastethernet1/0 192.168.121.113
ip route 10.2.2.12 255.255.255.255 fastethernet1/0 192.168.121.113
ip route 10.2.2.13 255.255.255.255 fastethernet1/0 192.168.121.113
```

Configuring Outside Dynamic NAT with MPLS VPNs Example

The following example shows how to configure outside dynamic NAT with MPLS VPNs:

```
!
ip nat pool outside 10.4.4.1 10.4.4.254 netmask 255.0.0.0
ip nat inside source static 192.168.121.113 10.2.2.1 vrf shop
ip nat inside source static 192.168.122.49 10.2.2.2 vrf shop
ip nat inside source static 192.168.121.113 10.2.2.3 vrf bank
ip nat inside source static 192.168.22.49 10.2.2.4 vrf bank
ip nat inside source static 192.168.121.113 10.2.2.5 vrf park
ip nat inside source static 192.168.22.49 10.2.2.6 vrf park
ip nat outside source list 1 pool outside
!
```

Configuring Outside Static NAT with MPLS VPNs Example

The following example shows how to configure outside static NAT with MPLS VPNs:

```
!
ip default-gateway 10.1.15.1
ip nat pool inside1 10.2.1.1 10.2.1.254 netmask 255.0.0.0
ip nat pool inside2 10.2.2.1 10.2.2.254 netmask 255.0.0.0
ip nat pool inside3 10.2.3.1 10.2.3.254 netmask 255.0.0.0
ip nat inside source list 1 pool inside2 vrf bank
ip nat inside source list 1 pool inside3 vrf park
ip nat inside source list 1 pool inside1 vrf shop
ip nat outside source static 192.168.88.2 10.4.4.1 vrf bank
ip nat outside source static 10.68.58.1 10.4.4.2 vrf park
ip nat outside source static 192.168.88.1 10.4.4.3 vrf shop
ip classless
ip route 172.16.10.0 255.255.255.0 fastethernet 1/0 192.168.121.113
ip route 172.16.11.0 255.255.255.0 Serial 2/1.1 192.168.121.113
ip route 172.16.12.0 255.255.255.0 fastethernet 0/0 192.168.121.113
ip route vrf shop 0.0.0.0 0.0.0.0 192.168.88.2 global
ip route vrf bank 0.0.0.0 0.0.0.0 192.168.88.2 global
ip route vrf park 0.0.0.0 0.0.0.0 192.168.88.2 global
no ip http server
!
access-list 1 permit 192.168.0.0 0.0.255.255
```

Where to Go Next

For more information about configuring IP applications and services, see the *IP SLAs Configuration Guide Cisco IOS XE Release 3S*.

Additional References

The following sections provide references related to NAT.

Related Documents

Related Topic	Document Title
NAT commands: complete command syntax, command mode, command history, defaults, usage guidelines and examples	*Cisco IOS IP Addressing Services Command Reference*
Configuring an access list	IP Access List Sequence Numbering
NAT high availability	“Configuring NAT for High Availability” module
Application-level gateways	“Using Application Level Gateways with NAT"
Maintain and monitor NAT	“Monitoring and Maintaining NAT” module
IP address conservation	“Configuring NAT for IP Address Conservation” module

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs[1]	Title
RFC 2547	BGP/MPLS VPNs

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, such as 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/techsupport

Feature Information for Integrating NAT with MPLS VPNs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6 Feature Information for Integrating NAT with MPLS VPNs

Feature Name	Releases	Feature Configuration Information
NAT Integration with MPLS VPNs feature	Cisco IOS XE Release 2.5	This feature allows multiple Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) to be configured on a single device to work together.

[1] Not all supported RFCs are listed.

Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Monitoring and Maintaining NAT

The Monitoring and Maintaining NAT feature enables the monitoring of Network Address Translation (NAT) by using translation information and statistics displays. It enables the logging of NAT translation to log and track system error messages and exceptions. The Monitoring and Maintaining NAT feature helps maintain NAT by clearing NAT translations before the timeout is expired.

This modules the Monitoring and Maintaining NAT feature.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Monitoring and Maintaining NAT

Before performing the tasks in this module, you must be familiar with the concepts described in the “Configuring NAT for IP Address Conservation” module and have NAT configured in your network.

Restrictions for Maintaining and Monitoring NAT

Syslog for Network Address Translation (NAT) is not supported.

Information About Monitoring and Maintaining NAT

NAT Display Contents

The two basic types of IP NAT translation information are described in the following sections:

Translation Entry Information

Translation entry information includes the following:

- Protocol of the port identifying the address.
- Legitimate IP address that represents one or more inside local IP addresses to the outside world.
- IP address assigned to a host on the inside network; probably not a legitimate address assigned by the Network Information Center (NIC) or the service provider.
- IP address of an outside host as it appears to the inside network; probably not a legitimate address assigned by the NIC or the service provider.
- IP address assigned to a host on the outside network by its owner.
- Time since the entry was created (in hours:minutes:seconds).
- Time since the entry was last used (in hours:minutes:seconds).
- Flags indicating the type of translation. Possible flags are as follows:
    - destination—Rotary translation.
    - extended—Extended translation.
    - outside—Outside translation.
    - static—Static translation.
    - timing out—Translation will be aged out or removed soon because of a TCP finish (FIN) or reset (RST) flag.

Statistical Information

Statistical information includes the following:

- Total number of translations that are active in the system. This number is incremented each time a translation is created and is decremented each time a translation is cleared or times out.
- List of interfaces that are marked as outside by using the **ip nat outside** command.
- List of interfaces that are marked as inside by using the **ip nat inside** command.
- Number of times the software does a translation table lookup and finds an entry.
- Number of times the software does a translation table lookup, fails to find an entry, and must try to create one.
- Cumulative count of translations that have expired since the device was booted.
- Information about dynamic mappings.
- Information about inside source translations.

- Access list number that is used for translations.
- Name of the address pool.
- Number of translations that use this address pool.
- IP network mask that is used by the address pool.
- Starting IP address in the address pool range.
- Ending IP address in the address pool range.
- Type of address pool. Possible types are generic or rotary.
- Number of addresses in the address pool that are available for translation.
- Number of addresses that are used for translation.
- Number of failed allocations from the pool.

Network Address Translation (NAT) does not support access control lists (ACLs) with the log option. Instead, you can use one of the following options:

- A physical interface or VLAN with the logging option
- NetFlow.

NAT-Forced Clear of Dynamic NAT Half-Entries

The NAT-Forced Clear of Dynamic NAT Half-Entries feature filters the display of the translation table by specifying an inside or outside address. This feature introduces the **clear ip nat translation forced** command that forcefully clears active dynamic Network Address Translation (NAT) half-entries that have child translations.

How to Monitor and Maintain NAT

Displaying NAT Translation Information

SUMMARY STEPS

1. **enable**
2. **show ip nat translations** [**verbose**]
3. **show ip nat statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations`	(Optional) Displays active NAT translations.
Step 3	**show ip nat statistics** **Example:** `Device# show ip nat statistics`	(Optional) Displays active NAT translation statistics.

Example:

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

```
Device# show ip nat translations

Pro Inside global         Inside local       Outside local        Outside global
tcp 192.168.1.1:514      192.168.2.3:53     192.168.2.22:256     192.168.2.22:256
tcp 192.168.1.1:513      192.168.2.2:53     192.168.2.22:256     192.168.2.22:256
tcp 192.168.1.1:512      192.168.2.4:53     192.168.2.22:256     192.168.2.22:256
Total number of translations: 3
```

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

```
Device# show ip nat translations verbose

Pro Inside global        Inside local       Outside local      Outside global
tcp 192.168.1.1:514      192.168.2.3:53     192.168.2.22:256     192.168.2.22:256
         create 04/09/11 10:51:48, use 04/09/11 10:52:31, timeout: 00:01:00
         Map-Id(In):1, Mac-Address: 0000.0000.0000 Input-IDB: GigabitEthernet0/3/1
          entry-id: 0x8ef80350, use_count:1
tcp 192.168.1.1:513      192.168.2.2:53     192.168.2.22:256     192.168.2.22:256
         create 04/09/11 10:51:48, use 04/09/11 10:52:31, timeout: 00:01:00
         Map-Id(In):1, Mac-Address: 0000.0000.0000 Input-IDB: GigabitEthernet0/3/1
          entry-id: 0x8ef801b0, use_count:1
tcp 192.168.1.1:512      192.168.2.4:53     192.168.2.22:256     192.168.2.22:256
         create 04/09/11 10:51:48, use 04/09/11 10:52:31, timeout: 00:01:00
         Map-Id(In):1, Mac-Address: 0000.0000.0000 Input-IDB: GigabitEthernet0/3/1
          entry-id: 0x8ef80280, use_count:1
Total number of translations: 3
```

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

```
Device# show ip nat statistics

Total active translations: 3 (0 static, 3 dynamic; 3 extended)
Outside interfaces:
GigabitEthernet0/3/0
Inside interfaces:
GigabitEthernet0/3/1
Hits: 3228980 Misses: 3
CEF Translated packets: 0, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool pool1 refcount 3
  pool pool1: netmask 255.255.255.0
  start 198.168.1.1 end 198.168.254.254
  type generic, total addresses 254, allocated 0 (0%), misses 0
  longest chain in pool: pool1's addr-hash: 0, average len 0,chains 0/256
  Pool stats drop: 0 Mapping stats drop: 0
  Port block alloc fail: 0
  IP alias add fail: 0
  Limit entry add fail: 0
```

- Examples, page 105

Examples

Displaying NAT Translations

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

```
Device# show ip nat translations

Pro Inside global         Inside local       Outside local        Outside global
tcp 192.168.1.1:514      192.168.2.3:53     192.168.2.22:256     192.168.2.22:256
tcp 192.168.1.1:513      192.168.2.2:53     192.168.2.22:256     192.168.2.22:256
tcp 192.168.1.1:512      192.168.2.4:53     192.168.2.22:256     192.168.2.22:256
Total number of translations: 3
```

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

```
Device# show ip nat translations verbose

Pro Inside global        Inside local       Outside local      Outside global
tcp 192.168.1.1:514      192.168.2.3:53     192.168.2.22:256     192.168.2.22:256
         create 04/09/11 10:51:48, use 04/09/11 10:52:31, timeout: 00:01:00
         Map-Id(In):1, Mac-Address: 0000.0000.0000 Input-IDB: GigabitEthernet0/3/1
          entry-id: 0x8ef80350, use_count:1
tcp 192.168.1.1:513      192.168.2.2:53     192.168.2.22:256     192.168.2.22:256
         create 04/09/11 10:51:48, use 04/09/11 10:52:31, timeout: 00:01:00
         Map-Id(In):1, Mac-Address: 0000.0000.0000 Input-IDB: GigabitEthernet0/3/1
          entry-id: 0x8ef801b0, use_count:1
tcp 192.168.1.1:512      192.168.2.4:53     192.168.2.22:256     192.168.2.22:256
         create 04/09/11 10:51:48, use 04/09/11 10:52:31, timeout: 00:01:00
         Map-Id(In):1, Mac-Address: 0000.0000.0000 Input-IDB: GigabitEthernet0/3/1
          entry-id: 0x8ef80280, use_count:1
Total number of translations: 3
```

Displaying NAT Statistics

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

```
Device# show ip nat statistics

Total active translations: 3 (0 static, 3 dynamic; 3 extended)
Outside interfaces:
GigabitEthernet0/3/0
Inside interfaces:
GigabitEthernet0/3/1
Hits: 3228980 Misses: 3
CEF Translated packets: 0, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool pool1 refcount 3
  pool pool1: netmask 255.255.255.0
  start 198.168.1.1 end 198.168.254.254
  type generic, total addresses 254, allocated 0 (0%), misses 0
  longest chain in pool: pool1's addr-hash: 0, average len 0,chains 0/256
  Pool stats drop: 0 Mapping stats drop: 0
  Port block alloc fail: 0
  IP alias add fail: 0
  Limit entry add fail: 0
```

Clearing NAT Entries Before the Timeout

By default, dynamic address translations time out from the NAT translation table. However, you can clear the translation entries before the default timeout. Perform this task to clear the translation entries before the timeout.

SUMMARY STEPS

1. **enable**
2. **clear ip nat translation inside** *global-ip local-ip*
3. **clear ip nat translation outside** *global-ip local-ip*
4. **clear ip nat translation udp inside** *global-ip global-port local-ip local-port* **outside** *local-ip local-port global-ip global-port*
5. **clear ip nat translation** {* | **forced** | [**inside** *global-ip local-ip*] [**outside** *local-ip global-ip*]}
6. **clear ip nat translation inside** *global-ip local-ip* [**forced**]
7. **clear ip nat translation outside** *local-ip global-ip* [**forced**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip nat translation inside** *global-ip local-ip* **Example:** `Device# clear ip nat translation inside 192.168.2.209 192.168.2.95`	(Optional) Clears a single dynamic half-entry that contains an inside translation, or both inside and outside translation that is created in a dynamic configuration. • A dynamic half-entry is cleared only if the entry does not have any child translations.
Step 3	**clear ip nat translation outside** *global-ip local-ip* **Example:** `Device# clear ip nat translation outside 192.168.2.100 192.168.2.80`	(Optional) Clears a single dynamic half-entry that contains an outside translation that is created in a dynamic configuration. • A dynamic half-entry is cleared only if the entry does not have any child translations.
Step 4	**clear ip nat translation udp inside** *global-ip global-port local-ip local-port* **outside** *local-ip local-port global-ip global-port* **Example:** `Device# clear ip nat translation udp inside 192.168.2.209 1220 192.168.2.195 1220 outside 192.168.2.13 53 192.168.2.132 53`	(Optional) Clears a UDP translation entry.

	Command or Action	Purpose
Step 5	**clear ip nat translation** {* | **forced** | [**inside** *global-ip local-ip*] [**outside** *local-ip global-ip*]} **Example:** `Device# clear ip nat translation *`	(Optional) Clears all dynamic translations (by using the ***** or the **forced** keyword), a single dynamic half-entry that contains an inside translation, or a single dynamic half-entry that contains an outside translation. • A single dynamic half-entry is cleared only if the entry does not have any child translations.
Step 6	**clear ip nat translation inside** *global-ip local-ip* [**forced**] **Example:** `Device# clear ip nat translation inside 192.168.2.209 192.168.2.95 forced`	(Optional) Forcefully clears a single dynamic half-entry along with its child translations that contains an inside translation that is created in a dynamic configuration, with or without its corresponding outside translation.
Step 7	**clear ip nat translation outside** *local-ip global-ip* [**forced**] **Example:** `Device# clear ip nat translation outside 192.168.2.100 192.168.2.101 forced`	(Optional) Forcefully clears a single dynamic half-entry along with its child translations that contains an outside translation that is created in a dynamic configuration.

Configuration Examples for Monitoring and Maintaining NAT

Example: Clearing NAT Entries Before the Timeout

The following sample output from the **show ip nat translations** command displays the NAT entries before and after the UDP entry is cleared:

```
Device# show ip nat translations

Pro Inside global          Inside local       Outside local      Outside global
tcp 192.168.2.20:1220     192.168.2.95:1220   192.168.2.22:53    192.168.2.20:53
tcp 192.168.2.20:11012    192.168.2.209:11012 171.69.1.220:23    192.168.2.20:23
udp 192.168.2.20:1067     192.168.2.20:1067   192.168.2.20:23    192.168.2.20:23
```

```
Device# clear ip nat translation udp inside

192.168.2.20:1067 192.168.2.20:1067 192.168.2.20:23 192.168.2.20:23
```

```
Device#show ip nat translations

Pro  Inside global      Inside local       Outside local      Outside global
tcp  192.168.2.20:1220  192.168.2.95:1220  192.168.2.22:53    192.168.2.20:53
tcp  192.168.2.20:11012 192.168.2.209:11012 171.69.1.220:23   192.168.2.20:23
```

Additional References for Maintaining and Monitoring NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
NAT commands	*Cisco IOS IP Addressing Command Reference*
NAT concepts, configuration tasks, and examples configurations	*IP Addressing: NAT Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Monitoring and Maintaining NAT

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 7 Feature Information for Monitoring and Maintaining NAT

Feature Name	Releases	Feature Information
NAT-Forced Clear of Dynamic NAT Half-Entries	Cisco IOS XE Release 2.4	The NAT-Forced Clear of Dynamic NAT Half-Entries feature filters the display of the translation table by specifying an inside or outside address. The following commands were introduced or modified: **clear ip nat translations forced**,**show ip nat translations**.

Configuring Stateful Interchassis Redundancy

The Stateful Interchassis Redundancy feature enables you to configure pairs of devices to act as backups for each other.

This module describes conceptual information about and tasks for configuring stateful interchassis redundancy.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Stateful Interchassis Redundancy

All application redundancy configurations, including Network Address Translation (NAT) rules that have redundancy group associations and mapping IDs, must be identical on both devices, or NAT sessions will not be synchronized between devices and NAT redundancy will not work.

Restrictions for Stateful Interchassis Redundancy

- By default, Network Address Translation (NAT) high availability (inter and intrabox) does not replicate HTTP sessions to the standby device. To replicate HTTP sessions on the standby device during a switchover, you must configure the **ip nat switchover replication http** command.

- During NAT payload translations with certain applications, there can be IP addresses in the payload that require NAT translation. The application-level gateway (ALG) for that specific application parses the packet for these IP addresses, NAT translates these addresses, and the ALG writes the translated addresses back into the packet.

  Fixup denotes the writing of the translated IP address back into the packet. The write back of data can change the length of a packet, which results in the adjustment of the packet's TCP sequence (SEQ) or acknowledgment (ACK) values by NAT for the life of the TCP connection. NAT writes the new TCP SEQ/ACK values into the packet during SEQ/ACK fixup.

  For example, during a TCP ALG session, SEQ/ACK values may require fixup with mainly ASCII applications such as Domain Name System (DNS), FTP/FTP64, H.323, Real Time Streaming Protocol (RTSP), and Session Initiation Protocol (SIP). This SEQ/ACK adjustment information gets associated with the NAT session and is synchronized to the standby device periodically.

  During a stateful switchover, if the SEQ/ACK information is not completely synchronized to the new active device it is likely that the TCP connection would be reset by endpoints of the application.

Information About Stateful Interchassis Redundancy

Stateful Interchassis Redundancy Overview

You can configure the Stateful Interchassis Redundancy feature to determine the active device from a group of devices, based on a number of failover conditions. When a failover occurs, the standby device seamlessly takes over, starts performing traffic forwarding services, and maintains a dynamic routing table.

Stateful Interchassis Redundancy Operation

You can configure pairs of devices to act as hot standbys for each other. Redundancy is configured on an interface basis. Pairs of redundant interfaces are known as redundancy groups (RGs). Redundancy occurs at an application level and does not require a complete physical failure of the interface or device for a switchover of the application to occur. When a switchover occurs, the application activity continues to run seamlessly on the redundant interface.

The first figure below depicts an active/standby load-sharing scenario. The figure shows how an RG is configured for a pair of devices that has one outgoing interface. The second figure depicts an active/active load-sharing scenario. The figure below shows how two RGs are configured for a pair of devices that have two outgoing interfaces. Group A on ASR1 is the standby RG and Group A on ASR 2 is the active RG.

In both cases, redundant devices are joined by a configurable control link and a data synchronization link. The control link is used to communicate the status of devices. The data synchronization link is used to transfer stateful information from Network Address Translation (NAT) and the firewall and synchronize the stateful database. The pairs of redundant interfaces are configured with the same unique ID number known as the redundant interface identifier (RII).

Figure 6 Redundancy Group Configuration—One Outgoing Interface

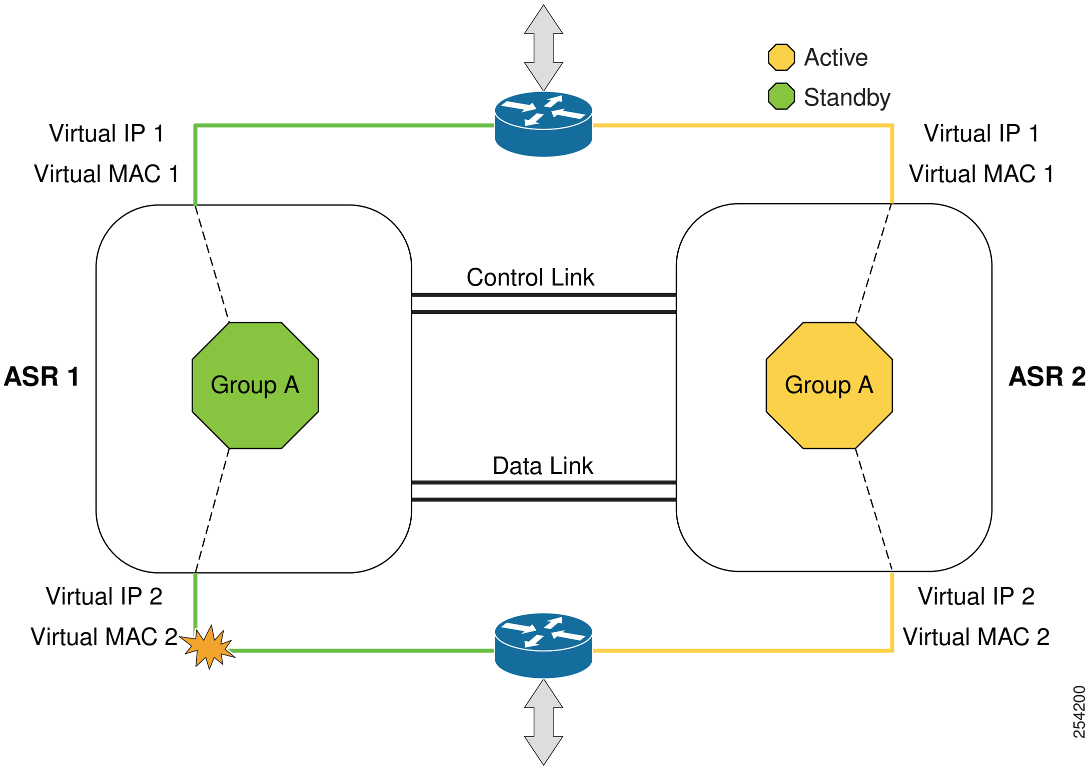
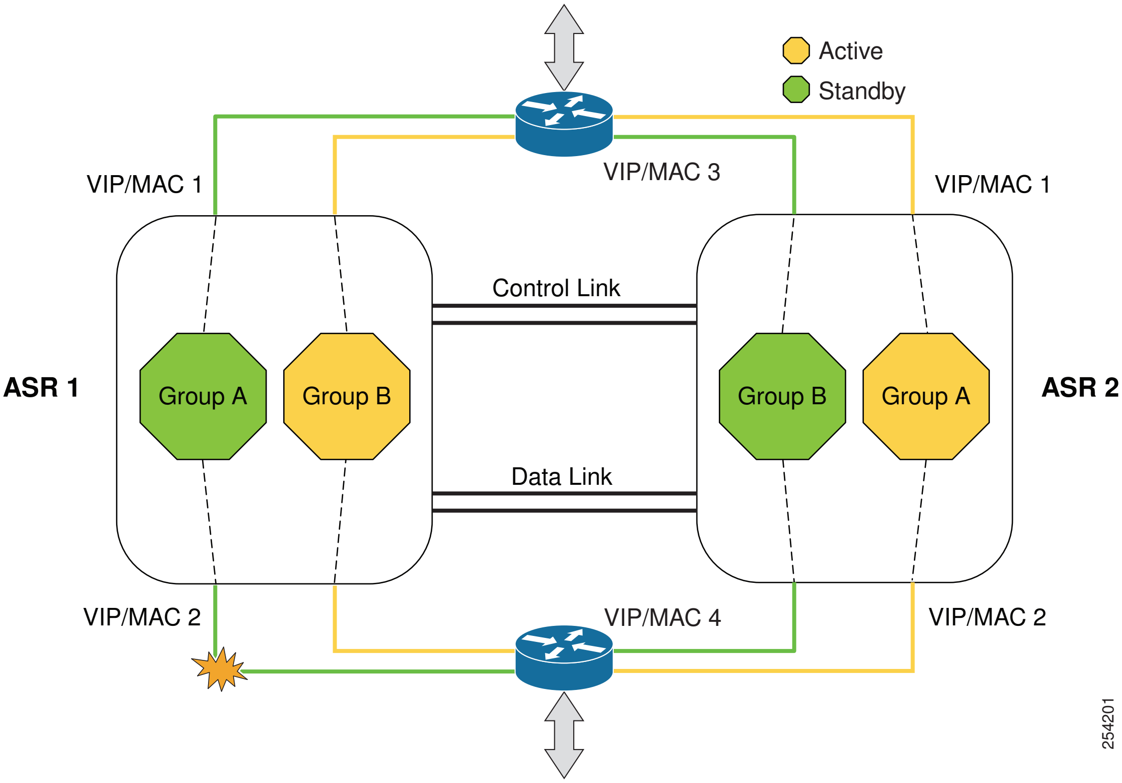


Figure 7 *Redundancy Group Configuration—Two Outgoing Interfaces*



The status of redundancy group members is determined through the use of hello messages sent over the control link. The software considers either device not responding to a hello message within a configurable amount of time to be a failure and initiates a switchover. For the software to detect a failure in milliseconds, control links run the failover protocol that is integrated with the Bidirectional Forwarding Detection (BFD) protocol. You can configure the following parameters for hello messages:

- Hello time—Interval at which hello messages are sent.
- Hold time—Amount of time before which the active or standby device is declared to be down.

The hello time defaults to 3 seconds to align with the Hot Standby Router Protocol (HSRP), and the hold time defaults to 10 seconds. You can also configure these timers in milliseconds by using the **timers hellotime msec** command.

To determine the pairs of interfaces that are affected by the switchover, you must configure a unique ID for each pair of redundant interfaces. This ID is known as the RII that is associated with the interface.

A switchover to the standby device can occur when the priority setting that is configured on each device changes. The device with the highest priority value acts as the active device. If a fault occurs on either the active or standby device, the priority of the device is decremented by a configurable amount known as the weight. If the priority of the active device falls below the priority of the standby device, a switchover occurs and the standby device becomes the active device. This default behavior can be overridden by disabling the preemption attribute for the RG. You can also configure each interface to decrease the priority when the Layer 1 state of the interface goes down. The priority that is configured overrides the default priority of an RG.

Each failure event that causes a modification of an RG priority generates a syslog entry that contains a time stamp, the RG that was affected, the previous priority, the new priority, and a description of the failure event cause.

A switchover also can occur when the priority of a device or interface falls below a configurable threshold level.

A switchover to the standby device occurs under the following circumstances:

- Power loss or a reload occurs on the active device (including reloads).
- The run-time priority of the active device goes below that of the standby device (with preempt configured).
- The run-time priority of the active device goes below that of the configured threshold.
- The redundancy group on the active device is reloaded manually. Use the **redundancy application reload group** *rg-number* command for a manual reload.

Associations with Firewalls and NAT

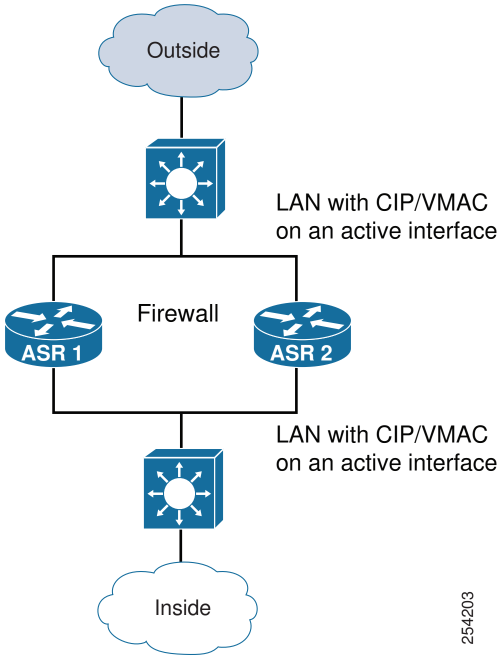
Firewalls use the association of the redundancy group with a traffic interface.

Network Address Translation (NAT) associates the redundancy group with a mapping ID.

LAN-LAN Topology

The figure below shows the LAN-LAN topology. In a LAN-LAN topology, all participating devices are connected to each other through LAN interfaces on both the inside and the outside. In this scenario, traffic is often directed to the correct firewall if static routing is configured on the upstream or downstream devices to an appropriate virtual IP address. Cisco ASR 1000 Aggregation Services Routers participate in dynamic routing with upstream or downstream devices. The dynamic routing configuration supported on LAN-facing interfaces must not introduce a dependency on the routing protocol convergence; otherwise, fast failover requirements will not be met.

Figure 8 LAN-LAN Topology



How to Configure Stateful Interchassis Redundancy

Configuring the Control Interface Protocol

The configuration for the control interface protocol consists of the following elements:

- Authentication information
- Group name
- Hello time
- Hold time
- Protocol instance
- Use of the bidirectional forwarding direction (BFD) protocol

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **mode sso**
5. **application redundancy**
6. **protocol** *number*
7. **name** *instance-name*
8. **timers hellotime** [**msec**] *number* **holdtime** [**msec**] *number*
9. **authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key* | **md5 key-chain** *key-chain-name*}
10. **bfd**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**mode sso** **Example:** `Device(config-red)# mode sso`	Sets the redundancy mode to stateful switchover (SSO) .
Step 5	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.
Step 6	**protocol** *number* **Example:** `Device(config-red-app)# protocol 4`	Specifies the protocol instance that will be attached to a control interface, and enters redundancy application protocol configuration mode.
Step 7	**name** *instance-name* **Example:** `Device(config-red-app-prot)# name rg1`	(Optional) Specifies an optional alias for the protocol instance.
Step 8	**timers hellotime** [**msec**] *number* **holdtime** [**msec**] *number* **Example:** `Device(config-red-app-prot)# timers hellotime 3 holdtime 10`	Specifies the interval between hello messages sent and the time before a device is declared to be down. • The default time for hello time is 3 seconds and for hold time is 10 seconds.

Step 9	**authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key* | **md5 key-chain** *key-chain-name*}
	Example:
	`Device(config-red-app-prot)# authentication text password`
		Specifies authentication information.

	Command or Action	Purpose
Step 10	**bfd** **Example:** `Device(config-red-app-prot)# bfd`	(Optional) Enables the integration of the failover protocol running on the control interface with the BFD protocol to achieve failure detection in milliseconds. • BFD is enabled by default.

Step 11	**end**
	Example:
	`Device(config-red-app-prot)# end`
		Exits redundancy application protocol configuration mode and enters privileged EXEC mode.

Configuring a Redundancy Group

Redundancy groups consist of the following configuration elements:

- The amount by which the priority will be decremented for each object.
- Faults (objects) that will decrement the priority.
- Failover priority.
- Failover threshold.
- Group instance.
- Group name.
- Initialization delay timer.
- The interface that is associated with the redundancy group (RG).
- The interface that is used as the control interface.
- The interface that is used as the data interface.
- The redundancy interface identifier (RII) number of the RG interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** {**1** | **2**}
6. **name** *group-name*
7. **preempt**
8. **priority** *number* **failover-threshold** *number*
9. **track** *object-number* [**decrement** *number* | **shutdown**]
10. **timers delay** *seconds* [**reload** *seconds*]
11. **control** *interface-name* **protocol** *instance*
12. **data** *interface-name*
13. To create another redundancy group, repeat Steps 3 through 12.
14. **end**
15. **configure terminal**
16. **interface** *type number*
17. **redundancy group** *number* **ip** *address* **exclusive** [**decrement** *number*]
18. **redundancy rii** *number*
19. **end**

DETAILED STEPS

	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	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.

	Command or Action	Purpose
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.

Step 5	**group** {**1** | **2**}
	Example:
	`Device(config-red-app)# group 1`
		Specifies the redundancy group instance and enters redundancy application group configuration mode.

Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name rg1`	(Optional) Specifies an optional alias for the protocol instance.
Step 7	**preempt** **Example:** `Device(config-red-app-grp)# preempt`	Enables preemption on the group and enables the standby device to preempt the active device regardless of which device has higher priority.
Step 8	**priority** *number* **failover-threshold** *number* **Example:** `Device(config-red-app-grp)# priority 120 failover-threshold 80`	Specifies the initial priority and failover threshold for the redundancy group.
Step 9	**track** *object-number* [**decrement** *number* | **shutdown**] **Example:** `Device(config-red-app-grp)# track 44 decrement 20`	Specifies the amount by which the priority of a redundancy group will be decremented if an event occurs. • You can track multiple objects that influence the priority of the redundancy group.
Step 10	**timers delay** *seconds* [**reload** *seconds*] **Example:** `Device(config-red-app-grp)# timers delay 10 reload 20`	Specifies the amount of time by which the redundancy group will delay role negotiations that start after a fault occurs or after the system is reloaded.
Step 11	**control** *interface-name* **protocol** *instance* **Example:** `Device(config-red-app-grp)# control GigabitEthernet0/1/0 protocol 1`	Specifies the control interface that is used by the redundancy group. • This interface is also associated with an instance of the control interface protocol.

	Command or Action	Purpose
Step 12	**data** *interface-name* **Example:** `Device(config-red-app-grp)# data GigabitEthernet0/1/2`	Specifies the data interface that is used by the redundancy group.
Step 13	To create another redundancy group, repeat Steps 3 through 12.	—

Step 14	**end**
	Example:
	`Device(config-red-app-grp)# end`
		Exits redundancy application group configuration mode and enters privileged EXEC mode.

Step 15	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 16	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Selects an interface to associate with the redundancy group and enters interface configuration mode.
Step 17	**redundancy group** *number* **ip** *address* **exclusive** [**decrement** *number*] **Example:** `Device(config-if)# redundancy group 1 ip 10.10.1.1 exclusive decrement 20`	Associates the interface with the redundancy group identified by the *number* argument.
Step 18	**redundancy rii** *number* **Example:** `Device(config-if)# redundancy rii 40`	Specifies a number for the RII associated with this interface. • This number must match the RII of the other interface in the redundancy group.

Step 19	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and enters privileged EXEC mode.

Configuring a Redundant Traffic Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ip nat outside**
6. **ip virtual-reassembly**
7. **negotiation auto**
8. **redundancy rii** *number*
9. **redundancy group** *number* **ip** *address* **exclusive** [**decrement** *number*]
10. **end**

DETAILED STEPS

	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 gigabitethernet 0/1/5`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.2 255.0.0.0`	Sets a primary or secondary IP address for an interface.
Step 5	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Configures the outside interface for IP address translation.

	Command or Action	Purpose
Step 6	**ip virtual-reassembly** **Example:** `Device(config-if)# ip virtual-reassembly`	Enables Virtual Fragmentation Reassembly (VFR) on an interface.
Step 7	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control of the Gigabit Ethernet interface.
Step 8	**redundancy rii** *number* **Example:** `Device(config-if)# redundancy rii 200`	Specifies a number for the redundancy interface identifier (RII) that is associated with this interface. • This number must match the RII of the other interface in the redundancy group.
Step 9	**redundancy group** *number* **ip** *address* **exclusive** [**decrement** *number*] **Example:** `Device(config-if)# redundancy group 1 ip 10.1.1.200 exclusive decrement 10`	Associates the interface with the redundancy group identified by the *number* argument.

Step 10	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and enters privileged EXEC mode.

Configuring NAT with Stateful Interchassis Redundancy

You must use a mapping ID to associate Network Address Translation (NAT) with a redundancy group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
4. **ip nat inside source list** {{*access-list-number* | *access-list-name*} | **route-map** *name*} **pool** *name* [**redundancy** *redundancy-id* [**mapping-id** *map-id* | **overload** | **reversible** | **vrf** *name*]]
5. **end**

DETAILED STEPS

	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 VPN-18 10.10.0.0 10.10.255.255 netmask 255.255.0.0`
		Defines a pool of IP addresses for NAT.

Step 4	**ip nat inside source list** {{*access-list-number* | *access-list-name*} | **route-map** *name*} **pool** *name* [**redundancy** *redundancy-id* [**mapping-id** *map-id* | **overload** | **reversible** | **vrf** *name*]] **Example:** `Device(config)# ip nat inside source list acl-18 pool VPN-18 redundancy 2 mapping-id 152`	Enables NAT of the inside source address. • You must use a mapping ID to associate NAT with the redundancy group.

Step 5	**end**
	Example:
	`Device(config)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Managing and Monitoring Stateful Interchassis Redundancy

All configuration commands in this task are optional. You can use the **show** commands in any order.

SUMMARY STEPS

1. **enable**
2. **redundancy application reload group** *number* [**peer** | **self**]
3. **show redundancy application group** [*group-id* | **all**]
4. **show redundancy application transport** {**clients** | **group** [*group-id*]}
5. **show redundancy application protocol** {*protocol-id* | **group** [*group-id*]}
6. **show redundancy application faults group** [*group-id*]
7. **show redundancy application if-mgr group** [*group-id*]
8. **show redundancy application control-interface group** [*group-id*]
9. **show redundancy application data-interface group** [*group-id*]
10. **show monitor event-trace rg_infra all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**redundancy application reload group** *number* [**peer** | **self**] **Example:** `Device# redundancy application reload group 2 self`	Forces the active redundancy group (RG) to reload and the standby RG to become the active RG. • Use the **redundancy application reload** command to verify if the redundancy configuration is working. You must enter this command on the active RG.
Step 3	**show redundancy application group** [*group-id* | **all**] **Example:** `Device# show redundancy application group 2`	Displays summary information for the specified group or for all groups.
Step 4	**show redundancy application transport** {**clients** | **group** [*group-id*]} **Example:** `Device# show redundancy application transport group 2`	Displays transport information for the specified group or for all groups.
Step 5	**show redundancy application protocol** {*protocol-id* | **group** [*group-id*]} **Example:** `Device# show redundancy application protocol 2`	Displays protocol information for the specified group or for all groups.

	Command or Action	Purpose
Step 6	**show redundancy application faults group** [*group-id*] **Example:** `Device# show redundancy application faults group 2`	Displays information about faults for the specified group or for all groups.
Step 7	**show redundancy application if-mgr group** [*group-id*] **Example:** `Device# show redundancy application if-mgr group 2`	Displays information about the interface manager (if-mgr) for the specified group or for all groups.
Step 8	**show redundancy application control-interface group** [*group-id*] **Example:** `Device# show redundancy application control-interface group IF-2`	Displays interface information associated with redundancy groups for the specified control interface.
Step 9	**show redundancy application data-interface group** [*group-id*] **Example:** `Device# show redundancy application data-interface group IF-2`	Displays interface information associated with redundancy groups for the specified data interface.
Step 10	**show monitor event-trace rg_infra all** **Example:** `Device# show monitor event-trace rg_infra all`	Displays event trace information associated with all redundancy groups.

Configuration Examples for Stateful Interchassis Redundancy

Example: Configuring the Control Interface Protocol

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# mode sso
Device(config-red)# application redundancy
Device(config-red-app)# protocol 4
Device(config-red-app-prot)# name rg1
Device(config-red-app-prot)# timers hellotime 3 holdtime 10
Device(config-red-app-prot)# authentication text password
Device(config-red-app-prot)# bfd
```

Example: Configuring a Redundancy Group

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name rg1
Device(config-red-app-grp)# preempt
Device(config-red-app-grp)# priority 120 failover-threshold 80
Device(config-red-app-grp)# track 44 decrement 20
Device(config-red-app-grp)# timers delay 10 reload 20
Device(config-red-app-grp)# control GigabitEthernet0/1/0 protocol 1
Device(config-red-app-grp)# data GigabitEthernet0/1/2
Device(config-red-app-grp)# end
Device# configure terminal
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# redundancy group 1 ip 10.10.1.1 exclusive decrement 20
Device(config-if)# redundancy rii 40
```

Example: Configuring a Redundant Traffic Interface

```
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/5
Device(config-if)# ip address 10.1.1.2 255.0.0.0
Device(config-if)# ip nat outside
Device(config-if)# ip virtual-reassembly
Device(config-if)# negotiation auto
Device(config-if)# redundancy rii 200
Device(config-if)# redundancy group 1 ip 10.1.1.200 exclusive decrement 10
```

Example: Configuring NAT with Stateful Interchassis Redundancy

```
Device# configure terminal
Device(config)# ip nat pool VPN-18 10.10.0.0 10.10.255.255 netmask 255.255.0.0
Device(config)# ip nat inside source list acl-18 pool VPN-18 redundancy 2 mapping-id 152
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
IP addressing commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
Fundamental principles of IP addressing and IP routing	*IP Routing Primer*

Standards and RFCs

Standards/RFCs	Title
RFC 791	*Internet Protocol*
RFC 1338	*Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*
RFC 1466	*Guidelines for Management of IP Address Space*
RFC 1716	*Towards Requirements for IP Routers*
RFC 1918	*Address Allocation for Private Internets*
RFC 3330	*Special-Use IP Addresses*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Stateful Interchassis Redundancy

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 8 Feature Information for Stateful Interchassis Redundancy

Feature Name	Releases	Feature Information
Stateful Interchassis Redundancy	Cisco IOS XE Release 3.1S	The Stateful Interchassis Redundancy feature enables you to configure pairs of devices to act as backups for each other.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Stateless Network Address Translation 64

The Stateless Network Address Translation 64 (NAT64) feature provides a translation mechanism that translates an IPv6 packet into an IPv4 packet and vice versa. The translation involves parsing the entire IPv6 header, including the extension headers, and extracting the relevant information and translating it into an IPv4 header. Similarly, the IPv4 header is parsed in its entirety, including the IPv4 options, to construct an IPv6 header. This processing happens on a per-packet basis on the interfaces that are configured for Stateless NAT64 translation.

The Stateless NAT64 translator enables native IPv6 or IPv4 communication and facilitates coexistence of IPv4 and IPv6 networks.

The Stateless NAT64 translator does not maintain any state information in the datapath. This translator is based on the IETF working group Behavior Engineering for Hindrance Avoidance (BEHAVE) drafts about the framework for IPv4/IPv6 translation. This draft describes the mechanism to translate an IPv6 packet to an IPv4 packet and vice versa, including the transport layer headers and Internet Control Message Protocol (ICMP).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Stateless Network Address Translation 64

The following restrictions apply to the Stateless NAT64 feature:

- Only valid IPv4-translatable addresses can be used for stateless translation.

- Multicast is not supported.
- Applications without a corresponding application layer gateway (ALG) may not work properly with the Stateless NAT64 translator.
- The translation of IPv4 options, IPv6 routing headers, hop-by-hop extension headers, destination option headers, and source routing headers are not supported.
- Fragmented IPv4 UDP packets that do not contain a UDP checksum are not translated.
- IPv6 packets with zero UDP checksum are not translated.

Information About Stateless Network Address Translation 64

Fragmentation of IP Datagrams in IPv6 and IPv4 Networks

In IPv4 networks, any intermediate router can do the fragmentation of an IP datagram. However, in IPv6 networks, fragmentation can be done only by the originating IPv6 host. Because fragmentation in IPv6 networks is done by the IPv6 hosts, the path maximum transmission unit (PMTU) discovery should also be done by the IPv6 hosts. However, a PMTU discovery is not possible across an IPv4 network where the routers are allowed to fragment the packets. In IPv4 networks, a Stateless NAT64 translator is used to fragment the IPv6 datagram and set the Don’t Fragment (DF) bits in the IPv4 header. Similarly, the translator can add the fragment header to the IPv6 packet if an IPv4 fragment is received.

Translation of ICMP for Stateless NAT64 Translation

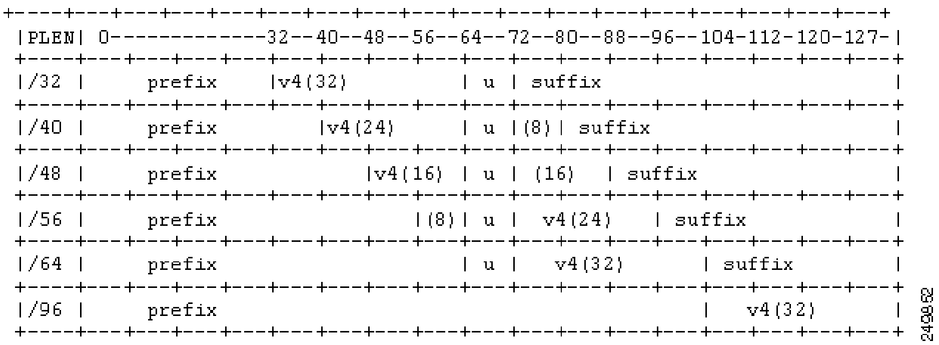
The IETF draft on the IP/ICMP translation algorithm describes the ICMP types or codes that should be translated between IPv4 and IPv6. ICMP errors embed the actual IP header and the transport header. Because the ICMP errors are embedded in the IP header, the IP header is not translated properly. For ICMP error packets, Stateless NAT64 translation should be applied twice: once for the outer header, and once again for the embedded header.

IPv4-Translatable IPv6 Address

IPv4-translatable IPv6 addresses are IPv6 addresses assigned to the IPv6 nodes for use with stateless translation. IPv4-translatable addresses consist of a variable-length prefix, an embedded IPv4 address, fixed universal bits (u-bits), and in some cases a suffix. IPv4-embedded IPv6 addresses are IPv6 addresses in which 32 bits contain an IPv4 address. This format is the same for both IPv4-converted and IPv4-translatable IPv6 addresses.

The figure below shows an IPv4-translatable IPv6 address format with several different prefixes and embedded IPv4 address positions.

Figure 9 IPv4-Translatable IPv6 Address Format



```
+----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 |PLEN| 0-------------32--40--48--56--64--72--80--88--96--104-112-120-127-|
 +----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 |/32 |     prefix    |v4(32)         | u | suffix                        |
 +----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 |/40 |     prefix        |v4(24)     | u |(8)| suffix                    |
 +----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 |/48 |     prefix            |v4(16) | u | (16)  | suffix                |
 +----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 |/56 |     prefix                |(8)| u |  v4(24)   | suffix            |
 +----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 |/64 |     prefix                    | u |   v4(32)      | suffix        |
 +----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 |/96 |     prefix                                        |   v4(32)      |
 +----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

- Prefixes Format, page 133

Prefixes Format

A set of bits at the start of an IPv6 address is called the format prefix. Prefix length is a decimal value that specifies how many of the leftmost contiguous bits of an address comprise the prefix.

An embedded IPv4 address is used to construct IPv4 addresses from the IPv6 packet. The Stateless NAT64 translator has to derive the IPv4 addresses that are embedded in the IPv6-translatable address by using the prefix length. The translator has to construct an IPv6-translatable address based on the prefix and prefix length and embed the IPv4 address based on the algorithm.

According to the IETF address format BEHAVE draft, a u-bit (bit 70) defined in the IPv6 architecture should be set to zero. For more information on the u-bit usage, see RFC 2464. The reserved octet, also called u-octet, is reserved for compatibility with the host identifier format defined in the IPv6 addressing architecture. When constructing an IPv6 packet, the translator has to make sure that the u-bits are not tampered with and are set to the value suggested by RFC 2373. The suffix will be set to all zeros by the translator. IETF recommends that the 8 bits of the u-octet (bit range 64-71) should be set to zero.

The prefix lengths of 32, 40, 48, 56, 64, or 96 are supported for Stateless NAT64 translation. The Well Known Prefix (WKP) is not supported. When traffic flows from the IPv4-to-IPv6 direction, either a WKP or a configured prefix can be added only in stateful translation.

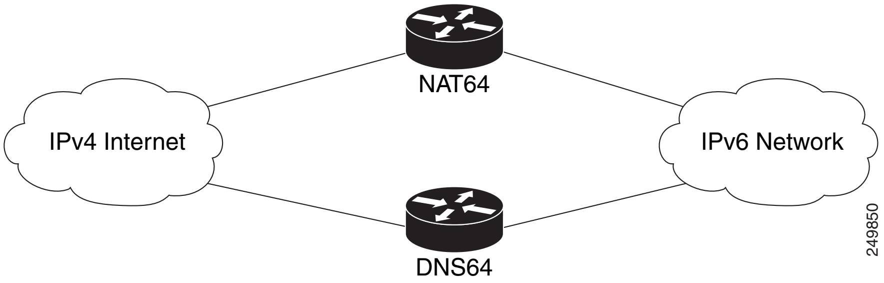
Supported Stateless NAT64 Scenarios

The IETF framework draft for IPv4/IPv6 translation describes eight different network communication scenarios for Stateless NAT64 translation. The following scenarios are supported by the Cisco IOS Stateless NAT64 feature and are described in this section:

- Scenario 1--an IPv6 network to the IPv4 Internet
- Scenario 2--the IPv4 Internet to an IPv6 network
- Scenario 5--an IPv6 network to an IPv4 network
- Scenario 6--an IPv4 network to an IPv6 network

The figure below shows stateless translation for scenarios 1 and 2. An IPv6-only network communicates with the IPv4 Internet.

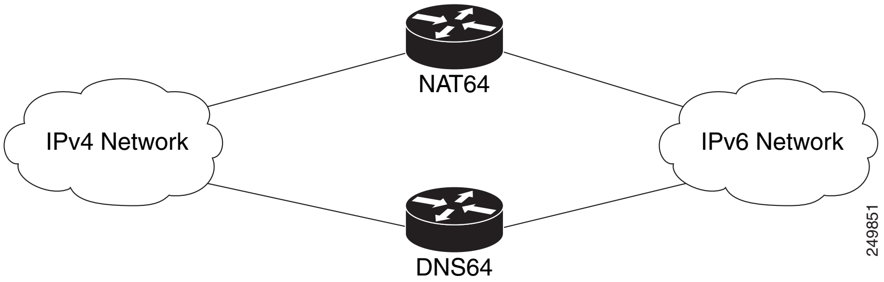
Figure 10 Stateless Translation for Scenarios 1 and 2



Scenario 1 is an IPv6 initiated connection and scenario 2 is an IPv4 initiated connection. Stateless NAT64 translates these two scenarios only if the IPv6 addresses are IPv4 translatable. In these two scenarios, the Stateless NAT64 feature does not help with IPv4 address depletion, because each IPv6 host that communicates with the IPv4 Internet is a globally routable IPv4 address. This consumption is similar to the IPv4 consumption rate as a dual-stack. The savings, however, is that the internal network is 100 percent IPv6, which eases management (Access Control Lists, routing tables), and IPv4 exists only at the edge where the Stateless translators live.

The figure below shows stateless translation for scenarios 5 and 6. The IPv4 network and IPv6 network are within the same organization.

Figure 11 Stateless Translation for Scenarios 5 and 6



The IPv4 addresses used are either public IPv4 addresses or RFC 1918 addresses. The IPv6 addresses used are either public IPv6 addresses or Unique Local Addresses (ULAs).

Both these scenarios consist of an IPv6 network that communicates with an IPv4 network. Scenario 5 is an IPv6 initiated connection and scenario 6 is an IPv4 initiated connection. The IPv4 and IPv6 addresses may not be public addresses. These scenarios are similar to the scenarios 1 and 2. The Stateless NAT64 feature supports these scenarios if the IPv6 addresses are IPv4 translatable.

Multiple Prefixes Support for Stateless NAT64 Translation

Network topologies that use the same IPv6 prefix for source and destination addresses may not handle routing correctly and may be difficult to troubleshoot. The Stateless NAT64 feature addresses these challenges in Cisco IOS XE Release 3.3S and later releases through the support of multiple prefixes for stateless translation. The entire IPv4 Internet is represented as using a different prefix from the one used for the IPv6 network.

How to Configure Stateless Network Address Translation 64

Configuring a Routing Network for Stateless NAT64 Communication

Perform this task to configure and verify a routing network for Stateless NAT64 communication.

- An IPv6 address assigned to any host in the network should have a valid IPv4-translatable address and vice versa.
- You should enable the **ipv6 unicast-routing** command for this configuration to work.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **description** *string*
6. **ipv6 enable**
7. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}
8. **nat64 enable**
9. **exit**
10. **interface** *type number*
11. **description** *string*
12. **ip address** *ip-address mask*
13. **nat64 enable**
14. **exit**
15. **nat64 prefix stateless** *ipv6-prefix/length*
16. **nat64 route** *ipv4-prefix/mask interface-type interface-number*
17. **ipv6 route** *ipv4-prefix/length interface-type interface-number*
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 5	**description** *string* **Example:** `Router(config-if)# description interface facing ipv6`	Adds a description to an interface configuration.
Step 6	**ipv6 enable** **Example:** `Router(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.

Step 7	**ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
	Example:
	`Router(config-if)# ipv6 address 2001:DB8::1/128`
		Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.

Step 8	**nat64 enable** **Example:** `Router(config-if)# nat64 enable`	Enables Stateless NAT64 translation on an IPv6 interface.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/2/0`	Configures an interface type and enters interface configuration mode.
Step 11	**description** *string* **Example:** `Router(config-if)# description interface facing ipv4`	Adds a description to an interface configuration.
Step 12	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 198.51.100.1 255.255.255.0`	Configures an IPv4 address for an interface.
Step 13	**nat64 enable** **Example:** `Router(config-if)# nat64 enable`	Enables Stateless NAT64 translation on an IPv4 interface.
Step 14	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Step 15	**nat64 prefix stateless** *ipv6-prefix*/*length*
	Example:
	`Router(config)# nat64 prefix stateless`
	`2001:0db8:0:1::/96`
		Defines the Stateless NAT64 prefix to be added to the IPv4 hosts to translate the IPv4 address into an IPv6 address.
		• The command also identifies the prefix that must be used to create the IPv4-translatable addresses for the IPv6 hosts.

	Command or Action	Purpose
Step 16	**nat64 route** *ipv4-prefix/mask interface-type interface-number* **Example:** `Router(config)# nat64 route 203.0.113.0/24 gigabitethernet 0/0/0`	Routes the IPv4 traffic towards the correct IPv6 interface.
Step 17	**ipv6 route** *ipv4-prefix/length interface-type interface-number* **Example:** `Router(config)# ipv6 route 2001:DB8:0:1::CB00:7100/120 gigabitethernet 0/0/0`	Routes the translated packets to the IPv4 address. • You must configure the **ipv6 route** command if your network is not running IPv6 routing protocols.

Step 18	**end**
	Example:
	`Router(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Configuring Multiple Prefixes for Stateless NAT64 Translation

Perform this task to configure multiple prefixes for Stateless NAT64 translation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **ipv6 address** {*ipv6-address /prefix-length* | *prefix-name sub-bits/prefix-length*}
6. **ipv6 enable**
7. **nat64 enable**
8. **nat64 prefix stateless v6v4** *ipv6-prefix/length*
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **negotiation auto**
13. **nat64 enable**
14. **exit**
15. **nat64 prefix stateless v4v6** *ipv6-prefix/length*
16. **nat64 route** *ipv4-prefix/mask interface-type interface-number*
17. **ipv6 route** *ipv6-prefix/length interface-type interface-number*
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Configures an interface type and enters interface configuration mode.

Step 5	**ipv6 address** {*ipv6-address* /*prefix-length* | *prefix-name sub-bits*/ *prefix-length*}
	Example:
	`Router(config-if)# ipv6 address 2001:DB8::1/128`
		Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.

Step 6	**ipv6 enable** **Example:** `Router(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.
Step 7	**nat64 enable** **Example:** `Router(config-if)# nat64 enable`	Enables Stateless NAT64 translation on an IPv6 interface.

	Command or Action	Purpose
Step 8	**nat64 prefix stateless v6v4** *ipv6-prefix/length* **Example:** `Router(config-if)# nat64 prefix stateless v6v4 2001:0db8:0:1::/96`	Maps an IPv6 address to an IPv4 host for Stateless NAT 64 translation. • The NAT64 prefix in the command is the same as the prefix of the source packet that is coming from the IPv6-to-IPv4 direction.
Step 9	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/2/0`	Configures an interface type and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 203.0.113.1 255.255.255.0`	Configures an IPv4 address for an interface.
Step 12	**negotiation auto** **Example:** `Router(config-if)# negotiation auto`	Enables the autonegotiation protocol to configure the speed, duplex, and automatic flow control on an interface.
Step 13	**nat64 enable** **Example:** `Router(config-if)# nat64 enable`	Enables Stateless NAT64 translation on an IPv4 interface.
Step 14	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 15	**nat64 prefix stateless v4v6** *ipv6-prefix/length* **Example:** `Router(config)# nat64 prefix stateless v4v6 2001:DB8:2::1/96`	Maps an IPv4 address to an IPv6 host for Stateless NAT 64 translation. • This command identifies the prefix that creates the IPv4-translatable addresses for the IPv6 hosts.
Step 16	**nat64 route** *ipv4-prefix/mask interface-type interface-number* **Example:** `Router(config)# nat64 route 203.0.113.0/24 gigabitethernet 0/0/0`	Routes the IPv4 traffic towards the correct IPv6 interface.
Step 17	**ipv6 route** *ipv6-prefix/length interface-type interface-number* **Example:** `Router(config)# ipv6 route 2001:DB8:0:1::CB00:7100/120 gigabitethernet 0/0/0`	Routes the translated packets to the IPv4 address. • You must configure the **ipv6 route** command if your network is not running IPv6 routing protocols.

Step 18	**end**
	Example:
	`Router(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Monitoring and Maintaining the Stateless NAT64 Routing Network

Perform this task to verify and monitor the Stateless NAT64 routing network. In the privileged EXEC mode, you can enter the commands in any order.

SUMMARY STEPS

1. **show nat64 statistics**
2. **show ipv6 route**
3. **show ip route**
4. **debug nat64** {**all** | **ha** {**all** | **info** | **trace** | **warn**} | **id-manager** | **info** | **issu** {**all** | **message** | **trace**} | **memory** | **statistics** | **trace** | **warn**}
5. **ping** [*protocol* [**tag**]] {*host-name* | *system-address*}

DETAILED STEPS

Step 1 **show nat64 statistics**
This command displays the global and interface-specific statistics of the packets that are translated and dropped.

Example:

```
Router# show nat64 statistics

NAT64 Statistics
Global Stats:
   Packets translated (IPv4 -> IPv6): 21
   Packets translated (IPv6 -> IPv4): 15
GigabitEthernet0/0/1 (IPv4 configured, IPv6 configured):
   Packets translated (IPv4 -> IPv6): 5
   Packets translated (IPv6 -> IPv4): 0
   Packets dropped: 0
GigabitEthernet1/2/0 (IPv4 configured, IPv6 configured):
   Packets translated (IPv4 -> IPv6): 0
   Packets translated (IPv6 -> IPv4): 5
   Packets dropped: 0
```

Step 2 **show ipv6 route**

This command displays the configured stateless prefix and the specific route for the IPv4 embedded IPv6 address pointing toward the IPv6 side.

Example:

```
Router# show ipv6 route

IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
ND - Neighbor Discovery
O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
LC  2001::1/128 [0/0] via FastEthernet0/3/4, receive
S   2001::1B01:10A/128 [1/0] via FastEthernet0/3/4, directly connected
S   3001::/96 [1/0] via ::42, NVI0
S   3001::1E1E:2/128 [1/0] via FastEthernet0/3/0, directly connected
LC  3001::C0A8:64D5/128 [0/0] via FastEthernet0/3/0, receive
L   FF00::/8 [0/0] via Null0, receive
```

Step 3 **show ip route**

This command displays the IPv4 addresses in the Internet that have reached the IPv4 side.

Example:

```
Router# show ip route

Codes: R - RIP derived, O - OSPF derived,
       C - connected, S - static, B - BGP derived,
       * - candidate default route, IA - OSPF inter area route,
       i - IS-IS derived, ia - IS-IS, U - per-user static route,
       o - on-demand routing, M - mobile, P - periodic downloaded static route,
       D - EIGRP, EX - EIGRP external, E1 - OSPF external type 1 route,
       E2 - OSPF external type 2 route, N1 - OSPF NSSA external type 1 route,
       N2 - OSPF NSSA external type 2 route
Gateway of last resort is 10.119.254.240 to network 10.140.0.0
O E2 10.110.0.0 [160/5] via 10.119.254.6, 0:01:00, Ethernet2
E    10.67.10.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
O E2 10.68.132.0 [160/5] via 10.119.254.6, 0:00:59, Ethernet2
O E2 10.130.0.0 [160/5] via 10.119.254.6, 0:00:59, Ethernet2
E    10.128.0.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
E    10.129.0.0 [200/129] via 10.119.254.240, 0:02:22, Ethernet2
E    10.65.129.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
E    10.10.0.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
E    10.75.139.0 [200/129] via 10.119.254.240, 0:02:23, Ethernet2
E    10.16.208.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
E    10.84.148.0 [200/129] via 10.119.254.240, 0:02:23, Ethernet2
```

```
E    10.31.223.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
E    10.44.236.0 [200/129] via 10.119.254.240, 0:02:23, Ethernet2
E    10.141.0.0 [200/129] via 10.119.254.240, 0:02:22, Ethernet2
E    10.140.0.0 [200/129] via 10.119.254.240, 0:02:23, Ethernet2
IPv6 Routing Table - default - 6 entries
```

Step 4 **debug nat64** {**all** | **ha** {**all** | **info** | **trace** | **warn**} | **id-manager** | **info** | **issu** {**all** | **message** | **trace**} | **memory** | **statistics** | **trace** | **warn**}

This command enables Stateless NAT64 debugging.

Example:

```
Router# debug nat64 statistics

NAT64 statistics debugging is on
Sep 16 18:26:24.537 IST: NAT64 (stats): Received stats update for IDB(FastEthernet0/3/5)
Sep 16 18:26:24.537 IST: NAT64 (stats): Updating pkts_translated_v4v6 from 94368894 to 95856998
(is_delta(TRUE) value(1488104))
Sep 16 18:26:24.537 IST: NAT64 (stats): Received stats update for IDB(FastEthernet0/3/4)
Sep 16 18:26:24.537 IST: NAT64 (stats): Updating pkts_translated_v6v4 from 7771538 to 7894088
(is_delta(TRUE) value(122550))
Sep 16 18:26:24.537 IST: NAT64 (stats): Received global stats update
Sep 16 18:26:24.537 IST: NAT64 (stats): Updating pkts_translated_v4v6 from 1718650332 to
1720138437 (is_delta(TRUE) value(1488105))
Sep 16 18:26:24.537 IST: NAT64 (stats): Updating pkts_translated_v6v4 from 1604459283 to
1604581833 (is_delta(TRUE) value(122550))
```

Step 5 **ping** [*protocol* [**tag**]] {*host-name* | *system-address*}

The following is a sample packet capture from the IPv6 side when you specify the **ping 198.168.0.2** command after you configure the **nat64 enable** command on both the IPv4 and IPv6 interfaces:

Example:

```
Router# ping 198.168.0.2

Time             Source            Destination         Protocol       Info
1 0.000000       2001::c6a7:2      2001::c6a8:2        ICMPv6         Echo request
Frame 1: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
  Arrival Time: Oct 8, 2010 11:54:06.408354000 India Standard Time
  Epoch Time: 1286519046.408354000 seconds
  [Time delta from previous captured frame: 0.000000000 seconds]
  [Time delta from previous displayed frame: 0.000000000 seconds]
  [Time since reference or first frame: 0.000000000 seconds]
  Frame Number: 1
  Frame Length: 118 bytes (944 bits)
  Capture Length: 118 bytes (944 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocol in frame: eth:1pv6:icmpv6: data]
Ethernet II, Src:Cisco_c3:64:94 (00:22:64:c3:64:94), Dst: Cisco_23:f2:30 (00:1f:6c:23:f2:30)
  Destination: Cisco_23:f2:30 (00:1f:6c:23:f2:30)
     Address: Cisco_23:f2:30 (00:1f:6c:23:f2:30)
     .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
     .... ...0 .... .... .... .... = LG bit: Globally unique address (factory default)
  Source: Cisco_c3:64:94 (00:22:64:c3:64:94)
    Address: Cisco_c3:64:94 (00:22:64:c3:64:94)
     .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
     .... ...0 .... .... .... .... = LG bit: Globally unique address (factory default)
  Type: IPv6 (0x86dd)
Internet Protocol Version 6, src: 2001::c6a7:2 (2001::c6a7:2), Dst: 2001::c6a8:2 (2001::c6a8:2)
   0110 .... = Version: 6
     [0110 .... = This field makes the filter “ip.version ==6” possible:: 6]
  .... 0000 0000 ... .... .... .... .... = Traffic class: 0x00000000
    .... 0000 00.. .... .... .... .... .... = Differentiated Services Field: Default (0x00000000)
    .... .... ..0. .... .... .... ... .... = ECN-Capable Transport (ECT): Not set
  .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
  Payload length: 64
  Next header: 64
  Hop limit: 64
  Source: 2001::c6a7:2 (2001::c6a7:2)
  [Source Teredo Server IPv4: 0.0.0.0 (0.0.0.0)]
  [Source Teredo Port: 6535]
  [Source Teredo Client IPv4: 198.51.100.1 (198.51.100.1)]
  Destination: 2001:c6a8:2 (2001::c6a8:2)
  [Destination Teredo Server IPv4: 0.0.0.0 {0.0.0.0)]
  [Destination Teredo Port: 65535]
  [Destination Teredo Client IPv4: 198.51.100.2 {198.51.100.2)]
Internet Control Message Protocol v6
  Type: 128 (Echo request)
  Code: 0 (Should always be zero)
  Checksum: 0xaed2 [correct]
  ID: 0x5018
  Sequence: 0x0000
  Data (56 bytes)
    Data: 069ae4c0d3b060008090a0b0c0d0e0f1011121314151617...
    [Length: 57]
```

Configuration Examples for Stateless Network Address Translation 64

Example Configuring a Routing Network for Stateless NAT64 Translation

The following example shows how to configure a routing network for Stateless NAT64 translation:

```
ipv6 unicast-routing
!
interface gigabitethernet 0/0/0
 description interface facing ipv6
 ipv6 enable
 ipv6 address 2001:DB8::1/128
 nat64 enable
!

interface gigabitethernet 1/2/0
 description interface facing ipv4
 ip address 198.51.100.1 255.255.255.0
 nat64 enable
!

nat64 prefix stateless 2001:0db8:0:1::/96
nat64 route 203.0.113.0/24 gigabitethernet 0/0/0
ipv6 route 2001:DB8:0:1::CB00:7100/120 gigabitethernet 0/0/0
```

Example: Configuring Multiple Prefixes for Stateless NAT64 Translation

```
ipv6 unicast-routing
!
interface gigabitethernet 0/0/0
```

```
 ipv6 address 2001:DB8::1/128
 ipv6 enable
 nat64 enable
 nat64 prefix stateless v6v4 2001:0db8:0:1::/96
!
interface gigabitethernet 1/2/0
 ip address 198.51.100.1 255.255.255.0
 negotiation auto
 nat64 enable
!
nat64 prefix stateless v4v6 2001:DB8:2::1/96
ipv6 route 2001:DB8:0:1::CB00:7100/120 gigabitethernet 0/0/0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
NAT commands	Cisco IOS IP Addressing Command Reference
NAT concepts, configuration tasks, and examples	Cisco IOS XE IP Addressing Services Configuration Guide

Standards

Standard	Title
Framework for IPv4/IPv6 translation	Framework for IPv4/IPv6 Translation draft-ietf-behave-v6v4-framework-03
IETF address format-10 BEHAVE draft	IPv6 Addressing of IPv4/IPv6 Translators draft-ietf-behave-address-format-10
IP/ICMP translation algorithm	IP/ICMP Translation Algorithm draft-ietf-behave-v6v4-xlate-05
IPv6 addressing of IPv4/IPv6 translators	IPv6 Addressing of IPv4/IPv6 Translators draft-ietf-behave-address-format-02

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1191	*Path MTU discovery*
RFC 1918	*Address Allocation for Private Internets*
RFC 2373	*IP Version 6 Addressing Architecture*
RFC 2464	*Transmission of IPv6 Packets over Ethernet Networks*
RFC 2765	*Stateless IP/ICMP Translation Algorithm (SIIT)*
RFC 2766	*Network Address Translation - Protocol Translation (NAT-PT)*
RFC 4787	*Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*
RFC 4966	*Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Stateless Network Address Translation 64

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 9 Feature Information for Stateless Network Address Translation 64

Feature Name	Releases	Feature Information
Stateless Network Address Translation 64	Cisco IOS XE Release 3.2S	The Stateless Network Address Translation 64 feature provides a translation mechanism that translates an IPv6 packet into an IPv4 packet and vice versa. The translation involves parsing the entire IPv6 header, including the extension headers, and extracting the relevant information and translating it into an IPv4 header. Similarly, the IPv4 header is parsed in its entirety, including the IPv4 options, to construct an IPv6 header. This processing happens on a per-packet basis on the interfaces that are configured for Stateless NAT64 translation. The following commands were introduced or modified: **clear nat64 ha statistics**, **clear nat64 statistics**, **debug nat64**, **nat64 enable**, **nat64 prefix**, **nat64 route**, **show nat64 adjacency**, **show nat64 ha status**, **show nat64 prefix stateless**, **show nat64 routes**, and **show nat64 statistics**.

Glossary

ALG—application-layer gateway or application-level gateway.

FP—Forward Processor.

IPv4-converted address—IPv6 addresses used to represent the IPv4 hosts. These have an explicit mapping relationship to the IPv4 addresses. This relationship is self-described by mapping the IPv4 address in the IPv6 address. Both stateless and stateful translators use IPv4-converted IPv6 addresses to represent the IPv4 hosts.

IPv6-converted address—IPv6 addresses that are assigned to the IPv6 hosts for the stateless translator. These IPv6-converted addresses have an explicit mapping relationship to the IPv4 addresses. This relationship is self-described by mapping the IPv4 address in the IPv6 address. The stateless translator uses the corresponding IPv4 addresses to represent the IPv6 hosts. The stateful translator does not use IPv6-converted addresses, because the IPv6 hosts are represented by the IPv4 address pool in the translator via dynamic states.

NAT—Network Address Translation.

RP—Route Processor.

stateful translation—In stateful translation a per-flow state is created when the first packet in a flow is received. A translation algorithm is said to be stateful if the transmission or reception of a packet creates or modifies a data structure in the relevant network element. Stateful translation allows the use of multiple translators interchangeably and also some level of scalability. Stateful translation is defined to enable the IPv6 clients and peers without mapped IPv4 addresses to connect to the IPv4-only servers and peers.

stateless translation—A translation algorithm that is not stateful is called stateless. A stateless translation requires configuring a static translation table, or may derive information algorithmically from the messages it is translating. Stateless translation requires less computational overhead than stateful translation. It also requires less memory to maintain the state, because the translation tables and the associated methods and processes exist in a stateful algorithm and do not exist in a stateless one. Stateless translation enables the IPv4-only clients and peers to initiate connections to the IPv6-only servers or peers that are equipped with IPv4-embedded IPv6 addresses. It also enables scalable coordination of IPv4-only stub networks or ISP IPv6-only networks. Because the source port in an IPv6-to-IPv4 translation may have to be changed to provide adequate flow identification, the source port in the IPv4-to-IPv6 direction need not be changed.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Stateful Network Address Translation 64

The Stateful Network Address Translation 64 feature provides a translation mechanism that translates IPv6 packets into IPv4 packets and vice versa. The stateful NAT64 translator algorithmically translates the IPv4 addresses of IPv4 hosts to and from IPv6 addresses by using the configured stateful prefix. In a similar manner, the IPv6 addresses of IPv6 hosts are translated to and from IPv4 addresses through Network Address Translation (NAT). Stateful Network Address Translation 64 (NAT64) also translates protocols and IP addresses. The Stateful NAT64 translator enables native IPv6 or IPv4 communication and facilitates coexistence of IPv4 and IPv6 networks.

This document explains how Stateful NAT64 works and how to configure your network for Stateful NAT64 translation.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring Stateful Network Address Translation 64

- For Domain Name System (DNS) traffic to work, you must have a separate working installation of DNS64.

Restrictions for Configuring Stateful Network Address Translation 64

- Applications without a corresponding application-level gateway (ALG) may not work properly with the Stateful NAT64 translator.
- IP Multicast is not supported.
- Stateful NAT64 supports only cold redundancy. There are two redundancy mechanisms: cold redundancy and hot redundancy. The redundancy mechanisms make the switchover of NAT64 boxes transparent to IPv6 hosts. In cold redundancy, the mapping of states is not synchronized among NAT64 boxes and the already established connections are interrupted during a switchover of NAT64 boxes.
- The translation of IPv4 options, IPv6 routing headers, hop-by-hop extension headers, destination option headers, and source routing headers is not supported.
- Virtual routing and forwarding (VRF)-aware NAT64 is not supported.
- When traffic flows from IPv6 to IPv4, the destination IP address that you have configured must match a stateful prefix to prevent hairpinning loops. However, the source IP address (source address of the IPv6 host) must not match the stateful prefix. If the source IP address matches the stateful prefix, packets are dropped.

  Hairpinning allows two endpoints inside Network Address Translation (NAT) to communicate with each other, even when the endpoints use only each other's external IP addresses and ports for communication.

Information About Stateful Network Address Translation 64

Stateful Network Address Translation 64

The Stateful NAT64 feature provides a translation mechanism that translates IPv6 packets into IPv4 packets and vice versa.

Stateful NAT64 supports Internet Control Message Protocol (ICMP), TCP, and UDP traffic. Packets that are generated in an IPv6 network and are destined for an IPv4 network are routed within the IPv6 network towards the Stateful NAT64 translator. Stateful NAT64 translates the packets and forwards them as IPv4 packets through the IPv4 network. The process is reversed for traffic that is generated by hosts connected to the IPv4 network and destined for an IPv6 receiver.

The Stateful NAT64 translation is not symmetric, because the IPv6 address space is larger than the IPv4 address space and a one-to-one address mapping is not possible. Before it can perform an IPv6 to an IPv4 translation, Stateful NAT64 requires a state that binds the IPv6 address and the TCP/UDP port to the IPv4 address. The binding state is either statically configured or dynamically created when the first packet that flows from the IPv6 network to the IPv4 network is translated. After the binding state is created, packets flowing in both directions are translated. In dynamic binding, Stateful NAT64 supports communication initiated by the IPv6-only node toward an IPv4-only node. Static binding supports communication initiated by an IPv4-only node to an IPv6-only node and vice versa. Stateful NAT64 with port overloading provides a 1:*n* mapping between IPv4 and IPv6 addresses.

According to the Behavior Engineering for Hindrance Avoidance (BEHAVE) drafts about the framework for Stateful NAT64 standards, when an IPv6 node initiates traffic through Stateful NAT64, and the incoming packet does not have an existing state, the following events happen:

- The source IPv6 address (and the source port) is associated with an IPv4 configured pool address (and port, based on the configuration).
- The destination IPv6 address is translated mechanically based on the BEHAVE translation draft using either the configured NAT64 stateful prefix or the Well Known Prefix (WKP).
- The packet is translated from IPv6 to IPv4 and forwarded to the IPv4 network.

When an incoming packet is stateful (if a state exists for an incoming packet), NAT64 identifies the state and uses the state to translate the packet.

When Stateful NAT64 is configured on an interface, Virtual Fragmentation Reassembly (VFR) is configured automatically.

Prefixes Format for Stateful Network Address Translation 64

A set of bits at the start of an IPv6 address is called the format prefix. Prefix length is a decimal value that specifies how many of the leftmost contiguous bits of an address comprise the prefix.

When packets flow from the IPv6 to the IPv4 direction, the IPv4 host address is derived from the destination IP address of the IPv6 packet that uses the prefix length. When packets flow from the IPv4 to the IPv6 direction, the IPv4 host address is constructed using the stateful prefix.

According to the IETF address format BEHAVE draft, a u-bit (bit 70) defined in the IPv6 architecture should be set to zero. For more information on the u-bit usage, see RFC 2464. The reserved octet, also called u-octet, is reserved for compatibility with the host identifier format defined in the IPv6 addressing architecture. When constructing an IPv6 packet, the translator has to make sure that the u-bits are not tampered with and are set to the value suggested by RFC 2373. The suffix will be set to all zeros by the translator. IETF recommends that the 8 bits of the u-octet (bit range 64–71) be set to zero.

- Well Known Prefix, page 151

Well Known Prefix

The Well Known Prefix 64:FF9B::/96 is supported for Stateful NAT64. During a stateful translation, if no stateful prefix is configured (either on the interface or globally), the WKP prefix is used to translate the IPv4 host addresses.

Stateful IPv4-to-IPv6 Packet Flow

The packet flow of IPv4-initiated packets for Stateful NAT64 is as follows:

- The destination address is routed to a NAT Virtual Interface (NVI).

  A virtual interface is created when Stateful NAT64 is configured. For Stateful NAT64 translation to work, all packets must get routed to the NVI. When you configure an address pool, a route is automatically added to all IPv4 addresses in the pool. This route automatically points to the NVI.
- The IPv4-initiated packet hits static or dynamic binding.

  Dynamic address bindings are created by the Stateful NAT64 translator when you configure dynamic Stateful NAT64. A binding is dynamically created between an IPv6 and an IPv4 address pool. Dynamic binding is triggered by the IPv6-to-IPv4 traffic and the address is dynamically allocated. Based on your configuration, you can have static or dynamic binding.
- The IPv4-initiated packet is protocol-translated and the destination IP address of the packet is set to IPv6 based on static or dynamic binding. The Stateful NAT64 translator translates the source IP address to IPv6 by using the Stateful NAT64 prefix (if a stateful prefix is configured) or the Well Known Prefix (WKP) (if a stateful prefix is not configured).
- A session is created based on the translation information.

All subsequent IPv4-initiated packets are translated based on the previously created session.

Stateful IPv6-to-IPv4 Packet Flow

The stateful IPv6-initiated packet flow is as follows:

- The first IPv6 packet is routed to the NAT Virtual Interface (NVI) based on the automatic routing setup that is configured for the stateful prefix. Stateful NAT64 performs a series of lookups to determine whether the IPv6 packet matches any of the configured mappings based on an access control list (ACL) lookup. Based on the mapping, an IPv4 address (and port) is associated with the IPv6 destination address. The IPv6 packet is translated and the IPv4 packet is formed by using the following methods:
  - Extracting the destination IPv4 address by stripping the prefix from the IPv6 address. The source address is replaced by the allocated IPv4 address (and port).
  - The rest of the fields are translated from IPv6-to-IPv4 to form a valid IPv4 packet.

  Note This protocol translation is the same for stateless NAT64 and described in the BEHAVE RFC draft.

- A new NAT64 translation is created in the session database and in the bind database. The pool and port databases are updated depending on the configuration. The return traffic and the subsequent traffic of the IPv6 packet flow will use this session database entry for translation.

IP Packet Filtering

Stateful Network Address Translation 64 (NAT64) filters IPv6 and IPv4 packets. All IPv6 packets that are transmitted into the stateful translator are filtered because statefully translated IPv6 packets consume resources in the translator. These packets consume processor resources for packet processing, memory resources (always session memory) for static configuration, IPv4 address resources for dynamic configuration, and IPv4 address and port resources for Port Address Translation (PAT).

Stateful NAT64 utilizes configured access control lists (ACLs) and prefix lists to filter IPv6-initiated traffic flows that are allowed to create the NAT64 state. Filtering of IPv6 packets is done in the IPv6-to-IPv4 direction because dynamic allocation of mapping between an IPv6 host and an IPv4 address can be done only in this direction.

Stateful NAT64 supports endpoint-dependent filtering for the IPv4-to-IPv6 packet flow with PAT configuration. In a Stateful NAT64 PAT configuration, the packet flow must have originated from the IPv6 realm and created the state information in NAT64 state tables. Packets from the IPv4 side that do not have a previously created state are dropped. Endpoint-independent filtering is supported with static Network Address Translation (NAT) and non-PAT configurations.

Differences Between Stateful NAT64 and Stateless NAT64

The table below displays the differences between Stateful NAT64 and Stateless NAT64.

Table 10 Differences Between Stateful NAT64 and Stateless NAT64

Supported Features	Stateful NAT64	Stateless NAT64
Address savings	*N*:1 mapping for PAT or overload configuration that saves IPv4 addresses.	One-to-one mapping—one IPv4 address is used for each IPv6 host).
Address space	IPv6 systems may use any type of IPv6 addresses.	IPv6 systems must have IPv4-translatable addresses (based on RFC 6052).
ALGs supported	FTP64	None
Protocols supported	ICMP, TCP, UDP	All
Standards	Draft-ieft-behave-v6v4-xlate-stateful-12	Draft-ietf-behave-v6v4-xlate-05
State creation	Each traffic flow creates a state in the NAT64 translator. The maximum number of states depends on the number of supported translations.	Traffic flow does not create any state in the NAT64 translator. Algorithmic operation is performed on the packet headers.

High-Speed Logging for NAT64

Depending on your release, Stateful NAT64 supports high-speed logging (HSL). When HSL is configured, NAT64 provides a log of packets that flow through routing devices (similar to the Version 9 NetFlow-like records) to an external collector. Records are sent for each binding (binding is the address binding between the local address and the global address to which the local address is translated) and when sessions are created and destroyed. Session records contain the full 5-tuple of information (the source IP address, destination IP address, source port, destination port, and protocol). A tuple is an ordered list of elements. NAT64 also sends an HSL message when a NAT64 pool runs out of addresses (also called pool exhaustion). Because the pool exhaustion messages are rate limited, each packet that hits the pool exhaustion condition does not trigger an HSL message.

Configure the **nat64 logging translations flow-export v9 udp destination** command to enable NAT64 HSL logging.

The table below describes the templates for HSL bind and session create or destroy. These fields (in the order they are displayed in the log) describe how the log collector must interpret the bytes in HSL records. The value for some of the fields varies based on whether the session is being created, destroyed, or modified.

Table 11 Templates for HSL Bind and Session Create or Destroy

Field	Format	ID	Value
Original IPv6 address	IPv6 address	27	Varies
Translated IPv4 address	IPv6 address	282	Varies
Translated IPv6 address	IPv4 address	225	Varies
Original IPv4 address	IPv4 address	12	Varies
Original IPv6 port	16-bit port	7	Varies
Translated IPv6 port	16-bit port	227	Varies
Translated IPv4 port	16-bit port	11	Varies
Original IPv4 port	16-bit port	228	Varies
Timestamp for an event	64 bits - milliseconds (This is a 64-bit field that holds the UNIX time, in milliseconds, when the event for the record occurred.)	323	Varies
VRF ID	32-bit ID	234	Zero
Protocol	8-bit value	4	Varies
Event	8-bit value	230	0–Invalid 1–Add event 2–Delete event

The table below describes the HSL pool exhaustion templates (in the order they are available in the template).

Table 12 Templates for HSL Pool Exhaustion

Field	Format	ID	Values
NAT pool ID	32-bit value	283	Varies
NAT event	8-bit value	230	3–Pool exhaust

FTP64 Application-Level Gateway Support

The FTP64 (or service FTP) application-level gateway (ALG) helps stateful Network Address Translation 64 (NAT64) to operate on Layer 7 data. FTP64 ALG translates IP addresses and the TCP port information embedded in the payload of an FTP control session.

NAT translates any TCP/UDP traffic that does not carry source and destination IP addresses in the application data stream. Protocols that embed the IP address information within the payload (or in the application data stream) require the support of an ALG. ALGs handle application data stream (Layer 7) protocol-specific services, such as translating embedded IP addresses and port numbers in the packet payload and extracting new connection or session information from control channels.

FTP64 is automatically enabled when Stateful NAT64 is enabled. Use the **no nat64 service ftp** command to disable the NAT64 FTP service.

Note The FTP64 ALG is not supported in Stateless NAT64 translation.

Note The FTP64 ALG does not support IPv4-compatible IPv6 addresses.

Based on *IPv6-to-IPv4 translation FTP considerations draft-ietf-behave-ftp64-02* and RFC 2228, the FTP64 ALG must switch to transparent mode (a device in a transparent mode is invisible in the network; however, this device can act as a bridge and inspect or filter packets), when commands and responses flow between the FTP client and the FTP server. When a client issues the FTP AUTH command, the FTP64 ALG transparently forwards all data on the control channel in both (ingress and egress) directions, until the end of the control channel session. Similarly, during an AUTH negotiation, the ALG must be in transparent mode, whether the negotiation is successful or not.

Based on RFC 6384, the behavior of the FTP64 ALG during a client-server communication is different. During an IPv6-to-IPv4 translation, the FTP64 ALG must transparently copy data transmitted over the control channel so that the transport layer security (TLS) session works correctly. However, the client commands and server responses are hidden from the FTP64 ALG. To ensure a consistent behavior, as soon as the initial FTP AUTH command is issued by a client, the FTP64 ALG must stop translating commands and responses and start transparently copying TCP data that is sent by the server to the client and vice versa. The FTP64 ALG must ignore the AUTH command and not go into transparent mode if the server response is in the 4*xx* or 5*xx* ranges, which comprise FTP error/warning messages.

Prior to CSCtu37975, when an IPv6 FTP client issues an FTP AUTH command, irrespective of whether the IPv4 FTP server accepts or rejects that authorization negotiation, the FTP64 ALG moves the AUTH session to transparent mode (or bypass mode). When a session is in transparent mode, NAT cannot perform translation on the packets within the session. With CSCtu37975, during a client-server communication, the FTP64 ALG's behavior is compliant with RFC 6384.

FTP64 NAT ALG Intrabox High Availability Support

Depending on your release, the FTP64 application-level gateway (ALG) adds high availability (HA) support for Stateful NAT64. The FTP64 NAT ALG Intrabox HA Support feature supports the stateful switchover between redundant Forward Processors (FPs) within a single chassis. The HA support provided by the FTP64 ALG is applicable to both intrabox HA and In-Service Software Upgrade (ISSU).

Use the **no nat64 service ftp** command to disable the NAT64 ALG service.

The FTP64 ALG synchronizes data when it receives the following messages:

- User authentication flag after 230 replies.
- ALG disable/enable flag after ALG ENABLE and ALG DISABLE messages are received.
- Fragment detection information after the first segmented packet is detected.
- Fragment detection information after the end of the segmentation is detected.

Note

- Stateful NAT64 supports only intrabox HA in some releases.
- FTP64 ALG statistics and FTP64 debug logs are not synchronized to the standby device by the FTP64 ALG.

Stateful NAT64—Intrachassis Redundancy

Depending on your release, support for the Stateful NAT64—Intrachassis Redundancy feature is available. When a second Forward Processor (FP) is available inside a single chassis, the Stateful NAT64—Intrachassis Redundancy feature enables you to configure the second FP as a standby entity. When you plug in the second FP, redundancy starts automatically with no explicit configuration. There is a short delay before the standby FP becomes the “hot standby” (which means that all sessions have been synchronized). The standby FP maintains a backup of the Stateful NAT64 session information, and when the active (first) FP fails, there is very little disruption of NAT64 sessions.

NAT64 redundancy information is sent to the standby FP in the following instances:

- When a session or a dynamic bind is created.
- When a session or a dynamic bind is deleted.
- During periodic updates. Based on the time elapsed, the active FP periodically updates the state information to the standby. Not all changes in the replicated objects are sent immediately to the standby at the time of change. The most critical updates are sent immediately, and other changes are communicated by periodic updates.

When a standby FP is inserted or when a standby FP recovers from a reload, the active FP performs a bulk synchronization to synchronize the standby FP with the active FP. NAT does an aggressive synchronization by which the active FP pushes all the state information forcefully to the standby FP.

In addition to NAT64 session information, application-specific information (application-level gateway [ALG] information) also has to be communicated to the standby FP. Each ALG has a per-session state that needs to be synchronized in the standby. The ALG triggers the sending of all ALG state information to the standby FP. NAT provides the mechanism for actually sending the ALG state and associates the state to a particular session.

HTTP sessions are not backed up on the standby FP. To replicate HTTP sessions on the standby FP during a switchover, you must configure the **nat64 switchover replicate http enable** command.

Note

The Stateful NAT64—Intrachassis Redundancy feature does not support box-to-box (B2B) redundancy or asymmetric routing.

How to Configure Stateful Network Address Translation 64

Based on your network configuration, you can configure static, dynamic, or dynamic Port Address Translation (PAT) Stateful NAT64.

Note You need to configure at least one of the configurations described in the following tasks for Stateful NAT64 to work.

- Configuring Static Stateful Network Address Translation 64, page 157
- Configuring Dynamic Stateful Network Address Translation 64, page 160
- Configuring Dynamic Port Address Translation Stateful NAT64, page 163
- Monitoring and Maintaining a Stateful NAT64 Routing Network, page 167

Configuring Static Stateful Network Address Translation 64

You can configure a static IPv6 address to an IPv4 address and vice versa. Optionally, you can configure static Stateful NAT64 with or without ports. Perform this task to configure static Stateful NAT64.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **description** *string*
6. **ipv6 enable**
7. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}
8. **nat64 enable**
9. **exit**
10. **interface** *type number*
11. **description** *string*
12. **ip address** *ip-address mask*
13. **nat64 enable**
14. **exit**
15. **nat64 prefix stateful** *ipv6-prefix/length*
16. **nat64 v6v4 static** *ipv6-address ipv4-address*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 5	**description** *string* **Example:** `Device(config-if)# description interface facing ipv6`	Adds a description to an interface configuration.
Step 6	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.

Step 7	**ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
	Example:
	`Device(config-if)# ipv6 address 2001:DB8:1::1/96`
		Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.

Step 8	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 translation on an IPv6 interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/2/0`	Configures an interface and enters interface configuration mode.
Step 11	**description** *string* **Example:** `Device(config-if)# description interface facing ipv4`	Adds a description to an interface configuration.
Step 12	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 209.165.201.1 255.255.255.0`	Configures an IPv4 address for an interface.
Step 13	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables NAT64 translation on an IPv4 interface.
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 15	**nat64 prefix stateful** *ipv6-prefix*/*length* **Example:** `Device(config)# nat64 prefix stateful 2001:DB8:1::1/96`	Defines the Stateful NAT64 prefix to be added to IPv4 hosts to translate the IPv4 address into an IPv6 address. • The Stateful NAT64 prefix can be configured at the global configuration level or at the interface level.
Step 16	**nat64 v6v4 static** *ipv6-address ipv4-address* **Example:** `Device(config)# nat64 v6v4 static 2001:DB8:1::FFFE 209.165.201.1`	Enables NAT64 IPv6-to-IPv4 static address mapping.

Step 17	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Configuring Dynamic Stateful Network Address Translation 64

A dynamic Stateful NAT64 configuration provides a one-to-one mapping of IPv6 addresses to IPv4 addresses in the address pool. You can use the dynamic Stateful NAT64 configuration when the number of active IPv6 hosts is less than the number of IPv4 addresses in the pool. Perform this task to configure dynamic Stateful NAT64.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **description** *string*
6. **ipv6 enable**
7. **ipv6** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}
8. **nat64 enable**
9. **exit**
10. **interface** *type number*
11. **description** *string*
12. **ip address** *ip-address mask*
13. **nat64 enable**
14. **exit**
15. **ipv6 access-list** *access-list-name*
16. **permit ipv6** *ipv6-address* **any**
17. **exit**
18. **nat64 prefix stateful** *ipv6-prefix/length*
19. **nat64 v4 pool** *pool-name start-ip-address end-ip-address*
20. **nat64 v6v4 list** *access-list-name* **pool** *pool-name*
21. **end**

DETAILED STEPS

	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	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 5	**description** *string* **Example:** `Device(config-if)# description interface facing ipv6`	Adds a description to an interface configuration.
Step 6	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.

Step 7	**ipv6** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}
	Example:
	`Device(config-if)# ipv6 2001:DB8:1::1/96`
		Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.

Step 8	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateful NAT64 translation on an IPv6 interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/2/0`	Configures an interface type and enters interface configuration mode

	Command or Action	Purpose
Step 11	**description** *string* **Example:** `Device(config-if)# description interface facing ipv4`	Adds a description to an interface configuration.
Step 12	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 209.165.201.24 255.255.255.0`	Configures an IPv4 address for an interface.
Step 13	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateful NAT64 translation on an IPv4 interface.
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 15	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list nat64-acl`	Defines an IPv6 access list and enters IPv6 access list configuration mode.
Step 16	**permit ipv6** *ipv6-address* **any** **Example:** `Device(config-ipv6-acl)# permit ipv6 2001:DB8:2::/96 any`	Sets permit conditions for an IPv6 access list.
Step 17	**exit** **Example:** `Device(config-ipv6-acl# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 18	**nat64 prefix stateful** *ipv6-prefix/length* **Example:** `Device(config)# nat64 prefix stateful 2001:DB8:1::1/96`	Enables NAT64 IPv6-to-IPv4 address mapping.

	Command or Action	Purpose
Step 19	**nat64 v4 pool** *pool-name start-ip-address end-ip-address* **Example:** `Device(config)# nat64 v4 pool pool1 209.165.201.1 209.165.201.254`	Defines the Stateful NAT64 IPv4 address pool.
Step 20	**nat64 v6v4 list** *access-list-name* **pool** *pool-name* **Example:** `Device(config)# nat64 v6v4 list nat64-acl pool pool1`	Dynamically translates an IPv6 source address to an IPv6 source address and an IPv6 destination address to an IPv4 destination address for NAT64.

Step 21	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Configuring Dynamic Port Address Translation Stateful NAT64

A Port Address Translation (PAT) or overload configuration is used to multiplex (mapping IPv6 addresses to a single IPv4 pool address) multiple IPv6 hosts to a pool of available IPv4 addresses on a first-come first-served basis. The dynamic PAT configuration conserves the IPv4 address space while providing connectivity to the IPv4 Internet. Configure the **nat64 v6v4 list** command with the **overload** keyword to configure PAT address translation. Perform this task to configure dynamic PAT Stateful NAT64.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **description** *string*
6. **ipv6 enable**
7. **ipv6** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
8. **nat64 enable**
9. **exit**
10. **interface** *type number*
11. **description** *string*
12. **ip address** *ip-address mask*
13. **nat64 enable**
14. **exit**
15. **ipv6 access-list** *access-list-name*
16. **permit ipv6** *ipv6-address* **any**
17. **exit**
18. **nat64 prefix stateful** *ipv6-prefix*/*length*
19. **nat64 v4 pool** *pool-name start-ip-address end-ip-address*
20. **nat64 v6v4 list** *access-list-name* **pool** *pool-name* **overload**
21. **end**

DETAILED STEPS

	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	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 5	**description** *string* **Example:** `Device(config-if)# description interface facing ipv6`	Adds a description to an interface configuration.
Step 6	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface.

Step 7	**ipv6** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
	Example:
	`Device(config-if)# ipv6 2001:DB8:1::1/96`
		Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.

Step 8	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateful NAT64 translation on an IPv6 interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/2/0`	Configures an interface type and enters interface configuration mode
Step 11	**description** *string* **Example:** `Device(config-if)# description interface facing ipv4`	Adds a description to an interface configuration.

	Command or Action	Purpose
Step 12	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 209.165.201.24 255.255.255.0`	Configures an IPv4 address for an interface.
Step 13	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateful NAT64 translation on an IPv6 interface.
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 15	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list nat64-acl`	Defines an IPv6 access list and places the device in IPv6 access list configuration mode.
Step 16	**permit ipv6** *ipv6-address* **any** **Example:** `Device(config-ipv6-acl)# permit ipv6 2001:db8:2::/96 any`	Sets permit conditions for an IPv6 access list.
Step 17	**exit** **Example:** `Device(config-ipv6-acl)# exit`	Exits IPv6 access list configuration mode and enters global configuration mode.
Step 18	**nat64 prefix stateful** *ipv6-prefix/length* **Example:** `Device(config)# nat64 prefix stateful 2001:db8:1::1/96`	Enables NAT64 IPv6-to-IPv4 address mapping.
Step 19	**nat64 v4 pool** *pool-name start-ip-address end-ip-address* **Example:** `Device(config)# nat64 v4 pool pool1 209.165.201.1 209.165.201.254`	Defines the Stateful NAT64 IPv4 address pool.

	Command or Action	Purpose
Step 20	**nat64 v6v4 list** *access-list-name* **pool** *pool-name* **overload** **Example:** `Device(config)# nat64 v6v4 list nat64-acl pool pool1 overload`	Enables NAT64 PAT or overload address translation.

Step 21	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Monitoring and Maintaining a Stateful NAT64 Routing Network

Use the following commands in any order to display the status of your Stateful Network Address Translation 64 (NAT64) configuration.

SUMMARY STEPS

1. **show nat64 aliases** [*lower-address-range upper-address-range*]
2. **show nat64 logging**
3. **show nat64 prefix stateful** {**global** | {**interfaces** | **static-routes**} [**prefix** *ipv6-address/prefix-length*]}
4. **show nat64 timeouts**

DETAILED STEPS

Step 1 **show nat64 aliases** [*lower-address-range upper-address-range*]

This command displays the IP aliases created by NAT64.

Example:

```
Device# show nat64 aliases

Aliases configured: 1
Address   Table ID  Inserted   Flags   Send ARP   Reconcilable   Stale   Ref-Count
10.1.1.1  0         FALSE      0x0030  FALSE      TRUE           FALSE   1
```

Step 2 **show nat64 logging**

This command displays NAT64 logging.

Example:

```
Device# show nat64 logging

NAT64 Logging Type

Method       Protocol   Dst. Address   Dst. Port   Src. Port
translation
flow export  UDP        10.1.1.1       5000        60087
```

Step 3 **show nat64 prefix stateful** {**global** | {**interfaces** | **static-routes**} [**prefix** *ipv6-address/prefix-length*]}

This command displays information about NAT64 stateful prefixes.

Example:

```
Device# show nat64 prefix stateful interfaces

Stateful Prefixes

Interface                NAT64   Enabled   Global Prefix
GigabitEthernet0/1/0     TRUE    TRUE      2001:DB8:1:1/96
GigabitEthernet0/1/3     TRUE    FALSE     2001:DB8:2:2/96
```

Step 4 **show nat64 timeouts**

This command displays statistics for NAT64 translation session timeout.

Example:

```
Device# show nat64 timeouts

NAT64 Timeout

Seconds    CLI Cfg    Uses 'All'    all flows
86400      FALSE      FALSE         udp
300        FALSE      TRUE          tcp
7200       FALSE      TRUE          tcp-transient
240        FALSE      FALSE         icmp
60         FALSE      TRUE
```

Configuration Examples for Stateful Network Address Translation 64

- Example: Configuring Static Stateful Network Address Translation 64, page 168
- Example: Configuring Dynamic Stateful Network Address Translation 64, page 169
- Example: Configuring Dynamic Port Address Translation Stateful NAT64, page 169

Example: Configuring Static Stateful Network Address Translation 64

```
Device# configure terminal
Device(config)# ipv6 unicast-routing
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# description interface facing ipv6
Device(config-if)# ipv6 enable
Device(config-if)# ipv6 address 2001:DB8:1::1/96
Device(confif-if)# nat64 enable
Device(config-fi)# exit
Device(config)# interface gigabitethernet 1/2/0
Device(config-if)# description interface facing ipv4
Device(config-if)# ip address 209.165.201.1 255.255.255.0
Device(config-if)# nat64 enable
Device(config-if)# exit
Device(config)# nat64 prefix stateful 2001:DB8:1::1/96
Device(config)# nat64 v6v4 static 2001:DB8:1::FFFE 209.165.201.1
Device(config)# end
```

Example: Configuring Dynamic Stateful Network Address Translation 64

```
Device# configure terminal
Device(config)# ipv6 unicast-routing
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# description interface facing ipv6
Device(config-if)# ipv6 enable
Device(config-if)# ipv6 2001:DB8:1::1/96
Device(config-if)# nat64 enable
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/2/0
Device(config-if)# description interface facing ipv4
Device(config-if)# ip address 209.165.201.24 255.255.255.0
Device(config-if)# nat64 enable
Device(config-if)# exit
Device(config)# ipv6 access-list nat64-acl
Device(config-ipv6-acl)# permit ipv6 2001:db8:2::/96 any
Device(config-ipv6-acl)# exit
Device(config)# nat64 prefix stateful 2001:db8:1::1/96
Device(config)# nat64 v4 pool pool1 209.165.201.1 209.165.201.254
Device(config)# nat64 v6v4 list nat64-acl pool pool1
Device(config)# end
```

Example: Configuring Dynamic Port Address Translation Stateful NAT64

```
enable
 configure terminal
  ipv6 unicast-routing
  interface gigabitethernet 0/0/0
   description interface facing ipv6
   ipv6 enable
   ipv6 2001:DB8:1::1/96
   nat64 enable
   exit
  interface gigabitethernet 1/2/0
   description interface facing ipv4
   ip address 209.165.201.24 255.255.255.0
   nat64 enable
   exit
  ipv6 access-list nat64-acl
   permit ipv6 2001:db8:2::/96 any
   exit
  nat64 prefix stateful 2001:db8:1::1/96
  nat64 v4 pool pool1 209.165.201.1 209.165.201.254
  nat64 v6v4 list nat64-acl pool pool1 overload
  end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Master Command List, All Releases*
NAT commands	*IP Addressing Services Command Reference*

Standards and RFCs

Standard/RFC	Title
Framework for IPv4/IPv6 Translation	*Framework for IPv4/IPv6 Translation draft-ietf-behave-v6v4-framework-06*
FTP ALG for IPv6-to-IPv4 translation	*An FTP ALG for IPv6-to-IPv4 translation draft-ietf-behave-ftp64-06*
IP/ICMP Translation Algorithm	*IP/ICMP Translation Algorithm draft-ietf-behave-v6v4-xlate-10*
IPv6 Addressing of IPv4/IPv6 Translators	*IPv6 Addressing of IPv4/IPv6 Translators draft-ietf-behave-address-format-07*
RFC 2228	*FTP Security Extensions*
RFC 2373	*IP Version 6 Addressing Architecture*
RFC 2464	*Transmission of IPv6 Packets over Ethernet Networks*
RFC 2765	*Stateless IP/ICMP Translation Algorithm (SIIT)*
RFC 2766	*Network Address Translation - Protocol Translation (NAT-PT)*
RFC 4787	*Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*
RFC 4966	*Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status*
RFC 6384	*An FTP Application Layer Gateway (ALG) for IPv6-to-IPv4 Translation*
Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers	*Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers draft-ietf-behave-v6v4-xlate-stateful-12*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Stateful Network Address Translation 64

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 13 Feature Information for Stateful Network Address Translation 64

Feature Name	Releases	Feature Information
FTP64 NAT ALG Intrabox HA Support	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, the FTP64 ALG adds HA support for Stateful NAT64. The FTP64 NAT ALG Intrabox HA Support feature supports the stateful switchover between redundant FPs within a single chassis. The HA support provided by the FTP64 ALG is applicable to both intrabox and interbox HA and In-Service Software Upgrade (ISSU).
Stateful NAT64 ALG—Stateful FTP64 ALG Support	Cisco IOS XE Release 3.4S	Cisco IOS XE Release 3.4S and later releases support FTP64 (or service FTP) ALGs. The FTP64 ALG helps Stateful NAT64 operate on Layer 7 data. An FTP ALG translates IP addresses and the TCP port information embedded in the payload of an FTP control session. The following commands were introduced or modified: **nat64 service ftp**.
Stateful NAT64—Intra-Chassis Redundancy	Cisco IOS XE Release 3.5S	Cisco IOS XE Release 3.5S and later releases support the Stateful NAT64—Intra-Chassis Redundancy feature. When a second Forward Processor (FP) is available inside a single chassis, the Stateful NAT64 Intra-Chassis Redundancy feature enables you to configure the second FP as a standby entity. The standby FP maintains a backup of the stateful NAT64 session information and when the active (first) FP fails, there is no disruption of NAT64 sessions. The following commands were introduced or modified: **nat64 switchover replicate http port**.

Feature Name	Releases	Feature Information
Stateful Network Address Translation 64	Cisco IOS XE Release 3.4S	The Stateful Network Address Translation 64 feature provides a translation mechanism that translates IPv6 packets into IPv4 packets and vice versa. The Stateful NAT64 translator, algorithmically translates the IPv4 addresses of IPv4 hosts to and from IPv6 addresses by using the configured stateful prefix. In a similar manner, the IPv6 addresses of IPv6 hosts are translated to and from IPv4 addresses through NAT. The following commands were introduced or modified: **clear nat64 statistics**, **debug nat64**, **nat64 logging**, **nat64 prefix stateful**, **nat64 translation**, **nat64 v4**, **nat64 v4v6**, **nat64 v6v4**, **show nat64 aliases**, **show nat64 limits**, **show nat64 logging**, **show nat64 mappings dynamic**, **show nat64 mappings static**, **show nat64 services**, **show nat64 pools**, **show nat64 prefix stateful**, **show nat64 statistics**, **show nat64 timeouts**, and **show nat64 translations**.

Glossary

ALG—application-layer gateway or application-level gateway.

FP—Forward Processor.

IPv4-converted address—IPv6 addresses used to represent the IPv4 hosts. These have an explicit mapping relationship to the IPv4 addresses. This relationship is self-described by mapping the IPv4 address in the IPv6 address. Both stateless and stateful translators use IPv4-converted IPv6 addresses to represent the IPv4 hosts.

IPv6-converted address—IPv6 addresses that are assigned to the IPv6 hosts for the stateless translator. These IPv6-converted addresses have an explicit mapping relationship to the IPv4 addresses. This relationship is self-described by mapping the IPv4 address in the IPv6 address. The stateless translator uses the corresponding IPv4 addresses to represent the IPv6 hosts. The stateful translator does not use IPv6-converted addresses, because the IPv6 hosts are represented by the IPv4 address pool in the translator via dynamic states.

NAT—Network Address Translation.

RP—Route Processor.

stateful translation—In stateful translation a per-flow state is created when the first packet in a flow is received. A translation algorithm is said to be stateful if the transmission or reception of a packet creates or modifies a data structure in the relevant network element. Stateful translation allows the use of multiple translators interchangeably and also some level of scalability. Stateful translation is defined to enable the IPv6 clients and peers without mapped IPv4 addresses to connect to the IPv4-only servers and peers.

stateless translation—A translation algorithm that is not stateful is called stateless. A stateless translation requires configuring a static translation table, or may derive information algorithmically from the messages it is translating. Stateless translation requires less computational overhead than stateful translation. It also requires less memory to maintain the state, because the translation tables and the associated methods and processes exist in a stateful algorithm and do not exist in a stateless one. Stateless translation enables the IPv4-only clients and peers to initiate connections to the IPv6-only servers or peers that are equipped with IPv4-embedded IPv6 addresses. It also enables scalable coordination of IPv4-only stub networks or ISP IPv6-only networks. Because the source port in an IPv6-to-IPv4 translation may have to be changed to provide adequate flow identification, the source port in the IPv4-to-IPv6 direction need not be changed.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

The Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT feature supports the forwarding of packets from a standby redundancy group to the active redundancy group for packet handling. If this feature is not enabled, the return TCP packets forwarded to the router that did not receive the initial synchronization (SYN) message are dropped because they do not belong to any known existing session.

This module provides an overview of asymmetric routing and describes how to configure asymmetric routing

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

- Asymmetric routing over Multiprotocol Label Switching (MPLS) and VPN is not supported.
- LANs that use virtual IP addresses and virtual MAC (VMAC) addresses do not support asymmetric routing.
- VPN routing and forwarding (VRF) is not supported.

Information About Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

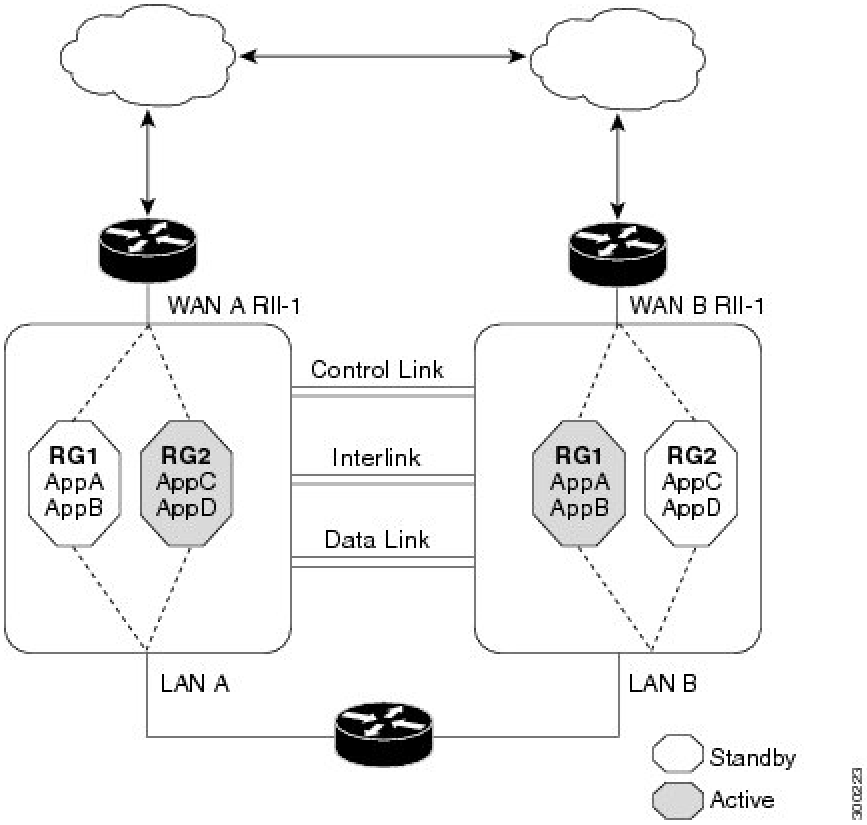
Asymmetric Routing Overview

Asymmetric routing occurs when packets from TCP or UDP connections flow in different directions through different routes. In asymmetric routing, packets that belong to a single TCP or UDP connection are forwarded through one interface in a redundancy group (RG), but returned through another interface in the same RG. In asymmetric routing, the packet flow remains in the same RG. When you configure asymmetric routing, packets received on the standby RG are redirected to the active RG for processing. If asymmetric routing is not configured, the packets received on the standby RG may be dropped.

Asymmetric routing determines the RG for a particular traffic flow. The state of the RG is critical in determining the handling of packets. If an RG is active, normal packet processing is performed. In case the RG is in a standby state and you have configured asymmetric routing and the **asymmetric-routing always-divert enable** command, packets are diverted to the active RG. Use the **asymmetric-routing always-divert enable** command to always divert packets received from the standby RG to the active RG.

The figure below shows an asymmetric routing scenario with a separate asymmetric-routing interlink interface to divert packets to the active RG.

Figure 12 ***Asymmetric Routing Scenario***



The following rules apply to asymmetric routing:

- 1:1 mapping exists between the redundancy interface identifier (RII) and the interface.
- 1:*n* mapping exists between the interface and an RG. (An interface can have multiple RGs.)
- 1:*n* mapping exists between an RG and applications that use it. (Multiple applications can use the same RG).
- 1:1 mapping exists between an RG and the traffic flow. The traffic flow must map only to a single RG. If a traffic flow maps to multiple RGs, an error occurs.
- 1:1 or 1:*n* mapping can exist between an RG and an asymmetric-routing interlink as long as the interlink has sufficient bandwidth to support all the RG interlink traffic.

Asymmetric routing consists of an interlink interface that handles all traffic that is to be diverted. The bandwidth of the asymmetric-routing interlink interface must be large enough to handle all expected traffic that is to be diverted. An IPv4 address must be configured on the asymmetric-routing interlink interface, and the IP address of the asymmetric routing interface must be reachable from this interface.

Note We recommend that the asymmetric-routing interlink interface be used for interlink traffic only and not be shared with high availability (HA) control or data interfaces because the amount of traffic on the asymmetric-routing interlink interface could be quite high.

Asymmetric Routing Support in Firewalls

For intrabox asymmetric routing support, the firewall does a stateful Layer 3 and Layer 4 inspection of Internet Control Message Protocol (ICMP), TCP, and UDP packets. The firewall does a stateful inspection of TCP packets by verifying the window size and order of packets. The firewall also requires the state information from both directions of the traffic for stateful inspection. The firewall does a limited inspection of ICMP information flows. It verifies the sequence number associated with the ICMP echo request and response. The firewall does not synchronize any packet flows to the standby redundancy group (RG) until a session is established for that packet. An established session is a three-way handshake for TCP, the second packet for UDP, and informational messages for ICMP. All ICMP flows are sent to the active RG.

The firewall does a stateless verification of policies for packets that do not belong to the ICMP, TCP, and UDP protocols.

The firewall depends on bidirectional traffic to determine when a packet flow should be aged out and diverts all inspected packet flows to the active RG. Packet flows that have a pass policy and that include the same zone with no policy or a drop policy are not diverted.

Note The firewall does not support the **asymmetric-routing always-divert enable** command that diverts packets received on the standby RG to the active RG. By default, the firewall forces all packet flows to be diverted to the active RG.

Asymmetric Routing in NAT

By default, when asymmetric routing is configured, Network Address Translation (NAT) processes non-ALG packets on the standby RG, instead of forwarding them to the active. The NAT-only configuration (that is when the firewall is not configured) can use both the active and standby RGs for processing packets. If you have a NAT-only configuration and you have configured asymmetric routing, the default asymmetric routing rule is that NAT will selectively process packets on the standby RG. You can configure the **asymmetric-routing always-divert enable** command to divert packets received on the standby RG to the active RG. Alternatively, if you have configured the firewall along with NAT, the default asymmetric routing rule is to always divert the packets to the active RG.

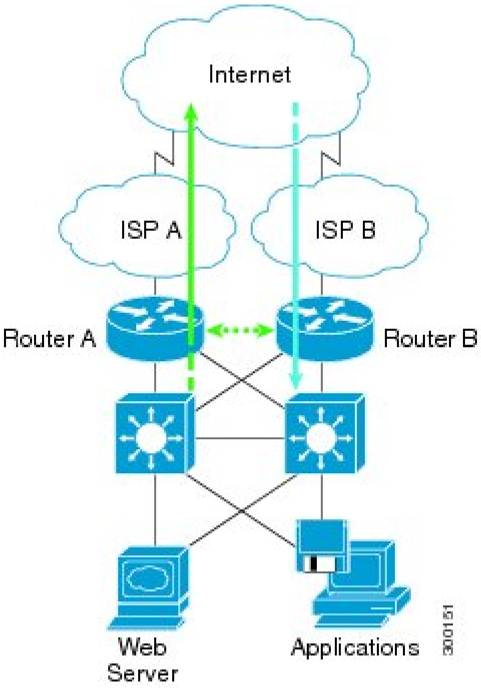
When NAT receives a packet on the standby RG and if you have not configured the diverting of packets, NAT does a lookup to see if a session exists for that packet. If a session exists and there is no ALG associated for that session, NAT processes the packet on the standby RG. The processing of packets on the standby RG when a session exists significantly increases the bandwidth of the NAT traffic.

ALGs are used by NAT to identify and translate payload and to create child flows. ALGs require a two-way traffic to function correctly. NAT must divert all traffic to the active RG for any packet flow that is associated with an ALG. This is accomplished by checking if ALG data that is associated with the session is found on the standby RG. If ALG data exits, the packet is diverted for asymmetric routing.

Asymmetric Routing in a WAN-LAN Topology

Asymmetric routing supports only a WAN-LAN topology. In a WAN-LAN topology, devices are connected through LAN interfaces on the inside and WAN interfaces on the outside. There is no control on the routing of return traffic received through WAN links. Asymmetric routing controls the routing of return traffic received through WAN links in a WAN-LAN topology. The figure below shows a WAN-LAN topology.

Figure 13 Asymmetric Routing in a WAN-LAN Topology



How to Configure Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

- Configuring a Redundancy Application Group and a Redundancy Group Protocol, page 179
- Configuring Data, Control, and Asymmetric Routing Interfaces, page 182
- Configuring a Redundant Interface Identifier and Asymmetric Routing on an Interface, page 185
- Configuring Dynamic Inside Source Translation with Asymmetric Routing, page 186

Configuring a Redundancy Application Group and a Redundancy Group Protocol

Redundancy groups consist of the following configuration elements:

- The amount by which the priority will be decremented for each object.

- Faults (objects) that decrement the priority
- Failover priority
- Failover threshold
- Group instance
- Group name
- Initialization delay timer

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **name** *group-name*
7. **priority** *value* [**failover threshold** *value*]
8. **preempt**
9. **track** *object-number* **decrement** *number*
10. **exit**
11. **protocol** *id*
12. **timers hellotime** {*seconds* | **msec** *msec*} **holdtime** {*seconds* | **msec** *msec*}
13. **authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key* [**timeout** *seconds*] | **key-chain** *key-chain-name*}
14. **bfd**
15. **end**

DETAILED STEPS

	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	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.

	Command or Action	Purpose
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a redundancy group and enters redundancy application group configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name group1`	Specifies an optional alias for the protocol instance.
Step 7	**priority** *value* [**failover threshold** *value*] **Example:** `Device(config-red-app-grp)# priority 100 failover threshold 50`	Specifies the initial priority and failover threshold for a redundancy group.
Step 8	**preempt** **Example:** `Device(config-red-app-grp)# preempt`	Enables preemption on the redundancy group and enables the standby device to preempt the active device. • The standby device preempts only when its priority is higher than that of the active device.
Step 9	**track** *object-number* **decrement** *number* **Example:** `Device(config-red-app-grp)# track 50 decrement 50`	Specifies the priority value of a redundancy group that will be decremented if an event occurs on the tracked object.
Step 10	**exit** **Example:** `Device(config-red-app-grp)# exit`	Exits redundancy application group configuration mode and enters redundancy application configuration mode.
Step 11	**protocol** *id* **Example:** `Device(config-red-app)# protocol 1`	Specifies the protocol instance that will be attached to a control interface and enters redundancy application protocol configuration mode.

	Command or Action	Purpose

Step 12	**timers hellotime** {*seconds* | **msec** *msec*} **holdtime** {*seconds* | **msec** *msec*}
	Example:
	`Device(config-red-app-prtcl)# timers hellotime 3 holdtime 10`
		Specifies the interval between hello messages sent and the time period before which a device is declared to be down.
		- Holdtime should be at least three times the hellotime.

Step 13	**authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key* [**timeout** *seconds*] | **key-chain** *key-chain-name*}
	Example:
	`Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100`
		Specifies authentication information.

Step 14	**bfd** **Example:** `Device(config-red-app-prtcl)# bfd`	Enables the integration of the failover protocol running on the control interface with the Bidirectional Forwarding Detection (BFD) protocol to achieve failure detection in milliseconds. - BFD is enabled by default.

Step 15	**end**
	Example:
	`Device(config-red-app-prtcl)# end`
		Exits redundancy application protocol configuration mode and enters privileged EXEC mode.

Configuring Data, Control, and Asymmetric Routing Interfaces

In this task, you configure the following redundancy group (RG) elements:

- The interface that is used as the control interface.
- The interface that is used as the data interface.
- The interface that is used for asymmetric routing.

Note Asymmetric routing, data, and control must be configured on separate interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **data** *interface-type interface-number*
7. **control** *interface-type interface-number* **protocol** *id*
8. **timers delay** *seconds* [**reload** *seconds*]
9. **asymmetric-routing interface** *type number*
10. **asymmetric-routing always-divert enable**
11. **end**

DETAILED STEPS

	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	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a redundancy group (RG) and enters redundancy application group configuration mode.

	Command or Action	Purpose
Step 6	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data GigabitEthernet 0/0/0`	Specifies the data interface that is used by the RG.
Step 7	**control** *interface-type interface-number* **protocol** *id* **Example:** `Device(config-red-app-grp)# control GigabitEthernet 0/0/2 protocol 1`	Specifies the control interface that is used by the RG. • The control interface is also associated with an instance of the control interface protocol.
Step 8	**timers delay** *seconds* [**reload** *seconds*] **Example:** `Device(config-red-app-grp)# timers delay 100 reload 400`	Specifies the time required for an RG to delay role negotiations that start after a fault occurs or the system is reloaded.
Step 9	**asymmetric-routing interface** *type number* **Example:** `Device(config-red-app-grp)# asymmetric-routing interface GigabitEthernet 0/1/1`	Specifies the asymmetric routing interface that is used by the RG.
Step 10	**asymmetric-routing always-divert enable** **Example:** `Device(config-red-app-grp)# asymmetric-routing always-divert enable`	Always diverts packets received from the standby RG to the active RG.

Step 11	**end**
	Example:
	`Device(config-red-app-grp)# end`
		Exits redundancy application group configuration mode and enters privileged EXEC mode.

Configuring a Redundant Interface Identifier and Asymmetric Routing on an Interface

Note

- You must not configure a redundant interface identifier (RII) on an interface that is configured either as a data interface or as a control interface.
- You must configure the RII and asymmetric routing on both active and standby devices.
- You cannot enable asymmetric routing on the interface that has a virtual IP address configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **redundancy rii** *id*
5. **redundancy group** *id* [**decrement** *number*]
6. **redundancy asymmetric-routing enable**
7. **end**

DETAILED STEPS

	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 GigabitEthernet 0/1/3`	Selects an interface to be associated with the redundancy group (RG) and enters interface configuration mode.
Step 4	**redundancy rii** *id* **Example:** `Device(config-if)# redundancy rii 600`	Configures the redundancy interface identifier (RII).

	Command or Action	Purpose
Step 5	**redundancy group** *id* [**decrement** *number*] **Example:** `Device(config-if)# redundancy group 1 decrement 20`	Enables the RG redundancy traffic interface configuration and specifies the amount to be decremented from the priority when the interface goes down. **Note** You need not configure an RG on the traffic interface on which asymmetric routing is enabled.
Step 6	**redundancy asymmetric-routing enable** **Example:** `Device(config-if)# redundancy asymmetric-routing enable`	Establishes an asymmetric flow diversion tunnel for each RG.

Step 7	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and enters privileged EXEC mode.

Configuring Dynamic Inside Source Translation with Asymmetric Routing

The following configuration is a sample dynamic inside source translation with asymmetric routing. You can configure asymmetric routing with the following types of NAT configurations—dynamic outside source, static inside and outside source, and Port Address Translation (PAT) inside and outside source translations. For more information on different types of NAT configurations, see the “Configuring NAT for IP Address Conservation” chapter.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ip nat outside**
6. **exit**
7. **redundancy**
8. **application redundancy**
9. **group** *id*
10. **asymmetric-routing always-divert enable**
11. **end**
12. **configure terminal**
13. **ip nat pool** *name start-ip end-ip* {*mask* | **prefix-length** *prefix-length*}
14. **exit**
15. **ip nat inside source list** *acl-number* **pool** *name* **redundancy** *redundancy-id* **mapping-id** *map-id*
16. **access-list** *standard-acl-number* **permit** *source-address wildcard-bits*
17. **end**

DETAILED STEPS

	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 gigabitethernet 0/1/3`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary IP address for an interface.

	Command or Action	Purpose
Step 5	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 7	**redundancy** **Example:** `Device(config)# redundancy`	Configures redundancy and enters redundancy configuration mode.
Step 8	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Configures application redundancy and enters redundancy application configuration mode.
Step 9	**group** *id* **Example:** `Device(config-red-app)# group 1`	Configures a redundancy group and enters redundancy application group configuration mode.
Step 10	**asymmetric-routing always-divert enable** **Example:** `Device(config-red-app-grp)# asymmetric-routing always-divert enable`	Diverts the traffic to the active device.

Step 11	**end**
	Example:
	`Device(config-red-app-grp)# end`
		Exits redundancy application group configuration mode and enters privileged EXEC mode.

Step 12	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose

Step 13	**ip nat pool** *name start-ip end-ip* {*mask* | **prefix-length** *prefix-length*}
	Example:
	`Device(config)# ip nat pool pool1 prefix-length 24`
		Defines a pool of global addresses.
		• Enters IP NAT pool configuration mode.

Step 14	**exit** **Example:** `Device(config-ipnat-pool)# exit`	Exits IP NAT pool configuration mode and enters global configuration mode.
Step 15	**ip nat inside source list** *acl-number* **pool** *name* **redundancy** *redundancy-id* **mapping-id** *map-id* **Example:** `Device(config)# ip nat inside source list pool pool1 redundancy 1 mapping-id 100`	Enables NAT of the inside source address and associates NAT with a redundancy group by using the mapping ID.
Step 16	**access-list** *standard-acl-number* **permit** *source-address wildcard-bits* **Example:** `Device(config)# access-list 10 permit 10.1.1.1 255.255.255.0`	Defines a standard access list for the inside addresses that are to be translated.

Step 17	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

Example: Configuring a Redundancy Application Group and a Redundancy Group Protocol

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name group1
Device(config-red-app-grp)# priority 100 failover threshold 50
Device(config-red-app-grp)# preempt
Device(config-red-app-grp)# track 50 decrement 50
Device(config-red-app-grp)# exit
Device(config-red-app)# protocol 1
Device(config-red-app-prtcl)# timers hellotime 3 holdtime 10
Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100
Device(config-red-app-prtcl)# bfd
Device(config-red-app-prtcl)# end
```

Example: Configuring Data, Control, and Asymmetric Routing Interfaces

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# data GigabitEthernet 0/0/0
Device(config-red-app-grp)# control GigabitEthernet 0/0/2 protocol 1
Device(config-red-app-grp)# timers delay 100 reload 400
Device(config-red-app-grp)# asymmetric-routing interface GigabitEthernet 0/1/1
Device(config-red-app-grp)# asymmetric-routing always-divert enable
Device(config-red-app-grp)# end
```

Example: Configuring a Redundant Interface Identifier and Asymmetric Routing on an Interface

```
Device# configure terminal
Device(config)# interface GigabitEthernet 0/1/3
Device(config-if)# redundancy rii 600
Device(config-if)# redundancy group 1 decrement 20
Device(config-if)# redundancy asymmetric-routing enable
Device(config-if)# end
```

Example: Configuring Dynamic Inside Source Translation with Asymmetric Routing

```
Device(config)# interface gigabitethernet 0/1/3
Device(config-if)# ip address 10.1.1.1 255.255.255.0
Device(config-if)# ip nat outside
Device(config-if)# exit
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# asymmetric-routing always-divert enable
Device(config-red-app-grp)# end
Device# configure terminal
Device(config)# ip nat pool pool1 prefix-length 24
Device(config-ipnat-pool)# exit
Device(config)# ip nat inside source list pool pool1 redundancy 1 mapping-id 100
Device(config)# access-list 10 permit 10.1.1.1 255.255.255.0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
Security commands	- *Cisco IOS Security Command Reference Commands A to C* - *Cisco IOS Security Command Reference Commands D to L* - *Cisco IOS Security Command Reference Commands M to R* - *Cisco IOS Security Command Reference Commands S to Z*
Firewall inter-chassis redundancy	"Configuring Firewall Stateful Inter-Chassis Redundancy" module
NAT inter-chassis redundancy	"Configuring Stateful Inter-Chassis Redundancy" module

Standards and RFCs

Standard/RFC	Title
No new or modified standards or RFCs are supported by this feature, and support for existing standards or RFCs has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 14 Feature Information for Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT

Feature Name	Releases	Feature Information
Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT	Cisco IOS XE Release 3.5S	The Interchassis Asymmetric Routing Support for Zone-Based Firewall and NAT feature supports the forwarding of packets from a standby redundancy group to the active redundancy group for packet handling. The following commands were introduced or modified: **asymmetric-routing**, **redundancy asymmetric-routing enable**.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

IP Multicast Dynamic NAT

The IP Multicast Dynamic Network Address Translation (NAT) feature supports the source address translation of multicast packets. You can use source address translation when you want to connect to the Internet, but not all your hosts have globally unique IP addresses. NAT translates the internal local addresses to globally unique IP addresses before sending packets to the outside network. The IP multicast dynamic translation establishes a one-to-one mapping between an inside local address and one of the addresses from the pool of outside global addresses.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for IP Multicast Dynamic NAT

The IP Multicast Dynamic NAT feature does not support:

- IPv4-to-IPv6 address translation.
- Multicast destination address translation.
- Port Address Translation (PAT) overloading for multicast.
- Source and destination address translation.
- Unicast-to-multicast address translation.

Information About IP Multicast Dynamic NAT

How NAT Works

A router configured with NAT will have at least one interface to the inside network and one to the outside network. In a typical environment, NAT is configured at the exit router between a stub domain and a backbone. When a packet leaves 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 and sends an Internet Control Message Protocol (ICMP) host unreachable packet.

Uses of NAT

NAT can be used for the following applications:

- When you want to connect to the Internet, but not all of 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 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 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 they are no longer in use.
- When you must 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.
- When 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.

NAT Inside and Outside Addresses

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

Similarly, *outside* refers to those networks to which the stub network connects, and which are generally not under the control of the organization. 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--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 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. The address is not necessarily legitimate; it was allocated from the address space 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 was allocated from a globally routable address or network space.

Dynamic Translation of 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 need to access the Internet. The dynamically configured pool IP address may be used as needed and 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 router, the router will add IP aliases for them so that it can answer Address Resolution Protocol (ARP) requests. However, a situation can arise where the router itself answers packets that are not destined for it, possibly causing a security issue. This can happen when an incoming Internet Control Message Protocol (ICMP) or UDP packet that is destined for one of the aliased addresses does not have a corresponding NAT translation in the NAT table, and the router itself runs a corresponding service, for example, the Network Time Protocol (NTP). Such a situation might cause minor security risks.

How to Configure IP Multicast Dynamic NAT

Configuring IP Multicast Dynamic NAT

Note IP multicast dynamic translation establishes a one-to-one mapping between an inside local address and one of the addresses from the pool of outside global addresses

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} [**type** {**match-host** | **rotary**}]
4. **access-list** *access-list-number* **permit** *source-address wildcard-bits* [**any**]
5. **ip nat inside source list** *access-list-number* **pool** *name*
6. **ip multicast-routing distributed**
7. **interface** *type number*
8. **ip address** *ip-address mask*
9. **ip pim sparse-mode**
10. **ip nat inside**
11. **exit**
12. **interface** *type number*
13. **ip address** *ip-address mask*
14. **ip pim sparse-mode**
15. **ip nat outside**
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} [**type** {**match-host** | **rotary**}] **Example:** `Router(config)# ip nat pool mypool 10.41.10.1 10.41.10.23 netmask 255.255.255.0`	Defines a pool of global addresses to be allocated as needed.

	Command or Action	Purpose
Step 4	**access-list** *access-list-number* **permit** *source-address wildcard-bits* [**any**] **Example:** `Router(config)# access-list 100 permit 10.3.2.0 0.0.0.255 any`	Defines a standard access list for the inside addresses that are to be translated.
Step 5	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Router(config)# ip nat inside source list 100 pool mypool`	Establishes dynamic source translation, specifying the access list defined in the prior step.
Step 6	**ip multicast-routing distributed** **Example:** `Router(config)# ip multicast-routing distributed`	Enables Multicast Distributed Switching (MDS).
Step 7	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 8	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 9	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables sparse mode operation of Protocol Independent Multicast (PIM) on an interface.
Step 10	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Indicates that the interface is connected to the inside network (the network that is subject to NAT translation).
Step 11	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 12	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/1`	Configures an interface and enters interface configuration mode.
Step 13	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.2.2.1 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 14	**ip pim sparse-mode** **Example:** `Router(config-if)# ip pim sparse-mode`	Enables sparse mode operation of PIM on an interface.
Step 15	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Indicates that the interface is connected to the outside network.

Step 16	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and enters privileged EXEC mode.

Configuration Examples for IP Multicast Dynamic NAT

Example: Configuring IP Multicast Dynamic NAT

```
Router# configure terminal
Router(config)# ip nat pool mypool 10.41.10.1 10.41.10.23 netmask 255.255.255.0
Router(config)# access-list 100 permit 10.3.2.0 0.0.0.255 any
Router(config)# ip nat inside source list 100 pool mypool
Router(config)# ip multicast-routing distributed
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# ip address 10.0.0.1 255.255.255.0
Router(config-if)# ip pim sparse-mode
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# interface gigabitethernet 0/0/1
Router(config-if)# ip address 10.2.2.1 255.255.255.0
Router(config-if)# ip pim sparse-mode
Router(config-if)# ip nat outside
Router(config-if)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
NAT commands	*Cisco IOS IP Addressing Services Command Reference*
Configuring NAT for IP address conservation	*Configuring NAT for IP Address Conservation* module

Standards and RFCs

Standard/RFC	Title
None	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IP Multicast Dynamic NAT

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 15 Feature Information for IP Multicast Dynamic NAT

Feature Name	Releases	Feature Information
IP Multicast Dynamic NAT	Cisco IOS XE Release 3.4S	The IP Multicast Dynamic Network Address Translation feature supports the source address translation of multicast packets. You can use source address translation when you want to connect to the Internet, but not all your hosts have globally unique IP addresses. NAT translates the internal local addresses to globally unique IP addresses before sending packets to the outside network. The IP multicast dynamic translation establishes a one-to-one mapping between an inside local address and one of the addresses from the pool of outside global addresses.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Match-in-VRF Support for NAT

The Match-in-VRF Support for NAT feature supports Network Address Translation (NAT) of packets that communicate between two hosts within the same VPN routing and forwarding (VRF) instance. In intra-VPN NAT, both the local and global address spaces for end hosts are isolated to their respective VPNs, and as a result, the translated addresses for the hosts overlap each other. The Match-in-VRF Support for NAT feature helps separate the address space for translated addresses among VPNs.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Match-in-VRF Support for NAT

The Match-in-VRF Support for NAT feature is not supported on interface overload configuration.

Information About Match-in-VRF Support for NAT

Match-in-VRF Support for NAT

In Cisco IOS XE Release 3.5S and later releases, the Match-in-VRF Support for NAT feature supports NAT of packets that communicate between two hosts within the same VPN.

The VRF-aware NAT enables communication between hosts in the private address space in different VPN routing and forwarding (VRF) instances and common servers in the Internet or the global domain. Because IP addresses of the inside hosts overlap with each other, the VRF-aware NAT facilitates communication between these hosts by converting overlapped inside IP addresses into globally unique addresses. The Match-in-VRF Support for NAT feature extends VRF-aware NAT by supporting intra-VPN NAT capability. In the intra-VPN NAT, both the local and global address spaces for end hosts are isolated to their respective VPNs, and as a result translated addresses for hosts overlap each other. To separate the address space for translated addresses among VPNs, configure the **match-in-vrf** keyword in the NAT mapping (**ip nat inside source** command) configuration. Both static and dynamic NAT configurations support the **match-in-vrf** keyword.

Note All NAT commands that support VRF support the **match-in-vrf** keyword. Because NAT outside rules (**ip nat outside source** command) support the match-in-VRF functionality by default, the **match-in-vrf** keyword is not supported by NAT outside rules.

In VRF-aware NAT, the IP alias and Address Resolution Protocol (ARP) entries for inside global addresses are configured in the global domain. For intra-VPN NAT, the IP alias and ARP entries for inside global addresses are configured in the VRF through which the translation happens. In intra-VPN NAT, configuration of the **match-in-vrf** keyword implies that at least one NAT outside interface is configured in the same VRF. The ARP entry in that VRF replies to the ARP request from the outside host.

If inside addresses are configured, the match-in-VRF is determined through inside mappings during the address translation of VRF traffic. If you have configured only outside mapping of IP addresses for address translations, the match-in-VRF will work. When a translation entry is created with both inside and outside mappings, the **match-in-vrf** keyword is determined by the inside mapping.

The Match-in-VRF Support for NAT feature supports the configuration of multiple dynamic mappings with the same IP address pool.

How to Configure Match-in-VRF Support for NAT

Configuring Static NAT with Match-in-VRF

Perform the following task to configure a static NAT translation and to enable NAT inside and outside traffic in the same VRF.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* [**vrf** *vrf-name* [**match-in-vrf**]] **Example:** `Router(config)# ip nat inside source static 10.10.10.1 172.16.131.1 vrf vrf1 match-in-vrf`	Establishes static translation between an inside local address and an inside global address. • The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF.
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for an interface.
Step 6	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 7	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vrf1`	Associates a VRF with an interface or subinterface.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies a different interface and enters interface configuration mode.
Step 10	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 11	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside. **Note** NAT outside rules support the match-in-VRF functionality by default.
Step 12	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vrf1`	Associates a VRF with an interface or subinterface.

	Command or Action	Purpose

Step 13	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Dynamic NAT with Match-in-VRF

Perform the following task to configure a dynamic NAT translation with the same address pool and to enable NAT inside and outside traffic in the same VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source list** *access-list-number* **pool** *pool-name* [**vrf** *vrf-name* [**match-in-vrf**]]
4. **access-list** *access-list-number* **permit source** [*source-wildcard*]
5. **ip nat inside source list** *access-list-number* **pool** *pool-name* **vrf** *vrf-name* [**match-in-vrf**]
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **ip nat inside**
9. **ip vrf forwarding** *vrf-name*
10. **exit**
11. **interface** *type number*
12. **ip address** *ip-address mask*
13. **ip nat outside**
14. **ip vrf forwarding** *vrf-name*
15. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip nat inside source list** *access-list-number* **pool** *pool-name* [**vrf** *vrf-name* [**match-in-vrf**]] **Example:** `Router(config)# ip nat inside source list 1 pool shared-pool vrf vrf1 match-in-vrf`	Enables multiple dynamic mappings to be configured with the same address pool. • The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF.
Step 4	**access-list** *access-list-number* **permit source** [*source-wildcard*] **Example:** `Router(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** *pool-name* **vrf** *vrf-name* [**match-in-vrf**] **Example:** `Router(config)# ip nat inside source list 1 pool shared-pool vrf vpn1`	Establishes dynamic source translation, specifying the access list defined in the previous step.
Step 6	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/1`	Specifies an interface and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 8	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 9	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vpn1`	Associates a VRF with an interface or subinterface.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 11	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0/0`	Specifies a different interface and enters interface configuration mode.
Step 12	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 13	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Marks the interface as connected to the outside. **Note** NAT outside rules support the match-in-VRF functionality by default.
Step 14	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vpn1`	Associates a VRF with an interface or subinterface.

Step 15	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and returns to global configuration mode.

Configuration Examples for Match-in-VRF Support for NAT

Example: Configuring Static NAT with Match-in-VRF

The following example shows how to configure a static NAT translation between the local IP address 10.10.10.1 and the global IP address 172.16.131.1. The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF.

```
Router# configure terminal
Router(config)# ip nat inside source static 10.10.10.1 172.16.131.1 vrf vrf1 match-in-vrf
Router(config)# interface gigabitethernet 0/0/1
Router(config-if)# ip address 10.114.11.39 255.255.255.0
Router(config-if)# ip nat inside
Router(config-if)# ip vrf forwarding vrf1
Router(config-if)# exit
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# ip address 172.31.232.182 255.255.255.240
Router(config-if)# ip nat outside
Router(config-if)# ip vrf forwarding vrf1
Router(config-if)# end
```

Example: Configuring Dynamic NAT with Match-in-VRF

The following example shows how to configure dynamic NAT mappings with the same address pool. The **match-in-vrf** keyword enables NAT inside and outside traffic in the same VRF.

```
Router# configure terminal
Router(config)# ip nat inside source list 1 pool shared-pool vrf vrf1 match-in-vrf
Router(config)# access-list 1 permit 192.168.34.0 0.0.0.255
Router(config)# ip nat inside source list 1 pool shared-pool vrf vpn1
Router(config)# interface gigabitethernet 0/0/1
Router(config-if)# ip address 172.31.232.182 255.255.255.240
Router(config-if)# ip nat inside
Router(config-if)# ip vrf forwarding vpn1
Router(config-if)# exit
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# ip address 172.31.232.182 255.255.255.240
Router(config-if)# ip nat outside
Router(config-if)# ip vrf forwarding vpn1
Router(config-if)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
NAT commands	*Cisco IOS IP Addressing Services Command Reference*
NAT for IP Address Conservation	"Configuring NAT for IP Address Conservation" module

Standards and RFCs

Standard/RFC	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Match-in-VRF Support for NAT

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 16 Feature Information for Match-in-VRF Support for NAT

Feature Name	Releases	Feature Information
Match-in-VRF Support for NAT	Cisco IOS XE Release 3.5S	The Match-in-VRF Support for NAT feature supports the NAT translation of packets that communicate between two hosts within the same VPN.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.