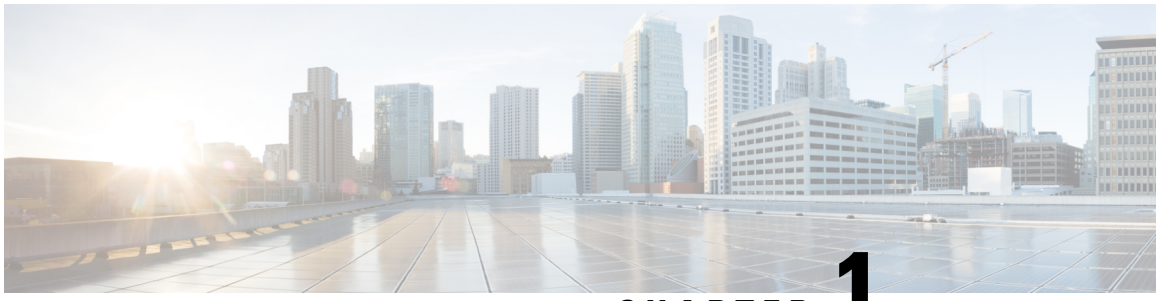
IP Addressing: NAT Configuration Guide, Cisco IOS Release 15M&T

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



CHAPTER 1

Configuring NAT for IP Address Conservation

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

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

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

Prerequisites for Configuring NAT for IP Address Conservation

Access Lists

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

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

NAT Requirements

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

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

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

Restrictions for Configuring NAT for IP Address Conservation

- It is not practical to use Network Address Translation (NAT) if a large number 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 not work at all through a NAT device.
- 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 with a NAT command, NAT will not support the **permit ip any any** command that is commonly used in the access list.
- NAT configuration is not supported on the access side of the Intelligent Services Gateway (ISG).
- On Cisco Catalyst 6500 Series Switches, if you have a NAT overload configuration, we recommend that you limit the number of NAT translations to less than 64512, by using the **ip nat translation max-entries** command. If the number of NAT translations is 64512 or more, a limited number of ports are available for use by local applications, which, in turn can cause security issues such as denial-of-service (DoS) attacks. The port numbers used by local applications can easily be identified by DoS attacks, leading to security threats. This restriction is specific to all NAT overload configurations (for example, interface

overload or pool overload configurations) that use a logical, loopback, or physical address for NAT configurations.

- Configuring zone-based policy firewall high availability with NAT and NAT high availability with zone-based policy firewalls is not recommended.
- If the NAT outside local address matches with any logical interface address, interface IP address, or a tunnel-configured address; then packets are software-switched.
- NAT outside interface is not supported on a VRF. However, NAT outside interface is supported in iWAN and is part of the Cisco Validated Design.
- The **acl-log** keyword will not work with an ACL used in NAT. The permit or deny functions for NAT ACL are used to filter the traffic according to the NAT rule. A rule like **permit tcp any any log** in the ACL used for NAT configuration is similar to **permit tcp any any**. Native ACL logging does not work in this ACL.
- BFD sessions may fail if you configure them to operate using an address that is also used for dynamic NAT. One common scenario is when you configure BFD on the same interface that you use to carry out interface-based dynamic NAT overload. To avoid this, you can instead employ a pool-based dynamic NAT overload configuration. However, even in this scenario, ensure that you do not use the chosen NAT pool address for BFD.

  When you configure BFD, we recommend you to use an address that does not overlap with NAT in order to avoid a conflict in case dynamic NAT is also configured on the device.

Information About Configuring NAT for IP Address Conservation

Benefits of Configuring NAT for IP Address Conservation

Network Address Translation (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 IP addresses, and if more than 254 clients are present or are planned, the scarcity of Class B addresses becomes a serious issue. NAT addresses these issues by mapping thousands of hidden internal addresses to a range of easy-to-get Class C addresses.

Sites that already have registered IP addresses for clients on an internal network may want to hide those addresses from the Internet so that hackers cannot directly attack clients. With client addresses hidden, a degree of security is established. 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). The expansion of Class A addresses occurs within the organization without a concern for addressing changes at the LAN or the Internet interface.

Cisco software can selectively or dynamically perform NAT. This flexibility allows 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, NAT allows the selection of internal hosts that are available for NAT.

A significant advantage of NAT is that it can be configured without requiring any changes to hosts or routers other than to those few routers on which NAT will be configured.

Purpose of NAT

NAT is a feature that allows the IP network of an organization to appear from the outside to use a different IP address space than what it is actually using. Thus, NAT allows an organization with nonglobally routable addresses to connect to the Internet by translating those addresses into a globally routable address space. NAT also allows a 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.

NAT supports all H.225 and H.245 message types, including FastConnect and Alerting, as part of the H.323 Version 2 specification. Any product that makes use of these message types will be able to pass through a Cisco NAT configuration without any static configuration. Full support for NetMeeting Directory (Internet Locator Service) is also provided through NAT.

How NAT Works

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

Uses of NAT

NAT can be used for the following scenarios:

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

NAT Inside and Outside Addresses

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

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

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

NAT supports the following VRFs:

Table 1: VRF NAT Support

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

This section describes the following topics:

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

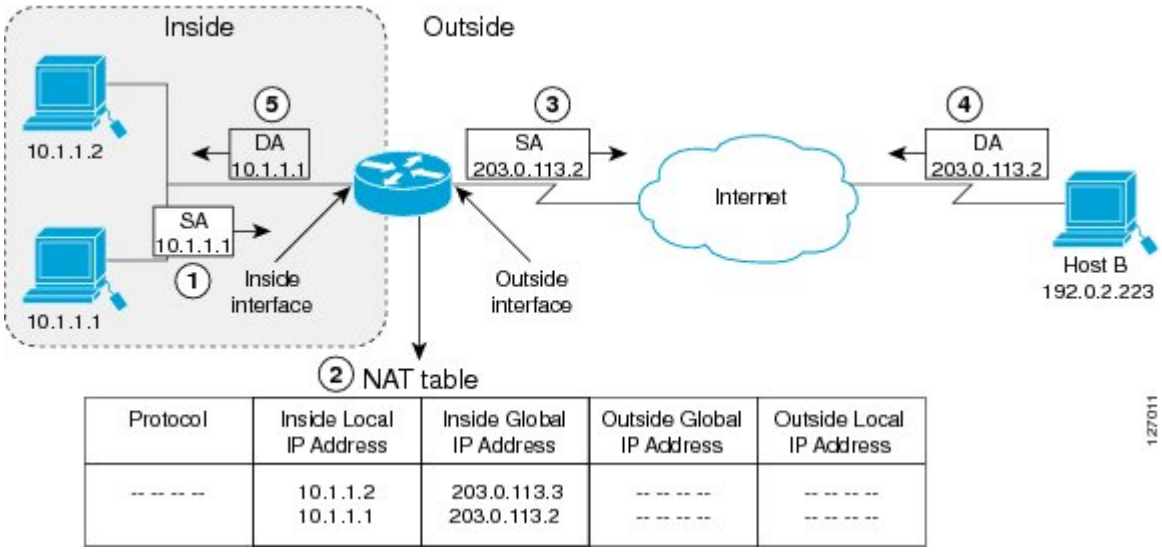
Inside Source Address Translation

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

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

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

Figure 1: NAT Inside Source Translation



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

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

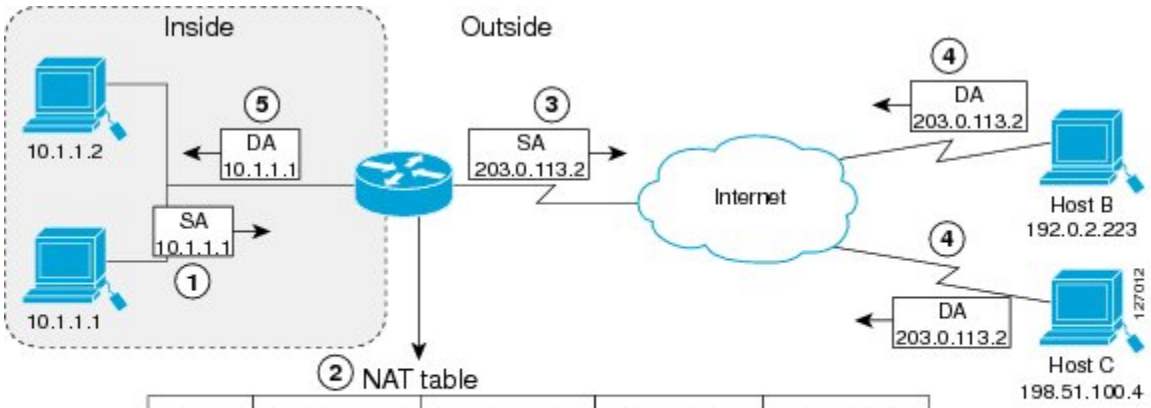
Host 10.1.1.1 receives the packet and continues the conversation. The device performs Steps 2 to 5 for each packet that it receives.

Overloading of Inside Global Addresses

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

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

Figure 2: NAT Overloading Inside Global Addresses



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

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

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

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

Types of NAT

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

The types of NAT include:

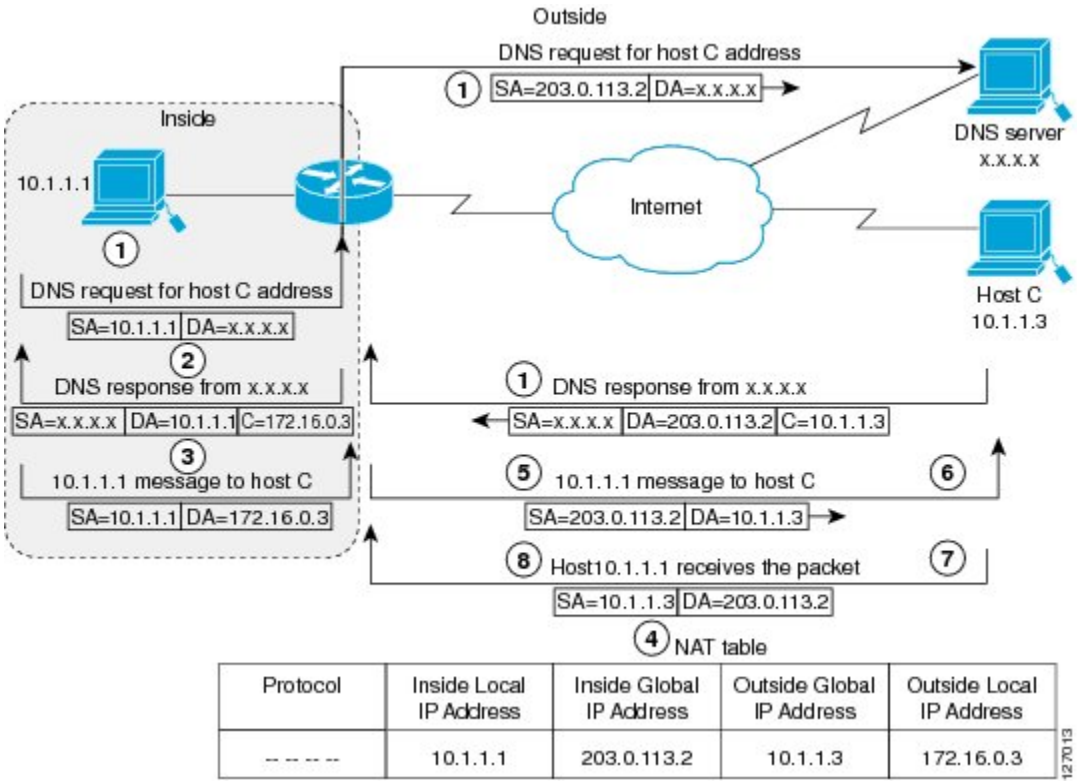
- Static address translation (static NAT)—Allows one-to-one mapping between local and global addresses.
- Dynamic address translation (dynamic NAT)—Maps unregistered IP addresses to registered IP addresses from a pool of registered IP addresses.
- Overloading—Maps multiple unregistered IP addresses to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

Address Translation of Overlapping Networks

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

The following figure shows how NAT translates overlapping networks.

Figure 3: NAT Translating Overlapping Addresses



The following steps describe how a device translates overlapping addresses:

1. Host 10.1.1.1 opens a connection to Host C using a name, requesting a name-to-address lookup from a Domain Name System (DNS) server.
2. The device intercepts the DNS reply, and translates the returned address if there is an overlap. That is, the resulting legal address resides illegally in the inside network. To translate the return address, the device creates a simple translation entry. This entry maps the overlapping address, 10.1.1.3 to an address from a separately configured, outside the local address pool.

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

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

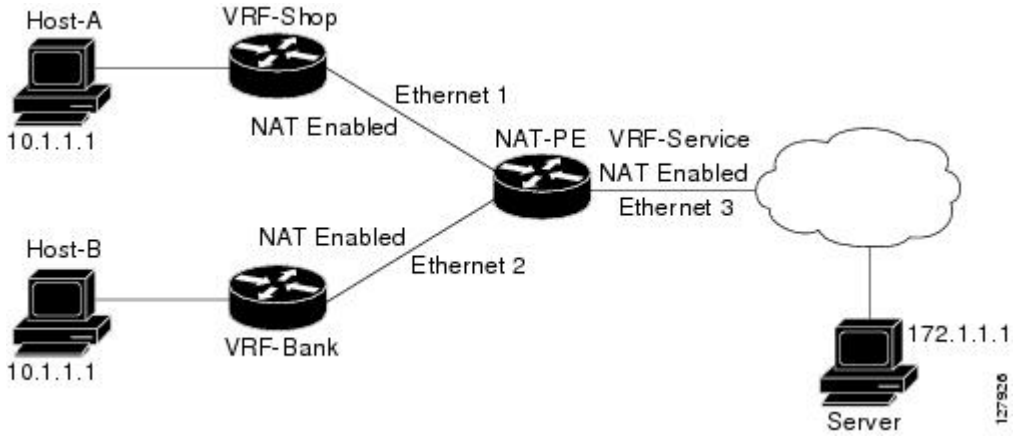
NAT Virtual Interface

The NAT Virtual Interface feature allows NAT traffic flows on the virtual interface, eliminating the need to specify inside and outside domains. When a domain is specified, translation rules are applied either before or after route decisions are applied, depending on the traffic flow from inside to outside or outside to inside. Translation rules are applied to a domain only after the route decision for a NAT Virtual Interface (NVI) is applied.

When a NAT pool is shared for translating packets from multiple networks connected to a NAT router, an NVI is created and a static route is configured that forwards all packets addressed to the NAT pool to the NVI. Standard interfaces connected to various networks are configured to determine if the traffic originating from and received on the interfaces needs to be translated.

The figure below shows a typical NVI configuration.

Figure 4: NAT Virtual Interface Typical Configuration



An NVI has the following benefits:

- A NAT table is maintained per interface for better performance and scalability.
- Domain-specific NAT configurations can be eliminated.

The following restrictions apply to an NVI configuration:

- Route maps are not supported.
- NVI is not supported in a NAT on-a-stick scenario. The term NAT on-a-stick implies the use of a single physical interface of a router for translation. NVI is designed for traffic from one VPN routing and forwarding (VRF) instance to another and not for routing between subnets in a global routing table. For more information, see the *Network Address Translation on a Stick* document.

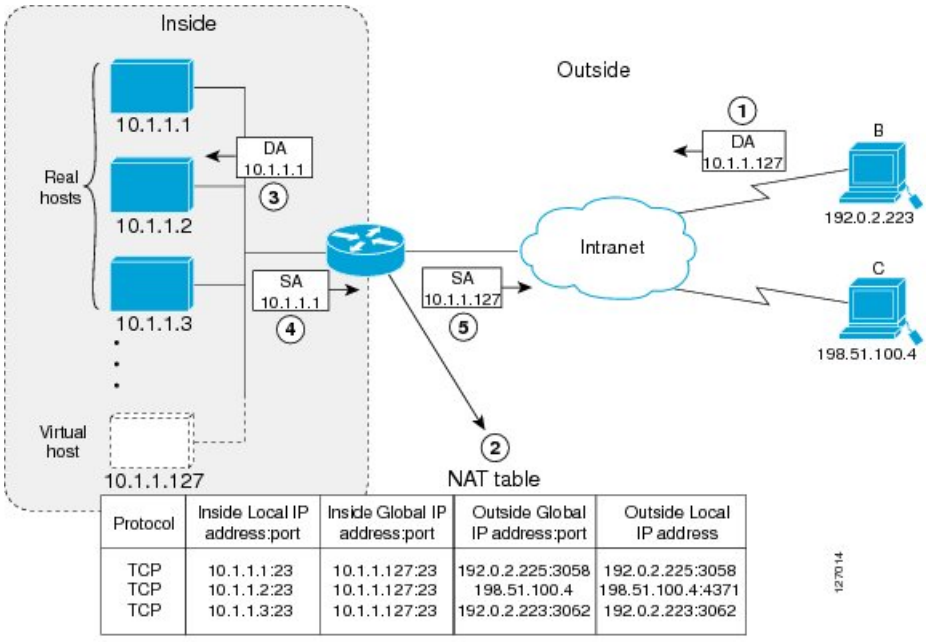
Note Use access-control list (ACL) to prevent inside hosts trying to establish an IPSec session to the same IPsec headend as the router.

Note NAT Virtual Interface gets dynamically created as part of NAT feature initialization and this interface is required for enabling the support for specific NAT usage scenarios. When a crypto module avails specific NAT services (APIs) to reserve transport ports that are of interest, the NAT feature is initialized creating a NAT Virtual interface.

TCP Load Distribution for NAT

Your organization may have multiple hosts that must communicate with a heavily used host. By using Network Address Translation (NAT), you can establish a virtual host on the inside network that coordinates load sharing among real hosts. Destination addresses that match an access list are replaced with addresses from a rotary pool. Allocation is done on a round-robin basis and only when a new connection is opened from the outside to inside the network. Non-TCP traffic is passed untranslated (unless other translations are configured). The following figure illustrates how TCP load distribution works.

Figure 5: 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	10.1.1.2:23	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

A device performs the following process when translating rotary addresses:

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

6. The device will allocate IP address 10.1.1.2 as the inside local address for the next connection request.

Route Map Overview

For NAT, a route map must be processed instead of an access list. A route map allows you to match any combination of access lists, next-hop IP addresses, and output interfaces to determine which pool to use. The ability to use route maps with static translations enables the NAT multihoming capability with static address translations. Multihomed internal networks can host common services such as the Internet and DNS, which are accessed from different outside networks. NAT processes route map-based mappings in lexicographical order. When static NAT and dynamic NAT are configured with route maps that share the same name, static NAT is given precedence over dynamic NAT. To ensure the precedence of static NAT over dynamic NAT, you can either configure the route map associated with static NAT and dynamic NAT to share the same name or configure the static NAT route map name so that it is lexicographically lower than the dynamic NAT route map name.

Benefits of using route maps for address translation 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.

NAT Route Maps Outside-to-Inside Support Feature

The NAT Route Maps Outside-to-Inside Support feature enables the deployment of a NAT route map configuration that allows IP sessions to be initiated from the outside to the inside.

An initial session from inside to outside is required to trigger a NAT. New translation sessions can then be initiated from the 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 it 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 entries) unless you configure the **ip nat inside source reversible** command.

The following restrictions apply to the NAT Route Maps Outside-to-Inside Support feature:

- Access lists with reversible route maps must be configured to match the inside-to-outside traffic.
- In Cisco IOS Release 12.2(33)SXI5, the NAT Route Maps Outside-to-Inside Support feature is supported only on Cisco ME 6500 series Ethernet switches.
- 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.

Static IP Address Support

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

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

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

RADIUS

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

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

Denial-of-Service Attacks

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

Viruses and Worms That Target NAT

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

How to Configure NAT for IP Address Conservation

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

Configuring Inside Source Addresses

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

Configuring Static Translation of Inside Source Addresses

Configure static translation of the inside source addresses to allow one-to-one mapping between an inside local address and an inside global address. Static translation is useful when a host on the inside must be accessible by a fixed address from the outside.

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

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
	`Device(config-if)# ip address 10.114.11.39 255.255.255.0`	
Step 6	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Specifies a different interface and enters the interface configuration mode.
Step 9	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 172.31.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 10	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 11	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode. **Note** Conditional translation is not supported with **ip nat outside source route-map** configuration.

Configuring Dynamic Translation of Inside Source Addresses

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool net-208 172.16.233.208 172.16.233.223 prefix-length 28`	Defines a pool of global addresses to be allocated as needed.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 192.168.34.0 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated.
Step 5	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Device(config)# ip nat inside source list 1 pool net-208`	Establishes dynamic source translation, specifying the access list defined in Step 4.
Step 6	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters an interface configuration mode.

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

Using NAT to Allow Internal Users Access to the Internet

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

SUMMARY STEPS

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

9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask*
12. **ip nat outside**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool net-208 192.168.202.129 192.168.202.158 netmask 255.255.255.224`	Defines a pool of global addresses to be allocated as needed.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 192.168.201.30 0.0.0.255`	Defines a standard access list permitting those addresses that are to be translated. • The access list must permit only those addresses that are to be translated. (Remember that there is an implicit "deny all" at the end of each access list.) Use of an access list that is too permissive can lead to unpredictable results.
Step 5	**ip nat inside source list** *access-list-number* **pool** *name* **overload** **Example:** `Device(config)# ip nat inside source list 1 pool net-208 overload`	Establishes dynamic source translation with overloading, specifying the access list defined in Step 4.
Step 6	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.201.1 255.255.255.240`	Sets a primary IP address for the interface.

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

Configuring Address Translation Timeouts

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

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

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

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

Changing the Translation Timeout

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

Changing the Timeouts When Overloading Is Configured

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

	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 translation** *seconds* **Example:** `Device(config)# ip nat translation 300`	(Optional) Changes the amount of time after which NAT translations time out. • The default timeout is 24 hours, and it applies to the aging time for half-entries. • The timeout configured using this command overrides the default TCP and UDP timeout values, unless explicitly configured.

	Command or Action	Purpose
Step 4	**ip nat translation udp-timeout** *seconds* **Example:** `Device(config)# ip nat translation udp-timeout 300`	(Optional) Changes the UDP timeout value. • The default is 300 seconds. This default value only applies if the general IP NAT translation timeout value (using the **ip nat translation** *seconds* command) is not configured.
Step 5	**ip nat translation dns-timeout** *seconds* **Example:** `Device(config)# ip nat translation dns-timeout 45`	(Optional) Changes the Domain Name System (DNS) timeout value.
Step 6	**ip nat translation tcp-timeout** *seconds* **Example:** `Device(config)# ip nat translation tcp-timeout 2500`	(Optional) Changes the TCP timeout value. • The default is 7440 seconds. This default value only applies if the general IP NAT translation timeout value (using the **ip nat translation** *seconds* command) is not configured.
Step 7	**ip nat translation finrst-timeout** *seconds* **Example:** `Device(config)# ip nat translation finrst-timeout 45`	(Optional) Changes the finish and reset timeout value. • **finrst-timeout**—The aging time after a TCP session receives both finish-in (FIN-IN) and finish-out (FIN-OUT) requests or after the reset of a TCP session.
Step 8	**ip nat translation icmp-timeout** *seconds* **Example:** `Device(config)# ip nat translation icmp-timeout 45`	(Optional) Changes the ICMP timeout value.
Step 9	**ip nat translation syn-timeout** *seconds* **Example:** `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 request is received on a TCP session. When a synchronous acknowledgment (SYNACK) request is received, the timeout changes to TCP timeout.
Step 10	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Allowing Overlapping Networks to Communicate Using NAT

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

This section contains the following tasks:

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

Configuring Static Translation of Overlapping Networks

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Device(config)# ip nat inside source static 192.168.121.33 10.2.2.1`	Establishes static translation between an inside local address and an inside global address.
Step 4	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.

	Command or Action	Purpose
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for the interface.
Step 6	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface ethernet 0`	Specifies an interface and enters the interface configuration mode.
Step 9	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for the interface.
Step 10	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 11	**end** **Example:** `Device(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

What to Do Next

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

Configuring 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.
- You want to communicate with those hosts or routers by 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Device(config)# ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24`	Defines a pool of global addresses to be allocated as needed.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 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:** `Device(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:** `Device(config)# interface ethernet 1`	Specifies an interface and enters the interface configuration mode.

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

Configuring the NAT Virtual Interface

The NAT Virtual Interface feature removes the requirement to configure an interface as either NAT inside or NAT outside. An interface can be configured to use or not use NAT.

Restrictions for NAT Virtual Interface

- Route maps are not supported.
- NVI is not supported in a *NAT on-a-stick* scenario. The term NAT on-a-stick implies the use of a single physical interface of a router for translation. NVI is designed for traffic from one VPN routing and forwarding (VRF) instance to another and not for routing between subnets in a global routing table. For more information on NAT on-a-stick, see http://www.cisco.com/en/US/tech/tk648/tk361/technologies_tech_note09186a0080094430.shtml.

Enabling a Dynamic NAT Virtual Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat enable**
5. **exit**
6. **ip nat pool** *name start-ip end-ip* **netmask** *netmask* **add-route**
7. **ip nat source list** *access-list-number* **pool** *name* **vrf** *name*
8. **ip nat source list** *access-list-number* **pool** *name* **overload**
9. **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 FastEthernet 1`	Configures an interface and enters interface configuration mode.
Step 4	**ip nat enable** **Example:** `Device(config-if)# ip nat enable`	Configures an interface that connects VPNs and the Internet for NAT.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **add-route** **Example:** `Device(config)# ip nat pool pool1 192.168.200.225 192.168.200.254 netmask 255.255.255.0 add-route`	Configures a NAT pool and the associated mappings.
Step 7	**ip nat source list** *access-list-number* **pool** *name* **vrf** *name* **Example:** `Device(config)# ip nat source list 1 pool pool1 vrf vrf1`	Configures a dynamic NVI without an inside or outside specification.

	Command or Action	Purpose
Step 8	**ip nat source list** *access-list-number* **pool** *name* **overload** **Example:** `Device(config)# ip nat source list 1 pool pool1 overload`	Configures an overloading NVI without an inside or outside specification.
Step 9	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Enabling a Static NAT Virtual Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat enable**
5. **exit**
6. **ip nat source static** *local-ip global-ip* **vrf** *name*
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 FastEthernet l`	Configures an interface type and enters interface configuration mode.
Step 4	**ip nat enable** **Example:** `Device(config-if)# ip nat enable`	Configures an interface that connects VPNs and the Internet for NAT.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 6	**ip nat source static** *local-ip global-ip* **vrf** *name* **Example:** `Device(config)# ip nat source static 192.168.123.1 192.168.125.10 vrf vrf1`	Configures a static NVI.
Step 7	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Configuring Server TCP Load Balancing

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

SUMMARY STEPS

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

DETAILED STEPS

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

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

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

Enabling Route Maps on Inside Interfaces

Before you begin

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**]| **static** *local-ip global-ip* [**route-map** *map-name*]} **Example:** `Device(config)# ip nat inside source static 192.168.201.6 192.168.201.21 route-map isp2`	Enables route mapping with static NAT configured on the NAT inside interface.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 5	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations`	(Optional) Displays active NAT.

Enabling NAT Route Maps Outside-to-Inside Support

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Device(config)# ip nat pool POOL-A 192.168.201.4 192.168.201.6 netmask 255.255.255.128`	Defines a pool of network addresses for NAT.
Step 4	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Device(config)# ip nat pool POOL-B 192.168.201.7 192.168.201.9 netmask 255.255.255.128`	Defines a pool of network addresses for NAT.
Step 5	**ip nat inside source route-map** *name* **pool** *name* [**reversible**] **Example:**	Enables outside-to-inside initiated sessions to use route maps for destination-based NAT.

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

Configuring NAT of External IP Addresses Only

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

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

Benefits of configuring NAT of external IP addresses only are:

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static network** *local-ip global-ip* [**no-payload**]}
4. **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**no-payload**]}

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** **network** *local-ip global-ip* [**no-payload**]} **Example:** `Device(config)# ip nat inside source static network 10.1.1.1 192.168.251.0/24 no-payload`	Disables the network packet translation on the inside host device.
Step 4	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** {**tcp** | **udp**} *local-ip local-port global-ip global-port* [**no-payload**]} **Example:** `Device(config)# ip nat inside source static tcp 10.1.1.1 2000 192.168.1.1 2000 no-payload`	Disables port packet translation on the inside host device.
Step 5	**ip nat inside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* [**overload**] | **static** [**network**] *local-network-mask global-network-mask* [**no-payload**]} **Example:** `Device(config)# ip nat inside source static 10.1.1.1 192.168.1.1 no-payload`	Disables packet translation on the inside host device.
Step 6	**ip nat outside source** {**list** {*access-list-number* | *access-list-name*} **pool** *pool-name* | **static** *local-ip global-ip* [**no-payload**]}	Disables packet translation on the outside host device.

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

Configuring the NAT Default Inside Server Feature

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

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

Note

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

SUMMARY STEPS

1. **enable**

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nat inside source static** *local-ip* **interface** *type number* **Example:** `Device(config)# ip nat inside source static 10.1.1.1 interface Ethernet 1/1`	Enables static NAT on the interface.
Step 4	**ip nat inside source static tcp** *local-ip local-port* **interface** *global-port* **Example:** `Device(config)# ip nat inside source static tcp 10.1.1.1 23 interface 23`	(Optional) Enables the use of telnet to the device from the outside.
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip nat translations** [**verbose**] **Example:** `Device# show ip nat translations`	(Optional) Displays active NAT.

Reenabling RTSP on a NAT Router

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

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

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

Configuring Support for Users with Static IP Addresses

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

Before you begin

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 1`	Configures an interface and enters an interface configuration mode.
Step 4	**ip nat inside** **Example:** `Device(config-if)# ip nat inside`	Marks the interface as connected to the inside.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 6	**ip nat allow-static-host** **Example:** `Device(config)# ip nat allow-static-host`	Enables static IP address support. • Dynamic Address Resolution Protocol (ARP) learning will be disabled on this interface, and NAT will control the creation and deletion of ARP entries for the static IP host.
Step 7	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **accounting** *list-name* **Example:** `Device(config)# ip nat pool pool1 172.16.0.0 172.16.0.254 netmask 255.255.255.0 accounting WLAN-ACCT`	Specifies an existing RADIUS profile name to be used for authentication of the static IP host.
Step 8	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Device(config)# ip nat inside source list 1 pool net-208`	Specifies the access list and pool to be used for static IP support. • The specified access list must permit all traffic.
Step 9	**access-list** *access-list-number* **deny ip** *source* **Example:** `Device(config)# access-list 1 deny ip 192.168.196.51`	Removes the traffic of the device from NAT. • The *source* argument is the IP address of the device that supports the NAT Static IP Support feature.
Step 10	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
Step 11	**show ip nat translations verbose** **Example:** `Device# show ip nat translations verbose`	(Optional) Displays active NAT translations and additional information for each translation table entry, including how long ago the entry was created and used.

Examples

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

```
Device# show ip nat translations verbose

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

Configuring Support for ARP Ping

When the NAT entry of the static IP client times out, the NAT entry and the secure ARP entry associations are deleted for the client. The ARP Ping feature enables the NAT entry and the secure ARP entry to not be deleted when the static IP client exists in the network where the IP address is unchanged after authentication.

An ARP ping is necessary to determine static IP client existence and to restart the NAT entry timer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat pool** *name start-ip end-ip* **prefix-length** *prefix-length* [**accounting** *method-list-name*] [**arp-ping**]
4. **ip nat translation arp-ping-timeout** [*seconds*]
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* **prefix-length** *prefix-length* [**accounting** *method-list-name*] [**arp-ping**] **Example:** `Device(config)# ip nat pool net-208 172.16.233.208 172.16.233.223 prefix-length 28 accounting radius1 arp-ping`	Defines a pool of IP addresses for NAT.
Step 4	**ip nat translation arp-ping-timeout** [*seconds*] **Example:** `Device(config)# ip nat translation arp-ping-timeout 600`	Changes the amount of time after each network address translation.
Step 5	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

Configuring the Rate Limiting NAT Translation Feature

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip nat translations** **Example:** `Device# show ip nat translations`	(Optional) Displays active NAT. • A specific host, access control list, or VRF instance generating an unexpectedly high number of NAT requests may be the source of a malicious virus or worm attack.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**ip nat translation max-entries** {*number* | **all-vrf** *number* | **host** *ip-address number* | **list** *listname number* | **vrf** *name number*} **Example:** `Device(config)# ip nat translation max-entries 300`	Configures the maximum number of NAT entries that are allowed from the specified source. • The maximum number of allowed NAT entries is 2147483647, although a typical range for a NAT rate limit is 100 to 300 entries. • When you configure a NAT rate limit for all VRF instances, each VRF instance is limited to the maximum number of NAT entries that you specify. • When you configure a NAT rate limit for a specific VRF instance, you can specify a maximum number of NAT entries for the named VRF instance that is greater than or less than that allowed for all VRF instances.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show ip nat statistics** **Example:** `Device# show ip nat statistics`	(Optional) Displays current NAT usage information, including NAT rate limit settings. • After setting a NAT rate limit, use the **show ip nat statistics** command to verify the current NAT rate limit settings. **Note** The CEF counters associated with the output of the **show ip nat statistics** command signify the number of packets that are translated and forwarded in the SW plane. Packets that require translation are punted to the SW plane in the absence of the corresponding NF shortcuts in the HW plane. This enables SW plane to carry out the translation and program the corresponding NF shortcuts in the HW in order to facilitate the HW translation for subsequent packets that match the given flow. A route-map based NAT rule does not maintain Half Entry mappings and this implies that every new packet flow that matches the given rule is directed to the SW plane for translation and forwarding. Such packets undergo translation in the SW plane. This in turn results in the increment of the afore mentioned CEF counters. This is an expected behavior when you employ a route-map-based NAT configuration. However, note that these packets that undergo translation in the SW result in the corresponding full flow NF shortcuts to be programmed in the HW. This is to facilitate the HW translation of subsequent packets that match the given flow.

Configuration Examples for Configuring NAT for IP Address Conservation

Example: Configuring Static Translation of Inside Source Addresses

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

```
ip nat pool net-208 172.31.233.208 172.31.233.223 prefix-length 28
ip nat pool net-10 10.0.1.0 10.0.1.255 prefix-length 24
ip nat inside source list 1 pool net-208
```

```
ip nat outside source list 1 pool net-10
!
interface gigabitethernet 0/0/0
 ip address 172.31.232.182 255.255.255.240
 ip nat outside
!
interface gigabitethernet 1/1/1
 ip address 10.114.11.39 255.255.255.0
 ip nat inside
!
access-list 1 permit 10.114.11.0 0.0.0.255
```

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

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

Example: Configuring Dynamic Translation of Inside Source Addresses

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

```
ip nat pool net-208 172.31.233.208 172.31.233.223 prefix-length 9
ip nat inside source list 1 pool net-208
!
interface gigabitethernet 0/0/0
 ip address 172.31.232.182 255.255.255.240
 ip nat outside
!
interface gigabitethernet 1/1/1
 ip address 192.168.1.94 255.255.255.0
 ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255
access-list 1 permit 192.168.2.0 0.0.0.255
!
```

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

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

```
access-list 1 permit 10.1.1.0 0.0.0.255
!
```

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

The following example shows how to create a pool of addresses that is named net-208. The pool contains addresses from 172.31.233.208 to 172.31.233.233. Access list 1 allows packets with SA from 192.168.1.0 to 192.168.1.255. If no translation exists, packets matching access list 1 is translated to an address from the pool. The router allows multiple local addresses (192.168.1.0 to 192.168.1.255) to use the same global address. The router retains port numbers to differentiate the connections.

```
ip nat pool net-208 172.31.233.208 172.31.233.233 netmask 255.255.255.240
access-list 1 permit 192.168.1.0 0.0.0.255
ip nat inside source list 1 pool net-208 overload
interface gigabitethernet 1/1/1
 ip address 192.168.201.1 255.255.255.240
 ip nat inside
!
interface gigabitethernet 0/0/0
 ip address 192.168.201.29 255.255.255.240
 ip nat outside
!
```

Example: Allowing Overlapping Networks to Communicate Using NAT

Example: Configuring the NAT Virtual Interface

Example: Enabling a Static NAT Virtual Interface

```
interface FastEthernet 1
 ip nat enable
!
ip nat source static 192.168.123.1 182.168.125.10 vrf vr1
!
```

Example: Enabling a Dynamic NAT Virtual Interface

```
interface FastEthernet 1
 ip nat enable
!
ip nat pool pool1 192.168.200.225 192.168.200.254 netmask 255.255.255.0 add-route
ip nat source list 1 pool pool1 vrf vrf1
ip nat source list 1 pool 1 vrf vrf2 overload
!
```

Example: Configuring Server TCP Load Balancing

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

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

Example: Enabling Route Maps on Inside Interfaces

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

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

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

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

Example: Configuring NAT of External IP Addresses Only

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

Example: Configuring Support for Users with Static IP Addresses

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

Example: Configuring NAT Static IP Support

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

```
interface gigabitethernet 1/1/1
 ip nat inside
!
ip nat allow-static-host
```

```
ip nat pool net-208 172.16.1.1 172.16.1.10 netmask 255.255.255.0 accounting WLAN-ACCT
ip nat inside source list 1 pool net-208
access-list 1 deny ip 192.168.196.51
```

Example: Creating a RADIUS Profile for NAT Static IP Support

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

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

Example: Configuring the Rate Limiting NAT Translation Feature

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

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

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

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

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

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

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

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

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

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

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

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

Example: Setting a Global NAT Rate Limit

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

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

Example: Setting NAT Rate Limits for a Specific VRF Instance

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

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

Example: Setting NAT Rate Limits for All VRF Instances

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

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

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

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

Example: Setting NAT Rate Limits for Access Control Lists

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

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

Example: Setting NAT Rate Limits for an IP Address

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

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

Where to Go Next

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

Additional References

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
Application-level gateways	*Using Application Level Gateways with NAT* module

Related Topic	Document Title
IP access list sequence numbering	IP Access List Sequence Numbering document
NAT-on-a-Stick technology note	Network Address Translation on a Stick technology note
NAT maintenance	*Monitoring and Maintaining NAT* module
RADIUS attributes overview	*RADIUS Attributes Overview and RADIUS IETF Attributes* module
Using HSRP and stateful NAT for high availability	*Configuring NAT for High Availability* module
Using NAT with MPLS VPNs	*Integrating NAT with MPLS VPNs* module

Standards and RFCs

Standard/RFC	Title
RFC 1597	Internet Assigned Numbers Authority
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)

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

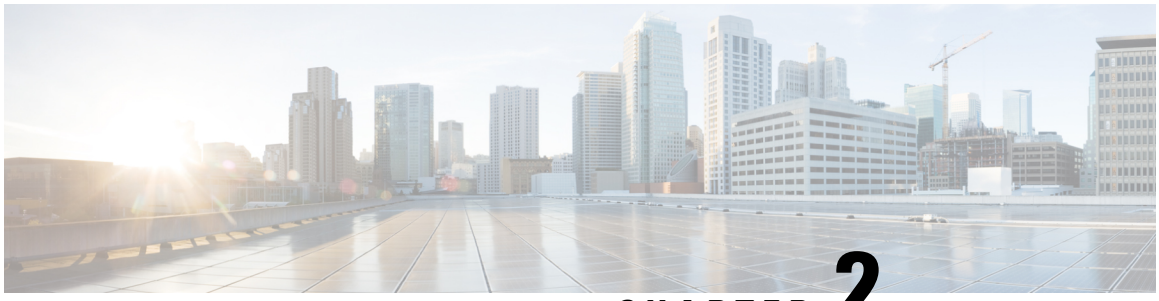
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 2: Feature Information for Configuring NAT for IP Address Conservation

Feature Name	Releases	Feature Information
NAT Ability to Use Route Maps with Static Translation	12.2.(4)T	The NAT Ability to Use Route Maps with Static Translation feature provides a dynamic translation command that can specify a route map to be processed instead of an access list. A route map allows you to match any combination of the 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.
NAT Default Inside Server	12.3(13)T	The NAT Default Inside Server feature enables forwarding of packets from outside to a specified inside local address.
NAT Route Maps Outside-to-Inside Support	12.2(33)SXI5 12.3(14)T	The NAT Route Maps Outside-to-Inside Support feature enables the deployment of a NAT route map configuration that allows IP sessions to be initiated from the outside to the inside.
NAT RTSP Support Using NBAR	12.3(7)T	The NAT RTSP Support Using NBAR feature is a client/server multimedia presentation control protocol that supports multimedia application delivery. Applications that use RTSP include WMS by Microsoft, QuickTime by Apple Computer, and RealSystem G2 by RealNetworks.
NAT Static and Dynamic Route Map Name-Sharing	15.0(1)M	The NAT Static and Dynamic Route Map Name-Sharing feature provides the ability to configure static and dynamic NAT to share the same route map name, while enforcing precedence of static NAT over dynamic NAT.
NAT Static IP Support	12.3(7)T	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 Translation of External IP Addresses Only	12.2(4)T 12.2(4)T2 15.0(1)S	Use the NAT Translation of External IP Addresses Only feature to configure NAT to ignore all embedded IP addresses for any application and traffic type.

Feature Name	Releases	Feature Information
NAT Virtual Interface	12.3(14)T	The NAT Virtual Interface feature removes the requirement to configure an interface as either Network Address Translation (NAT) inside or NAT outside. An interface can be configured to use or not use NAT.
Rate Limiting NAT Translation	12.3(4)T 15.0(1)S	The Rate Limiting NAT Translation feature provides the ability to limit the maximum number of concurrent Network Address Translation (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 denial-of-service attacks.
Support for ARP Ping in a Public Wireless LAN	12.4(6)T	The Support for ARP Ping in a Public Wireless LAN feature ensures that the NAT entry and the secure ARP entry from removal when the static IP client exists in the network, where the IP address is unchanged after authentication.



CHAPTER 2

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 ALGs 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. 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), remote login (rlogin), remote shell (rsh) protocol, and remote copy (rcp).

Specific protocols that embed the IP address information within the payload require the 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. You can use the **ip nat service dns-v6** command to control processing of IPv6 DNS packets by ALG

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.
- 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.

- Before performing the tasks in this module, you should verify that the Session Initiation Protocol (SIP) and H.323 are not disabled. SIP and H.323 are enabled by default.

Restrictions for Using Application-Level Gateways with NAT

- Network Address Translation (NAT) translates only embedded IPv4 addresses.
- Protocols that require application-level gateway (ALG) processing are not compatible with load balancing. All process-switched packets use the per-packet load-balancing algorithm. Process switching uses the per-packet load-balancing algorithm across equal-cost paths. As a result, every odd or even process-switched packet may be dropped by ISPs in a dual-ISP scenario due to a failed Unicast Reverse Path Forwarding (uRPF) check because these packets have the same source IP address (which is allocated by NAT or Port Address Translation [PAT]), but are routed to different outside interfaces. The packet drop causes excessive delay and retransmission of packets.
- In Cisco IOS Release 12.4 Mainline, the NAT ALG for Session Initiation Protocol (SIP) does not support the following T.38 session attributes in the Session Description Protocol (SDP): sqn, cdsc, and cpar. These session attributes are removed from the SDP header by the NAT ALG, which causes the SIP-based T.38 calls to fail. This restriction is applicable only to the Cisco IOS Release 12.4 mainline. As a workaround, upgrade to Cisco IOS Release 12.4(1)T and later releases.

Information About Using Application-Level Gateways with NAT

Benefits of Configuring NAT IPsec

- NAT enables customers to deploy private IP addresses within their network and translate the private IP addresses to public IP addresses when connecting to the Internet or when interconnecting with another corporate network.
- NAT support for the Session Initiation Protocol (SIP) adds the ability to deploy NAT on VoIP solutions based on SIP.
- With NAT ALGs, customers can control their IP address scheme and include complete support for H.323 v2 gatekeeper designs.
- 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 SPI entries are matched. Some third-party concentrators require both source ports and incoming ports to use port 500. Use the **ip nat service preserve-port** command to preserve the ports rather than changing them, which is required with regular NAT.

IPsec

IPsec is a set of extensions to the IP protocol 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 characteristics of these tunnels. When the IPsec peer receives a sensitive packet, it 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. You can enable IPsec packet processing using ESP with the **ip nat service ipsec-esp enable** command.

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. Such factors include the capabilities of the VPN server and client, the 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 router 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 Cisco IOS router 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.

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..

Voice and Multimedia over IP Networks

SIP is a protocol developed by the IETF Multiparty Multimedia Session Control (MMUSIC) Working Group. The Cisco SIP functionality equips Cisco routers to signal the setup of voice and multimedia calls over IP networks. SIP provides an alternative to H.323 within the 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 router configured with NAT to be translated and encoded back to the packet. An ALG is used with NAT to translate the SIP or SDP 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 Support of H.323 v2 RAS

Cisco IOS 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.

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

In Cisco IOS Release 12.2(2)T and later releases, embedded IP addresses can be inspected for potential address translation. Prior to Cisco IOS Release 12.2(2)T, NAT did not support H.323 v2 RAS messages.

NAT Support for H.323 v3 and v4 in v2 Compatibility Mode

H.323 is an ITU-T specification for transmitting audio, video, and data across packet networks. NAT supports four versions of the H.323 protocols: Version 1, Version 2, Version 3, and Version 4. The NAT Support for H.323 v3 and v4 in v2 Compatibility Mode feature enables NAT routers to support messages coded in H.323 Version 3 and Version 4 when these messages contain fields that are compatible with H.323 Version 2. This feature does not support H.323 capabilities introduced in H.323 Version 3 and Version 4, such as new message types or new fields that require address translation.

NAT H.245 Tunneling Support

The NAT H.245 Tunneling Support feature supports H.245 tunneling in H.323 ALGs. The H.245 tunneling supports H.245 tunnel messages that are needed to create a media channel setup.

For an H.323 call to take place, an H.225 connection on TCP port 1720 must be opened. When the H.225 connection is opened, the H.245 session is initiated and established. The H.323 connection can take place on a separate channel other than the H.225 or it can be done by using H.245 tunneling on the same H.225 channel whereby the H.245 messages are embedded in H.225 messages and sent on the previously established H.225 channel.

If the H.245 tunneled message is not understood by NAT, the media address or the port number is left untranslated by NAT, resulting in media traffic failure. The H.245 FastConnect procedures will not help if the H.245 tunneled message is not understood by NAT because FastConnect is terminated as soon as an H.245 tunneled message is sent.

NAT Support of Skinny Client Control Protocol

Cisco IP phones use the SCCP to connect with and register to Cisco CallManager.

To be able to configure Cisco IOS NAT between the IP phone and Cisco CallManager in a scalable environment, NAT needs to be able to detect the SCCP and understand the information passed within the messages. Messages flow back and forth that include IP address and port information used to identify other IP phone users with which a call can be placed.

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

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

NAT Support of SCCP Fragmentation

Skinny Client Control Protocol (SCCP) messages, also called Skinny control messages, are exchanged over TCP. If either the IP phone or the Cisco Unified CallManager is configured to have a TCP maximum segment size (MSS) lower than the Skinny control message payload, the Skinny control message is segmented across multiple TCP segments. Prior to the introduction of this feature, Skinny control message exchanges used to fail during TCP segmentation because the NAT Skinny ALG was not able to reassemble Skinny control messages. The NAT SCCP Fragmentation Support feature adds support for TCP segments for the NAT Skinny ALG and fragmented payloads that requires an IP translation or a port translation is no longer dropped.

Skinny control messages can also be IP fragmented by using Virtual Fragmentation Reassembly (VFR).

In Cisco IOS Release 15.1(3)T and later releases, NAT works with SCCP phones Version 17 and higher.

NAT Segmentation with Layer 4 Forwarding

The NAT Segmentation with Layer 4 Forwarding feature is implemented for the H.323, Skinny Client Control Protocol (SCCP), and the TCP Domain Name System (DNS) protocol. NAT supports the processing of segmented H.323, SCCP, or TCP DNS messages that are split across multiple packets.

Layer 4 forwarding or TCP proxy is responsible for session handling that includes setting sequence numbers in order, acknowledging the numbers in a packet, resegmenting the translated packet if it is larger than the maximum segment size (MSS), and handling retransmissions in case of packet loss. Layer 4 forwarding also handles out-of-order packets and these packets are buffered and not dropped. Layer 4 forwarding buffers received packets and notifies the NAT ALG when an in-order packet is available, sends acknowledgments to end hosts for received packets, and sends translated packets that it receives from the NAT ALG back into the output packet path.

Restrictions

The NAT Segmentation with Layer 4 Forwarding feature does not work when:

- Firewalls are configured using the **ip inspect name** command. (Context-Based Access Control (CBAC) firewalls are not supported. Zone-based firewalls are supported.)
- H.323, SCCP, or TCP DNS messages are larger than 18 KB.
- Multiprotocol Label Switching (MPLS) is configured.
- NAT and the Cisco Unified CallManager are configured on the same device. In this case, a colocated solution in Call Manager Express is used.
- NAT Virtual Interface (NVI) is configured.
- Stateful Network Address Translation (SNAT) is enabled.

Note Effective January 31, 2014, Stateful NAT is not available in Cisco IOS software. For more information, see End-of-Sale and End-of-Life Announcement for the Cisco IOS Stateful Failover of Network Address Translation (SNAT).

- The **match-in-vrf** keyword is configured along with the **ip nat inside source** command for packet translation.
- The packets are IPv6 packets.

How to Configure Application-Level Gateways with NAT

Configuring IPsec Through NAT

Configuring IPsec ESP Through NAT

IPsec ESP Through NAT provides the ability to support multiple concurrent IPsec ESP tunnels or connections through a Cisco IOS NAT device configured in Overload or PAT mode.

Perform this task to configure IPsec ESP through NAT.

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat** [**inside** | **outside**] **source static** *local-ip global-ip* [**vrf** *vrf-name*]
4. **exit**
5. **show ip nat translations**

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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**ip nat** [**inside** | **outside**] **source static** *local-ip global-ip* [**vrf** *vrf-name*] **Example:** `Router(config)# ip nat inside source static 10.10.10.10 192.168.30.30`	Enables static NAT.
Step 4	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show ip nat translations** **Example:** `Router# show ip nat translations`	(Optional) Displays active NATs.

Enabling the Preserve Port

Note This task is required by certain VPN concentrators. Cisco VPN devices generally do not use this feature.

This task is used for IPsec traffic using port 500 for the source port. Perform this task to enable port 500 to be preserved for the source port.

SUMMARY STEPS

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

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 service list** *access-list-number* **IKE preserve-port** **Example:** `Router(config)# ip nat service list 10 IKE preserve-port` **Note** When you configure the **ip nat service list** *list* **IKE preserve-port**, ensure that you define the access list for both in2out and out2in traffic.	Specifies IPsec traffic that matches the access list to preserve the port.

Enabling SPI Matching on the NAT Device

Note SPI matching is disabled by default.

Security parameter index (SPI) matching is used to establish VPN connections between multiple pairs of destinations. NAT entries are immediately placed in the translation table for endpoints matching the configured access list. SPI matching is available only for endpoints that choose SPIs according to the predictive algorithm implemented in Cisco IOS Release 12.2(15)T.

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

Before you begin

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

Note SPI matching must be configured on the NAT device and both endpoint devices.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service list** *access-list-number* **ESP spi-match**

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 service list** *access-list-number* **ESP spi-match** **Example:** `Router(config)# ip nat service list 10 ESP spi-match`	Specifies an access list to enable SPI matching. • This example shows how to enter ESP traffic matching list 10 into the NAT table, making the assumption that both devices are Cisco devices and are configured to provide matchable SPIs.

Enabling SPI Matching on Endpoints

Before you begin

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.

Enabling MultiPart SDP Support for NAT

The MultiPart SDP Support for NAT feature provides support for the multipart Session Description Protocol (SDP) in a SIP ALG. MultiPart SDP support for NAT is disabled by default.

Note NAT translates only embedded IPv4 addresses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service allow-multipart**
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 service allow-multipart** **Example:** `Device(config)# ip nat service allow-multipart`	Enables multipart SDP.
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.

Configuring NAT Between an IP Phone and Cisco CallManager

This section describes configuring Cisco's Skinny Client Control Protocol (SCCP) for Cisco IP phone to Cisco CallManager communication. The task in this section configures NAT between an IP phone and Cisco CallManager.

SUMMARY STEPS

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

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

Configuration Examples for Using Application-Level Gateways with NAT

Example: Specifying a Port for NAT Translation

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

Example: Enabling the Preserve Port

The following example shows how to configure TCP port 500 of the third-party concentrator. Access list 10 is configured.

```
ip nat service list 10 IKE preserve-port
access-list 10 permit 10.1.1.1
```

Example Enabling SPI Matching

The following example shows how to enable SPI matching. Access list 10 is configured:

```
ip nat service list 10 ESP spi-match
access-list 10 permit 10.1.1.1
```

Example: Enabling SPI Matching on Endpoints

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

Example: Enabling MultiPart SDP Support for NAT

```
ip nat service allow-multipart
```

Example: Specifying a Port for NAT Translation

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

Where to Go Next

- To learn about NAT and configure NAT for IP address conservation, see the “Configuring NAT for IP Address Conservation” module.
- To verify monitor, and maintain NAT, see the “Monitoring and Maintaining NAT” module.
- To integrate NAT with MPLS VPNs, see the “Integrating NAT with MPLS VPNs” module.
- To configure NAT for high availability, see the “Configuring NAT for High Availability” module.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IP Addressing Services Command Reference
IP access list sequence numbering	*IP Access List Sequence Numbering*
NAT IP address conservation	*Configuring NAT for IP Address Conservation*

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

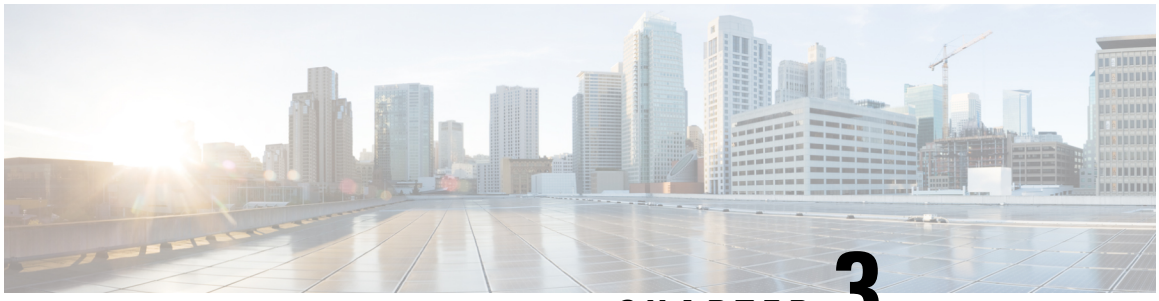
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 3: Feature Information for Using Application-Level Gateways with NAT

Feature Name	Releases	Feature Configuration Information
MultiPart SDP Support for NAT	15.0(1)M	The MultiPart SDP Support for NAT feature adds support for multipart SDP in a SIP ALG. This feature is disabled by default. The following commands were modified by this feature: **debug ip nat** and **ip nat service**.
NAT H.245 Tunneling Support	12.3(11)T	The NAT H.245 Tunneling Support feature allows H.245 tunneling in H.323 Application-Level Gateways (ALGs).
NAT Support for H.323 v2 RAS feature	12.2(2)T 15.0(1)S	NAT supports all H.225 and H.245 message types, including those sent in the RAS protocol.
NAT Support for H.323 v3 and v4 in v2 Compatibility Mode	12.3(2)T	The NAT Support for H.323 v3 and v4 in v2 Compatibility Mode feature enables NAT routers to support messages coded in H.323 Version 3 and Version 4 when these messages contain fields that are compatible with H.323 Version 2. This feature does not add support for H.323 capabilities introduced in H.323 Version 3 and Version 4, such as new message types or new fields that require address translation.

Feature Name	Releases	Feature Configuration Information
NAT Support for IPsec ESP—Phase II	12.2(15)T	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 router configured with NAPT.
NAT Support of SCCP Fragmentation	12.4(6)T 15.1(3)T	The NAT Support of SCCP Fragmentation feature adds support for TCP segments for the NAT Skinny ALG. A fragmented payload that requires an IP translation or a port translation is no longer be dropped. The following command was modified by this feature: **debug ip nat**.
NAT Support for SIP	12.2(8)T	NAT Support for SIP adds the ability to configure NAT on VoIP solutions based on SIP.
Support for applications that do not use H.323	12.2(33)XNC	NAT with an ALG will translate packets from applications that do not use H.323, as long as these applications use port 1720.
Support for IPsec ESP Through NAT	12.2(13)T	The 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.



CHAPTER 3

NAT Box-to-Box High-Availability Support

The NAT Box-to-Box High-Availability Support feature enables network-wide protection by making an IP network more resilient to potential link and router failures at the Network Address Translation (NAT) border.

NAT box-to-box high-availability functionality is achieved when you configure two NAT translators that reside across different devices as part of a redundancy group (RG) and function as a translation group. One member of the translation group acts as an active translator and the other member in the group acts as a standby translator. The standby translator takes over as the active translator in the event of any failures to the current active translator.

This module provides information about NAT box-to-box high-availability support and describes how to configure this 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.

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

Prerequisites for NAT Box-to-Box High-Availability Support

- Network Address Translation (NAT)-related configurations must be manually configured and the configuration must be identical on both devices, and associated to the same redundancy group (RG).

- RG must be shut down on both peer devices before you configure NAT.
- If devices are already in active/standby states, you must apply any additional configuration changes first on the standby device and then on the active device. To delete NAT configuration rules, you must apply the changes first on the active device and then on the standby device.

Restrictions for NAT Box-to-Box High-Availability Support

- Network Address Translation (NAT) configurations with the interface overload option are not supported.
- Both application redundancy (using redundancy groups [RGs]) and box-level redundancy cannot be configured on the same device.
- RG infrastructure with more than one RG peer is not supported.
- Multiprotocol Label Switching (MPLS) with Layer 3 VPN (L3VPN) configuration is not supported.
- NAT Virtual Interface (NVI) configuration is not supported.
- Only FTP application layer gateway (ALG) is supported. All other ALGs are not high-availability aware and are not expected to work correctly across failovers.

Note We recommend that you disable all other ALGs using the **no ip nat service** command, when using this feature.

Information About NAT Box-to-Box High-Availability Support

NAT Box-to-Box High-Availability Overview

The NAT Box-to-Box High-Availability Support feature enables network-wide protection by making an IP network resilient to potential link and router failures at the Network Address Translation (NAT) border.

The NAT Box-to-Box High-Availability Support feature leverages services provided by the redundancy group (RG) infrastructure present on the device to implement the high-availability functionality. The RG infrastructure defines multiple RGs to which applications can subscribe to and function in an active-standby mode across different devices. NAT box-to-box high-availability functionality is achieved when you configure two NAT translators, residing across different devices, to an RG and function as a translation group. One member of the translation group acts as an active translator and the other members of the translation group acts as a standby translator. The active translator is responsible for handling traffic that requires address translation. Additionally, the active translator informs the standby translator about packet flows that are being translated. The standby translator uses this information to create a duplicate translation database that equips the standby translator to take over as the active translator in the event of any failures to the active translator. Therefore, the application traffic flow continues unaffected as the translations tables are backed up in a stateful manner across the active and standby translators.

The NAT Box-to-Box High-Availability Support feature supports active-standby high-availability failover and asymmetric routing. The NAT Box-to-Box High-Availability Support feature supports the following NAT features:

- Simple Static NAT configuration

- Extended Static NAT configuration
- Network Static NAT configuration
- Dynamic NAT and Port Address Translation (PAT) configuration
- NAT inside source, outside source, and inside destination rules
- NAT rules for Virtual Routing and Forwarding (VRF) instances to IP
- NAT rules for VRF-VRF (within same VRF)

Reasons for Active Device Failover

The following are some of the reasons for the failover of an active device:

- Power loss or reload on the active device.
- Control interface for the redundancy group (RG) is shut down or the link to the interface is down.
- Data interface for the RG is shut down or the link to the interface is down.
- Tracked object failure.
- Protocol keepalive failure.
- The run-time priority of the active device is below the configured threshold. Run-time priority can go down in the following scenarios:
  - Traffic interface, that is assigned a Redundancy Interface Identifier (RII) value, is down.
  - Object tracked by the RG is down.
- RG on an active device is reloaded using the **redundancy application reload group** command in privileged EXEC mode.
- RG on an active device is shut down using the **group** command in redundancy application configuration mode.

NAT in Active-Standby Mode

In active-standby mode, the redundancy group (RG) that Network Address Translation (NAT) is part of remains in the standby mode on one device and active on a peer device. NAT in an RG that is in active mode translates the traffic according the configured translation rules.

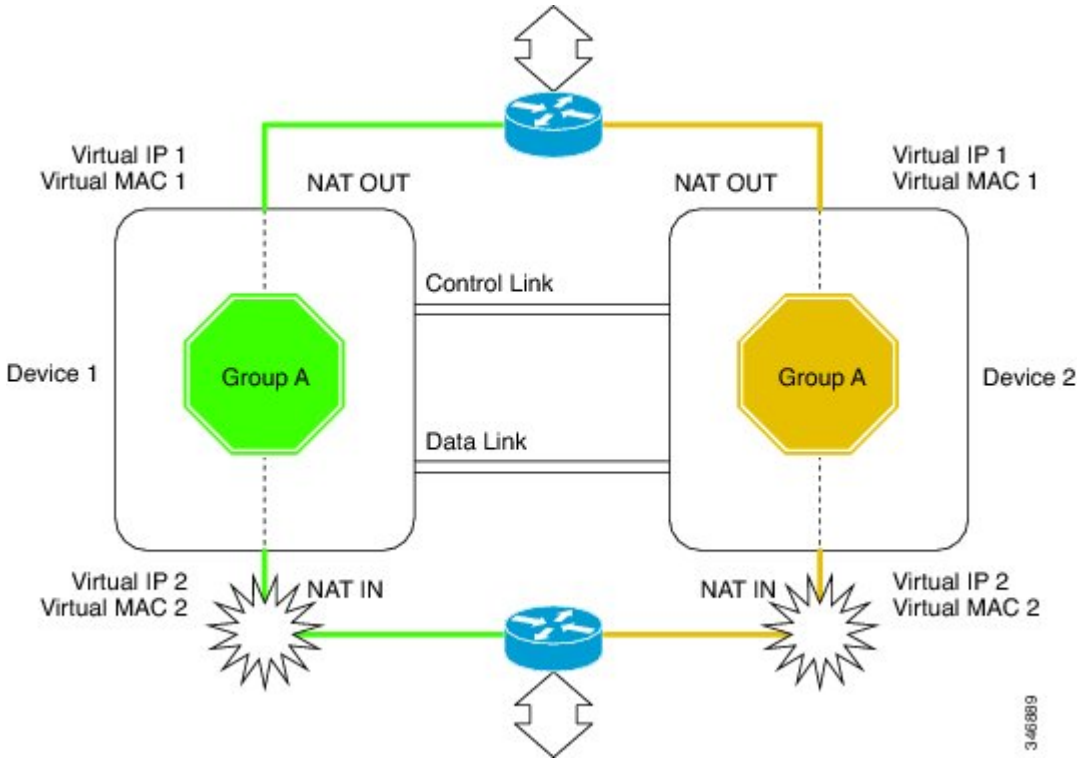
NAT does not actively perform any translations on the device where its RG is in the standby mode. In an RG, only one peer is in active mode at a given instance and the other peer is in standby mode. Applications that belong to the RG are active only on the device on which the RG is active. On all other devices, applications that belong to the RG are in the standby mode.

Note In a group of RG peers, only one peer can be active for a specific RG. Currently, the NAT Box-to-Box High-Availability Support feature supports only two peers in an RG and one RG in the RG infrastructure.

NAT Box-to-Box High-Availability Operation

The following figure illustrates the NAT box-to-box high-availability operation in a LAN-LAN topology. The green color represents an active device and the yellow color represents a standby device.

Figure 6: NAT Box-to-Box High Availability Operation

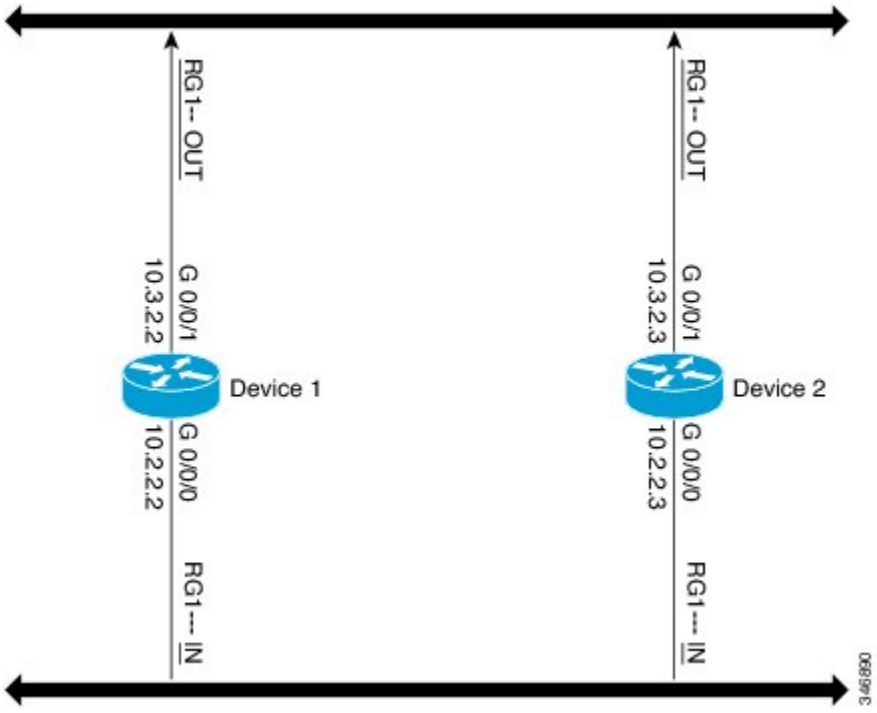


NAT Box-to-Box High-Availability LAN-LAN Topology

In a LAN-LAN topology, all participating devices are connected to each other through LAN interfaces on both the inside and the outside. The figure below shows the NAT box-to-box LAN-LAN topology. Network Address Translation (NAT) is in the active-standby mode and the peers are in one redundancy group (RG). All traffic or a subset of this traffic undergoes NAT translation.

Note Failover is caused by only those failures that the RG infrastructure listens to.

Figure 7: NAT Box-to-Box High-Availability LAN-LAN Topology

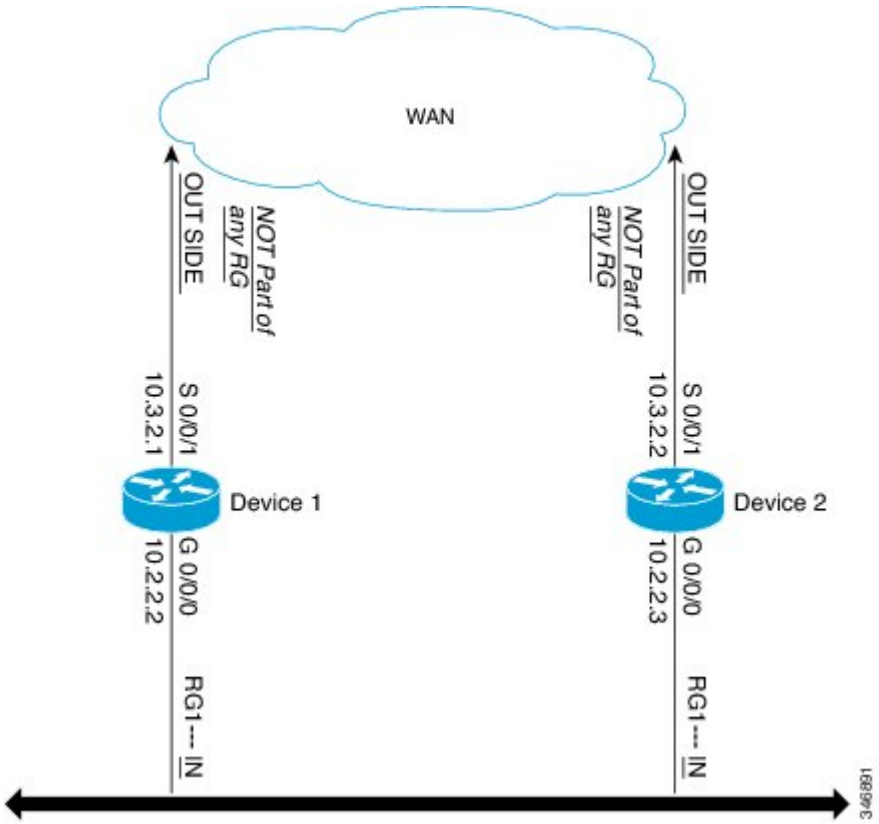


NAT Box-to-Box High-Availability WAN-LAN Topology

In a WAN-LAN topology, two 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. In most cases, WAN links are provided by different service providers. To utilize WAN links to the maximum, configure an external device to provide a failover.

In the following figure, inside interfaces are connected to a LAN while outside interfaces are connected to a WAN. The WAN interfaces cannot be made part of a redundancy group (RG) according to the current RG infrastructure. However, WAN interfaces may be configured in such a way that any failure on the WAN interfaces reduces the priority for the RG that is configured on that node, thereby triggering a failover.

Figure 8: NAT Box-to-Box High-Availability WAN-LAN Topology



Exclusive Virtual IP Addresses and Exclusive Virtual MAC Addresses

Virtual IP (VIP) addresses and virtual MAC (VMAC) addresses are used by security applications to control interfaces that receive traffic. An interface is paired with another interface, and these interfaces are associated with the same redundancy group (RG). The interface that is associated with an active RG exclusively owns the VIP and VMAC addresses.

The Address Resolution Protocol (ARP) process on the active device sends ARP replies for any ARP request for the VIP, and the Ethernet controller for the interface is programmed to receive packets destined for the VMAC.

When an RG failover occurs, the ownership of the VIP and VMAC changes. The interface that is associated with the newly active RG sends a gratuitous ARP message and programs the interface's Ethernet controller to accept packets destined for the VMAC.

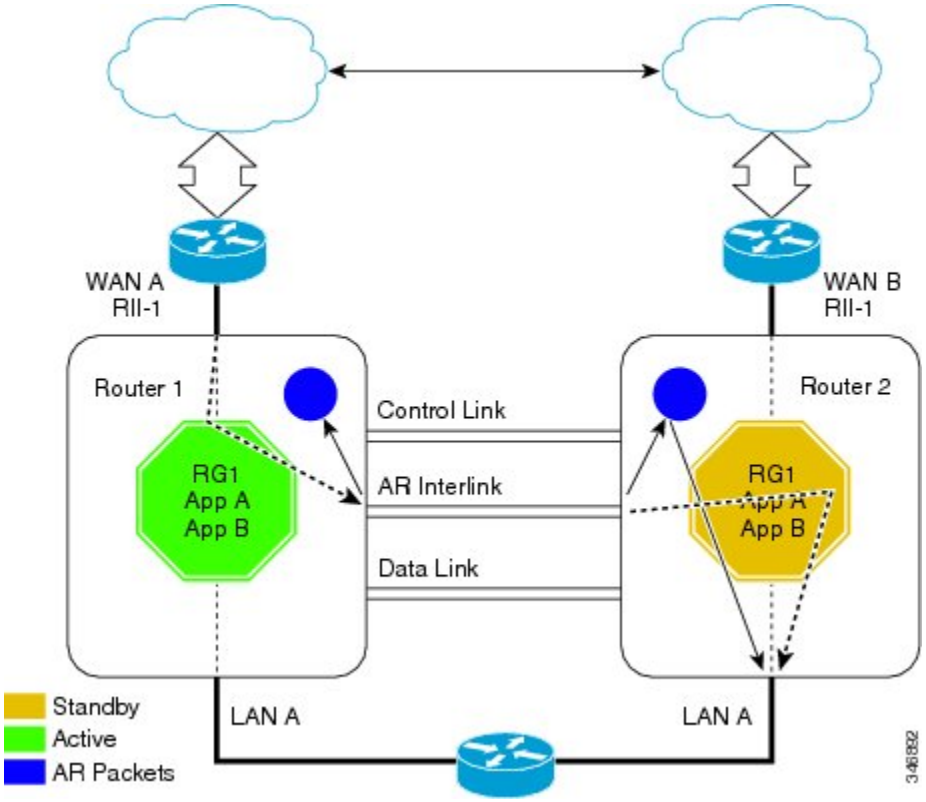
NAT Asymmetric Routing

In asymmetric routing, packets of a single connection or session flow through different routes in the forward and reverse directions. Asymmetric routing could occur due to link failures in the network, load balancing, a specific network configuration, and so on. Network Address Translation (NAT) provides session termination services and the associated dynamic session information. For NAT, if the return TCP segments are not forwarded to the same device that receives the initial synchronization (SYN) segment, the packet is dropped because it does not belong to any known session.

NAT Box-to-Box High Availability on Asymmetric-Routing Topology

The following figure shows asymmetrically routed packets being received on a standby device:

Figure 9: NAT Box-to-Box High Availability on Asymmetric-Routing Topology



Each routing device has an asymmetric routing (AR) module, which forwards the traffic received by the standby redundancy group (RG) using the module's AR interface. In the above illustration, the standby RG is RG1, on Router 1 with the Redundancy Interface Identifier (RII) configured as RII-1. The packet traffic that is received by RG1 is forwarded over the AR interface configured on Router 1 towards Router 2. This traffic is received by the AR module for RII-1 on Router 2 and is forwarded to RG1, which is active on Router 2.

Disabling NAT High Availability on Asymmetric-Routing Topology

When a packet ingresses Router 1 through the Redundancy Interface Identifier (RII), RII-1, Network Address Translation (NAT) identifies that packet as belonging to redundancy group (RG) RG1, which is in the standby state. If the asymmetric routing support is disabled, packets are not redirected to the active device by the standby device. Therefore, packets are dropped by default on the standby peer device.

Key Configuration Elements for NAT Box-to-Box High Availability Support

- Redundancy group (RG) Asymmetric Routing (AR) interface: A dedicated physical interface that provides connectivity between two peer devices. The redundancy infrastructure uses this interface to redirect AR packets from a standby device to an active device. The AR, control, and data interfaces can be configured on the same physical interface.

- Redundancy number: A unique identification number for each interface that is part of the RG infrastructure.
- RG priority: A numeric value that you can configure on the active or standby devices to control the switchover behavior. Each potential fault or error decrements the priority of the active device. The system switches over to the standby device when the priority value reaches the configured limit.
- RG control interface: A dedicated physical interface that provides connectivity between the two peer devices. The redundancy infrastructure uses this interface to exchange control information between the devices.
- RG data interface: A dedicated physical interface that provides connectivity between two peer devices. This interface is used by the redundancy infrastructure for data information exchange between devices, such as session information for NAT. Control and data interfaces can be configured on the same physical interface.
- Virtual IP address and virtual MAC address: The active device owns the virtual IP address and the virtual MAC address. Hosts or servers on the LAN that use the virtual IP address to reach the device which is currently in RG active state.
- RG decrement number: The priority value of an RG in local peer is decremented by the specified priority decrement number if the interface on which this configuration is applied goes down.
- RG infrastructure: Defines multiple RGs to which applications can subscribe and function in an active-standby mode across different routing devices. Currently, Network Address Translation (NAT) supports only one RG with an RG ID value of either 1 or 2.
- NAT mapping ID: A numeric value that is attached to all NAT rules that are associated to an RG. This value must be unique across different NAT rules and must be the same across NAT configurations on active and standby devices.

How to Configure NAT Box-to-Box High-Availability Support

- Perform configurations listed in this section on both the active and standby devices.
- The redundancy group (RG) ID must be the same for both devices.
- A unique redundancy interface identifier (RII) must be configured for each interface on a device that is part of the RG infrastructure.
- An RG ID and virtual IP address must be configured on each interface on a LAN.
- An RG ID and mapping ID must be configured for each Network Address Translation (NAT) statement.
- After configuring all NAT statements, you must enable RG.

Configuring a Redundancy Application Group

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**

4. **application redundancy**
5. **group** *id*
6. **name** *group-name*
7. **shutdown**
8. **priority** *value* [**failover threshold** *value*]
9. **preempt**
10. **track** *object-number* {**decrement** *value* | **shutdown**}
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`	Enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Enters redundancy application group configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name group1`	(Optional) Specifies an optional alias for the protocol instance.
Step 7	**shutdown** **Example:** `Device(config-red-app-grp)# shutdown`	(Optional) Shuts down a redundancy group manually.
Step 8	**priority** *value* [**failover threshold** *value*] **Example:** `Device(config-red-app-grp)# priority 100 failover threshold 50`	(Optional) Specifies the initial priority and failover threshold for a redundancy group.

	Command or Action	Purpose
Step 9	**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 the priority.
Step 10	**track** *object-number* {**decrement** *value* | **shutdown**} **Example:** `Device(config-red-app-grp)# track 200 decrement 200`	Specifies the priority value of a redundancy group that will be decremented if an event occurs.
Step 11	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group 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. This is an optional task. Perform this task only if you are configuring asymmetric routing for Network Address Translation (NAT).

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.
Step 6	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data GigabitEthernet 0/0/1`	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 1/0/0 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.

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

Enabling Data, Control and Asymmetric Routing Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **no shutdown**
6. **exit**
7. **interface** *type number*
8. **ip address** *ip-address mask*
9. **no shutdown**
10. **exit**
11. **interface** *type number*
12. **ip address** *ip-address mask*
13. **no shutdown**
14. **exit**

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 0/0/1`	Enters interface configuration mode for the data interface.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.2.3.2 255.255.255.0`	Assigns an IP address for the data interface.

	Command or Action	Purpose
Step 5	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Enables the interface.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/0`	Enters interface configuration mode for the control interface.
Step 8	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.10.2.5 255.255.255.255.0`	Assigns an IP address to the control interface.
Step 9	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Enables the interface.
Step 10	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	(Optional) Enters interface configuration mode for the asymmetric routing (AR) interface.
Step 12	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.5.1.5 255.255.255.255.0`	(Optional) Assigns an IP address to the AR interface.
Step 13	**no shutdown** **Example:** `Device(config-if)# no shutdown`	(Optional) Enables the interface.
Step 14	**exit** **Example:** `Device(config-if)# exit`	(Optional) Exits interface configuration mode and enters global configuration mode.

Configuring NAT Box-to-Box Interface Redundancy

Perform this task on the active and standby devices in the redundancy group to configure the Network Address Translation (NAT) box-to-box high-availability support.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ip nat inside**
6. **redundancy rii** *id*
7. **redundancy group** *id* **ip** *virtual-ip* [**exclusive**] [**decrement** *value*]
8. **exit**
9. **interface** *type number*
10. **ip address** *ip-address mask*
11. **ip nat outside**
12. **redundancy rii** *id* [**decrement** *number*]
13. **redundancy group** *id* **ip** *virtual-ip* [**exclusive**] [**decrement** *value*]
14. **exit**
15. **ip nat inside source static** *local-ip global-ip* [**redundancy** *rg-id* **mapping-id** *map-id*]
16. **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 2/0/2`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.1.27 255.255.255.0`	Assigns a virtual IP (VIP) address on the interface.
Step 5	**ip nat inside** **Example:**	Designates that traffic originating from the interface is subject to Network Address Translation (NAT).

	Command or Action	Purpose
	`Device(config-if)# ip nat inside`	
Step 6	**redundancy rii** *id* **Example:** `Device(config-if)# redundancy rii 100`	Configures a Redundancy Interface Identifier (RII) for redundancy group-protected traffic interfaces.
Step 7	**redundancy group** *id* **ip** *virtual-ip* [**exclusive**] [**decrement** *value*] **Example:** `Device(config-if)# redundancy group 1 ip 192.168.1.20 exclusive decrement 100`	Enables the redundancy group (RG) traffic interface configuration.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 9	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 10	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.5.54 255.255.255.255.0`	Assigns a virtual IP (VIP) address on the interface.
Step 11	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Designates that traffic destined for the interface is subject to NAT.
Step 12	**redundancy rii** *id* [**decrement** *number*] **Example:** `Device(config-if)# redundancy rii 101`	Configures an RII for redundancy group-protected traffic interfaces.
Step 13	**redundancy group** *id* **ip** *virtual-ip* [**exclusive**] [**decrement** *value*] **Example:** `Device(config-if)# redundancy group 1 ip 192.168.5.10 exclusive decrement 100`	Enables the redundancy group (RG) traffic interface configuration and specifies the decrement value number that is decremented from the priority when the state of the interface goes down.
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 15	**ip nat inside source static** *local-ip global-ip* [**redundancy** *rg-id* **mapping-id** *map-id*] **Example:**	Enables NAT redundancy of the inside source and associates the mapping ID to NAT high-availability redundancy.

	Command or Action	Purpose
	`Device(config)# ip nat inside source static 10.2.2.1 10.3.4.6 redundancy 1 mapping-id 120`	
Step 16	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuring Asymmetric Routing for NAT Box-to-Box High-Availability Support

Perform this task on the active and standby devices in the redundancy group to configure asymmetric routing support on Network Address Translation (NAT) Box-to-Box high availability.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **ip nat outside**
6. **redundancy rii** *id* [**decrement** *number*]
7. **redundancy asymmetric routing enable**
8. **exit**
9. **ip nat inside source static** *local-ip global-ip* [**redundancy** *RG-id* **mapping-id** *map-id*]
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 serial 0/0/1`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.1.27 255.255.255.0`	Assigns a virtual IP (VIP) address on the interface.

	Command or Action	Purpose
Step 5	**ip nat outside** **Example:** `Device(config-if)# ip nat outside`	Designates that traffic destined for the interface is subject to Network Address Translation (NAT).
Step 6	**redundancy rii** *id* [**decrement** *number*] **Example:** `Device(config-if)# redundancy rii 101`	Configures a Redundancy Interface Identifier (RII) for redundancy group-protected traffic interfaces.
Step 7	**redundancy asymmetric routing enable** **Example:** `Device(config-if)# redundancy asymmetric-routing enable`	Establishes an asymmetric flow diversion tunnel for each redundancy group (RG).
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 9	**ip nat inside source static** *local-ip global-ip* [**redundancy** *RG-id* **mapping-id** *map-id*] **Example:** `Device(config)# ip nat inside source static 10.2.2.1 10.3.4.6 redundancy 1 mapping-id 120`	Enables NAT redundancy of the inside source and associates the mapping ID to NAT high-availability redundancy.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and enters privileged EXEC mode.

Configuration Examples for NAT Box-to-Box High-Availability Support

Example: Configuring a Redundancy Application Group

The following example shows how to configure a redundancy group named group1 with priority and preempt attributes:

```
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 200 decrement 200
Device(config-red-app-grp)# 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 0/0/1
Device(config-red-app-grp)# control GigabitEthernet 1/0/0 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: Enabling Data, Control and Asymmetric Routing Interfaces

```
Device# configure terminal
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# ip address 10.2.3.2 255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# exit
Device(config)# interface Gigabitethernet 1/0/0
Device(config-if)# ip address 10.10.2.5 255.255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# exit
Device(config)# interface Gigabitethernet 0/1/1
Device(config-if)# ip address 10.5.1.5 255.255.255.255.0
Device(config-if)# no shutdown
Device(config-if)# end
```

Example: Configuring a NAT Box-to-Box High-Availability Support

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 2/0/2
Device(config-if)# ip address 192.168.1.27 255.255.255.0
Device(config-if)# ip nat inside
Device(config-if)# redundancy rii 100
Device(config-if)# redundancy group 1 ip 192.168.1.20 exclusive decrement 100
Device(config-if)# exit
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# ip address 192.168.5.54 255.255.255.255.0
Device(config-if)# ip nat outside
Device(config-if)# redundancy rii 101
Device(config-if)# redundancy group 1 ip 192.168.5.10 exclusive decrement 100
Device(config-if)# exit
Device(config)# ip nat inside source static 10.2.2.1 10.3.4.6 redundancy 1 mapping-id 120
Device(config-if)# end
```

Example: Configuring Asymmetric Routing for NAT Box-to-Box High-Availability Support

```
Device> enable
Device# configure terminal
Device(config)# interface serial 0/0/1
Device(config-if)# ip address 192.168.1.27 255.255.255.0
Device(config-if)# ip nat outside
Device(config-if)# redundancy rii 101
Device(config-if)# exit
Device(config)# ip nat inside source static 10.2.2.1 10.3.4.6 redundancy 1 mapping-id 120
Device(config-if)# end
```

Additional References for NAT Box-to-Box High-Availability Support

Related Documents

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

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

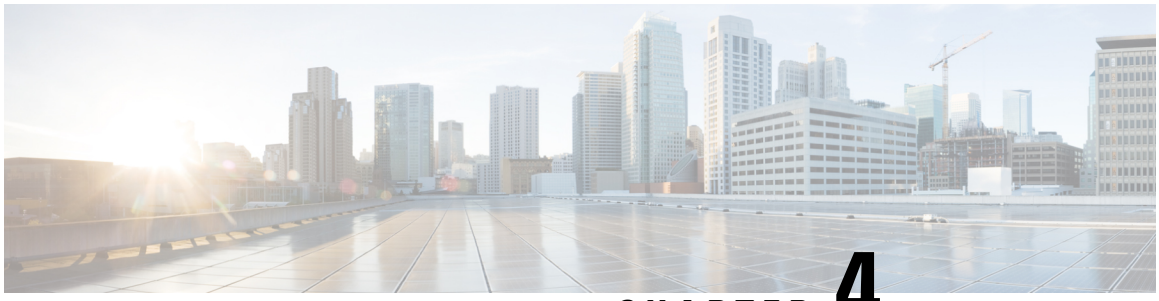
Feature Information for NAT Box-to-Box High-Availability Support

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 NAT Box-to-Box High Availability Support

Feature Name	Releases	Feature Configuration Information
NAT Box-to-Box High Availability Support	15.3(2)T	NAT Box-to-Box High-Availability Support feature makes an IP network more resilient to potential link and routing device failures at the Network Address Translation (NAT) border. NAT box-to-box high-availability functionality is achieved when you configure two NAT translators that reside across different devices as part of a redundancy group (RG) and function as a translation group. One member of the translation group acts as an active translator and the other member in the group acts as a standby translator. The standby translator takes over as the active translator in the event of any failures to the current active translator. The following commands were introduced or modified: **ip nat inside source**, **ip nat outside source**, **show ip nat redundancy**, **show ip nat translations redundancy**, **show redundancy application group**.



CHAPTER 4

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.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.

In Cisco IOS-XE release 17.4 release, support is introduced to map a VRF to an IPv4 to IPv6 prefix mapping. Multiple source and destination prefix can be mapped to a VRF.

The Stateless NAT64 translator does not maintain any state information in the datapath.

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.

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

Restrictions for Stateless Network Address Translation 64

The following restrictions apply to the Stateless NAT64 feature:

- Multiple prefixes are not supported.
- IPv4 and IPv6 virtual routing and forwarding (VRFs) instances are not supported.
- Redundancy is not supported.
- Applications without a corresponding application layer gateway (ALG) may not work properly with the Stateless NAT64 translator.
- Only valid IPv4-translatable addresses can be used for stateless translation.
- Multicast is not supported.
- 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.

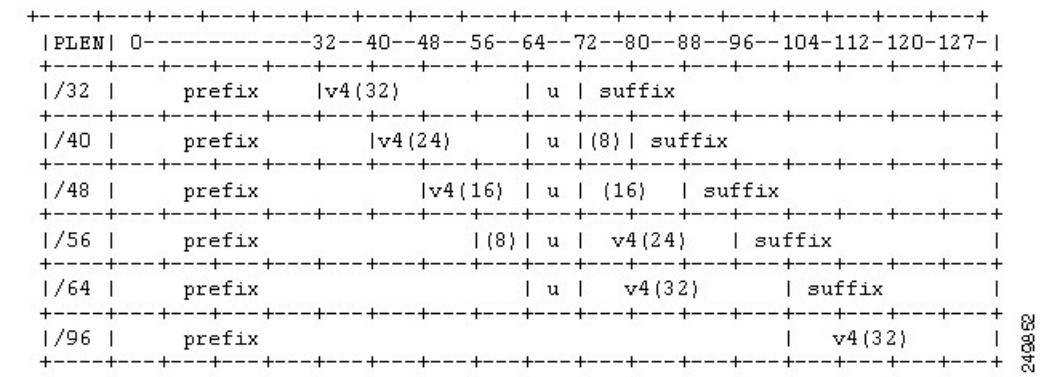
Information About Stateless Network Address Translation 64

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 10: IPv4-Translatable IPv6 Address Format



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.

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.

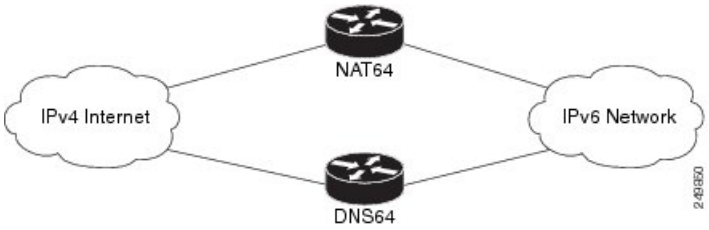
Supported Stateless NAT64 Scenarios

The 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.

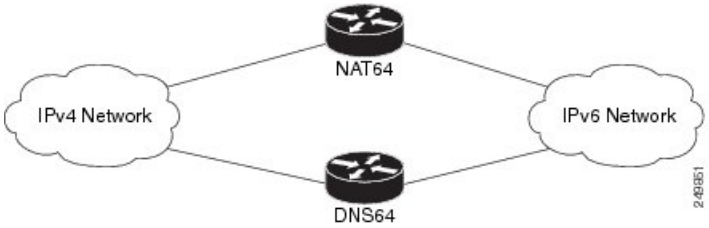
Figure 11: Stateless Translation for Scenarios 1 and 2



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 12: 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.

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. You can configure stateless NAT64 along with your NAT configuration: static, dynamic, or overload.

Before you begin

- 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:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
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::1/128`	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 Stateless NAT64 translation on an IPv6 interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:**	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface gigabitethernet 1/2/0`	
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 198.51.100.1 255.255.255.0`	Configures an IPv4 address for an interface.
Step 13	**nat64 enable** **Example:** `Device(config-if)# nat64 enable`	Enables Stateless NAT64 translation on an IPv4 interface.
Step 14	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 15	**nat64 prefix stateless** *ipv6-prefix/length* **Example:** `Device(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.
Step 16	**nat64 route** *ipv4-prefix/mask interface-type interface-number* **Example:** `Device(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:** `Device(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:** `Device(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:

```
Device# 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:

```
Device# 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:

```
Device# 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:

```
Device# debug nat64 statistics
```

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:

```
Device# 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
```

Additional References for Stateless Network Address Translation 64

Related Documents

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

Standards and RFCs

Standard/RFC	Document Title
RFC 6052	IPv6 Addressing of IPv4/IPv6 Translators
RFC 6144	Framework for IPv4/IPv6 Translation
RFC 6145	IP/ICMP Translation Algorithm

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 5: Feature Information for Stateless Network Address Translation 64

Feature Name	Releases	Feature Information
Stateless Network Address Translation 64	15.4(1)T	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.

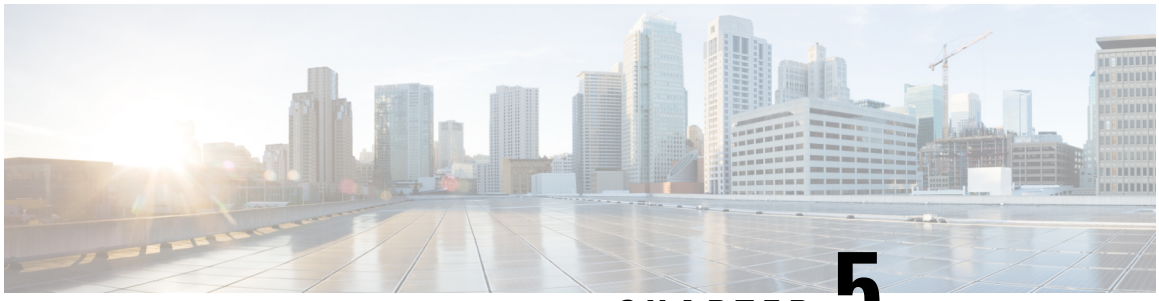
IPv6-converted address—IPv6 addresses that are assigned to the IPv6 hosts for the stateless translator. These IPv6-converted addresses have an explicit mapping relationship to the IPv4 addresses. This relationship is self-described by mapping the IPv4 address in the IPv6 address. The stateless translator 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.



CHAPTER 5

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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. 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.
- 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.

- Only TCP and UDP Layer 4 protocols are supported for header translation.
- Routemaps are not supported.
- Application-level gateways (ALGs) FTP and ICMP are not supported.
- In the absence of a pre-existing state in NAT 64, stateful translation only supports IPv6-initiated sessions.
- If a static mapping host-binding entry exists for an IPv6 host, the IPv4 nodes can initiate communication. In dynamic mapping, IPv4 nodes can initiate communication only if a host-binding entry is created for the IPv6 host through a previously established connection to the same or a different IPv4 host.

  Dynamic mapping rules that use Port-Address Translation (PAT), host-binding entries cannot be created because IPv4-initiated communication not possible through PAT.

- Both NAT44 (static, dynamic and PAT) configuration and stateful NAT64 configuration are not supported on the same interface.

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 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 NAT overload or Port Address Translation (PAT) provides a 1:*n* mapping between IPv4 and IPv6 addresses.

When an IPv6 node initiates traffic through Stateful NAT64, and the incoming packet does not have an existing state and 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.

Supported Stateful NAT64 Scenarios

The following scenarios are supported by the Stateful NAT64 feature and are described in this section:

- Scenario 1—an IPv6 network to the IPv4 Internet
- Scenario 3—an IPv6 Internet to an IPv4 network
- Scenario 5—an IPv6 network to an IPv4 network

Scenario 1

An IPv6-only network that communicates with a global IPv4 Internet. This type of network is also called a green-field network. In a green-field enterprise network only the the border between its network and the IPv4 Internet can be modified.

Translation is performed between IPv4 and IPv6 packets in unidirectional or bidirectional flows that are initiated from an IPv6 host towards an IPv4 host. Port translation is necessary on the IPv4 side for efficient IPv4 address usage. The stateful translator can service an IPv6 network of any size.

Both Stateful NAT64 and Stateless NAT64 support Scenario 1.

Scenario 3

Scenario 3 shows a legacy IPv4 network that provide services to IPv6 hosts. IPv6-initiated communication can be achieved through stateful translation in this scenario.

Translation is preformed between IPv4 and IPv6 packets in unidirectional or bidirectional flows that are initiated from an IPv6 host towards an IPv4 host. The stateful translator can service an IPv4 network using either private or public IPv4 addresses.

Note Do not use the Well-Known Prefix (WKP) for Scenario 3, because it would lead to using the WKP with non-global IPv4 addresses. Use a network-specific prefix (example, /96 prefix) in Scenario 3. For more information, see *RFC 6052*, section "3.4 Choice of Prefix for Stateful Translation Deployments"

Scenario 5

This scenario has an IPv4 and IPv6 network within the same organization. The IPv4 addresses used are either public IPv4 addresses or RFC 1918-compliant addresses. IPv6 addresses are either public IPv6 addresses or Unique Local Addresses (ULAs) as specified by RFC 4193.

Translation is performed between IPv6 and IPv4 packets in unidirectional or bidirectional flows that are initiated from an IPv6 host towards an IPv4 host. The stateful translator can service both IPv6 and IPv4 networks of any size; however neither networks should not be the Internet.

Both Stateful NAT64 and Stateless NAT64 support Scenario 5.

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

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.

- 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.

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

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**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**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.
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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.
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:**	Enables NAT64 IPv6-to-IPv4 address mapping.

	Command or Action	Purpose
	`Device(config)# nat64 prefix stateful 2001:DB8:1::1/96`	
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.
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.

	Command or Action	Purpose
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.
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

```
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
RFC 4291	*IP Version 6 Addressing Architecture*
RFC 6144	*Framework for IPv4/IPv6 Translation*
RFC 6052	*IPv6 Addressing of IPv4/IPv6 Translators*

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/support

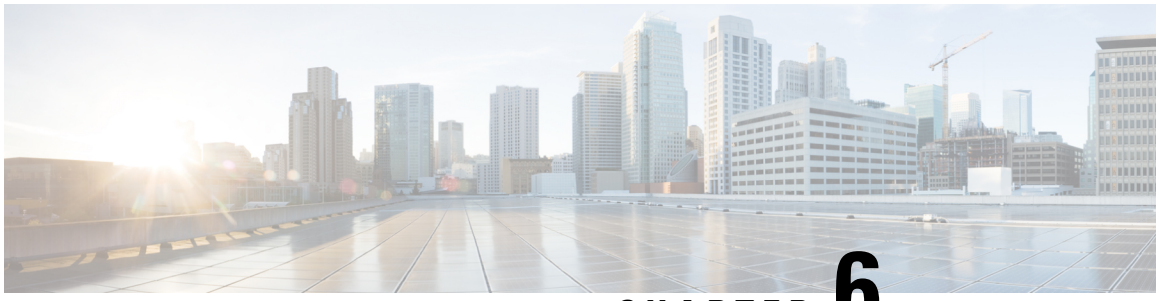
Feature Information for Stateful Network Address Translation 64

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 6: Feature Information for Stateful Network Address Translation 64

Feature Name	Releases	Feature Information
Stateful Network Address Translation 64	15.4(2)T	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**.



CHAPTER 6

Mapping of Address and Port Using Translation

The Mapping of Address and Port Using Translation feature provides connectivity to IPv4 hosts across IPv6 domains. Mapping of address and port using translation (MAP-T) is a mechanism that performs double translation (IPv4 to IPv6 and vice versa) on customer edge (CE) devices and border routers.

This module provides an overview of MAP-T and explains how to configure this feature.

Restrictions for Mapping of Address and Port Using Translation

- The mapping of address and port using translation (MAP-T) customer edge (CE) functionality is not supported.
- In Cisco IOS XE Denali 16.2 release, the support for MAP-T domains were extended to 10000 domains. For releases prior to Cisco IOS XE Denali 16.2, a maximum of 128 MAP-T domains are supported.
- Forwarding mapping rule (FMR) is not supported.

Information About Mapping of Address and Port Using Translation

Mapping of Address and Port Using Translation Overview

The Mapping of Address and Port Using Translation feature provides connectivity to IPv4 hosts across IPv6 domains. Mapping of address and port using translation (MAP-T) builds on the existing stateless IPv4 and IPv6 address translation techniques that are specified in RFCs 6052, 6144, and 6145.

MAP-T is a mechanism that performs double translation (IPv4 to IPv6 and vice versa) on customer edge (CE) devices and border routers. The Mapping of Address and Port Using Translation feature supports only the MAP-T border router functionality. This feature does not support the MAP-T CE functionality.

The Mapping of Address and Port Using Translation feature leverages the Network Address Translation 64 (NAT64) translation engine and adds the MAP-T border router function to the NAT64 stateless function. MAP-T is enabled on IPv4 and IPv6 interfaces. MAP-T uses IPv4 and IPv6 forwarding, IPv4 and IPv6 fragmentation functions, and NAT64 translation functions. A MAP-T domain is one or more MAP CE devices and a border router, all connected to the same IPv6 network.

A MAP-T CE device connects a user's private IPv4 address and the native IPv6 network to the IPv6-only MAP-T domain. The MAP-T border router uses the stateless IPv4/IPv6 translation to connect external IPv4 networks to all devices available in the one or more MAP-T domains. MAP-T requires only one IPv6 prefix per network and supports the regular IPv6 prefix/address assignment mechanisms. The MAP-T domain contains regular IPv6-only hosts or servers that have an IPv4-translatable IPv6 address. MAP-T does not require the operation of an IPv4 overlay network or the introduction of a non-native-IPv6 network device or server functionality.

A MAP-T configuration provides the following features:

- Retains the ability for IPv4 end hosts to communicate across the IPv6 domain with other IPv4 hosts.
- Permits both individual IPv4 address assignment and IPv4 address sharing with a predefined port range.
- Allows communication between IPv4-only and IPv6-enabled end hosts and native IPv6-only servers in domains that use IPv4-translatable IPv6 addresses.
- Allows the use of IPv6 native network operations, including the ability to classify IP traffic and perform IP traffic routing optimization policies such as routing optimization based on peering policies for IPv4 destinations outside the domain.

MAP-T Mapping Rules

Mapping rules define the mapping between an IPv4 prefix and an IPv4 address or between a shared IPv4 address and an IPv6 prefix/address. Each mapping of address and port using translation (MAP-T) domain uses a different mapping rule.

A MAP-T configuration has one basic mapping rule (BMR), one default mapping rule (DMR), and one or more forwarding mapping rules (FMRs) for each MAP-T domain. You must configure the DMR before configuring the BMR for a MAP-T domain.

The three types of mapping rules are described below:

- A BMR configures the MAP IPv6 address or prefix. The basic mapping rule is configured for the source address prefix. You can configure only one basic mapping rule per IPv6 prefix. The basic mapping rule is used by the MAP-T CE to configure itself with an IPv4 address, an IPv4 prefix, or a shared IPv4 address from an IPv6 prefix. The basic mapping rule can also be used for forwarding packets, where an IPv4 destination address and a destination port are mapped into an IPv6 address/prefix. Every MAP-T node (a CE device is a MAP-T node) must be provisioned with a basic mapping rule. You can use the **port-parameters** command to configure port parameters for the MAP-T BMR.
- A DMR is a mandatory rule that is used for mapping IPv4 information to IPv6 addresses for destinations outside a MAP-T domain. A 0.0.0.0/0 entry is automatically configured in the MAP rule table (MRT) for this rule.

- An FMR is used for forwarding packets. Each FMR results in an entry in the MRT for the rule IPv4 prefix. FMR is an optional rule for mapping IPv4 and IPv6 destinations within a MAP-T domain.

Note FMR is not supported by the Mapping of Address and Port Using Translation feature.

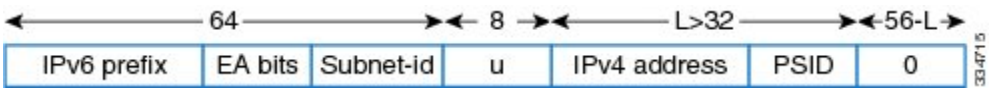
MAP-T Address Formats

The mapping of address and port using translation (MAP-T) customer edge (CE) device address format is defined by the IETF draft Mapping of Address and Port (MAP) . Address formats are used during mapping rule operations to construct the source and destination IPv6 addresses.

Note Forwarding mapping rule (FMR) is not supported by the Mapping of Address and Port Using Translation feature.

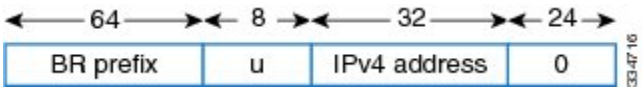
The figure below shows the mapped CE address format as defined in MAP-T configuration. This address format is used in basic mapping rule (BMR) and FMR operations.

Figure 13: IPv4-Translatable Address for BMR and FMR



The figure below shows the address format used by the MAP-T default mapping rule (DMR), an IPv4-translated address that is specific to MAP-T configuration.

Figure 14: IPv4-Translated Address for DMR



Packet Forwarding in MAP-T Customer Edge Devices

Note The Mapping of Address and Port Using Translation feature does not support the MAP-T customer edge (CE) functionality. The CE functionality is provided by third-party devices.

IPv4-to-IPv6 Packet Forwarding

A mapping of address and port using translation (MAP-T) CE device that receives IPv4 packets performs Network Address Translation (NAT) and creates appropriate NAT stateful bindings. The resulting IPv4 packets contain the source IPv4 address and the source transport number defined by MAP-T. This IPv4 packet is forwarded to the CE's MAP-T, which performs IPv4-to-IPv6 stateless translation. IPv6 source and destination addresses are then derived by the MAP-T translation, and IPv4 headers are replaced with IPv6 headers.

IPv6-to-IPv4 Packet Forwarding

A MAP-T CE device that receives an IPv6 packet performs its regular IPv6 operations. Only the packets that are addressed to the basic mapping rule (BMR) address are sent to the CE's MAP-T. All other IPv6 traffic is forwarded based on the IPv6 routing rules on the CE device. The CE device checks if the transport-layer destination port number of the packets received from MAP-T is in the range that was configured and forwards packets that confirm to the port number. The CE device drops all nonconforming packets and responds with an Internet Control Message Protocol Version 6 (ICMPv6) "Address Unreachable" message.

Packet Forwarding in Border Routers

IPv4-to-IPv6 Packet Forwarding

An incoming IPv4 packet is processed by the IPv4 input interface, and the destination route lookup routes the IPv4 packet to the mapping of address and port using translation (MAP-T) virtual interface. The border router compares the packet against the IPv4 prefix lookup unit (PLU) tree to obtain the corresponding basic mapping rule (BMR), the default mapping rule (DMR), and the forwarding mapping rule (FMR). Based on the BMR or FMR rules, the border router constructs the IPv6 destination address by encoding the embedded address (EA) bits and adding a suffix. The IPv6 source address is constructed from the DMR rule.

After the IPv6 source and destination addresses are constructed, the packet uses the Network Address Translation 64 (NAT64) IPv4-to-IPv6 translation to construct the IPv6 packet. A routing lookup is done on the IPv6 packet, and the packet is forwarded to the IPv6 egress interface for processing and transmission.

IPv6-to-IPv4 Packet Forwarding

An incoming IPv6 packet is processed by the IPv6 input interface, and the destination route lookup routes the IPv6 packet to the MAP-T virtual interface. The software compares the packet against the IPv6 PLU tree to obtain the corresponding BMR, DMR, and FMR rules. The border router checks whether the port-set ID (PSID) and the port set match. If the port-set ID and port set match, the DMR rule matches the packet destination of the IPv6 packet. Based on the BMR and FMR, the border router constructs the IPv4 source address and extracts the IPv4 destination address from the IPv6 destination address. The IPv6 packet uses the NAT64 IPv6-to-IPv4 translation engine to construct the IPv4 packet from the IPv6 packet. A routing lookup is done on the IPv4 packet, and the IPv4 packet is forwarded to the IPv4 egress interface for processing and transmission.

ICMP/ICMPv6 Header Translation for MAP-T

Mapping of address and port using translation (MAP-T) customer edge (CE) devices and border routers use the ICMP/ICMPv6 translation for address sharing of port ranges.

Unlike TCP and UDP, which provide two port fields to represent source and destination addresses, the Internet Control Message Protocol (ICMP) and ICMP Version 6 (ICMPv6) query message headers have only one ID field.

When an ICMP query message originates from an IPv4 host that exists beyond a MAP-T CE device, the ICMP ID field is exclusively used to identify the IPv4 host. The MAP-T CE device rewrites the ID field to a port-set value that is obtained through the basic mapping rule (BMR) during the IPv4-to-IPv6 translation, and the border router translates ICMPv6 packets to ICMP.

When a MAP-T border router receives an ICMP packet that contains an ID field that is bound for a shared address in the MAP-T domain, the MAP-T border router uses the ID field as a substitute for the destination port to determine the IPv6 destination address. The border router derives the destination IPv6 address by

mapping the destination IPv4 address without the port information for packets that do not contain the ID field, and the corresponding CE device translates the ICMPv6 packets to ICMP.

Path MTU Discovery and Fragmentation in MAP-T

Mapping of address and port using translation (MAP-T) uses path maximum transmission unit (MTU) discovery and fragmentation for IPv4-to-IPv6 translation because the size of IPv4 (more than 20 octets) and IPv6 (40 octets) headers is different. The MTU defines the largest size of a packet that an interface can transmit without the need to fragment the packet. IP packets larger than the MTU must go through IP fragmentation procedures.

When an IPv4 node performs path MTU discovery by setting the Don't Fragment (DF) bit in the packet header, path MTU discovery operates end-to-end across the MAP-T border router and customer edge (CE) translators. During IPv4 path MTU discovery, either the IPv4 device or the IPv6 device can send ICMP “Packet Too Big” messages to the sender. When IPv6 devices send these messages as Internet Control Message Protocol Version 6 (ICMPv6) errors, the packets that follow the message pass through the translator and result in an appropriate ICMP error message sent to the IPv4 sender.

When the IPv4 sender does not set the DF bit, the translator fragments the IPv4 packet and includes the packet with fragment headers to fit the packet in the minimum MTU 1280-byte IPv6 packets. When packets are fragmented, either by the sender or by IPv4 devices, the low-order 16 bits of the fragment identification are carried end-to-end across the MAP-T domain to ensure that packets are reassembled correctly.

How to Configure Mapping of Address and Port Using Translation

Configuring Mapping of Address and Port Using Translation

Before you begin

Prerequisites:

- Configure the **ipv6 enable** command on interfaces on which you configure the Mapping of Address and Port Using Translation feature.
- Configure the default mapping rule before you configure the basic mapping rule.
- While configuring mapping of address and port using translation (MAP-T), the default mapping rule (DMR) prefix, the IPv6 user prefix, and the IPv6 prefix plus the embedded address (EA) bits must be less than or equal to 64 bits, and the share ratio plus the contiguous ports plus the start port must be 16 bits.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **nat64 map-t domain** *number*
4. **default-mapping-rule** *ipv6-prefix/prefix-length*
5. **basic-mapping-rule**

6. **ipv6-prefix** *prefix/length*
7. **ipv4-prefix** *prefix/length*
8. **port-parameters share-ratio** *ratio* [**start-port** *port-number*]
9. **end**
10. **show nat64 map-t domain** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter you password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**nat64 map-t domain** *number* **Example:** `Device(config)# nat64 map-t domain 1`	Configures the Network Address Translation 64 (NAT64) mapping of address and port using translation (MAP-T) domain and enters NAT64 MAP-T configuration mode.
Step 4	**default-mapping-rule** *ipv6-prefix/prefix-length* **Example:** `Device(config-nat64-mapt)# default-mapping-rule 2001:DA8:B001:FFFF::/64`	Configures the default domain mapping rule for the MAP-T domain.
Step 5	**basic-mapping-rule** **Example:** `Device(config-nat64-mapt)# basic-mapping-rule`	Configures the basic mapping rule (BMR) for the MAP-T domain and enters NAT64 MAP-T BMR configuration mode.
Step 6	**ipv6-prefix** *prefix/length* **Example:** `Device(config-nat64-mapt-bmr)# ipv6-prefix 2001:DA8:B001::/56`	Configures an IPv6 address and prefix for the MAP-T BMR.
Step 7	**ipv4-prefix** *prefix/length* **Example:** `Device(config-nat64-mapt-bmr)# ipv4-prefix 202.1.0.128/28`	Configures an IPv4 address and prefix for the MAP-T BMR.
Step 8	**port-parameters share-ratio** *ratio* [**start-port** *port-number*] **Example:** `Device(config-nat64-mapt-bmr)# port-parameters share-ratio 16 start-port 1024`	Configures port parameters for the MAP-T BMR.

	Command or Action	Purpose
Step 9	**end** **Example:** `Device(config-nat64-mapt-bmr)# end`	Exits NAT64 MAP-T BMR configuration mode and returns to privileged EXEC mode.
Step 10	**show nat64 map-t domain** *number* **Example:** `Device# show nat64 map-t domain 1`	Displays MAP-T domain information.

Example:

The following is sample output from the **show nat64 map-t domain** command:

```
Device# show nat64 map-t domain 1

MAP-T Domain 1
Mode MAP-T
Default-mapping-rule
Ip-v6-prefix 2001:DA8:B001:FFFF::/64
Basic-mapping-rule
Ip-v6-prefix 2001:DA8:B001::/56
Ip-v4-prefix 202.1.0.128/28
Port-parameters
Share-ratio 16 Contiguous-ports 64 Start-port 1024
Share-ratio-bits 4 Contiguous-ports-bits 6 Port-offset-bits 6
```

Configuration Examples for Mapping of Address and Port Using Translation

Example: Configuring Mapping of Address and Port Using Translation

```
Device# configure terminal
Device(config)# nat64 map-t domain 1
Device(config-nat64-mapt)# $ping-rule 2001:DA8:B001:FFFF::/64
Device(config-nat64-mapt)# basic-mapping-rule
Device(config-nat64-mapt-bmr)# ipv6-prefix 2001:DA8:B001::/56
Device(config-nat64-mapt-bmr)# ipv4-prefix 202.1.0.128/28
Device(config-nat64-mapt-bmr)# $ters share-ratio 16 start-port 1024
Device(config-nat64-mapt-bmr)# end
```

Example: MAP-T Deployment Scenario

The following illustration shows a mapping of address and port using translation (MAP-T) deployment scenario.

The following is the configuration for the MAP-T deployment scenario:

```
Device(config)# nat64 map-t
Device(config)# nat64 map-t domain 1
Device(config-nat64-mapt)# $ping-rule 2001:DA8:B001:FFFF::/64
Device(config-nat64-mapt)# basic-mapping-rule
Device(config-nat64-mapt-bmr)# ipv6-prefix 2001:DA8:B001::/56
Device(config-nat64-mapt-bmr)# ipv4-prefix 202.1.0.128/28
Device(config-nat64-mapt-bmr)# $ters share-ratio 16 start-port 1024
Device(config-nat64-mapt-bmr)# end
```

At the PC:

An IPv4 packet goes from 202.1.0.130 to 11.1.1.1. At the customer edge (CE) device the Mapping of address and port mapping using translation (MAP-T) function translates the packet to Src: 2201:DA8:B001:2E:0:CA01:82:E00 Dest: 2001:DA8:B001:FFFF:B:0101:0100:0.

At the border router the MAP-T border router translates the packet to

Packet goes from 192.168.1.2 ---> 74.1.1.1, source 4000, destination port : 5000

At the CPE the MAP-T CE function translates the

packet to Src: 2201:DA8:B001:2E:0:CA01:82:E00 Dest: 2001:DA8:B001:FFFF:B:0101:0100:0.

At the BR the MAP-T BR function translates the packet to

Src:203.38.102.130 Dst:74.1.1.1 SrcPort:4000 DstPort:5000

From End device:

Src:74.1.1.1 Dst:203.38.102.130 SrcPort:4000 DstPort:5000

At the BR the MAP-T BR function translates the packet to

Src: 2201:DA8:B001:2E:0:CA01:82:E00 Dest: 2001:DA8:B001:FFFF:B:0101:0100:0.

At the CE the MAP-T CE function translates the packet from

Src: 2201:DA8:B001:2E:0:CA01:82:E00 Dest: 2001:DA8:B001:FFFF:B:0101:0100:0.

To

Src:74.1.1.1 Dst:203.38.102.130 SrcPort:4000 Dstport:5000

Additional References for Mapping of Address and Port Using Translation

Related Documents

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

Standards and RFCs

Standard/RFC	Title
MAP	Mapping of Address and Port (MAP)
MAP Translation	MAP Translation (MAP-T) - specification
RFC 6052	IPv6 Addressing of IPv4/IPv6 Translators
RFC 6144	Framework for IPv4/IPv6 Translation
RFC 6145	IP/ICMP Translation Algorithm

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 Mapping of Address and Port Using Translation

Glossary

EA bits—Embedded address bits. The IPv4 EA bits in the IPv6 address identify an IPv4 prefix/address (or part thereof) or a shared IPv4 address (or part thereof) and a port-set identifier.

IP fragmentation—The process of breaking a datagram into a number of pieces that can be reassembled later. The IP source, destination, identification, total length, and fragment offset fields, along with the More fragments and Don't Fragment (DF) flags in the IP header, are used for IP fragmentation and reassembly. A DF bit is a bit within the IP header that determines whether a device is allowed to fragment a packet.

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

IPv6-translatable address—IPv6 addresses that are assigned to IPv6 hosts for stateless translation. These IPv6-translatable addresses (also called IPv6-converted addresses) have an explicit mapping relationship to IPv4 addresses. This relationship is self-described by mapping the IPv4 address in the IPv6 address. The stateless translator uses corresponding IPv4 addresses to represent IPv6 hosts. The stateful translator does not

use IPv6-translatable addresses because IPv6 hosts are represented by the IPv4 address pool in the translator via dynamic states.

MAP rule—A set of parameters that define the mapping between an IPv4 prefix, an IPv4 address or a shared IPv4 address, and an IPv6 prefix or address. Each MAP domain uses a different mapping rule set.

MAP-T border router—A mapping of address and port using translation (MAP-T)-enabled router or translator at the edge of a MAP domain that provides connectivity to the MAP-T domain. A border relay router has at least one IPv6-enabled interface and one IPv4 interface connected to the native IPv4 network, and this router can serve multiple MAP-T domains.

MAP-T CE—A device that functions as a customer edge (CE) router in a MAP-T deployment. A typical MAP-T CE device that adopts MAP rules serves a residential site with one WAN-side interface and one or more LAN-side interfaces. A MAP-T CE device can also be referred to as a "CE" within the context of a MAP-T domain.

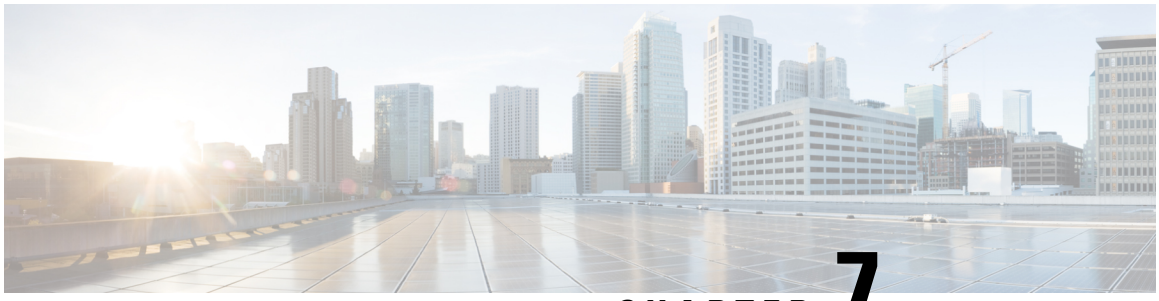
MAP-T domain—Mapping of address and port using translation (MAP-T) domain. One or more customer edge (CE) devices and a border router, all connected to the same IPv6 network. A service provider may deploy a single MAP-T domain or use multiple MAP domains.

MRT—MAP rule table. Address and port-aware data structure that supports the longest match lookups. The MRT is used by the MAP-T forwarding function.

path MTU—Path maximum transmission unit (MTU) discovery prevents fragmentation in the path between endpoints. Path MTU discovery is used to dynamically determine the lowest MTU along the path from a packet's source to its destination. Path MTU discovery is supported only by TCP and UDP. Path MTU discovery is mandatory in IPv6, but it is optional in IPv4. IPv6 devices never fragment a packet—only the sender can fragment packets.

stateful translation—Creates a per-flow state 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 enables IPv6 clients and peers without mapped IPv4 addresses to connect to IPv4-only servers and peers.

stateless translation—A translation algorithm that is not stateful. A stateless translation requires configuring a static translation table or may derive information algorithmically from the messages that 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 IPv4-only clients and peers to initiate connections to 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.



CHAPTER 7

Mapping of Address and Port Using Encapsulation

The MAP-E feature provides rules to define the mapping between an IPv6 prefix and an IPv4 address or between a shared IPv4 address and an IPv6 prefix/address. The MAP-E feature is supported by the Stateless NAT64 feature and does not change the system flow of the NAT64 client.

Feature Information for Mapping of Address and Port Using Encapsulation

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.

Restrictions for Mapping of Address and Port Using Encapsulation

- The MAP-E feature supports only a single basic mapping rule (BMR) per IPv6 prefix. This requires you to configure different mapping rules for every address and port translation.
- Default mapping rule (DMR) with 128 prefix must be configured before starting the MAP-E BMR configuration.
- This feature does not support BMR prefix length of 64, fragmentation, and local packet generation.

Information About Mapping of Address Port Using Encapsulation

Mapping of Address and Port Using Encapsulation

MAP-E refers to Mapping of Address and Port Encapsulation (MAP-E). The MAP-E feature enables you to configure mapping rules for translation between IPv4 and IPv6 addresses. Each mapping of address and port using MAP-E domain uses a different mapping rule. A MAP-E configuration comprises of one basic mapping rule (BMR), one default mapping rule (DMR), and one or more forwarding mapping rules (FMRs) for each MAP-E domain.

A BMR configures the MAP IPv6 address or prefix. You can configure only one BMR per IPv6 prefix. The MAP-E CE uses the BMR to configure itself with an IPv4 address, an IPv4 prefix, or a shared IPv4 address from an IPv6 prefix. A BMR can also be used for forwarding packets in such scenarios where an IPv4 source address and source port are mapped into an IPv6 address/prefix. Every MAP-E node (CE device is a MAP-E node) must be provisioned with a BMR. The BMR prefix along with the port parameter is used as tunnel source address. You can use the **port-parameters** command to configure port parameters for the MAP-E BMR.

A DMR prefix which matches with the interface address is recognized as hosts and a DMR prefix with a prefix length of 128 is recognized as the tunnel source address. A border relay IPv6 address is used as the tunnel destination address.

When you boot up a Customer Edge (CE) device for the first time, the CE sends an HTTP request to the rule server to acquire the MAP-E rules. After the CE receives the MAP-E rules, it saves a copy of the rules in a persistent storage, such as bootflash. When you reboot the router subsequently, the CE then detects the copy of MAP-E rules in the bootflash, so it does not send the HTTP request immediately. For a fixed IP in IP, the CE sends the request to the rule server only after the Dynamic Domain Name System (DDNS) reply is successfully received from the address resolution server.

Note In a fixed IP in IP, the IP in IP tunnel interface is used instead of a NAT64 configuration. Use the **nat64 provisioning mode** command to enable the tunnel interface.

Map Rule Request

Sl. No	Specifications	Remarks
1	HTTP	Versions: 1.0, 1.1, 2.0
2	HTTP	Method is GET
3	Communicate to IPv6 obtained by name resolution	IPv6 on the rule distribution server side is variable, so do not cache AAAA records.
4	Embed ipv6Prefix and ipv6PrefixLength in a query parameter.	ipv6Prefix=2404:050XXX&ipv6PrefixLength=YY
5	Embed API key in a query parameter.	Ex) code=Abag9k2RFgerkljgsirSDEFgwada

Map Rule Server Transmission of Data

Specifications	Information
Map Rule Specifications - Rule IPv6 prefix - Rule IPv6 prefix Length - Rule IPv4 prefix - Rule IPv4 prefix Length - EA - bits length - PSID offset - BR IPv6 Address	Essential information to generate Basic Mapping Rule (BMR) according to draft-ietf-softwire-map-03.
256	The maximum number of MAP rules that can be included in the transmitted data.
content-length	XXX (body size)
content-type	application/json; charset=utf-8

Map Rule Server URL Specification

Specifications	Remarks
URI https://rule.map.ocn.ad.jp/?ipv6Prefix=<address>&ipv6PrefixLength=<prefixLength>&code=<API Key>	Embed <IPv6 address> and <prefix length> allocated to CE • Example of URI: https://rule.map.ocn.ad.jp/?ipv6Prefix=2400:4050:XXX:&ipv6PrefixLength=YY&code=Abag9k2RFgerkljgsirSDEFgwada

The query parameter specification is as shown in the table below:

Map Rule Server Transmission of Data

Specifications	Information
Map Rule Specifications: - Rule IPv6 prefix - Rule IPv6 prefix length - Rule IPv4 prefix	Information to generate Basic Mapping Rule (BMR).

Specifications	Information
- Rule IPv4 prefix length - EA - bits length - PSID offset - BR IPv6 Address	
256	The maximum number of MAP rules that can be included in the transmitted data
Content-length	XXX (body size)
Content-type	application/json; charset=utf-8

Map Rule Server Response Parameters

Name	Description	Type	Byte	Remarks
brIpv6 Address	BRIPv6 address	string	39 (max)	Compliant with RFC 5952
eaBitLength	EAbit Length	string	2 (max)	Example: 25
ipv4Prefix	User IPv4 Prefix	string	15 (max)	Example: 10.0.0.0
ipv4PrefixLength	User IPv4 Prefix Length	string	2 (max)	Example: 16
ipv6Prefix RFC 5952	User IPv6 Prefix	String	39 (max)	Compliant with RFC 5952
pv6PrefixLength	User IPv6 Prefix Length	string	2 (max)	Example: 35
psIdOffset	PSID Offest	string	2 (max)	Example: 6

How to Configure Mapping of Address Port Using Encapsulation

Enable Tunnel Interface

Perform this task to enable tunnel interface. This task shows the **ipv6 nat prefix v4-mapped** command configured on a specified interface, but the command could alternatively be configured globally:

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 tunnel***tunnel-number*

Example:

```
Router(config)# interface tunnel0
```

Specifies the interface tunnel number.

Step 4 **nat64-mape**

Example:

```
Device(config)# nat64-mape
```

Specifies the MAP-E mapping rule and enters the basic mapping rule configuration mode.

Step 5 **ipv4** *ipv4-prefix*

Example:

```
Device(config-nat64-mape)# 10.1.1.0
```

Specifies the ipv4 address from rule server.

Step 6 **ip nat outside**

Example:

```
Device(config-nat64-mape)# ip nat outside
```

Specifies the ipv4 nat address.

Step 7 **ip virtual re-assembly in**

Example:

```
Device(config-nat64-mape)# ip virtual re-assembly in
```

Configures the virtual re-assembly.

Step 8 **ip tcp adjust-mss** *adjust-mss-number*

Example:

```
Device(config-nat64-mape)# ip tcp adjust-mss 1300
```

Specifies the TCP number.

Step 9 **tunnel source** *source-address*

Example:

```
Device(config-nat64-mape)# tunnel source 2001:22::0/128
```

Specifies the ipv6 tunnel source address.

Step 10 **tunnel mode** *ipv6-prefix*

Example:

```
Device(config-nat64-mape)# tunnel mode ipv6
```

Configures the ipv6 tunnel mode.

Step 11 **port-parameters share-ratio** *number* **port-offset-bits** *number*| **start-port** *port-number*| **no-eabits** *number*

Example:

```
Device(config-nat64-mape-bmr)# port-parameters share-ratio 2 port-offset-bits 5 start-port 1024
```

Specifies the values for port-parameters share-ratio, contiguous ports and start-port for MAP-E Basic Mapping Rule (BMR).

- If the share ratio is greater than 1, the configuration throws an error if the startport value is incorrect. The calculation is based on the share-ratio and port-offset bits. The configuration throws error and displays the value to be configured.
- If the share ratio is 1, there are no port-offset bits as the values is automatically set to 6 and the start port is set to 1024.

Step 12 **exit**

Example:

```
Device(config-nat64-mape-bmr)# exit
```

Exits basic mapping rule configuration mode and returns to MAP-E configuration mode.

Step 13 **default-mapping-rule** *ipv6 prefix*/*length*

Example:

```
Device(config-nat64-MAP-E-dmr)# default-mapping-rule 2001:22::0/128
```

Specifies the values of IPv6 prefix and length for MAP-E Default Mapping Rule (DMR).

Step 14 **mode map-e**

Example:

```
Device(config-nat64-MAP-E)# mode map-e
```

Specifies the value for MAP-E mode.

Step 15 **end**

Example:

```
Device(config-route-map)# end
```

Exits MAP-E configuration mode and returns to privileged EXEC mode.

Automatic Configuration of Address and Port Using Encapsulation

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 **nat64 provisioning mode** *mode id*

Example:

```
Device(config)# nat64 provisioning mode jp01
```

Specifies the nat64 MAP-E domain and enters the MAP-E configuration mode.

Step 4 **version draft-ietf-softwire-map-03**

Example:

```
Device(config-nat-provisioning)# version draft-ietf-softwire-map-03
```

Specifies the MAP version.

Step 5 **rule-server** *url*

Example:

```
Device(config-nat64-provisioning)# rule server rule-server 7
121111030251434B2B39342C36262349041F100259080A00745C53484E037C750B08050E58085E57020D555C0B054B0E4B1D34404B471316181C
```

Specifies the NAT64 rule server address.

Step 6 **api-key** *key-id*

Example:

```
Device(config-nat64-provisioning)# api-key api-key 7
85E57020D555C0B054B0E4B1D34404B471316181C
```

Specifies the NAT64 api key ID.

Step 7 **address-resolution-server** *url*

Example:

```
Device(config-nat64-provisioning)# address-resolution-server 7
00259080A00745C53484E037C750B08050E58085E57020D555C0B054B0E4B1D34404B471316181C
```

Specifies URL of the address resolution server.

Step 8 **exit**

Example:

```
Device(config-nat64-provisioning)# exit
```

Exits the NAT64 provisioning and returns to MAP-E configuration mode.

Step 9 **end**

Example:

```
Device(config)# end
```

Exits MAP-E configuration mode and returns to privileged EXEC mode.

Verifying Manual Mapping of Address and Port Using Encapsulation Configuration

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show nat64 MAP-E** [*domain number*]

Example:

```
Device# show nat64 MAP-E domain 1
  MAP-E Domain 1
   Mode MAP-E
   Default-mapping-rule
      Ip-v6-prefix 2001:22::/128
   Basic-mapping-rule
      Ip-v6-prefix 2001:100::/64
      Ip-v4-prefix 10.1.1.0/24
      Port-parameters
         Share-ratio 2   Contiguous-ports 1024   Start-port 1024
         Share-ratio-bits 1   Contiguous-ports-bits 10   Port-offset-bits 5
```

Displays MAP-E configuration.

Automatic Configuration of Address and Port Using Encapsulation

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 **nat64 provisioning mode** *mode id*

Example:

```
Device(config)# nat64 provisioning mode jp01
```

Specifies the nat64 MAP-E domain and enters the MAP-E configuration mode.

Step 4 **version draft-ietf-softwire-map-03**

Example:

```
Device(config-nat-provisioning)# version draft-ietf-softwire-map-03
```

Specifies the MAP version.

Step 5 **rule-server** *url*

Example:

```
Device(config-nat64-provisioning)# rule server rule-server 7
121111030251434B2B39342C36262349041F100259080A00745C53484E037C750B08050E58085E57020D555C0B054B0E4B1D34404B471316181C
```

Specifies the NAT64 rule server address.

Step 6 **api-key** *key-id*

Example:

```
Device(config-nat64-provisioning)# api-key api-key 7
85E57020D555C0B054B0E4B1D34404B471316181C
```

Specifies the NAT64 api key ID.

Step 7 **address-resolution-server** *url*

Example:

```
Device(config-nat64-provisioning)# address-resolution-server 7
00259080A00745C53484E037C750B08050E58085E57020D555C0B054B0E4B1D34404B471316181C
```

Specifies URL of the address resolution server.

Step 8 **exit**

Example:

```
Device(config-nat64-provisioning)# exit
```

Exits the NAT64 provisioning and returns to MAP-E configuration mode.

Step 9 **end**

Example:

```
Device(config)# end
```

Exits MAP-E configuration mode and returns to privileged EXEC mode.

Configuration Examples for Mapping of Address and Port Using Encapsulation

Example: Manual Mapping of Address and Port Using Encapsulation Configuration

The following example shows how to configure MAP-E:

```
enable
 configure terminal
 nat64 map-e domain 1
  basic-mapping-rule
   ipv6-prefix 4001:DB8::/40
   ipv4-prefix 50.50.50.0/24
   port-parameters share-ratio 1 start-port 1
  default-mapping-rule 3001:1::C0A8:105/128
   end
```

The following example shows shared IPv4 configurations:

```
enable
 configure terminal
  nat64 route 0.0.0.0/0 GigabitEthernet0/0/0
  nat64 provisioning mode jp01
  version draft-ietf-softwire-map-03
  rule-server 7
030C4F1F16556E034F0B1A5F0713181F4E0A797C78676F06315F4C215106080209055F4C1517495D1A41475951465A131357190E00090A

  api-key 7 050A070D23
 service-prefix 2400:4050::/30

   end


enable
 configure terminal
 nat64 map-e domain 1
  basic-mapping-rule
   ipv6-prefix 4001:DB8::/40
   ipv4-prefix 50.50.50.0/24
   port-parameters share-ratio 1 start-port 1
  default-mapping-rule 3001:1::C0A8:105/128
   end
```

Additional References for Mapping of Address and Port Using Encapsulation

Related Documents

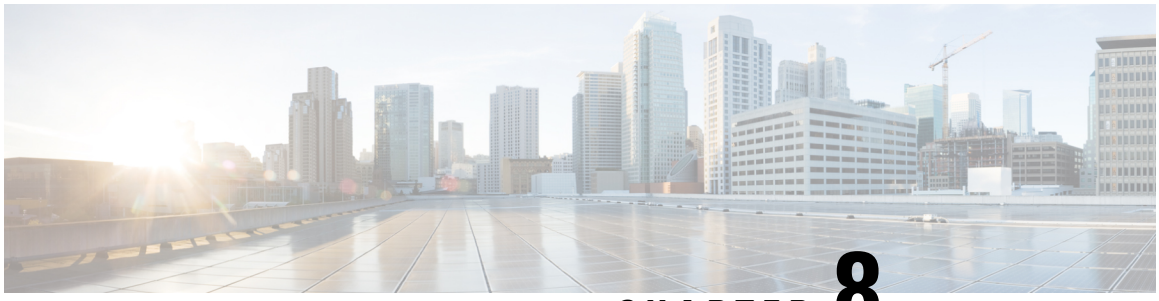
Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands	Cisco IOS IP Addressing Services Command Reference

Standards and RFCs

Standard/RFC	Title
MAP	Mapping of Address and Port (MAP)
MAP Encapsulation	MAP Encapsulation (MAP-E) - specification
RFC 6052	IPv6 Addressing of IPv4/IPv6 Translators
RFC 6144	Framework for IPv4/IPv6 Translation
RFC 6145	IP/ICMP Translation Algorithm

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



CHAPTER 8

Integrating NAT with MPLS VPNs

Network Address Translation (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.

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

Inside VPN to VPN with NAT is not supported.

Information About Integrating NAT with MPLS VPNs

Benefits of NAT Integration with MPLS VPNs

MPLS service providers would like to provide value-added services such as Internet connectivity, domain name servers (DNS), and voice over IP (VoIP) service to their customers. The providers require that their customers; IP addresses be different 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.

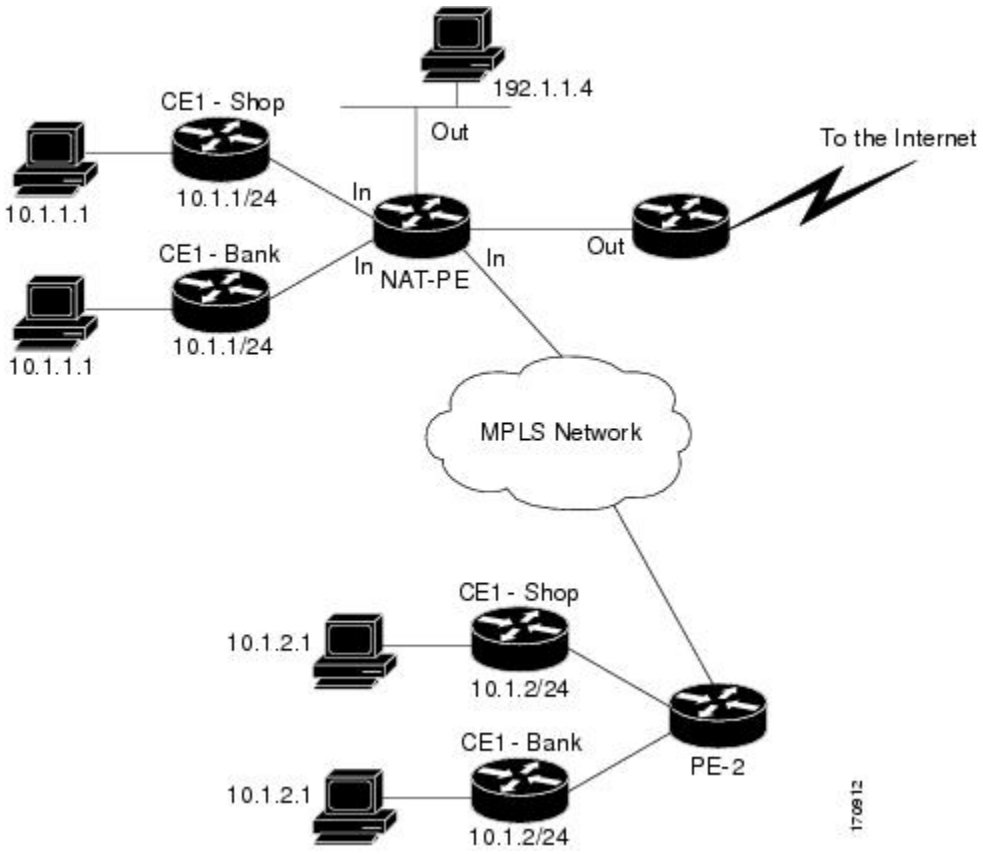
Scenarios for Implementing NAT on the PE Router

NAT could 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, 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 connected to the internet and centralized mail service is employed to do the address translation.

Figure 15: 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 wish to configure for your network:

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 2.2.2.10 2.2.2.10 netmask 255.255.255.0`	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.
Step 5	Repeat Step 4 for each VPN being configured	--
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 ethernet 0 168.58.88.2`	Allows NAT to be configured on a particular VPN.
Step 7	Repeat Step 6 for each VPN being configured.	--
Step 8	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.
Step 9	**show ip nat translations vrf** *vrf-name* **Example:**	(Optional) Displays the settings used by virtual routing/forwarding (VRF) table translations.

	Command or Action	Purpose
	`Router# show ip nat translations vrf shop`	

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 2.2.2.1 vrf shop`	Enables inside static translation on the VRF.
Step 4	Repeat Step 3 for each VPN being configured.	--
Step 5	**ip route vrf** *vrf-name* **prefix** *prefix mask next-hop-address* **global** **Example:**	Allows the route to be shared by several customers.

	Command or Action	Purpose
	`Router(config)# ip route vrf shop 0.0.0.0 0.0.0.0 168.58.88.2 global`	
Step 6	Repeat Step 5 for each VPN being configured.	--
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 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 outside** *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.

	Command or Action	Purpose
Step 3	**ip nat pool outside** *global-ip local-ip* **netmask** *netmask* **Example:** `Router(config)# ip nat pool outside 4.4.4.1 4.4.4.254 netmask 255.255.255.00`	Allows the configured VRF to be associated with the NAT translation rule.
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 2.2.2.1 vrf shop`	Allows the route to be shared by several customers.
Step 5	Repeat Step 4 for each VRF being configured.	Allows the route to be shared by several customers.
Step 6	**ip nat outside source static** *global-ip local-ip* **vrf** *vrf-name* **Example:** `Router(config)# ip nat outside source static 168.58.88.2 4.4.4.1 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 inside** *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 inside** *global-ip local-ip* **netmask** *netmask* **Example:** `Router(config)# ip nat pool inside1 2.2.1.1 2.2.1.254 netmask 255.255.255.0`	Allows the configured VRF to be associated with the NAT translation rule.
Step 4	Repeat Step 3 for each pool being configured.	--
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 inside2 vrf shop`	Allows the route to be shared by several customers.
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 168.58.88.2 4.4.4.1 vrf shop`	Allows the route to be shared by several customers.
Step 8	Repeat Step 7 for all VPNs being configured.	--
Step 9	**exit** **Example:** `Router(config)# exit`	Returns to privileged EXEC mode.

	Command or Action	Purpose
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

The following example shows configuring inside Dynamic NAT with MPLS VPNs.

```
!
ip nat pool inside 2.2.2.10 2.2.2.10 netmask 255.255.255.0
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 Ethernet1/3 168.58.88.2
ip route vrf bank 0.0.0.0 0.0.0.0 Ethernet1/3 168.58.88.2
ip route vrf park 0.0.0.0 0.0.0.0 Ethernet1/3 168.58.88.2
!
access-list 1 permit 192.168.0.0 0.0.255.255
```

Configuring Inside Static NAT with MPLS VPNs Example

The following example shows configuring inside static NAT with MPLS VPNs.

```
!
ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop
ip nat inside source static 192.168.122.49 2.2.2.2 vrf shop
ip nat inside source static 192.168.121.113 2.2.2.3 vrf bank
ip nat inside source static 192.168.22.49 2.2.2.4 vrf bank
ip nat inside source static 192.168.121.113 2.2.2.5 vrf park
ip nat inside source static 192.168.22.49 2.2.2.6 vrf park
ip nat inside source static 192.168.11.1 2.2.2.11 vrf shop
ip nat inside source static 192.168.11.3 2.2.2.12 vrf shop
ip nat inside source static 140.48.5.20 2.2.2.13 vrf shop
!
ip route 2.2.2.1 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.2 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.3 255.255.255.255 Serial2/1.1 192.168.121.113
ip route 2.2.2.4 255.255.255.255 Serial2/1.1 192.168.121.113
ip route 2.2.2.5 255.255.255.255 FastEthernet0/0 192.168.121.113
ip route 2.2.2.6 255.255.255.255 FastEthernet0/0 192.168.121.113
ip route 2.2.2.11 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.12 255.255.255.255 Ethernet1/0 192.168.121.113
ip route 2.2.2.13 255.255.255.255 Ethernet1/0 192.168.121.113
```

Configuring Outside Dynamic NAT with MPLS VPNs Example

The following example shows configuring outside dynamic NAT with MPLS VPNs.

```
!
ip nat pool outside 4.4.4.1 4.4.4.254 netmask 255.255.255.0
ip nat inside source static 192.168.121.113 2.2.2.1 vrf shop
ip nat inside source static 192.168.122.49 2.2.2.2 vrf shop
ip nat inside source static 192.168.121.113 2.2.2.3 vrf bank
ip nat inside source static 192.168.22.49 2.2.2.4 vrf bank
ip nat inside source static 192.168.121.113 2.2.2.5 vrf park
ip nat inside source static 192.168.22.49 2.2.2.6 vrf park
ip nat outside source list 1 pool outside
!
```

Configuring Outside Static NAT with MPLS VPNs Example

The following example shows configuring outside static NAT with MPLS VPNs.

```
!
ip default-gateway 10.1.15.1
ip nat pool inside1 2.2.1.1 2.2.1.254 netmask 255.255.255.0
ip nat pool inside2 2.2.2.1 2.2.2.254 netmask 255.255.255.0
ip nat pool inside3 2.2.3.1 2.2.3.254 netmask 255.255.255.0
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 168.58.88.2 4.4.4.1 vrf bank
ip nat outside source static 18.68.58.1 4.4.4.2 vrf park
ip nat outside source static 168.58.88.1 4.4.4.3 vrf shop
ip classless
ip route 192.170.10.0 255.255.255.0 Ethernet1/0 192.168.121.113
ip route 192.170.11.0 255.255.255.0 Serial2/1.1 192.168.121.113
ip route 192.170.12.0 255.255.255.0 FastEthernet0/0 192.168.121.113
ip route vrf shop 0.0.0.0 0.0.0.0 168.58.88.2 global
ip route vrf bank 0.0.0.0 0.0.0.0 168.58.88.2 global
ip route vrf park 0.0.0.0 0.0.0.0 168.58.88.2 global
no ip http server
!
access-list 1 permit 192.168.0.0 0.0.255.255
```

Where to Go Next

- To learn about Network Address Translation and configure NAT for IP address conservation, see the “Configuring NAT for IP Address Conservation” module.
- To verify, monitor, and maintain NAT, see the “Monitoring and Maintaining NAT” module.
- To use NAT with application level gateways, see the “Using Application Level Gateways with NAT” module.
- To configure NAT for high availability, see the “Configuring NAT for High Availability” module.

Additional References for Integrating NAT with MPLS VPNs

Related Documents

Related Topic	Document Title
IOS Commands	Cisco IOS Master Command List
NAT commands	Cisco IOS IP Addressing Services Command Reference

Standards and RFCs

Standard & RFC	Title
RFC 2547	*BGP/MPLS VPNs*

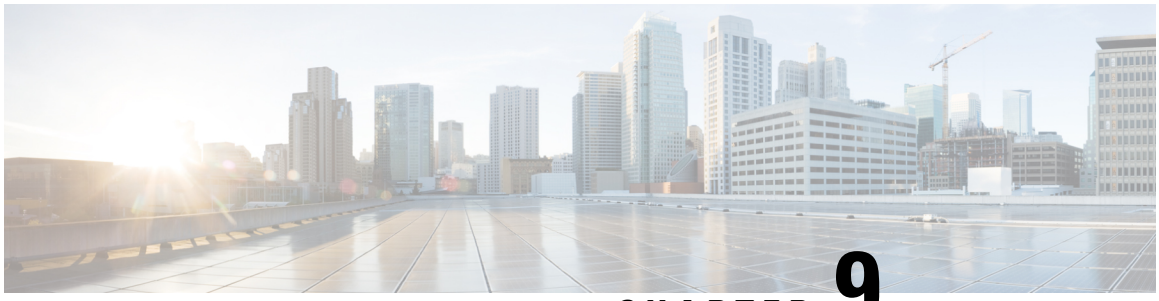
Technical Assistance

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 Integrating NAT with MPLS VPNs

Table 7: Feature Information for Integrating NAT with MPLS VPNs

Feature Name	Releases	Feature Configuration Information
Integrating NAT with MPLS VPNs	12.1(13)T 15.1(1)SY	The Integrating NAT with MPLS VPNs feature allows multiple Multiprotocol Label Switching (MPLS) VPNs to be configured on a single device to work together.



CHAPTER 9

Configuring Hosted NAT Traversal for Session Border Controller

The Cisco IOS Hosted NAT Traversal for Session Border Controller Phase-1 feature enables a Cisco IOS Network Address Translation (NAT) Session Initiation Protocol (SIP) Application Level Gateway (ALG) router to act as a Session Border Controller (SBC) on a Cisco Multiservice IP-to-IP gateway, ensuring a seamless delivery of VoIP services.

The Cisco IOS Hosted NAT Traversal for Session Border Controller Phase-2 feature provides registration throttling, media flow-through, and Stateful NAT (SNAT) support.

Note Effective January 31, 2014, Stateful NAT is not available in Cisco IOS software. For more information, see End-of-Sale and End-of-Life Announcement for the Cisco IOS Stateful Failover of Network Address Translation (SNAT).

Prerequisites for Configuring Cisco IOS Hosted NAT Traversal for Session Border Controller

- Before you configure the Cisco IOS Hosted NAT Traversal for Session Border Controller feature, you should understand the concepts documented in “Cisco IOS Hosted NAT Traversal for Session Border Controller Overview” section.

- 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 "Creating an IP Access List and Applying It to an Interface" module in the *Securing the Data Plane Configuration Guide*.
- Before performing the tasks in this module, you should verify that SIP has not been disabled. SIP is enabled by default.

Restrictions for Configuring Cisco IOS Hosted NAT Traversal for Session Border Controller

- Phase 1 supports flow-around mode for inside to inside media calls and flow-through for inside to outside media calls.
- If the intermediate routers between the inside phones and the NAT SBC are configured for Port Address Translation (PAT), the user agents (phones and proxy) must support symmetric signaling and symmetric and early media. The override port must be configured on the NAT SBC router. In the absence of support for symmetric signaling and symmetric and early media, the intermediate routers must be configured for non-PAT and the override address should be configured in the NAT SBC.

Information About Configuring Cisco IOS Hosted NAT Traversal for Session Border Controller

Voice and Multimedia over IP Networks

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

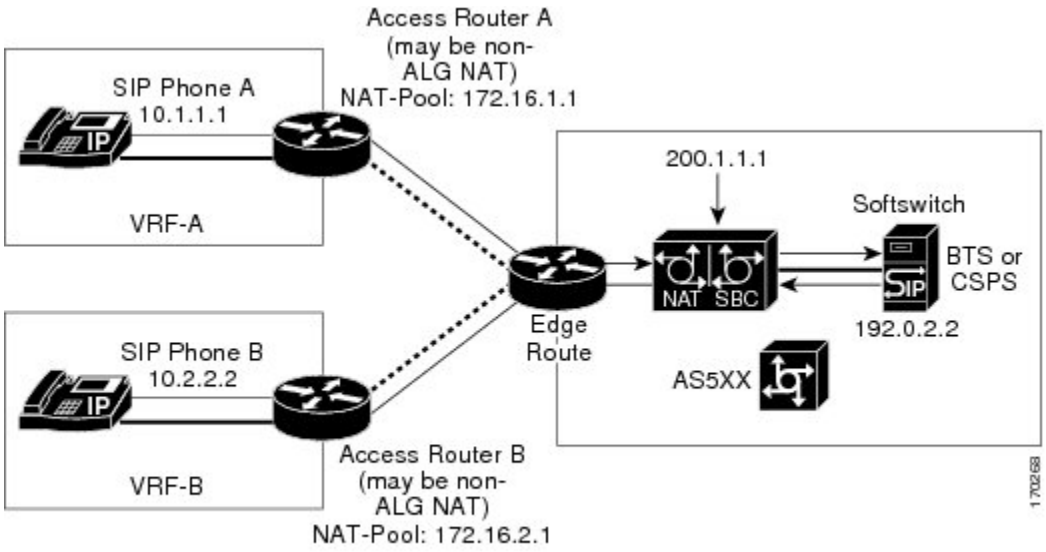
Session Description Protocol (SDP) describes multimedia sessions. SDP may be used in SIP message bodies to describe the multimedia sessions that are used for creating and controlling the multimedia sessions with two or more participants.

Cisco IOS Hosted NAT Traversal for Session Border Controller Overview

Private IP addresses and ports inserted in the packet payload by client devices, such as IP phones and video conferencing stations, are not routable in public networks using NAT. In addition, intermediate routers between the inside phones and the NAT SBC can have the non-ALG functionality. The hosted NAT traversal handles the signaling and the media streams involved in the setting up, conducting, and tearing down of calls that traverse these intermediate routers.

The figure below illustrates how the NAT SBC handles embedded SIP/SDP information for the address and port allocation by differentiating the overlapped embedded information.

Figure 16: NAT as a SIP Session Border Controller



The inside phones have the proxy configured as the NAT SBC's preconfigured address and port. NAT SBC has the Softswitch's address and port preconfigured as the proxy. The NAT SBC intercepts the packets destined from the inside phones to itself and translates the inside hosts and other information in the SIP/SDP payload and the IP/UDP destination address or port to the Softswitch's address and port, and vice versa.

SIP/SDP information is either a NAT or a PAT in order for the Real-Time Transport Protocol (RTP) flow to be directly between the phones in the NAT SBC inside domain.

The address-only fields are not translated by the NAT SIP ALG. The address-only fields are handled by the NAT SBC, except for the proxy-authorization and authorization translation, because these will break the authentication.

If the intermediate routers between the inside phones and the NAT SBC are configured to do a PAT, the user agents (phones and proxy) must support symmetric signaling and symmetric and early media. You must configure the override port on the NAT SBC router. In the absence of support for symmetric signaling and symmetric and early media, the intermediate routers must be configured without PAT and the override address should be configured in the NAT SBC.

The registration throttling support enables you to define the parameters in the Expires: header and the expires= parameter. It allows you to elect to not forward certain registration messages to the Softswitch.

How to Configure Cisco IOS Hosted NAT for Session Border Controller

Configuring Cisco IOS Hosted NAT for Session Border Controller

Perform this task to configure NAT for SBC.

Note When you use the NAT SBC feature and you want the call IDs to be translated, you must configure two address pools in such a way that the pool for SBC is accessed before the pool for the call IDs. Use the **ip nat pool** command to configure the address pools. Access lists are chosen in ascending order, so you should assign the list associated with the SBC pool a lower number than the list associated with the call ID pool.

Note The proxy of the inside phones must be set to 200.1.1.1. The VPN routing and forwarding (VRF) instance configuration as shown is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip nat inside**
5. **exit**
6. **interface** *type number*
7. **ip nat outside**
8. **exit**
9. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
10. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
11. **ip nat inside source list** *access-list-number* **pool** *name* [**vrf** *vrf-name*] [**overload**]
12. **ip nat outside source list** *access-list-number* **pool** *name*
13. **ip nat sip-sbc**
14. **proxy** *inside-address inside-port outside-address outside-port* **protocol udp**
15. **vrf-list**
16. **vrf-name** *vrf - name*
17. **exit**
18. **ip nat sip-sbc**
19. **call-id-pool** *call -id-pool*
20. **session -timeout** *seconds*
21. **mode allow -flow-around**
22. **override address**
23. **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	**interface** *type number* **Example:** `Router(config)# interface ethernet 1/1`	Specifies an interface and returns to interface configuration mode.
Step 4	**ip nat inside** **Example:** `Router(config-if)# ip nat inside`	Connects the interface to the inside network (the network subject to NAT translation).
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 6	**interface** *type number* **Example:** `Router(config)# interface ethernet 1/3`	Specifies an interface and enters interface configuration mode.
Step 7	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:** `Router(config)# ip nat pool inside-pool-A 172.16.0.1 172.16.0.10 prefix-length 16`	Defines a pool of global addresses to be allocated for the inside network. **Note** You must configure two address pools when you are using the NAT SBC feature and you want to translate the call IDs. In this step you are configuring the first address pool.
Step 10	**ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*} **Example:**	Defines a pool of global addresses to be allocated for the outside network.

	Command or Action	Purpose
	`Router(config)# ip nat pool outside-pool 203.0.113.1 203.0.113.10 prefix-length 24`	**Note** You must configure two address pools when you are using the NAT SBC feature and you want to translate the call IDs. In this step, you are configuring the second address pool.
Step 11	**ip nat inside source list** *access-list-number* **pool** *name* [**vrf** *vrf-name*] [**overload**] **Example:** `Router(config)# ip nat inside source list 1 pool inside-pool-A vrf vrfA overload`	Enables NAT of the inside source address and configures the access list for translation.
Step 12	**ip nat outside source list** *access-list-number* **pool** *name* **Example:** `Router(config)# ip nat outside source list 3 pool outside-pool`	Enables NAT of the outside source address and configures the access list for translation.
Step 13	**ip nat sip-sbc** **Example:** `Router(config)# ip nat sip-sbc`	Enters IP NAT SBC configuration mode.
Step 14	**proxy** *inside-address inside-port outside-address outside-port* **protocol udp** **Example:** `Router(config-ipnat-sbc)# proxy 200.1.1.1 5060 192.0.2.2 5060 protocol udp`	Configures the address or port that the inside phones will be referring to, and the outside proxy's address and port to which the NAT SBC translates the destination IP address and port.
Step 15	**vrf-list** **Example:** `Router(config-ipnat-sbc)# vrf-list`	(Optional) Enters IP NAT SBC VRF configuration mode.
Step 16	**vrf-name** *vrf - name* **Example:** `Router(config-ipnat-sbc-vrf)# vrf-name vrf1`	(Optional) Defines SBC VRF list names.
Step 17	**exit** **Example:** `Router(config-ipnat-sbc-vrf)# exit`	Exits IP NAT SBC VRF configuration mode and enters global configuration mode.
Step 18	**ip nat sip-sbc** **Example:**	Enters IP NAT SBC configuration mode.

	Command or Action	Purpose
	`Router(config)# ip nat sip-sbc`	

Step 19	**call-id-pool** *call -id-pool*
	Example:
	`Router(config-ipnat-sbc)# call-id-pool pool-name`
		Specifies a dummy pool name for the in to out SIP signaling packet's call ID that it will be translated to, and that a 1:1 association will be maintained rather than using the regular NAT pool.
		• This pool can be used in an overload scenario:
		  • NAT mapping with an appropriate access control list (ACL) and a NAT pool matching the pool name must be configured.
		  • This pool is not used for any other NAT processing except for call ID processing.

Step 20	**session -timeout** *seconds* **Example:** `Router(config-ipnat-sbc)# session-timeout 300`	Configures the timeout duration for NAT entries pertaining to SIP signaling flows. • The default is 5 minutes.
Step 21	**mode allow -flow-around** **Example:** `Router(config-ipnat-sbc)# mode allow-flow-around`	Enables flow-around for RTP. • This flow applies to traffic between phones in the inside domain.
Step 22	**override address** **Example:** `Router(config-ipnat-sbc)# override address`	Allows the NAT SBC to override the out to in traffic's destination IP during signaling or RTP traffic, or to override the address and port.
Step 23	**end** **Example:** `Router(config-ipnat-sbc)# end`	Exits IP NAT SBC configuration mode and enters privileged EXEC mode.

Configuration Examples for Configuring Cisco IOS Hosted NAT for Session Border Controller

Example Configuring Cisco IOS Hosted NAT Traversal for Session Border Controller

The following example shows how to configure the Cisco IOS Hosted NAT Traversal as Session Border Controller feature:

```
interface ethernet1/1
 ip nat inside
!
interface ethernet1/2
 ip nat inside
!
interface ethernet1/3
 ip nat outside
!
ip nat pool inside-pool-A 172.16.0.1 172.16.0.10 prefix-length 16
ip nat pool inside-pool-B 192.168.0.1 192.168.0.10 prefix-length 24
ip nat pool outside-pool 203.0.113.1 203.0.113.10 prefix-length 24
ip nat inside source list 1 pool inside-pool-A vrf vrfA overload
ip nat inside source list 2 pool inside-pool-B vrf vrfB overload
ip nat outside source list 3 pool outside-pool
!
! Access-list for VRF-A inside phones
access-list 1 permit 172.16.0.0 255.255.0.0
!
! Access-list for VRF-B inside phones
access-list 2 permit 192.0.2.0 255.255.255.0
!
access-list 3 permit 203.0.113.0 255.255.255.0
ip nat sip-sbc
 proxy 200.1.1.1 5060 192.0.2.2 5060 protocol udp
 vrf-list
  vrf-name vrfA
  vrf-name vrfB
  exit
 call-id-pool pool-name
 session-timeout 300
 mode allow-flow-around
 override address
```

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*
Configuring an IP access list	"Creating an IP Access List and Applying It to an Interface" module in the *Securing the Data Plane Configuration Guide*

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

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 Hosted NAT Traversal for Session Border Controller

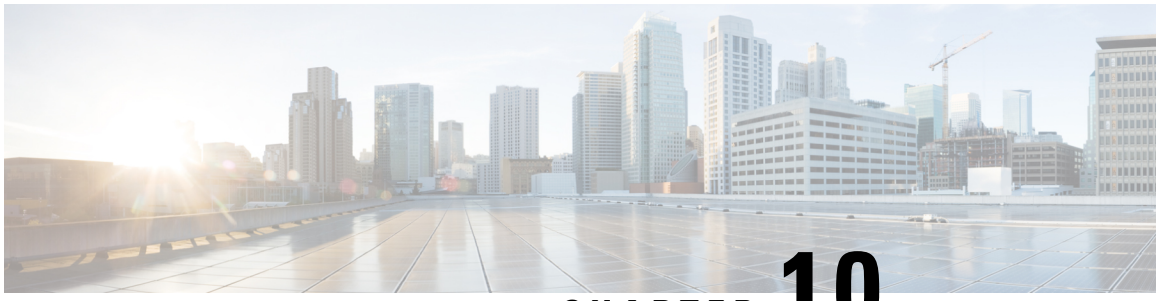
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train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

Feature Name	Releases	Feature Information
Cisco IOS Hosted NAT Traversal for Session Border Controller Phase-1	12.4(9)T	The Cisco IOS Hosted NAT Traversal for Session Border Controller feature provides transparency with the use of a proxy device on the NAT outside domain.
Hosted NAT Support for Session Border Controller Phase-2	12.4(15)T	The Hosted NAT Support for Session Border Controller Phase-2 feature provides registration throttling, media flow-through, and SNAT support. **Note** Effective January 31, 2014, Stateful NAT is not available in Cisco IOS software. For more information, see End-of-Sale and End-of-Life Announcement for the Cisco IOS Stateful Failover of Network Address Translation (SNAT).

Feature Name	Releases	Feature Information
NAT as SIP Session Border Controller Media Flow	12.4(9)T	The NAT as SIP Session Border Controller Media Flow feature provides support for media flow-around for RTP or RTCP exchanges between phones on the inside domain of the SBC.
NAT as SIP Session Border Controller Support for Address-Only Fields	12.4(9)T	The NAT as SIP Session Border Controller Support for Address-Only Fields feature provides support for the translation of SIP address-only fields.



CHAPTER 10

User Defined Source Port Ranges for PAT

The User Defined Source Port Ranges for PAT feature enables the specification of source port ranges for Port Address Translation (PAT) for SIP, H.323, and Skinny Real-Time Transport Protocol (RTP) and RTP Control Protocol (RTCP).

Restrictions for User Defined Source Port Ranges for PAT

- The size of port range that can be reserved is limited to a multiple of 64.
- The start port for the port range should also be a multiple of 64.

Information About User Defined Source Port Ranges for PAT

User Defined Source Port Ranges for PAT Overview

In order for VoIP traffic to not be in violation of the RTP standards and best practices, even/odd pairing of ports for RTP and RTCP traffic for SIP ALG, Skinny and H.323 has been made available.

Following is a scenario of what happens to VoIP traffic translated using PAT without user defined ports.

The first VoIP traffic getting translated using PAT, would request for port 16384 and would get to use port 16384 for its RTP traffic.

The second VoIP traffic stream getting translated using PAT would also request 16384 for its RTP. Since this port number is already in use by the first call, PAT would translate the 16384 source port for the second phone to 1024 (assuming the port was free) and this would be in violation of the RTP standards/best practices.

A third call would end up using port 1025 and others would increment from there.

Each call after the first call would end up having its inside source port translated to an external port assignment that is out of specifications for RTP, and this would continue until PAT binding fir the first call expires.

Problems associated with RTP traffic being assigned to a non-standard port by PAT:

- Inability for compressed RTP (cRTP) to be invoked in the return direction, as it only operates on RTP flows with compliant port numbers.
- Difficulty in properly classifying voice traffic for corresponding QoS treatment.
- Violation of standard firewall policies that specifically account for RTP/TRCP traffic by specified standard port range.

Even Port Parity

Cisco IOS NAT SIP gateways normally select the next available port+1 for SIP fixup in the NAT translations. The NAT gateway does not check for even/odd pair for RTP/TRCP port numbers, and as a result issues may arise with SIP user agents that are strictly following the encouraged even/odd parity for RTP/RTCP port numbers.

Even port parity for SIP, H.323, and skinny is supported by default and it can be turned off forcing the odd RTP ports allocation.

How to Configure User Defined Source Port Ranges for PAT

Configuring Source Port Ranges for PAT

Perform this task to assign a set of ports and associate a map to them.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat portmap** *mapname* **application** *application* **startport** *startport* **size** *size*
4. **ip nat inside source list** *list* - *name* **pool** *pool* - *name* **overload portmap** *portmap* - *name*

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 portmap** *mapname* **application** *application* **startport** *startport* **size** *size* **Example:** `Router(config)# ip nat portmap NAT-1 application sip-rtp startport 32128 size 128`	Defines the port map.
Step 4	**ip nat inside source list** *list - name* **pool** *pool - name* **overload portmap** *portmap - name* **Example:** `Router(config)# ip nat inside source list 1 pool A overload portmap NAT-1`	Associates the port map to the NAT configuration.

Configuring Even Port Parity

Even port parity for H.323, SIP, and skinny is supported by default and can be turned off forcing the odd ports allocation.

Perform this task to enable even port parity.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service allow-h323-even-rtp-ports** | **allow-sip-even-rtp-ports**| **allow-skinny-even-rtp-ports**

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 service allow-h323-even-rtp-ports**	**allow-sip-even-rtp-ports**| **allow-skinny-even-rtp-ports** **Example:** `Router(config)# ip nat service allow-h323-even-rtp-ports`	Establishes even port parity for H323, the SIP protocol, or the skinny protocol.

Configuration Examples for User Defined Source Port Ranges for PAT

Example User Defined Source Port Ranges for PAT

The following examples shows how to assign a set of ports and associate a map to them.

```
ip nat portmap NAT-I
 cisco-rtp-h323-low
 appl sip-rtp startport 32128 size 128
 appl sip-rtp startport 32000 size 64
ip nat inside source list 1 pool A overload portmap NAT-I
```

Macros have been defined to make port map configuration easier. The table below lists the name of the macros and the ports.

Table 9: Macro Names and Ports

Macro Name	Ports	Application
cisco-rtp-h323-low	16384-32767	H.323
cisco-rtp-h323-high	49152-65535	H.323
cisco-rtp-skinny-low	16384-32767	Skinny
cisco-rtp-skinny-high	49152-65535	Skinny
cisco-rtp-sip-low	16384-32767	SIP
cisco-rtp-sip-high	49152-65535	SIP

Example Even Port Parity

The following example enables even port parity for H.323.

```
ip nat service allow-h323-even-rtp-ports
```

The following example enables even port parity for SIP.

```
ip nat service allow-sip-even-rtp-ports
```

The following example enables even port parity for the skinny protocol.

```
ip nat service allow-skinny-even-rtp-ports
```

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, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
•	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

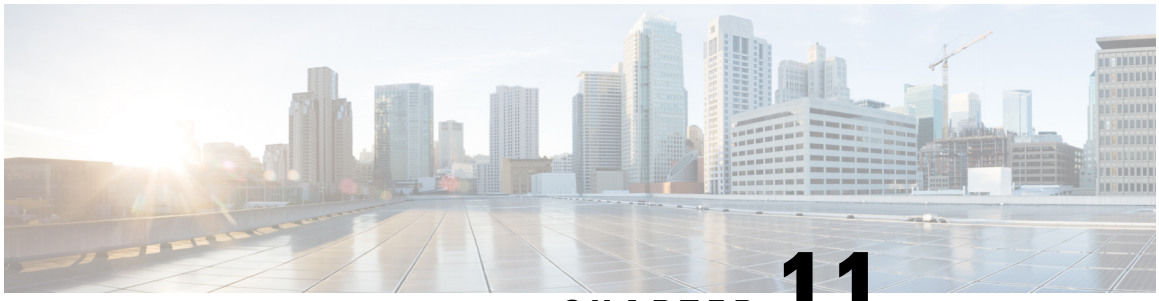
Feature Information for User Defined Source Port Ranges for PAT

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 10: Feature Information for User Defined Source Port Ranges for PAT

Feature Name	Releases	Feature Information
User Defined Source Port Ranges for PAT	12.4(11)T	The User Defined Source Port Ranges for PAT feature enables the specification of source port ranges for Port Address Translation (PAT) for SIP, H.323, and Skinny Real-Time Transport Protocol (RTP) and RTP Control Protocol (RTCP).



CHAPTER 11

FPG Endpoint Agnostic Port Allocation

When the Endpoint Agnostic Port Allocation feature is configured, an entry is added to the Symmetric Port Database. If the entry is already available, the port listed in the Symmetric Port Database is used and the packet is sent. This feature is only required if you need to configure NAT with pool overload or interface overload. Endpoint Agnostic Port Allocation is also known as Symmetric Port Allocation.

Information About Endpoint Agnostic Port Allocation

When a packet is being transmitted, the Symmetric Port Database is checked to see if the requested port is already allocated. If it has been allocated, it is checked if the source computer entry in the database matches the computer requesting the port. If this is true, the port listed in the Symmetric Port Database is used and the packet is sent.

If the computers do not match or if the requested port is not in the Symmetric Port Database, the feature continues checks to the NAT Port database for an entry matching the requested port. If no entry is found, this means that the port is available. A new entry is added to the NAT Port database, and to the existing NAT database, allocating the port to the requesting computer, and the packet is sent.

If no matching entry in the NAT Port database is found, it means that the port is busy, or otherwise unavailable. The next available port is found, which is allocated to the requesting computer. An entry is added to the NAT Port database with the requesting computer and the available port. An entry is added to the Symmetric Port database, with the requesting computer, the allocated port and the requested port and the packet is sent.

This feature is only required if you need to configure NAT with pool overload or interface overload. This feature is not applicable for other NAT configurations.

How to Configure Endpoint Agnostic Port Allocation

Configuring Endpoint Agnostic Port Allocation

Perform this task to configure NAT to support the Endpoint Agnostic Port Allocation feature.

Note This feature must be enabled by the user. It should be enabled before NAT is enabled. If it is enabled later, it will not translate the previously established connection. When this feature is disabled, it will not be seen in the output of the **show running-config**command.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface interface name**
4. **ip nat inside**
5. **exit**
6. **access list 1 permit** *ip address mask*
7. **ip nat inside source list 1 interface interface name**
8. **ip nat service enable-sym-port**
9. **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# configure terminal`	Enters global configuration mode.
Step 3	**interface interface name** **Example:** `Router (config)# interface Ethernet 0/0`	Configures the Ethernet 0/0 interface.
Step 4	**ip nat inside** **Example:**	Enables Network Address Translation (NAT) for the inside address.

	Command or Action	Purpose
	`Router (config-if)# ip nat inside`	
Step 5	**exit** **Example:** `Router (config-if)# exit`	Exits interface configuration mode.
Step 6	**access list 1 permit** *ip address mask* **Example:** `Router (config)# access list 1 permit 172.18.192.0.0.0.255`	Creates an access list called 1.
Step 7	**ip nat inside source list 1 interface interface name** **Example:** `Router (config)# ip nat inside source list 1 interface Ethernet 0/0`	Enables NAT for the inside source for access list 1 which is attached to the Ethernet interface.
Step 8	**ip nat service enable-sym-port** **Example:** `Router (config)# ip nat service enable-sym-port`	Enables the symmetric port allocation.
Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Verifying Endpoint Agnostic Port Support

To verify the Endpoint Agnostic Port Support feature, use the following command.

SUMMARY STEPS

1. **show ip nat translations**

DETAILED STEPS

show ip nat translations

Example:

```
Router# show ip nat translations

NAT Symmetric Port Database: 1 entries
```

```
public ipaddr:port [tableid] | port# [refcount][syscount] | localaddr:localport [flags]
172.18.192.69:1024 [0] | 1025 [1] [0] | 172.18.192.69:1024 [0]
```

Configuration Examples for Endpoint Agnostic Port Allocation

Configuring Endpoint Allocation Example

```
interface Ethernet0/0
 ip nat inside
 exit
access list 1 permit 172.18.192.0.0.0.255
ip nat inside source list 1 interface Ethernet0/0
ip nat service enable-sym-port
end
```

Additional References

Related Documents

Related Topic	Document Title
NAT configuration tasks	"Configuring NAT for IP Address Conservation" module
NAT maintenance	"Monitoring and Maintaining NAT" module
NAT commands: complete command syntax, command mode, command history, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Standards

Standard	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 IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

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

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

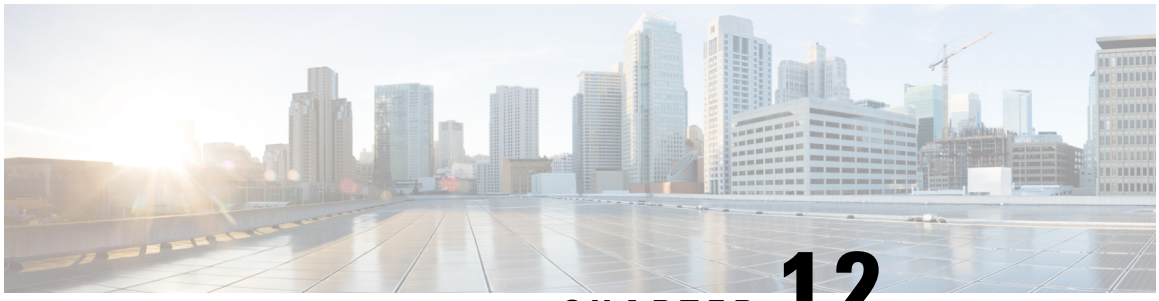
Feature Information for Endpoint Agnostic Port Allocation

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 11: Feature Information for NAT Endpoint Agnostic Port Allocation

Feature Name	Releases	Feature Information
FPG: Endpoint Agnostic Port Allocation	12.4(24)T	This feature was introduced.



CHAPTER 12

NAT Optimized SIP Media Path Without SDP

The NAT Optimized SIP Media Path Without SDP feature provides the ability to optimize the media path taken by a SIP VoIP session when NAT is used. NAT forces the VoIP traffic to take at least one extra hop in the network, which usually results in several additional hops being added to the path between two IP hosts.

The Message Digest 5 (MD5) algorithm is supported.

Information About the NAT Optimized SIP Media Path Without SDP Feature

Benefits of NAT Optimized SIP Media Path Without SDP

- The media path can be shortened, decreasing voice delay.
- More control of voice policy is possible because the media path is closer to the customer domain and not deep within the service provider cloud.
- Processes all packets sent through the NAT-enabled router, even those without the Session Description Protocol (SDP).

NAT Optimized SIP Media Path Without SDP Feature Design

Cisco IOS NAT will add the relevant translation information per SIP session within the SIP protocol messages. The SIP Application Layer Gateway support within Cisco IOS NAT will extract this translation information from the SIP packets and create NAT table entries.

The “piggybacking” of NAT translation information within the SIP call flows, the design of how users interact with the application when they talk to it, will allow the media path of a SIP VoIP session between two calling parties to take the optimized routing path between each other.

How to Configure NAT Optimized SIP Media Path Without SDP

Configuring a NAT Optimized SIP Media Path Without SDP Messages Including MD5 Authentication

Perform this task to configure messages with a NAT optimized SIP Media path including MD5 authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat piggyback-support sip-alg all-messages router** *router-id* [**md5-authentication** *md5-authentication-key*]

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 piggyback-support sip-alg all-messages router** *router-id* [**md5-authentication** *md5-authentication-key*] **Example:** `Router(config)# ip nat piggyback-support sip-alg all-messages router 100 md5-authentication md5-key`	Enables messages with a NAT optimized SIP Media path including MD5 authentication.

Configuring a NAT Optimized SIP Media Path Without SDP Messages

Perform this task to configure SDP messages with a NAT optimized SIP Media path without MD5 authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat piggyback-support sip-alg all-messages router** *router-id*

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 piggyback-support sip-alg all-messages router** *router-id* **Example:** `Router(config)# ip nat piggyback-support sip-alg all-messages router 100`	Enables messages with a NAT optimized SIP Media path without MD5 authentication.

Configuration Examples for NAT Optimized SIP Media Path Without SDP

Configuring a NAT Optimized SIP Media Path Without SDP Including MD5 Authentication Example

The following example shows how to configure a NAT optimized SIP media path without SDP including MD5 authentication:

```
ip nat piggyback-support sip-alg all-messages router 100 md5-authentication md5-key
```

Configuring a NAT Optimized SIP Media Path Without SDP or MD5 Authentication Example

The following example shows how to configure a NAT optimized SIP media path without SDP or MD5 authentication:

```
ip nat piggyback-support sip-alg all-messages router 100
```

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
Application-level gateways	*Using Application Level Gateways with NAT* module
IP access list sequence numbering	IP Access List Sequence Numbering document
NAT-on-a-Stick technology note	Network Address Translation on a Stick technology note
NAT maintenance	*Monitoring and Maintaining NAT* module
RADIUS attributes overview	*RADIUS Attributes Overview and RADIUS IETF Attributes* module
Using HSRP and stateful NAT for high availability	*Configuring NAT for High Availability* module
Using NAT with MPLS VPNs	*Integrating NAT with MPLS VPNs* module

Standards and RFCs

Standard/RFC	Title
RFC 1597	Internet Assigned Numbers Authority
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)

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

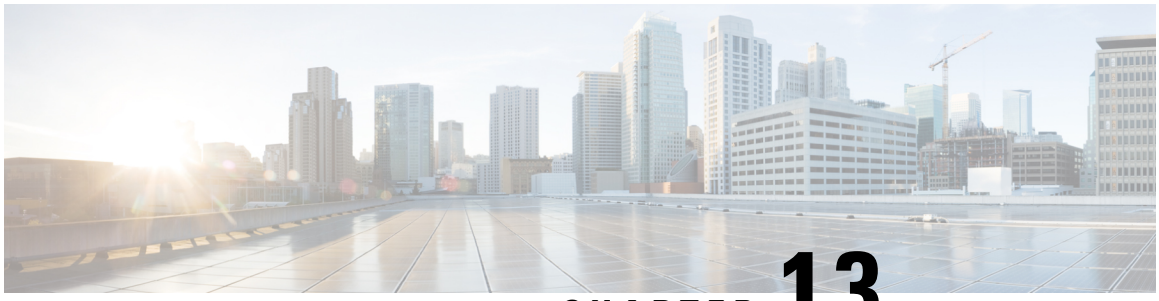
Feature Information for NAT Optimized SIP Media Path Without SDP

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 12: Feature Information for NAT Optimized SIP Media Path Without SDP

Feature Name	Releases	Feature Information
NAT Optimized SIP Media Path Without SDP	12.4(2)T	The NAT Optimized SIP Media Path Without SDP feature provides the ability to optimize the media path taken by a SIP VoIP session when NAT is used. NAT forces the VoIP traffic to take at least one extra hop in the network, which usually results in several additional hops being added to the path between two IP hosts.



CHAPTER 13

NAT Optimized SIP Media Path with SDP

The NAT Optimized SIP Media Path with SDP feature allows the creation of a shorter path for Session Initiation Protocol (SIP) media channels by distributing endpoint IP addressing information with Session Description Protocol (SDP) of SIP messages. This feature allows endpoints to communicate directly by using standard routing and eliminates the need for them to traverse through upstream NAT routers.

The Message Digest 5 (MD5) algorithm is supported.

Information About the NAT Optimized SIP Media Path with SDP Feature

Restrictions for NAT Optimized SIP Media Path with SDP

SIP messages may or may not have SDP. This feature processes SIP messages with SDP only. If a call exchange with SDP is certain to occur, this feature should be used.

Use the “NAT - Optimized SIP Media without SPD” feature for SIP messages without SPD. This feature processes all packets sent through the NAT-enabled router but is more CPU intensive than processing SIP messages with SPD.

Benefits of NAT Optimized SIP Media Path with SDP

- The media path can be shortened, decreasing voice delay.
- More control of voice policy is possible because the media path is closer to the customer domain and not deep within the service provider cloud.

NAT Optimized SIP Media Path with SDP Feature Design

The NAT Optimized SIP Media Path with SDP feature provides the ability to optimize the media path taken by a SIP VoIP session when NAT is used. NAT forces the VoIP traffic to take at least one extra hop in the network, which usually results in several additional hops being added to the path between two IP hosts.

Cisco IOS NAT will add the relevant translation information per SIP session within the SIP protocol messages. The SIP Application Layer Gateway support within Cisco IOS NAT will extract this translation information from the SIP packets and create NAT table entries.

The "piggybacking" of NAT translation information within the SIP call flows, the design of how users interact with the application when they talk to it, will allow the media path of a SIP VoIP session between two calling parties to take the optimized routing path between each other.

How to Configure NAT Optimized SIP Media Path with SDP

Configuring a NAT Optimized SIP Media Path with SDP Messages Including MD5 Authentication

Perform this task to configure SDP messages with a NAT optimized SIP Media path including MD5 authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat piggyback-support sip-alg sdp-only router** *router-id* **md5 -authentication** *md5-authentication-key*

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 piggyback-support sip-alg sdp-only router** *router-id* **md5 -authentication** *md5-authentication-key* **Example:**	Enables SDP messages with a NAT optimized SIP Media path including MD5 authentication.

	Command or Action	Purpose
	`Router(config)# ip nat piggyback-support sip-alg sdp-only router 100 md5-authentication md5-key`	

Configuring a NAT Optimized SIP Media Path with SDP Messages Without MD5 Authentication

Perform this task to configure SDP messages with a NAT optimized SIP Media path without MD5 authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat piggyback-support sip-alg sdp-only router** *router-id*

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 piggyback-support sip-alg sdp-only router** *router-id* **Example:** `Router(config)# ip nat piggyback-support sip-alg sdp-only router 100`	Enables SDP messages with a NAT optimized SIP Media path without MD5 authentication.

Configuration Examples for NAT Optimized SIP Media Path with SDP

Configuring a NAT Optimized SIP Media Path with SDP Including MD5 Authentication Example

The following example shows how to configure a NAT optimized SIP media path with SDP including MD5 authentication:

```
ip nat piggyback-support sip-alg sdp-only router 100 md5-authentication md5-key
```

Configuring a NAT Optimized SIP Media Path with SDP Without MD5 Authentication Example

The following example shows how to configure a NAT optimized SIP media path with SDP without MD5 authentication:

```
ip nat piggyback-support sip-alg sdp-only router 100
```

Additional References

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*
NAT Optimized SIP Media Path without SDP configuration tasks and conceptual information	"NAT - Optimized SIP Media without SPD" module

Standards

Standard	Title
None	--

MIBs

MIB	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

RFC	Title
None	--

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

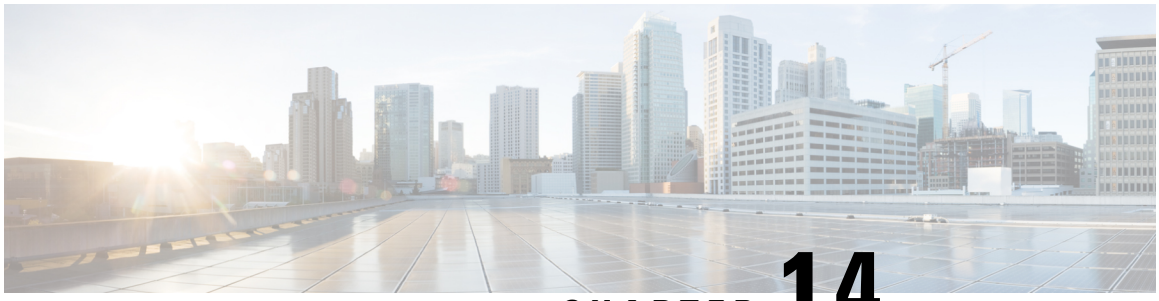
Feature Information for NAT Optimized SIP Media Path with SDP

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 <Phrase Based on Module Title>

Feature Name	Releases	Feature Information
NAT Optimized SIP Media Path with SDP	12.4(2)T	The NAT Optimized SIP Media Path with SDP feature allows the creation of a shorter path for Session Initiation Protocol (SIP) media channels by distributing endpoint IP addressing information with Session Description Protocol (SDP) of SIP messages. This feature allows endpoints to communicate directly by using standard routing and eliminates the need for them to traverse through upstream NAT routers.



CHAPTER 14

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.

Restrictions for Match-in-VRF Support for NAT

- The Match-in-VRF Support for NAT feature is not supported on interface overload configuration.
- The **match-in-vrf** keyword for intra-VPN NAT is not supported with CGN.

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.

The following table provides you information about VRF support for NAT:

NAT Inside Interface	NAT Outside Interface
Global	Global IPv4 (non-MPLS)
MPLS IP	VRF **Note** You must use the **match-in-vrf** keyword in the configuration to indicate that communication is occurring within the VRF.
VRF	VRF **Note** Both VRFs must be in the same inside interface for this configuration to work.
VRF	MPLS **Note** You must use the **match-in-vrf** keyword in the configuration to indicate that communication is occurring within the VRF.
VRF	Global IPv4 (non-MPLS)

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.
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.
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.

	Command or Action	Purpose
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.
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 for Static NAT Mapping with HSRP

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
NAT commands: complete command syntax, command mode, command history, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
IP Access List Sequence Numbering	*IP Access List Sequence Numbering* document
NAT configuration tasks	"Configuring NAT for IP Address Conservation" module
NAT maintenance	"Monitoring and Maintaining NAT" module
Using NAT with MPLS VPNs	"Integrating NAT with MPLS VPNs" module

Standards and RFCs

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

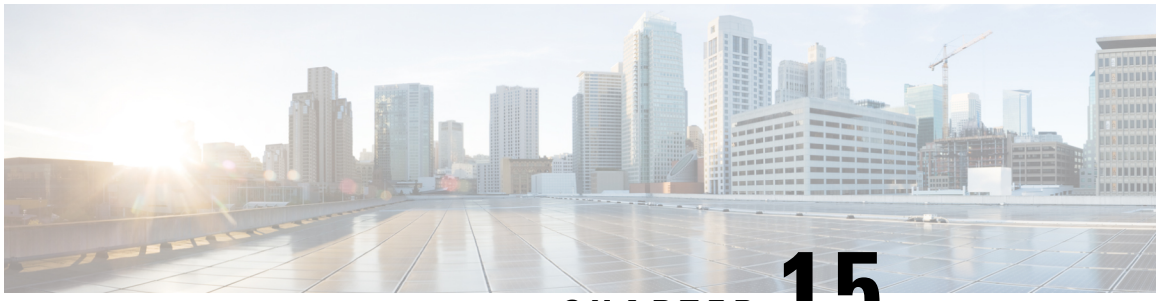
Technical Assistance

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

Table 14: 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.



CHAPTER 15

Monitoring and Maintaining NAT

This module describes how to:

- Monitor Network Address Translation (NAT) using translation information and statistical displays.
- Maintain NAT by clearing NAT translations before the timeout has expired.
- Enable the logging of NAT translation by way of syslog to log and track system error messages, exceptions, and other information.

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 Monitoring and Maintaining NAT

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

Information About Monitoring and Maintaining NAT

NAT Display Contents

There are two basic types of IP Network Address Translation (NAT) translation information:

Translation Entries

Translation entry information includes the following:

- The protocol of the port identifying the address.
- The legitimate IP address that represents one or more inside local IP addresses to the outside world.
- The IP address assigned to a host on the inside network; probably not a legitimate address assigned by the NIC or service provider.
- The IP address of an outside host as it appears to the inside network; probably not a legitimate address assigned by the NIC or service provider.
- The IP address assigned to a host on the outside network by its owner.
- The time since the entry was created (in hours:minutes:seconds).
- The time since the entry was last used (in hours:minutes:seconds).
- Flags indicating the type of translation. Possible flags are:
  - extended—Extended translation.
  - static—Static translation.
  - destination—Rotary translation.
  - outside—Outside translation.
  - timing out—Translation will no longer be used, due to a TCP finish (FIN) or reset (RST) flag.

Statistical Information

Statistical information includes the following:

- The total number of translations 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.
- A list of interfaces marked as outside with the **ip nat outside** command.
- A list of interfaces marked as inside with the **ip nat inside** command.
- The number of times the software does a translations table lookup and finds an entry.
- The number of times the software does a translations table lookup, fails to find an entry, and must try to create one.
- A cumulative count of translations that have expired since the router was booted.
- Information about dynamic mappings.
- Information about an inside source translation.
- The access list number being used for the translation.
- The name of the pool.
- The number of translations using this pool.
- The IP network mask being used in the pool.
- The starting IP address in the pool range.

- The ending IP address in the pool range.
- The type of pool. Possible types are generic or rotary.
- The number of addresses in the pool available for translation.
- The number of addresses being used.
- The number of failed allocations from the pool.

NAT does not support access control lists (ACLs) with the log option. The same functionality can be achieved by using one of the following options:

- By having a physical interface or virtual LAN (VLAN) with the logging option
- By using NetFlow

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.
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
```

Clearing NAT Entries Before the Timeout

By default, dynamic address translations will time out from the NAT translation table at some point. Perform this task to clear the entries before the timeout.

SUMMARY STEPS

1. **enable**
2. **clear ip nat translation inside** *global-ip local-ip* **outside** *local-ip global-ip*
3. **clear ip nat translation outside** *global-ip local-ip*

4. **clear ip nat translation** *protocol* **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* **outside** *local-ip global-ip* **Example:** `Device# clear ip nat translation inside 192.168.2.209 192.168.2.95 outside 192.168.2.100 192.168.2.101`	(Optional) Clears a single dynamic half-entry containing an inside translation or both an inside and outside translation created in a dynamic configuration. • A dynamic half-entry is cleared only if it 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 containing an outside translation created in a dynamic configuration. • A dynamic half-entry is cleared only if it does not have any child translations.
Step 4	**clear ip nat translation** *protocol* **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.
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 either all dynamic translations (with the ***** or **forced** keyword), a single dynamic half-entry containing an inside translation, or a single dynamic half-entry containing an outside translation. • A single dynamic half-entry is cleared only if it 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 192.168.2.209 192.168.2.195 forced`	(Optional) Forces the clearing of a single dynamic half-entry and its child translations containing an inside translation created in a dynamic configuration, with or without its corresponding outside translation. • A dynamic half-entry is always cleared, regardless of whether it has any child translations.

	Command or Action	Purpose
Step 7	**clear ip nat translation outside** *local-ip global-ip* [**forced**] **Example:** `Device# clear ip nat translation outside 192.168.2.100 192.168.2.80 forced`	(Optional) Forces the clearing of a single dynamic half-entry and its child translations containing an outside translation created in a dynamic configuration. • A dynamic half-entry is always cleared, regardless of whether it has any child translations.

Examples for Monitoring and Maintaining NAT

Example: Clearing UDP NAT Translations

The following example shows the Network Address Translation (NAT) entries before and after the UDP entry is cleared:

```
Device# show ip nat translation
Pro Inside global        Inside local         Outside local      Outside global
udp 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
tcp 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 outside
192.168.2.20:23 192.168.2.20:23
Device# show ip nat translation

Pro Inside global      Inside local        Outside local      Outside global
udp 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
```

Where to Go Next

- To configure NAT for use with application level gateways, see the “Using Application Level Gateways with NAT” module.
- To integrate NAT with MPLS VPNs, see the “Integrating NAT with MPLS VPNs” module.
- To configure NAT for high availability, see the “Configuring NAT for High Availability” module.

Additional References for Monitoring and Maintaining NAT

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

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
NAT for IP address conservation	"Configuring NAT for IP Address Conservation" module

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

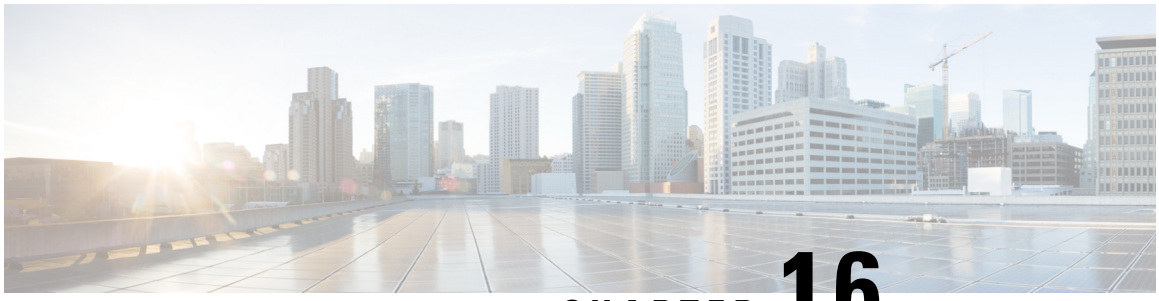
Feature Information for Monitoring and Maintaining NAT

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 15: Feature Information for Monitoring and Maintaining NAT

Feature Name	Releases	Feature Information
NAT—Forced Clear of Dynamic NAT Half-Entries	12.2(15)T	A second **forced** keyword was added to the **clear ip nat translation** command to enable the removal of half-entries regardless of whether they have any child translations.



CHAPTER 16

NAT-PT for IPv6

NAT—PT is an IPv6-to-IPv4 translation mechanism, as defined in RFC 2765 and RFC 2766, that allows IPv6-only devices to communicate with IPv4-only devices and vice versa.

This modules describes Network Address Translation (NAT)—Protocol Translation (PT) and explains how to configure the feature.

Prerequisites for NAT-PT for IPv6

Before implementing the NAT-PT for IPv6 feature, you must configure IPv4 and IPv6 on device interfaces that need to communicate between IPv4-only and IPv6-only networks.

Restrictions for NAT-PT for IPv6

- Network Address Translation (NAT)-Protocol Translation (PT) is not supported with Cisco Express Forwarding.
- NAT-PT supports only Domain Naming System (DNS), File Transfer Protocol (FTP), and Internet Control Message Protocol (ICMP) application-layer gateways (ALGs).
- NAT-PT does not provide end-to-end security to networks. The device on which NAT-PT is configured can be a single point of failure in the network.
- Bridge-group virtual interfaces (BVIs) in IPv6 are not supported with NAT-PT and wireless interfaces Dot11Radio.

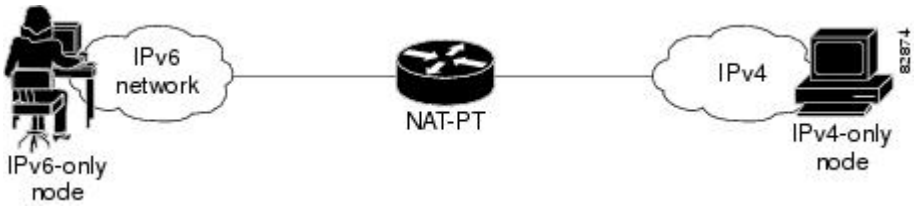
Information for NAT-PT for IPv6

NAT-PT Overview

Network Address Translation (NAT)-Port Translation (PT) for Cisco software based on RFC 2766 and RFC 2765 is a migration tool that helps customers transition their IPv4 networks to IPv6 networks. Using a protocol translator between IPv6 and IPv4 allows direct communication between hosts that use different network protocols. You can use static, dynamic, port address translation, IPv4-mapped definitions for NAT-PT operation.

The figure below shows that NAT-PT runs on a device that is configured between an IPv6 network and an IPv4 network that helps connect an IPv6-only node with an IPv4-only node.

Figure 17: NAT-PT Basic Operation



NAT-PT allows direct communication between IPv6-only networks and IPv4-only networks. Dual-stack networks (networks that have IPv4 and IPv6) can have some IPv6-only hosts configured to take advantage of the IPv6 autoconfiguration, global addressing, and simpler management features, and these hosts can use NAT-PT to communicate with existing IPv4-only networks in the same organization.

One of the benefits of NAT-PT is that no changes are required to existing hosts if NAT-PT is configured, because all NAT-PT configurations are performed at the NAT-PT device. Stable IPv4 networks can introduce an IPv6 network and use NAT-PT to communicate between these networks without disrupting the network. For a seamless transition, you can use FTP between IPv4 and IPv6 hosts.

When you configure IPv6, packet fragmentation is enabled by default, to allow IPv4 and IPv6 networks to resolve fragmentation problems. Without the ability to resolve fragmentation, connectivity can be intermittent when fragmented packets are dropped or not interpreted correctly.

We do not recommend the use of NAT-PT to communicate between a dual-stack host and an IPv6-only or IPv4-only host. We do not recommend the use of NAT-PT in a scenario in which an IPv6-only network tries to communicate with another IPv6-only network via an IPv4 backbone or vice versa, because NAT-PT requires a double translation. You can use tunneling techniques for communication in these scenarios.

You can configure one the following operations for NAT-PT, but not all four.

Static NAT-PT Operation

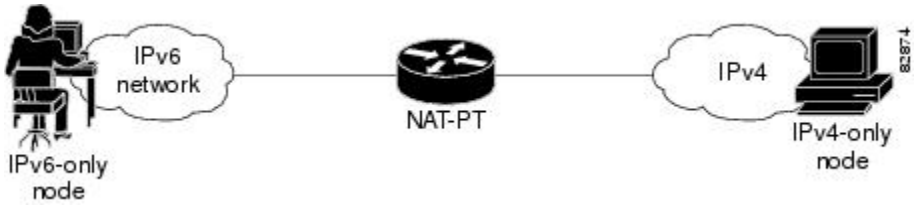
Static NAT-PT uses static translation rules to map an IPv6 address to an IPv4 address. IPv6 network nodes communicate with IPv4 network nodes using an IPv6 mapping of the IPv4 address that is configured on the NAT-PT device.

The figure below shows how the IPv6-only node named A can communicate with the IPv4-only node named C using NAT-PT. The NAT-PT device is configured to map the source IPv6 address for node A of 2001:DB8:bbbb:1::1 to the IPv4 address 192.168.99.2. NAT-PT is also configured to map the source address of IPv4 node C, 192.168.30.1 to 2001:DB8::a. When packets with a source IPv6 address of node A are received

at the NAT-PT device, these packets are translated to have a destination address that matches node C in the IPv4-only network. You can also configure NAT-PT to match a source IPv4 address and translate the packet to an IPv6 destination address to allow an IPv4-only host to communicate with an IPv6-only host.

If you have multiple IPv6-only or IPv4-only hosts, you may need to configure multiple static NAT-PT mappings. Static NAT-PT is useful when applications or servers require access to a stable IPv4 address, such as accessing an external IPv4 Domain Name System (DNS) server.

Figure 18: Static NAT-PT Operation

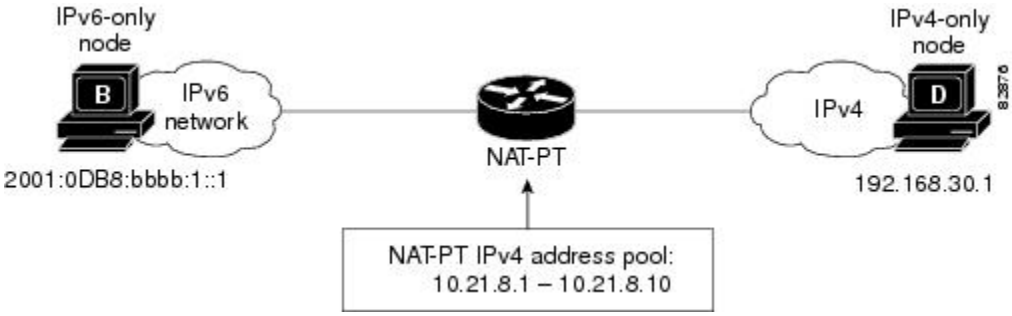


Dynamic NAT-PT Operation

Dynamic NAT-PT allows multiple NAT-PT mappings by allocating addresses from a pool of addresses. NAT-PT is configured with a pool of IPv6 and/or IPv4 addresses. At the start of a NAT-PT session a temporary address is dynamically allocated from this pool. The number of addresses available in the address pool determines the maximum number of concurrent sessions. The NAT-PT device records each mapping between addresses in a dynamic state table.

The figure below shows how dynamic NAT-PT operates. The IPv6-only node B can communicate with the IPv4-only node D using dynamic NAT-PT. The NAT-PT device is configured with an IPv6 access list, prefix list, or route map to determine which packets are to be translated by NAT-PT. A pool of IPv4 addresses--10.21.8.1 to 10.21.8.10 in the figure -- is also configured. When an IPv6 packet to be translated is identified, NAT-PT uses the configured mapping rules and assigns a temporary IPv4 address from the configured pool of IPv4 addresses.

Figure 19: Dynamic NAT-PT Operation



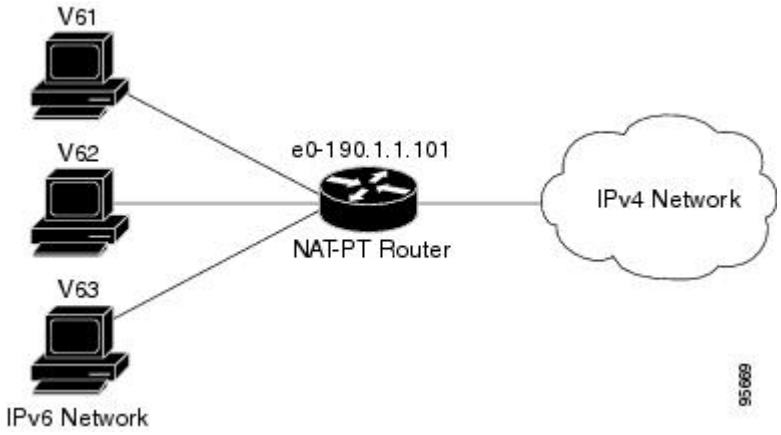
Dynamic NAT-PT translation operation requires at least one static mapping for the IPv4 Domain Name System (DNS) server.

After the IPv6 to IPv4 connection is established, reply packets going from IPv4 to IPv6 uses the previously established dynamic mapping to translate back from IPv4 to IPv6 and vice versa for an IPv4-only host.

Port Address Translation

Port Address Translation (PAT), also known as overload configuration, allows a single IPv4 address to be used among multiple sessions by multiplexing on the port number to associate several IPv6 users with a single IPv4 address. PAT can be accomplished through a specific interface or through a pool of addresses. The figure below shows multiple IPv6 addresses from the IPv6 network that is linked to a single IPv4 interface into the IPv4 network.

Figure 20: Port Address Translation



IPv4-Mapped Operation

You can send traffic from your IPv6 network to an IPv4 network without configuring the IPv6 destination address mapping. A packet that arrives at an interface is checked to discover if it has a NAT-PT prefix that was configured with the **ipv6 nat prefix v4-mapped** command. If the prefix matches, then an access-list check is performed to discover if the source address matches the access list or prefix list. If the prefix does not match, the packet is dropped. If the prefix matches, the source address translation is performed.

If a rule is configured for the source address translation, the last 32 bits of the destination IPv6 address is used as the IPv4 destination and a flow entry is created.

With an IPv4-mapping configuration on a device, when the Domain Name System (DNS) application-level gateway (ALG) IPv4 address is converted to an IPv6 address, the IPv6 address is processed and ALGs of the DNS packets from IPv4 network is translated into the IPv6 network.

How to Configure NAT-PT for IPv6

Configuring Basic IPv6 to IPv4 Connectivity for NAT-PT for IPv6

Perform this task to configure basic IPv6 to IPv4 connectivity for NAT-PT, which consists of configuring the NAT-PT prefix globally, and enable NAT-PT on an interface. For NAT-PT to be operational, NAT-PT must be enabled on both the incoming and outgoing interfaces.

An IPv6 prefix with a prefix length of 96 must be specified for NAT-PT to use. The IPv6 prefix can be a unique local unicast prefix, a subnet of your allocated IPv6 prefix, or even an extra prefix obtained from your

Internet service provider (ISP). The NAT-PT prefix is used to match a destination address of an IPv6 packet. If the match is successful, NAT-PT will use the configured address mapping rules to translate the IPv6 packet to an IPv4 packet. The NAT-PT prefix can be configured globally or with different IPv6 prefixes on individual interfaces. Using a different NAT-PT prefix on several interfaces allows the NAT-PT router to support an IPv6 network with multiple exit points to IPv4 networks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nat prefix** *ipv6-prefix* / *prefix-length*
4. **interface** *type number*
5. **ipv6 address** *ipv6-address* {/*prefix-length* | **link-local**}
6. **ipv6 nat**
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask* [**secondary**]
10. **ipv6 nat**

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 nat prefix** *ipv6-prefix* / *prefix-length* **Example:** `Router# ipv6 nat prefix 2001:DB8::/96`	Assigns an IPv6 prefix as a global NAT-PT prefix. • Matching destination prefixes in IPv6 packets are translated by NAT-PT. • The only prefix length supported is 96.
Step 4	**interface** *type number* **Example:** `Router(config)# interface ethernet 3/1`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 5	**ipv6 address** *ipv6-address* {/*prefix-length*	**link-local**} **Example:** `Router(config-if)# ipv6 address 2001:DB8:yyyy:1::9/64`	Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface.

	Command or Action	Purpose
Step 6	**ipv6 nat** **Example:** `Router(config-if)# ipv6 nat`	Enables NAT-PT on the interface.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode, and returns the router to global configuration mode.
Step 8	**interface** *type number* **Example:** `Router(config)# interface ethernet 3/3`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 9	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 192.168.30.9 255.255.255.0`	Specifies an IP address and mask assigned to the interface and enables IP processing on the interface.
Step 10	**ipv6 nat** **Example:** `Router(config-if)# ipv6 nat`	Enables NAT-PT on the interface.

Configuring IPv4-Mapped NAT-PT

Perform this task to enable customers to send traffic from their IPv6 network to an IPv4 network without configuring IPv6 destination address mapping. This task shows the **ipv6 nat prefix v4-mapped** command configured on a specified interface, but the command could alternatively be configured globally:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nat prefix** *ipv6-prefix* **v4-mapped** {*access-list-name* | *ipv6-prefix*}

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 ethernet 3/1`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**ipv6 nat prefix** *ipv6-prefix* **v4-mapped** {*access-list-name* | *ipv6-prefix*} **Example:** `Router(config-if)# ipv6 nat prefix 2001::/96 v4-mapped v4mapacl`	Enables customers to send traffic from their IPv6 network to an IPv4 network without configuring IPv6 destination address mapping.

Configuring Mappings for IPv6 Hosts Accessing IPv4 Hosts

Perform this task to configure static or dynamic IPv6 to IPv4 address mappings. The dynamic address mappings include assigning a pool of IPv4 addresses and using an access list, prefix list, or route map to define which packets are to be translated.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Configure one of the following commands:
   - **ipv6 nat v6v4 source** *ipv6-address ipv4-address*
   - **ipv6 nat v6v4 source** {**list** *access-list-name* | **route-map** *map-name*} **pool** *name*
4. **ipv6 nat v6v4 pool** *name start-ipv4 end-ipv4* **prefix-length** *prefix-length*
5. **ipv6 nat translation** [**max-entries** *number*] {**timeout** | **udp-timeout** | **dns-timeout** | **tcp-timeout** | **finrst-timeout** | **icmp-timeout**} {*seconds* | **never**}
6. **ipv6 access-list** *access-list-name*
7. **permit** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*}
8. **end**
9. **show ipv6 nat translations** [**icmp** | **tcp** | **udp**] [**verbose**]
10. **show ipv6 nat statistics**

DETAILED STEPS

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

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

Step 3	Configure one of the following commands:
	• **ipv6 nat v6v4 source** *ipv6-address ipv4-address*
	• **ipv6 nat v6v4 source** {**list** *access-list-name* | **route-map** *map-name*} **pool** *name*
	Example:
	`Device(config)# ipv6 nat v6v4 source 2001:DB8:yyyy:1::1 10.21.8.10`
	`Device(config)# ipv6 nat v6v4 source list pt-list1 pool v4pool`
		Enables a static IPv6 to IPv4 address mapping using NAT-PT.
		or
		Enables a dynamic IPv6 to IPv4 address mapping using NAT-PT.

Step 4	**ipv6 nat v6v4 pool** *name start-ipv4 end-ipv4* **prefix-length** *prefix-length* **Example:** `Device(config)# ipv6 nat v6v4 pool v4pool 10.21.8.1 10.21.8.10 prefix-length 24`	Specifies a pool of IPv4 addresses to be used by NAT-PT for dynamic address mapping.
Step 5	**ipv6 nat translation** [**max-entries** *number*] {**timeout** | **udp-timeout** | **dns-timeout** | **tcp-timeout** | **finrst-timeout** | **icmp-timeout**} {*seconds* | **never**} **Example:** `Device(config)# ipv6 nat translation udp-timeout 600`	(Optional) Specifies the time after which NAT-PT translations time out.
Step 6	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list pt-list1`	(Optional) Defines an IPv6 access list and enters IPv6 access list configuration mode. • The *access-list name* argument specifies the name of the IPv6 access control list (ACL). IPv6 ACL names cannot contain a space or quotation mark, or begin with a numeral.
Step 7	**permit** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} **Example:** `Device(config-ipv6-acl)# permit ipv6 2001:DB8:bbbb:1::/64 any`	(Optional) Specifies permit conditions for an IPv6 ACL.

	Command or Action	Purpose
Step 8	**end** **Example:** `Device(config-ipv6-acl)# end`	Exits IPv6 access list configuration mode, and returns to privileged EXEC mode.
Step 9	**show ipv6 nat translations** [**icmp** | **tcp** | **udp**] [**verbose**] **Example:** `Device# show ipv6 nat translations verbose`	(Optional) Displays active NAT-PT translations. • Use the optional **icmp**, **tcp**, and **udp** keywords to display detailed information about the NAT-PT translation events for the specified protocol. • Use the optional **verbose** keyword to display more detailed information about the active translations.
Step 10	**show ipv6 nat statistics** **Example:** `Device# show ipv6 nat statistics`	(Optional) Displays NAT-PT statistics.

Configuring Mappings for IPv4 Hosts Accessing IPv6 Hosts

Perform this optional task to configure static or dynamic IPv4 to IPv6 address mappings. The dynamic address mappings include assigning a pool of IPv6 addresses and using an access list, prefix list, or route map to define which packets are to be translated.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Configure one of the following commands:
   - **ipv6 nat v4v6 source** *ipv6-address ipv4-address*
   - **ipv6 nat v4v6 source list** {*access-list-number* | *name*} **pool** *name*
4. **ipv6 nat v4v6 pool** *name start-ipv6 end-ipv6* **prefix-length** *prefix-length*
5. **access-list** {*access-list-name* | *number*} {**deny** | **permit**} [*source source-wildcard*] [**log**]
6. **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	Configure one of the following commands:
	• **ipv6 nat v4v6 source** *ipv6-address ipv4-address*
	• **ipv6 nat v4v6 source list** {*access-list-number* | *name*} **pool** *name*
	Example:
	`Device(config)# ipv6 nat v4v6 source 10.21.8.11 2001:DB8:yyyy::2`
	`Device(config)# ipv6 nat v4v6 source list 1 pool v6pool`
		Enables a static IPv4 to IPv6 address mapping using NAT-PT.
		or
		Enables a dynamic IPv4 to IPv6 address mapping using NAT-PT.

Step 4	**ipv6 nat v4v6 pool** *name start-ipv6 end-ipv6* **prefix-length** *prefix-length* **Example:** `Device(config)# ipv6 nat v4v6 pool v6pool 2001:DB8:yyyy::1 2001:DB8:yyyy::2 prefix-length 128`	Specifies a pool of IPv6 addresses to be used by NAT-PT for dynamic address mapping.
Step 5	**access-list** {*access-list-name* | *number*} {**deny** | **permit**} [*source source-wildcard*] [**log**] **Example:** `Device(config)# access-list 1 permit 192.168.30.0 0.0.0.255`	Specifies an entry in a standard IPv4 access list.
Step 6	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring PAT for IPv6 to IPv4 Address Mappings

Perform this task to configure Port Address Translation (PAT) for IPv6 to IPv4 address mappings. Multiple IPv6 addresses are mapped to a single IPv4 address or to a pool of IPv4 addresses. Use an access list, a prefix list, or a route map to define which packets must be translated.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Configure one of the following commands:
   - **ipv6 nat v6v4 source** {**list** *access-list-name* \| **route-map** *map-name*} **pool** *name overload*
   - **ipv6 nat v6v4 source** {**list** *access-list-name* \| **route-map** *map-name*} **interface** *interface name* **overload**
4. **ipv6 nat v6v4 pool** *name start-ipv4 end-ipv4* **prefix-length** *prefix-length*
5. **ipv6 nat translation** [**max-entries** *number*] {**timeout** \| **udp-timeout** \| **dns-timeout** \| **tcp-timeout** \| **finrst-timeout** \| **icmp-timeout**} {*seconds* \| **never**}
6. **ipv6 access-list** *access-list-name*

7. **permit** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*}
8. **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	Configure one of the following commands:
	• **ipv6 nat v6v4 source** {**list** *access-list-name* | **route-map** *map-name*} **pool** *name overload*
	• **ipv6 nat v6v4 source** {**list** *access-list-name* | **route-map** *map-name*} **interface** *interface name* **overload**
	Example:
	`Device(config)# ipv6 nat v6v4 source 2001:DB8:yyyy:1::1 10.21.8.10`
	`Device(config)# ipv6 nat v6v4 source list pt-list1 pool v4pool overload`
		Enables a dynamic IPv6 to IPv4 address overload mapping using a pool address.
		or
		Enables a dynamic IPv6 to IPv4 address overload mapping using an interface address.

Step 4	**ipv6 nat v6v4 pool** *name start-ipv4 end-ipv4* **prefix-length** *prefix-length* **Example:** `Device(config)# ipv6 nat v6v4 pool v4pool 10.21.8.1 10.21.8.10 prefix-length 24`	Specifies a pool of IPv4 addresses to be used by NAT-PT for dynamic address mapping.
Step 5	**ipv6 nat translation** [**max-entries** *number*] {**timeout** | **udp-timeout** | **dns-timeout** | **tcp-timeout** | **finrst-timeout** | **icmp-timeout**} {*seconds* | **never**} **Example:** `Device(config)# ipv6 nat translation udp-timeout 600`	(Optional) Specifies the time after which NAT-PT translations time out.
Step 6	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list pt-list1`	(Optional) Defines an IPv6 access list and enters IPv6 access list configuration mode. • IPv6 ACL names cannot contain a space or quotation mark, or begin with a numeral.
Step 7	**permit** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]]	(Optional) Specifies permit conditions for an IPv6 ACL.

	Command or Action	Purpose
	{*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} **Example:** `Device(config-ipv6-acl)# permit ipv6 2001:DB8:bbbb:1::/64 any`	
Step 8	**end** **Example:** `Device(config-ipv6-acl)# end`	Exits IPv6 access list configuration mode and returns to privileged EXEC mode.

Verifying NAT-PT Configuration and Operation

These commands are optional. Use these commands in any order.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 nat translation ***
3. **debug ipv6 nat** [**detailed** | **port**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ipv6 nat translation *** **Example:** `Device# clear ipv6 nat translation *`	Clears dynamic Network Address Translation (NAT)-Port Translation (PT) entries from the dynamic translation state table. • Use the ***** keyword to clear all dynamic NAT-PT translations. **Note** Static translation configuration is not affected by this command.
Step 3	**debug ipv6 nat** [**detailed** | **port**] **Example:** `Device# debug ipv6 nat detail`	Displays debugging messages for NAT-PT translation events.

Configuration Examples for NAT-PT for IPv6

Example: Static NAT-PT Configuration

The following example configures the NAT-PT prefix globally, enables NAT-PT on two interfaces, and configures two static NAT-PT mappings. Ethernet interface 3/1 is configured as IPv6 only, and Ethernet interface 3/3 is configured as IPv4 only.

```
interface Ethernet3/1
 ipv6 address 2001:DB8:3002::9/64
 ipv6 enable
 ipv6 nat
!
interface Ethernet3/3
 ip address 192.168.30.9 255.255.255.0
 ipv6 nat
!
ipv6 nat v4v6 source 192.168.30.1 2001:DB8:0::2
ipv6 nat v6v4 source 2001:DB8:bbbb:1::1 10.21.8.10
ipv6 nat prefix 2001:DB8:0::/96
```

Example: Configuring IPv4-Mapped NAT-PT

The following example shows an access list that permits any IPv6 source address with the prefix 2001::/96 to enter the destination with the 2000::/96 prefix. The destination is translated to the last 32 bit of its IPv6 address; for example: source address is 2001::1 and destination address is 2000::192.168.1.1. The destination is translated to 192.168.1.1 in the IPv4 network.

```
interface gigabitethernet 3/1/1
 ipv6 nat prefix 2000::/96 v4-mapped v4map-acl
 ipv6 access-list v4map-acl
  permit ipv6 2001::/96 2000::/96
```

Example: Dynamic NAT-PT Configuration for IPv6 Hosts Accessing IPv4 Hosts

The following example configures the NAT-PT prefix globally, enables NAT-PT on two interfaces, and configures one static NAT-PT mapping (used, for example, to access a DNS server). A dynamic NAT-PT mapping is also configured to map IPv6 addresses to IPv4 addresses using a pool of IPv4 addresses named v4pool. The packets to be translated by NAT-PT are filtered using an IPv6 access list named pt-list1. The User Datagram Protocol (UDP) translation entries are configured to time out after 10 minutes. Ethernet interface 3/1 is configured as IPv6 only, and Ethernet interface 3/3 is configured as IPv4 only.

```
interface Ethernet3/1
 ipv6 address 2001:DB8:bbbb:1::9/64
 ipv6 enable
 ipv6 nat
!
interface Ethernet3/3
 ip address 192.168.30.9 255.255.255.0
 ipv6 nat
!
ipv6 nat v4v6 source 192.168.30.1 2001:DB8:0::2
ipv6 nat v6v4 source list pt-list1 pool v4pool
```

```
ipv6 nat v6v4 pool v4pool 10.21.8.1 10.21.8.10 prefix-length 24
ipv6 nat translation udp-timeout 600
ipv6 nat prefix 2001:DB8:1::/96
!
ipv6 access-list pt-list1
 permit ipv6 2001:DB8:bbbb:1::/64 any
```

Example: Dynamic NAT-PT Configuration for IPv4 Hosts Accessing IPv6 Hosts

The following example configures the NAT-PT prefix globally, enables NAT-PT on two interfaces, and configures one static NAT-PT mapping (used, for example, to access a DNS server). A dynamic NAT-PT mapping is also configured to map IPv4 addresses to IPv6 addresses using a pool of IPv6 addresses named v6pool. The packets to be translated by NAT-PT are filtered using an access list named pt-list2. Ethernet interface 3/1 is configured as IPv6 only, and Ethernet interface 3/3 is configured as IPv4 only.

```
interface Ethernet3/1
 ipv6 address 2001:DB8:bbbb:1::9/64
 ipv6 enable
 ipv6 nat
!
interface Ethernet3/3
 ip address 192.168.30.9 255.255.255.0
 ipv6 nat
!
ipv6 nat v4v6 source list 72 pool v6pool
ipv6 nat v4v6 pool v6pool 2001:DB8:0::1 2001:DB8:0::2 prefix-length 128
ipv6 nat v6v4 source 2001:DB8:bbbb:1::1 10.21.8.0
ipv6 nat prefix 2001:DB8:0::/96
!
access-list 72 permit 192.168.30.0 0.0.0.255
```

Example: Displaying Dynamic NAT-PT Translations

The following example shows how all dynamic NAT-PT translations are cleared from the dynamic translation state table using the **clear ipv6 nat translation *** command. After configuring the **clear** command, when you configure the **show ipv6 nat translations** command, only static translation configurations are displayed.

```
Device# clear ipv6 nat translation *

Device# show ipv6 nat translations

Prot  IPv4 source              IPv6 source
      IPv4 destination         IPv6 destination
---   ---                      ---
      192.168.123.2            2001:DB8::2

---   ---                      ---
      192.168.122.10           2001:DB8::10

---   192.168.124.8            2001:DB8:3::8
      ---                      ---

---   192.168.121.4            2001:DB8:5::4
      ---                      ---
```

Example: Displaying Active NAT-PT Translations

The following sample output from the **show ipv6 nat translations** command displays information about active Network Address Translation (NAT)-Port Translation (PT) translations:

```
Device# show ipv6 nat translations

Prot  IPv4 source              IPv6 source
      IPv4 destination         IPv6 destination
---   ---                      ---
      192.168.123.2            2001:DB8::2

---   ---                      ---
      192.168.122.10           2001:DB8::10

tcp   192.168.124.8,11047      2001:DB8:3::8,11047
      192.168.123.2,23         2001:DB8::2,23

udp   192.168.124.8,52922      2001:DB8:3::8,52922
      192.168.123.2,69         2001::2,69

udp   192.168.124.8,52922      2001:DB8:3::8,52922
      192.168.123.2,52922      2001:DB8::2,52922

---   192.168.124.8            2001:DB8:3::8
      192.168.123.2            2001:DB8::2

---   192.168.124.8            2001:DB8:3::8
      ---                      --- ---
```

Example: Displaying Information About NAT-PT Statistics

```
Router# show ipv6 nat statistics

Total active translations: 4 (4 static, 0 dynamic; 0 extended)
NAT-PT interfaces:
  Ethernet3/1, Ethernet3/3
Hits: 0  Misses: 0
Expired translations: 0
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	*Master Commands List, All Releases*
IPv6 commands	*IPv6 Command Reference*
Cisco IOS IPv6 features	*IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

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

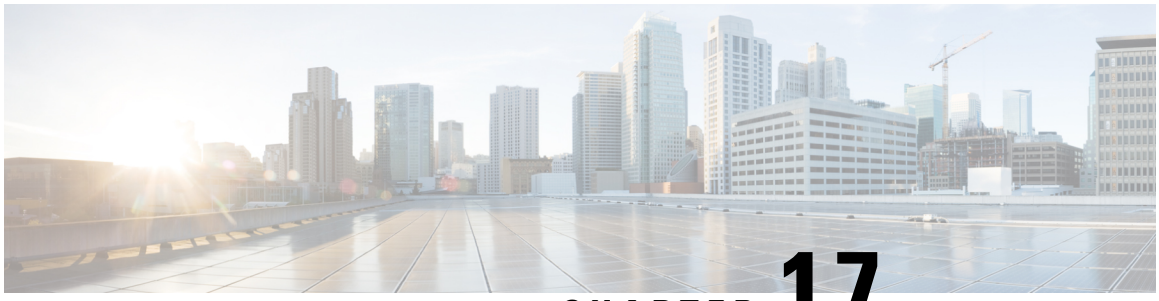
Feature Information for NAT-PT for IPv6

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 NAT-PT for IPv6

Feature Name	Releases	Feature Information
NAT-PT: Support for DNS ALG	12.2(13)T	IPv6 provides DNS ALG support.
NAT-PT: Support for FTP ALG	12.3(2)T	IPv6 provides FTP ALG support.
NAT-PT: Support for Fragmentation	12.3(2)T	Packet fragmentation is enabled by default when IPv6 is configured, allowing IPv6 and IPv4 networks to resolve fragmentation problems between the networks.
NAT-PT: Support for Overload	12.3(2)T	This feature allows a single IPv4 address to be used among multiple sessions by multiplexing on the port number to associate several IPv6 users with a single IPv4 address.



CHAPTER 17

NAT TCP SIP ALG Support

The NAT TCP SIP ALG Support feature allows embedded messages of the Session Initiation Protocol (SIP) passing through a device that is configured with Network Address Translation (NAT) to be translated and encoded back to the packet. An application-layer gateway (ALG) is used with NAT to translate the SIP or Session Description Protocol (SDP) messages.

This module describes the NAT TCP SIP ALG Support feature and explains how to configure it.

Prerequisites for NAT TCP SIP ALG Support

Layer 4 Forwarding (L4F) must be enabled for the feature to function.

Restrictions for NAT TCP SIP ALG Support

- Network Address Translation (NAT) translates only embedded IPv4 addresses.
- NAT application-layer gateway (ALG) fixup for Session Initiation Protocol (SIP) messages over TCP is not done when Layer 4 Forwarding (L4F) functionality is disabled. In this case, SIP messages are considered as TCP messages and only Layer 3 and Layer 4 fixups are done.
- As per RFC 5128, NAT TCP SIP ALG feature uses Endpoint-Independent mapping to perform address translations. This combination allows incoming SIP traffic from any external endpoint on the public network to a mapped public port. If you do not need Endpoint-Independent mapping, use ACL or Zone-based Policy Firewall to limit the scope of incoming traffic.

Information About NAT TCP SIP ALG Support

NAT TCP SIP ALG Support Overview

The NAT TCP SIP ALG Support feature allows embedded messages of the Session Initiation Protocol (SIP) passing through a device that is configured with Network Address Translation (NAT) to be translated and encoded back to the packet. An application-layer gateway (ALG) is used with NAT to translate the SIP or Session Description Protocol (SDP) messages. The NAT TCP SIP ALG Support feature adds NAT ALG support for fixing up TCP-based SIP messages.

Session Initiation Protocol (SIP) is an ASCII-based, application-layer control protocol that can be used to establish, maintain, and terminate calls between two or more endpoints. SIP is a protocol developed by IETF for multimedia conferencing over IP. SIP can be configured to operate over TCP-based transports. Cisco SIP implementation enables supported Cisco platforms to signal the setup of voice and multimedia calls over IP networks. SIP provides an alternative to H.323 within the VoIP internetworking software.

Like other VoIP protocols, SIP is designed to address functions of signaling and session management within a packet telephony network. Signaling allows call information to be carried across network boundaries. Session management provides the ability to control attributes of an end-to-end call.

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

SIP Messages

Entities that are present in a Session Initiation Protocol (SIP) deployment communicate with each other by using well-defined SIP messages that take the form of requests and responses. These SIP messages can contain embedded IP address or port information that might belong to a private domain, and such messages must be fixed up when they pass through a Network Address Translation (NAT) device. Fixup denotes the writing of the translated IP address back into the packet. This fixup is normally performed by an application-layer gateway (also called an application-level gateway) (ALG) module that resides on the NAT device.

By default, support for SIP is enabled on the standard TCP port 5060 to exchange SIP messages. You can also configure nonstandard ports for SIP to operate. NAT ALG accepts and attempts fixup operations on all TCP segments that originate from or are destined to the configured SIP port. SIP message processing involves performing the fixup operation on a complete SIP message. A TCP segment may carry multiple SIP messages. It is also possible that a SIP message is segmented and carried in two different TCP segments.

SIP messages are text based. Any adjustment that is made to the message as part of the ALG fixup can result in the message to increase or decrease in size. A change in the message size means that the ALG must make adjustments to the TCP sequence or acknowledgment numbers and keep track of the same. There are cases where the ALG must perform spoof acknowledgments and complete TCP retransmission.

TCP proxy is an essential component that terminates a TCP connection passing through NAT ALG and regenerates the TCP connection. This connection allows NAT ALG to modify the TCP payload without any TCP session handling issues.

The table below identifies the six available SIP request messages.

Table 17: SIP Request Messages

SIP Message	Purpose
ACK	Sent by calling party to confirm the receipt of a final response to INVITE.
BYE	Sent by calling party or called party to end a call.
CANCEL	Sent to end a call that has not yet been connected.
INVITE	Request sent from a User Agent Client (UAC) to initiate a session.
OPTIONS	Sent to query capabilities of UACs and network servers.
REGISTER	Sent by the client to register the address with a SIP proxy.

The table below identifies the available SIP response methods.

Table 18: SIP Response Messages

SIP Message	Purpose
1xx (Informational)	• 100 = Trying • 180 = Ringing • 181 = Call Is Being Forwarded • 182 = Queued • 183 = Session Progress
2xx (Successful)	• 200 = OK
3xx (Redirection)	• 300 = Multiple Choices • 301 = Moved Permanently • 302 = Moved Temporarily • 303 = See Other • 305 = Use Proxy • 380 = Alternative Service

SIP Message	Purpose
4xx (Request Failure)	• 400 = Bad Request • 401 = Unauthorized • 402 = Payment Required • 403 = Forbidden • 404 = Not Found • 405 = Method Not Allowed • 406 = Not Acceptable • 407 = Proxy Authentication Required • 408 = Request Timeout • 409 = Conflict • 410 = Gone • 411 = Length Required • 413 = Request Entity Too Large • 414 = Request URI Too Large • 415 = Unsupported Media Type • 420 = Bad Extension • 480 = Temporarily Not Available • 481 = Call Leg/Transaction Does Not Exist • 482 = Loop Detected • 483 = Too Many Hops • 484 = Address Incomplete • 485 = Ambiguous • 486 - Busy Here
5xx (Server Failure)	• 500 = Internal Server Error • 501 = Not Implemented • 502 = Bad Gateway • 503 = Service Unavailable • 504 = Gateway Timeout • 505 = SIP Version Not Supported

SIP Message	Purpose
6xx (Global Failure)	• 600 = Busy Anywhere • 603 = Decline • 604 = Does Not Exist Anywhere • 606 = Not Acceptable

SIP Functionality

Users in a SIP network are identified by unique SIP addresses. A SIP address is similar to an e-mail address and is in the format sip:*userID@gateway*.com. The *userID* can be either a username or an E.164 address. The *gateway* can be either a domain (with or without a hostname) or a specific internet IP address.

Note An E.164 address is a telephone number with a string of decimal digits, which uniquely indicates the public network termination point. This address contains all information that is necessary to route a call to a termination point.

Users register with a registrar server using their assigned SIP addresses. The registrar server provides SIP addresses to the location server on request. The registrar server processes requests from user-agent clients (UACs) for registration of their current locations.

When a user initiates a call, a SIP request is sent to a SIP server (either a proxy or a redirect server). The request includes the address of the caller (in the From header field) and the address of the intended called party (in the To header field).

A SIP end user might move between end systems. The location of the end user can be dynamically registered with the SIP server. The location server can use one or more protocols (including Finger, RWhois, and Lightweight Directory Access Protocol [LDAP]) to locate the end user. Because the end user can be logged in at more than one station and the location server can sometimes have inaccurate information, the location server might return more than one address for the end user. If the request is coming through a SIP proxy server, the proxy server tries each of the returned addresses until it locates the end user. If the request is coming through a SIP redirect server, the redirect server forwards all the addresses to the caller available in the Contact header field of the invitation response.

SIP Functionality with a Proxy Server

A proxy server receives Session Initiation Protocol (SIP) requests from a client and forwards them on the client's behalf. Proxy servers receive SIP messages and forward them to the next SIP server in the network. Proxy servers can provide functions such as authentication, authorization, network access control, routing, reliable request retransmission, and security.

SIP is a peer-to-peer protocol. The peers in a session are called user agents (UAs).

When communicating through a proxy server, the caller UA sends an INVITE request to the proxy server and then the proxy server determines the path and forwards the request to the called party. The called UA responds to the proxy server, which then forwards the response to the caller. When both parties respond with an acknowledgment (SIP ACK message), the proxy server forwards the acknowledgments to their intended party

and a session, or conference, is established between them. The Real-time Transfer Protocol (RTP) is then used for communication across the connection now established between the caller and called UA.

How to Configure NAT TCP SIP ALG Support

Specifying a Port for NAT TCP SIP ALG Support

Network Address Translation (NAT) support for Session Initiation Protocol (SIP) is enabled by default. SIP uses the default TCP port 5060 to exchange messages. If required, you can configure a different port to handle SIP messages.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat service sip tcp port** *port-number*
4. **end**
5. **debug ip nat sip**

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 sip tcp port** *port-number* **Example:** `Device(config)# ip nat service sip tcp port 8000`	Specifies a port number other than the default port.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	**debug ip nat sip** **Example:** `Device# debug ip nat sip`	Displays SIP messages that NAT recognizes and the embedded IP addresses contained in those messages.

Configuration Examples for NAT TCP SIP ALG Support

Example: Specifying a Port for NAT TCP SIP ALG Support

The following example shows how to configure the nonstandard port 8000:

```
Device(config)# ip nat service sip tcp port 8000
```

The following is sample output from the **debug ip nat sip** command:

```
Device# debug ip nat sip

May 23 14:11:17.243 IST: NAT-L4F:setting ALG_NEEDED flag in subblock for SIP message
May 23 14:11:17.243 IST: NAT-ALG: lookup=0 l7_bytes_recd=509 appl_type=7
May 23 14:11:17.243 IST: NAT-ALG: Complete SIP Message header of size: 376

May 23 14:11:17.243 IST: NAT-ALG: Message body length: 133
May 23 14:11:17.243 IST: NAT-ALG: Total SIP message length: 509
May 23 14:11:17.243 IST: NAT-ALG: after state machine:
May 23 14:11:17.243 IST: NAT-ALG: l7_bytes_recd=509
May 23 14:11:17.243 IST: NAT-ALG: remaining_hdr_sz=0
May 23 14:11:17.243 IST: NAT-ALG: remaining_payl_sz=0
May 23 14:11:17.243 IST: NAT-ALG: tcp_alg_state=0
May 23 14:11:17.243 IST: NAT-ALG: complete_msg_len=509
May 23 14:11:17.243 IST: NAT-SIP-TCP: Number of SIP messages received: 1
May 23 14:11:17.243 IST: NAT: SIP: [0] processing INVITE message
May 23 14:11:17.243 IST: NAT: SIP: [0] register:0 door_created:0
May 23 14:11:17.243 IST: NAT: SIP: [0] translated embedded address 192.168.122.3->10.1.1.1
May 23 14:11:17.243 IST: NAT: SIP: [0] register:0 door_created:0
May 23 14:11:17.243 IST: NAT: SIP: [0] translated embedded address 192.168.122.3->10.1.1.1
May 23 14:11:17.243 IST: NAT: SIP: [0] register:0 door_created:0
May 23 14:11:17.243 IST: NAT: SIP: [0] register:0 door_created:0
May 23 14:11:17.243 IST: NAT: SIP: Contact header found
May 23 14:11:17.243 IST: NAT: SIP: Trying to find expires parameter
May 23 14:11:17.243 IST: NAT: SIP: [0] translated embedded address 192.168.122.3->10.1.1.1
May 23 14:11:17.243 IST: NAT: SIP: [0] register:0 door_created:0
May 23 14:11:17.243 IST: NAT: SIP: [0] message body found
May 23 14:11:17.243 IST: NAT: SIP: Media Lines present:1
May 23 14:11:17.243 IST: NAT: SIP: Translated global m=(192.168.122.3, 6000) -> (10.1.1.1,
 6000)
May 23 14:11:17.243 IST: NAT: SIP: old_sdp_len:133 new_sdp_len :130
May 23 14:11:17.243 IST:  l4f_send returns 497 bytes
May 23 14:11:17.243 IST:  Complete buffer written to proxy
```

Additional Reference for NAT TCP SIP ALG Support

Related Documents

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

Standards and RFCs

Standard/RFC	Title
RFC 2543	*SIP: Session Initiation Protocol*

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

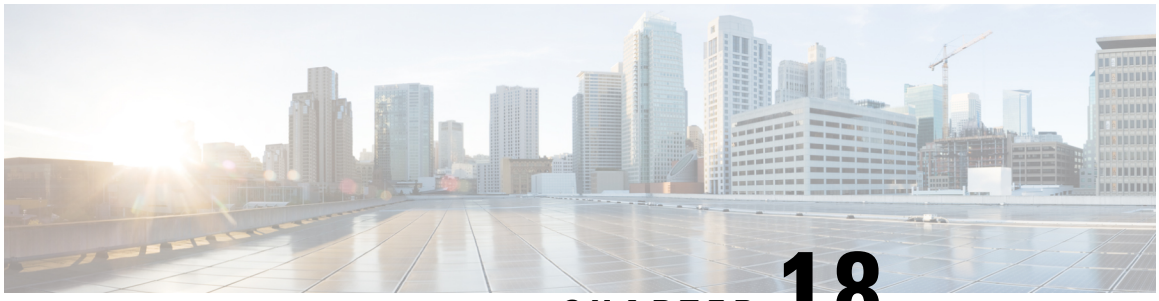
Feature Information for NAT TCP SIP ALG Support

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 19: Feature Information for NAT TCP SIP ALG Support

Feature Name	Releases	Feature Information
NAT TCP SIP ALG Support	15.3(1)T	The NAT TCP SIP ALG Support feature allows embedded messages of the Session Initiation Protocol (SIP) passing through a device that is configured with Network Address Translation (NAT) to be translated and encoded back to the packet. An application-layer gateway (ALG) is used with NAT to translate the SIP or Session Description Protocol (SDP) messages.



CHAPTER 18

NAT Routemaps Outside-to-Inside Support

The NAT Routemaps Outside-to-Inside Support feature enables you to configure a NAT routemap configuration that allows IP sessions to be initiated from outside the network to inside the network.

This module explains how to configure the NAT Routemaps Outside-to-Inside Support feature.

Restrictions for NAT Route Maps Outside-to-Inside Support

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

Information About NAT Route Maps Outside-to-inside Support

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

How to Enable NAT Route Maps Outside-to-inside Support

Enabling NAT Route Maps Outside-to-Inside Support

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device(config)# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Device(config)# ip nat pool POOL-A 192.168.201.4 192.168.201.6 netmask 255.255.255.128`	Defines a pool of network addresses for NAT.
Step 4	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* **Example:** `Device(config)# ip nat pool POOL-B 192.168.201.7 192.168.201.9 netmask 255.255.255.128`	Defines a pool of network addresses for NAT.
Step 5	**ip nat inside source route-map** *name* **pool** *name* [**reversible**] **Example:** `Device(config)# ip nat inside source route-map MAP-A pool POOL-A reversible`	Enables outside-to-inside initiated sessions to use route maps for destination-based NAT.

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

Configuration Examples for NAT Route Maps Outside-to-inside Support

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

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

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

Additional References for NAT Route Maps Outside-to-Inside Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NAT commands	Cisco IOS <<Technology>> Command Reference

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 NAT Route Maps Outside-to-Inside Support

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 20: Feature Information for NAT Route Maps Outside-to-Inside Support

Feature Name	Releases	Feature Information
NAT Route Maps Outside-to-Inside Support	12.3(14)T	The NAT Route Maps Outside-to-Inside Support feature enables you to configure a NAT route map configuration that allows IP sessions to be initiated from the outside to the inside. The following command was introduced or modified: **ip nat inside**.