IP Addressing Services Configuration Guide, Cisco Catalyst IE9300 Rugged Series Switches

First Published: 2022-04-26

Last Modified: 2023-07-14

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

Full Cisco Trademarks with Software License

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions, and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

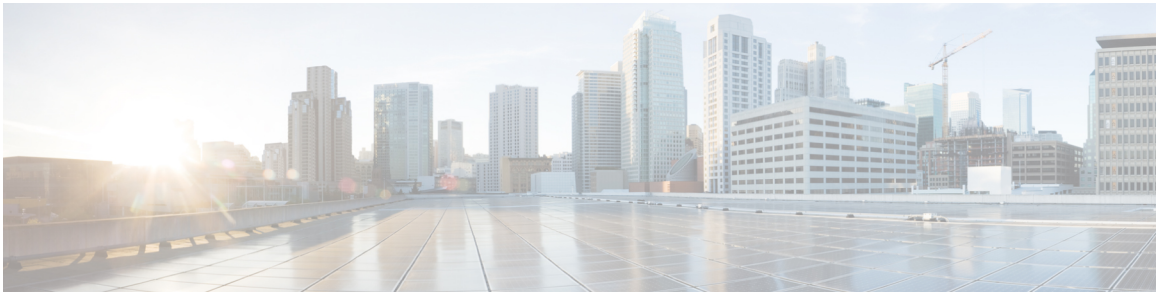
Cisco Bug Search Tool (BST) is a gateway to the Cisco bug-tracking system, which maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. The BST provides you with detailed defect information about your products and software.

Documentation Feedback

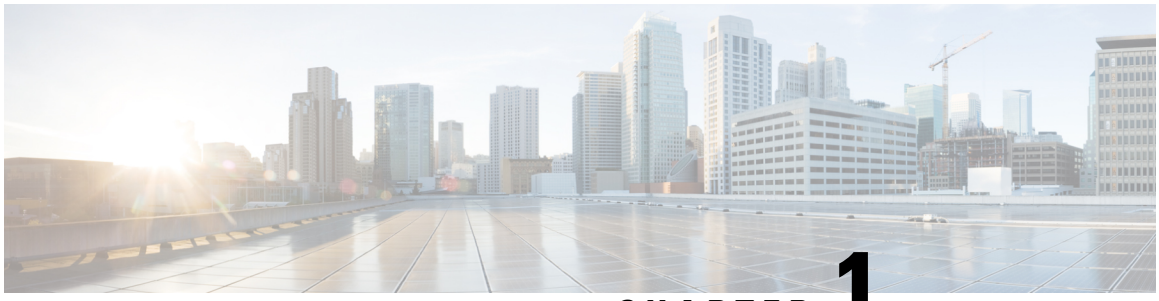
To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

Bias Free Language

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.



CONTENTS



CHAPTER 1

Layer 2 Network Address Translation

Layer 2 Network Address Translation

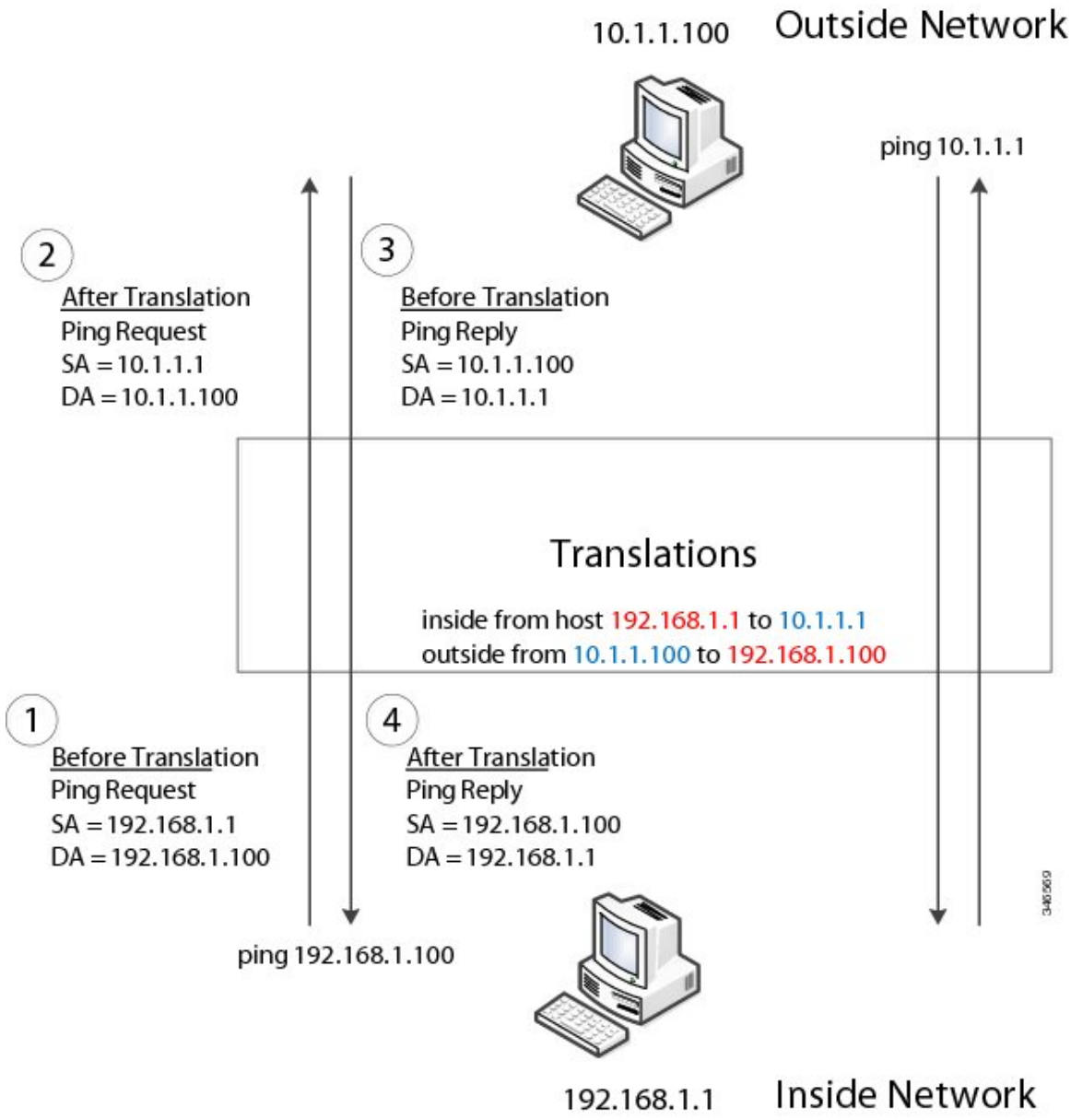
One-to-one Layer 2 NAT (Network Address Translation) is a service that allows the assignment of a unique public IP address to an existing private IP address (end device). The assignment enables the end device to communicate on both the private and public subnets. This service is configured in a NAT-enabled device and is the public "alias" of the IP address that is physically programmed on the end device. This is typically represented by a table in the NAT device.

Layer 2 NAT uses a table to translate IPv4 addresses both public-to-private, and private-to-public at line rate. Layer 2 NAT is a hardware-based implementation that provides the same high level of (bump-on-the-wire) wire-speed performance. This implementation also supports multiple VLANs through the NAT boundary for enhanced network segmentation.

In the following example, Layer 2 NAT translates addresses between sensors on a 192.168.1.x network and a line controller on a 10.1.1.x network.

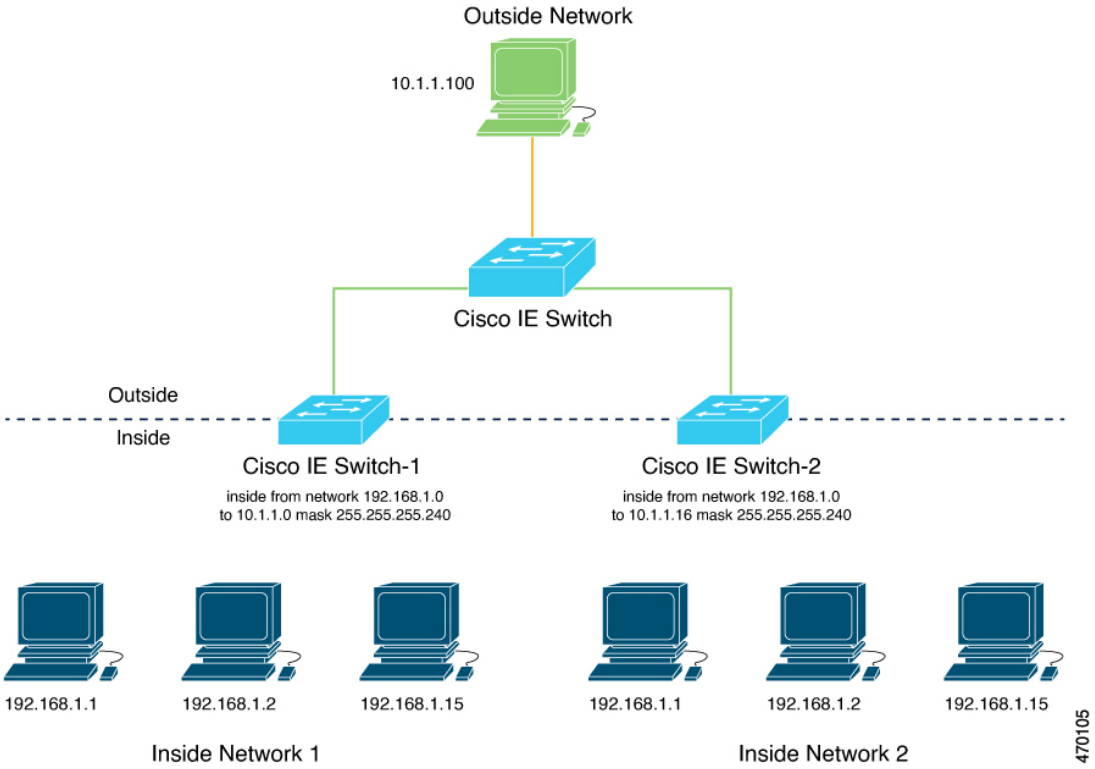
1. The 192.168.1.x network is the inside/internal IP address space and the 10.1.1.x network is the outside or external IP address space.
2. The sensor at 192.168.1.1 sends a ping request to the line controller by using an "inside" address, 192.168.1.100.
3. Before the packet leaves the internal network, Layer 2 NAT translates the source address (SA) to 10.1.1.1 and the destination address (DA) to 10.1.1.100.
4. The line controller sends a ping reply to 10.1.1.1.
5. When the packet is received on the internal network, Layer 2 NAT translates the source address to 192.168.1.100 and the destination address to 192.168.1.1.

Figure 1: Translating Addresses Between Networks



For large numbers of nodes, you can quickly enable translations for all devices in a subnet. In the scenario shown in the following figure, addresses from Inside Network 1 can be translated to outside addresses in the 10.1.1.0/28 subnet, and addresses from Inside Network 2 can be translated to outside addresses in the 10.1.1.16/28 subnet. All addresses in each subnet can be translated with one command. The benefit of using subnet-based translations saves in Layer L2 NAT rules. The switch has limits on the number of Layer 2 NAT rules. A rule with a subnet allows for multiple end devices to be translated with a single rule.

Figure 2: Inside-Outside Address Translation



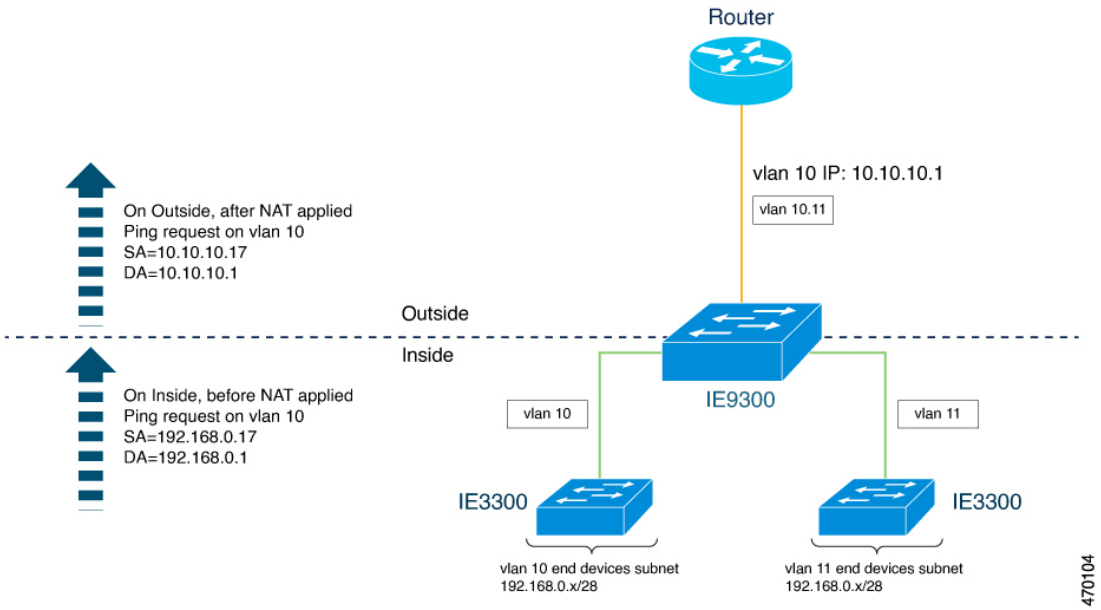
The following figure shows a Cisco Catalyst IE9300 Rugged Series Switch at the aggregation layer forwarding Ethernet packets based on Layer 2 MAC Addresses. In this example, the router is the Layer 3 gateway for all subnets and VLANs.

The L2NAT instance definitions use the **network** command to define a translated row for multiple devices in the same subnet. In this the case, it's a /28 subnet with last byte in the IP address starting with 16 and ending with 31. The gateway for the VLAN is the router with last byte of the IP address ending with .1. An outside host translation is provided for the router. The **network** command in the Layer 2NAT definition translates a subnet's worth of host with a single command, saving on Layer 2 NAT translation records.

The Gi1/0/25 uplink interface has Layer 2NAT translation instances for vlan10 and vlan 11 subnets. Interfaces can support multiple Layer 2 NAT instance definitions.

The downstream Cisco Catalyst IE3300 Rugged Series Switches are examples of access layer switches which do not perform L2NAT and rely on the upstream aggregation layer switch to do it.

Figure 3: NAT on the Cisco Catalyst IE9300 Rugged Series Switch



The following example shows the NAT configuration for the preceding diagram:

```
!
l2nat instance Subnet10-NAT
 instance-id 1
 permit all
 fixup all
 outside from host 10.10.10.1 to 192.168.0.1
 inside from network 192.168.0.0 to 10.10.10.16 mask 255.255.255.240
!
l2nat instance Subnet11-NAT
 instance-id 1
 permit all
 fixup all
 outside from host 10.10.11.1 to 192.168.0.1
 inside from network 192.168.0.0 to 10.10.11.16 mask 255.255.255.240
!
interface GigabitEthernet1/0/25
 switchport mode trunk
 l2nat Subnet10-NAT 10
 l2nat Subnet11-NAT 11
!
Interface vlan 1
  ip address 10.10.1.2
```

Guidelines and Limitations

The following list provides guidelines and limitations for using Layer 2 NAT with Cisco Catalyst IE9300 Rugged Series Switches.

Note For scale information, see the section NAT Performance and Scalability, on page 6 in this guide.

- Layer 2 NAT is supported for Cisco Catalyst IE9300 Rugged Series Switches in Cisco IOS XE Dublin 17.10.1 and later releases.
- Layer 2 NAT is supported for Cisco Catalyst IE9300 Rugged Series Switches only on stacked switches.
- Layer 2 NAT is disabled by default; it becomes enabled when you configure it. See Configure Layer 2 NAT, on page 6 in this guide.
- Layer 2 NAT applies only to unicast traffic. Untranslated unicast traffic, multicast traffic, and IGMP traffic are permitted.
- Layer 2 NAT is supported only on the uplink ports (25-28) and available in both Network Essentials and Network Advantage licenses.
- Layer 2 NAT supports one-to-one mapping between external and internal IP addresses.
- Layer 2 NAT can be applied to uplink interfaces in access or trunk mode.
- Only IPv4 addresses for Layer 2 traffic can be translated.
- Supported subnet masks on inside network translation are /24, /25, /26, /27, /28, and /32 only.
- Outside translation rule supports only host translations.
- ARP does not work transparently across Layer 2 NAT; however, the switch changes the IP addresses embedded in the payload of IP packets for the protocols to work. Embedded IP addresses are not translated.
- Statistics for debugging include the following statistics: entries for each translation, translated total ingress and egress for each instance, and for each interface. Also included are ARP fixup stats and the number of translations entries allocated in hardware.
- Layer 2 NAT does not support one-to-many and many-to-one IP address mapping.
- Layer 2 NAT cannot save on public IP addresses because public-to-private is a 1:1 translation. It is not 1:N NAT.
- If you configure a translation for a Layer 2 NAT host, do not configure it as a DHCP client.
- The management interface is behind the Layer 2 NAT function. Therefore this interface should not be on the private network VLAN. If it is on the private network VLAN, assign an inside address and configure an inside translation.
- Because Layer 2 NAT is designed to separate outside and inside addresses, we recommend that you do not configure addresses of the same subnet as both outside and inside addresses.
- Cisco Catalyst IE9300 Rugged Series Switch uplinks that support NAT instance configurations are Gig1/0/25 to Gig1/0/28.
- Layer 2 NAT is only for Layer 2 traffic; do not use it for packets undergoing routing
- Layer 2 NAT does not translate packets destined for CPU and packets coming from CPU. Management traffic should be on a different VLAN from the private network VLAN.

NAT Performance and Scalability

Layer 2 NAT translation and forwarding are performed in the hardware at line rate. The number of Layer 2NAT rules that are supported depends on the number of hardware entries that can be supported in hardware.

Scale depends on the number of inside/outside combinations. The following list provides scale examples.

- An instance with only inside rules can have a total of 128 translation rules.
- Multiple instances with one inside rule can have a total of 128 such instances applied to 128 different VLANs.
- Multiple instances with one inside rule and one outside rule can have a maximum of 64 instances.
- A single instance with one outside rule can have a maximum of 100 inside rules. The number of inside rules that can be supported reduces with increase in the outside rules.

Note We recommended that you use network translation rules to save on the number of rules.

Configure Layer 2 NAT

You must configure Layer 2 NAT instances that specify the address translations. Attach Layer 2 NAT instances to physical Ethernet interfaces, and configure which VLAN or VLANs the instances will be applied to. Layer 2 NAT instances can be configured from management interfaces (CLI/SNMP). You can view detailed statistics about the packets that are sent and received. See the section Verify the Configuration, on page 7 in this guide.

To configure Layer 2 NAT, follow these steps. Refer to the examples in Basic Inside-to-Outside Communications: Example, on page 8 and Duplicate IP Addresses Example, on page 11 in this guide for more details.

Step 1 Enter global configuration mode:

configure terminal

Step 2 Create a new Layer 2 NAT instance:

l2nat instance *instance_name* After creating an instance, you use this same command to enter the submode for that instance.

Step 3 Translate an inside address to an outside address:

inside from [*host* | *range* | *network*] *original ip* to *translated ip* [*mask*] *number* | *mask*

You can translate a single host address, a range of host addresses, or all the addresses in a subnet. Translate the source address for outbound traffic and the destination address for inbound traffic.

Step 4 Translate an outside address to an inside address:

outside from [*host* | *range* | *network*] *original ip* to *translated ip* [*mask*] *number* | *mask*

You can translate a single host address, a range of host addresses, or the addresses in a subnet. Translate the destination address for outbound traffic and the source address for inbound traffic.

Step 5 Exit config-l2nat mode:

exit

Step 6 Access interface configuration mode for the specified interface (uplink ports only on the IE 3400):

interface *interface-id*

Step 7 Apply the specified Layer 2 NAT instance to a VLAN or VLAN range. If this parameter is missing, the Layer 2 NAT instance applies to the native VLAN.

l2nat *instance_name* [*vlan* | *vlan_range*]

Step 8 Exit interface configuration mode:

end

Verify the Configuration

Perform the following commands to verify the Layer 2 NAT configuration.

Command	Purpose
show l2nat instance	Displays the configuration details for a specified Layer 2 NAT instance.
show l2nat interface	Displays the configuration details for Layer 2 NAT instances on one or more interfaces.
show l2nat statistics	Displays the Layer 2 NAT statistics for all interfaces.
show l2nat statistics interface	Displays the Layer 2 NAT statistics for a specified interface.
debug l2nat	Enables showing real-time Layer 2 NAT configuration details when the configuration is applied.
show platform hardware fed switch 1 fwd-asic resource tcam table pbr record 0 format 0 -	Displays the hardware entries.
-show platform hardware fed switch active fwd-asic resource tcam utilization | in PBR	Displays the hardware resource utilization.

The following is an example of output of the **show l2nat instance** and the **show l2nat statistics** commands:

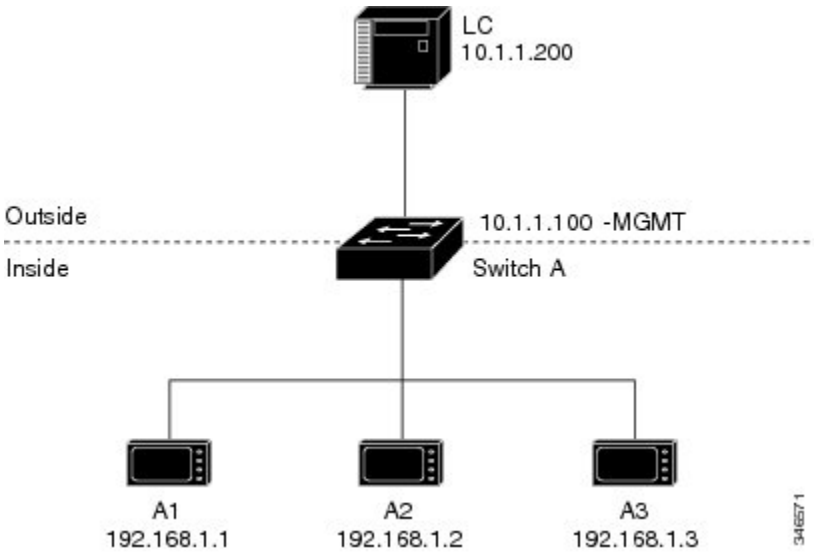
```
switch#show l2nat instance
l2nat instance test
```

```
fixup  : all
outside from host    10.10.10.200 to 192.168.1.200
inside  from host    192.168.1.1 to 10.10.10.1
l2nat instance test2
fixup  : all
inside  from host    1.1.1.1 to 2.2.2.2
outside from host    2.2.2.200 to 1.1.1.200

Switch#show l2nat interface
FOLLOWING INSTANCE(S) AND VLAN(s) ATTACHED TO ALL INTERFACES
=============================================================
l2nat Gi1/0/27 test
=============================================================

Switch#show l2nat statistics

STATS FOR INSTANCE: test (IN PACKETS)

TRANSLATED STATS (IN PACKETS)
================================================================================
INTERFACE DIRECTION VLAN   TRANSLATED
Gi1/0/27     EGRESS    50    0
Gi1/0/27     INGRESS   50    0
--------------------------------------------------------------------------------

PROTOCOL FIXUP STATS (IN PACKETS)
================================================================================
INTERFACE DIRECTION VLAN    ARP
Gi1/0/27     REPLY    50    0
Gi1/0/27     REQUEST   50    0
--------------------------------------------------------------------------------

PER TRANSLATION STATS (IN PACKETS)
================================================================================
TYPE    DIRECTION SA/DA ORIGINAL IP     TRANSLATED IP   COUNT
OUTSIDE INGRESS   SA    10.10.10.200    192.168.1.200   0
OUTSIDE EGRESS    DA    192.168.1.200   10.10.10.200    0
INSIDE  EGRESS    SA    192.168.1.1     10.10.10.1      0
INSIDE  INGRESS   DA    10.10.10.1      192.168.1.1     0
--------------------------------------------------------------------------------
================================================================================

TOTAL TRANSLATIONS ENTRIES IN HARDWARE: 4
TOTAL INSTANCES ATTACHED : 1
================================================================================
GLOBAL NAT STATISTICS
================================================================================
Total Number of TRANSLATED NAT  Packets   = 0
Total Number of ARP      FIX UP Packets   = 0
================================================================================
ad
```

Basic Inside-to-Outside Communications: Example

In this example, A1 must communicate with a logic controller (LC) that is directly connected to the uplink port. A Layer 2 NAT instance is configured to provide an address for A1 on the outside network (10.1.1.1) and an address for the LC on the inside network (192.168.1.250).

Figure 4: Basic Inside-to-Outside Communications



Now this communication can occur:

1. A1 sends an ARP request: SA: 192.168.1.1DA: 192.168.1.250.
2. Cisco Switch A fixes up the ARP request:SA:10.1.1.1DA: 10.1.1.200.
3. LC receives the request and learns the MAC Address of 10.1.1.1.
4. LC sends a response:SA: 10.1.1.200DA: 10.1.1.1.
5. Cisco Switch A fixes up the ARP response:SA: 192.168.1.250DA: 192.168.1.1.
6. A1 learns the MAC address for 192.168.1.250, and communication starts.

Note

- The management interface of the switch must be on a different VLAN from the inside network 192.168.1.x.
- See the section Basic Inside-to-Outside Communications: Configuration, on page 9 for the tasks to configure the example in this section.

Basic Inside-to-Outside Communications: Configuration

This section contains the steps to configure inside-to-outside communications as described in the preceding section. You create the Layer 2 NAT instance, add two translation entries, and then apply the instance to the interface. ARP fixups are enabled by default.

Before you begin

Read and understand the content in the section Basic Inside-to-Outside Communications: Example, on page 8.

Step 1 Enter configuration mode.

Example:

```
switch# configure
```

Step 2 Create a new Layer 2 NAT instance called A-LC.

Example:

```
switch(config)# l2nat instance A-LC
```

Step 3 Translate A1's inside address to an outside address.

Example:

```
switch(config-l2nat)# inside from host 192.168.1.1 to 10.1.1.1
```

Step 4 Translate A2's inside address to an outside address.

Example:

```
switch(config-l2nat)# inside from host 192.168.1.2 to 10.1.1.2
```

Step 5 Translate A3's inside address to an outside address.

Example:

```
switch(config-l2nat)# inside from host 192.168.1.3 to 10.1.1.3
```

Step 6 Translate the LC outside address to an inside address.

Example:

```
switch(config-l2nat)# outside from host 10.1.1.200 to 192.168.1.250
```

Step 7 Exit config-l2nat mode.

Example:

```
switch(config-l2nat)# exit
```

Step 8 Access interface configuration mode for the uplink port.

Example:

```
witch(config)# interface Gi1/1
```

Step 9 Apply this Layer 2 NAT instance to the native VLAN on this interface.

Example:

```
switch(config-if)# l2nat A-LC
```

Note For tagged traffic on a trunk, add the VLAN number when attaching the instance to an interface as follows:

l2nat *instance vlan*

Step 10 Return to privileged EXEC mode.

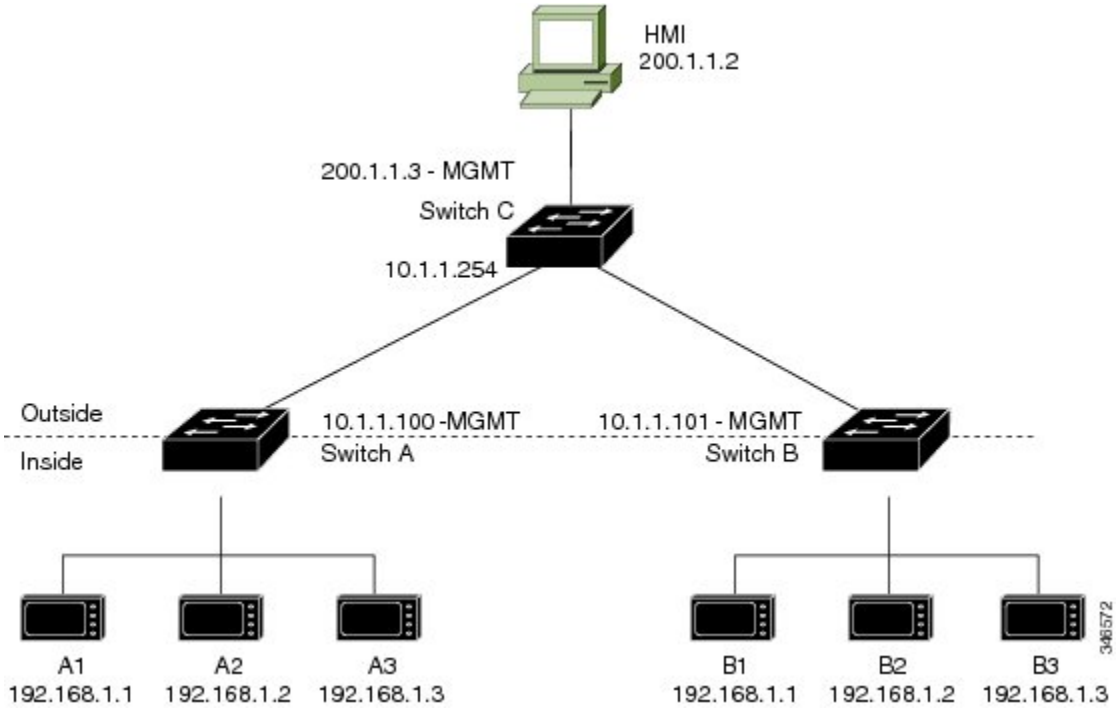
Example:

```
switch# end
```

Duplicate IP Addresses Example

In this scenario, two machine nodes are preconfigured with addresses in the 192.168.1.x space. Layer 2 NAT translates these addresses to unique addresses on separate subnets of the outside network. In addition, for machine-to-machine communications, the Node A machines need unique addresses on the Node B space and the Node B machines need unique addresses in the Node A space.

Figure 5: Duplicate IP Addresses



- Switch C needs an address in the 192.168.1.x space. When packets come into Node A or Node B, the 10.1.1.254 address of Switch C is translated to 192.168.1.254. When packets leave Node A or Node B, the 192.168.1.254 address of Switch C is translated to 10.1.1.254.
- Node A and Node B machines need unique addresses in the 10.1.1.x space. For quick configuration and ease of use, the 10.1.1.x space is divided into subnets: 10.1.1.0, 10.1.1.16, 10.1.1.32, and so on. Each subnet can then be used for a different node. In this example, 10.1.1.16 is used for Node A, and 10.1.1.32 is used for Node B.
- Node A and Node B machines need unique addresses to exchange data. The available addresses are divided into subnets. For convenience, the 10.1.1.16 subnet addresses for the Node A machines are translated to 192.168.1.16 subnet addresses on Node B. The10.1.1.32 subnet addresses for the Node B machines are translated to 192.168.1.32 addresses on Node A.
- Machines have unique addresses on each network:

Table 1: Translated IP Addresses

Node	Address in Node A	Address in Outside Network	Address in Node B
Switch A network address	192.168.1.0	10.1.1.16	192.168.1.16
A1	192.168.1.1	10.1.1.17	192.168.1.17
A2	192.168.1.2	10.1.1.18	192.168.1.18
A3	192.168.1.3	10.1.1.19	192.168.1.19
Cisco Switch B network address	192.168.1.32	10.1.1.32	192.168.1.0
B1	192.168.1.33	10.1.1.33	192.168.1.1
B2	192.168.1.34	10.1.1.34	192.168.1.2
B3	192.168.1.35	10.1.1.35	192.168.1.3
Switch C	192.168.1.254	10.1.1.254	192.168.1.254

Duplicate IP Addresses Configuration: Switch A

This section provides the steps for configuring Layer 2 NAT to translate the duplicated IP address of one machine node in an inside network to a unique address on a subnet of an outside network. This procedure is for Switch A in the section Duplicate IP Addresses Example, on page 11.

Before you begin

Read and understand the content in the section Duplicate IP Addresses Example, on page 11.

Step 1 Enter global configuration mode.

Example:

```
switch# configure
```

Step 2 Create a new Layer 2 NAT instance called A-Subnet.

Example:

```
switch(config)# l2nat instance A-Subnet
```

Step 3 Translate the Node A machines' inside addresses to addresses in the 10.1.1.16 255.255.255.240 subnet.

Example:

```
switch(config-l2nat)# inside from network 192.168.1.0 to 10.1.1.16 mask 255.255.255.240
```

Step 4 Translate the outside address of Switch C to an inside address.

Example:

```
switch(config-l2nat)# outside from host 10.1.1.254 to 192.168.1.254
```

Step 5 Translate the Node B machines' outside addresses to their inside addresses.

Example:

```
switch(config-l2nat)# outside from host 10.1.1.32 to 192.168.1.32
outside from host 10.1.1.33 to 192.168.1.33
outside from host 10.1.1.34 to 192.168.1.34
outside from host 10.1.1.35 to 192.168.1.35
```

Step 6 Exits config-l2nat mode.

Example:

```
switch(config-l2nat)# exit
```

Step 7 Access interface configuration mode for the uplink port.

Example:

```
switch(config)# interface Gi1/1
```

Step 8 Apply this Layer 2 NAT instance to the native VLAN on this interface.

Example:

```
switch(config-if)# l2nat A-Subnet
```

Note For tagged traffic on a trunk, add the VLAN number when attaching the instance to an interface as follows:

l2nat *instance vlan*

Step 9 Return to privileged EXEC mode.

Example:

```
switch# end
```

What to do next

Configure Layer 2 NAT to translate the duplicated IP address of Switch B in the section Duplicate IP Addresses Example, on page 11. See Duplicate IP Addresses Configuration: Switch B, on page 13.

Duplicate IP Addresses Configuration: Switch B

This section provides the steps for configuring Layer 2 NAT to translate the duplicated IP address of one machine node in an inside network to a unique address on a subnet of an outside network. This procedure is for Switch B in the section Duplicate IP Addresses Example, on page 11.

Before you begin

Read and understand the content in the section Duplicate IP Addresses Example, on page 11.

Step 1 Enter global configuration mode.

Example:

```
switch# configure
```

Step 2 Create a new Layer 2 NAT instance called B-Subnet.

Example:

```
switch(config)# l2nat instance B-Subnet
```

Step 3 Translate the Node B machines' inside addresses to addresses in the 10.1.1.32 255.255.255.240 subnet.

Example:

```
switch(config-l2nat)# inside from network 192.168.1.0 to 10.1.1.32 255.255.255.240
```

Step 4 Translate the outside address of Switch C to an inside address.

Example:

```
switch(config-l2nat)# outside from host 10.1.1.254 to
```

Step 5 Translate the Node A machines' outside addresses to their inside addresses.

Example:

```
switch(config-l2nat)# outside from host 10.1.1.16 to 192.168.1.16
outside from host 10.1.1.17 to 192.168.1.17
outside from host 10.1.1.18 to 192.168.1.18
outside from host 10.1.1.19 to 192.168.1.19
```

Step 6 Exit config-l2nat mode.

Example:

```
switch(config-l2nat)# exit
```

Step 7 Access interface configuration mode for the uplink port.

Example:

```
switch(config)# interface Gi1/1
```

Step 8 Apply this Layer 2 NAT instance to the native VLAN on this interface.

Example:

```
switch(config-if)# l2nat name1
```

Note For tagged traffic on a trunk, add the VLAN number when attaching the instance to an interface as follows:

l2nat *instance vlan*

Step 9 Show the configuration details for the specified Layer 2 NAT instance.

Example:

```
switch# show l2nat instance name1
```

Step 10 Show Layer 2 NAT statistics.

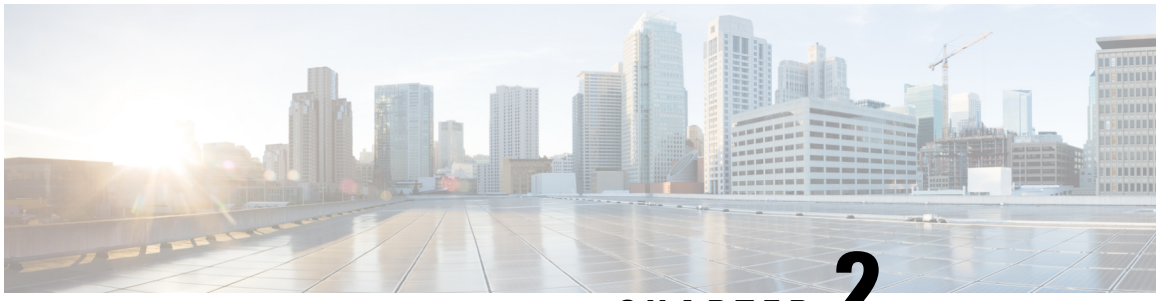
Example:

```
switch# show l2nat statistics
```

Step 11 Return to privileged EXEC mode.

Example:

```
switch# end
```



CHAPTER 2

Layer 3 Network Address Translation

Network Address Translation

Network Address Translation (NAT) is designed for IP address conservation. It enables private IP networks that use unregistered IP addresses to connect to the Internet. NAT operates on a device, usually connecting two networks together, and translates the private (not globally unique) addresses in the internal network into global routable addresses. It does so before packets are forwarded onto another network.

NAT can be configured to advertise only one address for the entire network to the outside world. This ability provides more security by effectively hiding the entire internal network behind that one address. NAT offers the dual functions of security and address conservation and is typically implemented in remote-access environments.

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

Finding Feature Information

Your software release may not support all the features described in this document. 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, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this chapter.

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

Benefits of Configuring NAT

Configuring NAT provides the following benefits:

- NAT Resolves the problem of IP depletion.

  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. In such cases, 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.

- NAT provides a layer of security by preventing the client IP address from being exposed to the outside network.

  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. 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 RFC 1918 addresses or registered addresses.

- NAT is designed for use on a variety of devices for IP address simplification and conservation. In addition, NAT allows the selection of internal hosts that are available for translation.

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

How NAT Works

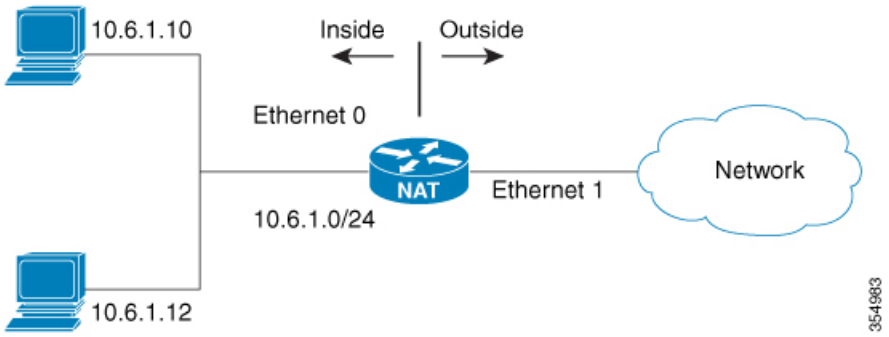
A device that is configured with NAT has at least one interface to the inside network and one to the outside network. In a typical environment, NAT is configured at the exit device between a stub domain and the backbone. When a packet leaves the domain, NAT translates the locally significant source address into a globally unique address. When a packet enters the domain, NAT translates the globally unique destination address into a local address.

Multiple inside networks could be connected to the device and similarly there might exist multiple exit points from the device towards outside networks. If NAT cannot allocate an address because it has run out of addresses,

it drops the packet and sends an Internet Control Message Protocol (ICMP) host unreachable packet to the destination.

Translation and forwarding are performed in the hardware switching plane, improving the overall throughput performance. For more details on performance, see the section Performance and Scale Numbers for NAT, on page 26.

Figure 6: NAT



Uses of NAT

You can use NAT in the following scenarios:

- To connect to the Internet when only a few of your hosts have globally unique IP address.

  NAT is configured on a device at the border of a stub domain (referred to as the inside network) and a public network such as the Internet (referred to as the outside network). NAT translates internal local addresses to globally unique IP addresses before sending packets to the outside network.

  As a solution to the connectivity problem, NAT is practical only when relatively few hosts in a stub domain communicate outside of the domain at the same time. In such cases, only a small subset of the IP addresses in the domain must be translated into globally unique IP addresses when outside communication is necessary, and these addresses can be reused.

- To renumber:

  Instead of changing the internal addresses, which can be a considerable amount of work, you can translate them by using NAT.

NAT Inside and Outside Addresses

The term *inside* in a NAT context refers to networks owned by an organization that must be translated. When NAT is configured, hosts within this network have addresses in one space (known as the local address space) that appears to those 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 generally not under the control of an organization. Hosts in outside networks can also 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 is probably not a routable IP address assigned by NIC or service provider.
- Inside global address: a global routable 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 routable IP address, it is allocated from the address space that is routable on the inside.
- Outside global address: the IP address 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.
- Inside Source Address Translation: translates an inside local address to inside global address.
- Outside Source Address Translation: translates the outside global address to outside local address.
- Static Port Translation: translates the IP address and port number of an inside/outside local address to the IP address and port number of the corresponding inside/outside global address.
- Static Translation of a given subnet: translates a specified range of subnets of an inside/outside local address to the corresponding inside/outside global address.
- Half Entry: represents a mapping between the local and global address/ports and is maintained in the translation database of NAT module. A half entry may be created statically or dynamically based on the configured NAT rule.
- Full Entry/Flow entry: represents a unique flow corresponding to a given session. In addition to the local to global mapping, it also maintains the destination information which fully qualifies the given flow. A Full entry is always created dynamically and maintained in the translation database of NAT module.

Types of NAT

You can configure NAT such that it advertises only a single address for your entire network to the outside world. The configuration 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 / PAT: Maps multiple unregistered IP addresses to a single registered IP address (many to one) using different Layer 4 ports. This method is also known as Port Address Translation (PAT). By using overloading, thousands of users can be connected to the Internet by using only one real global IP address.

Using NAT to Route Packets to the Outside Network (Inside Source Address Translation)

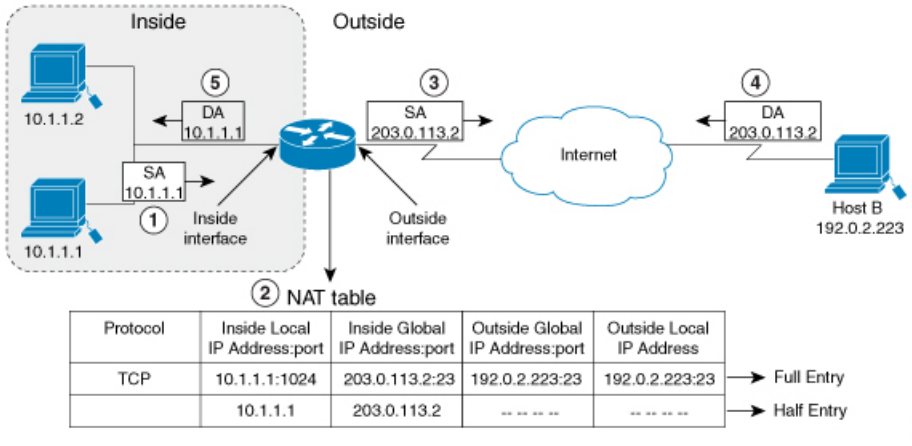
You can translate unregistered IP addresses into globally unique IP addresses when communicating outside your network.

You can configure static or dynamic inside source address translation 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. You can enable Static translation by configuring a static NAT rule as explained in the x section.
- Dynamic translation establishes a mapping between an inside local address and a pool of global addresses dynamically. Dynamic translation can be enabled by configuring a dynamic NAT rule and the mapping is established based on the result of the evaluation of the configured rule at run-time. You can employ an Access Control List (ACL), both Standard and Extended ACLs, to specify the inside local address. The inside global address can be specified through an address pool or an interface. Dynamic translation is enabled by configuring a dynamic rule as explained in the section Configuring Dynamic Translation of Inside Source Addresses, on page 29.

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

Figure 7: 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. NAT module intercepts the corresponding packet and attempts to translate the packet.

   The following scenarios are possible based on the presence or absence of a matching NAT rule:

   - If a matching static translation rule exists, the packet gets translated to the corresponding inside global address. Otherwise, the packet is matched against the dynamic translation rule, and in the event of a successful match, it gets translated to the corresponding inside global address. The NAT

module inserts a fully qualified flow entry corresponding to the translated packet, into its translation database. This facilitates fast translation and forwarding of the packets corresponding to this flow, in either direction.

- The packet gets forwarded without any address translation in the absence of a successful rule match.
- The packet is dropped in the event of failure to obtain a valid inside global address even-though we have a successful rule match.

Note If an ACL is employed for dynamic translation, NAT evaluates the ACL and ensures that only the packets that are permitted by the given ACL are considered for translation.

3. The device replaces the inside local source address of host 10.1.1.1 with the inside global address of the translation, 203.0.113.2, 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. The response packet from host B would be destined to the inside global address. The NAT module intercepts this packet and translates it back to the corresponding inside local address with the help of the flow entry that has been set up in the translation database.

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

Outside Source Address Translation

You can translate the source address of the IP packets that travel from outside of the network to inside the network. This type of translation is usually employed in conjunction with inside source address translation to interconnect overlapping networks.

This process is explained in the section Configuring Translation of Overlapping Networks, on page 34.

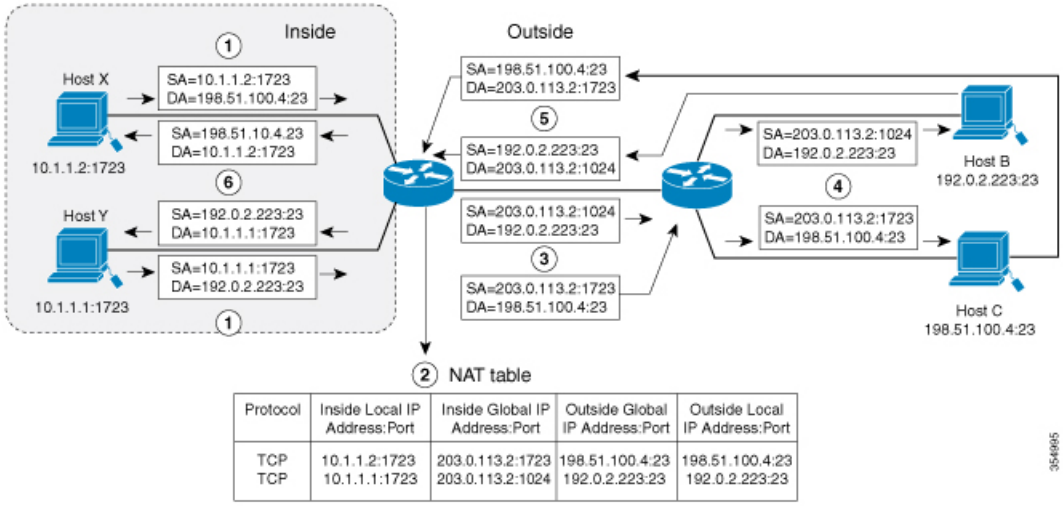
Port Address Translation

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 NAT configuration is called overloading or port address translation (PAT).

When overloading is configured, the device maintains enough information from higher-level protocols (for example, TCP or UDP port numbers) to translate the global address back to the correct local address. When multiple local addresses map to one global address, the TCP or UDP port numbers of each inside host distinguish between the local addresses.

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

Figure 8: PAT / NAT Overloading Inside Global Addresses



The device performs the following process in the overloading of inside global addresses, as shown in the figure above. Both Host B and Host C believe that they are communicating with a single host at address 203.0.113.2. However, 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 10.1.1.1:1723 opens a connection to Host B and the user at host 10.1.1.2:1723 opens a connection to Host C.
2. NAT module intercepts the corresponding packets and attempts to translate the packets.

   Based on the presence or absence of a matching NAT rule the following scenarios are possible:

   - If a matching static translation rule exists, then it takes precedence and the packets are translated to the corresponding global address. Otherwise, the packets are matched against dynamic translation rule and in the event of a successful match, they are translated to the corresponding global address. NAT module inserts a fully qualified flow entry corresponding to the translated packets, into its translation database, to facilitate fast translation and forwarding of the packets corresponding to this flow, in either direction.
   - The packets are forwarded without any address translation in the absence of a successful rule match.
   - The packets are dropped in the event of failure to obtain a valid inside global address even though we have a successful rule match.
   - Because this is a PAT configuration, transport ports help translate multiple flows to a single global address. (In addition to source address, the source port is also subjected to translation and the associated flow entry maintains the corresponding translation mappings.)

3. The device replaces inside local source address/port 10.1.1.1/1723 and 10.1.1.2/1723 with the corresponding selected global address/port 203.0.113.2/1024 and 203.0.113.2/1723 respectively and forwards the packets.
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, on port 1024. Host C receives the packet and responds to host 10.1.1.2 using the inside global IP address 203.0.113.2, on port 1723.

5. When the device receives the packets with the inside global IP address, it performs a NAT table lookup; the inside global address and port, and the outside address and port as keys; translates the addresses to the inside local addresses 10.1.1.1:1723 / 10.1.1.2:1723 and forwards the packets to host 10.1.1.1. and 10.1.1.2 respectively.

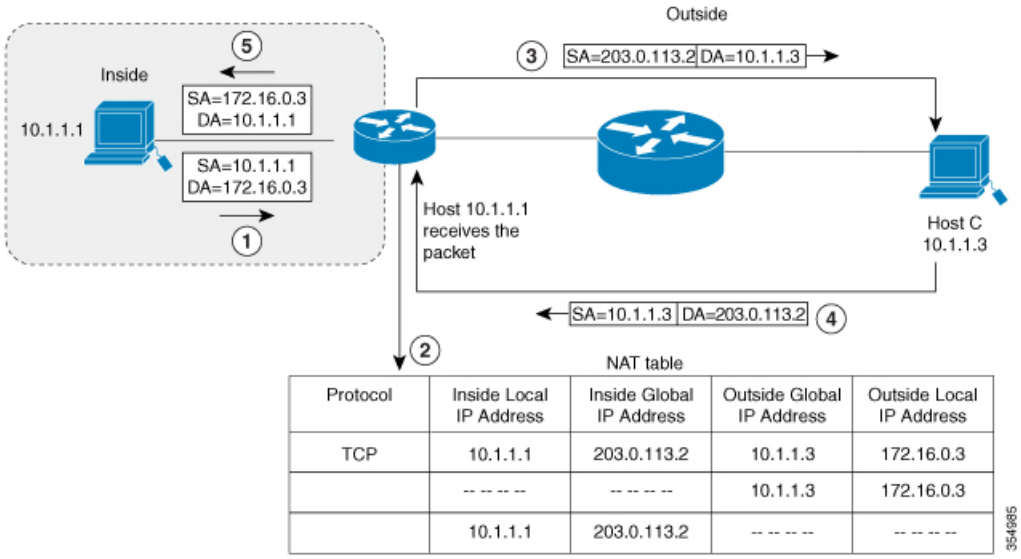
Host 10.1.1.1 and Host 10.1.1.2 receive the packet and continue the conversation. The device performs Step 2 to Step 5 for each packet it receives.

Overlapping Networks

Use 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 that is already legally owned and assigned to a different device on the Internet or outside the network.

The following figure depicts overlapping networks: the inside network and outside network both have the same local IP addresses (10.1.1.x). You need network connectivity between such overlapping address spaces with one NAT device to translate the address of a remote peer (10.1.1.3) to a different address from the perspective of the inside.

Figure 9: NAT Translating Overlapping Addresses



Notice that the inside local address (10.1.1.1) and the outside global address (10.1.1.3) are in the same subnet. To translate the overlapping address, first, the inside source address translation happens with the inside local address getting translated to 203.0.113.2 and a half entry is created in the NAT table. On the Receiving side, the outside source address is translated to 172.16.0.3 and another half entry is created. The NAT table is then updated with a full entry of the complete translation.

The following steps describe how a device translates overlapping addresses:

1. Host 10.1.1.1 opens a connection to 172.16.0.3.
2. The NAT module sets up the translation mapping of the inside local and global addresses to each other and the outside global and local addresses to each other.

3. The Source Address (SA) is replaced with inside global address and the Destination Address (DA) is replaced with outside global address.
4. Host C receives the packet and continues the conversation.
5. The device does a NAT table lookup, replaces the DA with inside local address, and replaces the SA with outside local address.
6. Host 10.1.1.1 receives the packet and the conversation continues using this translation process.

Limitations of NAT

- Some NAT operations are currently not supported in the hardware data plane. The following are such operations that are carried out in the relatively slower software data plane:
  - Translation of Internet Control Message Protocol (ICMP) packets
  - Translation of packets that require application layer gateway (ALG) processing
  - Packets that require both inside and outside translation
- The maximum number of sessions that can be translated and forwarded in the hardware in an ideal setting is limited to 192. Additional flows that require translation are handled in the software data plane at a reduced throughput.

Note Each translation consumes two entries in TCAM.

- A configured NAT rule might fail to get programmed into the hardware owing to resource constraint. This could result in packets that correspond to the given rule to get forwarded without translation.
- ALG support is currently limited to FTP, TFTP, and ICMP protocols. Also, although TCP SYN, TCP FIN and TCP RST are not part of ALG traffic, they are processed as part of ALG traffic.
- Dynamically created NAT flows age out after a period of inactivity. The number of NAT flows whose activity can be tracked is limited to 192.
- Port channel is not supported in NAT configuration.
- NAT does not support translation of fragmented packets.
- Explicit deny access control entry (ACE) in NAT ACL is not supported. Only explicit permit ACE is supported.
- NAT and PBR share the same TCAM space and they cannot co-exist.
- NAT configuration must be done without using route maps because route mapped NAT is not supported.
- NAT is not supported for multicast packets.

Performance and Scale Numbers for NAT

The maximum number of bidirectional NAT flows supported in hardware is limited to 192.

Address Only Translation

Note Using Address Only Translation optimizes the handling of flows and enhances the scale of the NAT feature.

You can use Address only Translation (AOT) functionality in situations that require only the address fields to be translated and not the transport ports. In such settings, enabling AOT functionality significantly increases the number of flows that can be translated and forwarded in the hardware at line-rate. This improvement is brought about by optimizing the usage of various hardware resources associated with translation and forwarding.

A typical NAT focused resource allocation scheme sets aside 384 TCAM entries for performing hardware translation. This places a strict upper limit on the number of flows that can be translated and forwarded at line-rate. Under AOT scheme, the usage of TCAM resource is highly optimized thereby enabling the accommodation of more number of flows in the TCAM tables and this provides a significant improvement in the hardware translation and forwarding scale.

AOT can be very effective in situations where majority of the flows are destined to a single or a small set of destinations. Under such favorable conditions, AOT can potentially enable line-rate translation and forwarding of all the flows originating from one or more given end-points. AOT functionality is disabled by default. It can be enabled using the **no ip nat create flow-entries** command. The existing dynamic flow can be cleared using the **clear ip nat translation** command. The AOT feature can be disabled using the **ip nat create flow-entries** command.

Restrictions for Address Only Translation

- AOT feature is expected to function correctly only in translation scenarios corresponding to simple inside static and inside dynamic rules. The simple static rule must be of the type **ip nat inside source static** *local-ip global-ip*, and the dynamic rule must be of the type **ip nat inside source list** *access-list* **pool** *name*.
- When AOT is enabled, the **show ip nat translation** command will not give visibility into all the NAT flows being translated and forwarded.

Configuring NAT

The tasks described in this section will help you configure NAT. Based on the desired configuration, you may need to configure more than one task.

Configuring Static Translation of Inside Source Addresses

Configure static translation of inside source address 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Use any of the following three commands depending on the requirement:
   - **ip nat inside source static** *local-ip global-ip*

     ```
     Switch(config)# ip nat inside source static 10.10.10.1 172.16.131.
     ```

   - **ip nat inside source static** *protocol local-ip port global-ip port*

     ```
     Switch(config)# ip nat inside source static tcp 10.10.10.1 1234 172.16.131.1 5467
     ```

   - **ip nat inside source static network** *local-ip global-ip* {**prefix_len** *len* | **subnet** *subnet-mask*}

     ```
     Switch(config)# ip nat inside source static network 10.10.10.1 172.16.131.1
     prefix_len 24
     ```

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:** `Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	Use any of the following three commands depending on the requirement: • **ip nat inside source static** *local-ip global-ip* `Switch(config)# ip nat inside source static 10.10.10.1 172.16.131.`	Establishes static translation between an inside local address and an inside global address. Establishes a static port translation between an inside local address and an inside global address. Establishes a static translation between an inside local address and an inside global address. You can specify a

	Command or Action	Purpose

	• **ip nat inside source static** *protocol local-ip port global-ip port*
	`Switch(config)# ip nat inside source static tcp 10.10.10.1 1234 172.16.131.1 5467`
	• **ip nat inside source static network** *local-ip global-ip* {**prefix_len** *len* | **subnet** *subnet-mask*}
	`Switch(config)# ip nat inside source static network 10.10.10.1 172.16.131.1 prefix_len 24`
		range of subnets to be translated to the inside global address, wherein the host portion of the IP address gets translated and the network portion of the IP remains the same.

Step 4	**interface** *type number* **Example:** `Switch(config)# interface GigabitEthernet 1/0/1`	Specifies an interface and enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **[secondary]** **Example:** `Switch(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:** `Switch(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 7	**exit** **Example:** `Switch(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Switch(config)# interface GigabitEthernet 1/0/2`	Specifies a different interface and enters interface configuration mode.
Step 9	**ip address** *ip-address mask* **[secondary]** **Example:** `Switch(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:** `Switch(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 11	**end** **Example:** `Switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Dynamic Translation of Inside Source Addresses

Dynamic translation establishes a mapping between an inside local address and a pool of global addresses dynamically. Dynamic translation can be enabled by configuring a dynamic NAT rule and the mapping is established based on the result of the evaluation of the configured rule at run-time. You can employ an ACL to specify the inside local address and the inside global address can be specified through an address pool or an interface.

Dynamic translation is useful when multiple users on a private network need to access the Internet. The dynamically configured pool IP address may be used as needed and is released for use by other users when access to the internet is no longer required.

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:** `Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* | **prefix-length** *prefix-length* **Example:** `Switch(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:**	Defines a standard access list permitting those addresses that are to be translated.

	Command or Action	Purpose
	`Switch(config)# access-list 1 permit 192.168.34.0 0.0.0.255`	
Step 5	**ip nat inside source list** *access-list-number* **pool** *name* **Example:** `Switch(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:** `Switch(config)# interface GigabitEthernet 1/0/1`	Specifies an interface and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **Example:** `Switch(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:** `Switch(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 9	**exit** **Example:** `Switch(config-if)#exit`	Exits the interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Switch(config)# interface GigabitEthernet 1/0/2`	Specifies an interface and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* **Example:** `Switch(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:** `Switch(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 13	**end** **Example:** `Switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring PAT

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* **[secondary]**
8. **ip nat inside**
9. **exit**
10. **interface** *type number*
11. **ip address** *ip-address mask* **[secondary]**
12. **ip nat outside**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**ip nat pool** *name start-ip end-ip* **netmask** *netmask* | **prefix-length** *prefix-length* **Example:** `Switch(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:** `Switch(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:** `Switch(config)# ip nat inside source list 1 pool net-208 overload`	Establishes dynamic source translation with overloading, specifying the access list defined in Step 4.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Switch(config)# interface GigabitEthernet 1/0/1`	Specifies an interface and enters interface configuration mode.
Step 7	**ip address** *ip-address mask* **[secondary]** **Example:** `Switch(config-if)# ip address 192.168.201.1 255.255.255.240`	Sets a primary IP address for an interface.
Step 8	**ip nat inside** **Example:** `Switch(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 9	**exit** **Example:** `Switch(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 10	**interface** *type number* **Example:** `Switch(config)# interface GigabitEthernet 1/0/2`	Specifies a different interface and enters interface configuration mode.
Step 11	**ip address** *ip-address mask* **[secondary]** **Example:** `Switch(config-if)#  ip address 192.168.201.29 255.255.255.240`	Sets a primary IP address for an interface.
Step 12	**ip nat outside** **Example:** `Switch(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 13	**end** **Example:** `Switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring NAT of External IP Addresses Only

By default, NAT translates the addresses embedded in the packet pay-load as explained in the section Using Application-Level Gateways with NAT, on page 38. There might be situations where the translation of the embedded address is not desirable and in such cases, NAT can be configured to translate the external IP address only.

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.

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

Configuring Translation of Overlapping Networks

Configure static translation of overlapping networks if your IP addresses in the stub network are legitimate IP addresses belonging to another network and you want to communicate with those hosts or routers using static translation.

Note For a successful NAT outside translation, the device should be configured with a route for the outside local address. You can configure the route either manually or using the **add-route** option associated with **ip nat outside source** {**static** | **list**} command. We recommend that you use the **add-route** option to enable automatic creation of the route.

SUMMARY STEPS

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

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	
Step 3	**ip nat inside source static** *local-ip global-ip* **Example:** `Switch(config)# ip nat inside source static 10.1.1.1 203.0.113.2`	Establishes static translation between an inside local address and an inside global address.
Step 4	**ip nat outside source static** *local-ip global-ip* **Example:** `Switch(config)# ip nat outside source static 172.16.0.3 10.1.1.3`	Establishes static translation between an outside local address and an outside global address.
Step 5	**interface** *type number* **Example:** `Switch(config)# interface GigabitEthernet 1/0/1`	Specifies an interface and enters interface configuration mode.
Step 6	**ip address** *ip-address mask* **Example:** `Switch(config-if)# ip address 10.114.11.39 255.255.255.0`	Sets a primary IP address for an interface.
Step 7	**ip nat inside** **Example:** `Switch(config-if)# ip nat inside`	Marks the interface as connected to the inside.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Switch(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**interface** *type number* **Example:** `Switch(config)# interface GigabitEthernet 1/0/2`	Specifies a different interface and enters interface configuration mode.
Step 10	**ip address** *ip-address mask* **Example:** `Switch(config-if)#  ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 11	**ip nat outside** **Example:** `Switch(config-if)# ip nat outside`	Marks the interface as connected to the outside.
Step 12	**end** **Example:** `Switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Address Translation Timeouts

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

By default, dynamically created translation entries time-out after a period of inactivity to enable the efficient use of various resources. You can change the default values on timeouts, if necessary. The following are the default time-out configurations associated with major translation types:

- Established TCP sessions: 24 hours
- UDP flow: 5 minutes
- ICMP flow: 1 minute

The default timeout values are adequate to address the timeout requirements in most of the deployment scenarios. However, these values can be adjusted/fine-tuned as appropriate. It is recommended not to configure very small timeout values (less than 60 seconds) as it could result in high CPU usage. Refer the x section for more information.

Based on your configuration, you can change the timeouts described in this section.

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**ip nat translation** *seconds* **Example:** `Switch(config)# ip nat translation 300`	(Optional) Changes the amount of time after which NAT translations time out. The default timeout is 24 hours, and it applies to the aging time for half-entries.
Step 4	**ip nat translation udp-timeout** *seconds* **Example:** `Switch(config)# ip nat translation udp-timeout 300`	(Optional) Changes the UDP timeout value.
Step 5	**ip nat translation tcp-timeout** *seconds* **Example:** `Switch(config)# ip nat translation tcp-timeout 2500`	(Optional) Changes the TCP timeout value. The default is 24 hours.
Step 6	**ip nat translation finrst-timeout** *seconds* **Example:** `Switch(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 7	**ip nat translation icmp-timeout** *seconds* **Example:** `Switch(config)# ip nat translation icmp-timeout 45`	(Optional) Changes the ICMP timeout value.
Step 8	**ip nat translation syn-timeout** *seconds* **Example:**	(Optional) Changes the synchronous (SYN) timeout value.

	Command or Action	Purpose
	`Switch(config)# ip nat translation syn-timeout 45`	The synchronous timeout or the aging time is used only when a SYN request is received on a TCP session. When a synchronous acknowledgment (SYNACK) request is received, the timeout changes to TCP timeout.
Step 9	**end** **Example:** `Switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Using Application-Level Gateways with NAT

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 the following:

- HTTP
- TFTP
- Telnet
- Archie
- Finger
- Network Time Protocol (NTP)
- Network File System (NFS)
- Remote login (rlogin)
- Remote shell (rsh)
- Remote copy (rcp)

NAT Application-Level Gateway (ALG) enables certain applications that carry address/port information in their payloads to function correctly across NAT domains. In addition to the usual translation of address/ports in the packet headers, ALGs take care of translating the address/ports present in the payload and setting up temporary mappings.

Best Practices for NAT Configuration

- In cases where both static and dynamic rules are configured, ensure that the local addresses specified in the rules do not overlap. If such an overlap is possible, then the ACL associated with the dynamic rule should exclude the corresponding addresses used by the static rule. Similarly, there must not be any overlap between the global addresses as this could lead to undesired behavior.
- Do not employ loose filtering such as **permit ip any any** in an ACL associated with NAT rule as this could result in unwanted packets being translated.

- Do not share an address pool across multiple NAT rules.
- Do not define the same inside global address in Static NAT and Dynamic Pool. This action can lead to undesirable results.
- Exercise caution while modifying the default timeout values associated with NAT. Small timeout values could result in high CPU usage.
- Exercise caution while manually clearing the translation entries as this could result in the disruption of application sessions.
- ALG packets traversing a NAT enabled interface will get punted to CPU, regardless of the packets being translated or not. Therefore, it is recommended to use dedicated interface(s) just for NAT traffic. For all other types of traffic that does not require NAT translation, use a different interface(s).

Troubleshooting NAT

This section explains the basic steps to troubleshoot and verify NAT.

- Clearly define what NAT is supposed to achieve.
- Verify that correct translation table exists using the **show ip nat translation** command.
- Verify that timer values are correctly configured using the **show ip nat translation verbose** command.
- Check the ACL values for NAT using the **show ip access-list** command.
- Check the overall NAT configuration using the **show ip nat statistics** command.
- Use the **clear ip nat translation** command to clear the NAT translational table entires before the timer expires.
- Use debug nat ip and debug nat ip detailed commands to debug NAT configuration.

For further information on Troubleshooting NAT, see Verifying NAT Operation and Basic NAT Troubleshooting on Cisco.com..

Feature History for Network Address Translation

This table provides release and related information for the features explained in this module.

These features are available in all the releases later to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Description
Cisco IOS XE Cupertino 17.7.1	Layer 3 Network Address Translation for Cisco Catalyst IE9300 Rugged Series Switches	NAT enables private IP networks that use unregistered IP address to connect to the internet. NAT operates on a device, usually connecting two networks together, and translates the private addresses in the internal network into global routable addresses, before packets are forwarded onto another network. Support for this feature was introduced for the following switch models: • IE-9310-26S2C-A • IE-9320-26S2C-A