Network Address Translation

Overview of NAT

Network Address Translation (NAT) is an IP address management feature that enables private IP networks using unregistered IP addresses to communicate with external networks, such as the Internet. NAT operates on a device—typically connecting two networks—and translates private (non-globally unique) addresses from the internal network into globally routable addresses before forwarding packets to the external network.

Cisco switches support NAT in both standalone and stack configurations. NAT is commonly implemented at the network edge in enterprise and remote-access environments to manage address conservation and enhance security.

Note NATis supported on stack setups, offering flexibility and scalability for enterprise deployments.

Benefits of Configuring NAT

NAT provides several benefits for enterprise networks:

- IP address conservation: NAT helps organizations address IPv4 address exhaustion by mapping multiple internal (private) addresses to a smaller pool of public (globally routable) addresses. This is especially useful for networks without enough registered IP addresses to assign to all devices.

- Enhanced security: NAT hides internal client addresses from external networks, reducing exposure to direct attacks. By presenting a single or limited set of external addresses, NAT limits the visibility of internal network topology.
- Flexible address management: NAT allows the use of RFC 1918 private addresses or registered addresses, enabling network administrators to manage address spaces efficiently based on organizational needs.
- Device-level deployment: NAT is implemented on specific devices, eliminating the need for changes across all endpoints. Only the devices configured with NAT require modification.
- Selective translation: NAT allows administrators to select which internal hosts are eligible for address translation, supporting granular control over network traffic.
- Stacking support:NAT operates in stack mode, supporting high-availability deployments and seamless expansion.

How NAT Works

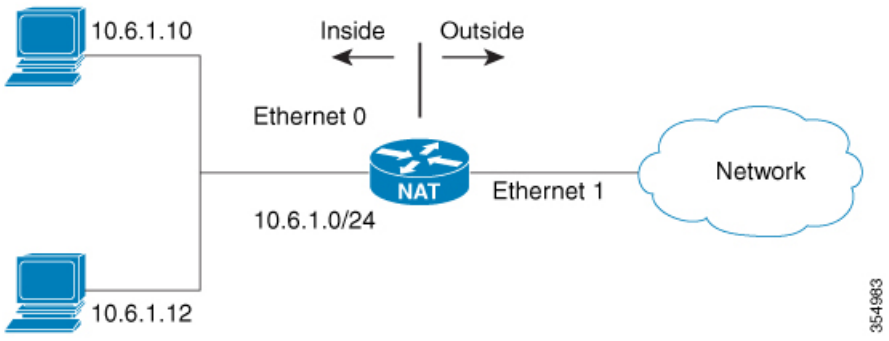
A device configured for NAT must have at least one interface connected to the internal (inside) network and at least one interface connected to the external (outside) network. Typically, NAT is deployed at the network edge, often at the boundary between a private stub domain and a public backbone network.

When a packet exits the internal network, NAT translates its locally significant source address to a globally unique external address. When a packet enters the internal network, NAT translates the globally unique destination address back to a local address. If multiple inside networks are connected or multiple external paths exist, NAT manages translation and forwarding accordingly.

If NAT runs out of available addresses for translation, it drops the affected packet and returns an Internet Control Message Protocol (ICMP) host unreachable message to the sender.

Note Translation and forwarding with NAT is processed in the hardware switching plane on Cisco Series Switches, which maximizes throughput.

Figure 1: Network Address Translation



NAT Usage Scenarios

Network Address Translation (NAT) is used in the following scenarios:

- Internet connectivity with limited public addresses: NAT enables an organization to connect to the Internet even if only a few hosts have globally unique IP addresses. NAT is most practical when only a small number of hosts need to communicate externally at the same time, allowing address reuse.
- Network renumbering: NAT allows organizations to avoid the complex and labor-intensive process of renumbering internal addresses. Instead, internal addresses are translated as needed, allowing seamless migration or coexistence with new address schemes.

NAT Address Types

NAT uses the following address definitions:

Inside local address: The IP address assigned to a host on the inside network. This is typically not a routable address assigned by a Network Information Center (NIC) or service provider.

Inside global address: A globally routable IP address representing one or more inside local addresses to the outside world. This address is assigned by the NIC or service provider.

Outside local address: The address of an external host as it appears to the inside network. This may not be globally routable and is allocated from the inside network's address space.

Outside global address: The globally routable address assigned to an external host by its owner, typically allocated from a public address space.

NAT Entry Definitions

NAT maintains translation state information using the following entry types:

Half Entry: A mapping between local and global addresses and/or ports, maintained in the NAT translation database. Half entries may be created statically or dynamically based on configured NAT rules.

Full Entry (Flow Entry): Represents a unique session flow, including both local-to-global mapping and destination information. Full entries are always created dynamically and are maintained in the translation database for the duration of the session.

NAT on Layer 3 Multi-chassis EtherChannel

NAT can be enabled on Layer 3 Multi-chassis EtherChannel (MEC) interfaces using the interface port-channel command, allowing translation across aggregated links for high availability and redundancy.

Supported Translation

Only VRF-to-Global translations are supported, where the NAT inside interface belongs to a specific VRF and the NAT outside interface is in the global routing table.

Intra-VRF NAT (inside and outside interfaces within the same VRF) and Inter-VRF NAT (inside and outside interfaces in different non-global VRFs) are not supported.

NAT behavior is undefined in unsupported scenarios. Only deploy VRF-to-Global NAT translation in your network.

Types of Network Address Translation

NetworkAddress Translation (NAT) supports several methods to translate IP addresses betweenprivate and public networks. Each method serves a different use case and provides varying levels of address conservation and flexibility.

Static NAT

Static NAT (Static Address Translation) provides a permanent, one-to-one mapping between a private (inside local) IP address and a public (inside global) IP address.

Use static NAT when a device inside your network must always be reachable from the outside using the same public IP address (for example, servers or network devices requiring fixed external access).

Dynamic NAT

Dynamic NAT (Dynamic Address Translation) maps private (inside local) addresses to public (inside global) addresses selected from a pool. The mapping is created dynamically when an inside device initiates a connection and is maintained only for the duration of the session.

Dynamic NAT is useful when you have more inside hosts than public addresses, but not all hosts need simultaneous external access.

Port Address Translation (PAT) / NAT Overloading

PAT, also known as NAT overloading, allows multiple inside local addresses to share a single inside global address. Each session is uniquely identified by a combination of the IP address and Layer 4 port number.

PAT conserves public IP addresses and enables thousands of inside hosts to access external networks using just one or a few public addresses.

Static Port Translation

Static port translation creates a fixed mapping between a specific IP address/port pair on the inside and a specific IP address/port pair on the outside.

Use static port translation when both IP address and port mapping need to be preserved for consistent access, such as for specific services hosted internally.

Subnet (Network) Static Translation

Subnet static translation allows a range of inside local subnets to be mapped to a corresponding range of inside global subnets. The mapping is defined so that each host in the subnet has a predictable translation.

This is useful when a group of internal addresses needs to be mapped to a different address block externally, such as during network migrations or for overlapping address resolution.

Inside Source Address Translation

Inside source address translation translates the source address of packets from internal hosts (inside local) to an external, globally routable address (inside global).

Allows private hosts to initiate connections to the public network and appear as routable addresses.

1. Connection Initiation

The user at host 10.1.1.1 initiates a connection to an external host (Host B) located on the outside network.

2. NAT Rule Evaluation and Translation Decision

   The NAT module intercepts the outbound packet and evaluates it against the configured NAT rules:

   - If a matching static translation rule exists, the packet is translated to the specified inside global address.
   - If no static rule matches, the packet is checked against dynamic translation rules. If a matching dynamic rule is found, the packet is translated using an available inside global address from the configured pool.
   - Upon a successful translation, the NAT module creates and stores a fully qualified flow entry in its translation database. This enables fast bidirectional translation and forwarding for future packets in the same flow.
   - If no rule matches, the packet is forwarded without translation.
   - If a matching dynamic rule is found but no valid inside global address is available, the packet is dropped.

Note If an Access Control List (ACL) is used for dynamic NAT, only packets permitted by the ACL are considered for translation.

3. Source Address Replacement and Fowarding

   The device replaces the source IP address (inside local) of host 10.1.1.1 with the inside global address (for example, 203.0.113.2). Only packet-relevant checksums are updated; all other packet fields remain unchanged. The translated packet is then forwarded to the outside network.

4. Translation Flow Entry Maintenance

   The NAT module maintains a fully qualified flow entry for the translated session in its translation database. This entry facilitates fast hardware translation and forwarding for all subsequent packets in this flow, in both directions.

5. Response from Outside Host

   Host B receives the packet and sends a response addressed to the inside global IP address (203.0.113.2).

6. Reverse Translation

   The NAT module intercepts the response packet destined for the inside global address. Usingthe stored flow entry, the module translates the destination address back to the original inside local address (10.1.1.1), and forwards the packet to the internal host.

7. Ongoing Conversation

   Host 10.1.1.1 receives the response and communication continues. The NAT device repeats Steps 2–6 for each packet in the session.

Outside Source Address Translation

Outside source address translation translates the source address of packets from external hosts (outside global) to an internal, locally significant address (outside local).

Used in advanced scenarios, such as when connecting to overlapping networks or allowing external devices to appear as unique addresses within your private network. For a detailed process, see the section on Overlapping Networks.

Port Address Translation (PAT)

Port Address Translation (PAT), also knownas NAT Overloading, allows multiple internal hosts to share a single inside global IP address for external communications by differentiating flows using Layer 4 (TCP or UDP) port numbers. This method conserves address space while providing connectivity for many users.

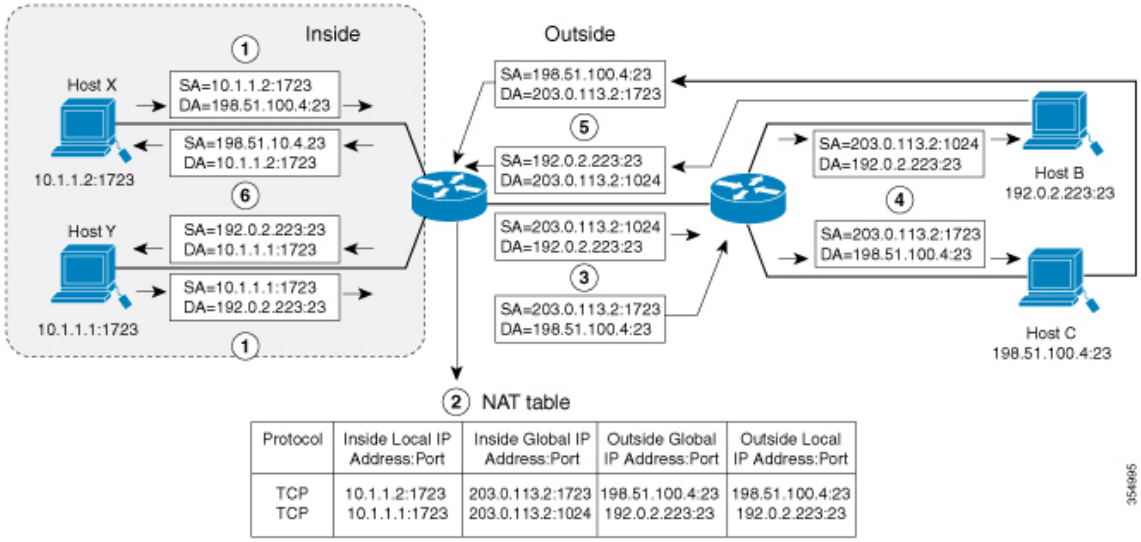
How PAT Works

The NAT device tracks each session based on the combination of inside local address, inside global address, and the transport protocol port number.

When multiple internal hosts communicate with external hosts, the NAT device assigns a unique port number for each session, ensuring correct mapping between local and global addresses.

Return traffic uses the inside global address and the appropriate port number, allowing the NAT device to identify the corresponding inside local address and port.

Figure 2: PAT/ NAT Overloading Inside Global Addresses



Port Address Translation (PAT), also called overloading, enables multiple inside hosts to share a single inside global IP address by using unique Layer 4 port numbers as differentiators. This allows many users to access external resources while conserving public IP address space.

How PAT (Overloading) Works

1. Multiple Connections Initiated

The user at Host Y (10.1.1.1) opens a connection to Host B. Theuser at Host X (10.1.1.2) opens a connection to Host C.

2. NAT Rule Evaluation and Translation

   The NAT module intercepts the packets and evaluates them against configured NAT rules.

   - If a matching static translation rule exists, it takes precedence; the packets are translated to the corresponding global address.
   - If no static rule matches, dynamic translation rules are checked. If a match is found,translation uses the available global address.
   - For each translation, the NAT module creates a fully qualified flow entry (including address and port mappings) in the translation database for fast, bidirectional translation.
   - If no matching rule is found, packets are forwarded untranslated.
   - If a matching rule is found but no valid inside global address is available, packets are dropped.

   In PAT configurations, the device also translates the source port, enabling multiple flows to share a single global address.

3. Address and Port Translation

   The device replaces the inside local source addresses and ports (10.1.1.1:1723 and 10.1.1.2:1723) with the single inside global address, assigning unique source ports (e.g., 203.0.113.2:1024 for Host Y and 203.0.113.2:1723 for Host X). The packets are then forwarded to their respective outside destinations.

4. Response Handling

   Host B responds to Host Y by sending a packet to 203.0.113.2:1024. Host C responds to Host X by sending a packet to 203.0.113.2:1723.

5. Reverse Translation and Delivery

   When the device receives return packets with the inside global address and port, it looks up the translation table entry using both the address and port, as well as the outside address and port.

   The device then translates the destination back to the correct inside local address and port (10.1.1.1:1723 or 10.1.1.2:1723) and forwards the packet to Host Y or Host X.

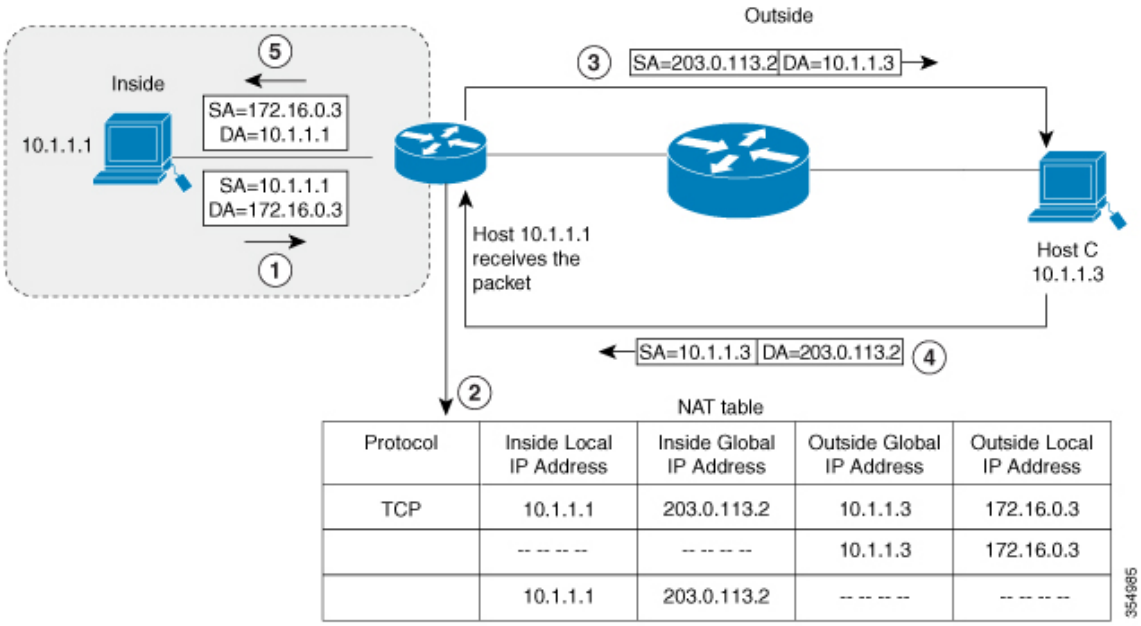
6. Ongoing Communication

   Host Y and Host X continue communication with their respective external hosts. The NAT device repeats Steps 2–5 for each packet in the session.

Overlapping Networks

Overlapping networks occur when internal and external networks use the same IP addressranges, often due to the use of unofficial or duplicate address assignments.

Figure 3: NAT Translating Overlapping Addresses



Overlapping Address Translation Process

When the inside local address and the outside global address belong to the same subnet (an overlapping network scenario), NAT enables communication by translating both sides of the conversation. The following steps describe how NAT handles overlapping addresses:

- Session Initiation

  Host 10.1.1.1 opens a connection to 172.16.0.3.

- Translation Mapping Creation

  The NAT module creates translation mappings:

  Inside local address (10.1.1.1) is mapped to an inside global address (203.0.113.2), creating a "half entry" in the NAT table.

  On the receiving side, the outside global address (10.1.1.3) is translated to an outside local address (172.16.0.3), creating another half entry.

- Address Replacement and Forwarding

  The device replaces the source address (SA) with the inside global address and the destination address (DA) with the outside global address before forwarding the packet.

- Communication Continues

  Host C receives the packet and continues the session.

- Reverse Translation on Return Packets

  The device performs a NAT table lookup for return traffic, replacing the DA with the inside local address and the SA with the outside local address.

- Session Maintenance

  Host 10.1.1.1 receives the reply and the translated conversation continues.

Note Half entries are created for each direction initially. Once both halves are established, the NAT table is updated with a full entry for the session, enabling efficient bidirectional translation.

Application-Level Gateways with NAT

NAT translates TCP/UDP traffic that does not carry source/destination IP addresses in the application payload. However, some protocols embed address or port information insidethe payload, which requires special handling.

The NAT Application-Level Gateway (ALG) feature enables proper translationfor applications that embed address or port information within the payload. Cisco Series switches support ALGs for FTP, TFTP, and ICMP only.

The ALG processes both packet headers and relevant payload fields, establishing temporary mappings as needed.

Best Practices for NAT Configuration

Follow these best practices while configuring NAT.

- Avoid overlapping local addresses in static and dynamic rules. If unavoidable, ensure dynamic rule ACLs exclude static rule addresses.
- Do not use loose filtering (e.g., permit ip any any) in NAT ACLs, as this can cause unintended traffic to be translated.
- Do not share address pools across multiple NAT rules.
- Do not define the same inside global address in both static NAT and dynamic pools.
- Be cautious with timeout values; small values can cause high CPU usage.
- Avoid manual clearing of translations during active sessions, as this may disrupt applications.
- Before reconfiguring NAT during active use:
  - Stop traffic matching the configuration (use ACLs or shut interfaces).
  - Clear existing translation entries.
  - Apply configuration changes and re-enable traffic.

Limitations of NAT

These are the limitations of NAT on the Cisco C9000 Series Smart Switches.

- Route map-based NAT is not supported.
- Twice NAT is not supported on Cisco C9350 Series Smart Switches.
- VRF-aware NAT is not supported.
- If an ACL contains a deny statement for a specific host and traffic from that host matches the corresponding ACE, you will observe continuous punt events for that traffic.
- Pool with type match-host not supported.
- After 13.5k flows, packets will be software forwarded.

How to Configure NAT

Use these tasks to for different NAT configurations.

Configuration Examples for Network Address Translation

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
```

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

!

!
```

Configuring Static Translation of Inside Source Addresses

Static translation provides a permanent, one-to-one mapping between an inside local addressand an inside global address. This is required for hosts that must be consistently accessible from outside the network.

Procedure

	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 commands depending on the requirement: • **ip nat inside source static** *local-ip global-ip* `Switch(config)# ip nat inside source static 10.10.10.1 172.16.131.1` • **ip nat inside source static** *protocol local-ip port global-ip port*	• 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.

	Command or Action	Purpose
	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 • **ip nat inside source static** *local-ip global-ip* **vrf** *vrf-name* Switch(config)# ip nat inside source static 10.10.10.1 172.16.131.1 vrf vrf1	• Allows a static translation mapping for an entire subnet without the need for specifying multiple individual translation rules. You can specify the translation mapping for the desired subnet. The actual translation is performed by translating the network portion of the address with the host the portion remaining unchanged. • Makes the static translation VRF aware and associates the given rule with the specified VRF.
Step 4	**interface** *type number* **Example:** Switch(config)# interface ethernet 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 0/0/0	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.

	Command or Action	Purpose
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

Dynamictranslation establishes a mapping between an inside local address and a pool of global addresses dynamically. Dynamic translation can be enabled by configuring a

dynamicNAT 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 addressand 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 releasedfor use by other users when access to the internet is no longer required.

Procedure

	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:** `Switch(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* **vrf** *vrf-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. Using the **vrf** keyword makes the dynamic translation VRF aware and associates the given rule with the specified VRF.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Switch(config)# interface ethernet 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 ethernet 0`	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 Port Address Translation by Overloading of Global Addresses

NATmodule supports dynamic PAT configurations through address pools and interface, as described in the following tasks.

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

Procedure

	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* [**vrf** *vrf-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. Using the **vrf** keyword makes the dynamic translation VRF aware and associates the given rule with the specified VRF.
Step 6	**interface** *type number* **Example:** `Switch(config)# interface ethernet 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.

	Command or Action	Purpose
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 ethernet 0`	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 Port Address Translation by Overloading an Interface

Perform the following task to allow your internal users access to the internet and conserve addresses in the inside global addresstepy overloading an interface.

Procedure

	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.

	Command or Action	Purpose
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* [**vrf** *vrf-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. Using the **vrf** keyword makes the dynamic translation VRF aware and associates the given rule with the specified VRF.
Step 6	**interface** *type number* **Example:** `Switch(config)# interface ethernet 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 ethernet 0`	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.

	Command or Action	Purpose
Step 13	**end** **Example:** `Switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Network Address Translation of External IP Addresses Only

By default, NAT translates the addresses embedded in the packet pay-load. 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.

Procedure

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

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

Configuring 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 aroute 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.

Procedure

	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 ethernet 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.
Step 8	**exit** **Example:** `Switch(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 9	**interface** *type number* **Example:** `Switch(config)# interface ethernet 0`	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
- UDPflow: 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

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 duringa reset, change the default TCP timeout by using the **ip nat translation tcp-timeout** command.

Procedure

	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:** `Switch(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.

	Command or Action	Purpose
Step 9	**end** **Example:** `Switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Network Address Translation on Layer 3 Port Channel

Procedure

	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	**interface port-channel** *port-channel-number* **Example:** `Switch(config)# interface port-channel 10`	Enters port-channel interface mode.
Step 4	**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 5	**ip nat inside** **Example:** `Switch(config-if)# ip nat inside`	Connects the interface to the inside network, which is subject to NAT.
Step 6	**interface port channel** *port-channel-number* **Example:** `Switch(config)# interface port-channel 11`	Enters port-channel interface mode.
Step 7	**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 8	**ip nat outside** **Example:**	Connects the interface to the outside network.

	Command or Action	Purpose
	`Switch(config-if)# ip nat outside`	
Step 9	**end** **Example:** `Switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

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 translations** command.
- Verify that timer values are correctly configured using the **show ip nat translations 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 translations** command to clear the NAT translational table entires before the timer expires.
- Use **debug ip nat** and **debug ip nat detailed** commands to debug NAT configuration.