AT&T

NAT Configuration Guide, 17.2.0

November 2017

Supporting AT&T Vyatta Network Operating System

Contents

Copyright Statement

The training materials and other content provided herein for assistance in training on the Vyatta vRouter may have references to Brocade as the Vyatta vRouter was formerly a Brocade product prior to AT&T's acquisition of Vyatta. Brocade remains a separate company and is not affiliated to AT&T.

About This Guide

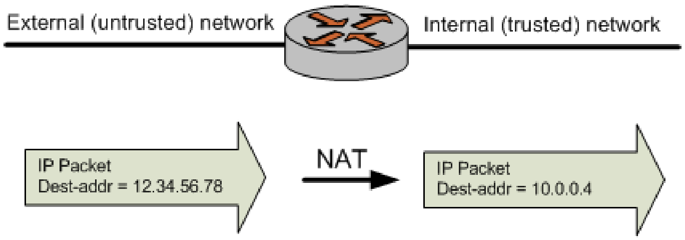
This guide describes how to configure NAT on AT&T products that run on the AT&T Vyatta Network Operating System (referred to as a virtual router, vRouter, or router in the guide).

NAT Overview

What is NAT?

Network Address Translation (NAT) is a service that modifies address, port, or both types of information within network packets as they pass through a computer or network device. The device that performs NAT on the packets can be the source of the packets, the destination of the packets, or an intermediate device on the path between the source and destination devices.

Figure 1: An example of a device that performs NAT



NAT was originally designed to help conserve the number of IP addresses used by the growing number of devices accessing the Internet, but it also has important applications in network security.

The computers on an internal network can use any of the addresses set aside by the Internet Assigned Numbers Authority (IANA) for private addressing (refer to RFC 1918). These reserved IP addresses are not in use on the Internet, so an external machine does not directly route to them. The following addresses are reserved for private use:

- 10.0.0.0 through 10.255.255.255 (CIDR: 10.0.0.0/8)
- 172.16.0.0 through 172.31.255.255 (CIDR: 172.16.0.0/12)
- 192.168.0.0 through 192.168.255.255 (CIDR: 192.168.0.0/16)

A NAT-enabled router can hide the IP addresses of an internal network from the external network by replacing the internal, private IP addresses with public IP addresses that have been provided to it. These public IP addresses are the only addresses that are ever exposed to the external network. The router can manage a pool of multiple public IP addresses from which it can dynamically choose when performing address replacement.

Be aware that, although NAT can minimize the possibility that internal computers make unsafe connections to the external network, it provides no protection to a computer that, for one reason or another, connects to an untrusted machine. Therefore, you should always combine NAT with packet filtering and other features of a complete security policy to fully protect your network.

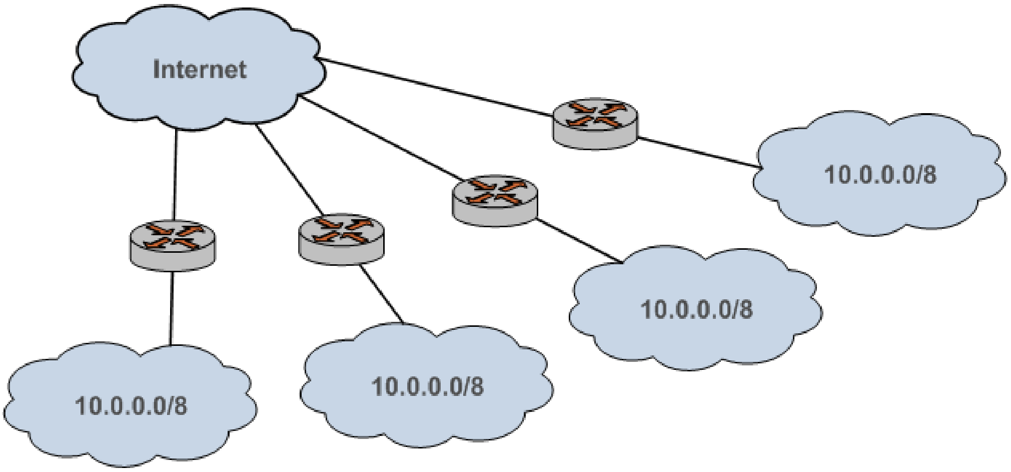
For more information, refer to *IPsec Site-to-Site VPN Reference Guide*.

Benefits of NAT

NAT confers the following advantages:

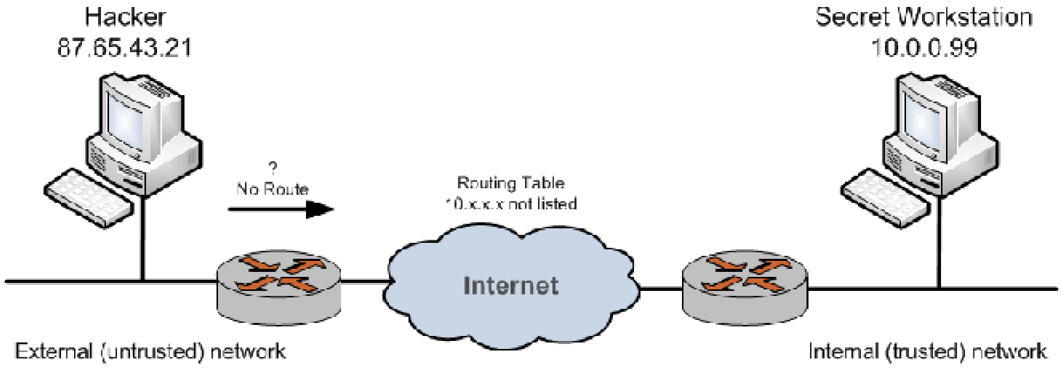
- NAT conserves public Internet address space.
  Any number of hosts within a local network can use private IP addresses instead of consuming public IP addresses. The addresses of packets that are transmitted from this network to the public Internet are translated to the appropriate public IP address. This translation means that the same private IP address space can be reused within any number of private networks, as shown in the following figure.

Figure 2: Reusing private address space



- NAT enhances security.
  IP addresses within a private (internal) network are hidden from the public (external) network. This hiding of addresses makes it more difficult for hackers to initiate an attack on an internal host. However, private network hosts are still vulnerable to attack; therefore, NAT is typically combined with firewall functionality.

Figure 3: NAT combined with firewall



- NAT is seamless.
  Standard client/server network services work without modification through a NAT-enabled device.
- NAT facilitates network migration from one address space to another.
  The address space within a private network that is having NAT performed on it is independent of the public IP address. This independence means that the private network can be moved to a new public IP address without changing network configurations within the private network. Likewise, the addressing within the private network can change without affecting the public IP address.
- NAT simplifies routing.
  NAT reduces the need to implement more complicated routing schemes within larger local networks.

Types of NAT

NAT has three main types:

- Source NAT. This NAT is also called SNAT. “Masquerade” NAT is a special type of SNAT.
- Destination NAT. This NAT is also called DNAT.
- Bidirectional NAT. When both SNAT and DNAT are configured, the result is bidirectional NAT.

Source NAT (SNAT)

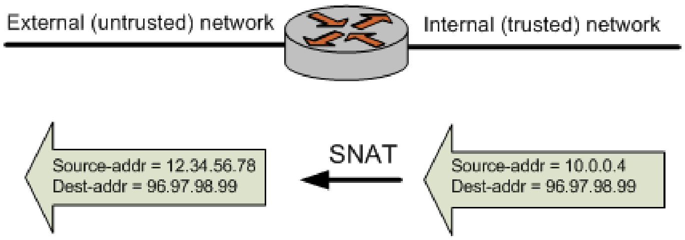
Source NAT (SNAT) is the most common form of NAT. SNAT changes the source address of the packets passing through the AT&T Vyatta vRouter. SNAT is typically used when an internal (private) host needs to initiate a session to an external (public) host; in this case, the device that is performing NAT changes the private IP address of the source host to some public IP address, as shown in the following figure. In “masquerade” NAT (a common type of SNAT), the source address of the outgoing packet is replaced with the primary IP address of the outbound interface. The destination address of return packets is automatically translated back to the IP address of the source host.

Note: SNAT is performed after the routing decision is made.

The device that is performing NAT tracks information about the traffic flow so that traffic from the flow can be correctly forwarded to and from the source host.

Note: If an IP-in-IP, IP-in-IP6, IP6-in-IP6, SIT, or OpenVPN tunnel is configured as an outbound interface for SNAT, you cannot use a local address as a translation address.

Figure 4: Source NAT (SNAT)

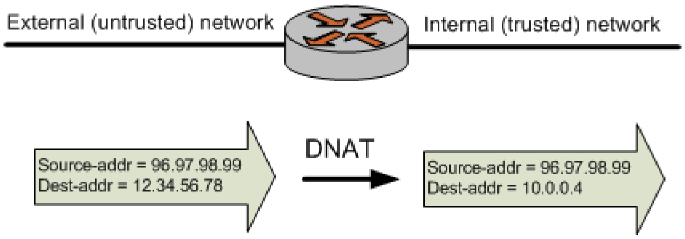

External (untrusted) network
Internal (trusted) network
Source-addr = 12.34.56.78
Dest-addr = 96.97.98.99
SNAT
Source-addr = 10.0.0.4
Dest-addr = 96.97.98.99

Destination NAT (DNAT)

While SNAT changes the source address of packets, destination NAT (DNAT) changes the destination address of packets passing through the AT&T Vyatta vRouter. DNAT is typically used when an external (public) host needs to initiate a session with an internal (private) host; for example, when a subscriber accesses a news service, as shown in the following figure. The source address of return packets is automatically translated back to the IP address of the source host.

Note: DNAT is performed before the routing decision is made.

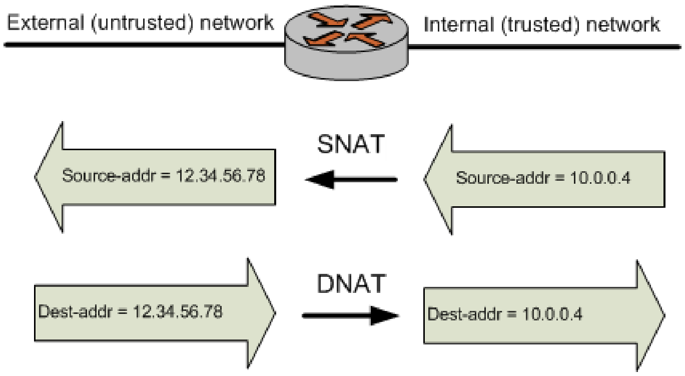
Figure 5: Destination NAT (DNAT)


External (untrusted) network
Internal (trusted) network
Source-addr = 96.97.98.99
Dest-addr = 12.34.56.78
DNAT
Source-addr = 96.97.98.99
Dest-addr = 10.0.0.4

Bidirectional NAT

Bidirectional NAT is just a scenario in which both SNAT and DNAT are configured at the same time. Bidirectional NAT is typically used when internal hosts need to initiate sessions with external hosts and external hosts need to initiate sessions with internal hosts. The following figure shows an example of bidirectional NAT.

Figure 6: Bidirectional NAT

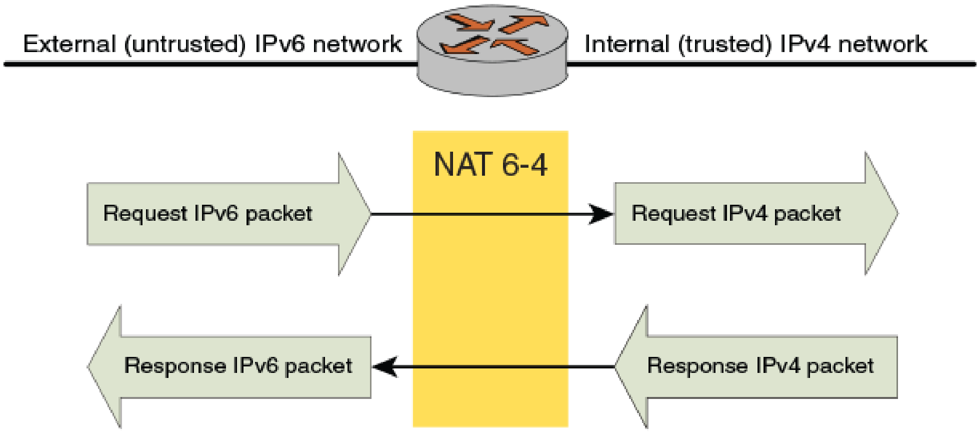

External (untrusted) network
Internal (trusted) network
SNAT
Source-addr = 12.34.56.78
Source-addr = 10.0.0.4
DNAT
Dest-addr = 12.34.56.78
Dest-addr = 10.0.0.4

NAT 6-4

The AT&T Vyatta vRouter supports NAT 6-4, a translator that allows IPv6-only networks to communicate with IPv4-only networks. The router forwards the IPv6 packets to the translator, which converts them into IPv4 packets. The response IPv4 packets flow in the opposite direction with the NAT6-4 translator converting IPv4 addresses into IPv6.

The following figure shows the flow of packets, which originate in an IPv6-only network.

Figure 7: NAT 6-4 packet flow



IPv4 and IPv6 address formats

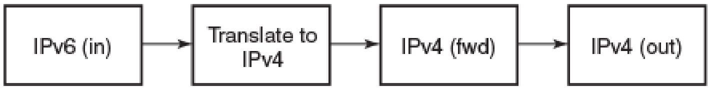
To perform NAT 6-4 translation, the router rewrites an incoming IPv6 packet into an IPv4 address space for the source and destination of the packet. This rewrite requires that the router modify the Ethernet and the IP headers.

For more information about the IPv4 and IPv6 address formats, refer to RFC 6052 (section 2.2).

Packet processing

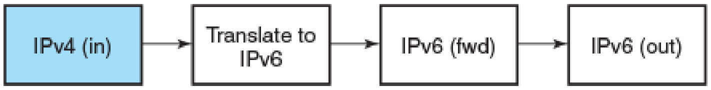
The following figure shows the forwarding pipeline stages for processing of incoming IPv6 packets. Incoming IPv6 packets are processed by IPv6 (in) and IPv4 (in), and IPv4 (out) firewalls.

Figure 8: Forwarding pipeline stages



The following figure shows the reverse pipeline stages for processing incoming IPv4 packets. Outgoing IPv4 packets are processed by the IPv4 (in), and IPv6 (in) and IPv6 (out) firewalls.

Figure 9: Reverse pipeline stages



NAT 6-4 sessions

You can display information about sessions that are created as a result of NAT 6-4 conversion by using the `show session-table` command, as shown in the following example. This example shows a NAT 6-4 session that is created to support an ICMP session.

```
vyatta@vyatta# run show session-table
TCP state codes: SS - SYN SENT, SR - SYN RECEIVED, ES - ESTABLISHED,
                 FW - FIN WAIT, CW - CLOSE WAIT, LA - LAST ACK,
                 TW - TIME WAIT, CL - CLOSE, LI - LISTEN

CONN ID   Source        Destination   Protocol  TIMEOUT   Intf              Parent
1         198.18.4.200  20.20.10.4    icmp [1]  14        dp0p224p1    0
```

What NAT 6-4 supports

The AT&T Vyatta vRouter NAT 6-4 translator supports the following:

- TCP and UDP protocols and Internet Control Message Protocol (ICMP) echo requests.

  Note: ICMP is supported only for echo requests and responses.

- Working with firewalls (stateful and stateless).
- Working with DNAT and SNAT.
- Address formats as specified in section 2.2 of RFC 6052.
- Stateful connection tracking and validation (by way of the NPF session table).
- Selective packet filtering of source and destination prefixes on the inbound interface of the IPv6 network.

What NAT 6-4 does not support

The AT&T Vyatta vRouter NAT 6-4 translator does not support the following:

- Bidirectional NAT (SNAT and DNAT).
- Sessions initiated from an IPv4 network.
- Multihomed behavior in an IPv4 network.

Supported protocols

NAT 6-4 supports the following protocols:

- TCP
- UDP
- ICMP

Main NAT 6-4 operations

The NAT 6-4 translator maintains state and, for enabled interfaces, inspects the incoming IPv6 packets that match the configured filter (the filter specifies the IPv6 prefix) to determine whether the packet requires translation between IP versions.

Inspecting packets involves performing a lookup against configured IPv6 prefixes, deriving IPv4 addresses from IPv4 addresses, and performing lookups of IPv4 addresses against a table of converted addresses for the reverse flow. Translation implies changes to the Ethernet frame, the IP header, and updates to the TCP, UCP, and ICMP checksums. In brief, NAT 6-4 is highly invasive.

One-way translation

The AT&T Vyatta vRouter NAT 6-4 translator supports one-way translation with the IPv6 network initiating communication with the IPv4 network.

Interaction between NAT, routing, firewall, and DNS

One of the most important concepts to understand when working with NAT is the processing order of the various services that might be configured within the AT&T Vyatta vRouter. If the processing order of the services is not considered, the results achieved might not be what you expect. This section covers the following topics:

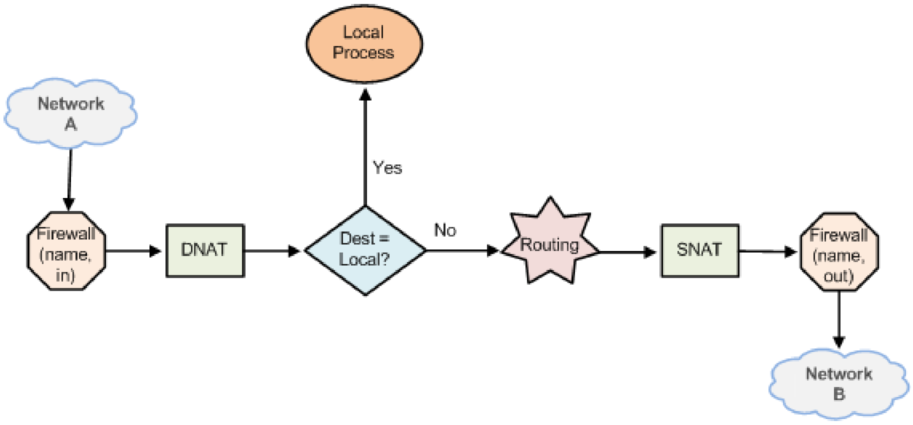
- Traffic flow through firewall, NAT, and routing *(page 14)*
- Interaction between NAT and routing *(page 14)*
- Interaction between NAT and firewall *(page 16)*
- Interaction between NAT and DNS *(page 18)*

Traffic flow through firewall, NAT, and routing

For example, if you are using DNAT, you should take care not to set up the system to route packets based on particular external addresses. This routing method would not have the expected result because the addresses of external packets would have all been changed to internal addresses by DNAT before routing.

The following figure shows the traffic flow between NAT, routing, and firewall within the AT&T Vyatta vRouter.

Figure 10: Traffic flow through the AT&T Vyatta vRouter



Interaction between NAT and routing

When considering NAT in relation to routing, it is important to be aware how routing decisions are made with respect to DNAT and SNAT. The scenarios in this section illustrate this point.

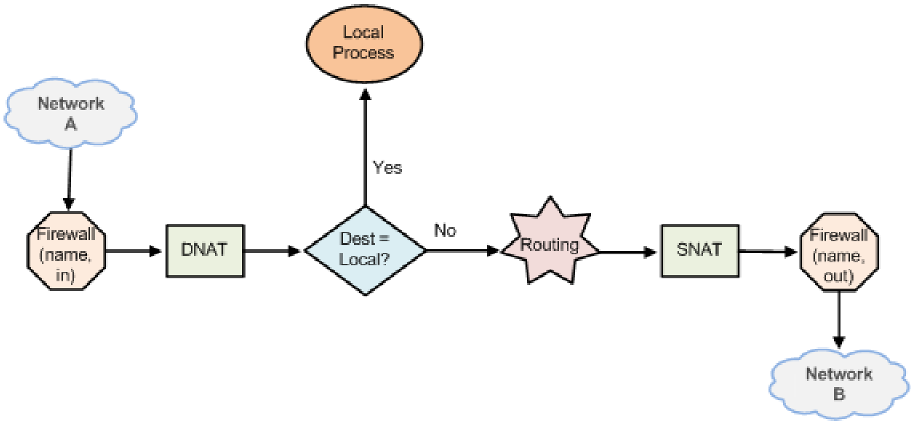
Scenario 1a: DNAT—Packets passing through the AT&T Vyatta vRouter

In this scenario, packets originate in Network A and pass through the AT&T Vyatta vRouter.

Note: DNAT—routing decisions are based on translated destination address.

Note that DNAT operates on the packets before the routing decision. This sequence means that routing decisions based on the destination address are made relative to the translated destination address—not the original destination address; refer to the following figure.

Figure 11: Pass-through DNAT routing decisions

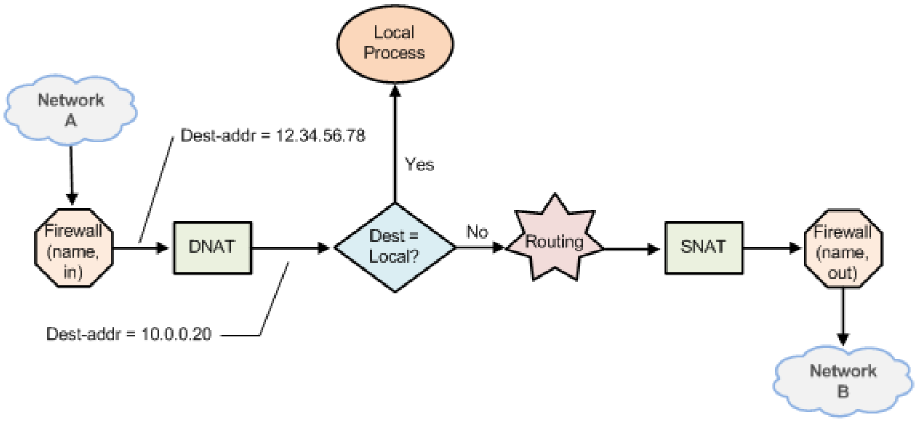


Scenario 1b: DNAT—Packets destined for the AT&T Vyatta vRouter

In this scenario, packets are destined for a process within the AT&T Vyatta vRouter. For packets destined for the AT&T Vyatta vRouter, routing decisions based on the destination address are made relative to the translated destination address—not the original destination address.

Again, because DNAT operates on the packets before the routing decision, routing decisions based on destination address are made on the translated destination address—not the original destination address; refer to the following figure.

Figure 12: AT&T Vyatta vRouter-destined DNAT routing decisions



Scenario 2a: SNAT—Packets passing through the AT&T Vyatta vRouter

On the other hand, routing decisions are made before SNAT. This sequence means that routing decisions based on the source address are made on the original source address—not the translated source address.

Note: SNAT routing decisions are based on original source address.

Scenario 2b: SNAT—Packets originating from the AT&T Vyatta vRouter

In this scenario, packets originate with a process in the AT&T Vyatta vRouter. Again, because routing decisions are made before SNAT, operations based on source address are made on the translated source address—not the original source address.

Interaction between NAT and firewall

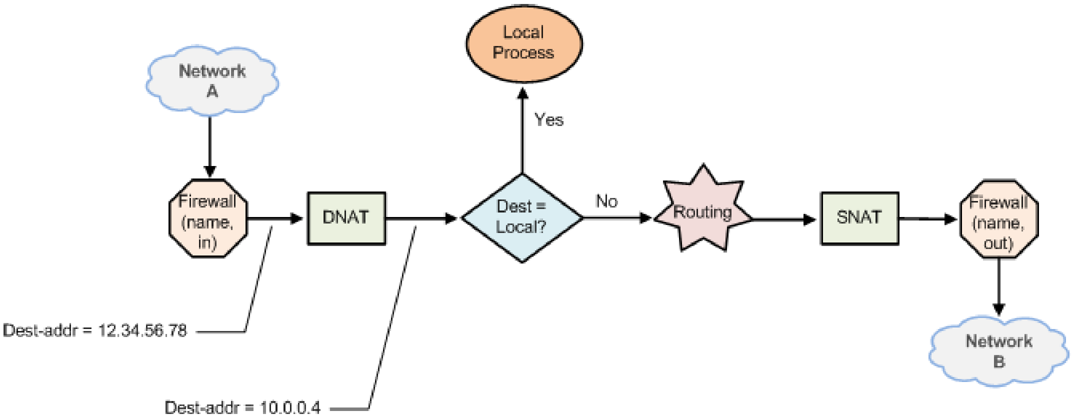
When considering NAT in relation to firewall, it is important to understand the traffic flow between NAT and firewall. In particular, it is important to keep in mind that firewall rule sets are evaluated at different points in the traffic flow. The scenarios in this section illustrate this point.

Scenario 1a: DNAT—Packets passing through the AT&T Vyatta vRouter

In this scenario, packets originate in Network A and pass through the AT&T Vyatta vRouter. Note the following rule applications:

- For firewall rule sets applied to inbound packets on an interface, the firewall rules are applied before DNAT (that is, on the translated destination address).
- For rule sets applied to outbound packets on an interface, the firewall rules are applied after DNAT (that is, on the translated destination address); refer to the following figure.

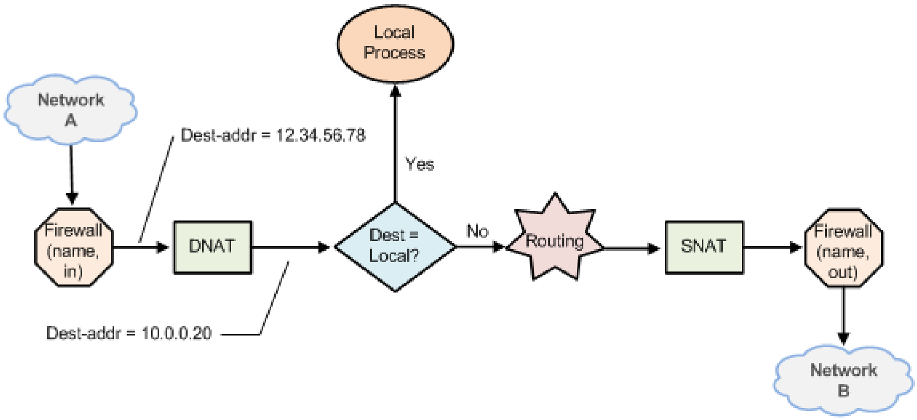
Figure 13: Pass-through DNAT firewall decisions



Scenario 1b: DNAT—Packets destined for the AT&T Vyatta vRouter

In this scenario, packets are destined for a process within the AT&T Vyatta vRouter. When firewall rule sets are applied to locally bound packets on an interface, the firewall rules are applied before DNAT (that is, on the translated destination address); refer to the following figure.

Figure 14: AT&T Vyatta vRouter-destined DNAT firewall decisions

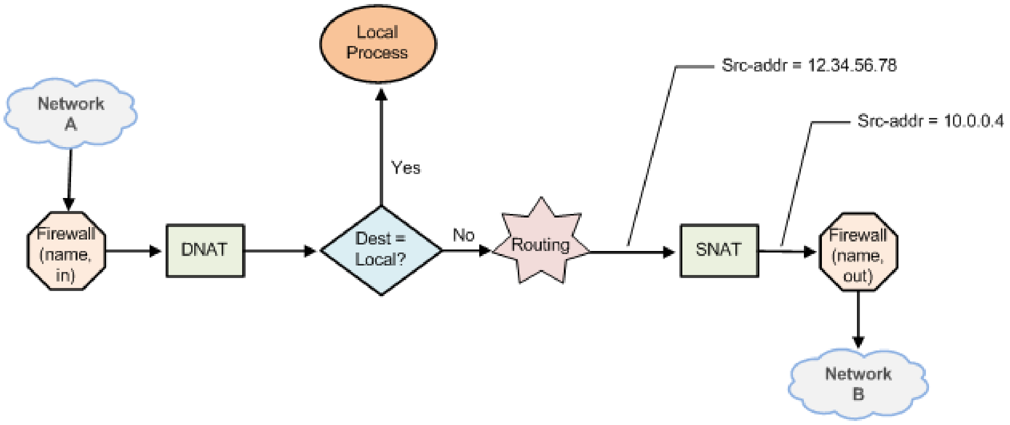


Scenario 2a: SNAT—Packets passing through the AT&T Vyatta vRouter

Firewall rules are applied before DNAT. This sequence means that firewall decisions based on source address are made on the translated source address—not the original source address. This order of evaluation is true for both inbound and outbound packets; refer to the following figure.

> **Note:** SNAT firewall rules are applied on original source address.

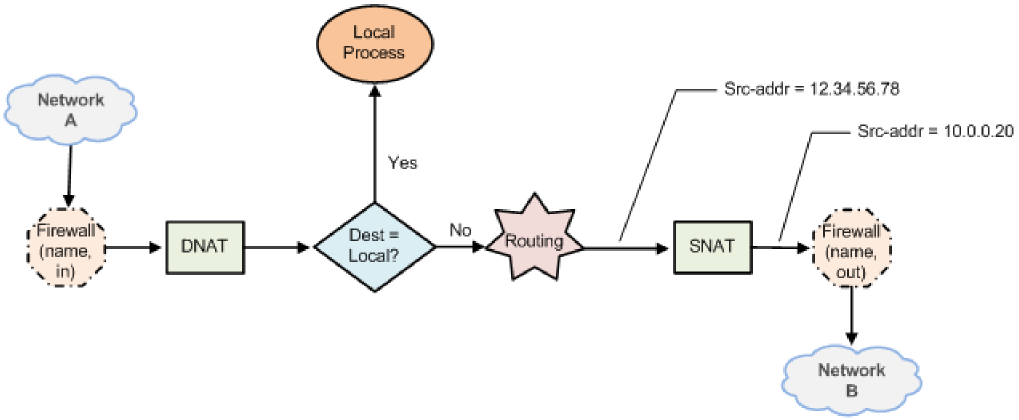
Figure 15: Pass-through SNAT firewall decisions



Scenario 2b: SNAT—Packets originating from the AT&T Vyatta vRouter

In this scenario, packets originate with a process in the AT&T Vyatta vRouter. Firewall rule sets are not involved.

Figure 16: AT&T Vyatta vRouter-originated SNAT firewall decisions



Interaction between NAT and DNS

NAT and DNS can be combined in various scenarios involving load balancing. These scenarios can include additional load-balancing switches that operate at higher protocol layers (Layers 4 through 7). For example, a large bank might have many web servers with transactions load-balanced across them.

In these cases, the NAT configuration must be carefully considered to achieve the desired results. Discussion of DNS and load-balancing scenarios is beyond the scope of this guide.

NAT rules

NAT is configured as a series of NAT "rules." Each rule instructs NAT to perform a network address translation that you require. NAT rules are numbered and are evaluated in numerical order. The NAT rule number can be changed by using the `rename` and `copy` commands.

> **Note:** Changes to NAT rules affect only connections established after the changes are made. Those connections that are already established at the time a change is made are not affected.

> **Note:** Leave a gap between NAT rule numbers.

It is advisable to create your NAT rules leaving "space" between the numbers. For example, you might initially create your set of NAT rules numbered 10, 20, 30, and 40. This way, if you need to insert a new rule later and you want it to run in a particular sequence, you can insert it between existing rules without having to change any other rules.

Creating a SNAT rule

The AT&T Vyatta vRouter allows you to configure SNAT and DNAT rules. To implement bidirectional NAT, you define a NAT rule for SNAT and one for DNAT. The following example shows how to define a SNAT rule, rule 10.

```
vyatta@vyatta# set service nat source rule 10
```

Traffic filters

Filters control which packets have the NAT rules applied to them. Five different filters can be applied within a NAT rule: **`outbound-interface`**, **`inbound-interface`**, **`protocol`**, **`source`**, and **`destination`**.

The "outbound-interface" filter

The **`outbound-interface`** filter applies only to SNAT rules. It specifies the outbound traffic flow to which NAT applies.

Filtering outbound traffic

The following example shows how to apply a SNAT rule, rule 20, to outbound traffic on the dp0p1p2 interface.

```
vyatta@vyatta# set service nat source rule 20 outbound-interface dp0p1p2
```

The "inbound-interface" filter

The **`inbound-interface`** filter applies only to DNAT rules. It specifies the inbound traffic flow to which NAT applies.

Filtering inbound interface

The following example shows how to apply a DNAT rule, rule 20, to inbound traffic on the dp0p1p1 interface.

```
vyatta@vyatta# set service nat destination rule 20 inbound-interface dp0p1p1
```

The protocol filter

The **`protocol`** filter specifies the protocols to which NAT applies. NAT applies only to packets of the specified protocol. The default protocol is **`all`** protocols. The **`protocol`** filter can be used in SNAT and DNAT rules.

Filtering packets by protocol

The following example shows how to apply a SNAT rule, rule 10, to TCP protocol packets. Only TCP packets have address translation performed.

```
vyatta@vyatta# set service nat source rule 10 protocol tcp
```

The "source" filter

The **`source`** filter specifies the packets to which NAT applies based on their source address, port, or both. NAT applies only to packets that have a source address, port, or both that match that defined in the filter.

If the **`source`** filter is not specified, then by default, the rule matches packets arriving from any source address and port. The **`source`** filter can be used in SNAT and DNAT rules.

Filtering packets by source address

The following example shows how to apply a SNAT rule, rule 10, to packets with a source address of 10.0.0.4. Only packets with a source address of 10.0.0.4 have address translation performed.

```
vyatta@vyatta# set service nat source rule 10 source address 10.0.0.4
```

Filtering packets by source network address and port

The following example shows how to apply a SNAT rule, rule 20, to packets with a source network of 10.0.0.0/24 and a port of 80. Only packets with a source address on the 10.0.0.0/24 subnet with a source port of 80 have address translation performed.

```
vyatta@vyatta# set service nat source rule 20 source address 10.0.0.0/24

vyatta@vyatta# set service nat source rule 20 source port 80
```

The "destination" filter

The **`destination`** filter specifies the packets to which NAT applies based on their destination address, port, or both. NAT applies only to packets that have a destination address, port, or both that match that defined in the filter.

If the `destination` filter is not specified, then by default, the rule matches packets sent to any destination address and port. The `destination` filter can be used in SNAT and DNAT rules.

Filtering packets by destination address

The following example shows how to apply a SNAT rule, rule 30, to packets with a destination address of 12.34.56.78. Only packets with a destination address of 12.34.56.78 have address translation performed.

```
vyatta@vyatta# set service nat source rule 30 destination address 12.34.56.78
```

Address conversion: translation addresses

The translation address defines the address conversion that takes place. It specifies the information that is substituted into the packet for the original address.

Source address translations

SNAT rules substitute the translation address for the source address of a packet. Port translation is also available and can be specified as part of the translation address.

Note that the translation address must be set either to one of the addresses defined on the outbound interface or to `masquerade`, indicating that the primary IP address of the outbound interface is to be used as the translation address.

Substituting a source IP address

The following example shows how to apply a SNAT rule, rule 10, that substitutes the address of 12.34.56.78 as the source IP address of outbound packets that match its filter criteria.

```
vyatta@vyatta# set service nat source rule 10 translation address 12.34.56.78
```

Substituting a range of source IP addresses

The following example shows how to apply a SNAT rule, rule 20, that substitutes the addresses 12.34.56.64 through 12.34.56.79 as the range of source IP addresses for outbound packets that match its filter criteria.

```
vyatta@vyatta# set service nat source rule 20 translation address 12.34.56.64-12.34.56.79
```

Substituting the primary address of an outbound interface

The following example shows how to apply a SNAT rule, rule 30, that substitutes the primary address of the outbound interface as the source IP address of outbound packets that match its filter criteria.

```
vyatta@vyatta# set service nat source rule 30 translation address masquerade
```

Destination address translations

DNAT rules substitute the destination address of a packet with the `translation` address. Port translation is also available and can be specified as part of the translation address.

Substituting a destination IP address

The following example shows how to apply a DNAT rule, rule 40, that substitutes the address of 10.0.0.4 as the destination IP address of inbound packets that match its filter criteria.

```
vyatta@vyatta# set service nat destination rule 40 translation address 10.0.0.4
```

Substituting a range of destination IP addresses

The following example shows how to apply a DNAT rule, rule 50, that substitutes the addresses 10.0.0.0 through 10.0.0.3 as the range of destination IP addresses for inbound packets that match its filter criteria.

```
vyatta@vyatta# set service nat destination rule 50 translation address 10.0.0.0-10.0.0.3
```

Multiple Address Ranges for NAT

You can specify the name of a resource address group as the translation address for a NAT rule.

You can create a NAT rule with specific translation addresses, address ranges, or both. Each entry listed in the resource address group is used to create a set of mappings based on the port range.

The number of translation mappings is based on the number of addresses or ports. For ***address-group*** entries specified in CIDR format, for example, 2.2.2.0/24, the number of addresses is based on the network address and broadcast address. For example, a CIDR of 2.2.2.0/24 results in an address range of 2.2.2.1 to 2.2.2.254. A single address entry in the address group specifies a single address.

> **Note:** The address range for a CIDR entry does not include the broadcast address or network address.

For resource address groups specified for a NAT rule, the number of address mappings depends on the number and type of address group entries multiplied by the range of ports specified for the rule. If a port range is not specified for the rule, the default port range from 1 through 65535 is used. For example, a NAT rule that specifies a resource address group with two addresses and no port range results in 131,070 mappings.

For source NAT rules, the addresses specified in a resource group are used in ascending numerical order. The next address in the address group is referenced only when all the mappings implied by a resource address group entry have been consumed.

You can also dynamically add and delete address group entries, which takes effect immediately on the next NAT mapping allocations.

> **Note:** IPv6 addresses in resource address groups are ignored.

Session and packet logging

You can configure the vRouter for the following types of logging:

- Session logging. Configure stateful rules to log session state transitions.
- Per packet logging. Log every packet that matches a network packet filter rule, such as a firewall rule or NAT rule.

> **Note:** Per-packet logging generates large amounts of output and can negatively affect the performance of the entire system. Use per packet logging only for debugging purposes.

When logging is enabled, all log messages can be accessed by using the **`show dataplane log`** command.

Session Logging

A stateful firewall rule is created by adding the **state enabled** keywords to a firewall rule. By design, all NAT rules are stateful rules.

When a flow matches either a stateful firewall rule or a NAT rule, a session is created. The session tracks the state transitions of its IP protocol.

For UDP, ICMP, and all non-TCP flows, a session transitions to four states over the lifetime of the flow. For each transition, you can configure the product to log a message. TCP has a larger number of state transitions, each of which can be logged.

Use the **security firewall session-log** command to configure firewall session logging. When logging is configured, a log message is generated for each state transition.

Per packet logging for debugging

You can set up filtering rules so that each packet matched by the rule is logged.

AT&T recommends limiting per packet logging to debugging. Per packet logging occurs in the forwarding paths and can greatly reduce the throughput of the system and dramatically increase the disk space used for the log files. For all operational purposes, use stateful session logging instead of per packet logging.

To implement per packet logging for debugging purposes, you can include the **log** keyword when specifying a rule. When the logging option is specified, a log message containing the parameters of the packet is generated and logged.

NAT MIB Overview

The AT&T implementation of NAT supports the following Simple Management Network Protocol (SNMP) management information bases (MIBs): NAT-MIB, RFC4008, *Definitions of Managed Objects for Network Address Translators (NAT)*.

For a list of all supported MIBs, refer to AT&T Vyatta Network Operating System Remote Management Configuration Guide.

It is assumed for the NAT MIB that NAT is configured on a per-interface basis, with each interface explicitly labeled as internal or external. The MIB provides information on the NAT configuration and translated traffic. The following table describes some key terms that are used in the MIB and the vRouter that are equivalent to the MIB.

Table 1: NAT MIB Terminology

RFC 4008 Terminology	vRouter Terminology	RFC 4008 Definition
Address map	Rule	Per-interface statement consulted by NAT to determine the translation function to run, if any, whenever a session starts.
Binding	Translation	Description of a translation function run by NAT: • pre-NAT IP address or post-NAT IP address • transport protocol, pre-NAT IP address, pre-NAT port, post-NAT IP address and post-NAT port
Session	Session	Set of Traffic that is managed as a unit for translation: • TCP and UDP sessions are uniquely identified by source IP address, source TCP/UDP port, target IP address and target TCP/UDP port. • ICMP query sessions are identified by source IP address, ICMP query ID, target IP address. • All other sessions are characterized by source IP address, target IP address, protocol.

RFC 4008 Terminology	vRouter Terminology	RFC 4008 Definition
NAT session	N/A	Association between the pre-NAT and post-NAT versions of the same session.

Using the NAT MIB on the AT&T Vyatta vRouter

The NAT MIB implementation on the AT&T Vyatta vRouter is a read-only implementation. Consequently SNMP SET commands and SNMP traps are not supported.

Philosphical differences between the NAT MIB and the AT&T Vyatta vRouter NAT architecture mean that you must consider the following facts when interpreting the MIB contents.

- MIB objects referring to a local, internal, or private interface are derived from pre-NAT data for SNAT rules and post-NAT data for DNAT rules.
- MIB objects referring to a global, external, or public interface are derived from post-NAT data for SNAT rules and pre-NAT data for DNAT rules.
- MIB objects referring to inbound NAT traffic are derived from DNAT rules only.
- MIB objects referring to outbound NAT traffic are derived from SNAT rules only.

NAT6-4, the use of port-groups or address-groups, and the masquerade NAT translation address are currently not supported in the MIB. Furthermore, the following MIB objects are currently not supported on the AT&T Vyatta vRouter.

- natNotifThrottlingInterval (1.3.6.1.2.1.123.1.2.1)
- natInterfaceDiscards (1.3.6.1.2.1.123.1.3.1.5)
- natAddrMapDiscards (1.3.6.1.2.1.123.1.4.1.18)
- natAddrBindMapIndex (1.3.6.1.2.1.123.1.6.1.8)
- natAddrBindSessions (1.3.6.1.2.1.123.1.6.1.9)
- natAddrBindInTranslates (1.3.6.1.2.1.123.1.6.1.12)
- natAddrBindOutTranslates (1.3.6.1.2.1.123.1.6.1.13)
- natAddrPortBindMapIndex (1.3.6.1.2.1.123.1.8.1.11)
- natAddrPortBindSessions (1.3.6.1.2.1.123.1.8.1.12)
- natAddrPortBindInTranslates (1.3.6.1.2.1.123.1.8.1.15)
- natAddrPortBindOutTranslates (1.3.6.1.2.1.123.1.8.1.16)
- natSessionUpTime (1.3.6.1.2.1.123.1.9.1.7)
- natSessionAddrMapIndex (1.3.6.1.2.1.123.1.9.1.8)
- natSessionInTranslates (1.3.6.1.2.1.123.1.9.1.22)
- natSessionOutTranslates (1.3.6.1.2.1.123.1.9.1.23)
- natProtocolDiscards (1.3.6.1.2.1.123.1.10.1.4)

NAT Configuration Examples

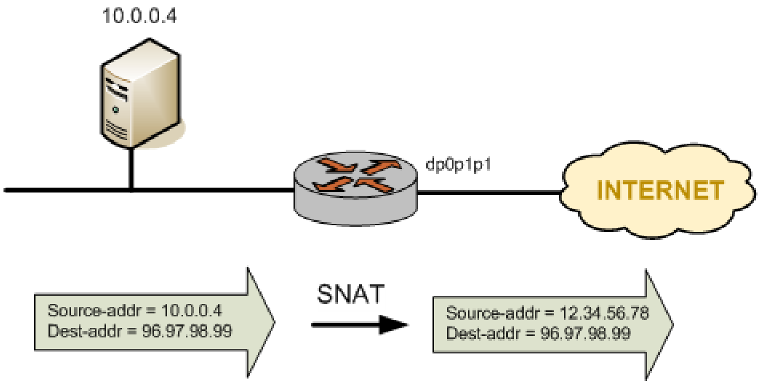
> **Note:** Each NAT rule in these examples could be independently deployed on a system. These examples are not intended to be deployed together. For that reason, all rules in the examples are given the same rule number (rule 10).

Source NAT (one-to-one)

The following figure shows an example of source NAT (SNAT) in which a single "inside" source address is translated to a single "outside" source address. This example has the following characteristics:

- An internal news server, a Network News Time Protocol (NTTP) device, needs to connect to an external news server.
- The external news server accepts connections only from known clients.
- The internal news server does not receive connections from outside the local network.

Figure 17: Source NAT (one-to-one)



To configure NAT in this way, perform the following steps in configuration mode.

Table 2: Configuring source NAT (one-to-one)

Step	Command
Create SNAT rule 10.	`vyatta@vyatta# set service nat source rule 10`

Apply this rule to packets coming from the 10.0.0.4 address and egressing through the dp0p1p1 interface.	`vyatta@vyatta# set service nat source rule 10 source address 10.0.0.4`
	`vyatta@vyatta# set service nat source rule 10 outbound-interface dp0p1p1`

Use 12.34.56.78 as the source address in outgoing packets. Make sure that the translation address is an address defined on the outbound interface if it is part of the connected subnet on that interface. This ensures that the AT&T Vyatta vRouter replies to ARP requests from remote devices for the translation address.	`vyatta@vyatta# set service nat source rule 10 translation address 12.34.56.78`

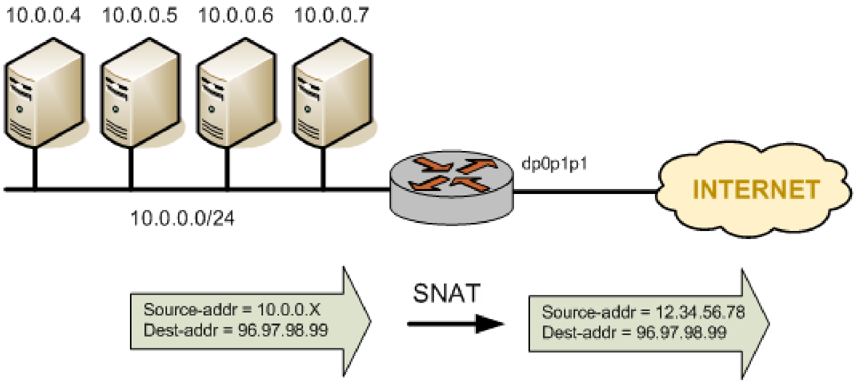
Step	Command
Commit the change.	`vyatta@vyatta# commit`

Show the configuration.	`vyatta@vyatta# show nat source rule 10`
	`outbound-interface dp0p1p1`
	`source {`
	`    address 10.0.0.4`
	`}`
	`translation {`
	`    address 12.34.56.78`
	`}`

Source NAT (many-to-one)

The following figure shows an example of SNAT in which many different “inside” addresses are dynamically translated to a single “outside” address. In this example, all hosts on the 10.0.0.0/24 subnet show the same source address externally.

Figure 18: Source NAT (many-to-one)



To configure NAT in this way, perform the following steps in configuration mode.

Table 3: Configuring source NAT (many-to-one)

Step	Command
Create SNAT rule 10.	`vyatta@vyatta# set service nat source rule 10`

Apply this rule to packets coming from the 10.0.0.0/24 network and egressing through the eht0 interface.	`vyatta@vyatta# set service nat source rule 10`
	` source address 10.0.0.0/24`
	`vyatta@vyatta# set service nat source rule 10`
	` outbound-interface dp0p1p1`

Use 12.34.56.78 as the source address in outgoing packets. Make sure that the translation address is an address defined on the outbound interface if it is part of the connected subnet on that interface. This ensures that the AT&T Vyatta vRouter replies to ARP requests from remote devices for the translation address.	`vyatta@vyatta# set service nat source rule 10`
	` translation address 12.34.56.78`

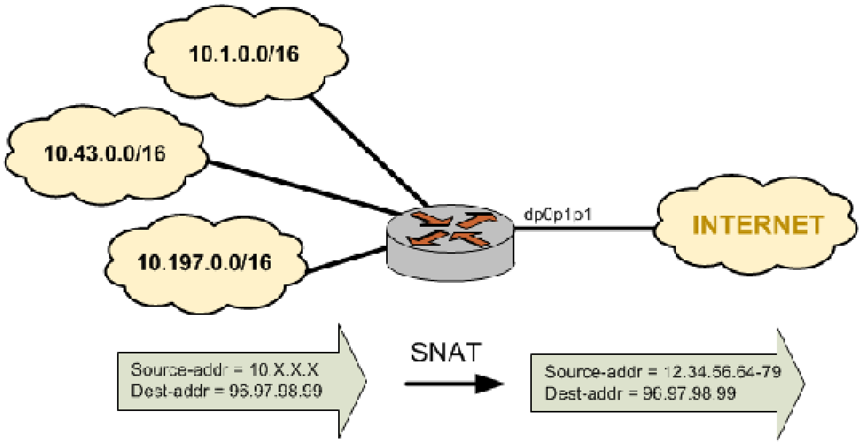
Step	Command
Commit the change.	`vyatta@vyatta# commit`

Show the configuration.	`vyatta@vyatta# show nat source rule 10`
	`outbound-interface dp0p1p1`
	`source {`
	`    address 10.0.0.0/24`
	`}`
	`translation {`
	`    address 12.34.56.78`
	`}`

Source NAT (many-to-many)

In many-to-many translations, a number of private addresses are mapped to a number of public addresses. This mapping provides a way of reducing the possibility of port exhaustions that are possible in a many-to-one scenario. For this reason, the mapping can provide more capacity for outbound translations. The following figure shows a large private address space (a /8 network prefix, here represented as three /16 subnets) mapped to a small range of external addresses.

Figure 19: Source NAT (many-to-many)



To configure NAT in this way, perform the following steps in configuration mode.

Table 4: Configuring source NAT (many-to-many)

Step	Command
Create SNAT rule 10.	`vyatta@vyatta# set service nat source rule 10`

Apply this rule to packets coming from any host on the 10.0.0.0./8 network and egressing through the dp0p1p1 interface.	`vyatta@vyatta# set service nat source rule 10`
	` source address 10.0.0.0/8`
	`vyatta@vyatta# set service nat source rule 10`
	` outbound-interface dp0p1p1`

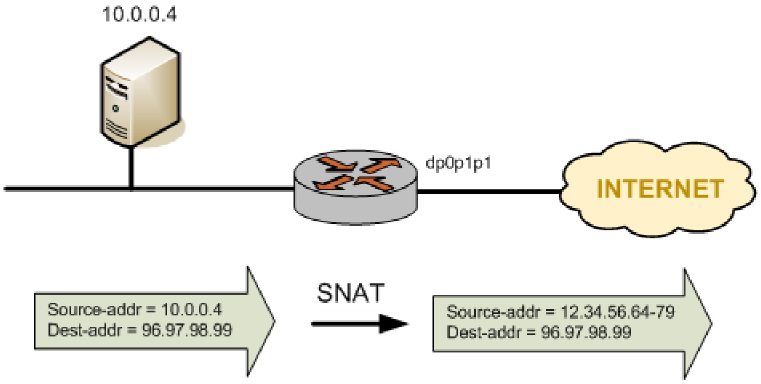
Step	Command
Choose an address in the range 12.34.56.64 through 12.34.56.79 as the source address in outgoing packets. Note that the translation address should be an address defined on the outbound interface if it is part of the connected subnet on that interface. This ensures that the AT&T Vyatta vRouter replies to ARP requests from remote devices for one of the translation addresses.	`vyatta@vyatta# set service nat source rule 10 translation address 12.34.56.64-12.34.56.79`
Commit the change.	`vyatta@vyatta# commit`

Show the configuration.	`vyatta@vyatta# show nat source rule 10`
	`outbound-interface dp0p1p1`
	`source {`
	`    address 10.0.0.0/8`
	`}`
	`translation {`
	`    address 12.34.56.64-12.34.56.79`
	`}`

Source NAT (one-to-many)

The scenario described in this section is less common. In this scenario, a single test-source device behind the NAT device appears externally to be multiple devices, as shown in the following figure. One application of this scenario might be to test an upstream load-balancing device.

Figure 20: Source NAT (one-to-many)



To configure NAT in this way, perform the following steps in configuration mode.

Table 5: Configuring source NAT (one-to-many)

Step	Command
Create SNAT rule 10.	`vyatta@vyatta# set service nat source rule 10`

Step	Command

Apply this rule to packets coming from the 10.0.0.4 address and egressing through the dp0p1p1 interface.	`vyatta@vyatta# set service nat source rule 10 source address 10.0.0.4`
	`vyatta@vyatta# set service nat source rule 10 outbound-interface dp0p1p1`

Choose an address in the range 12.34.56.64 through 12.34.56.79 as the source address in outgoing packets. Note that the translation address should be an address defined on the outbound interface if it is part of the connected subnet on that interface. This ensures that the AT&T Vyatta vRouter replies to ARP requests from remote devices for one of the translation addresses.	`vyatta@vyatta# set service nat source rule 10 translation address 12.34.56.64-12.34.56.79`
Commit the change.	`vyatta@vyatta# commit`

Show the configuration.	`vyatta@vyatta# show nat source rule 10`
	`outbound-interface dp0p1p1`
	`source {`
	`address 10.0.0.4`
	`}`
	`translation {`
	`address 12.34.56.64-12.34.56.79`
	`}`

Masquerade NAT

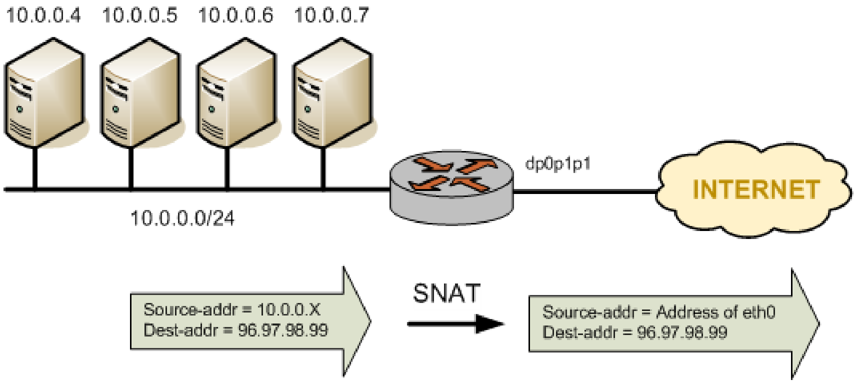
Masquerade NAT is a special application of source NAT. It is typically used when the Internet-facing interface has a dynamic IP address provided by a mechanism such as DHCP. In these cases, configuring a static translation address is not appropriate as the address assigned to the interface can change. Specifying **masquerade** as the translation address instructs the system to use the IP address currently assigned to the outbound interface as the translation address.

Masquerade NAT rules typically consist of match conditions that contain the following characteristics:

- The source network (usually the private IP network assigned to LAN devices)
- The outbound interface (the Internet-facing interface that is assigned the dynamic IP address)

The following figure shows an example of masquerade NAT.

Figure 21: Masquerade NAT



To configure NAT in this way, perform the following steps in configuration mode.

Table 6: Configuring masquerade NAT

Step	Command
Create SNAT rule 10.	`vyatta@vyatta# set service nat source rule 10`

Apply this rule to packets coming from any host on the 10.0.0.0/24 network and egressing through the dp0p1p1 interface.	`vyatta@vyatta# set service nat source rule 10`
	`source address 10.0.0.0/24`
	`vyatta@vyatta# set service nat source rule 10`
	`outbound-interface dp0p1p1`

Use the IP address of the outbound interface as the outside address.	`vyatta@vyatta# set service nat source rule 10`
	`translation address masquerade`

Commit the change.	`vyatta@vyatta# commit`

Show the configuration.	`vyatta@vyatta# show nat source rule 10`
	`outbound-interface dp0p1p1`
	`source {`
	`    address 10.0.0.0/24`
	`}`
	`translation {`
	`    address masquerade`
	`}`

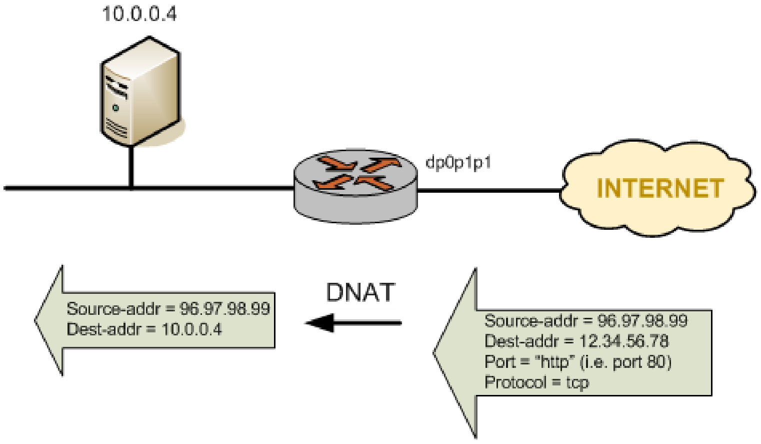
Destination NAT (one-to-one)

Destination NAT (DNAT) is used when only inbound traffic is expected.

Scenario 1: Packets destined for an internal web server

For example, DNAT might be used in a scenario in which a corporate web server needs to be reachable from external locations but never initiates outbound sessions, as shown in the following figure.

Figure 22: Destination NAT (one-to-one)



To configure NAT in this way, perform the following steps in configuration mode.

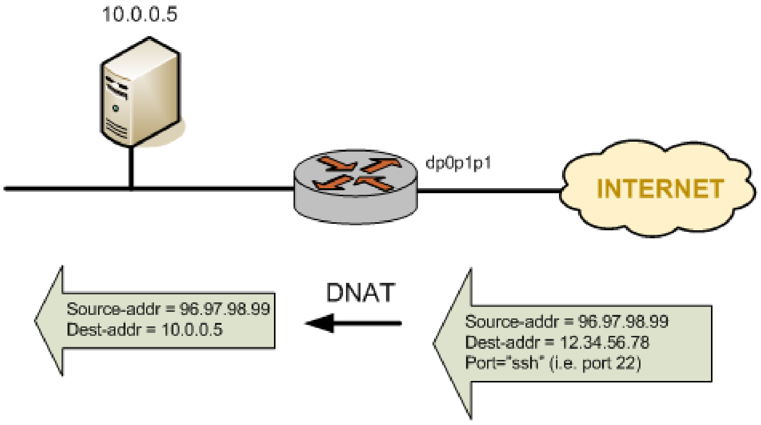
Table 7: Configuring destination NAT (one-to-one)

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>Create DNAT rule 10.</td><td><pre>vyatta@vyatta# set service nat destination
 rule 10</pre></td></tr>
<tr><td>Apply this rule to all incoming TCP packets on the dp0p1p1 interface bound for the 12.34.56.78 address on the HTTP port.</td><td><pre>vyatta@vyatta# set service nat destination
 rule 10 inbound-interface dp0p1p1
vyatta@vyatta# set service nat destination
 rule 10 destination address 12.34.56.78
vyatta@vyatta# set service nat destination
 rule 10 destination port http
vyatta@vyatta# set service nat destination
 rule 10 protocol tcp</pre></td></tr>
<tr><td>Forward traffic to the 10.0.0.4 address.</td><td><pre>vyatta@vyatta# set service nat destination
 rule 10 translation address 10.0.0.4</pre></td></tr>
<tr><td>Commit the change.</td><td><pre>vyatta@vyatta# commit</pre></td></tr>
<tr><td>Show the configuration.</td><td><pre>vyatta@vyatta# show nat destination rule 10
 destination {
     address 12.34.56.78
     port http
 }
 inbound-interface dp0p1p1
 protocols tcp
 translation {
     address 10.0.0.4
 }</pre></td></tr>
</table>

Scenario 2: Packets destined for an internal SSH server

In this scenario, all traffic destined for the SSH port is passed through to a host containing an SSH server, as shown in the following figure.

Figure 23: Destination NAT (one-to-one): filtering on port name


10.0.0.5
dp0p1p1
INTERNET
DNAT
Source-addr = 96.97.98.99
Dest-addr = 10.0.0.5
Source-addr = 96.97.98.99
Dest-addr = 12.34.56.78
Port="ssh" (i.e. port 22)

To configure NAT in this way, perform the following steps in configuration mode.

Table 8: Configuring destination NAT (one-to-one): filtering port name

Step	Command
Create DNAT rule 10.	`vyatta@vyatta# set service nat destination rule 10`

Apply this rule to all incoming packets on the dp0p1p1 interface bound for the 12.34.56.78 address on the SSH port.	`vyatta@vyatta# set service nat destination rule 10 inbound-interface dp0p1p1`
	`vyatta@vyatta# set service nat destination rule 10 protocol tcp`
	`vyatta@vyatta# set service nat destination rule 10 destination address 12.34.56.78`
	`vyatta@vyatta# set service nat destination rule 10 destination port ssh`

Forward traffic to the 10.0.0.5 address.	`vyatta@vyatta# set service nat destination rule 10 translation address 10.0.0.5`
Commit the change.	`vyatta@vyatta# commit`

Show the configuration.	`vyatta@vyatta# show nat destination rule 10`
	`destination {`
	`    address 12.34.56.78`
	`    port ssh`
	`}`
	`inbound-interface dp0p1p1`
	`protocol tcp`
	`translation {`
	`    address 10.0.0.5`
	`}`

Destination NAT (one-to-many)

Another application where DNAT might be used is a scenario in which there are multiple instances (each on a different port) of the server inside a private network. To configure NAT for this particular scenario, perform the following steps in configuration mode.

Table 9: Configuring destination NAT (one-to-many)

Step	Command
Create DNAT rule 10.	`vyatta@vyatta# set service nat destination rule 10`

Apply this rule to all incoming packets on the dp0p1p1 interface bound for the 12.34.56.78 address on a well know http port.	`vyatta@vyatta# set service nat destination rule 10 inbound-interface dp0p1p1`
	`vyatta@vyatta# set service nat destination rule 10 destination port http`
	`vyatta@vyatta# set service nat destination rule 10 destination address 12.34.56.78`
	`vyatta@vyatta# set service nat destination rule 10 protocol tcp`

Step	Command

Forward traffic to internal host address 10.0.0.64 across ports 2000-2019 and across 20 instances in this case.	`vyatta@vyatta# set service nat destination`
	` rule 10 translation address 10.0.0.64`
	`vyatta@vyatta#  set service nat destination`
	` rule 10 translation port 2000-2019`

	`vyatta@vyatta# set service nat destination`
	` rule 10 inbound-interface dp0p192p1`

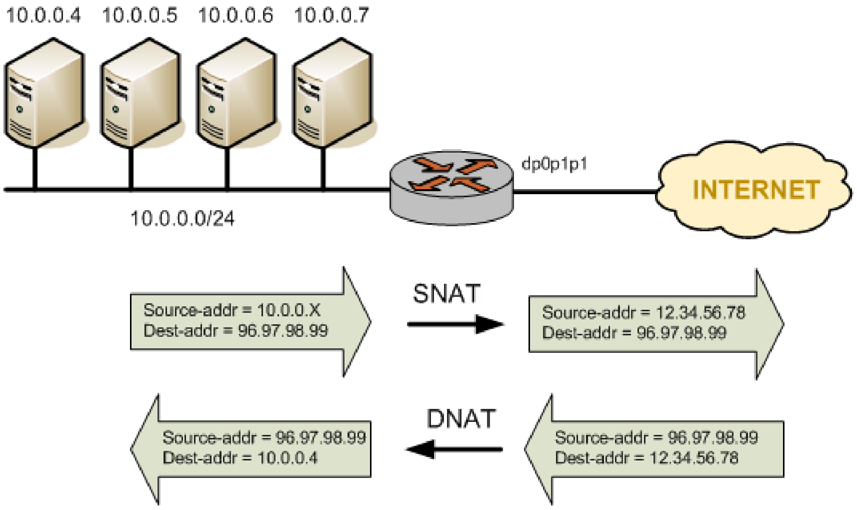
Commit the change.	`vyatta@vyatta# commit`

Show the configuration.	`vyatta@vyatta# show nat destination rule 10`
	` destination {`
	`     address 12.34.56.78`
	`     port http`
	` }`
	` inbound-interface dp0p1p1`
	` protocol tcp`
	` translation {`
	`     address 10.0.0.64`
	`     port 2000-2019`
	` }`

Bidirectional NAT

Bidirectional NAT is simply a combination of source and destination NAT. A typical scenario might use SNAT on the outbound traffic of an entire private network and DNAT for specific internal services (for example, mail or web); refer to the following figure.

Figure 24: Bidirectional NAT



To configure NAT in this way, perform the following steps in configuration mode. Note that source and destination rule numbers are independent. In the example, this independence is highlighted by creating “source rule 10” and “destination rule 10.”

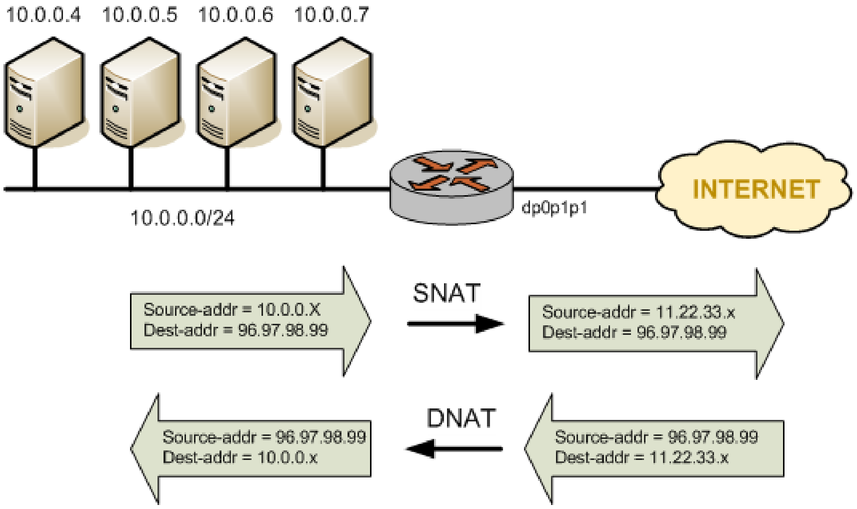
Table 10: Configuring bidirectional NAT

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>Create SNAT rule 10.</td><td><pre>vyatta@vyatta# set service nat source rule 10</pre></td></tr>
<tr><td>Apply this rule to packets coming from any host on the 10.0.0.0/24 network and egressing through the dp0p1p1 interface.</td><td><pre>vyatta@vyatta# set service nat source rule 10
 source address 10.0.0.0/24
vyatta@vyatta# set service nat source rule 10
 outbound-interface dp0p1p1</pre></td></tr>
<tr><td>Use 12.34.56.78 as the source address in outgoing packets.</td><td><pre>vyatta@vyatta# set service nat source rule 10
 translation address 12.34.56.78</pre></td></tr>
<tr><td>Create DNAT rule 10.</td><td><pre>vyatta@vyatta# set service nat destination
 rule 10</pre></td></tr>
<tr><td>Apply this rule to all incoming TCP packets on the dp0p1p1 interface bound for the 12.34.56.78 address, port 80 (that is, HTTP traffic).</td><td><pre>vyatta@vyatta# set service nat destination
 rule 10 inbound-interface dp0p1p1
vyatta@vyatta# set service nat destination
 rule 10 destination address 12.34.56.78
vyatta@vyatta# set service nat destination
 rule 10 destination port 80
vyatta@vyatta# set service nat destination
 rule 10 protocol tcp</pre></td></tr>
<tr><td>Forward traffic to the 10.0.0.4 address (that is, the web server).</td><td><pre>vyatta@vyatta# set service nat destination
 rule 10 translation address 10.0.0.4</pre></td></tr>
<tr><td>Commit the change.</td><td><pre>vyatta@vyatta# commit</pre></td></tr>
<tr><td>Show the configuration.</td><td><pre>vyatta@vyatta# show nat source rule 10
 outbound-interface dp0p1p1
 source {
     address 10.0.0.0/24
 }
 translation {
     address 12.34.56.78
 }
vyatta@vyatta# show nat destination rule 10
 destination {
     address 12.34.56.78
     port 80
 }
 inbound-interface dp0p1p1
 protocol tcp
 translation {
     address 10.0.0.4
 }</pre></td></tr>
</table>

Mapping of address ranges

The AT&T Vyatta vRouter supports the mapping of an entire network of addresses to another network of addresses. This mapping means that you do not have to manually enter many NAT rules. For example, you can map the 10.0.0.0/24 network to the 11.22.33.0/24 network, which maps 10.0.0.1 through 11.22.33.1, 10.0.0.2 through 11.22.33.2, and so on. The networks must be the same size, that is, they must have the same network mask, as shown in the following figure.

Figure 25: Mapping of address ranges



To configure NAT in this way, perform the following steps in configuration mode.

Table 11: Mapping address ranges

Step	Command
Create SNAT rule 10.	`vyatta@vyatta# set service nat source rule 10`

Apply this rule to packets coming from any host on the 10.0.0.0/24 network and egressing through the dp0p1p1 interface.	`vyatta@vyatta# set service nat source rule 10 source address 10.0.0.0/24`
	`vyatta@vyatta# set service nat source rule 10 outbound-interface dp0p1p1`

Use 11.22.33.*x* as the source address in outgoing packets.	`vyatta@vyatta# set service nat source rule 10 translation address 11.22.33.0/24`
Create destination (DNAT) rule 10.	`vyatta@vyatta# set service nat destination rule 10`

Apply this rule to packets destined for any host on the 11.22.33.0/24 network and ingressing through the dp0p1p1 interface.	`vyatta@vyatta# set service nat destination rule 10 destination address 11.22.33.0/24`
	`vyatta@vyatta# set service nat destination rule 10 inbound-interface dp0p1p1`

Use 10.0.0.*x* as the destination address in incoming packets.	`vyatta@vyatta# set service nat destination rule 10 translation address 10.0.0.0/24`
Commit the change.	`vyatta@vyatta# commit`

Step	Command

Show the configuration.	`vyatta@vyatta# show nat source rule 10`
	` outbound-interface dp0p1p1`
	` source {`
	`     address 10.0.0.0/24`
	` }`
	` translation {`
	`     address 11.22.33.0/24`
	` }`
	`vyatta@vyatta# show nat destination rule 10`
	` destination {`
	`     address 11.22.33.0/24`
	` }`
	` inbound-interface dp0p1p1`
	` translation {`
	`     address 10.0.0.0/24`
	` }`

The "exclude" option

Sometimes it is desirable to exclude packets from NAT that match certain criteria. This exclusion can be accomplished by using the **`exclude`** option.

The following example shows how to use the **`exclude`** option to exclude a subset of traffic (packets coming from 192.168.0.0/24 and destined for 172.16.50.0/24 through the dp0p1p1 interface from translation. Note that rule 10 excludes certain traffic from translation and rule 20 performs a translation on the traffic that meets its filter criteria and is not excluded by rule 10.

Table 12: Excluding packets from NAT by using the exclude option

Step	Command
Create SNAT rule 10.	`vyatta@vyatta# set service nat source rule 10`

Apply this rule to packets coming from any host on the 192.168.0.0/24 network, going to the 172.16.50.0/24 network, and egressing through the dp0p1p1 interface.	`vyatta@vyatta# set service nat source rule 10`
	` source address 192.168.0.0/24`
	`vyatta@vyatta# set service nat source rule 10`
	` destination address 172.16.50.0/24`
	`vyatta@vyatta# set service nat source rule 10`
	` outbound-interface dp0p1p1`

Exclude packets from NAT that match the filter criteria in this rule.	`vyatta@vyatta# set service nat source rule 10`
	` exclude`

Create SNAT rule 20.	`vyatta@vyatta# set service nat source rule 20`

Apply this rule to packets coming from any host on the 192.168.0.0/24 network and egressing through the dp0p1p1 interface.	`vyatta@vyatta# set service nat source rule 20`
	` source address 192.168.0.0/24`
	`vyatta@vyatta# set service nat source rule 20`
	` outbound-interface dp0p1p1`

Use the primary IP address of the outbound interface as the translation address.	`vyatta@vyatta# set service nat source rule 20`
	` translation address masquerade`

Step	Command
Commit the change.	`vyatta@vyatta# commit`

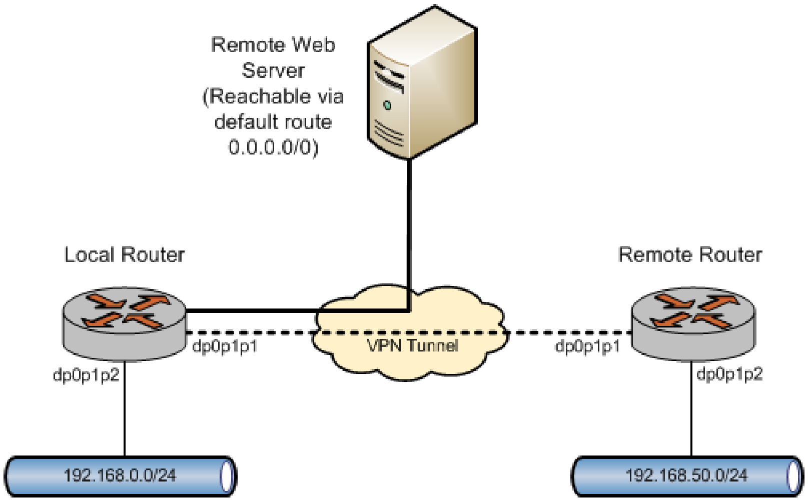
Show the configuration.	`vyatta@vyatta# show nat source`
	` rule 10 {`
	`     destination {`
	`         address 172.16.50.0/24`
	`     }`
	`     exclude`
	`     outbound-interface dp0p1p1`
	`     source {`
	`         address 192.168.0.0/24`
	`     }`
	` }`
	` rule 20 {`
	`     outbound-interface dp0p1p1`
	`     source {`
	`         address 192.168.0.0/24`
	`     }`
	`     translation {`
	`         address masquerade`
	`     }`
	` }`

Source NAT and VPN: using the "exclude" option

When a packet is matched against the source NAT (including masquerade NAT) filter criteria, the source address of the packet is modified before it is forwarded to its destination. This means that source NAT rules are applied before the VPN process compares the packets against the VPN configuration. If the source network that is configured for source NAT is also configured to use a site-to-site VPN connection using the same externally facing interface, the packets are not recognized by the VPN process because the source address has been changed. Consequently, they are not placed into the VPN tunnel for transport.

To account for this behavior, packets destined for a VPN tunnel must be excluded from having NAT applied. You can do this by using an exclusion rule, as shown in the following figure.

Figure 26: Source NAT and VPN



To configure NAT in this way, perform the following steps in configuration mode.

Table 13: Configuring masquerade NAT to bypass a VPN tunnel

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>Create SNAT rule 10.</td><td><pre>vyatta@vyatta# set service nat source rule 10</pre></td></tr>
<tr><td>Apply this rule to packets coming from any host on the 192.168.0.0/24 network, going to the 192.168.50.0/24 network, and egressing through the dp0p1p1 interface.</td><td><pre>vyatta@vyatta# set service nat source rule 10
 source address 192.168.0.0/24
vyatta@vyatta# set service nat source rule 10
 destination address 192.168.50.0/24
vyatta@vyatta# set service nat source rule 10
 outbound-interface dp0p1p1</pre></td></tr>
<tr><td>Exclude packets from NAT translation that match the filter criteria in this rule.</td><td><pre>vyatta@vyatta# set service nat source rule 10
 exclude</pre></td></tr>
<tr><td>Create SNAT rule 20.</td><td><pre>vyatta@vyatta# set service nat source rule 20</pre></td></tr>
<tr><td>Apply this rule to packets coming from any host on the 192.168.0.0/24 network and egressing through the dp0p1p1 interface.</td><td><pre>vyatta@vyatta# set service nat source rule 20
 source address 192.168.0.0/24
vyatta@vyatta# set service nat source rule 20
 outbound-interface dp0p1p1</pre></td></tr>
<tr><td>Use the primary IP address of the outbound interface as the translation address.</td><td><pre>vyatta@vyatta# set service nat source rule 20
 translation address masquerade</pre></td></tr>
<tr><td>Commit the change.</td><td><pre>vyatta@vyatta# commit</pre></td></tr>
<tr><td>Show the configuration.</td><td><pre>vyatta@vyatta# show nat source
 rule 10 {
     destination {
         address 192.168.50.0/24
     }
     exclude
     outbound-interface dp0p1p1
     source {
     address 192.168.0.0/24
     }
 }
 rule 20 {
     outbound-interface dp0p1p1
     source {
         address 192.168.0.0/24
     }
     translation {
         address masquerade
     }
 }</pre></td></tr>
</table>

The negation operator

Another way to exclude a subset of traffic from being translated is by using the negation operator (exclamation mark [!]). The following example shows how to provide the same functionality as in the previous example but use the negation operator instead of the **`exclude`** option.

Note: You can use the negation operator with IP addresses but not with port addresses.

Table 14: Configuring masquerade NAT to exclude a subset of traffic by using the negation operator

Step	Command
Create SNAT rule 10.	`vyatta@vyatta# set service nat source rule 10`

Apply this rule to packets coming from any host on the 192.168.0.0/24 network, not going to the 192.168.50.0/24 network, and egressing through the dp0p1p1 interface.	`vyatta@vyatta# set service nat source rule 10 source address 192.168.0.0/24`
	`vyatta@vyatta# set service nat source rule 10 destination address !192.168.50.0/24`
	`vyatta@vyatta# set service nat source rule 10 outbound-interface dp0p1p1`

Use the primary IP address of the outbound interface as the translation address.	`vyatta@vyatta# set service nat source rule 10 translation address masquerade`
Commit the change.	`vyatta@vyatta# commit`

Show the configuration.	`vyatta@vyatta# show nat source`
	`rule 10 {`
	`    destination {`
	`        address !192.168.50.0/24`
	`    }`
	`    outbound-interface dp0p1p1`
	`    source {`
	`        address 192.168.0.0/24`
	`    }`
	`    translation {`
	`        address masquerade`
	`    }`
	`}`

Note that you should take extreme care using when combining more than one negation operator rule. NAT rules are evaluated sequentially, and a sequence of rules that use the negation operator may result in unexpected behavior.

Consider the set of two NAT rules shown in the following example.

Multiple source NAT rules that use the negation operator: unexpected behavior

```
rule 10 {
    destination {
        address !192.168.50.0/24
    }
    outbound-interface dp0p1p1
    source {
        address 192.168.0.0/24
    }
    translation {
    address masquerade
    }
}
rule 20 {
    destination {
        address !172.16.50.0/24
    }
    outbound-interface dp0p1p1
    source {
        address 192.168.0.0/24
    }
    translation {
        address masquerade
    }
}
```

This combination of rules does not exclude the 192.168.50.0/24 and 172.16.50.0/24 networks. As previously explained, these NAT rules are evaluated sequentially; when a packet arrives, it is tested against the first rule and if it does not match, it is tested against the second rule, and so on until it matches a rule.

In the example, a packet with a destination in 192.168.50.0/24 does not meet the match criteria in rule 10, which matches all packets with a destination not in 192.168.50.0/24. As a result, the packet “falls through” to rule 20. A packet with a destination in 192.168.50.0/24 does match rule 20 because it is not in 172.16.50.0/24; therefore, the packet has NAT applied, which is not the desired result.

Similarly, a packet with a destination in 172.16.50.0/24 is matched and has NAT applied by rule 10.

Address and port groups

The following example shows how to configure address groups and applying NAT rules to them.

Table 15: Configuring address groups and applying NAT rules

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>Configure address and port to join a group named foo.</td><td>

```
vyatta@vyatta# set resources group port-group
 bar port 1
vyatta@vyatta# set resources group port-group
 bar port 121
vyatta@vyatta# commit
vyatta@vyatta# show resources
resources {
        group {
                address-group foo {
                        address 1.1.1.0/24
                        address 2.2.0.0/16
                        address 12.32.223.3
                }
                port-group bar {
                        port 1
                        port 121
                                }
        }
}
```

</td></tr>
<tr><td>Create a source NAT rule.</td><td>

```
vyatta@vyatta# set service nat source rule
 200 source address foo
vyatta@vyatta# set service nat source rule
 200 source port bar
vyatta@vyatta# set service nat source rule
 200 protocol tcp
vyatta@vyatta# set service nat source rule
 200 translation address 20.20.10.0/24
vyatta@vyatta# set service nat source rule
 200 translation port http
vyatta@vyatta# set service nat source rule
 200 outbound-interface dp0s224
```

</td></tr>
</table>

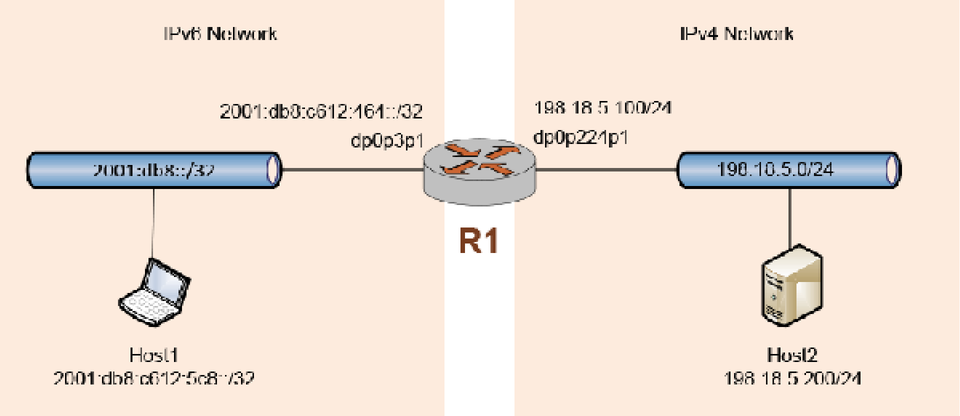
Step	Command
Commit the changes.	`vyatta@vyatta# commit`

Show the NAT configuration.	
	```
	vyatta@vyatta# show service nat source rule
	 200
	outbound-interface dp0s224
	protocol tcp
	source {
	        address foo
	        port bar
	}
	translation {
	        address 20.20.10.0/24
	        port http
	}
	```

Configuring NAT 6-4

The following figure shows a NAT 6-4 configuration example. In this example, Host1, a host that resides in an external IPv6 network, sends requests to Host2, a host that resides in an internal IPv4 network. The request enters the router through the dp0p3p1 data plane interface for which NAT 6-4 translation is enabled.

Figure 27: NAT 6-4 configuration example



To configure NAT 6-4 as shown in this figure, perform the following steps in configuration mode. NAT 6-4 configuration involves the following steps:

- Creating a NAT 6-4 rule.
- Specifying the IPv6 routing prefix of the destination IPv4 addresses.
- Specifying the data plane interface through which the inbound IPv6 request packets pass.
- Specifying the IPv6 routing prefix of the source IPv6 addresses.

Table 16: Configuring NAT 6-4

Step	Command

On R1, specify 1 as the IPv6-to-IPv4 NAT rule and specify 2001:db9::/32 as the routing prefix for destination addresses.	
	```
	vyatta@R1# set service nat ipv6-to-ipv4 rule
	 1 destination prefix 2001:db9::/32
	```

Step	Command

For rule 1, specify the inbound interface.	`vyatta@R1# set service nat ipv6-to-ipv4 rule`
	` 1 inbound-interface dp0p3p1`

For rule 1, specify the routing prefix for source addresses.	`vyatta@R1# set service nat ipv6-to-ipv4 rule`
	` 1 source prefix 2001:db8::/32`

Run the `show service nat` command.	`vyatta@R1# show service nat`
	`nat {`
	`  ipv6-to-ipv4 {`
	`    rule 100 {`
	`      destination {`
	`        prefix 2001:db9::/32`
	`      }`
	`    inbound-interface dp0p3p1`
	`    source {`
	`       prefix 2001:db8::/32`
	`    }`
	`   }`
	`  }`
	`}`

To verify that your NAT 6-4 setup works, ping Host2 from Host1.	`vyatta@host1# run ping 2001:db9:c612:05c8::`
	`PING 2001:db9:c612:05c8::`
	`(2001:db9:c612:5c8::) 56 data bytes`
	`64 bytes from 2001:db9:c612:5c8::: icmp_seq=1`
	` ttl=63 time=0.950 ms`

On Host2, run the following command to capture the ping traffic on the eth1 interface.

```
vyatta@host2:~$ sudo tshark -i eth1
Running as user "root" and group "root". This could be dangerous.
Capturing on eth1
 0.000000 198.18.4.200 -> 198.18.5.200 ICMP Echo (ping) request
 0.000025 198.18.5.200 -> 198.18.4.200 ICMP Echo (ping) reply
```

For a TCP-based flow (SSH from Host1 to Host2), run the following command:

```
vyatta@host1# ssh vyatta@2001:db9:c612:05c8::

Welcome to Vyatta
vyatta@2001:db9:c612:05c8::'s password:
Welcome to Vyatta
Version: 999.daisyse.12170009
Description: 999.daisyse.12170009
Copyright: 2006-2013 Vyatta, Inc.
Last login: Wed Sep 24 23:07:35 2014 from 192.168.122.1
vyatta@host2:~$
```

On Host2, run the following command to capture the SSH traffic on the eth1 interface.

```
vyatta@host2:~$ sudo tshark -i eth1
Running as user "root" and group "root". This could be dangerous.
Capturing on eth1
 73.000922 198.18.4.200 -> 198.18.5.200 TCP 46468 > ssh [SYN] Seq=0
Win=14400 Len=0 MSS=1440 TSV=2698800 TSER=0 WS=7
 73.000959 198.18.5.200 -> 198.18.4.200 TCP ssh > 46468 [SYN, ACK] Seq=0
Ack=1 Win=14480 Len=0 MSS=1460 TSV=2698617 TSER=2698800 WS=7
 73.002098 198.18.4.200 -> 198.18.5.200 TCP 46468 > ssh [ACK] Seq=1 Ack=1
```

```
Win=14464 Len=0 TSV=2698800 TSER=2698617
 73.006947 198.18.5.200 -> 198.18.4.200 SSH Server Protocol:
SSH-2.0-OpenSSH_5.5p1 Debian-6+squeeze3\r
```

Both flows create sessions on DUT, as shown in the following example.

```
vyatta@vyatta# run show session-table
TCP state codes: SS - SYN SENT, SR - SYN RECEIVED, ES - ESTABLISHED,
                 FW - FIN WAIT, CW - CLOSE WAIT, LA - LAST ACK,
                 TW - TIME WAIT, CL - CLOSE, LI - LISTEN

CONNID Source               Destination      Protocol  TIMEOUTIntf   Parent
5      198.18.4.200:46468   198.18.5.200:22  tcp [6]   TW 237        dp0p4p10
6      198.18.4.200         198.18.5.200     icmp [1]  28            dp0p4p10
```

NAT Commands

clear nat

Clears counters for active NAT rules.

Syntax:
`clear nat`

Counters for all NAT rules.

Operational mode

Use this command to clear counters for active NAT rules.

resources group address-group

Specifies the name of an address resource group as the translation address for a NAT rule.

Syntax:
`set resources group address-group` *address-group-name* **`address`** *member-address*

Syntax:
`delete resources group address-group` *address-group-name* **`address`** *member-address*

address-group-name
The name of a resource address group

member-address
Member address

Configuration mode

```
resources {
  group {
   address-group address-group-name {
    address member-address
    address member-address
    address member-address
   }
  }
}
```

Use the `set` form of this command to specify the name of a resource address group as the translation address for a NAT rule.

You can create a NAT rule with specific translation addresses, address ranges, or both. Each entry listed in the resource address group is used to create a set of mappings based on the port range.

> **Note:** This feature traverses the entries specified in the resource address group in numerical ascending order and not in the order the addresses were configured.

Use the `delete` form of this command to remove the address-group.

service nat

Enables NAT on the system.

Syntax:

`set service nat`

Syntax:
`delete service nat`

Syntax:
`show service nat`

Configuration mode

```
service {
    nat {
    }
}
```

Use the `set` form of this command to enable, create, or modify the NAT configuration.

Use the `delete` form of this command to remove NAT configuration and disable NAT on the system.

Use the `show` form of this command to view NAT configuration.

service nat destination rule <rule-number>

Defines a NAT destination rule number.

Syntax:
`set service nat destination rule` *rule-number*

Syntax:
`delete service nat destination` **`rule`** *rule-number*

Syntax:
`show service nat destination` **`rule`** *rule-number*

rule-number
Multi-node. The rule number for NAT that ranges from 1 through 9999.

Configuration mode

```
service {
    nat {
        destination {
            rule rule-number
        }
    }
}
```

Use this command to define a NAT rule number. The *number* argument defines the destination NAT rule. Destination NAT rules translate the destination IP address. Destination rules typically ingress from the untrusted to the trusted network. For destination NAT rules, the translation address typically defines an IP address inside the trusted network. This address is substituted for the original destination IP address in ingressing packets.

NAT rules are executed in numeric order. To allow insertion of more rules in the future, choose rule numbers in increments of ten, such as 10, 20, 30, 40, and so on. The numbers must be separated by a comma.

Use the `set` form of this command to define a NAT rule number.

Use the `delete` form of this command to remove a NAT rule number.

Use the `show` form of this command to view a NAT rule number.

service nat destination rule <rule-number> description <description>

Creates a brief description of a NAT destination rule.

Syntax:
`set service nat destination rule` *rule-number* **`description`** *description*

Syntax:
`delete service nat destination rule` *rule-number* **`description`**

Syntax:
`show service nat destination rule` *rule-number* **`description`**

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

description
A description for the rule. If the description contains spaces, it must be enclosed in double quotation marks.

Configuration mode

```
service {
    nat {
        destination {
            rule rule-number
                description description {
            }
        }
    }
}
```

Use this command to provide a description of an NAT rule to quickly determine the purpose of the rule when viewing the configuration.

Use the `set` form of this command to provide a description of a rule.

Use the `delete` form of this command to remove the description of a rule.

Use the `show` form of this command to view the description of a rule.

service nat destination rule <rule-number> destination <destination>

Specifies a destination an address, a port, or both, to match in a NAT destination rule.

Syntax:
`set service nat destination rule` *rule-number* **`destination`** { **`address`** *address* | **`port`** *port* }

Syntax:
`delete service nat destination rule` *rule-number* **`destination`** [**`address`** *address* | **`port`** *port*]

Syntax:
`show service nat destination rule` *rule-number* **`destination`**

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

address
A destination address to match. Address formats are as follows:

ip-address: An IP address.

ip-address/prefix: An IPv4 network address, where 0.0.0.0/0 matches any network.

`!`*ip-address*: All IPv4 addresses except the one specified.

`!`*ip-address/prefix*: All IPv4 network addresses except the one specified.

port

A destination port to match. Port formats are as follows:

port-name: The name of an IP service; for example, http. You can specify any service name in the `etc/services` file.

port-number: A port number. The number ranges from 1 through 65535.

start-end: A range of ports; for example, 1001-1005.

Configuration mode

```
service {
    nat {
        destination {
            rule rule-number {
                destination {
                    address address
                    port port
                }
            }
        }
    }
}
```

Use care when employing more than one exclusion rule (using the `!` symbol), that is, when combining more than one negation operator. NAT rules are evaluated sequentially, and a sequence of exclusion rules could result in unexpected behavior.

Use the `set` form of this command to specify a destination address and port to match a NAT rule (destination filter).

Use the `delete` form of this command to remove a destination filter.

Use the `show` form of this command to view a destination filter.

service nat destination rule <rule-number> disable

Disables a NAT destination rule.

Syntax:
`set service nat destination rule` *rule-number* **`disable`**

Syntax:
`delete service nat destination rule` *rule-number* **`disable`**

Syntax:
`show service nat destination rule` *rule-number* **`disable`**

The rule is enabled.

rule-number

Multi-node. The rule number for NAT that ranges from 1 through 9999.

Configuration mode

```
service {
    nat {
        destination {
            rule rule-number {
                disable
            }
        }
    }
}
```

Use the `set` form of this command to disable a NAT rule.

Use the `delete` form of this command to return a rule to its enabled state.

Use the `show` form of this command to view a rule.

service nat destination rule <rule-number> exclude

Creates an exclusion rule, which excludes from address translation packets that match this destination rule.

Syntax:
`set service nat destination rule` *rule-number* **`exclude`**

Syntax:
`delete service nat destination rule` *rule-number* [**`exclude`**]

Syntax:
`show service nat destination rule` *rule-number*

rule-number
Multi-node. The rule number for NAT that ranges from 1 through 9999.

Configuration mode

```
service {
    nat {
        destination {
            rule rule-number {
                exclude
            }
        }
    }
}
```

Use this command to create an exclusion rule, which excludes from address translation packets that match this rule. Exclusion can be used in scenarios in which certain types of traffic (for example, VPN traffic) should not be translated.

Use the `set` form of this command to specify that packets matching this rule are excluded from NAT.

Use the `delete` form of this command to remove an exclusion rule.

Use the `show` form of this command to view an exclusion rule.

service nat destination rule <rule-number> inbound-interface

Applies DNAT rules to the inbound traffic of an interface.

Syntax:
`set service nat destination rule` *rule-number* **`inbound-interface`**

Syntax:
`delete service nat destination rule` *rule-number* [**`inbound-interface`**]

Syntax:
`show service nat destination rule` *rule-number*

rule-number

Multi-node. The rule number for NAT that ranges from 1 through 9999.

Configuration mode

```
service {
    nat {
        destination {
            rule rule-number {
                inbound-interface {
                }
            }
        }
    }
}
```

Applied to the inbound data plane interface. SNAT or masquerade NAT is performed on traffic transmitted from this interface. This attribute is not configurable for source rules.

You can specify an individual virtual interface instead of an entire interface. To do this, refer to the virtual interface by using the *int* .*vif* notation. For example, to refer to the 40 virtual interface on the dp0p160p0 interface, use dp0p160p1.40.

This command can be used only on source NAT rules (that is, NAT rules with a rule type of **`source`**). It does not apply to rules with a rule type of **`destination`**.

Use the `set` for of this command to specify the data plane interface on which inbound traffic has DNAT rules applied.

Use the `delete` form of this command to remove an inbound interface.

Use the `show` form of this command to view an inbound interface.

service nat destination rule <rule-number> log

Enables logging of entries for matches with a NAT destination rule.

Syntax:
`set service nat destination rule` *rule-number* **`log`**

Syntax:
`delete service nat destination rule` *rule-number* **`log`**

Syntax:
`show service nat destination rule` *rule-number* **`log`**

Log entries are not generated for matched rules.

rule-number

The numeric identifier of a rule. The identifier ranges from 1 through 9999.

Configuration mode

```
service {
    nat {
        destination {
            rule rule-number {
                log
            }
        }
    }
}
```

Use care when enabling this feature because it can create very large log files and quickly fill a disk.

Use the `set` form of this command to enable logging of entries for matches with a NAT destination rule.

Use the `delete` form of this command to restore the default NAT logging, that is, the logging of NAT destination entries is not generated.

Use the `show` form of this command to view the state of NAT logging.

service nat destination rule <rule-number> protocol

Specifies one or more protocols on which NAT destination rule is performed.

Syntax:

```
set service nat destination rule rule-number protocol protocol
```

Syntax:

```
delete service nat destination rule rule-number protocol protocol
```

Syntax:

```
show service nat destination rule rule-number protocol
```

rule-number

The numeric identifier of a rule. The identifier ranges from 1 through 9999.

protocol

A protocol or protocols on which to perform NAT. Any protocol literals or numbers listed in `/etc/protocols` can be used. Protocols such as TCP, UDP, L2TP, or IPSec ESP can be matched individually.

`all`: Supported for all protocols.

Configuration mode

```
service {
    nat {
        destination {
            rule rule-number {
                protocol protocol
            }
        }
    }
}
```

Use care when employing more than one exclusion rule (using the **!** symbol), that is, when combining more than one negation operator. NAT rules are evaluated sequentially, and a sequence of exclusion rules could result in unexpected behavior.

Use the `set` form of this command to specify one or more protocols on which NAT destination rule is performed.

Use the `delete` form of this command to remove a protocol from a NAT destination rule.

Use the `show` form of this command to view a protocol for a NAT destination rule.

service nat destination rule <rule-number> source <address>

Specifies a source address and port to match in a NAT destination rule.

Syntax:

```
set service nat destination rule rule-number source { address address | port port }
```

Syntax:
`delete service nat destination` **`rule`** *rule-number* **`source`** `[` **`address`** `|` **`port`** `]`

Syntax:
`show service nat destination` **`rule`** *rule-number* **`source`** `[` **`address`** `|` **`port`** `]`

rule-number

The numeric identifier of a rule. The identifier ranges from 1 through 9999.

address

A source address to match. Address formats are as follows:

ip-address: An IP address.

ip-address/prefix: A network address, where 0.0.0.0/0 matches any network.

`!`*ip-address*: All IP addresses except the one specified.

`!`*ip-address/prefix*: All network addresses except the one specified.

port

A source port to match. Port formats are as follows:

port-name: The name of an IP service; for example, http. You can specify any service name in the `etc/services` file.

port-number: A port number. The number ranges from 1 through 65535.

start`-`*end*: A range of ports; for example, 1001-1005.

Configuration mode

```
service {
    nat {
        destination {
            rule rule-number {
                source {
                    address address
                    port port
                }
            }
        }
    }
}
```

Use care when employing more than one exclusion rule (using the `!` symbol), that is, when combining more than one negation operator. NAT rules are evaluated sequentially, and a sequence of exclusion rules could result in unexpected behavior.

Use the `set` form of this command to specify a source address and port to match in a NAT rule (source filter).

Use the `delete` form of this command to remove a source filter.

Use the `show` form of this command to view a source filter.

service nat destination rule <rule-number> translation <address>

Specifies a translated address or port address in a NAT rule.

Syntax:
`set service nat destination` **`rule`** *rule-number* **`translation`** `{` **`address`** *address* `|` **`port`** *port* `}`

Syntax:
`delete service nat destination` **`rule`** *rule-number* **`translation`** `[` **`address`** `|` **`port`** `]`

Syntax:
show service nat destination **rule** *rule-number* **translation** [**address** | **port**]

rule-number

The numeric identifier of a NAT rule. The identifier ranges from 1 through 9999.

address ***address***

An IP address or range of addresses to substitute for the original address or addresses. Address formats are as follows:

ip-address: An IP address.

ip-address/prefix: A network. This argument is typically used in bidirectional NAT to translate one network of addresses to another.

ip-address-range: A range of network IP address.

ip-address-group: The name of a resource address group.

port ***port***

An IP port to substitute for the original port. It cannot be used when the **source address** or **destination address** and the **translation address** are IPv4 subnets. Port formats are as follows:

port-number: A port number. The number ranges from 1 through 65535.

start-*end*: A range of contiguous ports; for example, 1001-1005.

Configuration mode

```
service {
    nat {
        destination {
            rule rule-number {
                translation {
                    address address
                    port port
                }
            }
        }
    }
}
```

Use the set form of this command to configure a translated address or port address for a NAT rule.

Use the delete form of this command to remove a translated address or a port address from a NAT rule.

Use the show form of this command to view a translated address or port address for a NAT rule.

service nat ipv6-to-ipv4 rule <rule-number> destination prefix <dest_addr_prefix>

Specifies the NAT IPv6-to-IPv4 rule to apply to the IPv6 prefix for the destination.

Syntax:
set service **nat** **ipv6-to-ipv4** **rule** *rule-number* **destination** **prefix** *dest_addr_prefix*

Syntax:
delete service **nat** **ipv6-to-ipv4** **rule** *rrule-number* **destination** **prefix** *dest_addr_prefix*

Syntax:
show service nat

rule-number

The NAT rule to apply. The identifier ranges from 1 through 9999.

dest_addr_prefix

The IPv6 address prefix of the destination node in the IPv4 network.

Configuration mode

```
service {
    nat {
        ipv6-to-ipv4 {
            rule rule-number {
                destination {
                    prefix dest_addr_prefix
                }
            }
        }
    }
}
```

Use the `set` form of this command to specify the NAT IPv6-to-IPv4 rule to apply to the IPv6 prefix for the destination.

Use the `delete` form of this command to delete the NAT IPv6-to-IPv4 rule.

Use the `show` form of this command to display the configured NAT IPv6-to-IPv4 rules.

service nat ipv6-to-ipv4 rule <rule-number> inbound-interface <inbound_intf>

Specifies the service IPv6-to-IPv4 rule to apply to the inbound interface.

Syntax:
set service **nat ipv6-to-ipv4 rule** *rule-number* **inbound-interface** *inbound_intf*

Syntax:
delete service **nat ipv6-to-ipv4 rule** *rule-number* **inbound-interface** *inbound_intf*

Syntax:
show service nat

rule-number
The rule to apply to the inbound IPv6 traffic. The identifier ranges from 1 through 9999.

inbound_intf
The data plane interface for which to enable NAT 6-4 translation.

Configuration mode

```
service {
    nat {
        ipv6-to-ipv4 {
            rule rule-number {
                inbound-interface inbound_intf
            }
        }
    }
}
```

Use the `set` form of this command to specify the service IPv6-to-IPv4 rule to apply to the inbound interface.

Use the `delete` form of this command to delete the service IPv6-to-IPv4 rule.

Use the `show` form of this command to display the configured IPv6-to-IPv4 rules.

service nat ipv6-to-ipv4 rule <rule-number> source prefix <src_addr_prefix>

Specifies the service IPv6-to-IPv4 rule to apply to the source prefix for the destination.

Syntax:

```
set service nat ipv6-to-ipv4 rule rule-number source prefix src_addr_prefix
```

Syntax:

```
delete service nat ipv6-to-ipv4 rule rule-number source prefix src_addr_prefix
```

Syntax:

```
show service nat
```

rule-number

The rule to apply to the inbound IPv6 traffic. The identifier ranges from 1 through 9999.

src_addr_prefix

The IPv6 prefix that matches the incoming IPv6 addresses.

Configuration mode

```
service {
    nat {
        ipv6-to-ipv4 {
            rule rule-number {
                source {
                    prefix src_addr_prefix
                }
            }
        }
    }
}
```

Use the `set` form of this command to specify the service IPv6-to-IPv4 rule to apply to the source prefix for the destination.

Use the `delete` form of this command to delete the service IPv6-to-IPv4 rule.

Use the `show` form of this command to display the configured IPv6-to-IPv4 rules.

service nat source rule <rule-number> description <description>

Provides a brief description for a NAT source rule.

Syntax:

```
set service nat source rule rule-number description description
```

Syntax:

```
delete service nat source rule rule-number description
```

Syntax:

```
show service nat source rule rule-number description
```

rule-number

The numeric identifier of a rule. The identifier ranges from 1 through 9999.

description

A description for the rule. If the description contains spaces, it must be enclosed in double quotation marks.

Configuration mode

```
service {
    nat {
        source {
            rule rule-number {
                description description
            }
        }
    }
}
```

Providing a description for an NAT rule can help you to quickly determine the purpose of the rule when viewing the configuration.

Use the `set` form of this command to provide a description of a rule.

Use the `delete` form of this command to remove the description of a rule.

Use the `show` form of this command to view the description of a rule.

service nat source rule <rule-number> destination <address>

Specifies a destination address and port to match in a NAT source rule.

Syntax:
`set service nat source rule` *rule-number* **`destination`** { **`address`** *address* | **`port`** *port* }

Syntax:
`delete service nat source rule` *rule-number* **`destination`** [**`address`** | **`port`**]

Syntax:
`show service nat source rule` *rule-number* **`destination`** [**`address`** | **`port`**]

rule-number

The numeric identifier of a rule. The identifier ranges from 1 through 9999.

address

A destination address to match. Address formats are as follows:

ip-address: An IP address.

ip-address/prefix: An IPv4 network address, where 0.0.0.0/0 matches any network.

`!`*ip-address*: All IPv4 addresses except the one specified.

`!`*ip-address/prefix*: All IPv4 network addresses except the one specified.

port

A destination port to match. Port formats are as follows:

port-name: The name of an IP service; for example, http. You can specify any service name in the `etc/services` file.

port-number: A port number. The number ranges from 1 through 65535.

start-end: A range of ports; for example, 1001-1005.

Configuration mode

```
service {
    nat {
        source {
            rule rule-number {
                address address
                port port
            }
        }
    }
}
```

Use care when employing more than one exclusion rule (using the `!` symbol), that is, when combining more than one negation operator. NAT rules are evaluated sequentially, and a sequence of exclusion rules could result in unexpected behavior.

Use the `set` form of this command to specify a destination address and port to match a NAT source rule (destination filter).

Use the `delete` form of this command to remove a destination filter for a NAT source rule.

Use the `show` form of this command to view a destination filter for a NAT source rule.

service nat source rule <rule-number> disable

Disables a NAT rule.

Syntax:
`set service nat source rule` *rule-number* **`disable`**

Syntax:
`delete service nat source rule` *rule-number* **`disable`**

Syntax:
`show source nat rule` *rule-number* **`disable`**

The rule is enabled.

rule-number
Multi-node. The rule number for NAT that ranges from 1 through 9999.

Configuration mode

```
service {
    nat {
        source {
            rule rule-number {
                disable
            }
        }
    }
}
```

Use the `set` form of this command to disable a NAT rule.

Use the `delete` form of this command to return a rule to its enabled state.

Use the `show` form of this command to view a rule.

service nat source rule <rule-number> exclude

Creates an exclusion rule, which excludes from address translation packets that match this source rule.

Syntax:
`set service nat source rule` *rule-number* **`exclude`**

Syntax:

`delete service nat source rule` *rule-number* **`exclude`**

Syntax:
`show service nat source rule` *rule-number*

rule-number
Multi-node. The rule number for NAT that ranges from 1 through 9999.

Configuration mode

```
service {
    nat {
        source {
            rule rule-number {
                exclude
            }
        }
    }
}
```

Use this command to create an exclusion rule, which excludes from address translation packets that match this rule. Exclusion can be used in scenarios in which certain types of traffic (for example, VPN traffic) should not be translated.

Use the `set` form of this command to specify that packets matching this rule are excluded from NAT.

Use the `delete` form of this command to remove an exclusion rule.

Use the `show` form of this command to view an exclusion rule.

service nat source rule <rule-number> log

Enables logging of entries for matches with a NAT source rule.

Syntax:
`set service nat source rule` *rule-number* **`log`**

Syntax:
`delete service nat source rule` *rule-number* **`log`**

Syntax:
`show service nat source rule` *rule-number* **`log`**

Log entries are not generated for matched rules.

rule-number
The numeric identifier of a rule. The identifier ranges from 1 through 9999.

Configuration mode

```
service {
    nat {
        source {
            rule rule-number {
                log
            }
        }
    }
}
```

Use care when enabling this feature because it can create very large log files and quickly fill a disk.

Use the `set` form of this command to enable logging of entries for matches with a NAT source rule.

Use the `delete` form of this command to restore the default NAT logging, that is, the logging of NAT source entries is not generated.

Use the `show` form of this command to view the state of NAT logging.

service nat source rule <rule-number> outbound-interface <interface>

Specifies an interface on which outbound traffic has SNAT rules applied.

Syntax:
`set service nat source rule` *rule-number* **`outbound-interface`** *interface*

Syntax:
`delete service nat source rule` *rule-number* **`outbound-interface`** *interface*

Syntax:
`show service nat source rule` *rule-number* **`outbound-interface`** *interface*

rule-number

The numeric identifier of a rule. The identifier ranges from 1 through 9999.

interface

Not configurable for **`destination`** rules. The outbound interface. SNAT or masquerade NAT is performed on traffic transmitted from this interface.

You can specify an individual virtual interface instead of an entire interface. To do this, refer to the virtual interface by using *int.vif* notation. For example, to refer to the 40 virtual interface on the dp0p160p0 interface, use dp0p160p1.40.

Configuration mode

```
service {
    nat {
        source {
            rule rule-number {
                outbound-interface interface
            }
        }
    }
}
```

This command can be used only on source NAT rules (that is, NAT rules with a rule type of **`source`**). It does not apply to rules with a rule type of **`destination`**.

Use the `set` form of this command to specify the data plane interface on which outbound traffic has SNAT rules applied.

Use the `delete` form of this command to remove an outbound interface.

Use the `show` form of this command to view an outbound interface.

service nat source rule <rule-number> protocol <protocol>

Specifies one or more protocols on which NAT source rule is performed.

Syntax:
`set service nat source` **`rule`** *rule-number* **`protocol`** *protocol*

Syntax:
`delete service nat source` **`rule`** *rule-number* **`protocol`** *protocol*

Syntax:

`show service nat source rule` *rule-number* **`protocol`** *protocol*

rule-number

The numeric identifier of a rule. The identifier ranges from 1 through 9999.

protocol

A protocol or protocols on which to perform NAT. Any protocol literals or numbers listed in `/etc/protocols` can be used. Protocols such as TCP, UDP, L2TP, or IPSec ESP can be matched individually.

Configuration mode

```
service {
    nat {
        source {
            rule rule-number {
                protocol protocol
            }
        }
    }
}
```

Use care when employing more than one exclusion rule (using the `!` symbol), that is, when combining more than one negation operator. NAT rules are evaluated sequentially, and a sequence of exclusion rules could result in unexpected behavior.

Use the `set` form of this command to specify one or more protocols on which NAT source rule is performed.

Use the `delete` form of this command to remove a protocol from a NAT source rule.

Use the `show` form of this command to view a protocol for a NAT source rule.

service nat source rule <rule-number> source <address>

Specifies a source address and port to match in a NAT source rule.

Syntax:

`set service nat source rule` *rule-number* **`source`** { **`address`** *address* | **`port`** *port* }

Syntax:

`delete service nat source rule` *rule-number* **`source`** [**`address`** | **`port`**]

Syntax:

`show service nat source rule` *rule-number* **`source`** [**`address`** | **`port`**]

rule-number

The numeric identifier of a rule. The identifier ranges from 1 through 9999.

address

A source address to match. Address formats are as follows:

ip-address: An IP address.

ip-address/prefix: A network address, where 0.0.0.0/0 matches any network.

`!`*ip-address*: All IP addresses except the one specified.

`!`*ip-address/prefix*: All network addresses except the one specified.

port

A source port to match. Port formats are as follows:

port-name: The name of an IP service; for example, http. You can specify any service name in the `etc/services` file.

port-number: A port number. The number ranges from 1 through 65535.

start-end: A range of ports; for example, 1001-1005.

Configuration mode

```
service {
    nat {
        source {
            rule rule-number {
                source {
                    address address
                    port port
                }
            }
        }
    }
}
```

Use care when employing more than one exclusion rule (using the **!** symbol), that is, when combining more than one negation operator. NAT rules are evaluated sequentially, and a sequence of exclusion rules could result in unexpected behavior.

Use the `set` form of this command to specify a source address and port to match in a NAT source rule (source filter).

Use the `delete` form of this command to remove a source filter.

Use the `show` form of this command to view a source filter.

service nat source rule <rule-number> translation <address>

Specifies a translated address or port address for a NAT rule.

Syntax:
set service nat source **rule** *rule-number* **translation** { **address** *address* | **port** *port* }

Syntax:
delete service nat source **rule** *rule-number* **translation** [**address** | **port**]

Syntax:
show service nat source **rule** *rule-number* **translation** [**address** | **port**]

rule-number
The numeric identifier of a NAT rule. The identifier ranges from 1 through 9999.

`address` ***address***
An IP address or an IP address/prefix to substitute for the original address or addresses. Address formats are as follows:

ip-address: An IP address.

ip-address/prefix: A network. This argument is typically used in bidirectional NAT to translate one network of addresses to another.

ip-address-range: A range of network IP address.

ip-address-group: The name of a resource address-group.

masquerade: A format that is available only when rule type is set to **source**. It specifies that the source IP address is to be set to the primary IP address on the outbound interface.

`port` ***port***
An IP port to substitute for the original port. It cannot be used when the **source address** or **destination address** and the **translation address** are IPv4 subnets. Port formats are as follows:

port-number: A port number. The number ranges from 1 through 65535.

start-end: A range of contiguous ports; for example, 1001-1005.

Configuration mode

```
service {
    nat {
      source {
        rule rule-number {
            translation {
                address address
                port port
            }
         }
      }
    }
}
```

Use this command to specify a translated address or port address for a NAT rule. A translated address or a port address must be specified for each rule.

Use the `set` form of this command to configure a translated address or port address for a NAT rule.

Use the `delete` form of this command to remove a translated address or port address from a NAT rule.

Use the `show` form of this command to view a translated address or port address for a NAT rule.

show nat destination

Displays configured destination NAT (DNAT) rules, statistics, or translations.

Syntax:

show nat destination [**rules** | **statistics** | **translations**]

rules

Destination NAT rules.

statistics

Destination NAT statistics such as address and port information.

translations

Destination NAT translations.

Operational mode

Use this command to display the NAT rules you have configured. You can use this command for troubleshooting, to confirm whether traffic is matching the NAT rules as expected.

The following example shows how to display configured destination NAT rules.

```
vyatta@vyatta:~$ show nat destination rules
-----------------------------
NAT Rulesets Information
-----------------------------
----------------------------------------------------------------------------
DESTINATION
rule   intf        match                                   translation
----   ----        -----                                   -----------
120    dp0s5    proto tcp to 172.16.139.100 port 80 ipv4 tag 0 dynamic 10.0.0.102
port 1-65535 <-any
```

The following example shows how to display current statistics for destination NAT.

```
vyatta@vyatta:~$ show nat destination statistics
rule    pkts            bytes                   interface       used/total
----    ----            -----                   ---------       ----------
```

```
120     14              1036                    dp0s5           2/65535
```

Note:

The used/total column refers to the translation space as defined by the NAT rule. The value is equivalent to the number of addresses multiplied by the number of ports. DNAT can exceed the translation space while SNAT cannot. In SNAT, if the translation space is exhausted, the remaining packets are dropped.

The following example shows how to display destination NAT translation information.

```
vyatta@vyatta:~$ show nat destination translations
Pre-NAT                  Post-NAT                Prot       Timeout

172.16.139.100:80        10.0.0.102:80           tcp        25
```

show nat source

Displays configured source NAT (SNAT) rules.

Syntax:

`show nat source` [**`rules`** | **`statistics`** | **`translations`**]

`rules`
    Source NAT rules.
`statistics`
    Source NAT statistics such as address and port information.
`translations`
    Source NAT translations.

Operational mode

Use this command to display the NAT rules you have configured. You can use this command for troubleshooting, to confirm whether traffic is matching the NAT rules as expected.

The following example shows how to display source rules for NAT.

```
vyatta@vyatta:~$ show nat source rules
--------------------------------------
NAT Rulesets Information
------------------------
-----------------------------------------------------------------------
SOURCE
rule    intf        match       translation
----    ----        -----       -----------
20      dp0s5       proto 1 from 10.0.0.102 to 172.16.140.200 tag 0 dynamic any ->
172.16.139.100
30      dp0s5        from 10.0.0.0/24 ipv4 tag 0          dynamic any -> masquerade
```

The following example shows how to display current statistics for source NAT.

```
vyatta@vyatta:~$ show nat source statistics
rule   pkts   bytes  interface  used/total
----   ----   -----  ---------  ----------
1      111    20006  dp0s5      1/65535
2      0      0      dp0s5      0/11
```

Note:

The used/total column refers to the translation space as defined by the NAT rule. The value is equivalent to the number of addresses multiplied by the number of ports. DNAT can exceed the translation space while SNAT cannot. In SNAT, if the translation space is exhausted, the remaining packets are dropped.

The following example shows how to display source NAT translation information.

```
vyatta@vyatta:~$ show nat source translations
Pre-NAT               Post-NAT                Prot     Timeout

10.0.0.101:56803      172.16.139.100:56803    tcp      86375
10.0.0.102:48635      172.16.139.100:48635    tcp      0
10.0.0.102:56279      172.16.139.100:56279    tcp      0
10.0.0.102:56432      172.16.139.100:56432    tcp      4
10.0.0.102            172.16.139.100          icmp     59
```

Related commands

The following table lists related commands in other guides.

Related Commands Documented Elsewhere	
resources group address#group <group#name>	Defines a group of IP addresses that are referenced in firewall rules. (Refer to *Basic Routing Reference Guide*)
resources group port#group <group#name>	Defines a group of ports that are referenced in firewall rules. (Refer to *Basic Routing Reference Guide*)

VRF Support

VRF support for flow monitoring

NAT is independent of routing instances. However, because interfaces can be bound to a routing instance, if you want a VRF NAT, you must assign NAT rules to the interfaces that make up the routing instance.

Command support for VRF routing instances

VRF allows an AT&T Vyatta vRouter to support multiple routing tables, one for each VRF routing instance. Some commands in this guide support VRF and can be applied to particular routing instances.

Use the guidelines in this section to determine correct syntax when adding VRF routing instances to commands. For more information about VRF, refer to AT&T Vyatta Network Operating System Basic Routing Configuration Guide. This guide includes an overview of VRF, VRF configuration examples, information about VRF-specific features, and a list of commands that support VRF routing instances.

Adding a VRF routing instance to a Configuration mode command

For most Configuration mode commands, specify the VRF routing instance at the beginning of a command. Add the appropriate VRF keywords and variable to follow the initial action (**`set`**, **`show`**, or **`delete`**) and before the other keywords and variables in the command.

Example: Configuration mode example: syslog

The following command configures the syslog logging level for the specified syslog host. The command does not include a VRF routing instance, so the command applies to the default routing instance.

```
vyatta@R1# set system syslog host 10.10.10.1 facility all level debug
vyatta@R1# show system syslog
syslog {
    host 10.10.10.1 {
            facility all {
                    level debug
            }
    }
}
```

The following example shows the same command with the VRF routing instance (GREEN) added. Notice that **`routing routing-instance GREEN`** has been inserted between the basic action (**set** in the example) and the rest of the command. Most Configuration mode commands follow this convention.

```
vyatta@R1# set routing routing-instance GREEN system syslog host 10.10.10.1 facility all
 level debug
vyatta@R1# show routing
routing {
     routing-instance GREEN {
             system {
                     syslog {
                            host 11.12.13.2:514 {
                                     facility all {
                                             level debug
                                     }
                             }
                     }
             }
     }
}
```

Example: Configuration mode example: SNMP

Some features, such as SNMP, are not available on a per-routing instance basis but can be bound to a specific routing instance. For these features, the command syntax is an exception to the convention of specifying the routing instance at the beginning of Configuration mode commands.

The following example shows how to configure the SNMPv1 or SNMPv2c community and context for the RED and BLUE routing instances. The first two commands specify the RED routing instance as the context for community A and BLUE routing instance as the context for community B. The subsequent commands complete the configuration.

For more information about configuring SNMP, refer to AT&T Vyatta Network Operating System Remote Management Configuration Guide.

```
vyatta@R1# set service snmp community commA context RED
vyatta@R1# set service snmp community commB context BLUE
vyatta@R1# set service snmp view all oid 1
vyatta@R1# set service snmp community commA view all
vyatta@R1# set service snmp community commB view all
vyatta@R1# show service snmp community
 community commA {
        context RED
        view all
 }
 community commB {
        context BLUE
        view all
 }
[edit]
vyatta@vyatta#
```

Adding a VRF routing instance to an Operational mode command

The syntax for adding a VRF routing instance to an Operational mode command varies according to the type of command parameters:

- If the command does not have optional parameters, specify the routing instance at the end of the command.
- If the command has optional parameters, specify the routing instance after the required parameters and before the optional parameters.

Example: Operational mode examples without optional parameters

The following command displays dynamic DNS information for the default routing instance.

```
vyatta@vyatta:~$ show dns dynamic status
```

The following command displays the same information for the specified routing instance (GREEN). The command does not have any optional parameters, so the routing instance is specified at the end of the command.

```
vyatta@vyatta:~$ show dns dynamic status routing-instance GREEN
```

Example: Operational mode example with optional parameters

The following command obtains multicast path information for the specified host (10.33.2.5). A routing instance is not specified, so the command applies to the default routing instance.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 detail
```

The following command obtains multicast path information for the specified host (10.33.2.5) and routing instance (GREEN). Notice that the routing instance is specified before the optional **detail** keyword.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 routing-instance GREEN detail
```

Example: Operational mode example output: SNMP

The following SNMP **show** commands display output for routing instances.

```
vyatta@vyatta:~$ show snmp routing-instance
Routing Instance SNMP Agent is Listening on for Incoming Requests:
Routing-Instance            RDID
-----------------           ----
RED                         5

vyatta@vyatta:~$ show snmp community-mapping
SNMPv1/v2c Community/Context Mapping:
Community                   Context
---------                   -------
commA                       'RED'
commB                       'BLUE'
deva                        'default'


vyatta@vyatta:~$ show snmp trap-target
SNMPv1/v2c Trap-targets:
Trap-target                   Port   Routing-Instance Community
-----------                   ----   ---------------- ---------
1.1.1.1                              'RED'           'test'


vyatta@vyatta:~$ show snmp v3 trap-target
SNMPv3 Trap-targets:
Trap-target                   Port   Protocol Auth Priv Type   EngineID              Routing-
Instance User
-----------                   ----   -------- ---- ---- ----   --------
 ---------------- ----
2.2.2.2                       '162'  'udp'    'md5      'infor                       'BLUE'
         'test'
```

List of Acronyms

Acronym	Description
ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMVPN	dynamic multipoint VPN
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers

Acronym	Description
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP Security
IPv4	IP Version 4
IPv6	IP Version 6
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Internet Standard Multicast
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
mGRE	multipoint GRE
MIB	Management Information Base
MLD	Multicast Listener Discovery
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbor Discovery
NHRP	Next Hop Resolution Protocol
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode
PIM-SM	PIM Sparse Mode
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM

Acronym	Description
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PTMU	Path Maximum Transfer Unit
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RSA	Rivest, Shamir, and Adleman
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPT	Shortest Path Tree
SSH	Secure Shell
SSID	Service Set Identifier
SSM	Source-Specific Multicast
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	virtual private network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access