- Network Address Translation -

NAT (Network Address Translation)

The rapid growth of the Internet resulted in a shortage of available IPv4 addresses. In response, a specific subset of the IPv4 address space was designated as *private,* to temporarily alleviate this problem.

A **public address** can be routed on the Internet. Thus, devices that must be Internet-accessible must be configured with (or *reachable* by) public addresses. Allocation of public addresses is governed by the Internet Assigned Numbers Authority (IANA).

A **private address** is intended for internal use within a home or organization, and can be freely used by anyone. However, private addresses can *never be routed* on the Internet. In fact, Internet routers are configured to immediately drop traffic with private addresses.

Three private address ranges were defined in RFC 1918, one for each IPv4 class:

- Class A - **10.x.x.x /8**
- Class B - **172.16.x.x /12**
- Class C - **192.168.x.x /24**

It is possible to *translate* between private and public addresses, using **Network Address Translation (NAT).** NAT allows a host configured with a private address to be *stamped* with a public address, thus allowing that host to communicate across the Internet. It is also possible to translate multiple privately-addressed hosts to a single public address, which conserves the public address space.

NAT provides an additional benefit – hiding the specific addresses and addressing structure of the internal (or *private*) network.

Note: NAT is *not* restricted to private-to-public address translation, though that is the most common application. NAT can also perform public-to-public address translation, as well as private-to-private address translation.

NAT is only a temporarily solution to the address shortage problem. IPv4 will eventually be replaced with IPv6, which supports a vast address space.

Both Cisco IOS devices and PIX/ASA firewalls support NAT.

* * *

Types of NAT

NAT can be implemented using one of three methods:

Static NAT – performs a static one-to-one translation between two addresses, or between a *port* on one address to a port on another address. Static NAT is most often used to assign a public address to a device behind a NAT-enabled firewall/router.

Dynamic NAT – utilizes a **pool** of global addresses to dynamically translate the outbound traffic of clients behind a NAT-enabled device.

NAT Overload or **Port Address Translation (PAT) –** translates the outbound traffic of clients to unique port numbers off of a *single* global address. PAT is necessary when the number of internal clients exceeds the available global addresses.

NAT Terminology

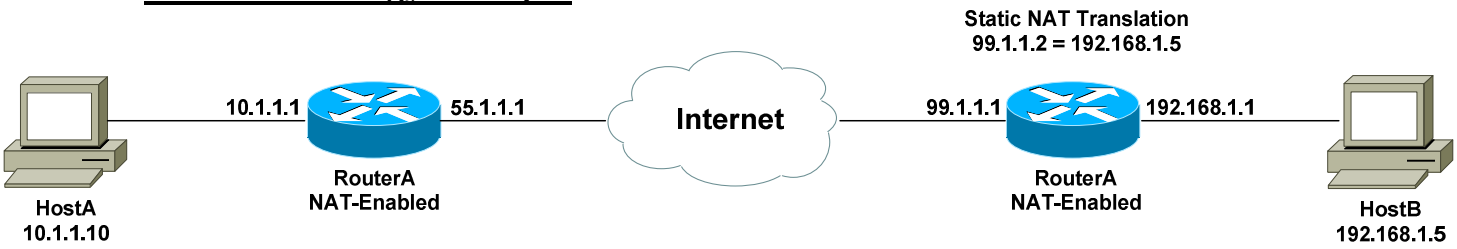
Specific terms are used to identify the various NAT addresses:

- **Inside Local –** the specific IP address assigned to an *inside* host behind a NAT-enabled device (usually a *private* address).
- **Inside Global –** the address that identifies an *inside* host to the *outside* world (usually a *public* address). Essentially, this is the dynamically or statically-assigned public address assigned to a private host.
- **Outside Global** – the address assigned to an *outside* host (usually a *public* address).
- **Outside Local** – the address that identifies an *outside* host to the *inside* network. Often, this is the **same** address as the Outside Global. However, it is occasionally necessary to translate an outside (usually *public*) address to an inside (usually *private*) address.

For simplicity sake, it is generally acceptable to associate **global** addresses with **public** addresses, and **local** addresses with **private** addresses. However, remember that public-to-public and private-to-private translation is still possible. **Inside** hosts are within the local network, while **outside** hosts are external to the local network.

* * *

NAT Terminology Example



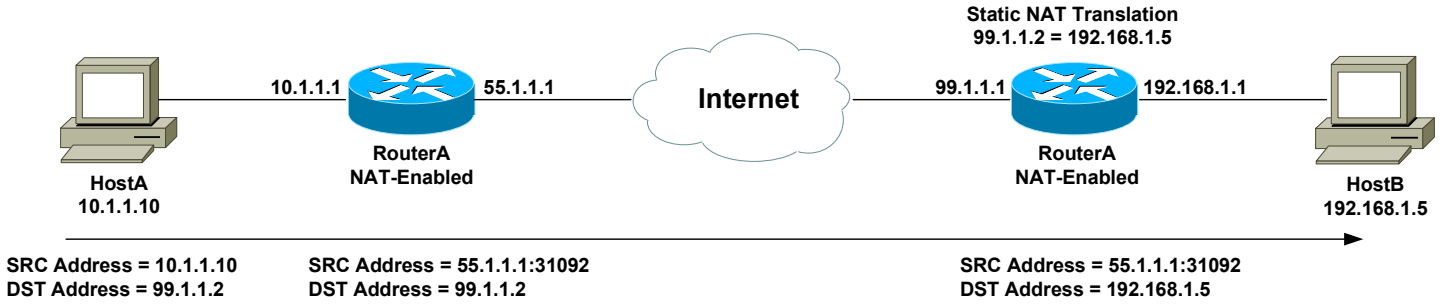
Consider the above example. For a connection *from* HostA *to* HostB, the NAT addresses are identified as follows:

- **Inside Local Address** - 10.1.1.10
- **Inside Global Address** - 55.1.1.1
- **Outside Global Address** – 99.1.1.2
- **Outside Local Address –** 99.1.1.2

HostA's configured address is *10.1.1.10*, and is identified as its *Inside Local* address. When HostA communicates with the Internet, it is stamped with RouterA's public address, using PAT. Thus, HostA's *Inside Global* address will become *55.1.1.1*.

When HostA communicates with HostB, it will access HostB's *Outside Global* address of *99.1.1.2*. In this instance, the *Outside Local* address is also *99.1.1.2*. HostA is never aware of HostB's configured address.

It is possible to map an address from the local network (such as 10.1.1.5) to the global address of the remote device (in this case, 99.1.1.2). This may be required if a legacy device exists that will only communicate with the local subnet. In this instance, the *Outside Local* address would be *10.1.1.5*.



The above example demonstrates how the source (SRC) and destination (DST) IP addresses within the Network-Layer header are translated by NAT.

(Reference: *http://www.cisco.com/warp/public/556/8.html*)

* * *

Configuring Static NAT

The first step to configure **Static NAT** is to identify the *inside* (usually private) and *outside* (usually public) interfaces:

Router(config)# *int e0/0*
Router(config-if)# *ip nat inside*

Router(config)# *int s0/0*
Router(config-if)# *ip nat outside*

To statically map a public address to a private address, the syntax is as follows:

Router(config)# *ip nat inside source static 172.16.1.1 158.80.1.40*

This command performs a *static* translation of the *source* address *172.16.1.1* (located on the *inside* of the network), to the outside address of *158.80.1.40.*

Configuring Dynamic NAT

When configuring **Dynamic NAT**, the *inside* and *outside* interfaces must first be identified:

Router(config)# *int e0/0*
Router(config-if)# *ip nat inside*

Router(config)# *int s0/0*
Router(config-if)# *ip nat outside*

Next, a *pool* of global addresses must be specified. Inside hosts will dynamically choose the next available address in this pool, when communicating outside the local network:

Router(config)# *ip nat pool POOLNAME 158.80.1.1 158.80.1.50 netmask 255.255.255.0*

The above command specifies that the *pool* named *POOLNAME* contains a range of public addresses from *158.80.1.1* through *158.80.1.50*.

Finally, a list of private addresses that are **allowed** to be dynamically translated must be specified:

Router(config)# *ip nat inside source list 10 pool POOLNAME*
Router(config)# *access-list 10 permit 172.16.1.0 0.0.0.255*

The first command states that any *inside* host with a *source* that matches access-*list 10* can be translated to any address in the *pool* named *POOLNAME*.

The *access-list* specifies any host on the *172.16.1.0* network.

* * *

Configuring NAT Overload (or PAT)

Recall that **NAT Overload** (or **PAT**) is necessary when the number of internal clients exceeds the available global addresses. Each internal host is translated to a unique port number off of a *single* global address.

Configuring NAT overload is relatively simple:

```
Router(config)# int e0/0
Router(config-if)# ip nat inside

Router(config)# int s0/0
Router(config-if)# ip nat outside

Router(config)# ip nat inside source list 10 interface Serial0/0 overload
Router(config)# access-list 10 permit 172.16.1.0 0.0.0.255
```

Any *inside* host with a *source* that matches access-*list 10* will be translated with *overload* to the IP address configured on the *Serial0/0* interface.

Troubleshooting NAT

To view all current static and dynamic translations:

```
Router# show ip nat translations
```

To view whether an interface is configure as an *inside* or *outside* NAT interface, and to display statistical information regarding active NAT translations:

```
Router# show ip nat statistics
```

To view NAT translations in real-time:

```
Router# debug ip nat
```

To clear all dynamic NAT entries from the translation table:

```
Router# clear ip nat translation
```

* * *