Why NAT?

Before we can discuss how NAT works, we must discuss the purpose of NAT and answer the question, "*Why NAT*?"

This article is a part of a series on Network Address Translation (**NAT**). Use the navigation boxes to view the rest of the articles.

Network Address Translation

- **Why NAT?**
- NAT Terminology
- Static NAT
- Static PAT
- Dynamic PAT
- Dynamic NAT
- Policy NAT and Twice NAT
- NAT Terminology Disambiguation

In the original plan for the Internet, every host was meant to have its own unique IP address. This means if you had a network which had 30 hosts, you would need 30 unique IP addresses for each host to *access the Internet*, or *to be accessed from the Internet*.

IP addresses are a finite resource – 32 bits allows for roughly 4.2 billion possible IP address combinations.

As the Internet grew in popularity, the industry realized there would one day be more hosts on the Internet than there were IP addresses available.

The long term, permanent solution was to create a larger address range, and IPv6 was born which is an addressing scheme that uses 128 bits. However, transitioning to IPv6 would prove to be a complicated and slow process, so a short term solution had to also be implemented: RFC 1918 was created to reduce the rate of IPv4 address utilization and delay the inevitable exhaustion of addresses.

RFC 1918

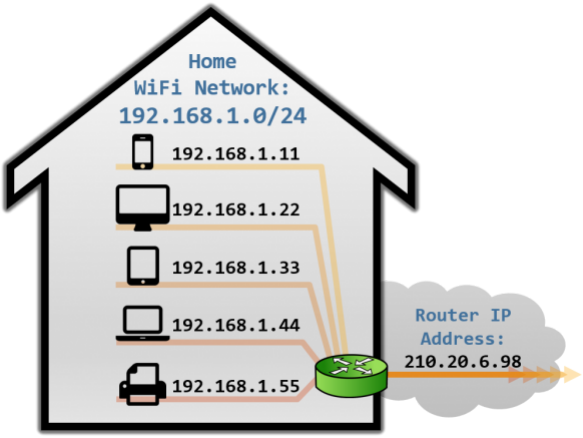
RFC 1918 designated three different address sets that were considered free to use and reuse by any organization:

- **`10.0.0.0 /8`** – any IP address in the range of **`10 . # . # . #`**
- **`172.16.0.0 /12`** – any IP address in the range of **`172 .[16–31]. # . #`**
- **`192.168.0.0 /16`** – any IP address in the range of **`192 . 168 . # . #`**

These addresses were labeled as **Private** addresses, and were deemed unroutable on the Internet. All the remaining addresses remained **Public** addresses, and able to be routed on th

Internet.

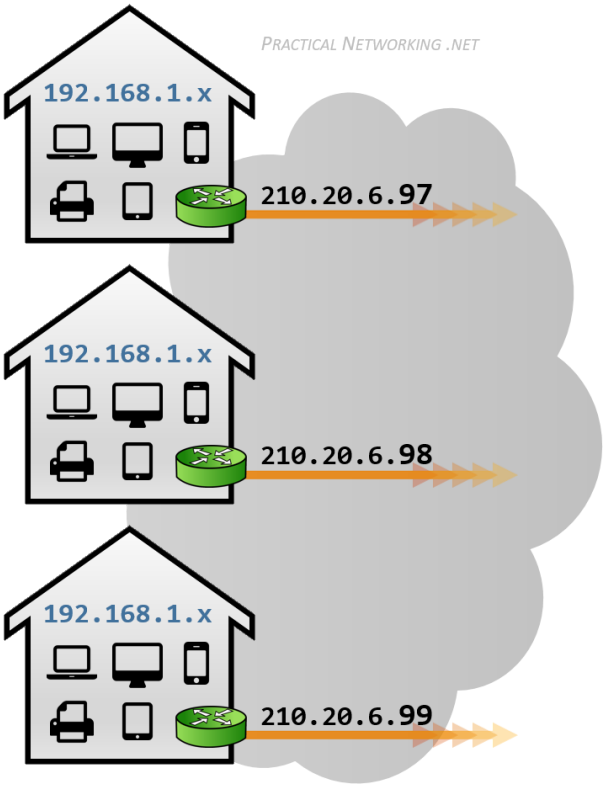
With RFC 1918, if you had 30 hosts on your network, all 30 of them would use 30 unique *Private* IP addresses, but for Internet facing traffic, all 30 could share a **single** *Public* address. Allowing you to conserve 29 Public addresses.



This is exactly what happens on WiFi networks. Whether it is a home WiFi network, or a coffee shop, or airport, each device on the network has a private IP address from one of the private ranges above. When these devices speak to the Internet, they all share the IP address assigned to the WiFi Router.

These Private addresses can be reused with each deployment without fear of duplicate addresses on the Internet. So long as the Public address(es) they are sharing are unique.

For example, a lot of home WiFi networks use the common range of `192.168.1.0/24` for each of their internal address ranges. The home Wifi router then translates each independent set of *Private* `192.168.1.0/24` addresses into unique *Public* addresses.



The idea is anyone can use these addresses, or even re-use these addresses, for as many hosts as they like on their internal network. NAT can then translate the multitude of hosts using *Private* addresses into a much smaller set of *Public* addresses – thereby curbing the rate of which IPv4 addresses are being utilized.

Private addresses are theoretically infinite, since they can be reused with each deployment. Public addresses are finite, and tracked by the Internet Authority for Assigned Numbers (IANA) to ensure no organization inadvertently uses duplicate Public addresses.

Consequently, the concept of **Network Address Translation was born to facilitate the translation between Private addresses and Public addresses**.

> Traditionally, NAT exists to translate Private IPv4 addresses into Public IPv4 addresses. For the sake of simplicity, this article series will describe NAT from this perspective. However, in reality, it does not matter whether the IP addresses being translated are public or private. NAT could easily occur from private addresses to other private addresses or from public addresses to other public addresses.

NAT Terminology

As discussed before, Network Address Translation, or NAT, is a process that involves translating Private IP addresses into Public IP addresses. There are different operations within NAT and understanding each of them requires understanding NAT terminology.

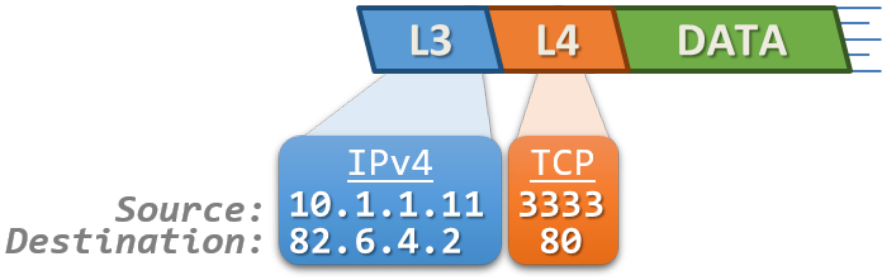
This article is a part of a series on Network Address Translation (**NAT**). Use the navigation boxes to view the rest of the articles.

Network Address Translation

- Why NAT?
- **NAT Terminology**
- Static NAT
- Static PAT
- Dynamic PAT
- Dynamic NAT
- Policy NAT and Twice NAT
- NAT Terminology Disambiguation

NAT vs PAT

Network **A**ddress **T**ranslation and **P**ort **A**ddress **T**ranslation differ by modifying different headers in a data packet.



Network Address Translation, or **NAT**, implies a **translation of an IP address to another IP address**.

NAT in and of itself only affects the L3 header, which in today's world will be the IPv4 header. While NAT can modify an IPv6 header as well, it really isn't common, as IPv6 was created in such a way to avoid the need for NAT altogether.

Port Address Translation, or **PAT**, implies a **translation of an IP address and Port to another IP address and Port**.

PAT affects *both* the L3 header and the L4 header. Which means the IPv4 Header, as well as *either* the TCP or UDP header, will be modified.

You could consider PAT as a subset of NAT (i.e., Network Address Translation along with a Port translation), but there isn't really a common use case for a Port translation only without an accompanying IP address translation as well. Therefore, nearly every instance of a PAT will also typically include an IP address translation as well.

In summary, a NAT modifies *only* the L3 header, and a PAT modifies *both* the L3 and L4 header. Or, said another way, a NAT modifies only the IP, and a PAT modifies *both* the IP and Port.

Static vs Dynamic

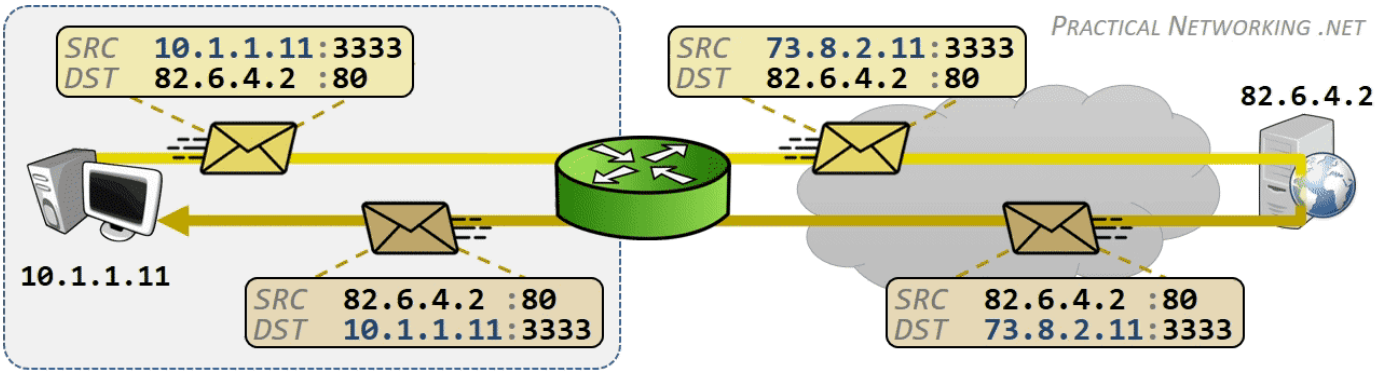
Both a NAT and a PAT can exist in two forms: Static or Dynamic. **These two terms designate whether the *post*-translation attributes of the packet are explicitly defined** by the administrator or determined by the translation device.

In either case, the *pre*-translation attributes are explicitly defined. This is how the NAT device knows which packets should be translated in the first place.

To help define the terms, we will look at an example of a Static translation and Dynamic translation below. The examples will use a Router as a NAT device, but many other devices can also perform address translation (Firewalls, Load Balancers, etc).

Static Translations

In a **Static** translation, the ***post*-translation attributes are explicitly defined** by the administrator (`IP` address for a NAT, or `IP:Port` for a PAT). A Static translation implies the pre-translation `IP` or `IP:Port` will *permanently* map to the same, constant post-translation `IP` or `IP:Port`.



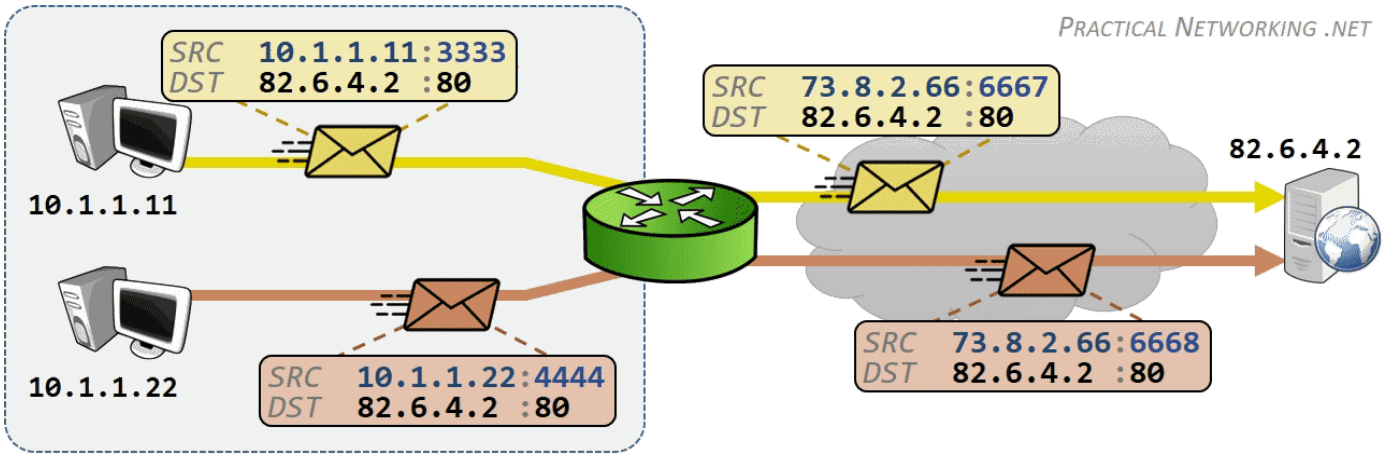
In this example, only the IP address is changing (NAT), and the mapping between pre-translation and post-translation is permanent (Static) – the IP address **`10.1.1.11`** will always be translated to **`73.8.2.11`** (and *vice versa*). Hence, this is an example of a **Static NAT**.

A **Static** mapping is sometimes referred to as a **One-to-One** translation – implying that in a Static translation, a single `IP` or `IP:Port` can only ever appear as another single `IP` or `IP:Port`.

Dynamic Translations

In a **Dynamic** translation, **the post-translation attributes are selected by the router *at the time that the packet is received*** – the final post-translation attributes are not permanently mapped to pre-translation attributes.

Of course, the scope of post-translation attributes must be defined by the administrator, but the exact mapping is determined by the device, at the time the packet is received.



In this example, both the IP address and the Port are changing (which makes this a PAT), and the mapping between the pre-translation and post-translation is not explicitly defined by the administrator (which makes this a Dynamic translation). Hence, this is an example of a **Dynamic PAT**.

When the packet from **`10.1.1.11`** arrived on the Router, the Router chose a new source port of **`6667`**. When the packet from **`10.1.1.22`** arrived on the Router, the Router chose a new source port of **`6668`**. Both hosts are sharing the public IP address **`73.8.2.66`**.

There is no guarantee that the *next* connections initiated by either host will have port numbers of `6667` or `6668` – they will very likely be something else randomly chosen by the Router, at the time the next packet is received by the Router.

A **Dynamic** mapping is sometimes referred to as a **One-to-Many** or **Many-to-One** translation – implying that in a Dynamic translation, many addresses can appear as one, or one address can appear as many.

Both Static NAT and the Dynamic PAT will be explored in more detail in later articles in this series.

Combining Terms

In total, we discussed two sets of two terms: NAT and PAT, and Static and Dynamic. When these are combined, they create four possible variants of Network Address Translation:

- Static NAT
- Static PAT
- Dynamic PAT
- Dynamic NAT

Each of the four combinations above account for every type of Network Address Translation that exists. They will also each be the subject of their own article in this series.

Static NAT

According to the definitions outlined in the NAT Terminology article, a **Static NAT** implies a translation of just the IP address, where the post-translation IP addresses are explicitly defined.

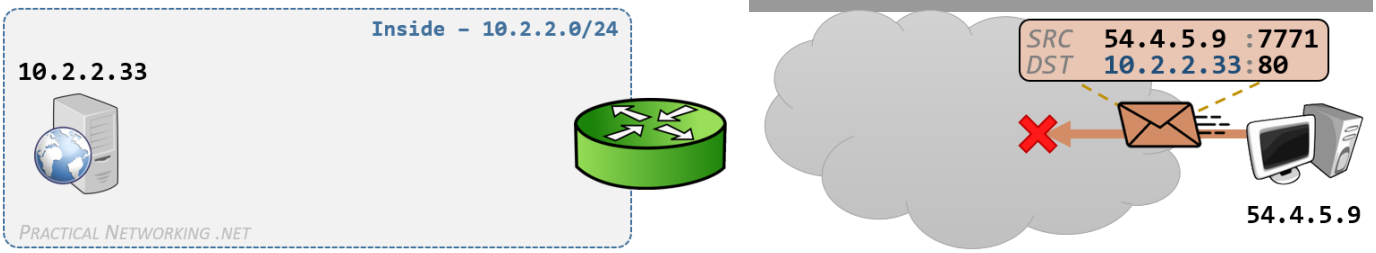
This article is a part of a series on Network Address Translation (**NAT**). Use the navigation boxes to view the rest of the articles.

Network Address Translation

- Why NAT?
- NAT Terminology
- **Static NAT**
- Static PAT
- Dynamic PAT
- Dynamic NAT
- Policy NAT and Twice NAT
- NAT Terminology Disambiguation

Making Internal Resources Accessible

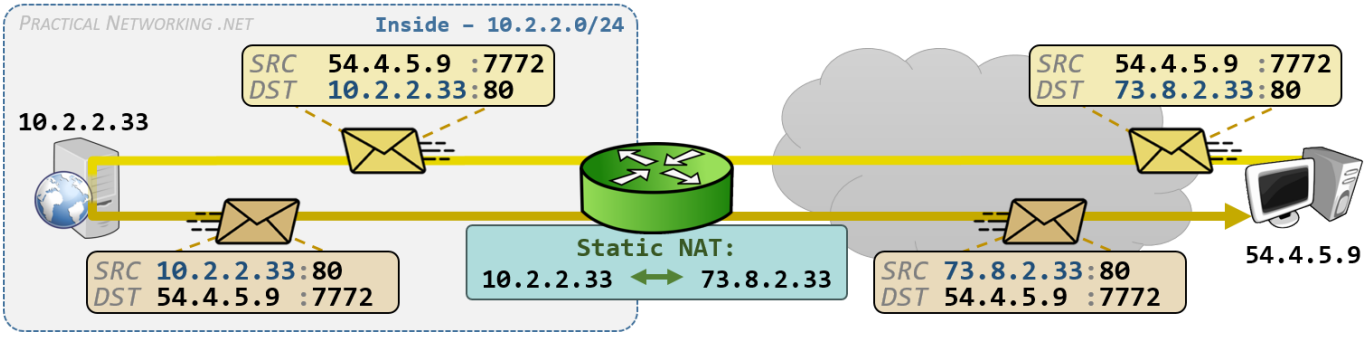
The typical use case for a Static NAT is for a Server on a Private IPv4 network to be reached externally from the Internet.



In this example, the Internet host **`54.4.5.9`** needs to connect to the web server on the Inside network. The web server is on an internal network and is therefore configured with the Private IP address of **`10.2.2.33`**.

If the Internet host attempts to send a packet to the IP address of the server (`10.2.2.33`), the packet will be dropped when it reaches the Internet. Recall, Private IP addresses are not routable on the Internet.

For a host on the Internet to reach the server, a Static NAT must be configured on the NAT device. In our example, the Router in front of the `10.2.2.0/24` network will be the NAT device, and we will configure it to translate the private IP address **`10.2.2.33`** to the Public IP address **`73.8.2.33`**.



Now, the Internet host can send a packet to the correlating Public IP address (`73.8.2.33`) which will be routed through the Internet to the NAT device. The Router (acting as our NAT device) will then translate the packet to the Server's private IP address (`10.2.2.33`). When the web server responds, the router will un-translate the packet back to the original IP address of `73.8.2.33`.

The Static NAT allowed the internal host with the private IP address to be accessed by an external host.

With that in mind, there are three additional points that must be made regarding Static NAT.

Source or Destination

Whether the Source or Destination of the packet is translated is dependent on the direction the packet is traveling. The **inbound packet has its *Destination IP* translated** (from the Internet to the server). The **outbound packet has its *Source IP* translated** (from the server to the Internet).

Either way, the *one* IP address `10.2.2.33` always maps to the *one* IP address `73.8.2.33`. This is why a Static NAT is also sometimes called a *one-to-one* NAT.

Conserving IP Addresses

If you had 30 servers on the Inside network, each with their own Private IP address, and you wanted to use Static NAT, then you would need 30 unique Public IP addresses for the translations.

We discussed earlier that the original intent of Network Address Translation was to conserve Public IPv4 addresses. However, as you can see, **a Static NAT does not actually conserve any Public IPv4 addresses**. Instead, the **primary purpose of a Static NAT is to expose a server with a Private IP address to the public Internet**.

Bidirectional

Finally, in the example above, the initial packet was sent from the Internet host. But it could have easily been sent from the server on the Inside network. Regardless of who initiated the connection, the Static NAT would cause the Source of the outbound packets or the Destination of the inbound packets to be translated.

The key point is that **a Static NAT translation is bidirectional**. Whether the internal host or the external host sent the first packet, it would “pass through” the Static NAT. There are variations of NAT which we will discuss later in this article series where the translation will not be bidirectional.

Static PAT

According to the definitions outlined in the NAT Terminology article, a **Static PAT** implies a translation of the IP address and Port, where the post-translation attributes are explicitly defined.

There are multiple use cases for a Static PAT, but they all have one thing in common – a need to manually change the TCP or UDP port as a packet moves through a router or firewall.

This article is a part of a series on Network Address Translation (**NAT**). Use the navigation boxes to view the rest of the articles.

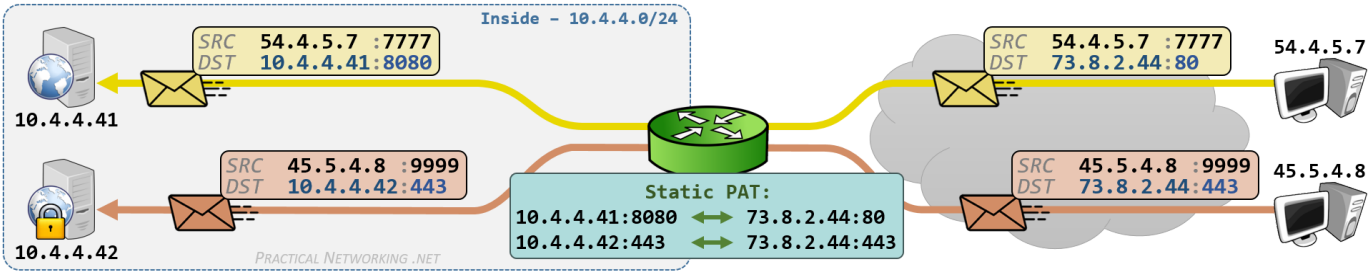
Network Address Translation

- Why NAT?
- NAT Terminology
- Static NAT
- **Static PAT**
- Dynamic PAT
- Dynamic NAT
- Policy NAT and Twice NAT
- NAT Terminology Disambiguation

Multiple Servers using one Public IP Address

One specific use case for Static PAT is to use a *single* Public IP address to host *multiple* services on *different* internal servers. This is in contrast with a Static NAT which would only allow you to use a single Public IP address to host *multiple* services on the *same* server.

This illustration will show how Static PAT can enable the single IP address **`73.8.2.44`** to host two different services (HTTP and HTTPS) using two separate internal servers (**`10.4.4.41`** and **`10.4.4.42`**):



The Router is acting as our translation device and is configured with two Static PAT entries. Since these are *static* translations, the mapping is explicitly defined: **`73.8.2.44:80`** will always be translated to **`10.4.4.41:8080`** and **`73.8.2.44:443`** will always be translated to **`10.4.4.42:443`**.

Two hosts somewhere on the Internet both make a request to the *same* IP address (`73.8.2.44`) – one request using **`HTTP`** (port `80`), one request using **`HTTPS`** (port `443`).

When their packets arrive on the Router, they get translated and sent to two *different servers* for processing.

The single Public IP address (**`73.8.2.44`**) is hosting two services (HTTP and HTTPS) served by two different internal servers (**`10.4.4.41`** and **`10.4.4.42`**).

If you use a Static PAT in this way (where one public IPv4 address is used to host multiple services on *multiple servers*), then you are conserving IPv4 address space. You could theoretically have 10 (or 50, or 200, or 1000+) different servers, each hosting a different service, all using a single Public IPv4 address.

Non-Standard Ports

The same illustration above also provides yet another use case for Static PAT – the **`10.4.4.41`** server is hosting HTTP traffic on a non-standard port (**`8080`**).

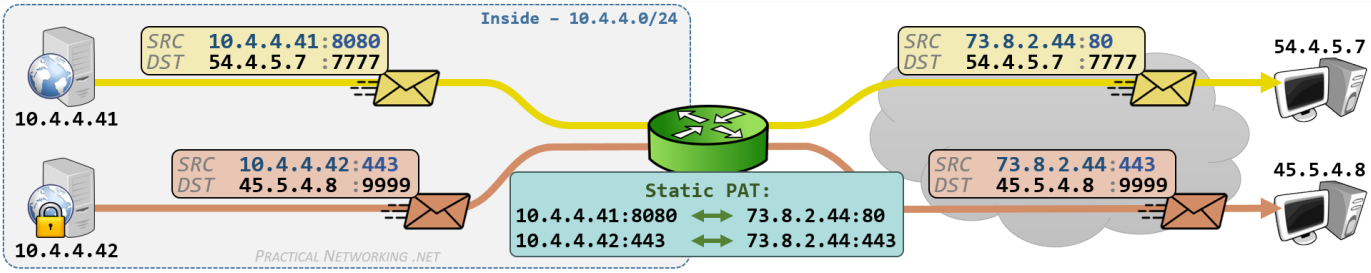
Without the port translation, hosts on the Internet would have to specify the non-standard port in their web browser by visiting "`www.site.com:8080`". Instead, with the Static PAT, the users can just type "`www.site.com`" (which implicitly assumes the standard web port of **`80`**) and the router automatically translates the packet to port **`8080`** instead.

This could work in reverse as well. Where a non-standard port is used on the outside, but a standard port is used on the inside server.

For example, the standard port associated with SSH traffic is `TCP/22`. Malicious users routinely scan the entire IPv4 address space for servers listening on port `TCP/22` to look for all SSH servers in hopes of finding some with weak passwords. A common strategy is to host SSH on a non-standard port in an effort to hide your SSH server from this mass scanning that occurs on port `22`.

Response Traffic

The response traffic from these hosts would be untranslated by the router. Since this is the outbound traffic, the source of the packet will be translated. Whereas on the inbound packet above, the destination of the packet was translated.



Once again, since the pre-translation `IP:Port` and post-translation `IP:Port` in a Static PAT are explicitly defined, the initial packet could have come from either the Internet hosts or the inside hosts. Therefore, a **Static PAT translation is bidirectional**.

Selectively Punching Holes

There is one final use-case for Static PAT, which is possibly the least common of the three. A Static PAT allows you to selectively "punch holes" through a particular Public IP address.

When we looked at a Static NAT, only the IP address is translated – the port numbers are left untouched. Which means, *every* port is explicitly mapped to *every* port. It's possible that you may want only ports `80` and `443` to get through, but not port `21` or `23` (or any other). A Static NAT does not allow you to choose.

Whereas instead, with a Static PAT, if you only create a translation for port `80` and `443`, only those paths through the router will exist. Protecting your internal servers from unwanted access attempts.

In this context, Static PAT is sometimes referred to as **Port Forwarding**: a specific port on an external address is forwarded to a specific port on an internal address.

Admittedly, the same effect can be attained by creating a Static NAT and applying an access-list or security policy to only allow the desired traffic through. As such, this is *a* use-case for Static PAT, but by no means is it the only way to attain the same effect.

The most common usage of selectively punching holes is to create a *bi*directional path through *uni*directional NAT translation. This will make more sense when we discuss Dynamic PAT in the next article.

Dynamic PAT

According to the definitions outlined in the NAT Terminology article, a **Dynamic PAT** implies a translation of the IP address and Port, where the post-translation attributes are selected by the router.

Dynamic PAT is the most common of the types of address translation we will discuss in this article series. Dynamic PAT is used any time multiple internal hosts need to share a single public IP address.

This article is a part of a series on Network Address Translation (**NAT**). Use the navigation boxes to view the rest of the articles.

Network Address Translation

- Why NAT?
- NAT Terminology
- Static NAT
- Static PAT
- **Dynamic PAT**
- Dynamic NAT
- Policy NAT and Twice NAT
- NAT Terminology Disambiguation

On a small scale, this is exactly what your home Wi-Fi router does. You may have 5-25 unique devices on your home network, each of them with their own private IP address. But when any of them try to speak with the Internet, they all share the single, unique public IP address assigned to your router.

The same type of translation happens with the Wi-Fi at coffee shops, or restaurants, or airports. This was the exact same example that was provided in the "*Why NAT?*" article – the illustrations are examples of a Dynamic PAT.

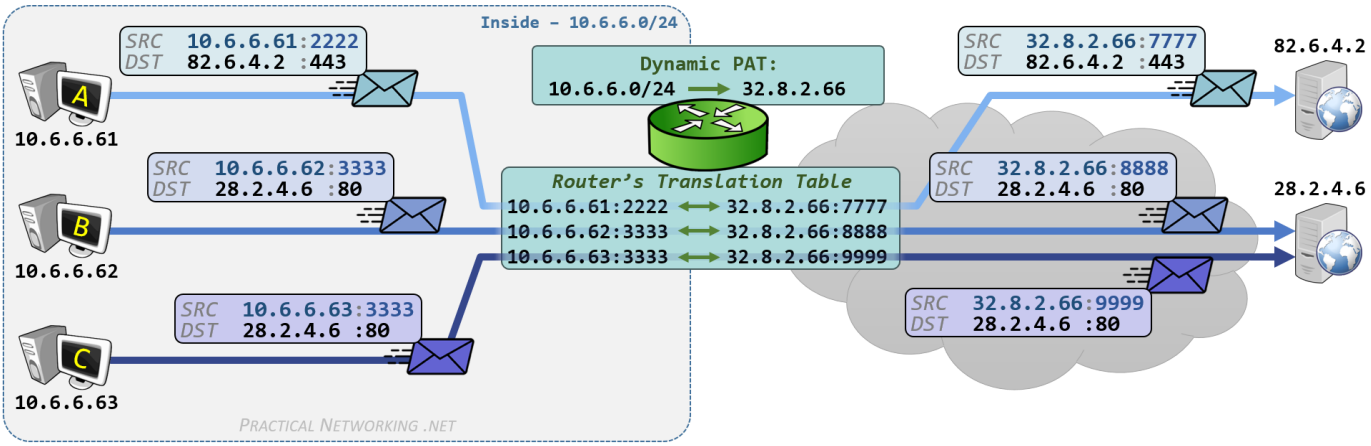
Of all the types of Network Address Translation, a Dynamic PAT is the most conducive to conserving IP address space. It is not uncommon to have hundreds of internal hosts sharing one public IP address.

Dynamic PAT is often referred to as a *many-to-one* or *one-to-many* translation, implying the many hosts on the Wi-Fi network are sharing the one Public IP address on the Internet.

Of course, this simple example referred to earlier hasn't quite shown how ports are translated, or how the Router selected the post-translation attributes. To illustrate those concepts, we will have to look at the packet flow through a Dynamic PAT in more detail.

Packet Flow – Outbound Traffic

The image below illustrates what is occurring at the packet level:



The Router is serving as our translation device, and is configured with a Dynamic PAT which translates any IP address on the Inside network (`10.6.6.0/24`) to the IP address `32.8.2.66`. When packets are translated, the Router makes note of the attributes of the original *and* translated packet in the Router's Translation Table.

Hosts A, B, and C each send a packet. They each use their own, unique Private IP address as the Source IP address, and they each randomly select a Source Port.

There are approximately 60,000 port numbers that can be chosen, and it is entirely feasible for two different hosts to randomly select the *same* source port (as is the case with Host B and Host C above).

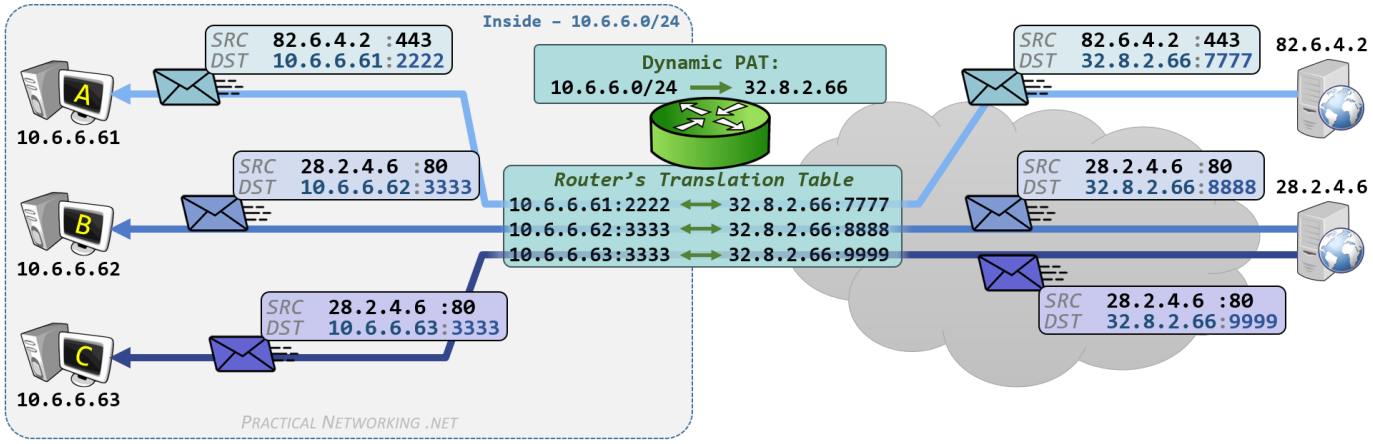
Notice the *configuration* of the Dynamic *PAT* does not include specifying a port number. Even though the ports are not explicitly set in the Router's configuration, this translation is still classified as a PAT because the port is dynamically changed by the Router.

In our example above, upon reception of each packet, the Router translates the source IP address of each packet to `32.8.2.66` (as explicitly configured), and randomly selects a new, unique source port number for each packet (`7777`, `8888`, and `9999`). The Router translated the port (PAT) and the Router selected the new source port (Dynamic).

Each specific mapping is recorded in the Router's Translation Table. This translation table will be used to "un-translate" the response packets when they return from the Internet.

Packet Flow – Response Traffic

When the two webservers respond to the three packets illustrated in the example above, the packet flow will resemble the following:



The response traffic from the web servers simply reverses the source and destination from the initial packet. Each web server sends the response traffic to the destination of the shared IP address (`32.8.2.66`), with the destination port number which the Router had selected in the original outbound traffic.

When the packets arrive on the Router, it matches them against the translation table to know how to "un-translate" the packet to their original attributes to get them to the appropriate host:

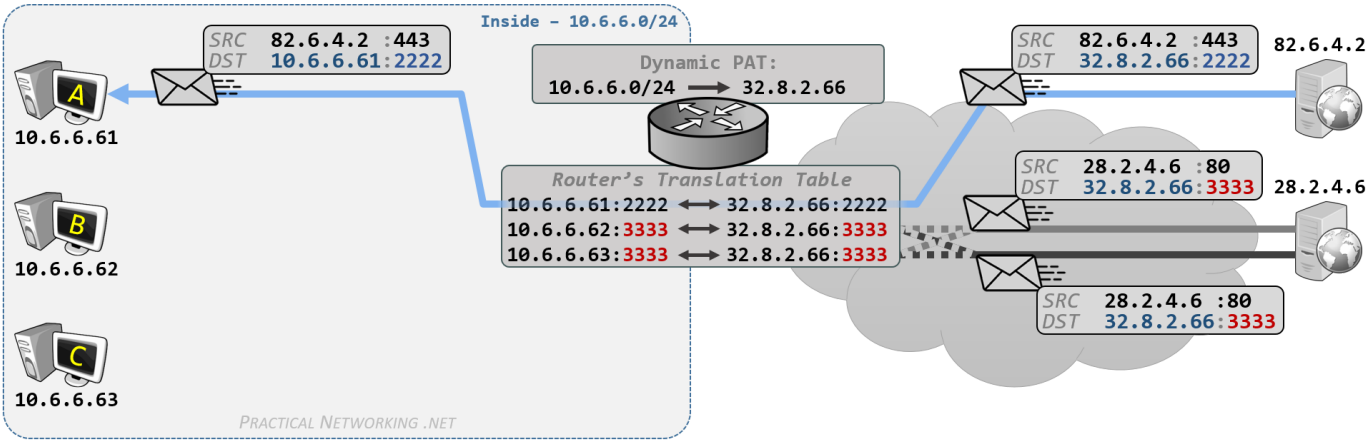
- The response packet sent to `32.8.2.66:7777` is forwarded to Host A (`10.6.6.61:2222`)
- The response packet sent to `32.8.2.66:8888` is forwarded to Host B (`10.6.6.62:3333`)
- The response packet sent to `32.8.2.66:9999` is forwarded to Host C (`10.6.6.63:3333`)

Why was the source port re-randomized?

In the last section, we pointed out that the router selected a new, random source port for the outbound packet. This re-randomizing of the source port is *crucial* to enabling successful communication through a Dynamic PAT.

Had the router *not* re-randomized the source port number, the outbound post-translation packets from Host B and Host C would have looked identical – they both would have had a Source IP of **`32.8.2.66`** and a Source port of **`3333`**.

Which means the response traffic for *both* packets from the **`28.2.4.6`** server would have looked identical – the Destination IP would have been **`32.8.2.66`** and the Destination port would have been **`3333`**.



When the identical packets arrive, the router would have no way of distinguishing which packet should be untranslated to Host B (`10.6.6.62`) or which should be translated to Host C (`10.6.6.63`). The router would have no choice but to drop both packets.

This would cause packets to drop anytime two hosts happen to pick the same source port, which happens often enough that no host would be content with the connectivity (or lack thereof) provided through a Dynamic PAT.

For this reason, it is *imperative* that the Router ensures every packet sent through a Dynamic PAT uses a unique source port number. This allows the return packets to be distinguishable from one another, and allows the Router to forward the return traffic to the appropriate host.

Some NAT devices assure unique source ports by re-randomizing the source port for *all connections* when doing a Dynamic PAT translation. Some NAT devices do this by re-randomizing the source port only when duplicate ports are chosen by the inside hosts.

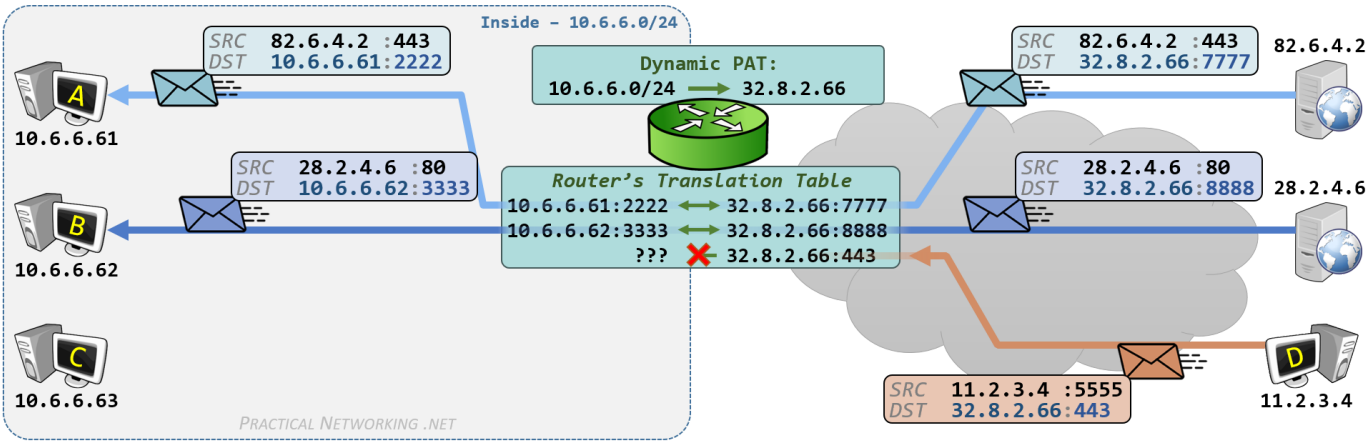
Regardless of the method used, so long as each connection's packets can be identified by both unique IP Address and Port, for both Source and Destination, the response traffic can be successfully un-translated to the appropriate initial host.

Unidirectional

As discussed before, a Dynamic PAT allows many internal hosts to share the same the same public IP address. One of the side effects of multiple hosts sharing a single IP address is the translation only works in one direction.

In the example above, Hosts A, B, and C initiated some traffic to external hosts. When the external hosts *responded*, the Router had entries in its translation table which allowed it to "un-translate" the packets and send them to the appropriate hosts.

If, however, a *new connection was initiated from an external host* and destined to the shared IP address, the router will have no way of knowing which internal host was the intended target of the packet.



Not knowing whom to deliver the packet to, the Router has no choice but to drop the packet. As such, **a Dynamic PAT only succeeds if the *internal* host sends the first packet**. If the *external* host sends the first packet, it will be dropped when it reaches the translation device.

This is what is meant by a Dynamic PAT being a *unidirectional* translation – traffic will flow through a Dynamic PAT only if the internal host initiates the connection.

This is in contrast to Static NAT and Static PAT, which are both bi-directional – traffic can be translated whether it was initiated by the external host or the internal host.

Keep in mind, this is not a "feature" of Dynamic PAT so much as it is a "side effect" of multiple hosts sharing a single IP address. Since it is possible for hosts to pick identical source ports, the router must change the source port during the translation, which means the packets arriving from the Internet can only make it back through the Dynamic PAT due to the entry in the translation table, which a packet initiated from an external network would not have.

If there is a need for *certain* ports to be accessible through a *shared* IP address, this can be achieved by using a Static PAT to selectively punch holes through the shared address of a Dynamic PAT.

Dynamic NAT

According to the definitions outlined in the NAT Terminology article, a Dynamic NAT implies a translation of just the IP address, where the post-translation attributes are selected by the router.

In a Dynamic NAT, a multitude of hosts with private IP addresses can share an equal or fewer amount of public IP addresses.

It may seem very similar to a Dynamic PAT, but the major difference is this is a *NAT* – the port number is not changing, only the IP address. Which means a single public IP address cannot be shared among multiple internal Hosts at the same time (as occurs with a Dynamic PAT).

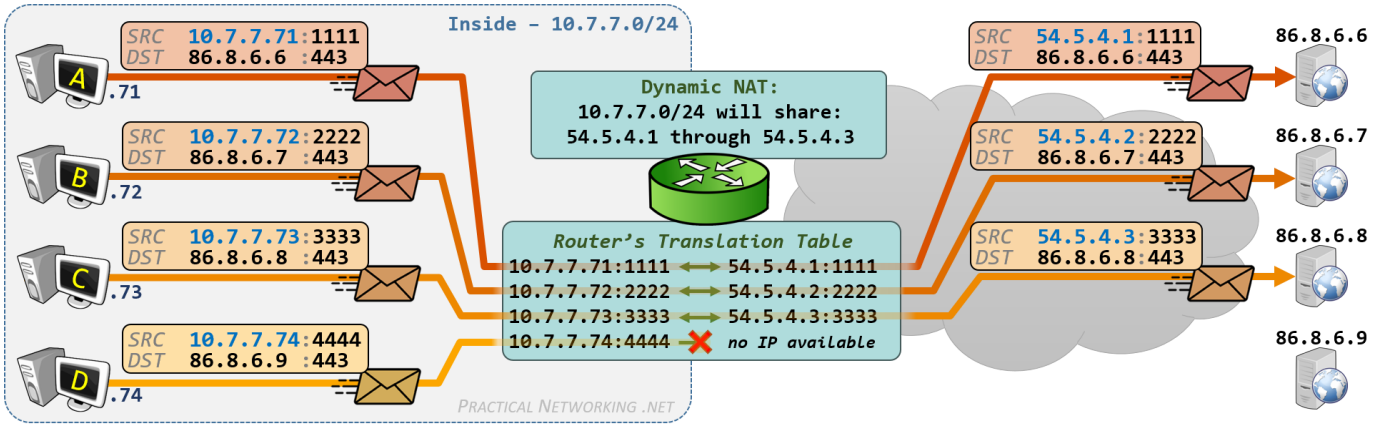
It is best explained with an illustration.

This article is a part of a series on Network Address Translation (**NAT**). Use the navigation boxes to view the rest of the articles.

Network Address Translation

- Why NAT?
- NAT Terminology
- Static NAT
- Static PAT
- Dynamic PAT
- **Dynamic NAT**
- Policy NAT and Twice NAT
- NAT Terminology Disambiguation

Dynamic NAT Illustration



In the image we have a Router with an Inside network (**`10.7.7.0/24`**) with four hosts (**`.71`**, **`.72`**, **`.73`**, **`.74`**). The Router is configured with a Dynamic NAT which states the hosts on the Inside network can share three public IP addresses: **`54.5.4.1`**, **`54.5.4.2`**, and **`54.5.4.3`**.

Host A (**`10.7.7.71`**) initiates a connection to **`86.8.6.6`**, and the Router assigns Host A the public IP **`54.5.4.1`**.

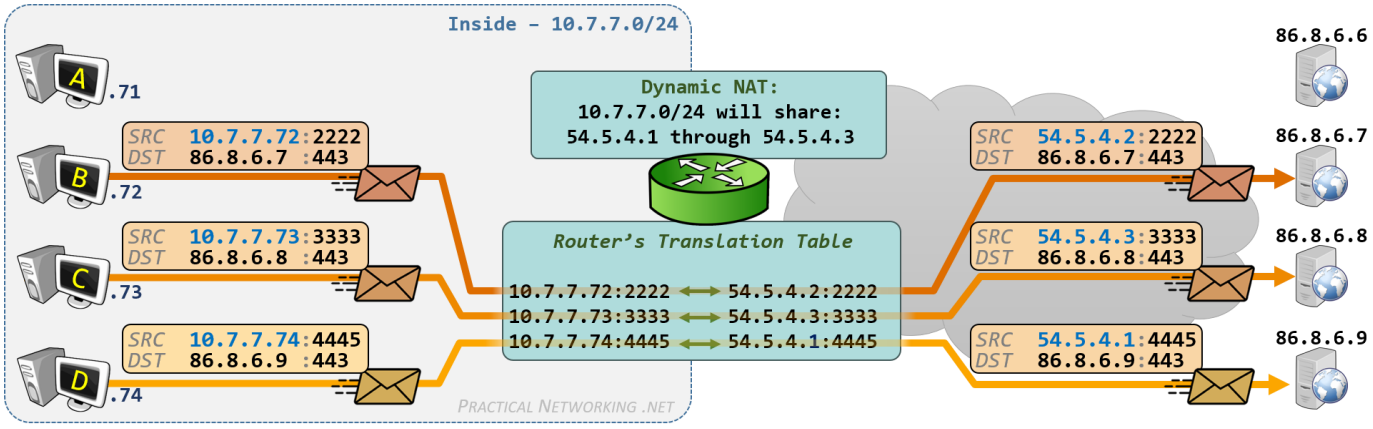
Host B (`10.7.7.72`) initiates a connection to `86.8.6.7`, and the Router assigns Host B the public IP `54.5.4.2`.

Host C (`10.7.7.73`) initiates a connection to `86.8.6.8`, and the Router assigns Host C the public IP `54.5.4.3`.

At this point, all the shared IP addresses have been used. Because of this, when Host D (`10.7.7.74`) attempts to initiate a connection to `86.8.6.9`, the packet is dropped because there are no available public IP addresses the router can use to translate Host D's private IP address.

While Host A/B/C have active connections through the Dynamic NAT, communication to those hosts are Bidirectional. Which means any host on the Internet can send packets to `54.5.4.1`, `.2`, and `.3` to reach Host A/B/C, respectively. We will expand on this in a moment.

When Host A is finished with its connection, the IP address it was assigned (`54.5.4.1`) becomes available again for the next internal host to use:



Here, we see Host D can now initiate a connection through the Dynamic NAT and receives the next available IP address.

In all cases, since this is a Dynamic *NAT*, only the IP address changed – the source port picked by the internal host remains the source port in the packet after translation.

Additionally, a Dynamic NAT has the potential to conserve IP addresses if configured as above where multiple internal hosts are sharing fewer Public IP addresses. However, you'll see in a moment that Dynamic NAT is not always configured in that fashion.

Benefits and Use Cases for Dynamic NAT

The main use case for a Dynamic NAT is that while the translation is active it has the benefit of being bidirectional, just like a Static NAT.

For example, in the images above, Host B (`10.7.7.72`) has an active connection and was assigned the public IP address `54.5.4.2`. So long as the connection is active in the Router's translation able, any host on the Internet can send packets to `54.5.4.2` and they will reach Host B.

In a way, a Dynamic NAT assigns a temporary "dedicated IP" to each internal host (so long as IP addresses are available). Or, said another way, a Dynamic NAT creates a temporary Static NAT.

There are two primary use cases for Dynamic NAT. The first is to allow for protocols which create a secondary, dynamic connection back to the client. The second is if you need a Bidirectional mapping of Private IPs to Public IPs, but don't particularly care about the explicit mapping between the two.

File Transfer Protocol and Dynamic NAT

The initial intent of a Dynamic NAT was to allow for protocols which create a second, dynamic connection back to the client. The main example of which is the File Transfer Protocol, or FTP.

FTP clients initiate outbound connections to FTP servers over destination port `TCP/21`. This connection serves as what FTP considers the *control channel*.

Over the control channel, a FTP client makes a request for a file and provides a random port number to the Server. The FTP Server then *initiates a second connection back to the client* from source port `TCP/20`, to the destination port *provided by the client in the control channel*. It is over this second connection that the file is actually transferred – this second connection is what FTP considers the *data channel*.

The issue is the data channel is a connection initiated from an external host on the Internet, destined to a host behind the Router. In a Dynamic PAT, which only allows connections initiated from the internal hosts, the data channel connection would be dropped.

But with a Dynamic NAT, the inbound data channel connection would be able to pass through the translation and the clients on the Inside server would be able to successfully use FTP to access files on the Internet.

The above describes the classic implementation of FTP known as *Active FTP*. There is a more modern implantation of FTP known as *Passive FTP* which does not require FTP clients to sit behind a Dynamic NAT, and instead allows them to sit behind the much more ubiquitous Dynamic PAT.

Dynamic Bidirectional Mappings

Beyond the case of dynamic protocols described above, one other usage for a Dynamic NAT is if you have an equal number of Public IP addresses as you do Private hosts, and don't particularly care which host get which public IP address, so long as each host gets one.

An example of such a case would be if the Router above could be configured to Dynamic NAT the entire `10.7.7.0/24` network into the entire `54.5.4.0/24` network. All 256 IP addresses in the Private range would receive an associated IP address on the Public range.

This would be the same effect of creating 256 individual Static NAT entries, except since the Dynamic NAT is *Dynamic*, there wouldn't be an explicit mapping of a Private IP to a Public IP. The Router would be choosing which Private addresses map to which Public addresses.

If a particular deployment doesn't necessarily care for a permanent, explicit mapping of private to public IP addresses, then Dynamic NAT could be used as a type of short cut to configuring 256 individual Static NAT entries.

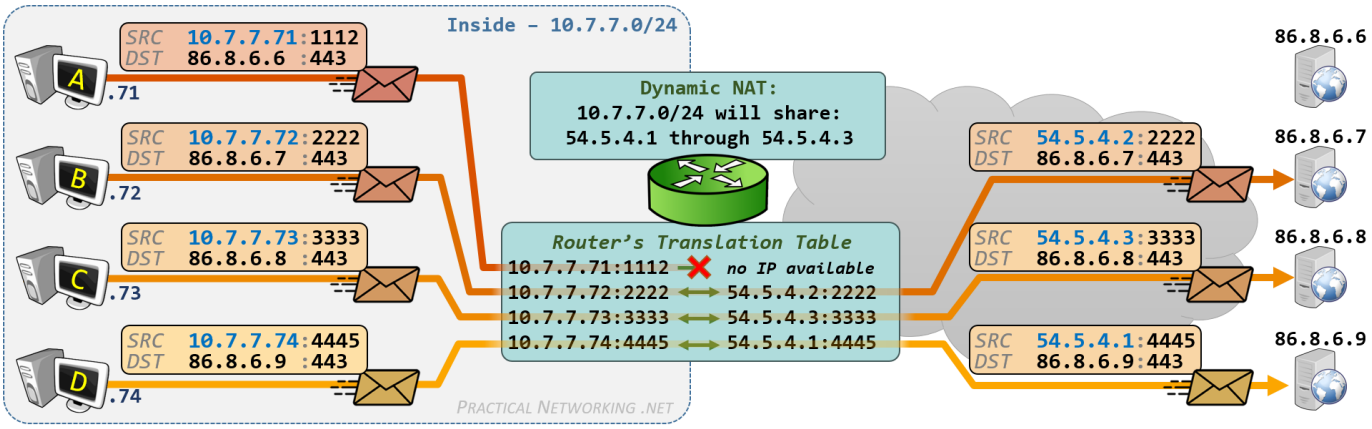
When configured in this manner, a Dynamic NAT does not actually conserve any IP addresses, since it would be necessary to have a public IP address for each private host.

Disadvantages of Dynamic NAT

Despite the potential use cases outlined above, in the grand scheme of things, a Dynamic NAT is the least common type of translation deployed. This is due to the mapping created by a Dynamic NAT being temporary by nature, and therefore inconsistent.

In the first illustration above, Host A/B/C received the IP addresses `54.5.4.1`, `54.5.4.2`, `54.5.4.3` respectively. A moment later, in the second illustration, Host A's connection terminated, and Host D received the IP address `54.5.4.1`. If a moment after that, Host A

attempted to communicate, there would be no available IP addresses and Host A's packet would be dropped:



From Host A's perspective, there was connectivity one moment, and no connectivity the next. This creates a generally poor experience for the user. And some of the most difficult for the network administrator to troubleshoot, as the connectivity issue is intermittent.

Of course, running out of available addresses and losing connectivity would only occur when there are less public IP addresses available in your translation pool than you have internal hosts – as is the case above with four internal hosts sharing three public IP addresses.

If you had a similar number of internal hosts and external IP addresses, as discussed in our second use-case example, you wouldn't run into the inconsistent connectivity problem. However, you would still run into the issue of inconsistent IP addresses.

For example, if there were no Host D in our illustration and there were just Hosts A/B/C sharing the IP addresses **`54.5.4.1`**, **`54.5.4.2`**, **`54.5.4.3`**. Host A may get **`54.5.4.1`** for the first connection, **`54.5.4.2`** for the next, and **`54.5.4.3`** for the third. At any given time, Host A would have connectivity, but there is no telling which public IP address Host A would receive at any given time.

Strictly speaking, this isn't intrinsically a bad thing if you are using a Dynamic NAT for the specific case described above where you don't necessarily need an explicit mapping.

But non-deterministic configurations can lead to unexpected and unintended results. So as a general rule in the Network Engineering field, deterministic designs are more favorable than non-deterministic designs.

Hence, if you have the public IP addresses available to give each of your private hosts a unique address, it is generally looked at as more favorable to configure multiple Static NAT translations instead of a single Dynamic NAT.

It should be noted that often when discussing address translation people will use the term *Dynamic NAT* when they actually mean *Dynamic PAT*. For the reasons mentioned above, Dynamic NAT is rarely used in production. If a single IP address is *shared* among many internal users, and if the *port number changes*, than it is indeed a Dynamic ***PAT***.

Policy NAT and Twice NAT

Every type of NAT we have discussed so far have two things in common. The first is that only the source of the packet is used to *make a NAT decision*. The second is that only the *source of the outbound packet* is translated. **Policy NAT** and **Twice NAT** are two ways of performing any type of NAT that expand beyond these two facts.

This article is a part of a series on Network Address Translation (**NAT**). Use the navigation boxes to view the rest of the articles.

Network Address Translation

- Why NAT?
- NAT Terminology
- Static NAT
- Static PAT
- Dynamic PAT
- Dynamic NAT
- **Policy NAT and Twice NAT**
- NAT Terminology Disambiguation

Summary of the types of NAT

First, let's quickly recap what we learned in the previous articles:

NAT vs **PAT** – these terms define whether just the IP address portion of the packet, or the IP address *and* Port number are being translated

Static vs **Dynamic** – these terms define whether the post-translation attributes are explicitly defined by the administrator, or ephemerally determined by the router.

When combined, this provides four possible variations of Network Address Translation:

- **Static NAT** – Translation of just the IP address where the administrator explicitly defines the IP address after translation
- **Static PAT** – Translation of the IP address and Port, where the administrator explicitly defines the IP address and Port after translation
- **Dynamic PAT** – Translation of the IP address and Port, where the router determines the new IP address and Port after translation
- **Dynamic NAT** – Translation of just the IP address, where the router determines the new IP address after translation

Decision Criteria

To configure each type of NAT above, we must define for the router exactly what traffic should be translated, and what it should be translated to.

If we review the configuration applied in the Static NAT or Dynamic PAT articles, we essentially instructed the Router to perform the following translations:

- If the *source IP address* is `10.2.2.33`, translate the source IP statically to `73.8.2.33`
- If the *source IP address* is `10.6.6.0/24` translate the source IP to `32.8.2.66` and dynamically pick a unique Source port

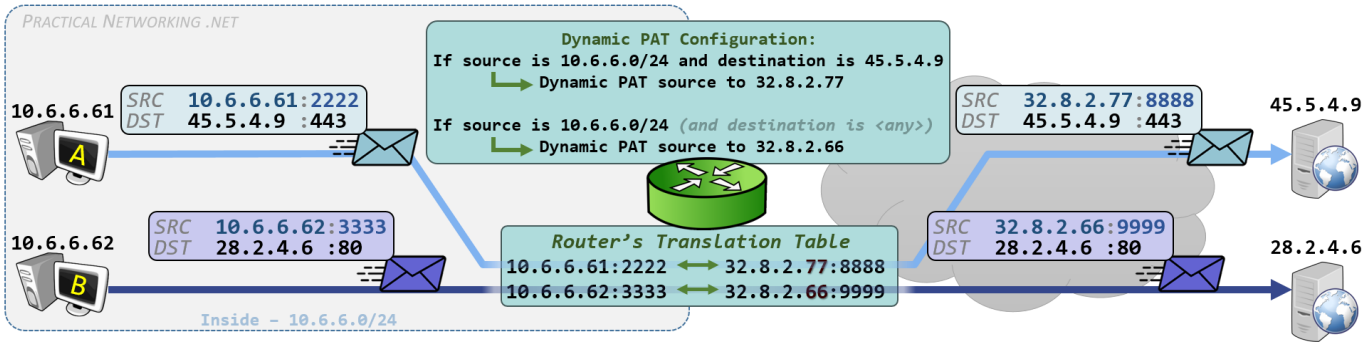
Notice in both cases we are making a decision to perform address translation based solely upon matching the *source IP address* of the packet – the *destination* address was not considered.

This is fine if you want all traffic from the Inside servers translated the same way for *every* destination they may speak to. However, there are times when you want to translate traffic to a certain destination one way, then translate traffic to a different destination a completely different way.

In such cases, when you need to conditionally translate traffic based upon the destination of the packet, you will need to use what is known as a Policy NAT.

Policy NAT

The following example is the same illustration as we used in the Dynamic PAT article, except we've added one additional, conditional translation to the configuration:



There are two parts to the Router's configuration. The first part of the configuration produces this behavior:

- If the *source IP address* is **`10.6.6.0/24`** *and the destination IP address is* **`45.5.4.9`**, translate the source IP using Dynamic PAT to the address **`32.8.2.77`**

The additional configuration tells the router to translate a packet based upon the criteria of matching *both* the *Source and Destination* of the packet. In the industry, this is referred to as a **Policy NAT**.

A **Policy NAT is simply any of the four NAT types we discussed prior in this article series, except the NAT decision requires matching both the *Source and Destination*** of a packet.

By contrast, every example of address translation thus far made a NAT decision based upon only the *source* of the packet.

The specific illustration immediately above was an example of a **Policy Dynamic PAT** – A translation decision based upon matching the source and destination of the packet (Policy), with the router determining the attributes after translation (Dynamic), which translated the source IP address and port (PAT).

The second part of the configuration produces this behavior:

- If the *source IP address* is **`10.6.6.0/24`**, *and the destination IP address is* **`<anything>`**, translate the source IP using Dynamic PAT to the address **`32.8.2.66`**

The second configuration item in the illustration above is simply a regular, Dynamic PAT.

Every traditional Dynamic PAT implies matching for *any* destination. Whereas the *Policy Dynamic PAT* in the first example would only match for *specific* destinations.

Twice NAT

In each example of the traditional four types of NAT we've discussed in this article series, only one "side" of the packet was being modified: the Source of the outbound packet or the Destination of the inbound packet.

Moreover, in the prior section we discussed Policy NAT: making a NAT *decision* based upon matching both the source and destination of traffic. However, even in a Policy NAT, once the decision was made, only one side of the packet was being modified.

If you refer back to the Policy Dynamic PAT example, when Host A (**`10.6.6.61`**) was speak to the Server, we translated **`10.6.6.61`** using a Dynamic PAT into **`32.8.2.77`**. Notice the

Server's IP address (`45.5.4.9`) was never translated, only the client's – only one side of the packet was changed (the source of the outbound packet).

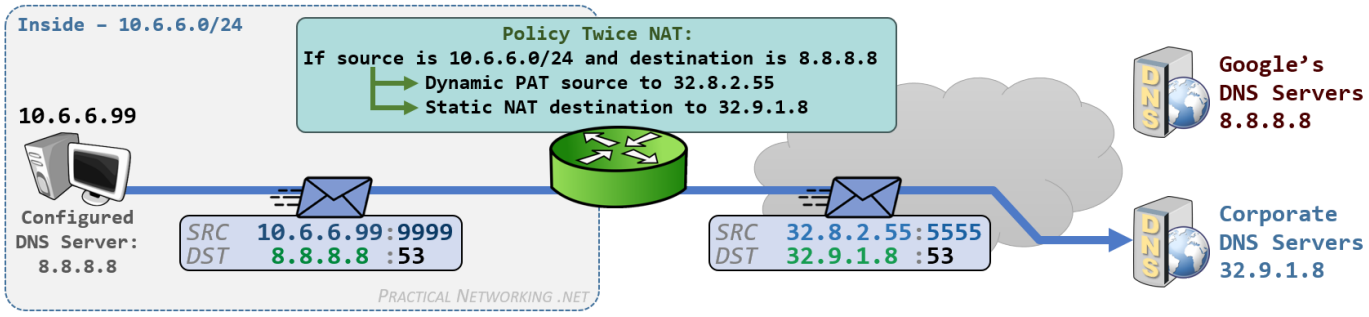
There are occasions where you need to translate *both* sides of the packet – this type of translation is referred to as a **Twice NAT**. The term comes from the fact that you are performing NAT twice: once on the source of the packet and another time on the destination of the packet.

There are many use cases for Twice NAT, we will provide one examples below. Another example will be illustrated in a separate article.

Changing the Destination with Twice NAT

At the core of it, a **Twice NAT is a type of NAT where both the Source and Destination of the packet will be translated**. Take this scenario as an example.

You are in charge of a Router with hosts on a private network (`10.6.6.0/24`) that have chosen to use Google's Public DNS Resolving Server (`8.8.8.8`). However, company policy states DNS requests must be made using the Corporate DNS server (`32.9.1.8`). One option is to manually verify every user's DNS configuration, but that does not scale. Instead, another option would be to translate any outbound requests to `8.8.8.8` into a request for `32.9.1.8`.



Notice the configuration is making a decision based upon matching a Source of `10.6.6.0/24` and a Destination of `8.8.8.8` – this makes the configuration a Policy NAT. Furthermore, the configuration is *translating the source* using a Dynamic PAT, *and the destination* using a Static NAT – this makes the configuration a *Twice NAT*, since we are doing *two instances* of *address translation*.

The packet sent by the host is sourced from a private IP address and destined to Google's DNS servers. But after crossing the router, the packet is now sourced from a public IP address and destined to the Corporate DNS servers.

The internal host is still configured to use Google's DNS servers, but their traffic is automatically being redirected to the corporate DNS servers. The internal host will not know that anything is different, and unless they go out of their way to validate the DNS responses, they will have no idea that the response is coming from the corporate DNS server and not Google's DNS server.

Summary of New Terms

In this article, we unpacked and compared the ideas of a **Policy NAT** and a **Twice NAT**. As a quick summary:

- A **Policy NAT** is any translation that occurs based upon *matching* both the Source and Destination of traffic.
- A **Twice NAT** is any translation that involves *translating* both the Source and Destination of traffic.

These two terms can be combined, giving us a Policy Twice NAT. Which is a type of NAT which makes a decision based upon the Source and Destination of a packet (Policy NAT), and translates both the Source and Destination of a packet (Twice NAT).

NAT Terminology Disambiguation

If you've made it here, then at this point you should have a solid understanding of each of the four types of translation that can exist: Static NAT, Static PAT, Dynamic PAT, and Dynamic NAT. Moreover, you have an understanding of Policy NAT and Twice NAT, which are simply two different ways of implementing the four types of NAT.

This article is a part of a series on Network Address Translation (**NAT**). Use the navigation boxes to view the rest of the articles.

Network Address Translation

- Why NAT?
- NAT Terminology
- Static NAT
- Static PAT
- Dynamic PAT
- Dynamic NAT
- Policy NAT and Twice NAT
- **NAT Terminology Disambiguation**

The definitions and examples provided in this article series encompass every type of address translation that can possibly exist. That said, for marketing efforts, most vendors use their own distinct terminology to refer to each type of NAT we have discussed.

NAT Disambiguation Table

	Static NAT	Static PAT	Dynamic PAT	Dynamic NAT
RFC 2663	Static Address Assignment	Realm Specific Address and Port IP	Network Address Port Translation (NAPT)	Basic NAT
Wikipedia	Full Cone NAT	Full Cone NAT	Symmetric NAT	Address Restricted Cone NAT
Cisco Routers	Static NAT	Static PAT	Dynamic PAT	Dynamic NAT
Cisco ASA/ASA-X	Static NAT	Static PAT	Dynamic PAT	Dynamic NAT
F5 LTM	NAT	Virtual Server	SNAT	n/a
Juniper	Static NAT	Static NAT with Port Mapping	Source NAT	Source NAT with Disable PAT Argument