Introduction

- Welcome to Jeremy's IT Lab, a free, complete course for the CCNA.
- This video will finish up the topic of NAT, Network Address Translation.
- NAT is topic 4.1 of the CCNA exam.
- In Day 44, NAT was introduced and static NAT was covered.
- This video will focus on dynamic NAT.
- After this video, you should have no problems answering any NAT questions on the CCNA exam.

What We'll Cover in This Video

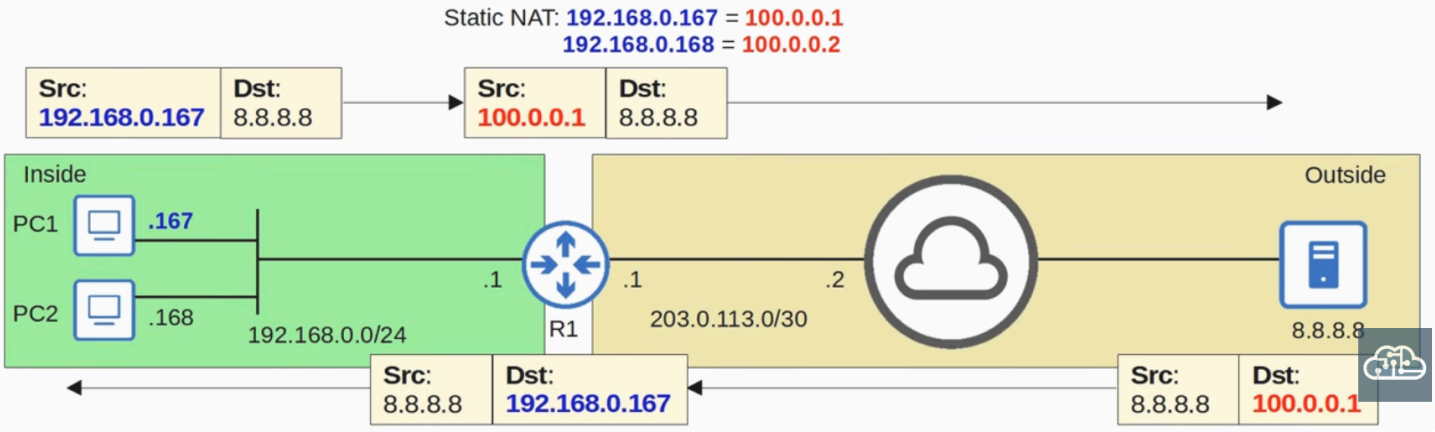
- One more point about static NAT that was not mentioned in the last video.
- Dynamic NAT, in which you don't manually map addresses one-to-one, but instead the router automatically makes the mappings for you.
- Another very important type of NAT: dynamic PAT (Port Address Translation), which translates not only the IP address, but also the port number.
    - Dynamic PAT is extremely useful because it allows many devices to share a single public IP address.
- A bonus practice question from Boson Software's ExSim for CCNA — watch until the end of the video.

Review of Static NAT

- Static NAT involves statically configuring one-to-one mappings of private IP addresses to public IP addresses.

- When traffic from the internal host is sent to the outside network, the router will translate the source address.

Example: Static NAT Mapping Setup



- Inside local address 192.168.0.167 is mapped to inside global address 100.0.0.1.
- Inside local address 192.168.0.168 is mapped to inside global address 100.0.0.2.

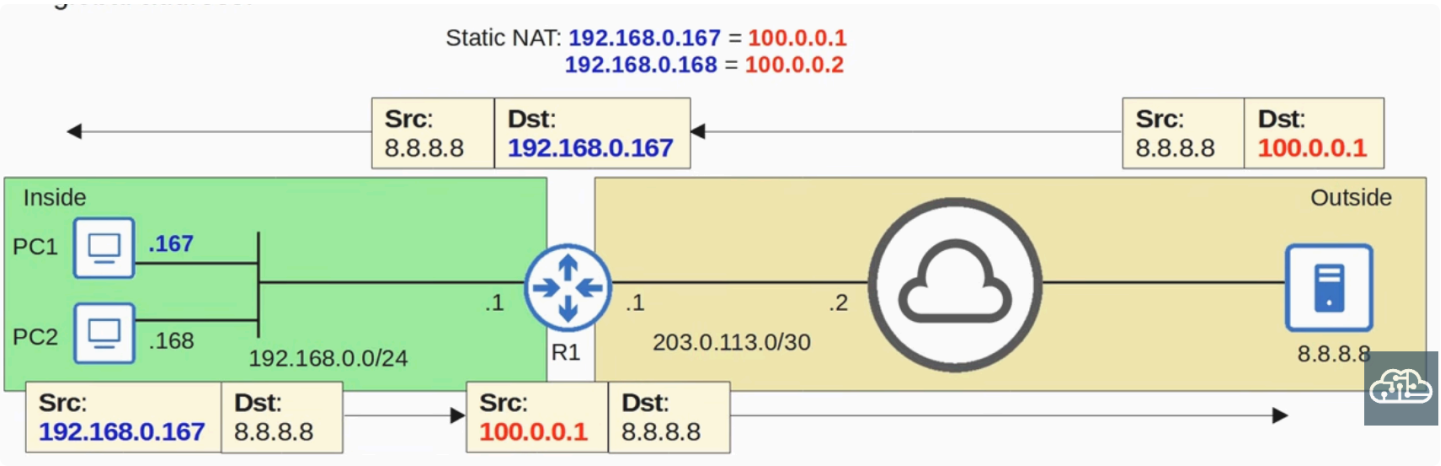
Example: PC1 Accessing the Server at 8.8.8.8

- PC1 wants to access the server at 8.8.8.8.
- PC1 sends a packet with source IP 192.168.0.167, a private IP address.
- R1 translates the source address to 100.0.0.1, a public IP address.
- When the server sends its reply, the destination is the public IP address.
- R1 translates the destination IP of the reply back to 192.168.0.167 and forwards it to PC1.

One Additional Important Point About Static NAT

- The one-to-one mapping of IP addresses doesn't only allow the internal host to access external resources.
- It also allows external hosts to access the internal host via the inside global address.

Example: External Host Initiating Communication



- Without PC1 initiating communication with the server, the server could send a packet to destination IP 100.0.0.1.
- Because of the one-to-one IP address mapping, R1 would translate it to 192.168.0.167 and forward it to PC1.
- PC1 would then reply.
- So, static NAT works two ways: not just from inside to outside, but also from outside to inside.
- Note: This point was not mentioned in Day 44.

Moving On to Dynamic NAT

- The professor now moves on to the topic of dynamic NAT.

Dynamic NAT

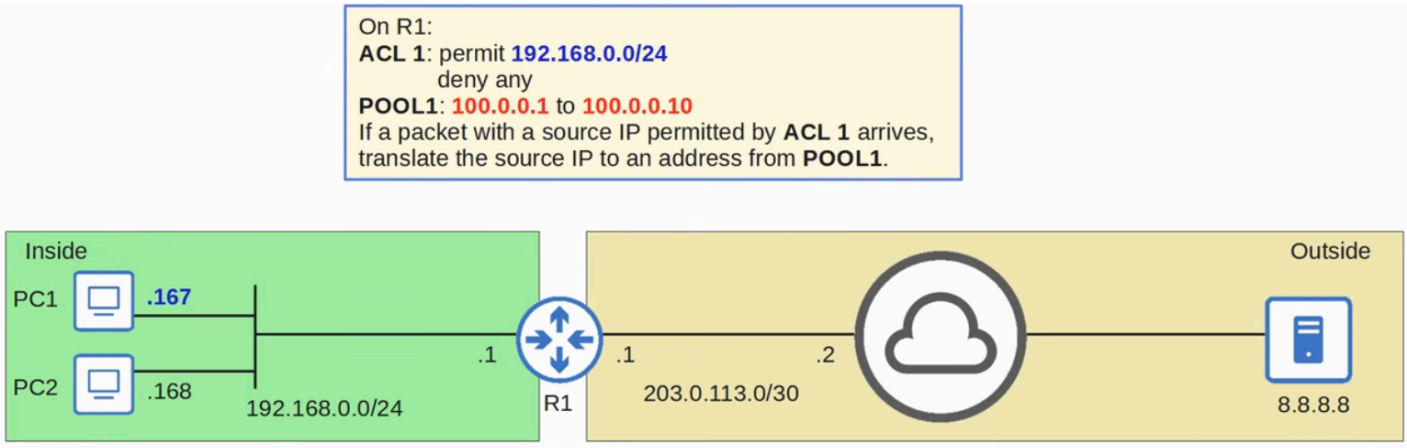
- In dynamic NAT, the router dynamically maps inside local addresses to inside global addresses as needed.
- You don't manually specify "map this IP address to this IP address, that IP address to that IP address."
- The router makes those mappings automatically, and then clears the mapping when it is no longer needed.

How It Works in Cisco IOS

- An ACL is used to identify which traffic should be translated.

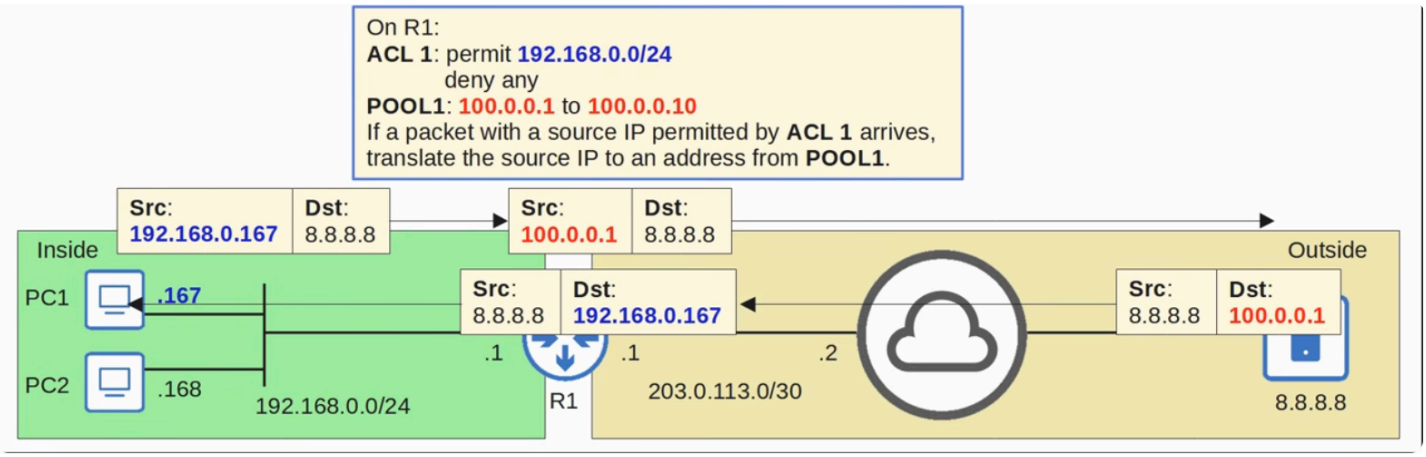
- This is a totally different use of an ACL, but it is a very common use.
- ACLs can be used to indicate which traffic should be forwarded and which should be blocked, but they can also be used just to identify traffic.
- If the source IP of a packet is permitted by the ACL, the source IP will be translated by NAT.
- If the source IP is denied by the ACL, the source IP will NOT be translated.
- However, that doesn't mean the traffic will be dropped — the ACL is simply being used to identify which traffic should be translated, not to drop traffic.
- The ACL will not be applied to an interface with the `ip access-group` command.
- A NAT pool is used to define the available inside global addresses that can be used for translation.

Example: ACL 1 and NAT POOL1 on R1



- On R1, ACL 1 is configured to permit 192.168.0.0/24 and deny all other traffic.
- NAT POOL1 is configured with a range from 100.0.0.1 to 100.0.0.10.
- If a packet with a source IP permitted by ACL 1 arrives, R1 translates the source IP to an address from POOL1.

Example: PC1 Sending a Packet



- PC1 sends a packet, and its IP address is permitted by ACL 1.
- R1 translates the source IP to an address from POOL1, 100.0.0.1.
- The server sends a reply, and R1 translates the address back.
- This looks like the same process as in static NAT.
- The difference is that, instead of manually configuring a mapping from 192.168.0.167 to 100.0.0.1, R1 did it automatically when the packet from PC1 arrived.
- Important point:
  - If the source IP address is not permitted by the ACL, it doesn't mean the packet will be dropped.
  - It just means it won't be translated.

Additional Points About Dynamic NAT

- Although the mappings are dynamically assigned, they are still one-to-one: one inside local IP address per inside global IP address.
- If there aren't enough inside global IP addresses available — if they are all currently being used — it is called "NAT pool exhaustion."

Example: NAT Pool Exhaustion

- In the previous example, the ACL specified a /24 subnet of inside local addresses, but the NAT pool only had 10 inside global addresses.
- So, there can only be 10 translations active at a time.

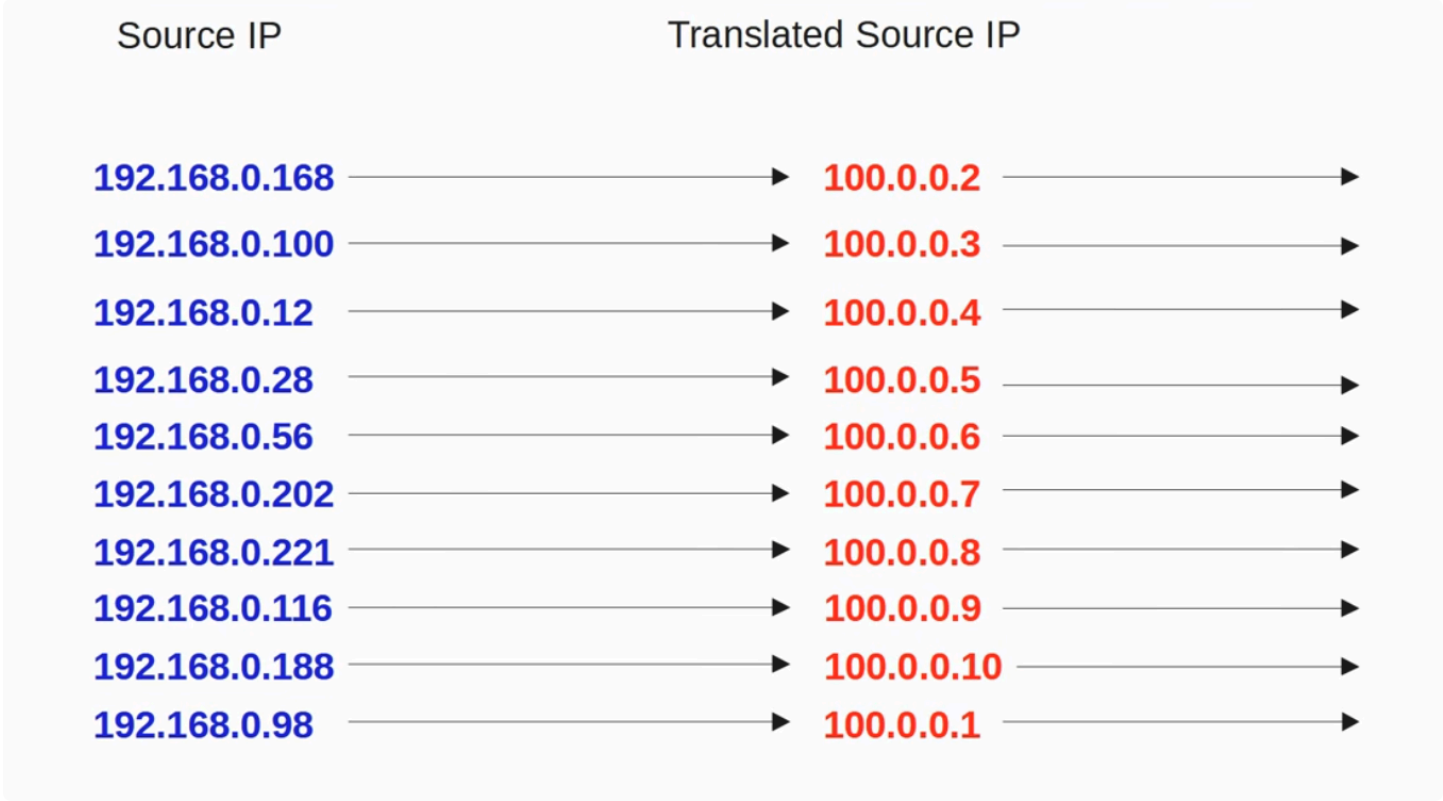
- What happens if there are no available addresses, but a packet arrives and needs NAT?
  - If a packet from another inside host arrives and needs NAT but there are no available addresses, the router will drop the packet.
  - The host will be unable to access outside networks until one of the inside global IP addresses becomes available.
- How does an address become available?
  - Dynamic NAT entries will time out automatically if not used.
  - Or you can clear them manually.

Example: Address Becoming Available

- In the previous example, if PC1 stops communicating with external hosts for a while, the inside global address 100.0.0.1 would become available again for a different host to use.
- You can use the `clear ip nat translation` command anytime to manually clear translations.

```
R1# clear ip nat translation
```

Demonstration of NAT Pool Exhaustion

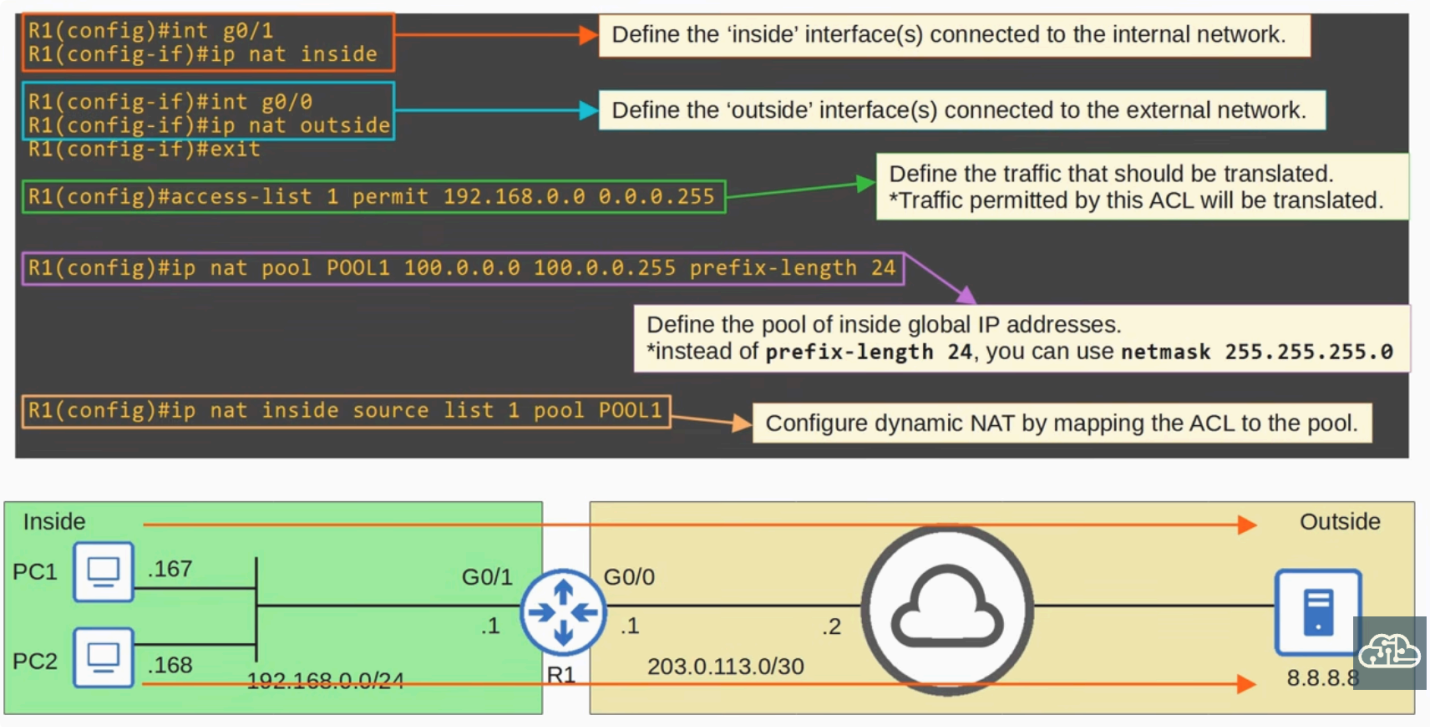


- A packet with source IP 192.168.0.167 arrives at R1.
  - It is translated to 100.0.0.1, and then forwarded.
- A packet with source IP 192.168.0.168 arrives.
  - It is translated to 100.0.0.2, and then forwarded.
- Some other hosts send traffic, and now those ten inside global IP addresses, 100.0.0.1 through .10, are all being used.
- If host 192.168.0.98 tries to send some traffic to the Internet:
  - There are no addresses available, so the router will drop the packet.
- But perhaps 192.168.0.167 stops communicating over the Internet.
  - After some time, its dynamic translation is removed.
- If 192.168.0.98 tries to send that traffic again:
  - The 100.0.0.1 address is now available, and it can reach the Internet.

Difference Between Static NAT and Dynamic NAT

- Although both are one-to-one mappings:
  - Static NAT mappings are permanent.
  - Dynamic NAT mappings are temporary; they will time out when they are no longer needed, and then another host can use that same public IP address.
- However, hosts are still unable to use the same public IP address at the same time.
- To do that, you have to use PAT, Port Address Translation.

Dynamic NAT Configuration



- First, just like in static NAT, you must define the inside and outside interfaces.
- Then define the traffic that should be translated.
  - Traffic permitted by this ACL will be translated.
  - This ACL tells R1 to translate all traffic from 192.168.0.0/24.

- Then define the pool of inside global IP addresses.
  - The command is `ip nat pool`, then the pool name.
  - After that you specify the first IP address in the range, and then the last IP address in the range.
  - Example: a range from 100.0.0.0 to 100.0.0.255.
  - Finally, you have to specify the prefix length, either with the `prefix-length` option or the `netmask` option.
  - IOS uses this to check if both addresses — the first and last address of the range — are in the same subnet.
  - If they aren't, the command will be rejected.

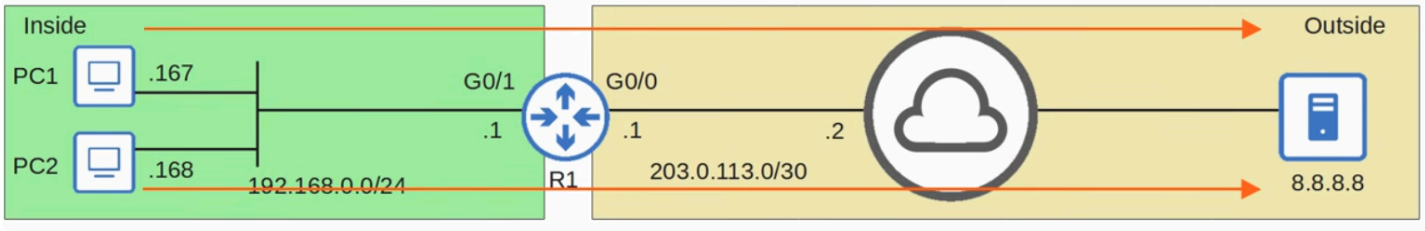
```
R1(config)# ip nat pool POOL1 100.0.0.0 100.0.0.255 prefix-length 24
```

- Finally, configure dynamic NAT by mapping the ACL to the pool with `ip nat inside source`.
  - Instead of `static`, use `list`, followed by the ACL number or name.
  - Then use `pool`, followed by the pool name.

```
R1(config)# ip nat inside source list 1 pool POOL1
```

- And that's all there is to dynamic NAT configuration.
- It's a bit more complicated than static NAT configuration, but it's not so difficult.

Example: Sending Traffic and Checking R1's Translation Table



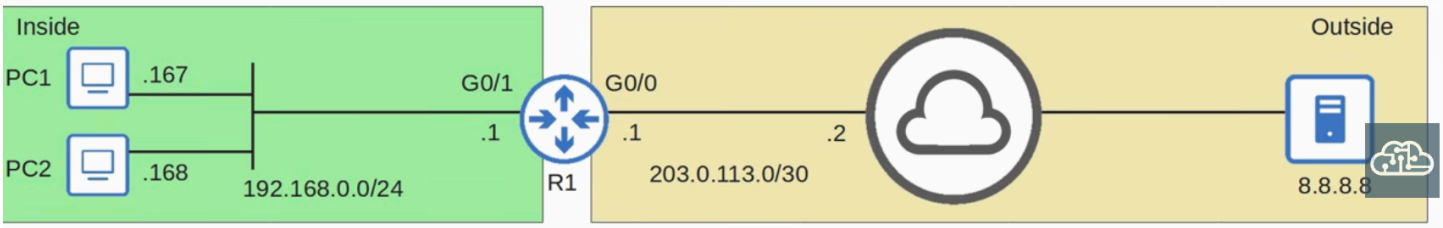
- With the configurations complete, the professor sent some traffic from PC1 and PC2 to 8.8.8.8.
- He then checks R1's translation table.

Viewing R1's NAT Translation Table

```
R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 100.0.0.1:3       192.168.0.167:3    8.8.8.8:3          8.8.8.8:3
udp 100.0.0.1:58685    192.168.0.167:58685 8.8.8.8:53        8.8.8.8:53
--- 100.0.0.1          192.168.0.167      ---                ---
icmp 100.0.0.2:3       192.168.0.168:3    8.8.8.8:3          8.8.8.8:3
udp 100.0.0.2:49536    192.168.0.168:49536 8.8.8.8:53        8.8.8.8:53
--- 100.0.0.2          192.168.0.168      ---                ---

! below is about 1 minute later

R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 100.0.0.1          192.168.0.167      ---                ---
--- 100.0.0.2          192.168.0.168      ---                ---
```



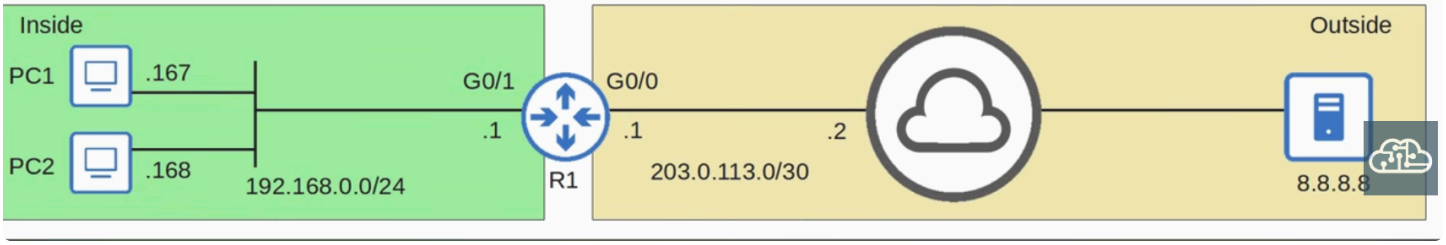
- After sending pings as well as DNS traffic to the server at 8.8.8.8, R1's translation table contains three entries for each mapping:
  - Three for 192.168.0.167 to 100.0.0.1.
  - Three for 192.168.0.168 to 100.0.0.2.
- When the router dynamically creates the inside local to inside global mappings, these entries are created.

- When those mappings are actually used — when translations are made — separate entries like UDP entries or ICMP entries are made.
- The UDP and ICMP entries will be cleared after about a minute.
- The original dynamic mappings themselves have a default timeout value of 24 hours.
- Each time a translation is made, the timer resets.
- So, dynamic NAT mappings actually last quite a long time.
- You can change the default timers if you want, but that's not something you need to know for the CCNA.
- Note: Although these entries look like the static NAT entries from Day 44, if you use the `clear ip nat translation` command, these entries will be cleared because they are not static — they are dynamic.

```
R1# clear ip nat translation
```

SHOW IP NAT STATISTICS and Key Show Commands

```
R1#show ip nat statistics
Total active translations: 6 (0 static, 6 dynamic; 4 extended)
Peak translations: 6, occurred 00:00:30 ago
Outside interfaces:
  GigabitEthernet0/0
Inside interfaces:
  GigabitEthernet0/1
Hits: 32  Misses: 0
CEF Translated packets: 20, CEF Punted packets: 12
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool POOL1 refcount 6
 pool POOL1: netmask 255.255.255.0
        start 100.0.0.0 end 100.0.0.255
        type generic, total addresses 256, allocated 2 (0%), misses 0
[output omitted]
```



- The professor now shows the output of `show ip nat statistics`.
- Note: He entered this command when all 6 of those translation entries were active, so it displays 6 total active translations.
    - 6 dynamic entries.
    - 4 extended.
- All of those entries were dynamic.
- The extended entries are those UDP and ICMP temporary entries that were cleared after a minute.
- The details of extended entries aren't something you need to know for the CCNA.
- What he really wants to point out: you can confirm the dynamic mapping from ACL 1 to POOL1 in the output.
- Just like with static NAT, the two SHOW commands you should know for dynamic NAT are `show ip nat translations` and `show ip nat statistics`.
- You should be familiar with both of these commands for the CCNA exam.

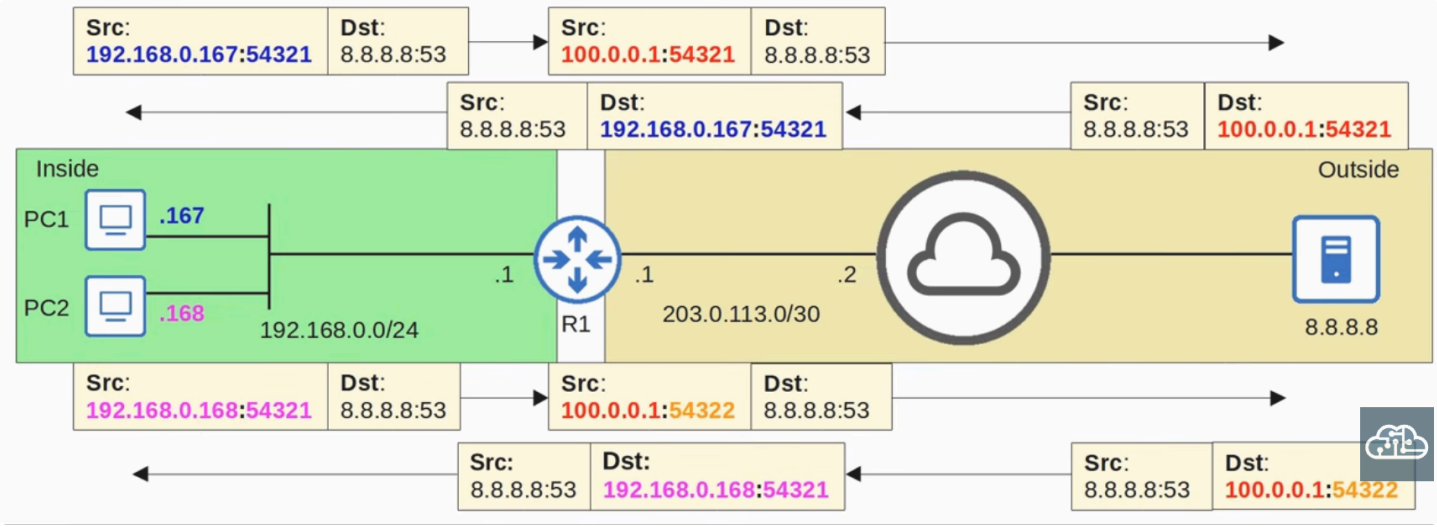
Verification:

```
R1# show ip nat statistics
R1# show ip nat translations
```

PAT (Port Address Translation) / NAT Overload

- PAT, Port Address Translation, is also known as NAT overload.
- It translates both the IP address and the port number of a packet, if necessary.
- Purpose of translating the port number:
    - By using a unique port number for each communication flow between internal hosts and external hosts, a single public IP address can be used by many different internal hosts.

- TCP and UDP port numbers are 16 bits in length, which means there are over 65,000 port numbers.
- The router will keep track of which inside local address is using which inside global address and port.
- The router keeps track of the communication flows by using unique port numbers for each flow.

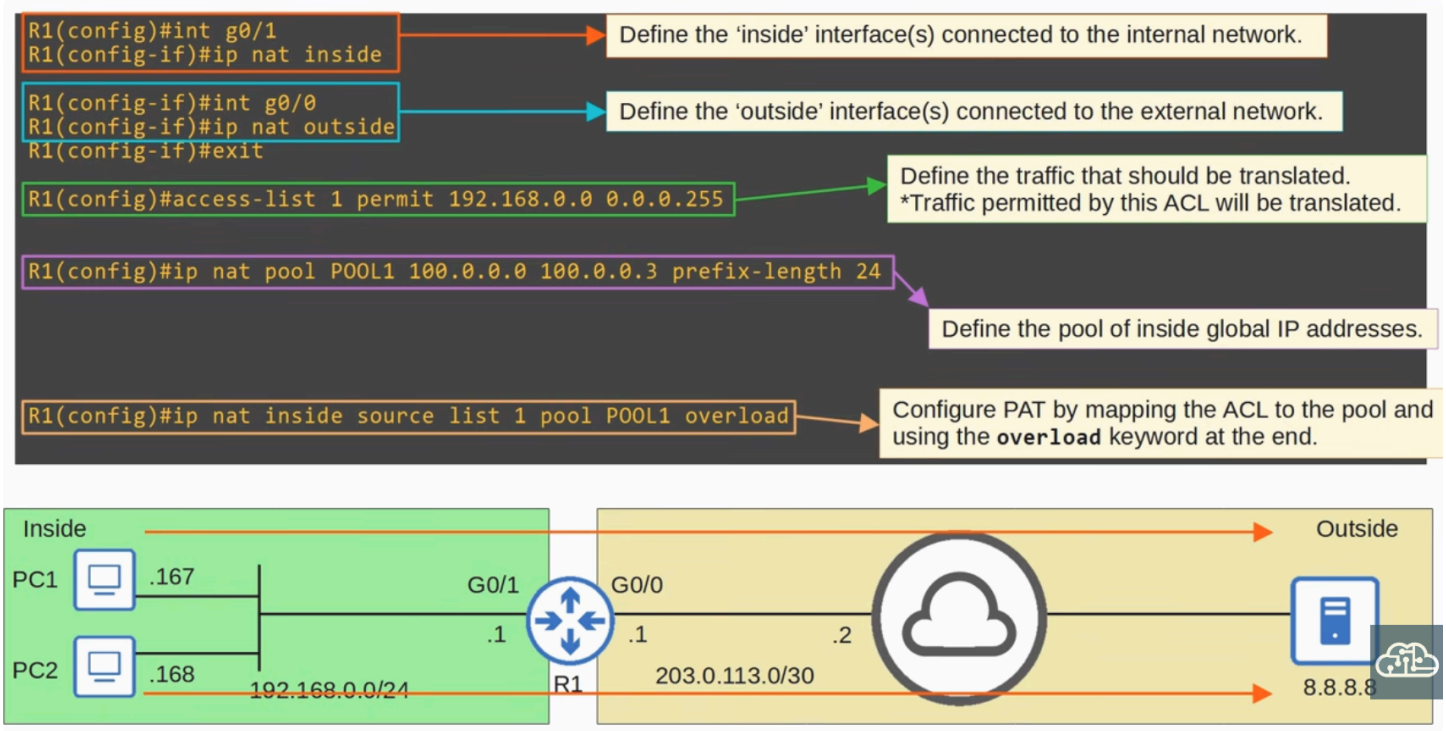


Example: PC1 and PC2 Sending DNS Messages to 8.8.8.8

- PC1 sends a DNS message to 8.8.8.8.
  - Source IP is 192.168.0.167.
  - It selects a random source port, 54321.
- PC2 also sends a DNS message to 8.8.8.8.
  - Source IP is 192.168.0.168.
  - It selects a random source port, and it happens to choose the same source port as PC1, 54321.
- R1 translates the source IP of PC1's message to 100.0.0.1, and keeps the source port the same, 54321.
- For PC2's message, R1 also translates the source IP to 100.0.0.1, but it changes the source port to 54322.

- Why does it do that?
  - It's so it can keep track of the communication flows.
  - If both flows use the same source IP and source port, when the replies come from the server, R1 won't know which reply to send to which PC.
- The server sends the replies to R1.
  - Both have destination IP 100.0.0.1.
  - One has destination port 54321, and the other 54322.
- Because of the unique port numbers, R1 knows to send one reply to PC1 and the other reply to PC2.
- In this example, both PCs selected the same random source port number because the professor wanted to show the router translating the source port.
- However, if PC2 selected a different random source port than PC1, then R1 would have no need to translate the source port.
  - It would simply use the source port PC2 selected and use that to keep track of the sessions.
- PAT is very useful for preserving public IP addresses, because many inside hosts can share a single public IP.
- It is used in networks all over the world.
- Of the three types covered — static NAT, dynamic NAT, and PAT — PAT is the most widely used because it allows so many inside hosts to share a single public IP address.

PAT Configuration



- Configuring PAT is basically the same as dynamic NAT — you just add one keyword.
- First, define the inside and outside interfaces.
- Then, define which traffic should be translated by using an ACL.
  - Again, traffic permitted by the ACL will be translated.
- Then define the pool of inside global IP addresses.

Example: NAT Pool for PAT

- This time the professor defined a smaller range, from 100.0.0.0 to 100.0.0.3.
- Chances are, unless the internal network is very large, you won't need more than a single IP address, but this provides some scalability, some room for growth.
- Note: He still specified a prefix length of 24 bits.
  - As long as the specified address range is in the same subnet, the specified prefix length doesn't really matter.
  - 100.0.0.0 through 100.0.0.3 all fit in the subnet 100.0.0.0/24, so the command is valid.

```
R1(config)# ip nat pool POOL1 100.0.0.0 100.0.0.3 prefix-length 24
```

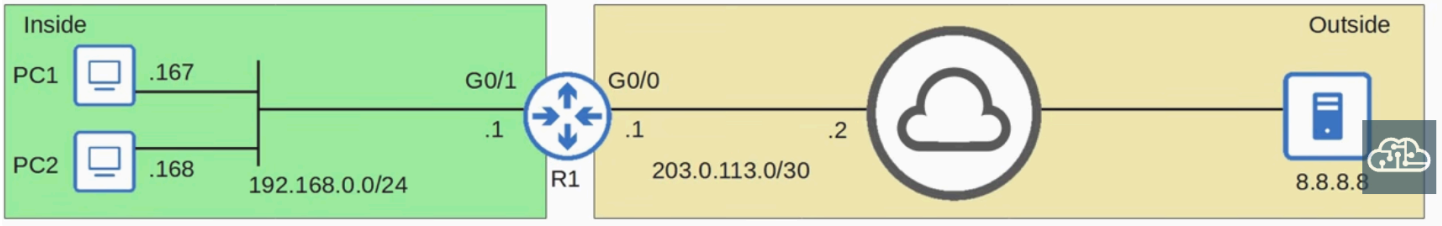
- Finally, the NAT statement itself.
  - It's the same as for dynamic NAT, except at the end you use the keyword `overload`.

```
R1(config)# ip nat inside source list 1 pool POOL1 overload
```

- So, PC1 and PC2 both send DNS requests to 8.8.8.8.
  - The professor then takes a look at the NAT translations and statistics on R1.

Viewing R1's NAT Translations for PAT

```
R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
udp 100.0.0.1:63925    192.168.0.167:63925 8.8.8.8:53        8.8.8.8:53
udp 100.0.0.1:59549    192.168.0.168:59549 8.8.8.8:53        8.8.8.8:53
R1#show ip nat statistics
Total active translations: 2 (0 static, 2 dynamic; 2 extended)
Peak translations: 2, occurred 00:00:03 ago
Outside interfaces:
  GigabitEthernet0/0
Inside interfaces:
  GigabitEthernet0/1
Hits: 4  Misses: 0
CEF Translated packets: 0, CEF Punted packets: 4
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 3] access-list 1 pool POOL1 refcount 2
 pool POOL1: netmask 255.255.255.0
        start 100.0.0.0 end 100.0.0.3
        type generic, total addresses 4, allocated 1 (25%), misses 0
```

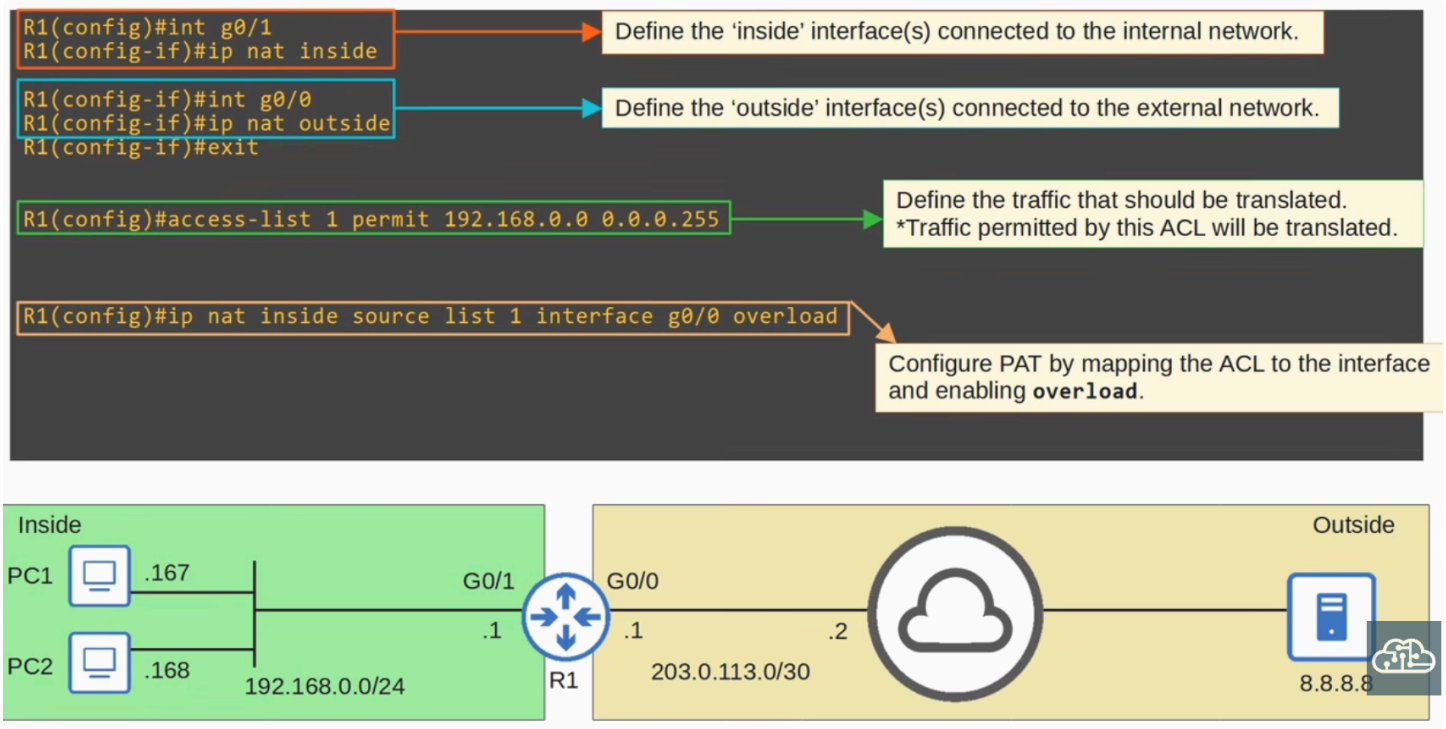


- Here are the translations on R1.
- One big difference compared to regular dynamic NAT:
  - Those one-to-one dynamic mapping entries aren't displayed.
  - That's because, when using PAT, there are no one-to-one mappings — they're many to one.
  - Many inside local addresses to one inside global address.
- Both 192.168.0.167 and .168 are using 100.0.0.1.
- They both selected different source port numbers, so R1 didn't need to translate the port numbers.
  - R1 will use port number 63925 to track the communication flow between PC1 and 8.8.8.8.
  - R1 will use port number 59549 to track the communication flow between PC2 and 8.8.8.8.

Example: SHOW IP NAT STATISTICS

- The professor shows the output of `show ip nat statistics`.
- He notes you can pause the video to check out the output if interested, but he moves on to show one more useful way to configure PAT.

Configuring PAT with the Router's Own Public IP Address

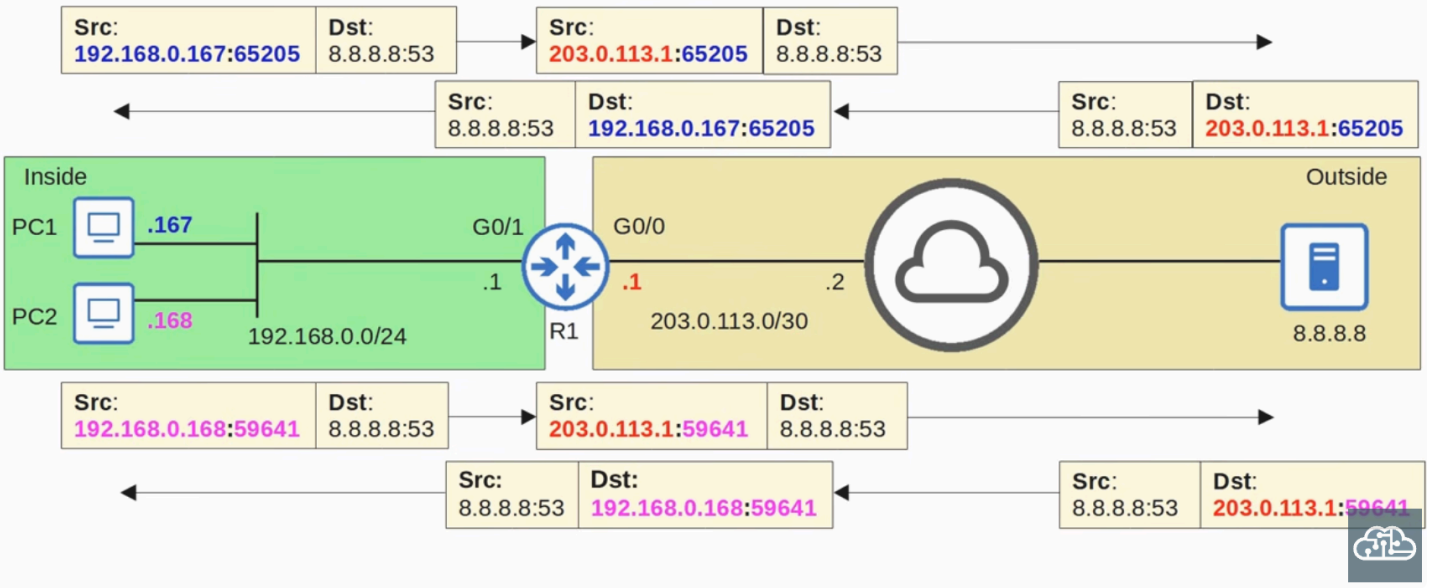


- One more way to configure PAT, and probably the more common way, is to configure the router to use its own public IP address when translating the source IP of packets.
- As always, you must configure the inside and outside interfaces on the router.
- Then, just like before, use an ACL to define which traffic should be translated.
- Here's the difference with the previous method of configuring PAT:
  - Instead of specifying a POOL and then the OVERLOAD keyword, you specify INTERFACE, then the outside interface, g0/0 in R1's case, and then OVERLOAD.

```
R1(config)# ip nat inside source list 1 interface g0/0 overload
```

- R1 will translate the source IP address of packets from 192.168.0.0/24 to the IP address of its G0/0 interface, 203.0.113.1.
- It will also use a unique source port number for each traffic flow, so it can keep track of them and multiple internal hosts can share that single IP address.

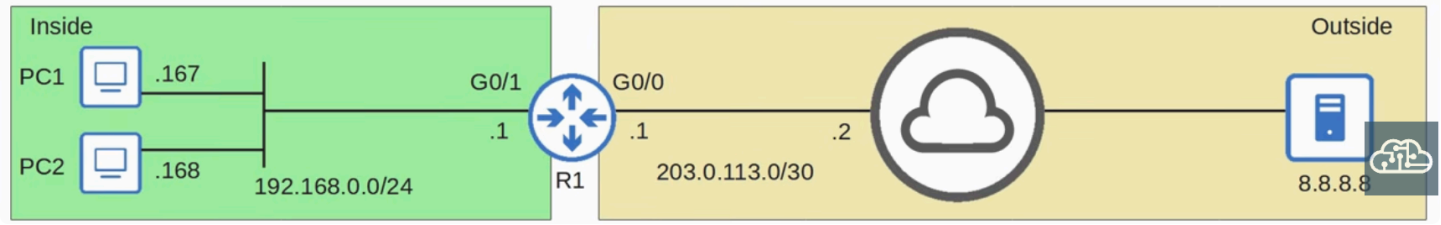
Demonstration: PAT Using the Router's Own Public IP Address



- PC1 and PC2 both send DNS requests to 8.8.8.8.
- Notice that they both selected a different random source port:
  - 65205 for PC1.
  - 59641 for PC2.
- R1 translates the source IP address of both packets, changing them to the IP of R1's G0/0 interface, 203.0.113.1.
- Because both source port numbers are already different, there is no need to translate the port numbers.
- The server sends the replies, and R1 translates them back and sends them to their correct destinations.
- Because of the unique port numbers, R1 knows which to translate back to 192.168.0.167 and which to translate back to 192.168.0.168.

R1's NAT Translation Table After PAT with Interface

```
R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
udp 203.0.113.1:65205  192.168.0.167:65205 8.8.8.8:53        8.8.8.8:53
udp 203.0.113.1:59641  192.168.0.168:59641 8.8.8.8:53        8.8.8.8:53
R1#show ip nat statistics
Total active translations: 2 (0 static, 2 dynamic; 2 extended)
Peak translations: 2, occurred 00:36:30 ago
Outside interfaces:
  GigabitEthernet0/0
Inside interfaces:
  GigabitEthernet0/1
Hits: 12  Misses: 0
CEF Translated packets: 0, CEF Punted packets: 12
Expired translations: 4
Dynamic mappings:
-- Inside Source
[Id: 4] access-list 1 interface GigabitEthernet0/0 refcount 2
```



- Here is R1's NAT translation table after the demonstration.
- Notice that both 192.168.0.167 and .168 are using R1's IP address, 203.0.113.1.
- Also notice that, as the professor said, there was no need for R1 to translate the source port numbers, because they were already unique.

Example: SHOW IP NAT STATISTICS Output

- The professor shows the output of `show ip nat statistics`.
- He notes you can pause and check the command out if you want, but going in depth into all of the output isn't necessary.

Verification:

```
R1# show ip nat statistics
```

Review and Practice of the New Commands

- These are the new commands covered in this video.
- If you don't remember these commands, go back in the video to review them.
- Make sure to practice the commands in Packet Tracer:
  - Make your own labs.
  - Use the professor's practice labs.

Command	What it does
`ip nat pool <name> <start-ip> <end-ip> prefix-length <#>`	Creates a pool of **public IP addresses** using a CIDR prefix.
`ip nat pool <name> <start-ip> <end-ip> netmask <mask>`	Creates the same NAT pool, but specifies a **subnet mask** instead.
`ip nat inside source list <ACL> pool <name>`	**Dynamic NAT:** translates private IPs matched by the ACL to available public IPs in the pool.
`ip nat inside source list <ACL> pool <name> overload`	**PAT:** allows many private hosts to share the public IPs in the pool by using different port numbers.
`ip nat inside source list <ACL> interface <interface> overload`	**PAT:** allows many private hosts to share the **single public IP of the router's interface**.

Review of the Video

- Before moving on to the quiz, the professor reviews what was covered in this video.
- First, he demonstrated how static NAT not only allows inside hosts to access external networks, but it also allows hosts on external networks to access the internal hosts using the mapped inside global IP address.

- Then he covered dynamic NAT and dynamic PAT.
- Dynamic PAT is probably the most common type of NAT.
- It allows many internal hosts to share a single public IP, all at the same time.

Example: Home Router

- The professor's home router, for example, has a single public IP address.
- All of the devices in his home — such as his PC, phones connected to Wi-Fi, etc. — all share the single public IP address when accessing the Internet.
- He reminds viewers to watch until the end of the quiz for a bonus question from Boson Software's ExSim for CCNA.
- He then moves on to the quiz.
