IT 341: Introduction to System Administration

- **Private IP Addresses and the Internet**
- **Using IP Addresses to Communicate Over the Internet**
- **Network Address Translation**

Private IP Addresses and the Internet

- For one computer to talk to another over the Internet, both machines must be assigned IP addresses
- But most computers on a network are assigned **private IP addresses**, which routers cannot send out over the Internet since they are not unique
- So…how can machines assigned private IP addresses talk to *other* machines over the Internet?
- For this to happen, the gateway used by these machines must provide a service called **Network Address Translation**

Using IP Addresses to Communicate over the Internet

- There are two protocols for packets on the Internet
  - TCP
  - UDP
- Universal Datagram Protocol (UDP) only goes in one direction:
  - A packet is sent to another machine and...
  - ...there is no way to know if the packet has arrived
- Most of what we on the Internet is done using ***TCP***

Using IP Addresses to Communicate over the Internet

- Transmission Control Protocol (TCP) sets up a communication channel between two machines
- So each package must have two pieces of information
  - A destination socket
  - A return socket
- A `socket` has three pieces of information:
  - IP address
  - Port number
  - Packet protocol
- When you ask your browser to get you a web page, it sends out a request for data to a web server using the HTTP protocol

Using IP Addresses to Communicate over the Internet

- Let's assume that the machine running the browser also has a public IP address
- The machine running the browser must know
  - the *IP addresses* of the web server
  - which *port* to use on that IP address
- All web servers listen for HTTP requests on port 80
- If the client machines were trying to get a web page from **nytimes.com**, whose web server has the IP address 170.149.172.130, it would send the request to port 80 at this address:

  `170.149.172.130:80`

Using IP Addresses to Communicate over the Internet

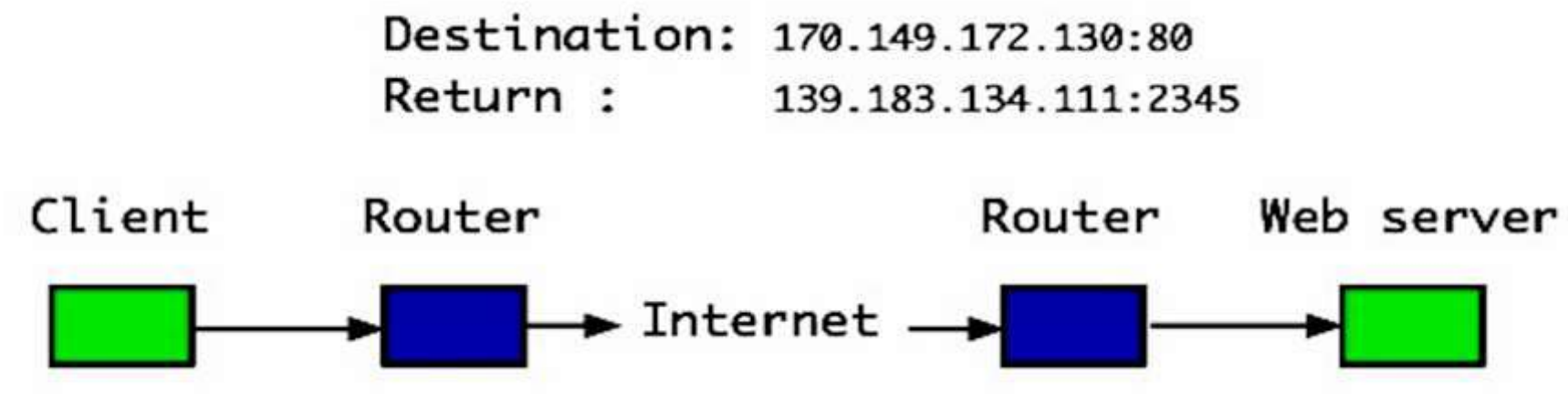
- Since the client machine needs to get information back from the web server, it needs to provide a return socket to the web server
- This return address also consists of an IP address and a port
- The **IP address** will be the address of the client machine, but the **port number** will be chosen randomly
- Let's say the client machine has IP address 139.183.134.111 and that it chooses port 2345 to receive the web page
  - The client machine will send its HTTP request to

    `170.149.172.130:80`
  - giving the following as the return address:

    `139.183.134.111:2345`

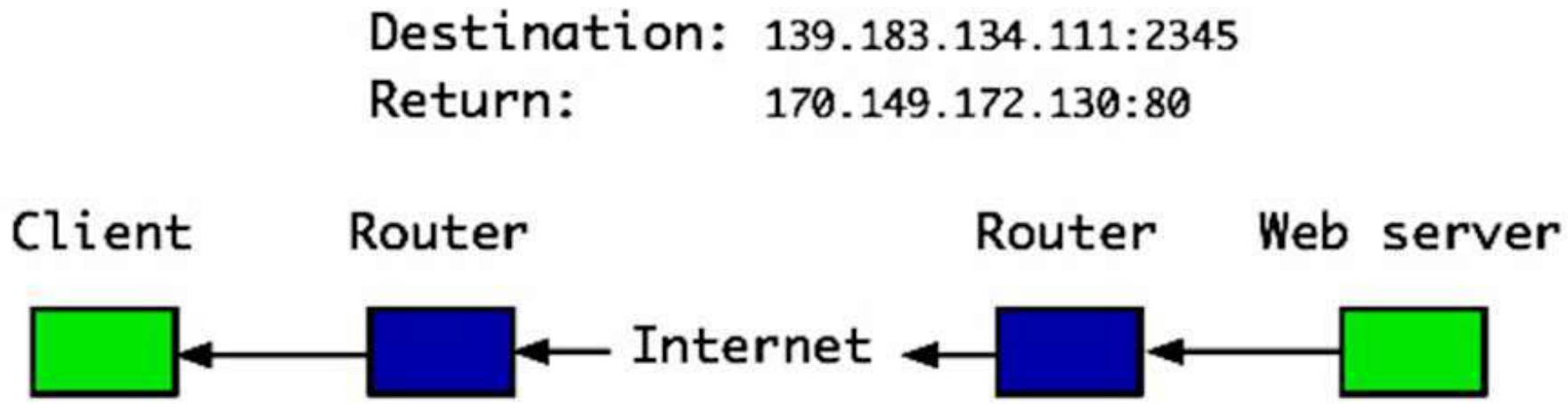
Using IP Addresses to Communicate over the Internet

- The TCP packet goes to the router of the client's network, and since the destination IP is public, the router sends it out over the Internet
- Eventually, the request arrives at the router for the New York Times - which sends it to the web server.
- The process looks like this:



Using IP Addresses to Communicate over the Internet

- When the web server replies with the requested page, information flows in the other direction:



Network Address Translation

- However, if a machine has a private IP address, things are more complicated
- The client machine can still send the HTTP request to the same IP address at the same port, but it cannot use its IP address for the return address
- No machine can contact another using a private IP address because private IP addresses are not routable over the Internet
- This is where **Network Address Translation**, usually abbreviated NAT, comes in

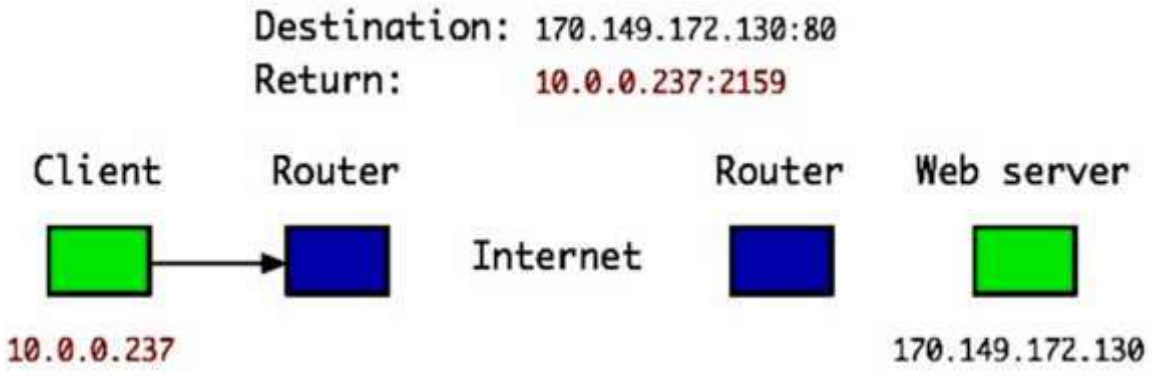
Network Address Translation

- **NAT** is a service provided by the router
- It takes the packet sent to another machine and changes the return socket
- It changes the IP part of the return socket to its own IP address, but it changes the port part of the return socket as well
- Here is how it works…
  - Let's say the client has the private IP address **10.0.0.237** and that it has created a HTTP request for the New York Times web site, which has the TCP address

    `170.149.172.130`
  - Since this request is sent to a web server, the port is **80**

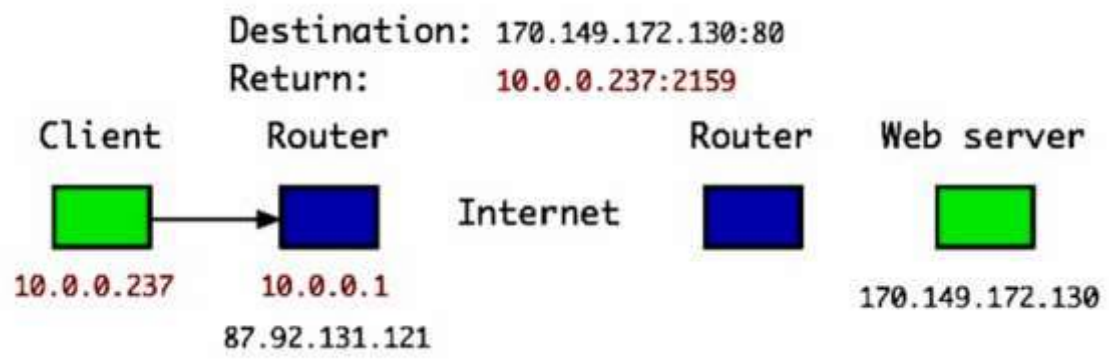
Network Address Translation

- So, the destination of this request is: `170.149.172.130:80`
- The web browser has to pick a random port for the return message
- Let's say it picks port 2159 So the return address of for the request is `10.0.0.237:2159`
- The client sends this request out on the local network - where it is caught by the router. So, the situation looks like this:



Network Address Translation

- The router has two IP addresses - one for each of its two Ethernet cards
- One **NIC** connects to the local network and has a private IP address
- The other NIC connects to the Internet and has a public IP address
- Let's say the private address is **10.0.0.1** and the public address is **87.92.131.121**, so the situation looks like this:

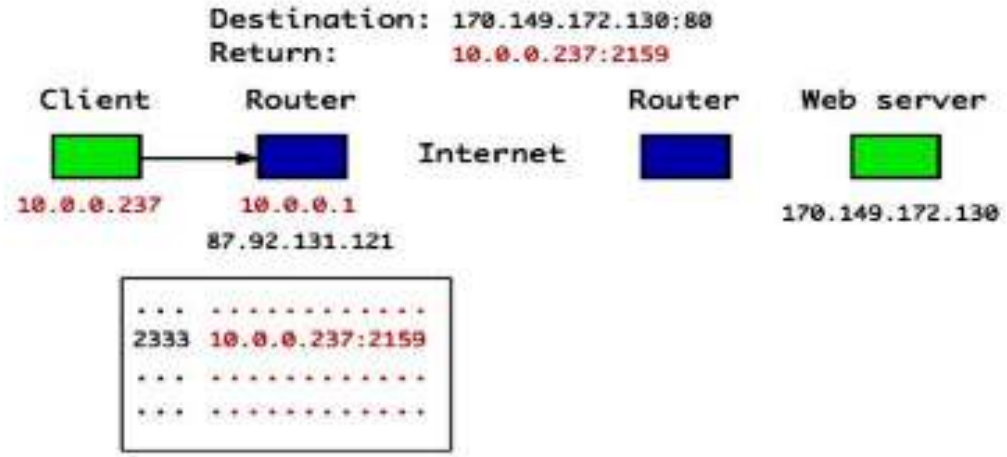


Network Address Translation

- NAT changes the return socket
- It changes the IP part of the return socket to its own public IP address
- But what about the port number?
- NAT picks its own random port number (say 2333) to create the new return socket
  - `87.92.131.121:2333`
- This is a valid return IP address for sending the request over the Internet
- But what happens where the reply comes back to this address?

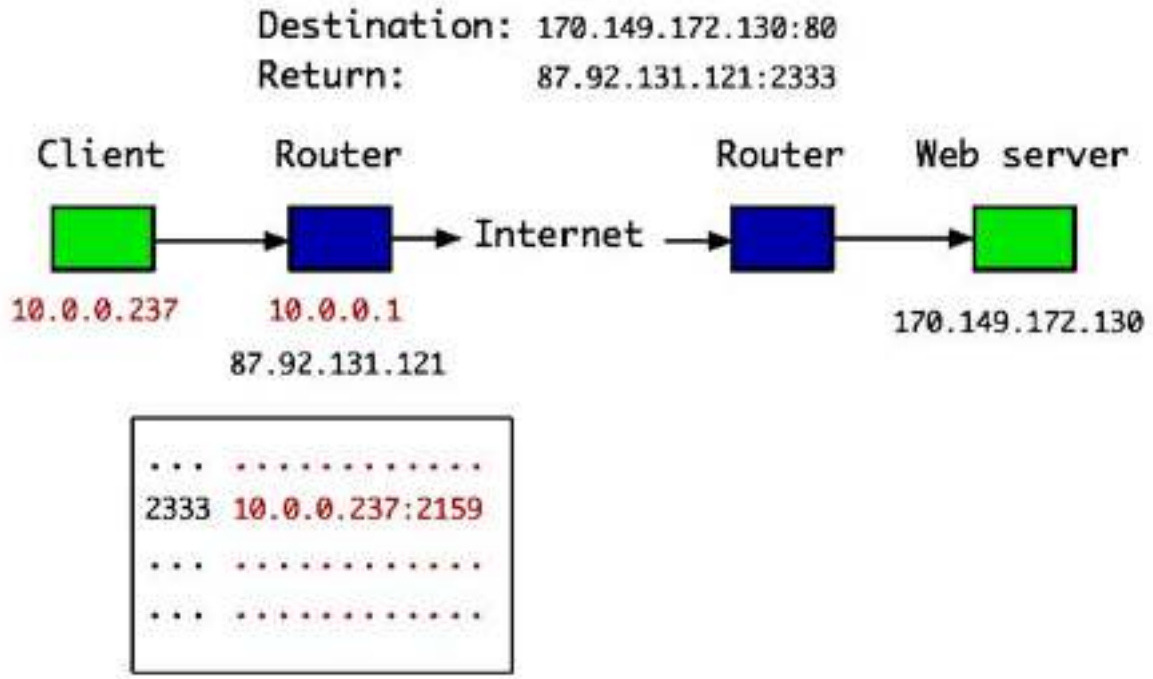
Network Address Translation

- NAT needs to know where to forward the reply
- In order to do this, it creates an entry in a table
- The key for this table is the port number the router chose for public return socket
- The value associated with this key is the true return socket
- So, here is our situation now:



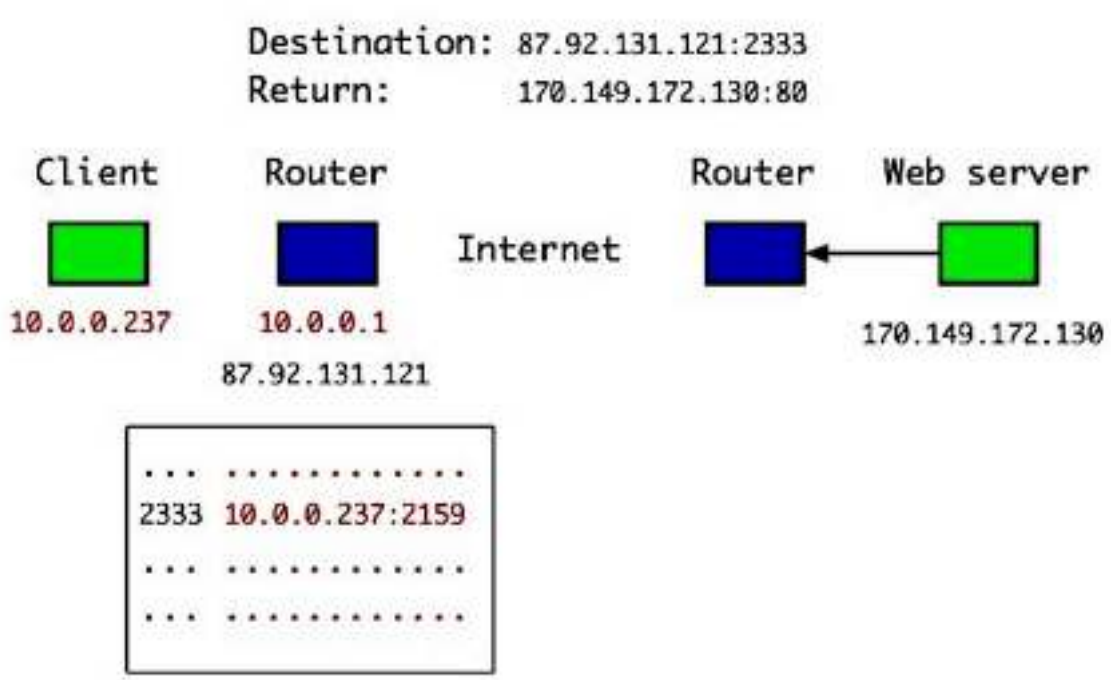
Network Address Translation

- Now that the request has a proper public address, it can be sent out over the Internet



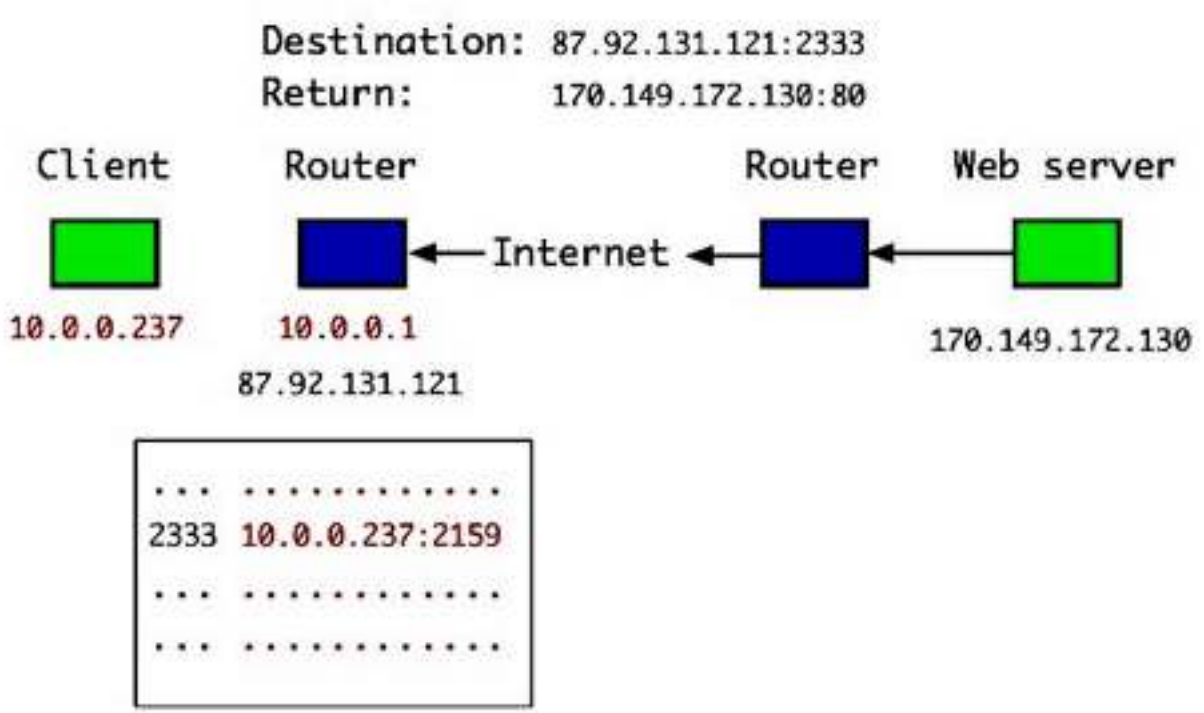
Network Address Translation

- When the reply is sent back, the original return socket is the now the destination socket, and the new return socket points to the web server:



Network Address Translation

- The reply eventually reaches the router that performed the original Network Address Translation:



Network Address Translation

- Now, the router has to figure out where to forward the reply
- It takes the port number on the destination address - and uses it to find the **real** destination address
- Then it changes the destination address to the original reply address and sends the packet on its way: