NAT (Network Address Translation)
Jokes

- ARP
- Ghostbusters
Introduction to Nat

- Grew up in Lexington, KY
- Enjoy stargazing, cycling, and mushroom hunting
- Met Mario once (long time ago)
Network Address Translation

- Not very old (only in heavy use since the late 90s)
- A protocol to map from private addresses to public addresses, and vice-versa
- Utilizes port numbers as secondary addressing information
- Most common type of NAT is actually NAPT (Network Address Port Translation)
- Other type of NAT is “Basic NAT” (which we won’t really be discussing)
Any IP network that isn’t directly connected to the internet
IP addresses can be assigned however we want!
However, generally these ranges are used:
• 10.0.0.0 – 10.255.255.255
• 172.16.0.0 – 172.31.255.255
• 192.168.0.0 – 192.168.255.255
NAT Operation

- Each NAT device (router) has an address translation table.
- For outbound packets, a new table entry is made, choosing an arbitrary source port number (TCP/IP headers rewritten).
- For inbound packets, the table is consulted to rewrite the packet headers and re-route to an internal host.
- Phone analogy.
Why is NAT necessary?

- Not enough IP addresses to go around
- We want some hosts not to be publicly accessible
Types of NAT

- **Full-cone NAT**
  - Accepts data through any previously used port

- **Address-restricted-cone NAT**
  - Only accepts data through previously used ports if the source IP matches a system we’ve already sent to

- **Port-restricted-cone NAT**
  - Like the above, but uses source ports too

- **Symmetric NAT**
  - Mappings are unique to external hosts: a different public port is used for each external host
NAT is great!

But it has issues

Like what?

• Breaks end-to-end connectivity
• Should just use IPv6
• Rewrites packet headers
• Even requires new TCP checksum!
• Initial issue: how do you connect to a host behind a NAT if it hasn’t talked to you first?
You’re behind a NAT, and you need an external host’s packets to get to you

Example: running a web host behind a NAT

You can’t necessarily send an outbound packet first to write the NAT table

Major issue for games and P2P

Solutions?

• Port forwarding (manually adding tables to the address translation table)
Two hosts behind NATs need a way to exchange data directly.
They know each other’s IPs, but not each other’s communication ports.
They both connect to a known server that exchanges the data for them.
They can now communicate.
Often used for multiplayer games.
UFnP and IGD

1. UPnP: Universal Plug and Play
   - Set of protocols for networked devices to perform discovery automatically

2. IGD: Internet Gateway Device protocol
   - NAT protocol that can perform automatic port mapping
   - Allows a host inside a network to tell the router which public port it wants to use for communication
   - Also gives mechanisms for finding public IP address and checking existing port mappings
   - Games can rely on this protocol to configure NAT tables such that users can be mapped with known ports and communication can take place
Old Name: Simple Traversal of UDP through NAT
New Name: Session Traversal Utilities for NAT
Protocol for NAT traversal
Attempt to standardize NAT traversal by establishing NAT categories and methods for checking for/communicating with each
Traversal Using Relays Around NAT
- Similar to earlier punchthrough algorithm
- A server sits between two hosts behind NATs
- The server relays data between the two hosts
Interactive Connectivity Establishment
Protocol that utilizes STUN and TURN to perform NAT punchthrough
Used often in VoIP