Challenges of Network Address Translation in P2P networks

Ramakrishnan Muthukrishnan ram@leastauthority.com

https://mastodon.radio/@vu3rdd

2019/Nov/07

Networks

Modern internet is mostly around centralized servides. Google/Twitter/Facebook...

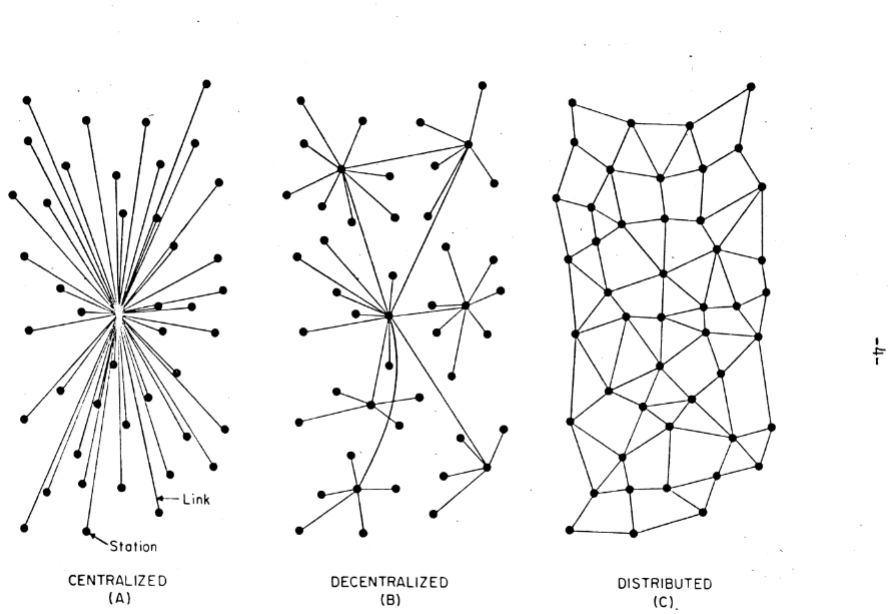
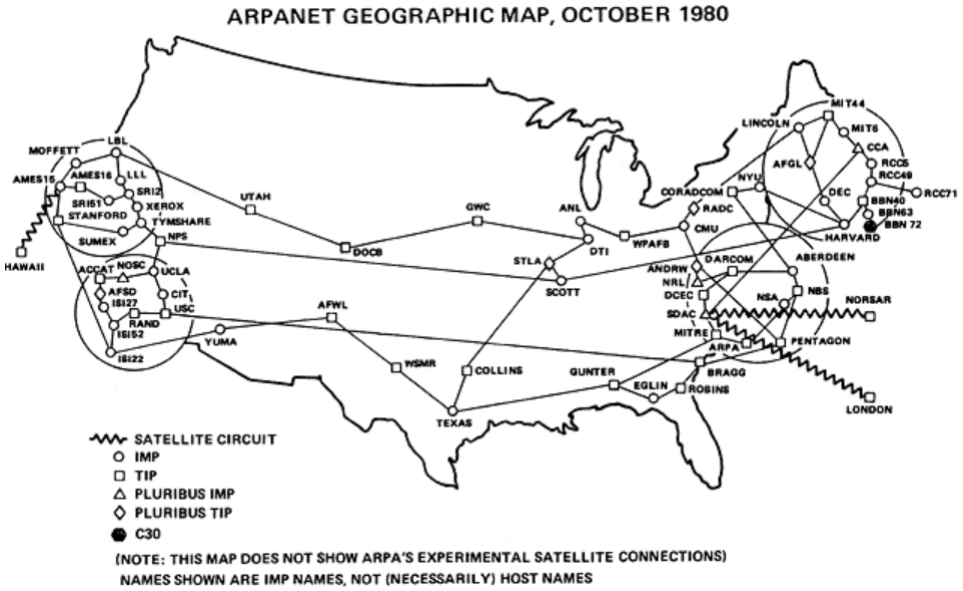


FIG. 1 — Centralized, Decentralized and Distributed Networks

Origins of the internet

The internet was conceived in the late 1960s fueled by the Cold War and ways to communicate in the case of Nuclear wars. They also had the purpose of *sharing computer resources*. It was very much a network of networks. Protocols like the Email, Usenet, IRC are all designed to run in a decentralized manner.

It was designed in such a way that new "nodes" can be added. Nodes are today's equivalent of routers and there were a few 10s or 100s of users (with "hosts" connected to the nodes) behind each node.



Modern Internet

Email and FTP made the usage of internet very appealing to users outside the Government and then later, the hyperlinks and world wide web make the internet explode in the popularity among non-scientific community. People started building commerce around internet.

Modern internet is a highly commercialized system where several corporations like Facebook/Google/Twitter run highly centralized services. Most of the computers on the internet are not providing any service. It just uses them. Such computers are called "Clients" and the ones providing any service is called a "Server".

Every computer "on the internet" need to have an IP address. IP address is a 32-bit field in the IP header of the IP packet, usually represented in group of four 8-bit numbers A.B.C.D. The original designers did not anticipate that world would need more that ~3.2 million addresses. But then we have computers of various size like desktop computers, laptops, smart phones, internet connected door bells and fridges etc.. and each of them need an IP address.

As the popularity of the internet increased the 32-bit addresses started running out. Since most computers were only acting as "Clients", it was decided that something needs to be done in short term and long term.

P2P

The ideal state of the web would be peer to peer. This would eliminate use of large data centers, censorships, algorithmic timelines and manipulation of the minds of people and many other evils caused by corporate control of the internet.

In the ideal case, all data would be content addressable, replicated and seeded (a la bittorrent, ipfs, tahoe-lafs). Each peer would make their own decisions about what data to share and who it wants to share it to.

IPv4/IPv6

In the mid-1990s, people realized the problems and proposed two RFCs until a new addressing scheme is designed and implemented:

- https://www.ietf.org/rfc/rfc1631.txt
- https://www.ietf.org/rfc/rfc1918.txt

These two RFCs are the origins of network address translation (NAT). rfc1631 specifies a process to translate a single public address to multiple private addresses. rfc1918 specifies the private IP addresses and their layouts.

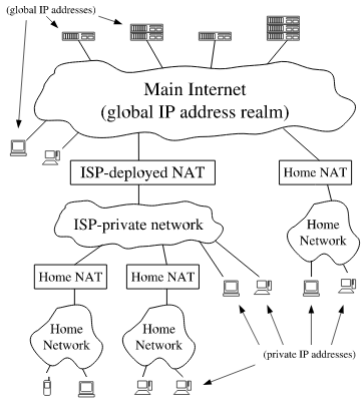
The new addressing scheme was IPv6, which uses 128-bit addresses, represented as 8 groups (*eg: 2001:0db8:0000:0000:0000:8a2e:0370:7334*). Unfortunately, IPv6 and IPv4 do not interoperate, so direct communication between IPv4 hosts and IPv6 hosts are not possible.

So, we are still stuck with IPv4 and NAT.

Network Address Translation

Somewhat similar to office telephone system. One public address with a number of extension numbers. An external caller cannot reach an internal number directly. However, any internal caller can call an external number and conversations can happen after that.

The new internet topology



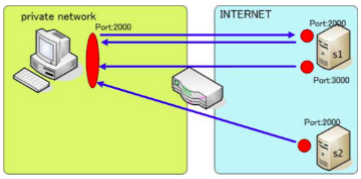
Hosts within a private network can talk to each other, but hosts in two different private networks cannot talk to each other! This breaks P2P.

Network Address Translation

- Router keeps a mapping of the session data (i.e. src (ip, port) -> dst (ip, port).
- Router rewrites the private src ip and/or port with its own (public) ip and possibly remaps the ports in each packet and creates a mapping.
- No standardization on how ports are allocated and how these mappings are done within the router.
- Most common one is the so called "outbound" NAT, that allows only outbound packets from an endpoint in the private network. Inward packets are dropped unless the NAT identifies them as being a part of a session initiated from the private network.
- This is a problem for p2p protocols. What if the two peers are both behind different NATs?
- NAT Traversal is a way to make the session "outbound" to both the NATs.
- Most reliable but least efficient solution to NAT is to use an external *Relay* server.
- are there ways to minimize the use of a relay server?

NAT Variations (1)

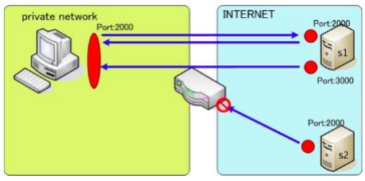
Full Cone NAT (one-to-one NAT)



- Full Cone: similar to manual port mapping.
- Once an internal address (iAddr:iPort) is mapped to an external address (eAddr:ePort), any packets from iAddr:iPort are rewritten to eAddr:ePort at the router.
- *Any* external host (not just the one for which the session mapping was created for) can send packets to iAddr:iPort by sending it to the destination address eAddr:ePort.

NAT Variations (2)

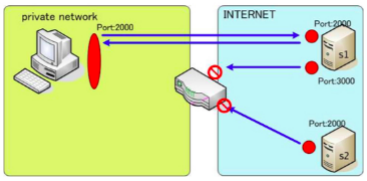
Address Restricted Cone NAT



- Once an internal address (iAddr:iPort) is mapped to an external address (eAddr:ePort), any packets from iAddr:iPort are rewritten to eAddr:ePort at the router.
- An external host (hAddr:any) can send packets to iAddr:iPort by sending packets to eAddr:ePort only if iAddr:iPort has previously sent a packet to hAddr:any. "Any" means the port number doesn't matter.

NAT Variations (3)

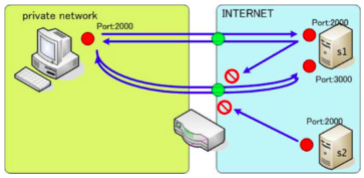
Port Restricted Cone NAT



- Once an internal address (iAddr:iPort) is mapped to an external address (eAddr:ePort), any packets from iAddr:iPort are rewritten to eAddr:ePort at the router.
- An external host (hAddr:hPort) can send packets to iAddr:iPort by sending packets to eAddr:ePort only if iAddr:iPort has previously sent a packet to hAddr:hPort.
- The port used by the internal host is the same port used by the NAT for the external mapping.

NAT Variations (4)

Symmetric NAT



- Each request from the same internal iAddr:iPort to a specific destination hAddr:hPort is mapped to a unique external source address eAddr:ePort; if the same internal host sends a packet even with the same source address and port but to a different destination, a different mapping is used by the router.
- Only an external host that receives a packet from an internal host can send a packet back.

Or in other words, the other above approaches (#1-#3) preserves the source port number. But for symmetric NAT, a random port is chosen for every new connection. This makes port prediction very hard and most NAT holepunching techniques fail for this case.

If both the peers are located behind NATs, they are unable to contact each other to know the mapping.

NAT: A few usage scenarios

Both the users within the same NAT.

```
           Internet
              |
              |
  userA <---> NAT <---> userB
```

Direct connections work, no need to use the relay

NAT: A few usage scenarios (Contd)

One of the users outside the NAT on public internet

```
         Internet
            |\
            | \
userA <---> NAT \
                 \
                  userB
```

Let us say, user A is in a NAT and user B is outside the NAT but on public internet.

Direct connections still work, no need to use the relay. Why? Peer-to-Peer nodes start a server and a client, each node tries to connect to each other. The succeeding socket is used for transfering files.

NAT: A few usage scenarios (Contd)

```
          Internet
          /      \
         /        \
    userA         userB
```

Both users have a public IP address. No problem.

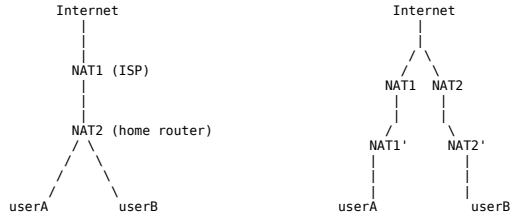
NAT: A few usage scenarios (Contd)

```
           Internet
              |
              |
              |
             / \
            /   \
           /     \
```

```
        /       \
userA <-> NAT      NAT <-> userB
```

Each user is behind a different NAT - most common scenarios. Here we will need to attempt hole punching.

NAT: A few usage scenarios (Contd)



ISPs provide NAT'ed IP to its customers, each customer would have a wireless router which is its own different NAT. Source IP gets re-written everytime the packet passed through the NAT router.

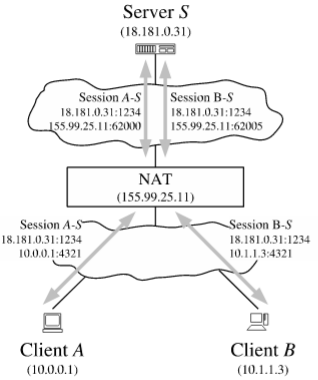
NAT Traversal techniques

Most popular one is the hole punching technique. So, we will discuss that first.

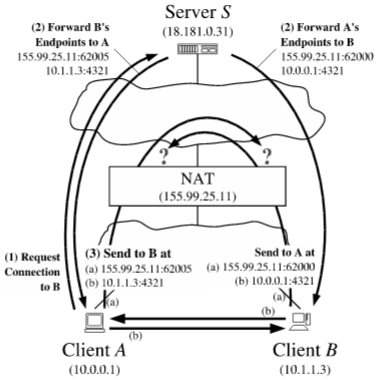
Hole punching

Hole punching relies on a rendezvous server (or an *introducer*) that knows both the parties and has a session with each of them and hence know their public and private (ip, port) pairs.

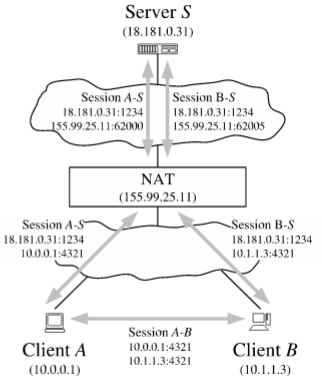
Hole punching, peers behind a common NAT



Before Hole Punching

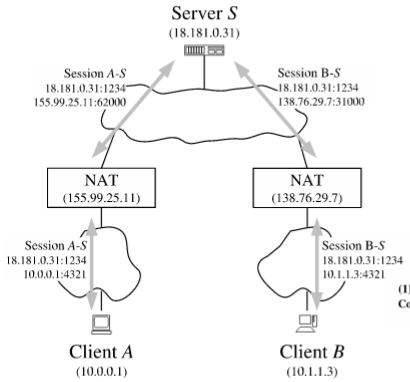


The Hole Punching Process

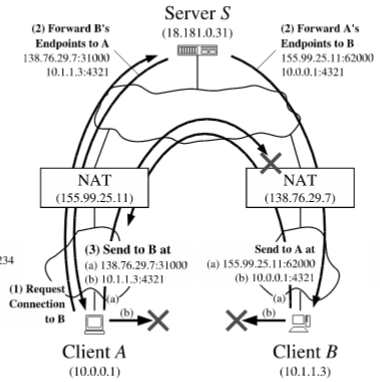


After Hole Punching

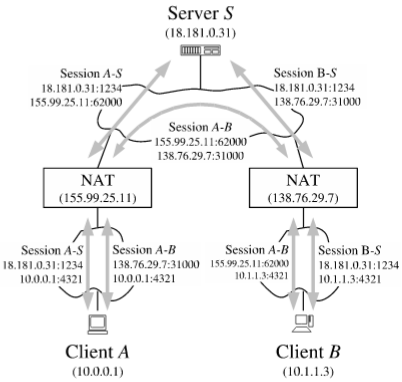
Hole punching, peers behind different NATs



Before Hole Punching

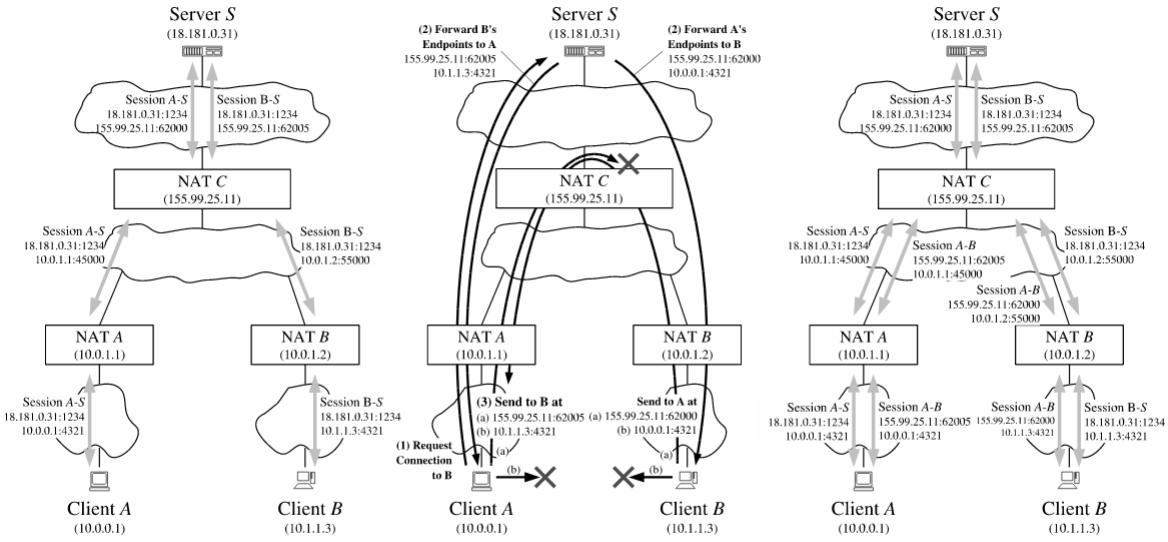


The Hole Punching Process



After Hole Punching

Hole punching, peers behind multiple levels of NATs



- common to have multi-level NATs (often called carrier-grade NAT) among big ISPs.
- behaves line the previous case, but very much depends on the behaviour of the NATs at each level.

NAT Traversal techniques (Continued)

Traversal Using Relays around NAT - TURN

- Most Cisco routers implement symmetric NAT unfortunately, which means, we cannot use any of the NAT traversal techniques.
- We need to fall back to an external relay.
- Applications end up implementing their own custom relay servers.

NAT Traversal techniques (Continued)

Session Traversal Utilities for NAT - STUN

- Specified by RFC-5389.
- Formalizes the rendezvous server messages to discover the transport addresses.
- provides behavioural detection of NAT (RFC-5780).

Interactive Connectivity Establishment - ICE

- specified as RFC-8445
- work with STUN and TURN servers to discover candidate transport addresses.
- comprehensive. Used by most VoIP/webrtc/IoT products.

Others

- NAT PmP
- UPnP
- PCP ...

challenges in TCP NAT Traversal.

- symmetric NAT (each outbound connection from a node may end up using different port numbers, even to the same destination (ip, port).
- enumerating candidates can take a lot of time
- connection attempts to candidates can time out for various reasons that make networks unreliable in general.
- Trickle ICE

Thanks

Questions?

Afew relevant RFCs and references to read

- UDP Hole Punching in TomP2P for NAT Traversal
- Peer-to-Peer Communication Across Network Address Translators
- RFC 5128: https://tools.ietf.org/html/rfc5128 (State of P2P over NAT)
- RFC 3489: https://tools.ietf.org/html/rfc3489 (STUN)
- RFC 2663: https://tools.ietf.org/html/rfc2663 (NAT Terminology)
- RFC 4787: https://tools.ietf.org/html/rfc4787 (UDP NAT traversal)
- RFC 5382: https://tools.ietf.org/html/rfc5382 (TCP NAT traversal)
- RFC 5245: https://tools.ietf.org/html/rfc5245 (ICE)