



CoVim-Enhanced

第六組

B04902017 李立譽 大家躺好了 我來凱瑞 有人沒A+我請喝飲料
B04902025 施博瀚
B04902037 顏子斌
B04902043 謝宏祺
B04902045 孫凡耘
B04902096 陳力榕

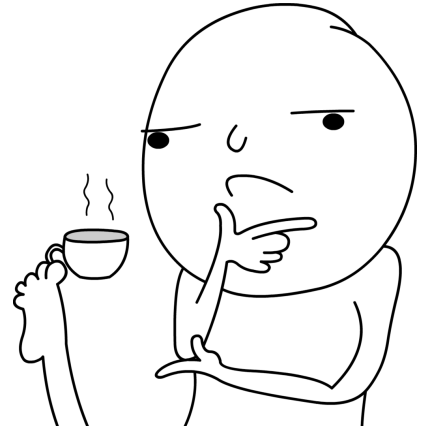


Outline

1. Motivation
2. Technical
 - a. NAT
 - b. UDP hole punching
 - c. TCP hole punching
3. Implementation
 - a. Design & Architecture
 - b. Difficulties
4. Demo

Motivation

Not long ago, Microsoft introduces Code Sharing functionality.
However, not many people use VSCode while coding.
So why not make this functionality on Vim?





NAT

- Network Address Translation
- According to behavior: Full cone NAT, Address Restricted cone NAT ...
- Cause problems for P2P connection
- Hole punching



Symmetric NAT

- A particular private address and destination address pair corresponds to a unique public address
- Hole punching may not work!



UDP Hole Punching

Three parts:

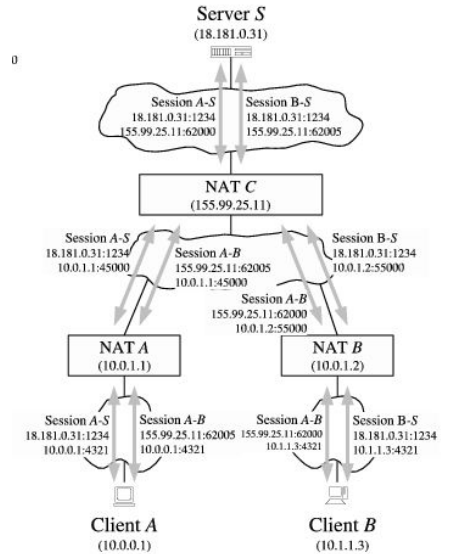
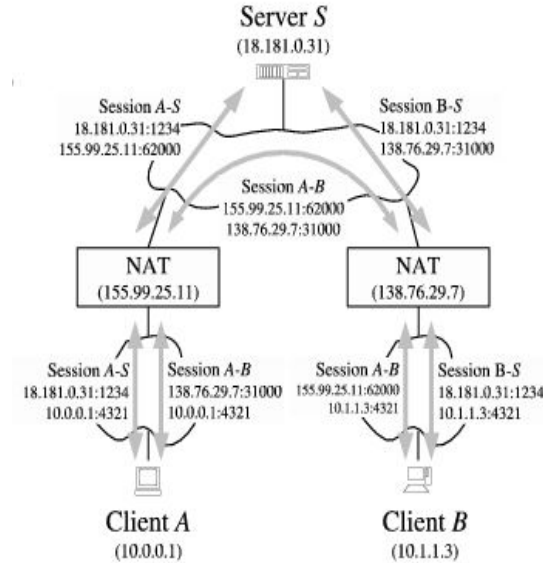
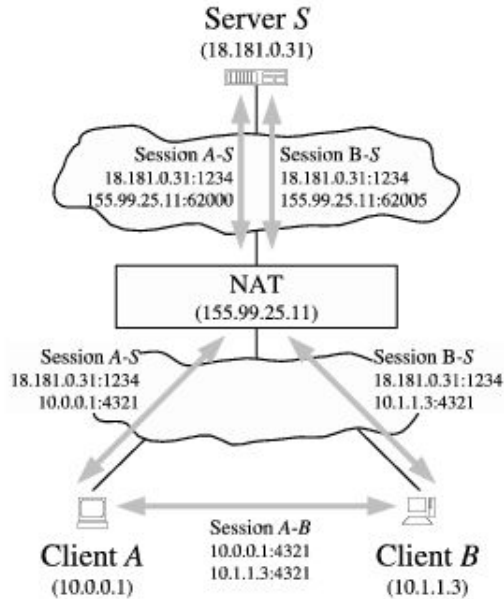
- Rendezvous server S
- Client A
- Client B



UDP Hole Punching

- Step 1 : A ask S to establish session with B
- Step 2 : S replies to A with B's public and private endpoint; S send B a connecting request(A's info). => A and B know each others' info.
- Step 3 : A starts sending UDP packet to B(both private and public), waits for valid response from B and "locks in".
- Step 4 : B also starts sending UDP packet to A(according to info from step2), waits for valid response from A and "locks in".

UDP Hole Punching





TCP Hole Punching

- Not as well-understood as UDP hole punching
- Same process with that of UDP
- *Direction* is decided by the initial SYN packet
- Two situation when A's SYN packet dropped by B and B's SYN packet get through to A :
 - A notice that it's connect session endpoint matches one of outbound session and then the connect session enters the connected state.
 - A notice that there is a active listen socket and thus create a new stream socket with A's connect attempt failing.



TCP Hole Punching

- Simultaneous TCP Open
- Sequential Hole Punching :
 - For those not supporting `SO_REUSEADDR`
 - Cost much time, and requiring the clients to open fresh connections to `S` for each new P2P connection to be forged.



Design & Architecture

- Twisted: event-driven networking engine
- Threading and Multiprocessing
- Safe communication(shared secret key)



Design

To host a new CoVim session: `:CoVim host [port] [name]`

To add a connection(HOST ONLY): `:CoVim add [token]`

To generate your token: `:CoVim start [port]`

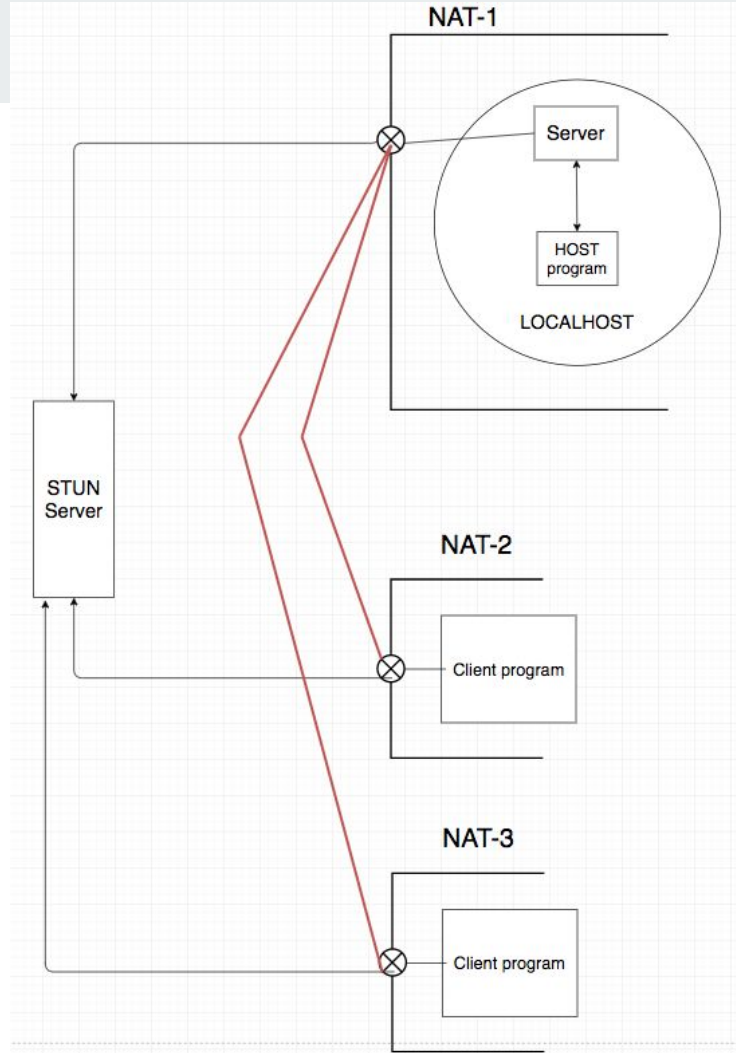
To connect to a host: `:CoVim connect [server_token] [name]`

To disconnect: `:CoVim disconnect`

To quit Vim while CoVim is connected: `:CoVim quit` or `:qall!`

Design & Architecture

- Establish connection(TCP hole punching)
- Hand socket connection to Twisted(Protocol)
- Server “listens” for any “movement” from all clients and broadcast it
- Server quits after all users are disconnected





Difficulties

- Different NAT behavior
- Different platform has different behavior(socket)
- Threading and Multiprocessing



Demo!

- Demo co-editing of vim with and without Public IP



Q&A

- Any questions?



Design & Architecture - PyStun

- Python based STUN client on Github
- Obtain public address of client inside NAT for TCP hole punching
- Resulting public private address pair used as key for CoVim to establish connection to remote host



STUN

- Session Traversal Utilities for NAT
- Allow machine behind NAT to find its own public private address mapping
- Only can't be used under Symmetric NAT

