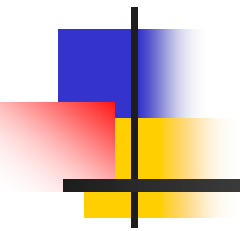
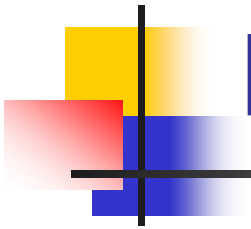
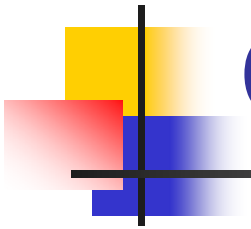


# VoIP + NAT



# References

- “SIP, NAT and Firewalls”, Fredrik Thernelius
- Baruch Sterman and David Schwartz, “NAT Traversal in SIP”, Deltathree
- “STUN – Simple Traversal of UDP Through Network Address Translators”, RFC 3489, IETF
- “An Extension to the SIP for Symmetric Response Routing”, RFC 3581, IETF

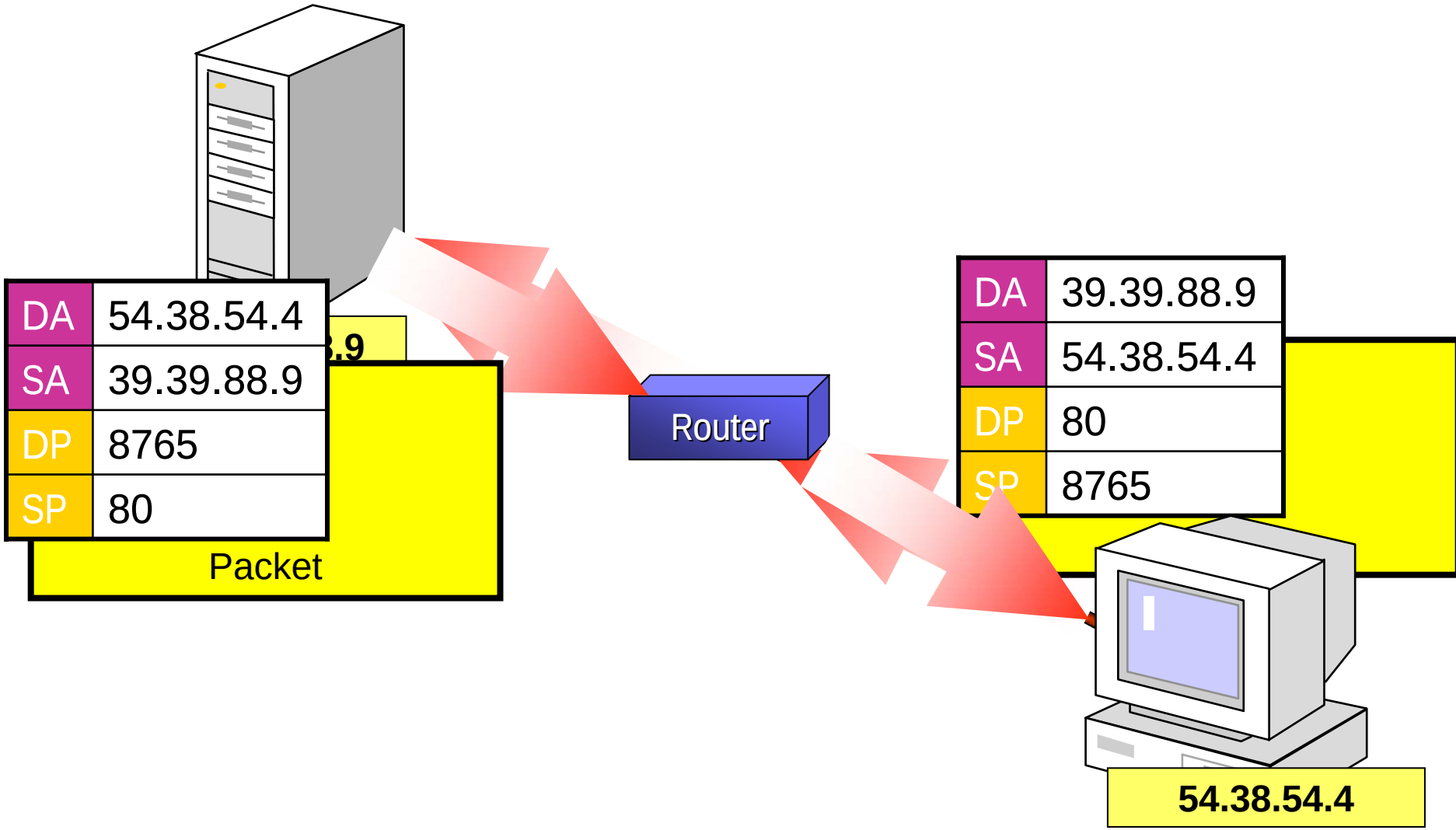


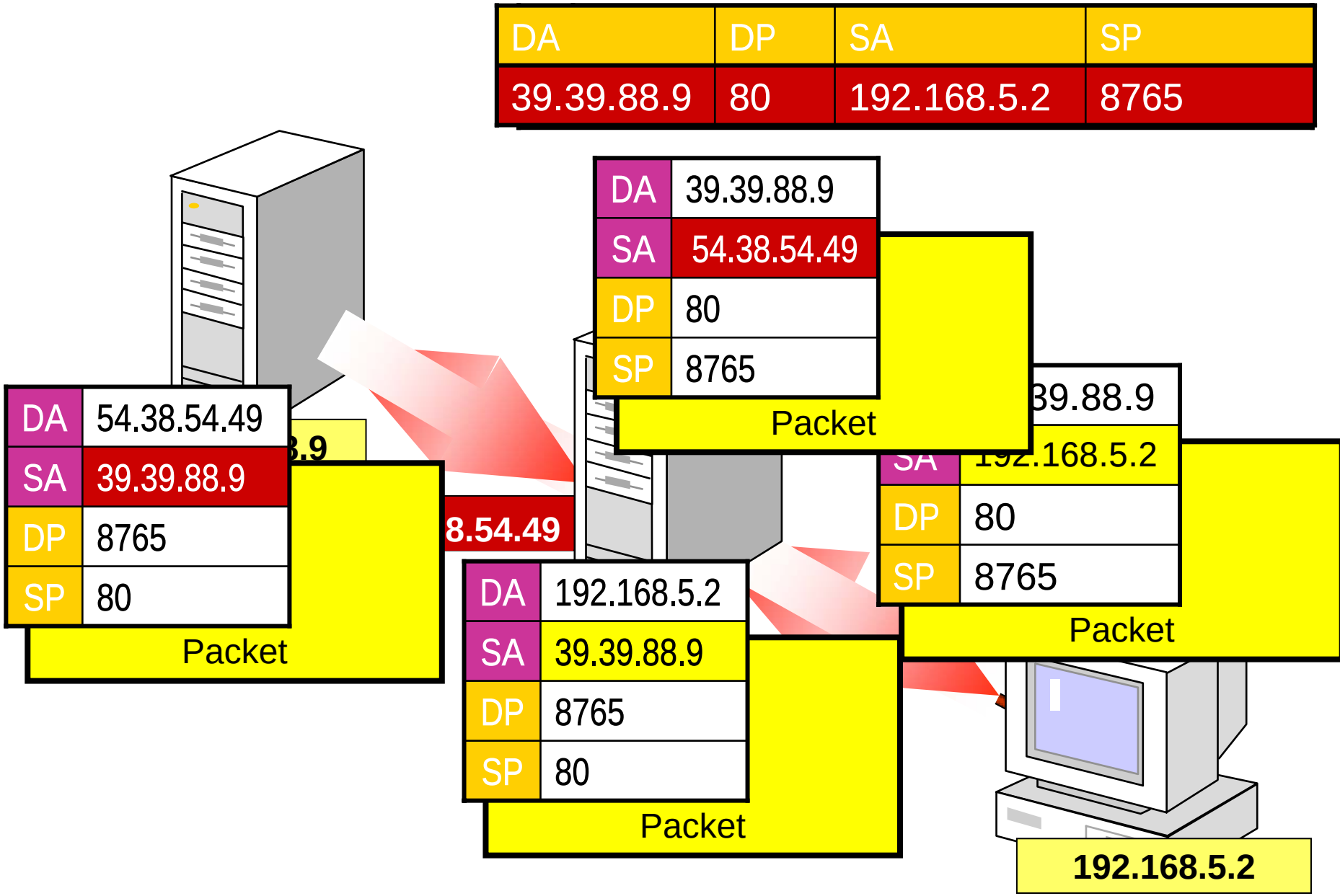
# Outline

- Introduction
- The Problem of VoIP + NAT
- Possible Solutions for VoIP + NAT

# What is NAT?

- NAT - Network Address Translation
  - RFC 3022 - Traditional IP Network Address Translator (Traditional NAT)
  - RFC 1918 - Address Allocation for Private Internets (BCP 5)
  - RFC 2993 - Architectural Implications of NAT
  - RFC 3027 - Protocol Complications with the IP Network Address Translator
  - RFC 3235 - Network Address Translator (NAT)-Friendly Application Design Guidelines
- Convert Network Address (and Port) between private and public realm
- Works on IP layer
- Transparent to Application

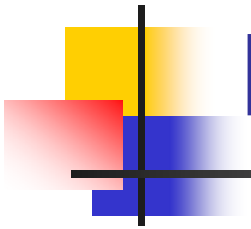

DA
54.38.54.4
SA
39.39.88.9
DP
8765
SP
80
Packet
Router
DA
39.39.88.9
SA
54.38.54.4
DP
80
SP
8765
54.38.54.4


DA
DP
SA
SP
39.39.88.9
80
192.168.5.2
8765
DA 54.38.54.49
SA 39.39.88.9
DP 8765
SP 80
Packet
DA 39.39.88.9
SA 54.38.54.49
DP 80
SP 8765
Packet
DA 192.168.5.2
SA 39.39.88.9
DP 8765
SP 80
Packet
DP 80
SP 8765
Packet
192.168.5.2

# Flavors of NAT [1/3]

## Static NAT

- Requiring same number of globally IP addresses as that of hosts in private environment
- Mapping between internal IP addresses and external addresses is set manually
  - This mapping intends to stay for a long period of time



# Flavors of NAT [2/3]

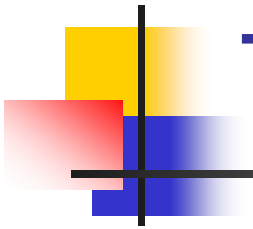
## Dynamic NAT

- Collect the public IP addresses into an IP address pool
- A host connecting to the outside network is allocated an external IP address from the address pool managed by NAT

# Flavors of NAT [3/3]

**NAPT (Network Address and Port Translation)**

- A special case of Dynamic NAT
  - Use port numbers as the basics for the address translation
- The mechanism most commonly used

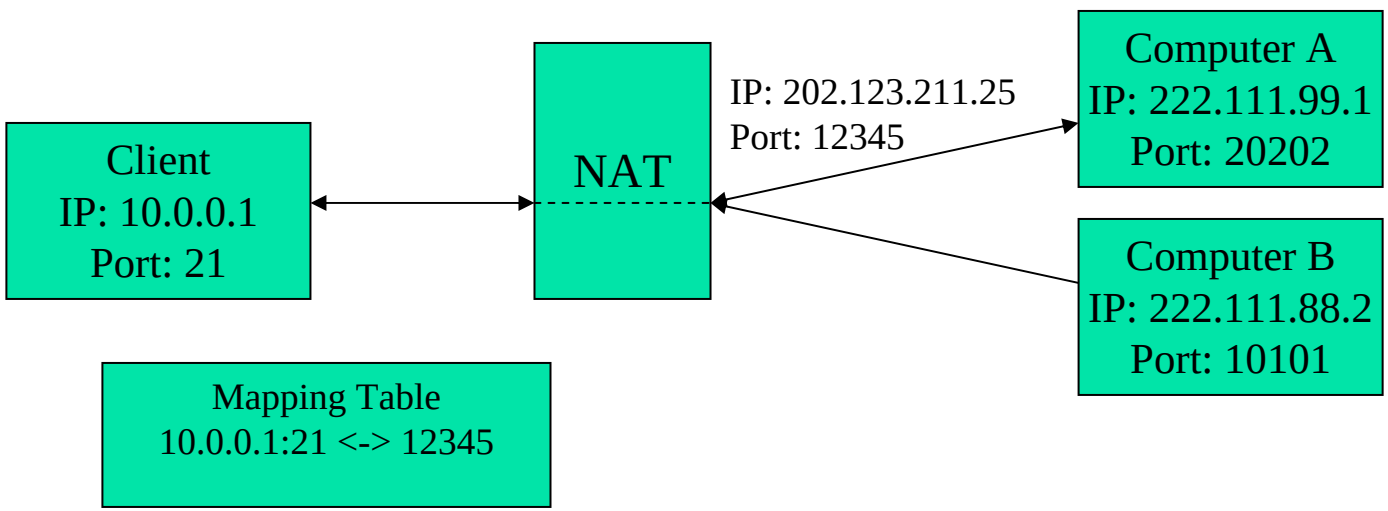


# Types of NAT

- Full Cone
- Restricted Cone
- Port Restricted Cone
- Symmetric

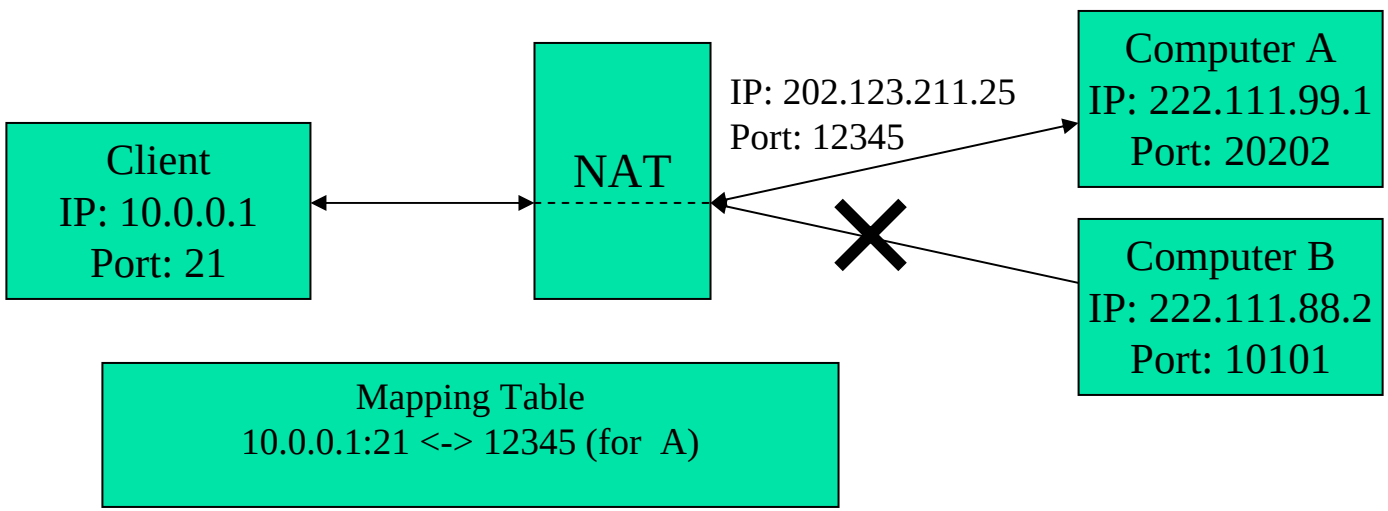
# Full Cone NAT

- Client sends a packet to public address A.
- NAT allocates a public port (12345) for private port (21) on the client.
- Any incoming packet (from A or B) to public port (12345) will dispatch to private port (21) on the client.



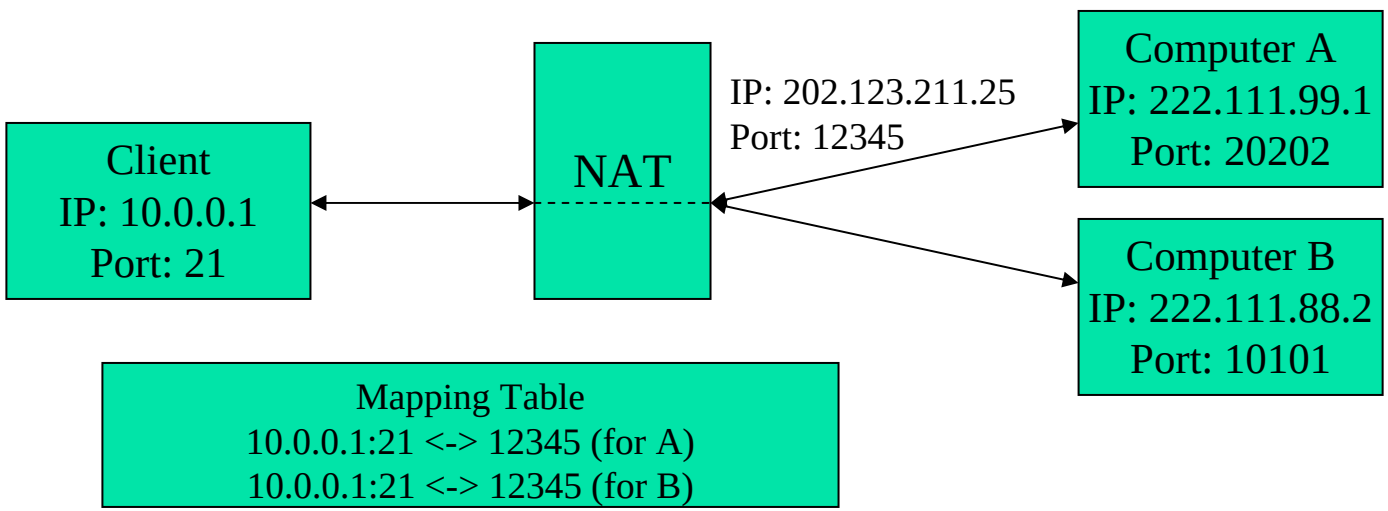
# Restricted Cone NAT [1/2]

- Client sends a packet to public address A.
- NAT allocate a public port (12345) for private port (21) on the client.
- Only incoming packet from A to public port (12345) will dispatch to private port (21) on the client.



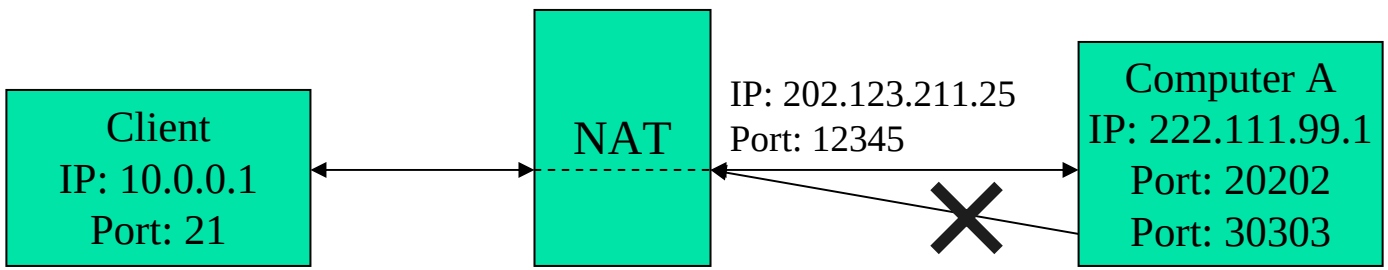
# Restricted Cone NAT [2/2]

- Client sends another packet to public address B.
- NAT will reuse allocated public port (12345) for private port (21) on the client.
- Incoming packet from B to public port (12345) will now dispatch to private port (21) on the client.



# Port Restricted Cone NAT

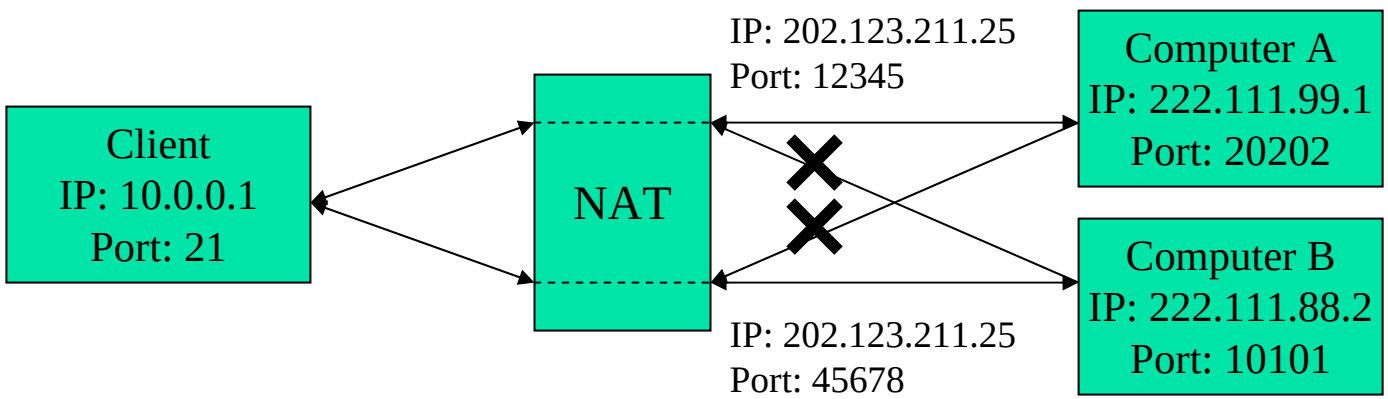
- Client sends a packet to public address A at port 20202.
- NAT will allocate a public port (12345) for private port (21) on the client.
- Only incoming packet from address A and port 20202 to public port (12345) will dispatch to private port (21) on the client.



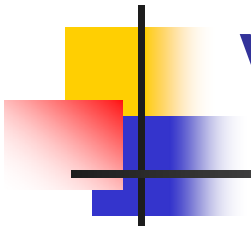

Mapping Table
10.0.0.1:21 <-> 12345 (for A : 20202)
*10.0.0.1:21 <-> 12345 (for A : 30303)*

# Symmetric NAT

- NAT allocates a public port each time the client sends a packet to different public address and port
- Only incoming packet from the original mapped public address and port will dispatch to private port on client

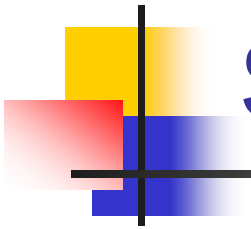



Mapping Table
10.0.0.1:21 <-> 12345 (for A : 20202)
10.0.0.1:21 <-> 45678 ( for B : 10101)



# VoIP Protocol and NAT

- NAT converts IP addresses on IP layer
- Problem 1:
  - SIP, H.323, Megaco and MGCP are application layer protocol but contain IP address/port info in messages, which is not translated by NAT
- Problem 2:
  - Private client must send a outgoing packet first (to create a mapping on NAT) to receive incoming packet



# Solving NAT Traversal Problems

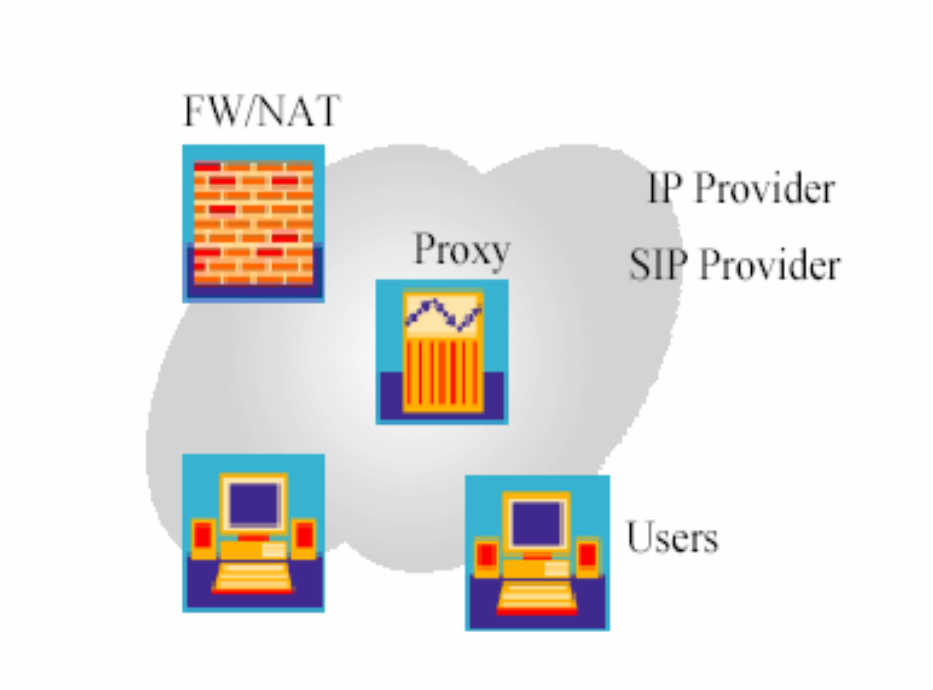
- Objectives
  - Discover mapped public IP & port for private IP & port
  - Use mapped public IP & port in application layer message
  - Keep this mapping valid
- Issues
  - NAT will automatically allocate a public port for a private address & port if needed.
  - NAT will release the mapping if the public port is “idle”
    - No TCP connection on the port
    - No UDP traffic on the port for a period (1 min~5 min)
  - Keep a TCP connection to destination
  - Send UDP packets to destination every specified interval

# NAT Solutions

- IPv6 (Internet Protocol Version 6)
- UPnP (Universal Plug-and-Play)
  - UPnP Forum - http://www.upnp.org/
- Proprietary protocol by NAT/Firewall
  - SIP ALG (Application Level Gateway)
  - No standard now
- SIP extensions for NAT traversal
  - RFC 3581
  - Works for SIP only, can not help RTP to pass through NAT
- STUN (Simple Traversal of UDP Through Network Address Translators)
  - RFC 3489
  - Works except symmetric NAT
- TURN (Traversal Using Relay NAT)
  - draft-rosenberg-midcom-turn-04
  - for symmetric NAT

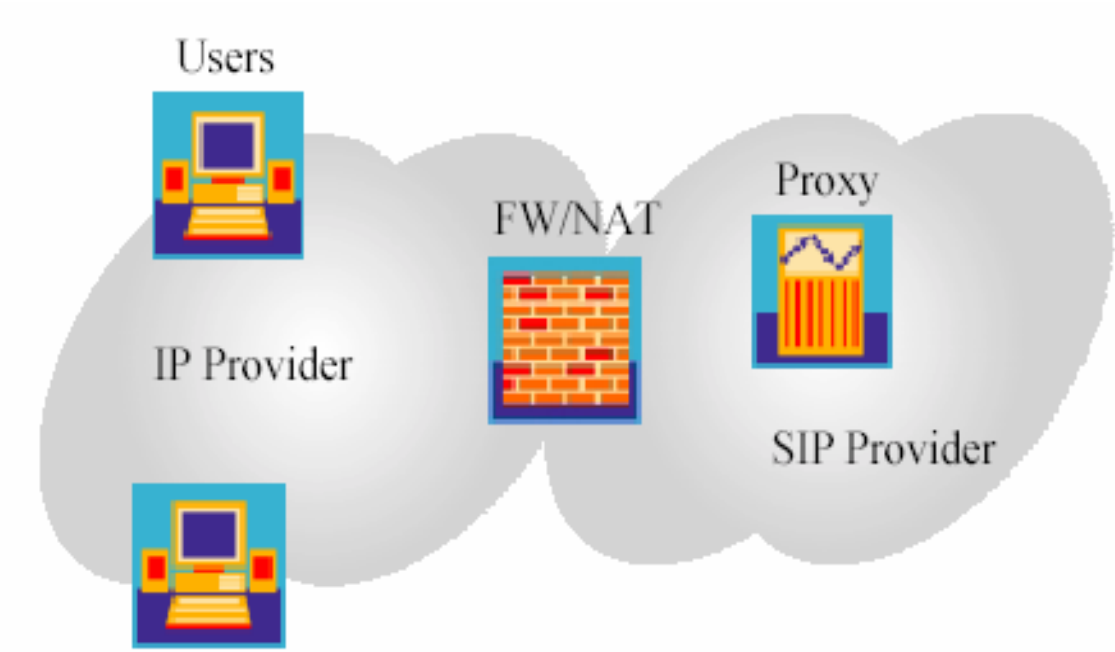
# Two Distinct Cases – NAT Deployment [1/2]

Case I : SIP Provider is the IP Network Provider



# Two Distinct Cases – NAT Deployment [2/2]

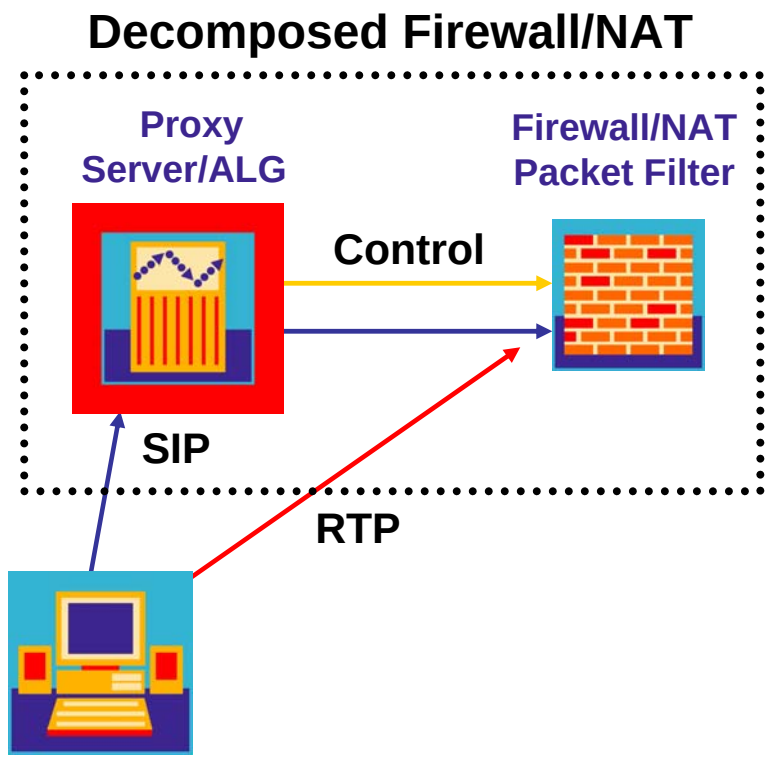
Case II : SIP Provider is **NOT** IP Network Provider



# Solution for Case I – ALG [1/2]

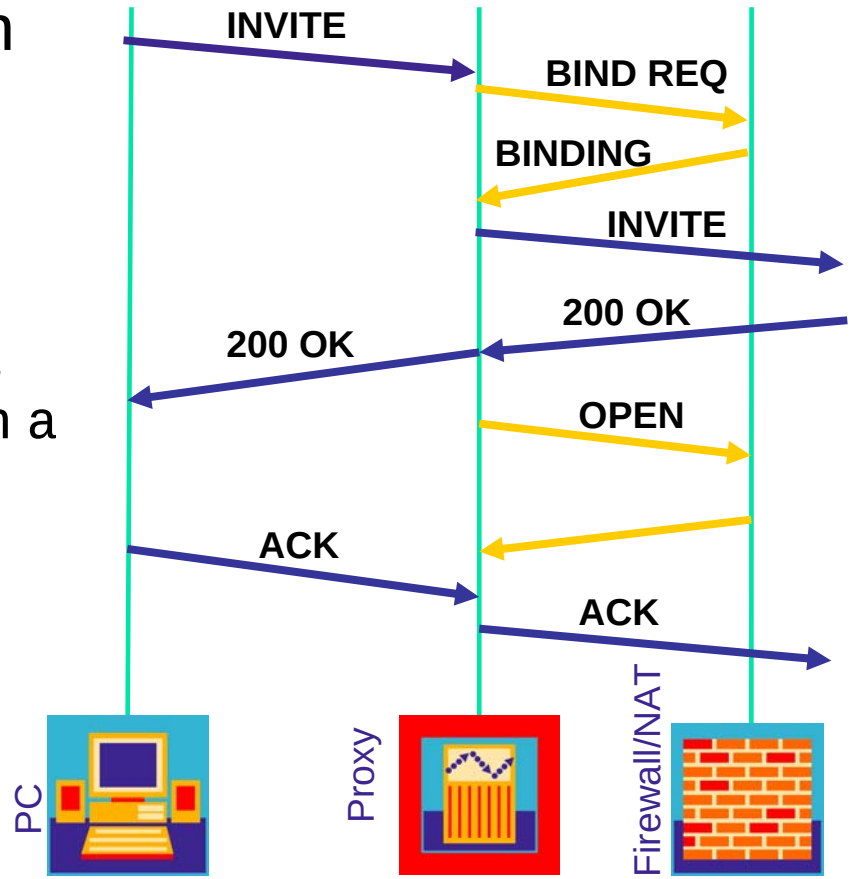
## Separate Application Layer NAT from IP Layer NAT

- Like MEGACO Decomposition
  - **MG = Packet Filter**
  - **MGC = Firewall Control Proxy**
- Advantage
  - **Better scaling**
  - **Load balancing**
  - **Low cost**
  - **Expertise problem solved**



# Solution for Case I – ALG [2/2]

- Control Protocol Between Application Layer NATs and IP Layer NATs
- Main Requirements
  - **Binding Request**: give a private address and obtain a public address
  - **Binding Release**
  - **Open Hole** (firewall)
  - **Close Hole** (firewall)



# Proposed Solution for Case II

**Much harder problem**

- No way to control firewall or NAT
- Cascading NATs
- Variable firewall NAT behaviors

**Proposed Solution**

- Make SIP "NAT-Friendly"
  - Minor extensions
  - Address the issues for SIP only, not RTP
  - Accepted by IETF (RFC 3581)
- Develop a protocol for traversal of UDP through NAT
  - Work for RTP
  - Also support other applications

# SIP Extension to NAT Friendly

## Client Behavior

- Include an “**rport**” parameter in the Via header
  - This parameter MUST have no value
  - It serves as a flag
- The client SHOULD retransmit its INVITE every 20 seconds
  - Due to UDP NAT binding period and to keep the binding fresh

# SIP Extension to NAT Friendly [2/2]

## Server Behavior

- Examine the Via header field value of the request.
  - If it contains an "**rport**" parameter,
    - A "**received**" parameter
    - An "**rport**" parameter
- The response MUST be sent to the IP address listed in the "received" parameter, and the port in the "rport" parameter.

# Example [1/2]

Client A: 10.1.1.1
Proxy B: 68.44.10.3
NAT C: 68.44.20.1

- A issues request
  INVITE sip:user@domain SIP/2.0
  Via: SIP/2.0/UDP 10.1.1.1:4540;**rport**;branch=z9hG4bKkjshdyff

- A→C (mapping port 9988)→B
  INVITE sip:user@domain SIP/2.0
  Via: SIP/2.0/UDP proxy.domain.com;branch=z9hG4bKkjsh77
  Via: SIP/2.0/UDP 10.1.1.1:4540;
  **received=68.44.20.1;rport=9988**;
  branch=z9hG4bKkjshdyff

# Example [2/2]

3) Server B receives the response
SIP/2.0 200 OK
Via: SIP/2.0/UDP proxy.domain.com;branch=z9hG4bKkjsh77
Via: SIP/2.0/UDP
10.1.1.1:4540;**received=68.44.20.1**;**rport=9988**;
branch=z9hG4bKkjshdyff

3) B (68.44.10.3:5060) → C (68.44.20.1:9988) → A
SIP/2.0 200 OK
Via: SIP/2.0/UDP
10.1.1.1:4540;**received=68.44.20.1**;**rport=9988**;
branch=z9hG4bKkjshdyff