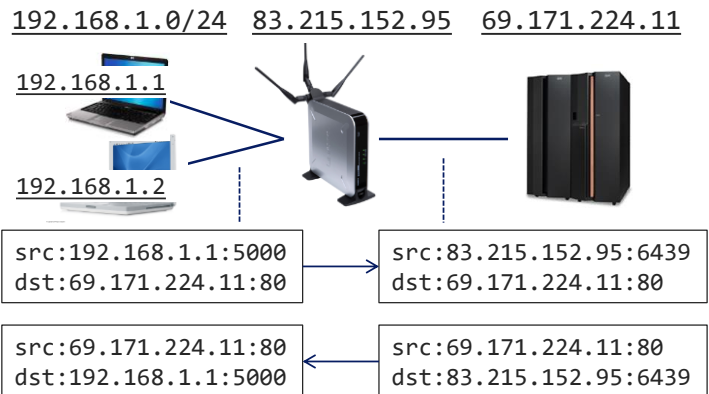# Network Address Translation

Motivation: local network uses just one IP address as far as outside world is concerned:

* Range of addresses not needed from ISP: just one IP address for all devices
* Can change addresses of devices in local network without notifying outside world
* Can change ISP without changing addresses of devices in local network

Network address translation (NAT)

* Connections initiated internally
* Address translation table maps host addr:port to virtual addr:port
* 16-bit port-number field: over 60,000 simultaneous connections

What if I want to run a web server behind a NAT? How will it be reached?

192.168.1.0/24   83.215.152.95   69.171.224.11

192.168.1.1

192.168.1.2

```
src:192.168.1.1:5000
dst:69.171.224.11:80
```
```
src:83.215.152.95:6439
dst:69.171.224.11:80
```
```
src:69.171.224.11:80
dst:192.168.1.1:5000
```
```
src:69.171.224.11:80
dst:83.215.152.95:6439
```

NAT translation table

192.168.1.1:5000 ⟷ 83.215.152.95:6439
192.168.1.2:5000 ⟷ 83.215.152.95:7645

Mountains & Minds

193

NAT traversal problem

* Client wants to connect to server with address 192.168.1.1
  – server address 192.168.1.1 local to LAN (client can't use it as destination addr)
  – only one externally visible NATed address: 83.215.152.95

* Solution1: statically configure NAT to forward incoming connection requests at given port to server
  – e.g., (83.215.152.95, port 80) always forwarded to 192.168.1.1 port 80

192.168.1.0/24   83.215.152.95   69.171.224.11

192.168.1.1   ?

Mountains & Minds

194

1

NAT traversal problem

- Solution 2: Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol.
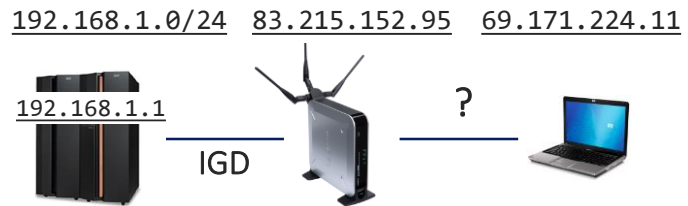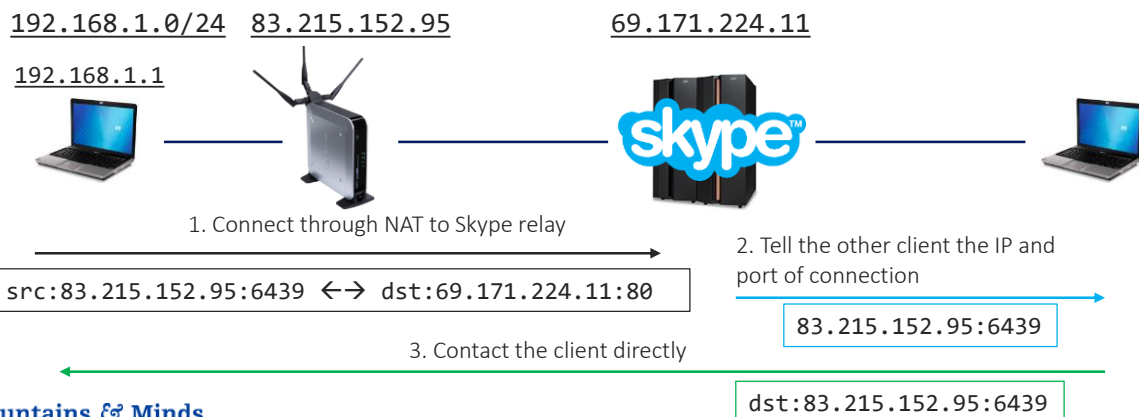
- IGD allows NATed host to:
  - learn public IP address (83.215.152.95)
  - add/remove port mappings (with lease times)

  - i.e., automate static NAT port map configuration

192.168.1.0/24   83.215.152.95   69.171.224.11

192.168.1.1

IGD   ?

Mountains & Minds

195

NAT traversal problem

- Solution 3: relaying (used in Skype)
  - NATed client establishes connection to relay
  - external client connects to relay
  - relay bridges packets between to connections

192.168.1.0/24   83.215.152.95   69.171.224.11

192.168.1.1

1. Connect through NAT to Skype relay

`src:83.215.152.95:6439 ←→ dst:69.171.224.11:80`

2. Tell the other client the IP and port of connection

`83.215.152.95:6439`

3. Contact the client directly

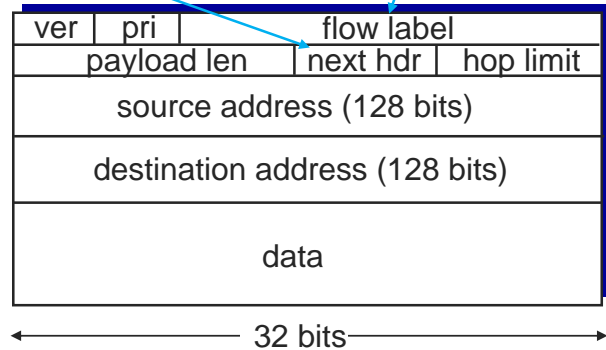`dst:83.215.152.95:6439`

Mountains & Minds

196

2

IPv6

- Motivation
  - 32-bit address space already allocated
  - Header format helps speed processing/forwarding
  - Header changes to facilitate QoS

- Differences from IPv4?
  - Fixed-length 40 byte header
  - Options: allowed, but outside of header, indicated by "Next Header" field
  - No fragmentation allowed
    - ICMPv6 returns "Packet too big" error
  - Checksum: removed entirely to reduce processing time at each hop

Allows flexible options and a fixed length header

QoS labeling

ver	pri	flow label	
payload len		next hdr	hop limit
source address (128 bits)			
destination address (128 bits)			
data			

←——— 32 bits ———→

What are the pros and cons of an expanded "Flow Label" field?
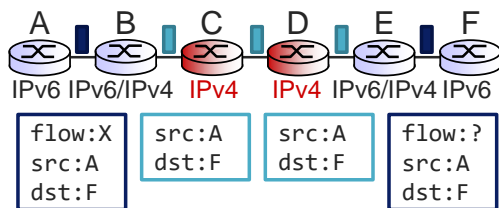
Mountains & Minds

197

Transition from IPv4 to IPv6

How to upgrade all hosts and router from IPv4 to IPv6 simultaneously?
- Basically cannot – *flag day* too complicated!

Dual-stack Approach

A B C D E F
IPv6 IPv6/IPv4 IPv4 IPv4 IPv6/IPv4 IPv6

```
flow:X    src:A    src:A    flow:?
src:A     dst:F    dst:F    src:A
dst:F                       dst:F
```

- IPv6 packets translated to IPv4 in the network
- Loss of IPv6-specific header information

Tunneling Approach

A B C D E F
IPv6 IPv6 IPv4 IPv4 IPv6 IPv6

IPv4 tunnel

```
flow:X   src:B      src:B      flow:X
src:A    dst:E      dst:E      src:A
dst:F                          dst:F
         flow:X     flow:X
         src:A      src:A
         dst:F      dst:F
```
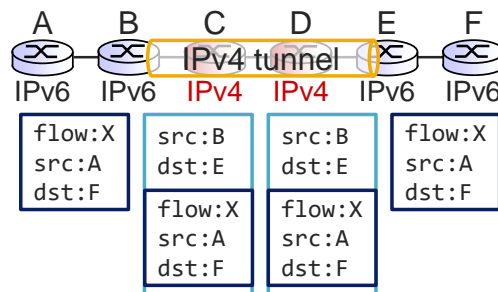
- IPv6 packets encapsulated by IPv4
- Decapsulated at IPv6 capable routers

Mountains & Minds

198

3