ECS503
IPv6

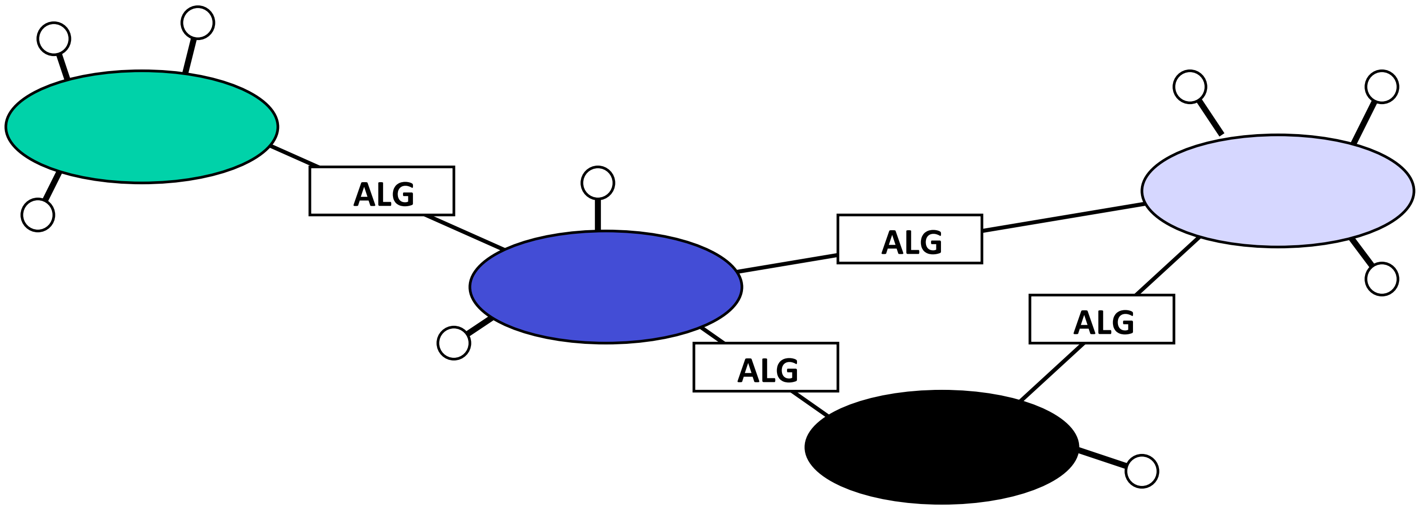
Prof. Steve Uhlig
steve@eecs.qmul.ac.uk
Office: Eng202

Dr. Felix Cuadrado
felix@eecs.qmul.ac.uk
Office: E207

IPv6

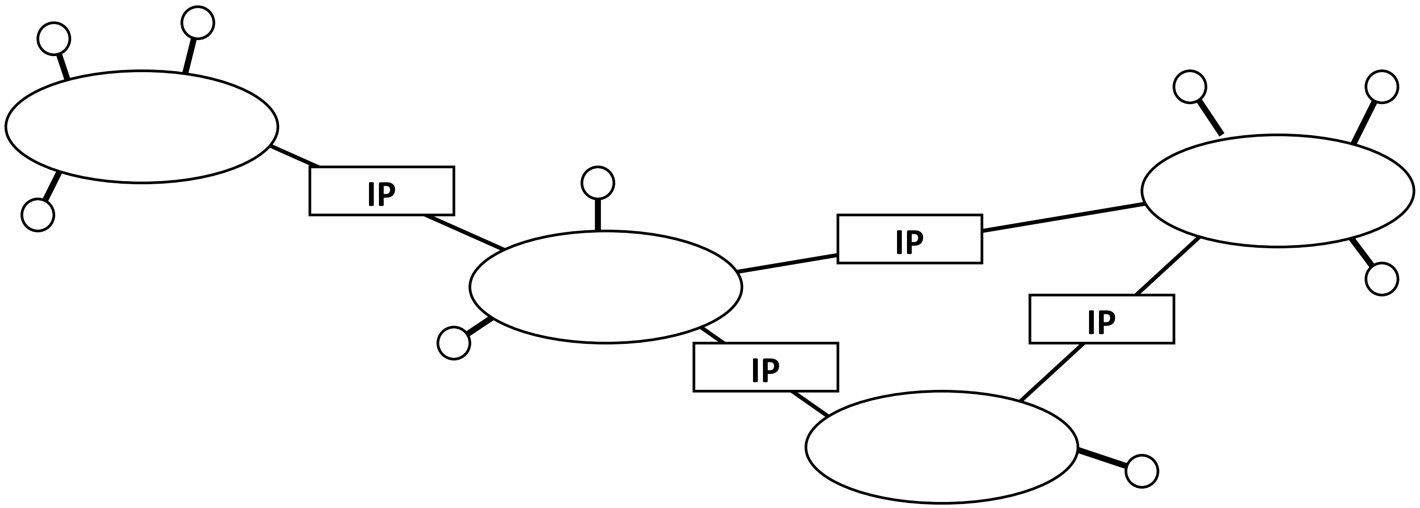
- **Why**
- What
- How
- Deployment
- NAT

Life before IP



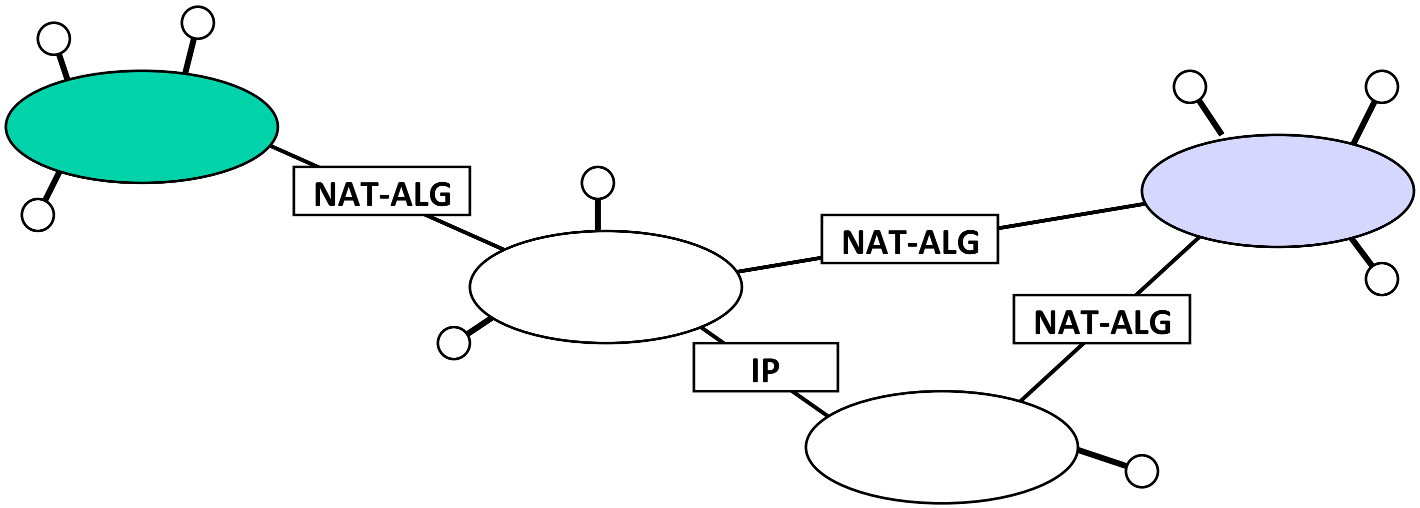
- Application-layer gateways
  - Difficult to deploy new internet-wide applications
  - Hard to diagnose and remedy end-to-end problems
  - Inhibits dynamic routing around failures
  - No global addressability
  - Ad-hoc, application-specific solutions

The IP Solution



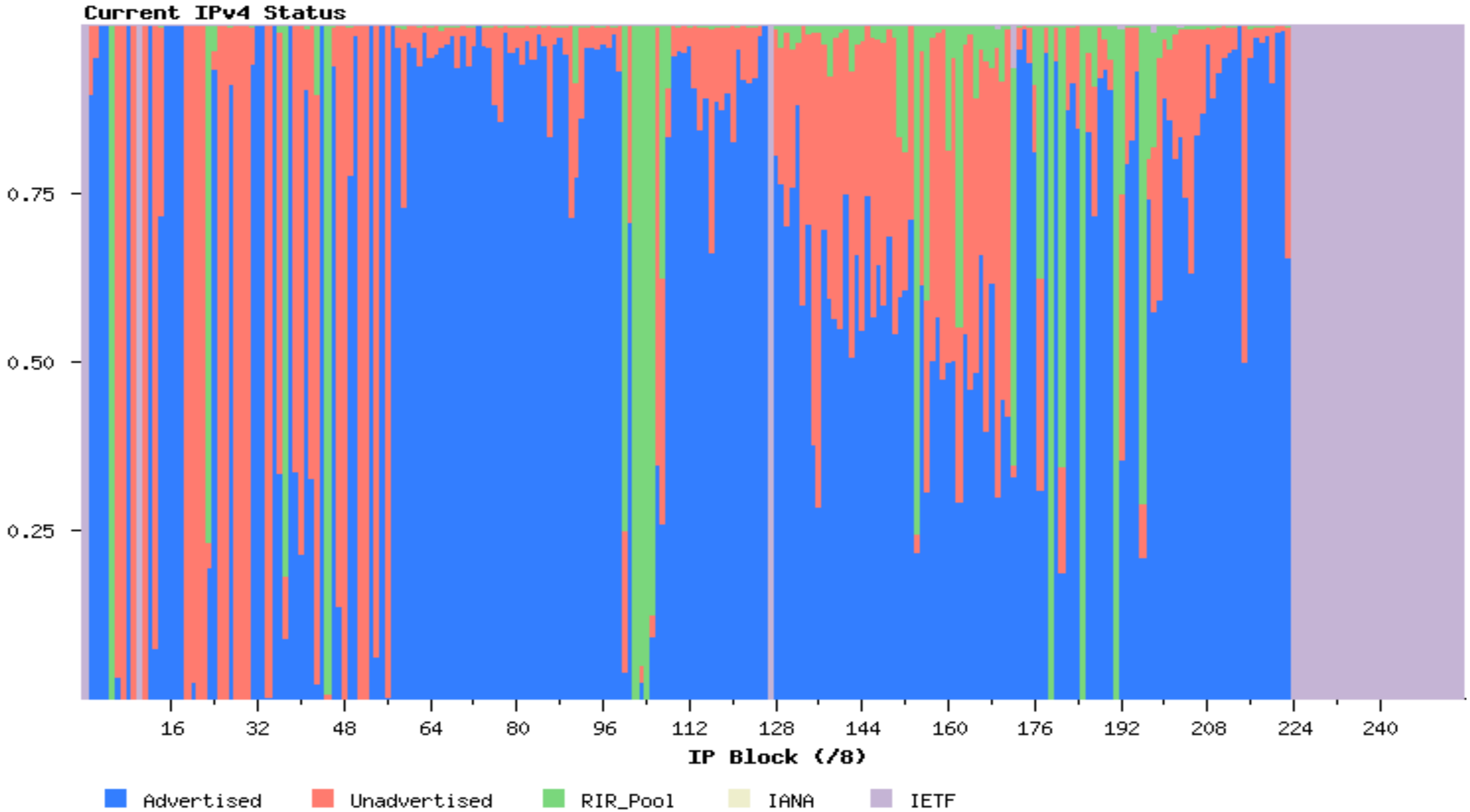
- Internet-layer gateways & global addresses
  - Simple, application-independent network service
  - Easy to route around failures
  - ISPs no longer have monopoly on providing new services
  - Internet became a platform for rapid, competitive innovation

The Internet Today



- NAT and App-layer gateways
  - Difficult to deploy new internet-wide applications
  - Hard to diagnose and remedy end-to-end problems
  - Inhibits dynamic routing around failures
  - No global addressability
  - Ad-hoc, application-specific (or ignorant!) solutions

IPv4 address space status



How much of the IPv4 space is left?

- Not much time left before we run out of IPv4 address space (1-2 years)
- IPv4 addresses are being rationed
  - Consumption statistics tell us nothing about the real demand for addresses, or the hardship created by withholding them
  - The difficulty in obtaining addresses is why many of the NAT-ALGs exist
- New kinds of Internet devices will be much more numerous, and not adequately handled by NATs
  - Mobile phones
  - Cars
  - Home appliances

Why not NAT?

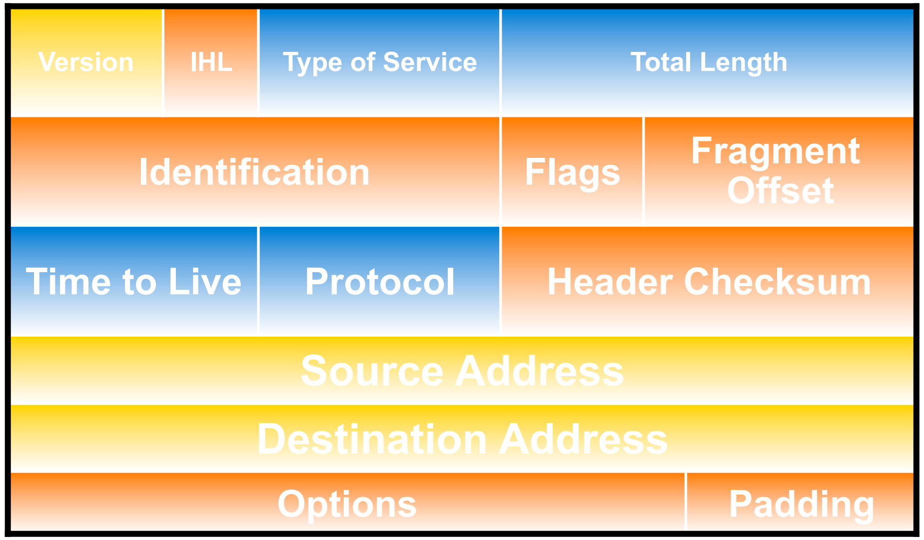
- Not for large numbers of "servers", i.e., devices that are "called" by others (e.g., IP phones)
- Break most current IP multicast and IP mobility protocols
- Break many existing applications
- Limit the market for new applications and services
- Compromise the performance, robustness, security, and manageability of the Internet

IPv6

- Why
- **What**
- How
- Deployment
- NAT

IPv4 and IPv6 Headers

IPv4



IPv6

Legend

- field's name kept from IPv4 to IPv6
- fields not kept in IPv6
- Name & position changed in IPv6
- New field in IPv6

What changed?

Streamlined

- Fragmentation fields moved out of base header: path MTU
- IP options moved out of base header: not used
- Header Checksum eliminated: redundant
- Header Length field eliminated: fixed
- Length field excludes IPv6 header
- Alignment changed from 32 to 64 bits

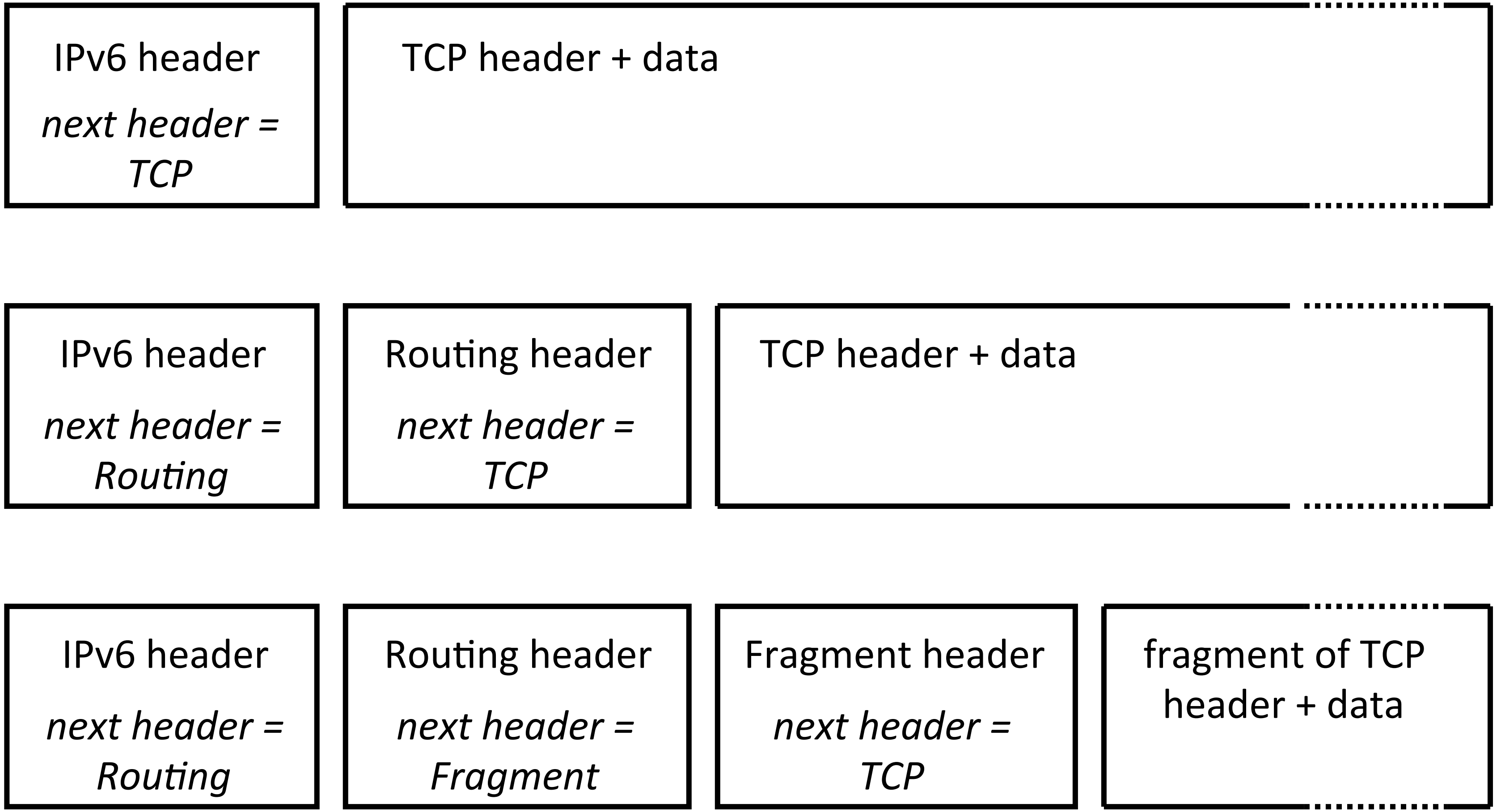
Revised

- Time to Live => Hop Limit
- Protocol => Next Header
- Precedence & TOS => Traffic Class
- Addresses increased 32 bits => 128 bits

Extended

- Flow Label field added

Extension Headers



Address Types

- Unicast (one-to-one)
  - global
  - link-local
  - site-local
  - compatible (IPv4, IPX, NSAP)
- Multicast (one-to-many)
- Anycast (one-to-nearest)
- Reserved

Address Type Prefixes

address type	binary prefix
IPv4-compatible	0000...0 (96 zero bits)
global unicast	001
link-local unicast	1111 1110 10
site-local unicast	1111 1110 11
multicast	1111 1111

all other prefixes reserved (approx. 7/8ths of total)

anycast addresses allocated from unicast prefixes

Global Unicast Addresses

001	TLA	NLA*	SLA*	interface ID

public topology (45 bits)	site topology (16 bits)	interface identifier (64 bits)

- TLA = Top-Level Aggregator
  NLA* = Next-Level Aggregator(s)
  SLA* = Site-Level Aggregator(s)
- All subfields variable-length
- TLAs may be assigned to ISPs or IXPs

Link-Local & Site-Local Unicast Addresses

Link-local addresses for use during auto-configuration and when no routers are present:

1111111010	0	interface ID

Site-local addresses for independence from changes of TLA / NLA*:

1111111010	0	SLA*	interface ID

Multicast Addresses

11111111	flags	scope	group ID
8	4	4	112 bits

Low-order flag indicates permanent transient group; three other flags reserved

scope field:
- 1 - node local
- 2 - link-local
- 5 - site-local
- 8 - organization-local
- B - community-local
- E - global

(all other values reserved)

Routing

- Same "longest-prefix match" routing as IPv4 CIDR
- Straightforward changes to existing IPv4 routing protocols to handle bigger addresses
  - unicast: OSPF, RIP-II, IS-IS, BGP4+,
  - multicast: MOSPF, PIM, …
- Can use Routing header with anycast addresses to route packets through particular regions, e.g., for provider selection, policy, performance, etc.

Serverless Autoconfiguration (“Plug-n-Play”)

- Hosts can construct their own addresses:
  - subnet prefix(es) learned from periodic multicast advertisements from neighboring router(s)
  - interface IDs generated locally, e.g., using MAC addresses
- Other IP-layer parameters also learned from router adverts (e.g., router addresses, recommended hop limit, etc.)
- Higher-layer info (e.g., DNS server and NTP server addresses) discovered by multicast / anycast-based service-location protocol
- DHCP also available for those who want more control

Auto-Reconfiguration ("Renumbering")

- New address prefixes can be introduced, and old ones withdrawn
  - Assume some overlap period between old and new
  - Hosts learn prefix lifetimes and preferably from router advertisements
  - Old TCP connections can survive until end of overlap; new TCP connections can survive beyond overlap
- Router renumbering protocol, to allow domain-interior routers to learn of prefix introduction / withdrawal
- New DNS structure to facilitate prefix changes

Other Features of IPv6

- Flow label for more efficient flow identification (avoids having to parse the transport-layer port numbers)
- Neighbor unreachability detection protocol for hosts to detect and recover from first-hop router failure
- More general header compression (handles more than just IP+TCP)
- Security ("IPsec") & differentiated services ("diff-serv") QoS features — same as IPv4

IPv6

- Why
- What
- **How**
- Deployment
- NAT

IPv4-IPv6 Co-Existence / Transition

Queen Mary University of London

- 3 non-exclusive options:

  (1) dual-stack techniques, to allow IPv4 and IPv6 to co-exist in the same devices and networks

  (2) tunneling techniques, to avoid order dependencies when upgrading hosts, routers, or regions

  (3) translation techniques, to allow IPv6-only devices to communicate with IPv4-only devices

- Expect all of these to be used, in combination

Dual-Stack Approach

- When adding IPv6 to a system, do not delete IPv4
  - Multi-protocol approach: familiar and well-understood (e.g., for AppleTalk, IPX)
  - In most cases, IPv6 will be bundled with new OS releases, not an extra-cost add-on
- Applications choose IP version to use
  - When initiating, based on DNS response: if (dest has AAAA or A6 record) use IPv6, else use IPv4
  - When responding, based on version of initiating packet
- Allows indefinite co-existence of IPv4 and IPv6, and gradual, app-by-app upgrades to IPv6 usage

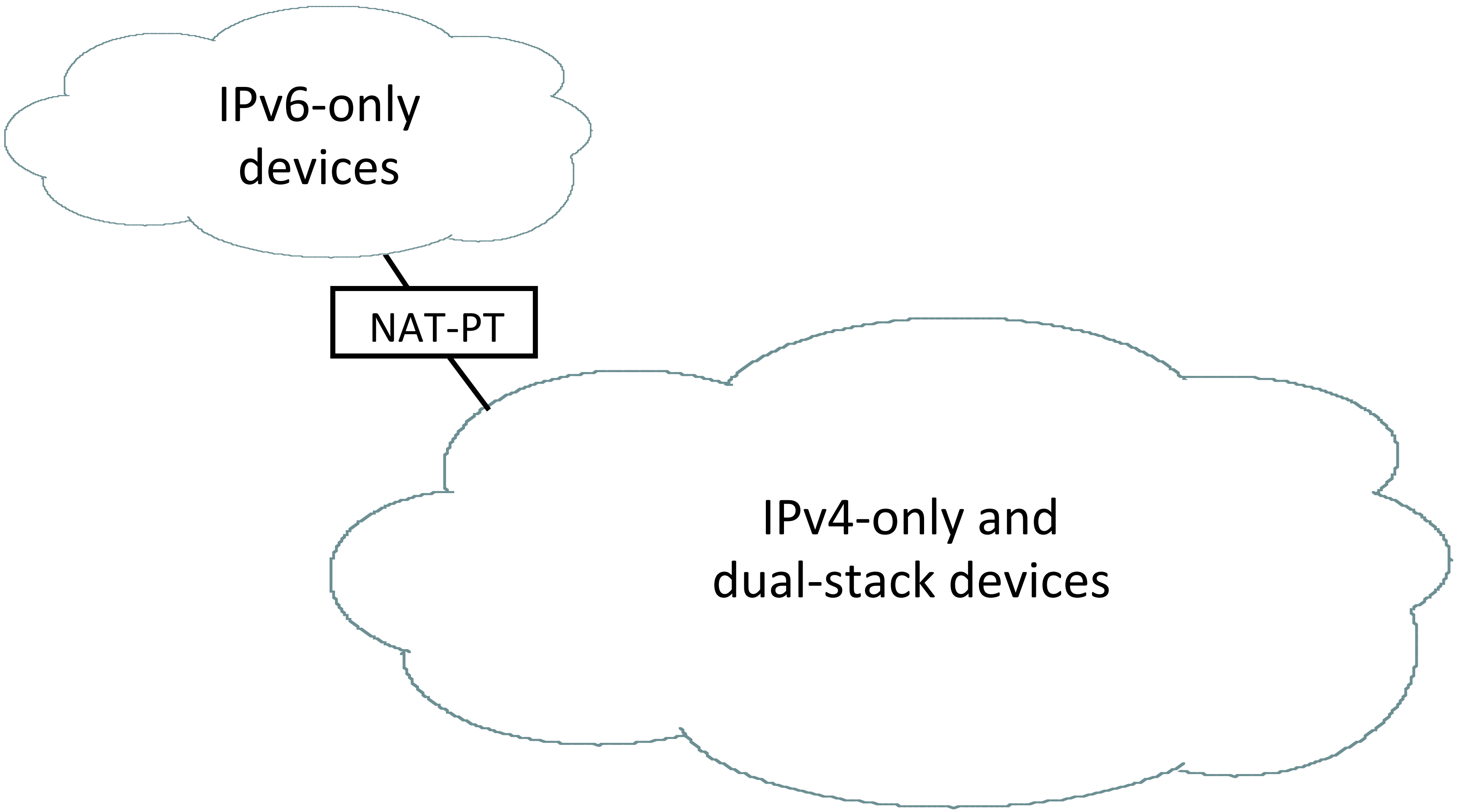
Tunnels: dealing with non-IPv6 Routers/Switches

- Encapsulate IPv6 packets inside IPv4 packets (or MPLS frames)
- Many methods exist for establishing tunnels:
  - manual configuration
  - "tunnel brokers" (using web-based service to create a tunnel)
  - "6-over-4" (intra-domain, using IPv4 multicast as virtual LAN)
  - "6-to-4" (inter-domain, using IPv4 addr as IPv6 site prefix)
- Can view this as:
  - IPv6 using IPv4 as a virtual link-layer, or
  - IPv6 VPN (virtual public network), over the IPv4 Internet (becoming "less virtual" over time, we hope)

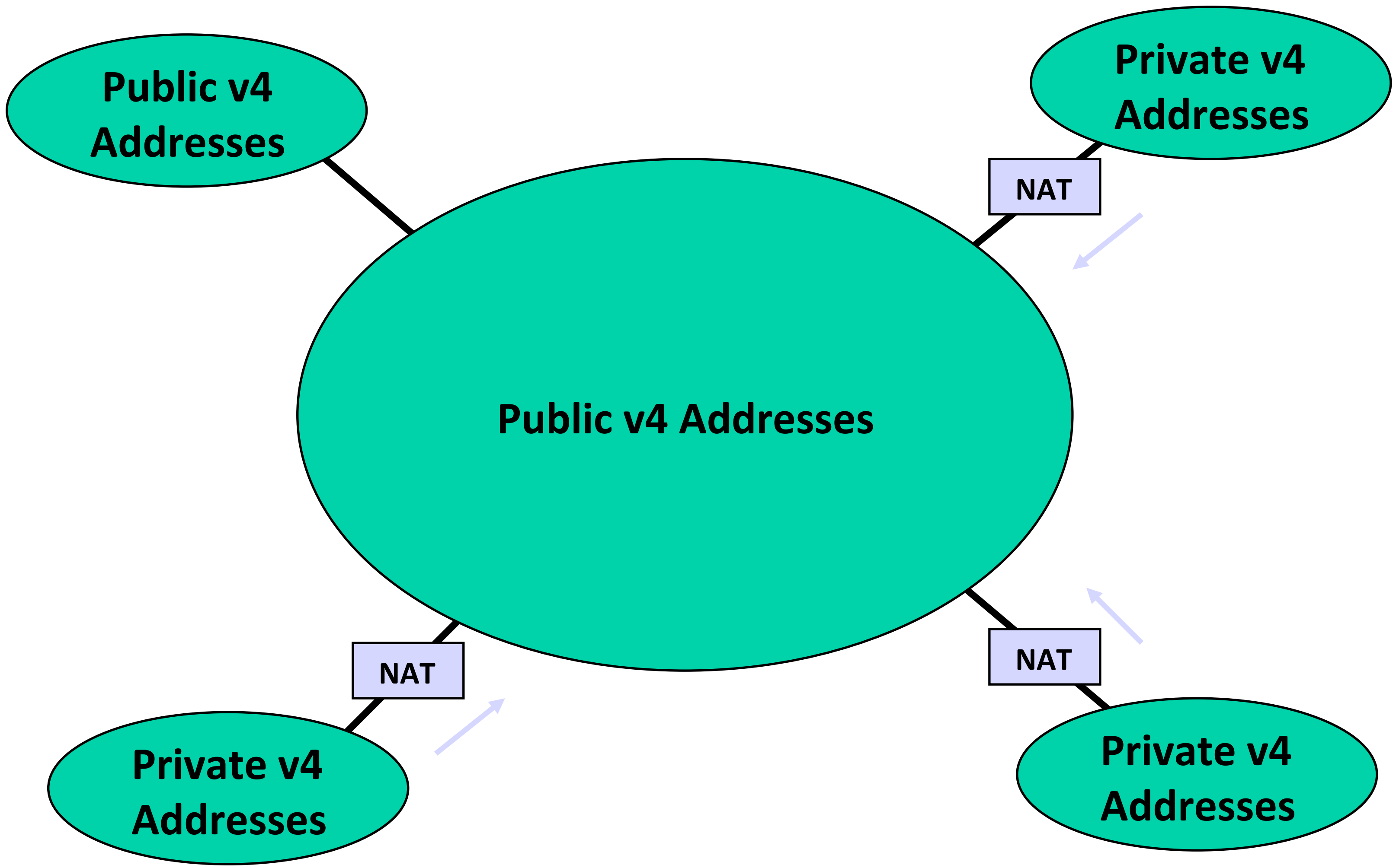
Translation

- IPv6-IPv4 protocol translation for:
  - new kinds of Internet devices (e.g., cell phones, cars, appliances)
- NAT extension: translate header format as well as addresses
  - IPv6 nodes behind a translator get full IPv6 functionality when talking to other IPv6 nodes located anywhere
  - They get the normal (i.e., degraded) NAT functionality when talking to IPv4 devices
- Alternative: transport-layer relay or app-layer gateways

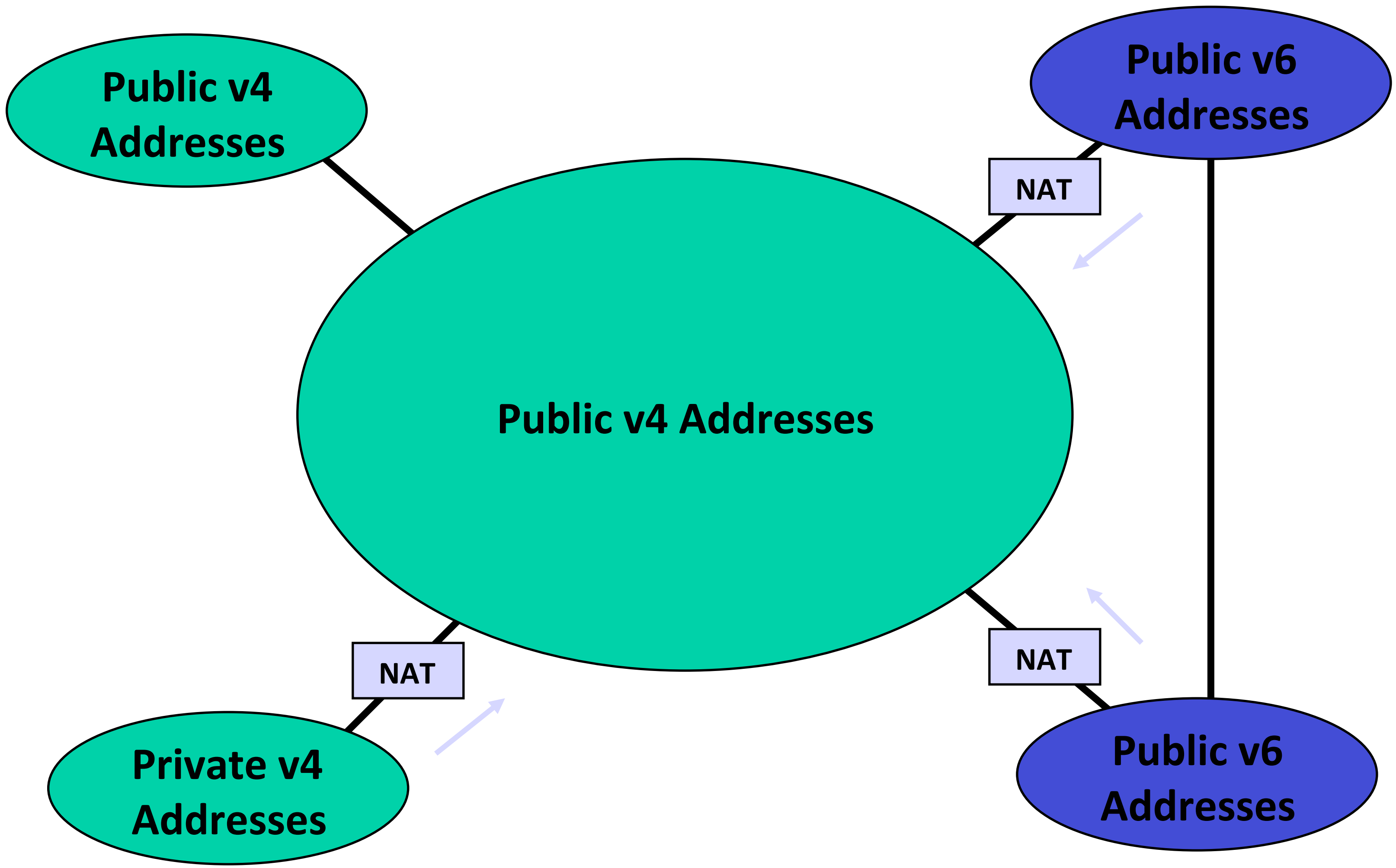
NAT and Protocol Translation (NAT-PT)



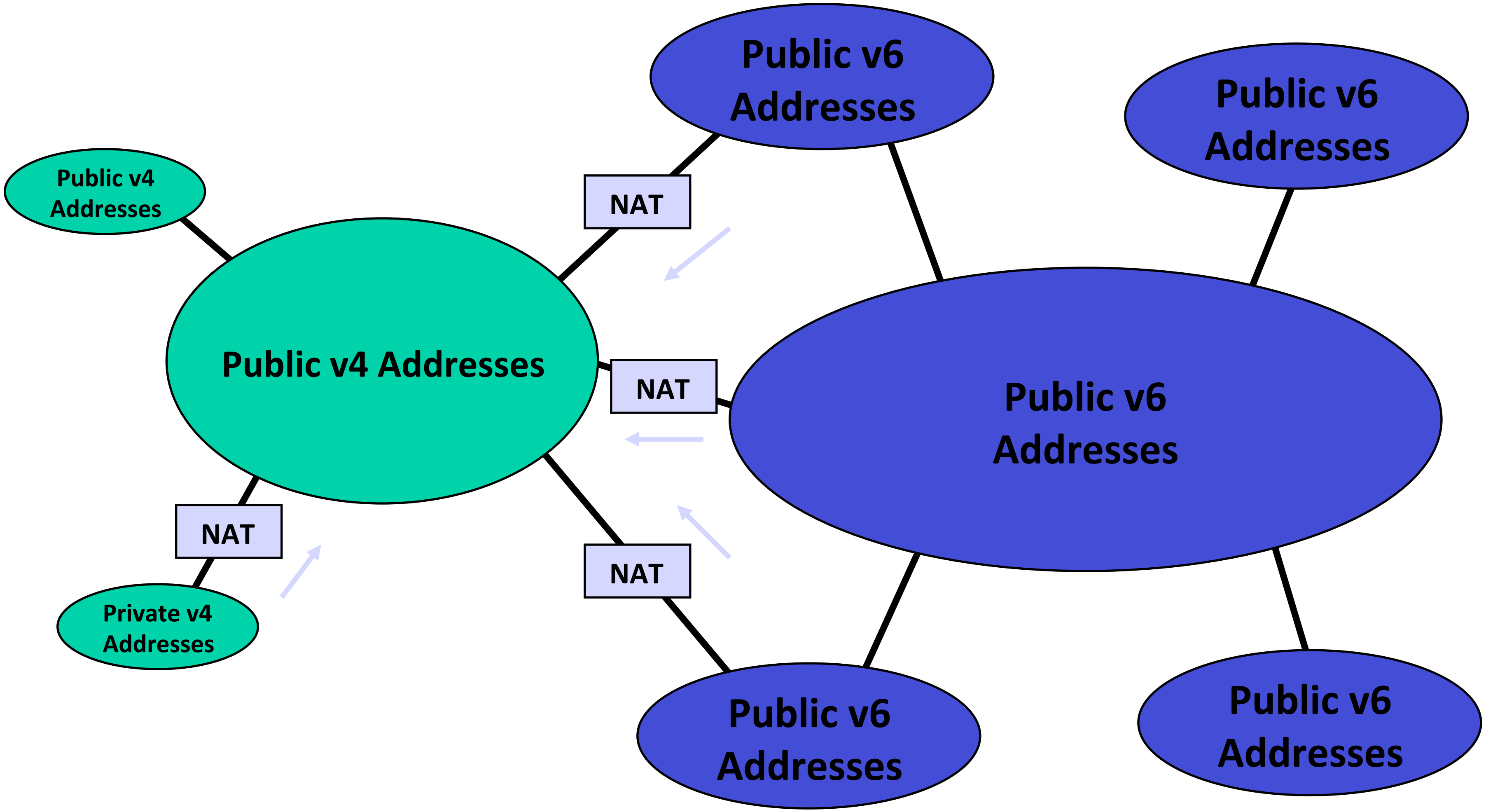
The IPv4 Internet Today



Introducing IPv6



Expanding IPv6



IPv6

- Why
- What
- How
- **Deployment**
- NAT

Status

- Check status at: http://ipv6.com/articles/deployment/IPv6-Deployment-Status.htm
- Commercial ISPs: deployed dual-stack but waiting…
- Asia is strong
- Traffic: not much
- Efforts such as IPv6 day: http://www.worldipv6day.org/

IPv6 deployment issues

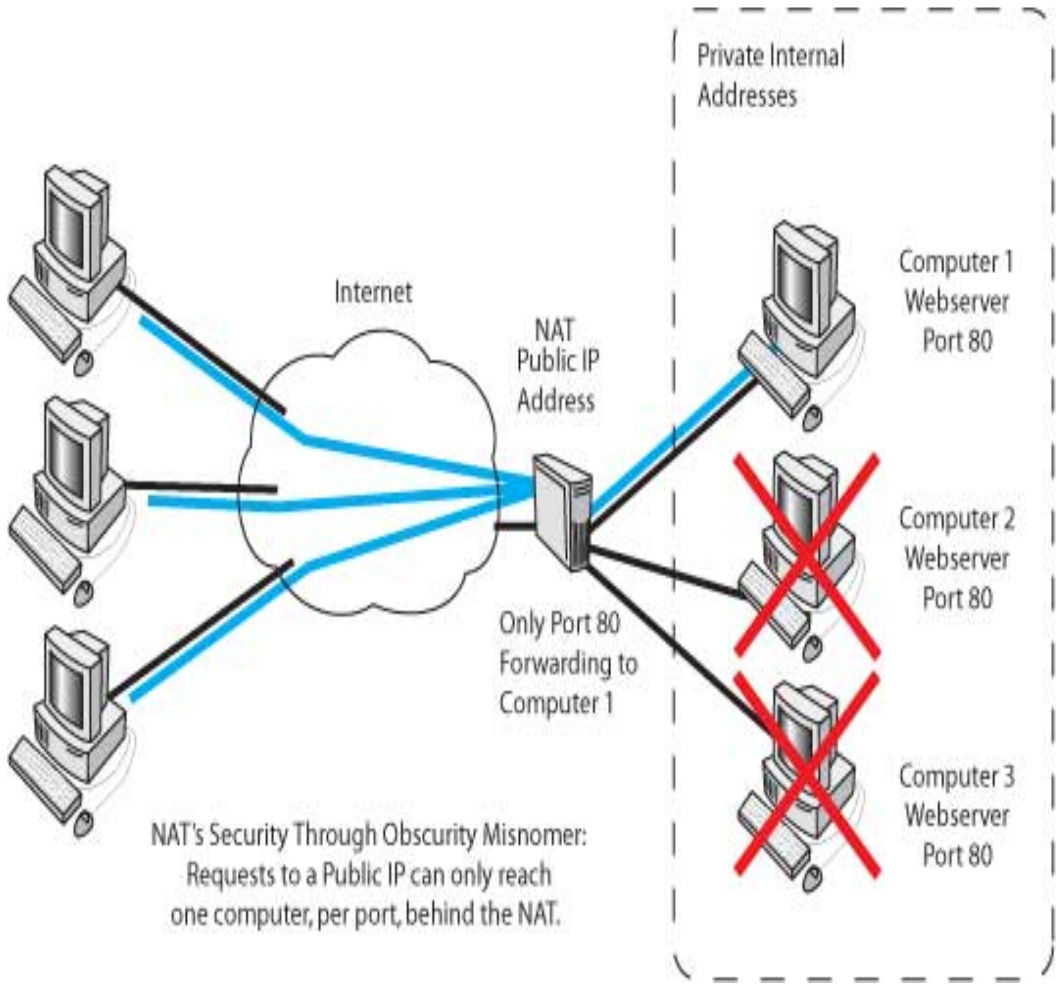
- NAT: background
- NAT traversal techniques
- Discussion: Carrier-grade NAT
- IPv6 World Day
- http://test-ipv6.com/

IPv6

- Why
- What
- How
- Deployment
- **NAT**

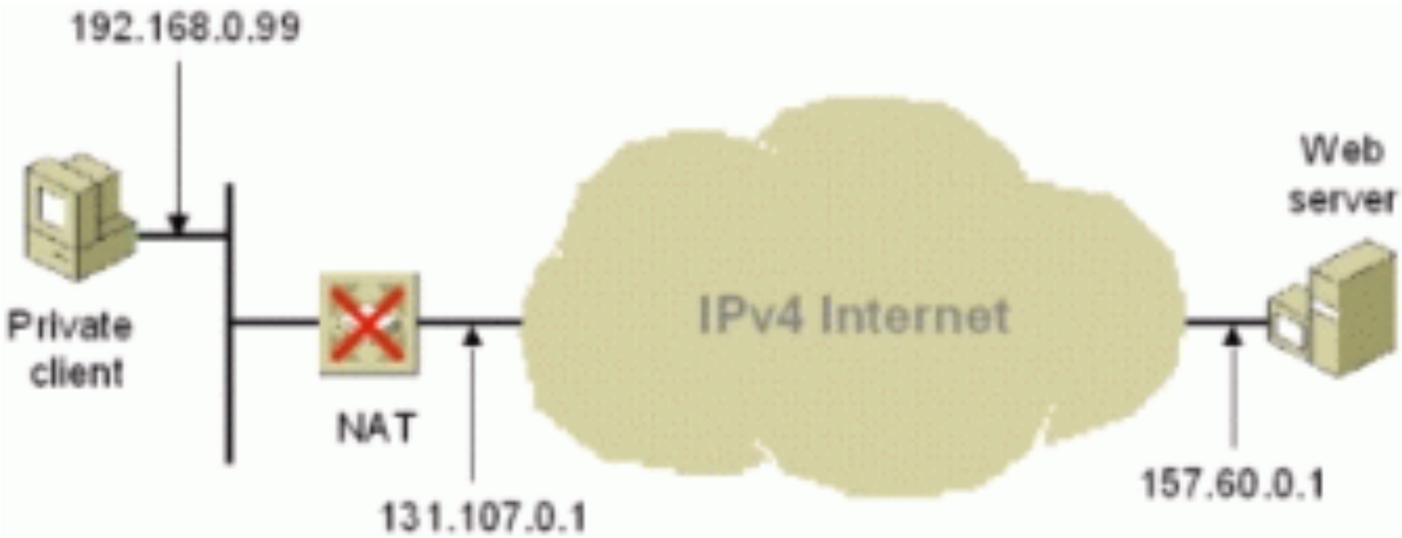
NAT

- Translation of IP address and TCP/UDP ports by a router
- Why?
  - Sharing Internet connectivity
  - Usage of private IP space
  - Access/provide resources without proxy
  - Security



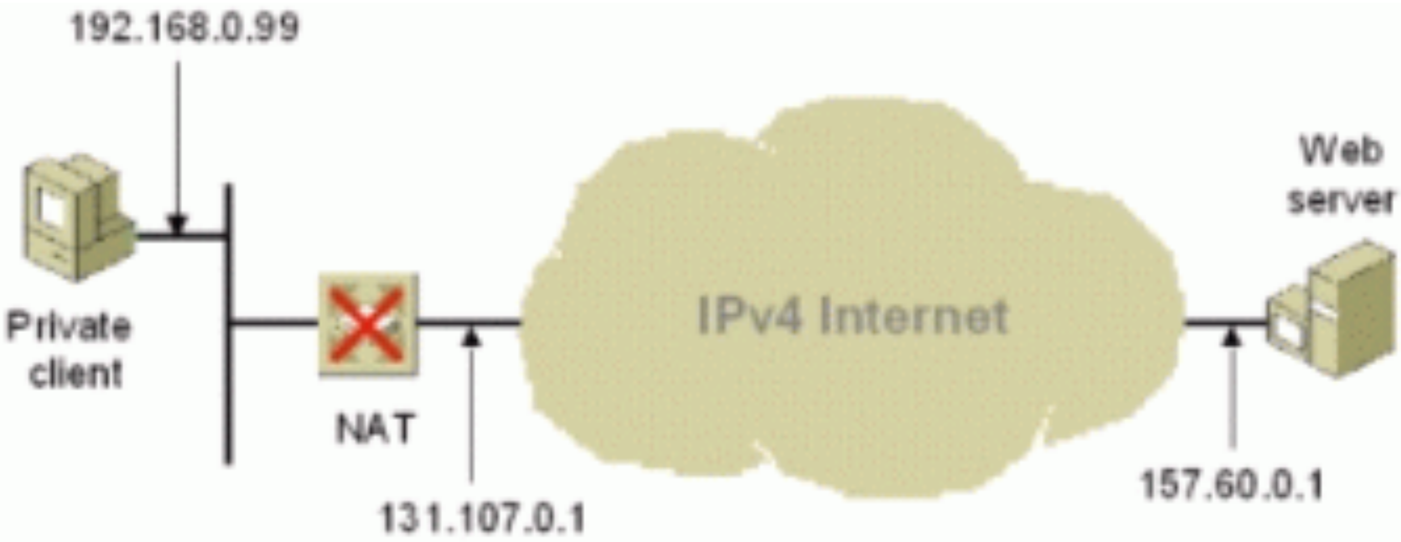
Static NAT

- Maps an unregistered IP address to a registered one, statically
- Useful for Web server or data-centers
- NAT bindings not removed



Dynamic NAT

- Mapping between IP and ports dynamic
- Useful when host on private network initiates communication
- NAT bindings end when communication ends



Types of NAT

- Traditional or outbound NAT: allows hosts on the private IP space to access the Internet
- Bi-directional or two-way NAT: allows both sides to initiate communication
- Twice NAT: IP addresses on both sides are remapped
- Multihomed NAT: one or multiple NAT boxes that share state for failover

NAT traversal: motivation

- How do applications talk when hosts are behind a NAT?
  - Well-known ports, e.g., HTTP
  - Dynamic ports, e.g., P2P

NAT traversal: Skype

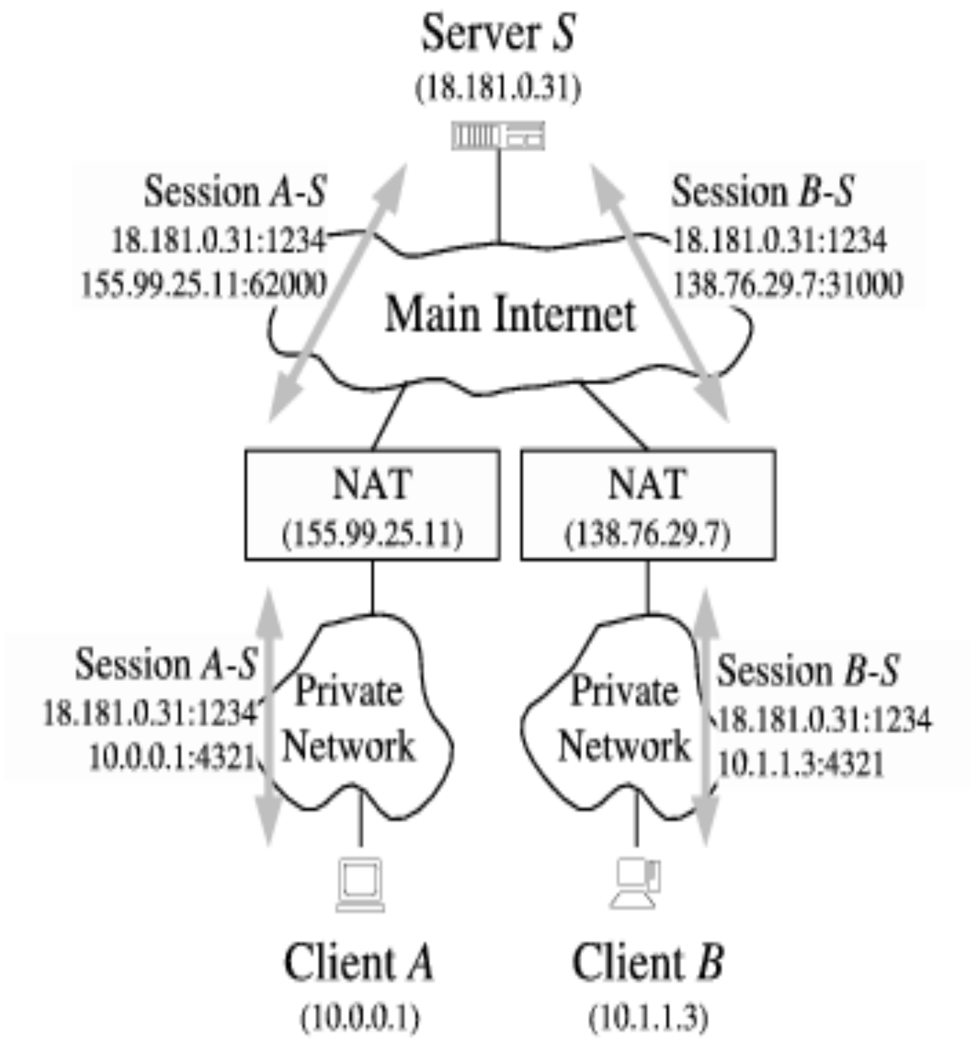
- With cooperation of the NAT
  - SOCKS5/HTTP proxy
- Without cooperation of the NAT
  - TCP/UDP relay: Skype P2P network will do the job
  - Native NAT traversal: hole punching

SOCKS/UPnP

- Client-server protocol that allows a client behind a NAT/firewall to connect to a server in the Internet
- Operations: bind and connect
- Widely supported by browsers, e.g., Mozilla
- Not always supported by NAT

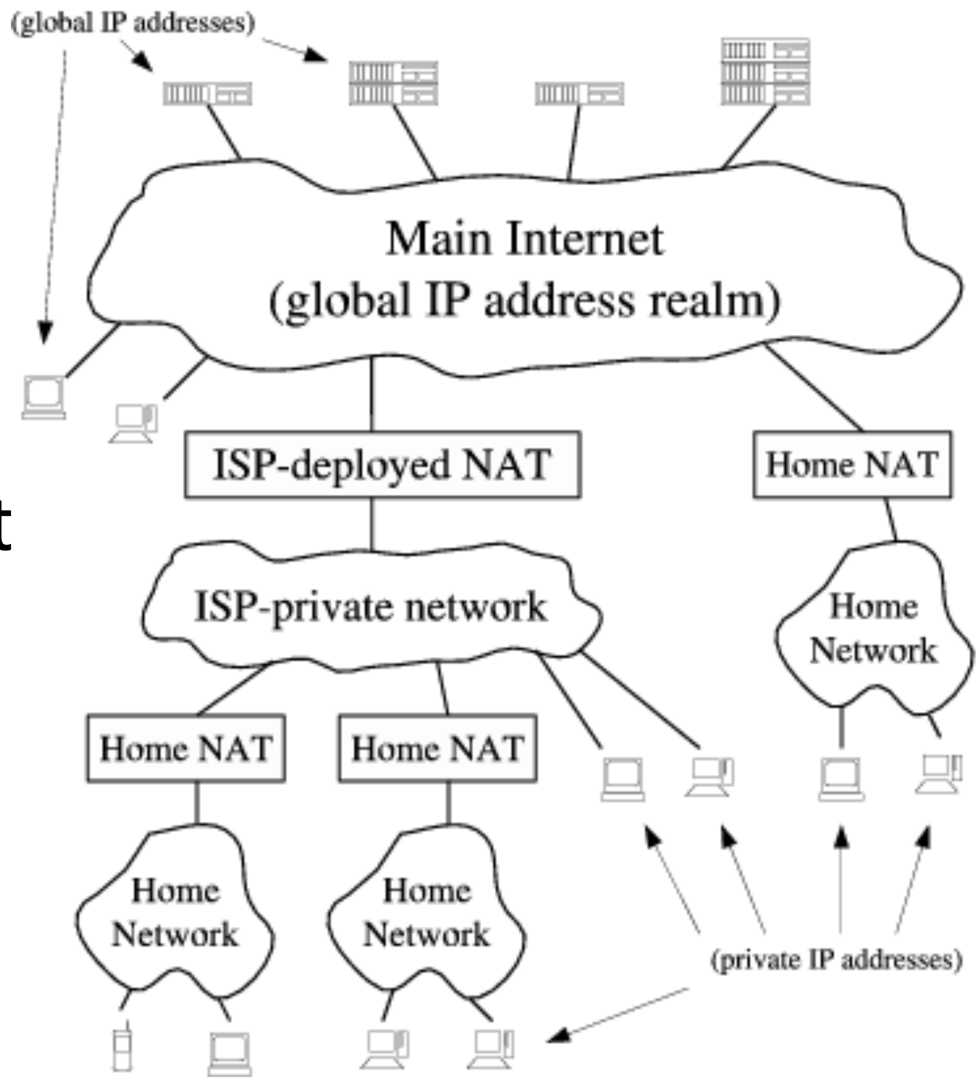
Relaying

- Use a server S to relay packets between A and B
- Reliable but not efficient: S is a bottleneck and point of failure



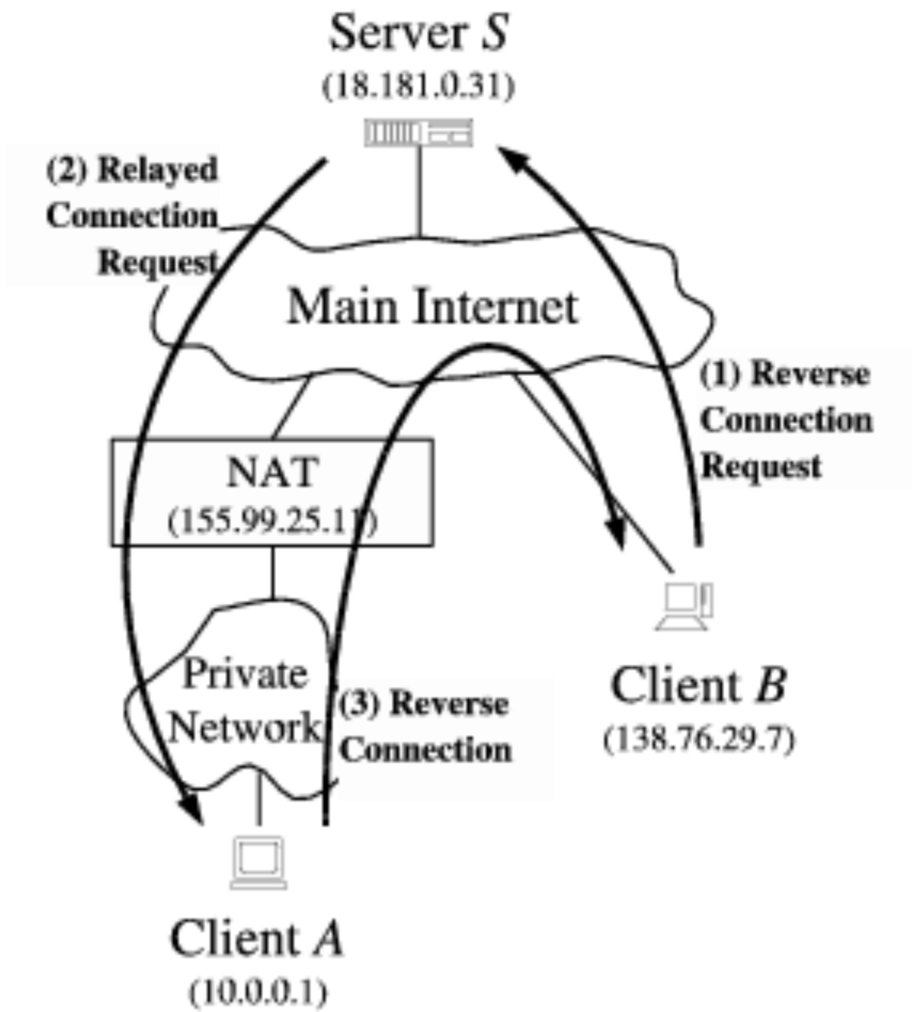
Hole punching

- NATs do not always cooperate
- Making assumptions about NAT/firewalls presence in the Internet is hard
- Need a more generic solution: hole punching



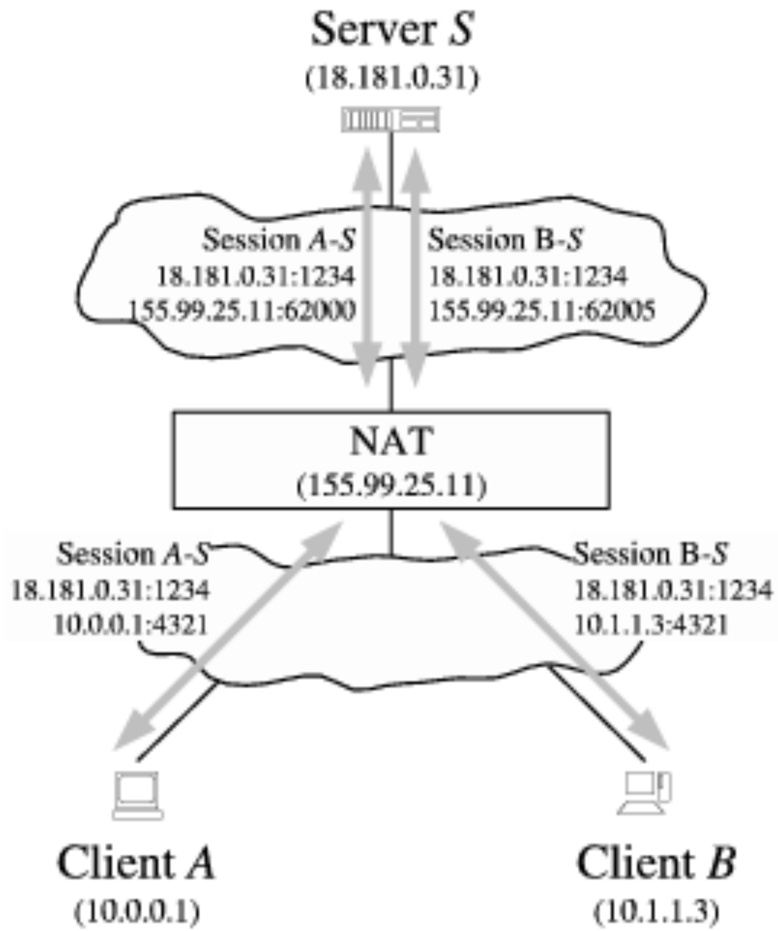
Connection reversal

- If B is not behind a NAT
- Let S relay the connection request
- A can then directly contact B by reversing the connection



Hole punching (same NAT)

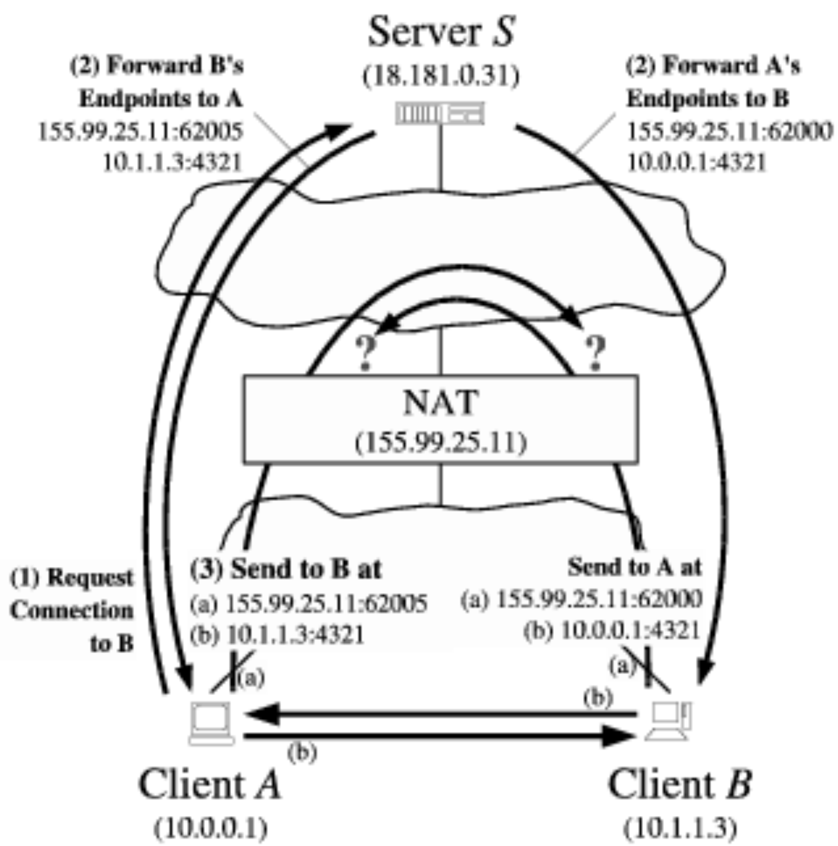
1. Clients A and B contact Server S
2. S receives both A and B’s private IP and ports



Before Hole Punching

Hole punching (same NAT)

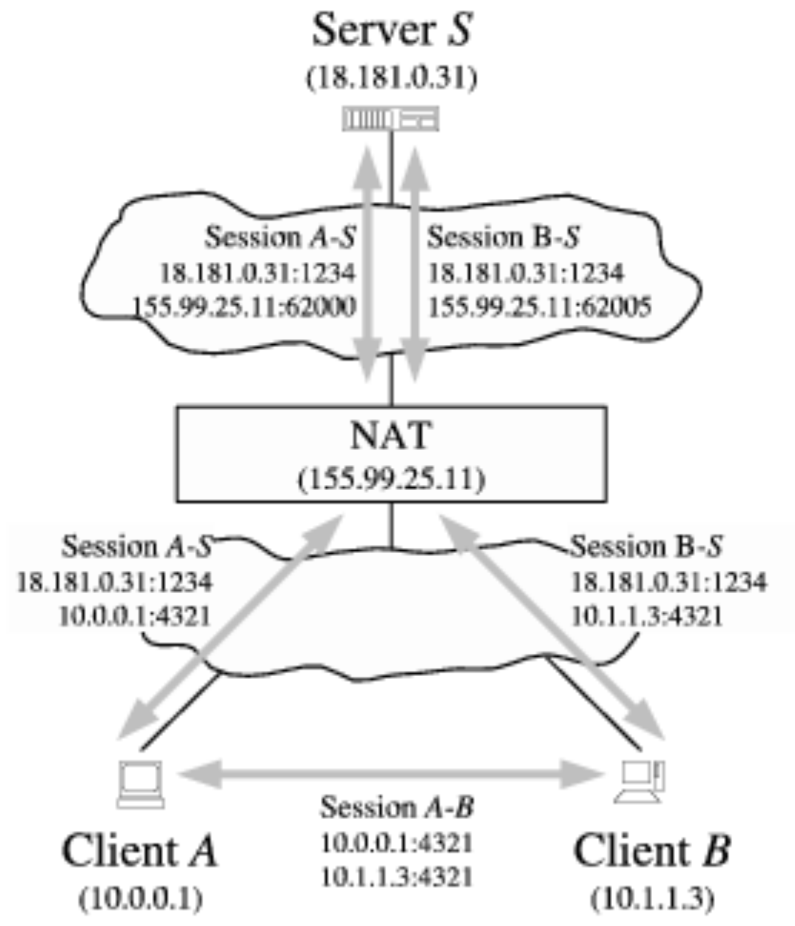
1. A and B ask S help
2. S sends them back each other's private and public IP/ports
3. A and B try to send UDP packets towards the public and private IP/ports
4. Public packets get dropped in private network, private get through



The Hole Punching Process

Hole punching (same NAT)

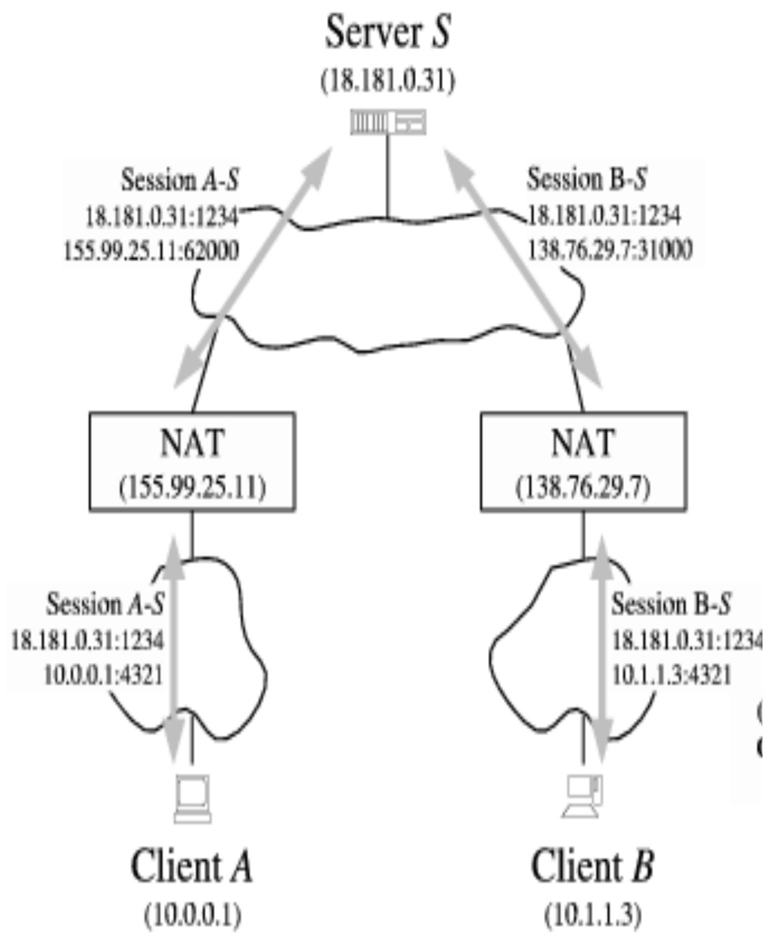
- A and B both have a session between each other within the private network



After Hole Punching

Hole punching (≠ NAT)

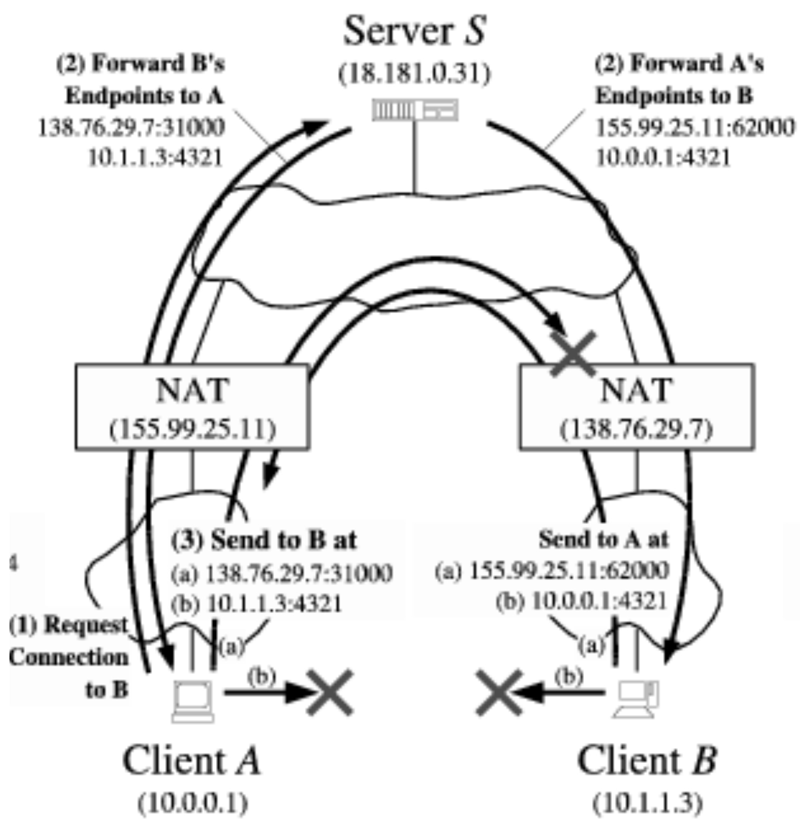
1. Clients A and B contact Server S
2. S receives both A and B's private IP and ports



Before Hole Punching

Hole punching (≠ NAT)

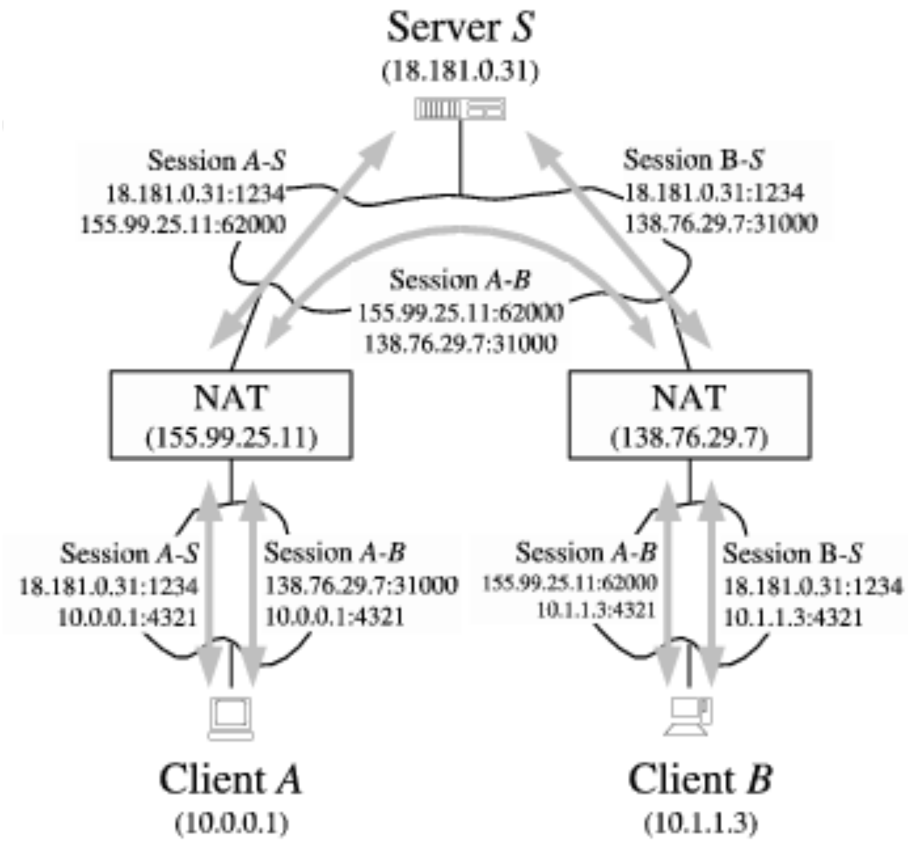
1. A and B ask S help
2. S sends them back each other's private and public IP/ports
3. A and B try to send UDP packets towards the private and public IP/ports
4. Private packets get dropped, public get through



The Hole Punching Process

Hole punching (≠ NAT)

- A and B both have a session between each other, as well as through S



After Hole Punching

Carrier-grade NAT

- NAT is already deployed everywhere
- Why not have ISPs deploy NATs for ALL hosts?
- Pro's
  - IPv6 deployment solved
  - NAT is there anyway
- Con's
  - Breaks e2e principle
  - Stateful: security and reliability
  - Forget about well-known ports
  - Still need more IPv4 addresses