CS-340 Introduction to Computer Networking

Lecture 9: NAT and IPv6

Steve Tarzia

Many diagrams & slides are adapted from those by J.F Kurose and K.W. Ross

Last Lecture: IPv4 Addressing

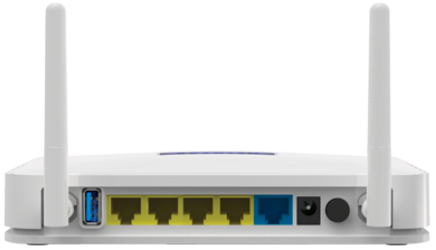
- *IP routing* gets packets to their destination on the net.
- Each router has a *forwarding table* mapping addresses → outbound links.
  - Uses *longest prefix matching*.
- IPv4 *fragments* packets larger than *MTU*. Are reassembled at the destination.
- IPv4 header is 20 bytes (UDP header is 8 bytes or TCP header is 20 bytes)
- IP *subnets* define ranges of address that can communicate directly
  - *CIDR notation* (123.100.16.0/28) specifies a range of addresses
    - Used both for specifying subnets and for routing rules.
    - /28 or 255.255.255.240 is called a *subnet mask*.
- Host's IP configuration is: *address*, *subnet mask*, *gateway*, and *DNS server*
  - *Gateway* is IP address of the router who will route packets outside the subnet
  - *DHCP* allows newly-arriving machines to request an IP configuration.

Network Address Translation (NAT)

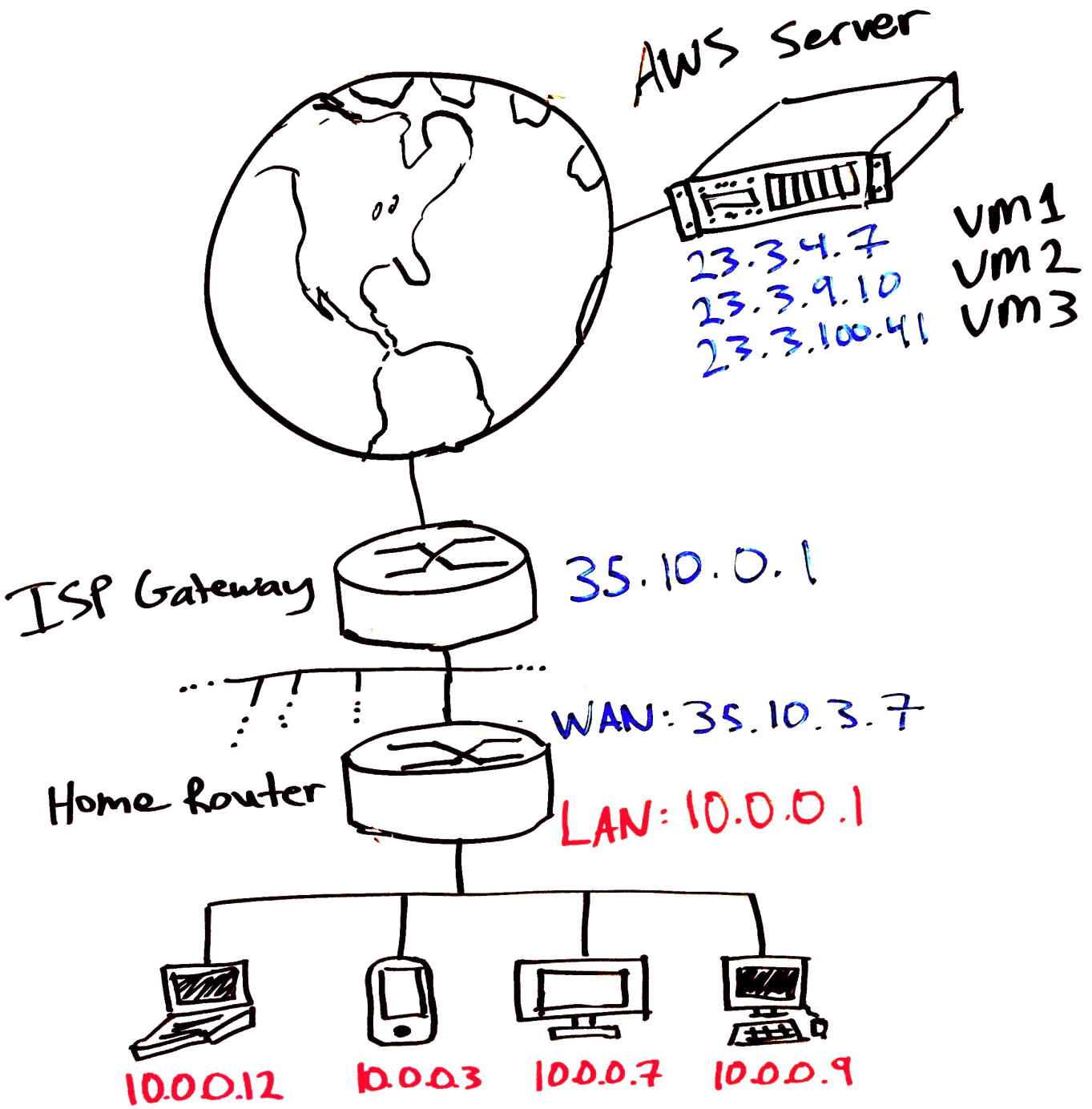
- IPv4 addresses are in short supply (4 billion). ISP often will give you just one address, so how to connect multiple devices?
- ***Key insight***: OS already shares one IP address with multiple independent processes running on one machine, using *port numbers*.

- Make the entire local network look like one big host:

Public ports map to ⟨local_IP_address, port⟩ pairs on the local network.



- NAT router must track this mapping and translate IP addresses on packets leaving/entering local network.



Public Internet

Private LAN

Looks like one big

35.10.3.7 to outside world

Public/Private address translation:

- Source address is **private**
- Dest. address is **public**

Translate private to public

- Source address is **public**
- Dest. address is **public**

- Source address is **public**
- Dest. address is **public**

Translate public to private

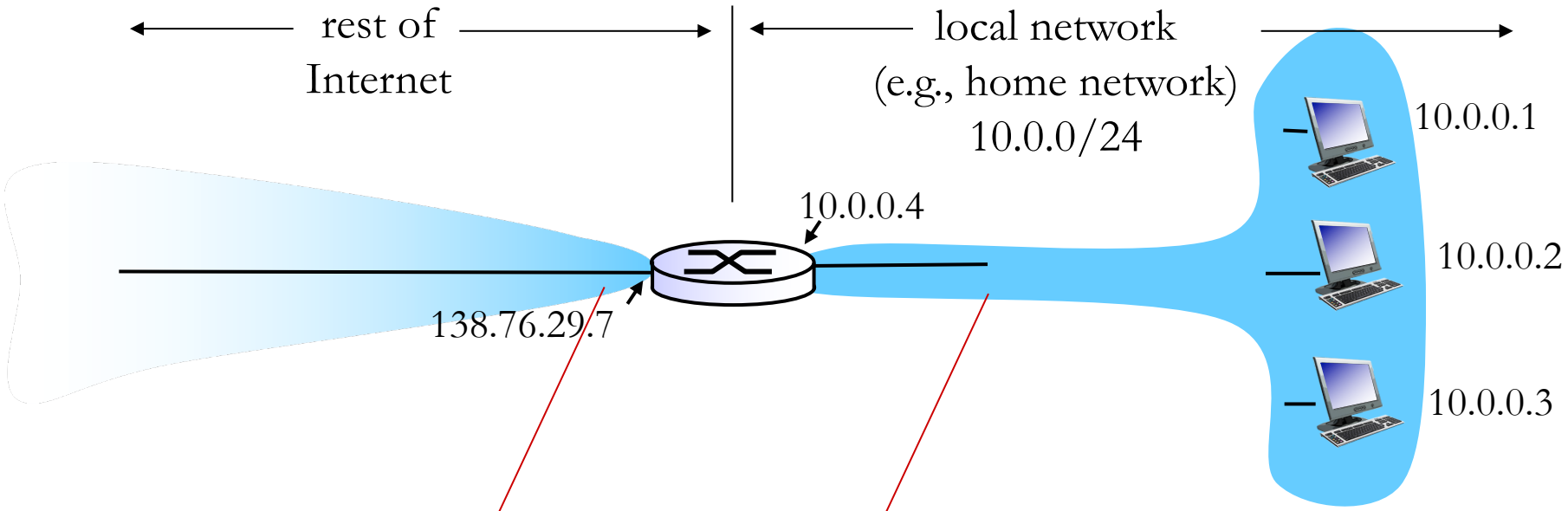
- Source address is **public**
- Dest. address is **private**

Private versus public IP addresses

- **Public IP addresses** specify locations on the Internet.
  - Internet is sometime called the *wide-area network (WAN)*
  - Eg: front-end servers, university campus, home router WAN
  - Only one machine has the public IP address 54.245.121.172
- **Private IP addresses** are meaningful only on a local network.
  - Called *local-area network (LAN)*
  - Eg.: home, office, back-end server
  - *Usually* behind a NAT, to give hosts access to the Internet.
  - Always within: 10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16
  - Eg., millions of machines have the *private* IP address 192.168.0.100

NAT high-level view

- Local 10.0.0.0/24 subnet is not part of the public Internet.
- *Outgoing* packets can reach destination easily
- *Incoming* packets must somehow be routed to the correct local machine. How?



all datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7, different source port numbers for different local machines

datagrams with source or destination in this network have a 10.0.0.0/24 address for source, destination (as usual)

NAT implementation

NAT router takes these actions:

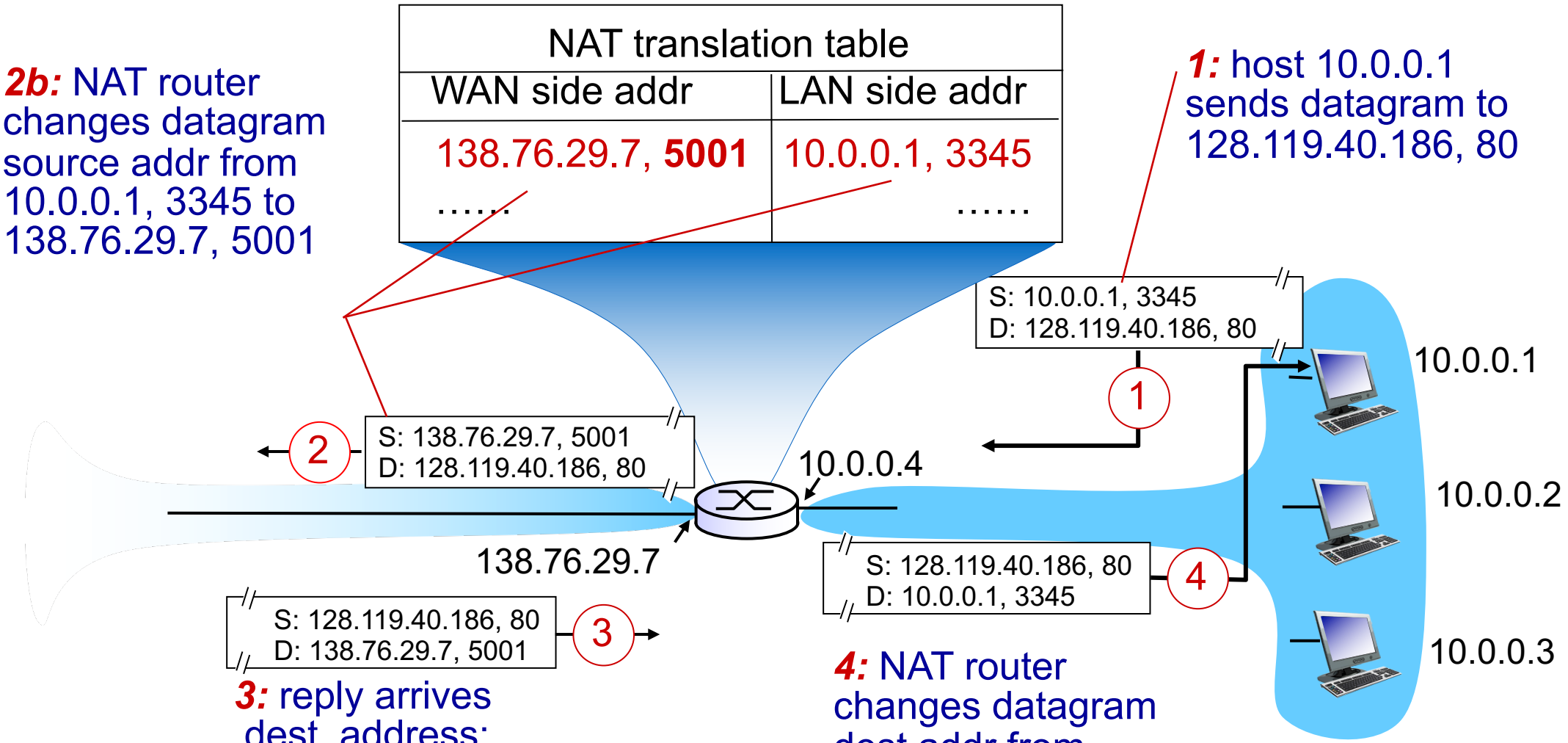
- **Outgoing datagrams:** ***replace*** (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #).
  - Remote host will respond using (public NAT IP address, new port #) as destination address.
  - ***Remember*** (in NAT translation table) every (source IP address, port#) to (NAT IP address, new port #) translation pair.
- **Incoming datagrams:** ***replace*** (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table.

NAT example

2a: NAT router randomly chooses the unused public port 5001 for the connection, and stores the translation.

2b: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80



3: reply arrives dest. address: 138.76.29.7, 5001

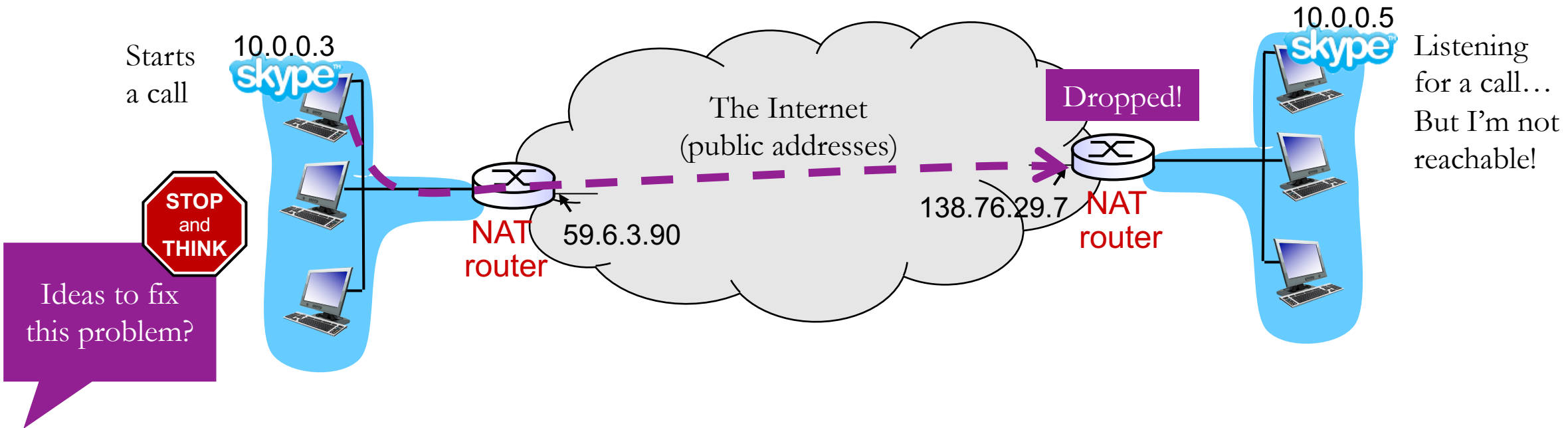
4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

NAT difficulties

- Local clients can only be reached by public IPs that they recently contacted. In other words, cannot easily run services behind a NAT.
  - Special NAT-router configuration called *port forwarding* determines where to send unsolicited packets.
- Some protocols (eg., SIP) advertise IP address and port in payload of packet, which will not be translated by NAT.
- Temporarily-inactive TCP connections may need to send keepalive packets that the NAT router does not "forget" the port mapping.
  - NAT's port mappings timeout eventually to make room for later connections.

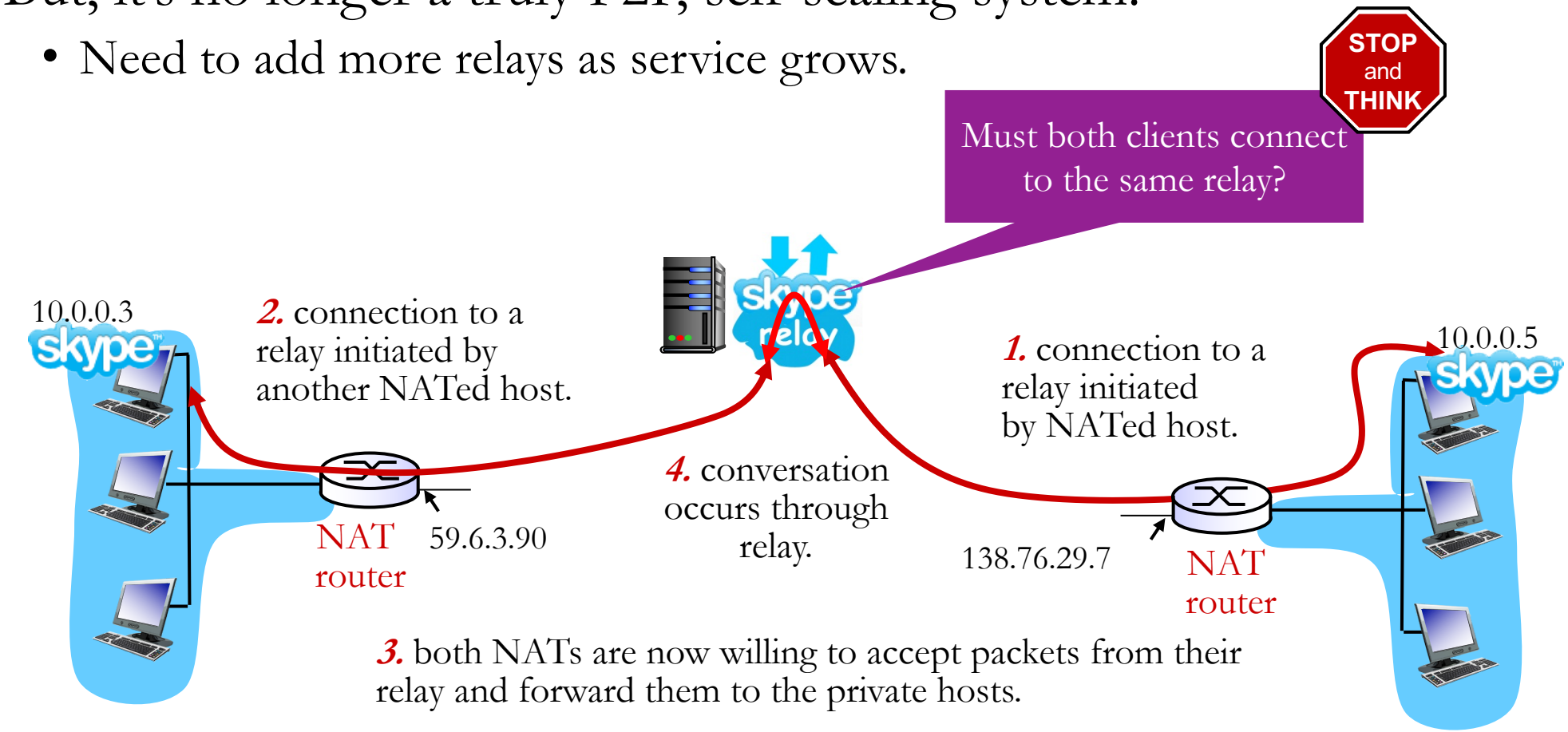
Peer-to-peer communication behind a NAT

- Neither of the two peers can be reached directly.
  - NAT only works if private address contacts a public address.
  - Private IP address cannot listen for new connections from Internet.
  - NAT will discard any inbound packets not associated with an already-established connection.



Peer-to-peer NAT solution:

- Both parties must communicate through a **relay** server.
  - Relay does no processing, so it can handle lots of traffic
  - But, it's no longer a truly P2P, self-scaling system.
    - Need to add more relays as service grows.

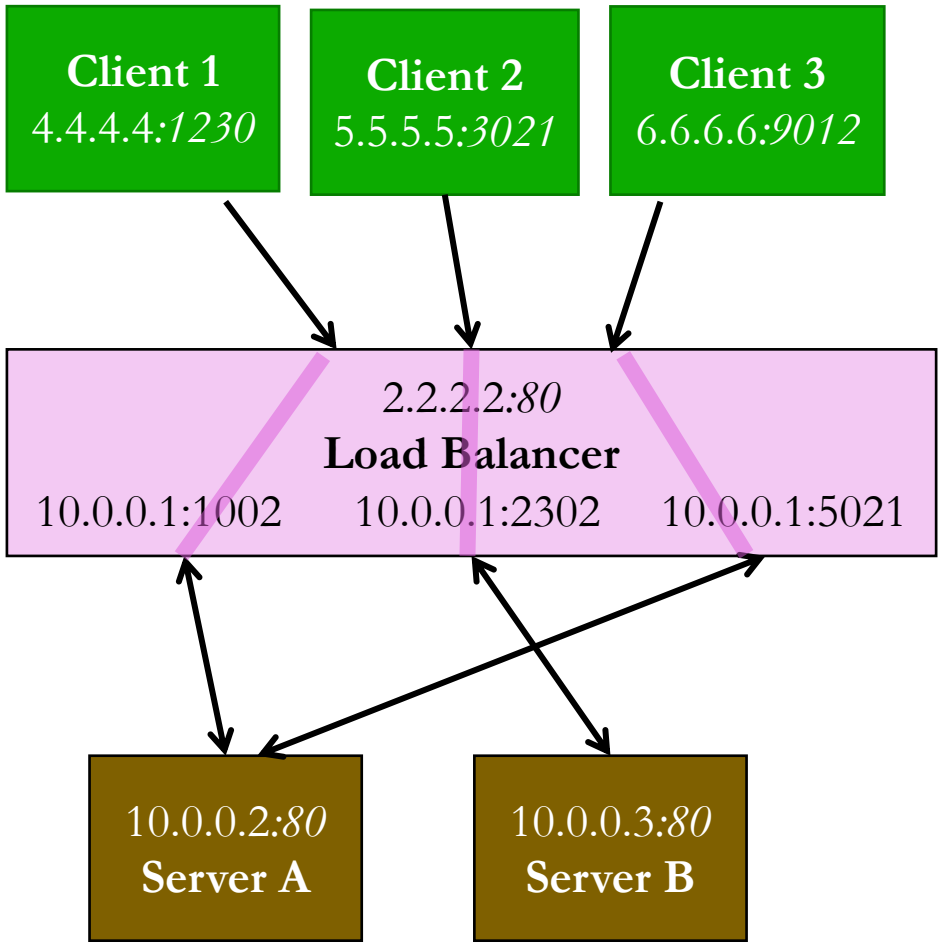


Other benefits to NAT *(besides sharing scarce addresses)*

- **Security**: local devices are not publicly reachable by "strangers."
  - For an outside IP address to contact a local device, the local device must first send a packet to that outside address, causing the NAT to create a new public port mapping for the connection.
- **Configuration isolation**: ISP and public IP address can change without reconfiguring local devices.
- **Load balancing***: NAT can be used to make several servers work in parallel to handle work destined for one IP address.
  - However, LB NAT works slightly differently that home NAT:
  - Create entries in the translation table in response to new **inbound** requests from the public.

NAT Load Balancer *(for scaling and fault tolerance)*

- A type of NAT device that proxies requests to multiple equivalent servers.
  - Load balancer maintains IP address and port mappings, like a traditional NAT.
- Makes multiple servers (with private IP addresses) appear like one big machine with one IP address.
- Allows a single IP address to handle lots of requests: port mapping and relaying is easy, whereas responding to requests may require lots of computation or I/O.
- Load balancer may monitor *health* and *load* of servers to inform its choice of server.
- Called a "layer 4" (TCP/UDP) load balancer.



Multiple kinds of load balancers

- **NAT LB** forwards individual packets and just maps ports.
- **HTTP Reverse Proxy** relays full HTTP requests.
- **DNS LB** directs users to different IP addresses.

Content Delivery Networks (CDNs) combine these two.

- **IP Anycast** assigns one IP address to multiple machines around the world, and the closest one accepts the traffic (covered in next chapter).

- Take CS-310 Scalable Software Architectures for more details on LBs.

Middleboxes

Middleboxes are the category of network devices that transform, filter, or inspect packets but are not routers (not just forwarding). Eg.:

- Network Address Translators (**NATs**)
- **Load Balancers**
- **Firewalls** drop traffic using simple rules:
  - Certain ports or source addresses may be blocked. Eg., censor a domain.
- **Deep Packet Inspection** firewalls looks for app-specific behavior.
  - For example, access Wordpress admin page with HTTP GET /wp-admin
  - Censor certain search terms on a permitted website.
- **Intrusion Detection Systems** (like firewalls, but more long-term)
  - Gather traffic information for offline analysis to detect multi-step attacks.

Intermission

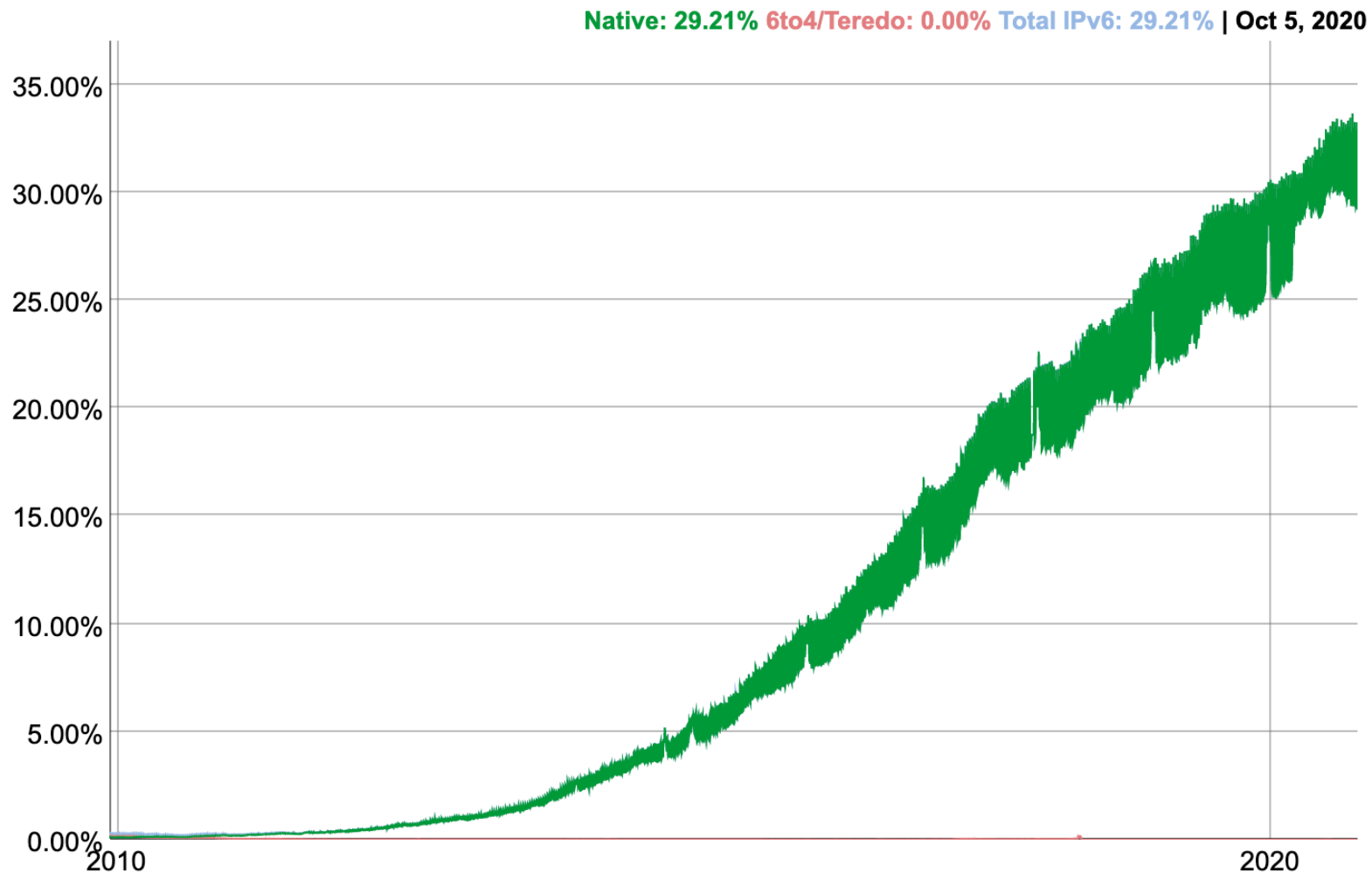
IPv6

- A much better solution than NAT for the IPv4 address shortage.
- Uses 128-bit addresses instead of 32-bit.
  - Expressed in hex:
    `a39b:239e:ffff:29a2:0021:20f1:aaa2:2112`
- Invented in early 1990s when IPv4 address shortage was foreseen.
- 27 years later, IPv6 is slowly being adopted.
- But IPv4 is still the standard, with NAT being used extensively.

Adoption of IPv6: https://www.google.com/intl/en/ipv6/statistics.html

IPv6 Adoption

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.



IPv6 address notation rules

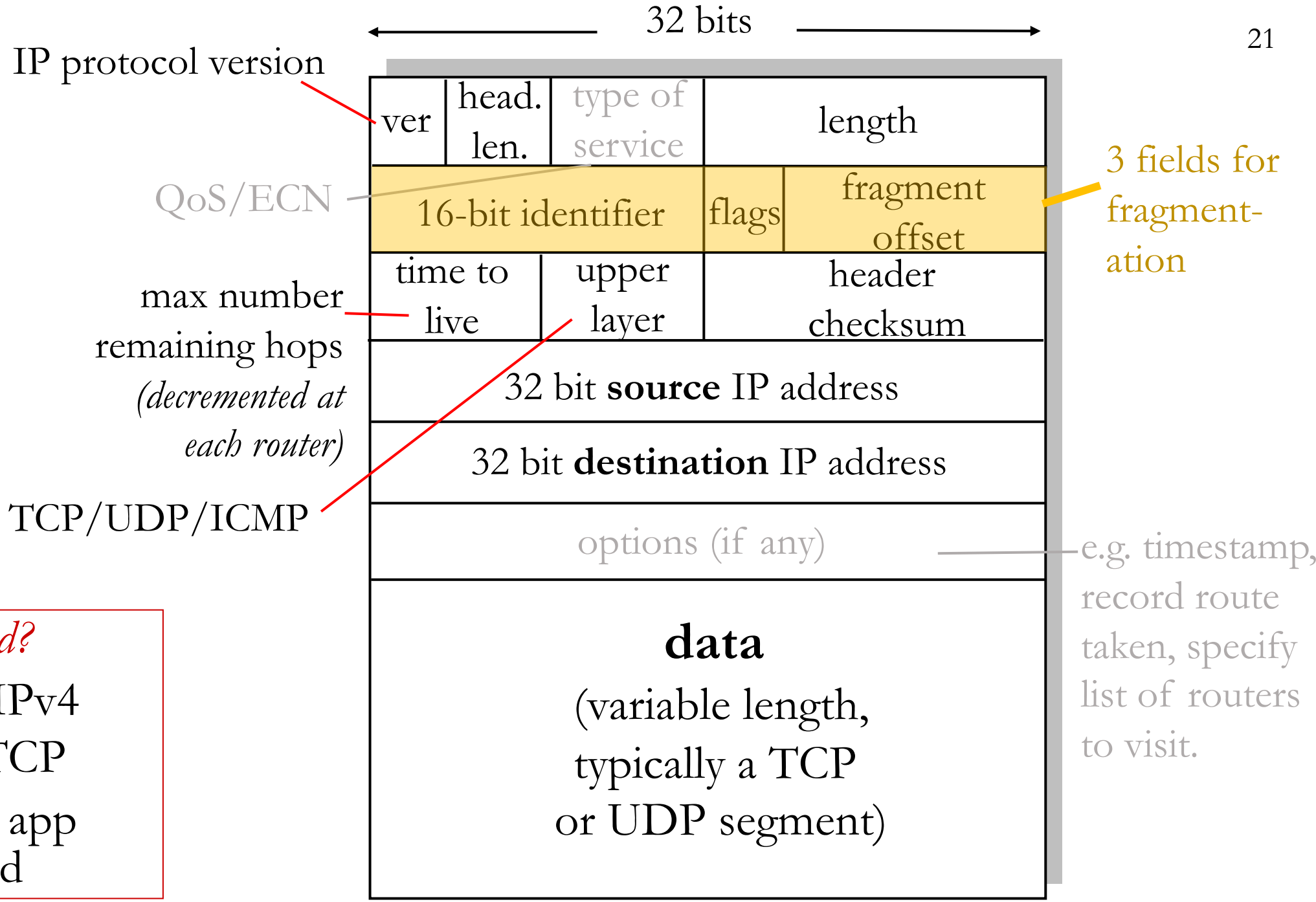
- Groups of zeros can be replaced with "::"
  - Can only use "::" once
- Leading zeros within each 16-bit group can be omitted.

`0000:0000:0000:0000:0000:0000:0000:0001` → `::1` (localhost)

`2345:1001:0023:1003:0000:0000:0000:0000` → `2345:1001:23:1003::`

`aecb:0222:0000:0000:0000:0000:0000:0010` → `aecb:222::10`

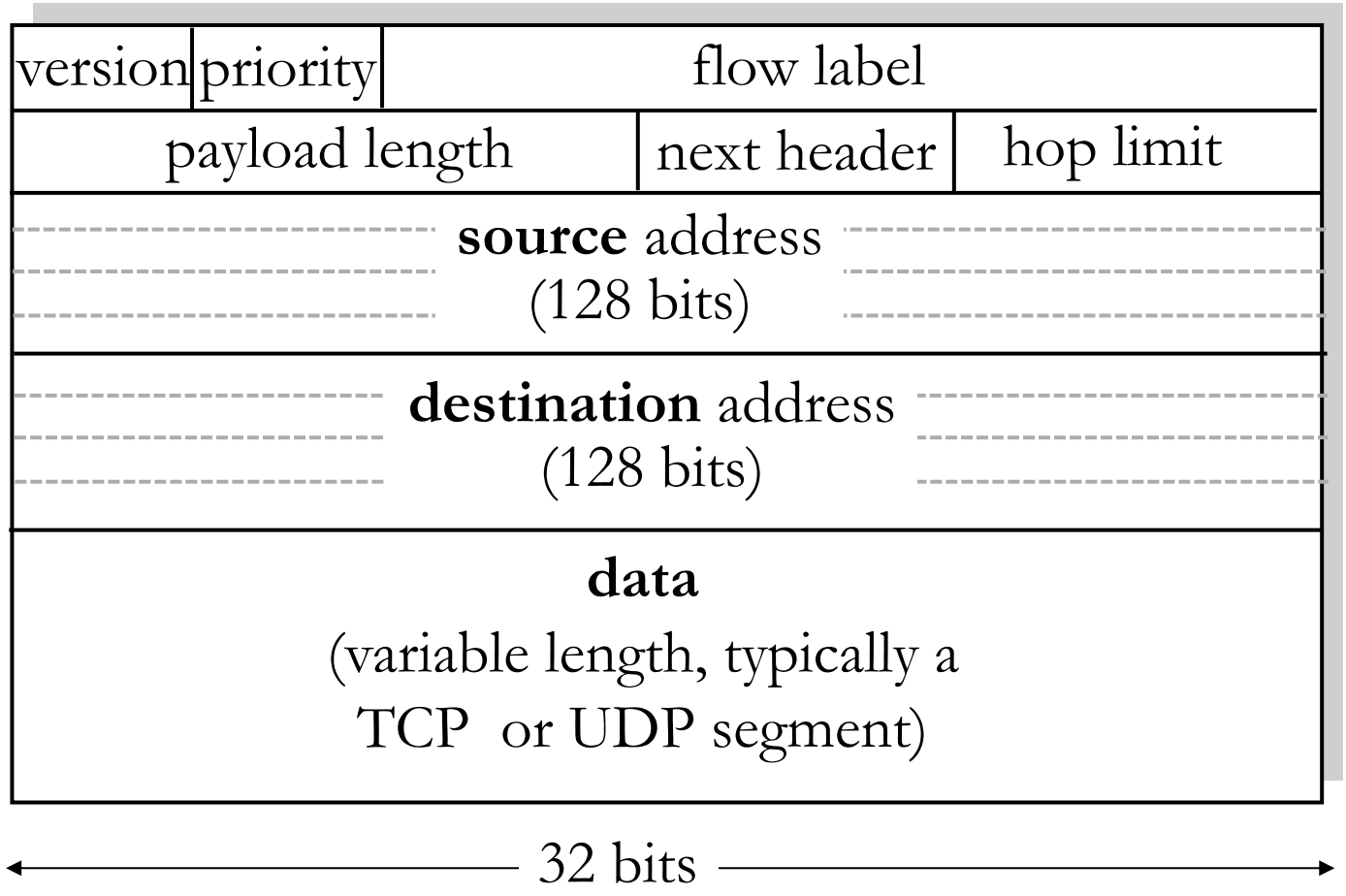
IPv4 datagram



how much overhead?

- **20 bytes** of IPv4
- 20 bytes of TCP
- = 40 bytes + app layer overhead

IPv6 datagram format



- **Priority**: like "type of service" in IPv4.
- **Flow label**: ambiguous
- **Next header**: TCP, UDP
- **Hop limit** = TTL

how much overhead?

- **40 bytes** of IPv6
- 20 bytes of TCP
- = 60 bytes + app layer overhead

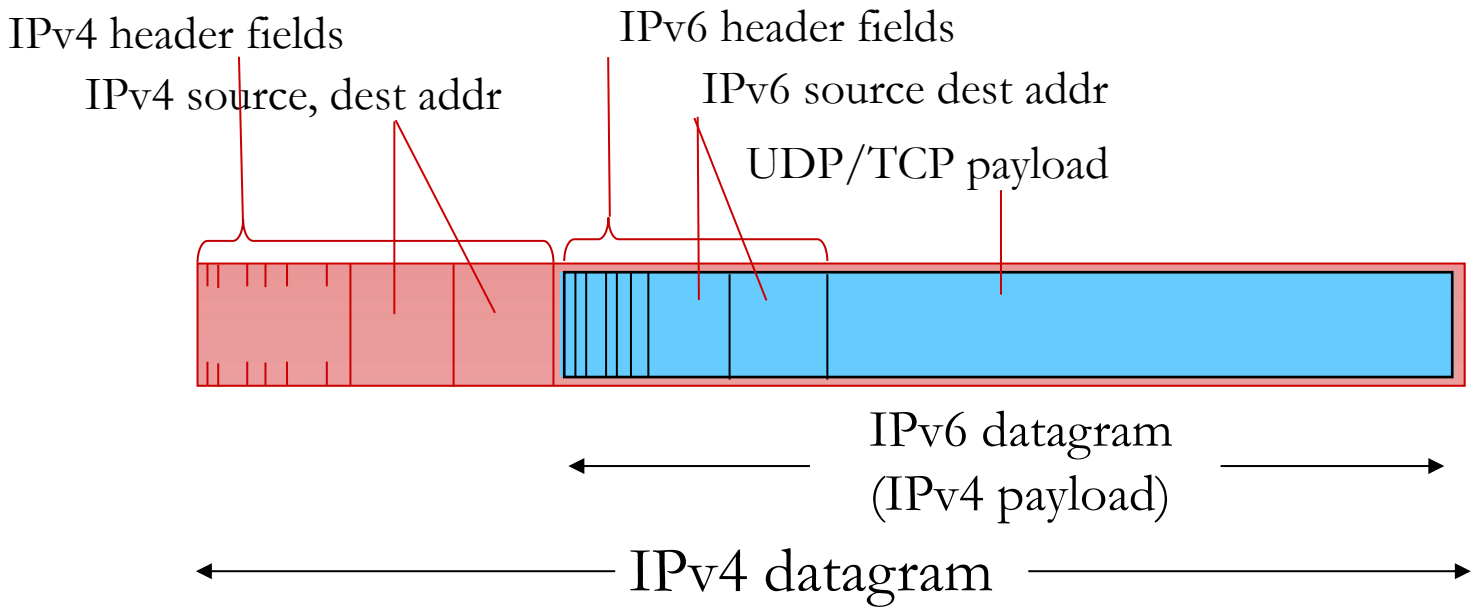
Other improvements in IPv6

- Checksum field was dropped
  - Checksum already exists above in TCP/UDP & below in Ethernet.
  - TTL is decremented at each router, so each router must recalculate checksum!
- Fragmentation support was dropped
  - Routers should be simple and fast, let endpoints deal w/fragmentation (TCP)

✗ But extra 20 bytes header overhead of IPv6 is a tough pill to swallow.
No real incentive to adopt IPv6 if you already have enough addresses.

IPv4 and IPv6 interoperability

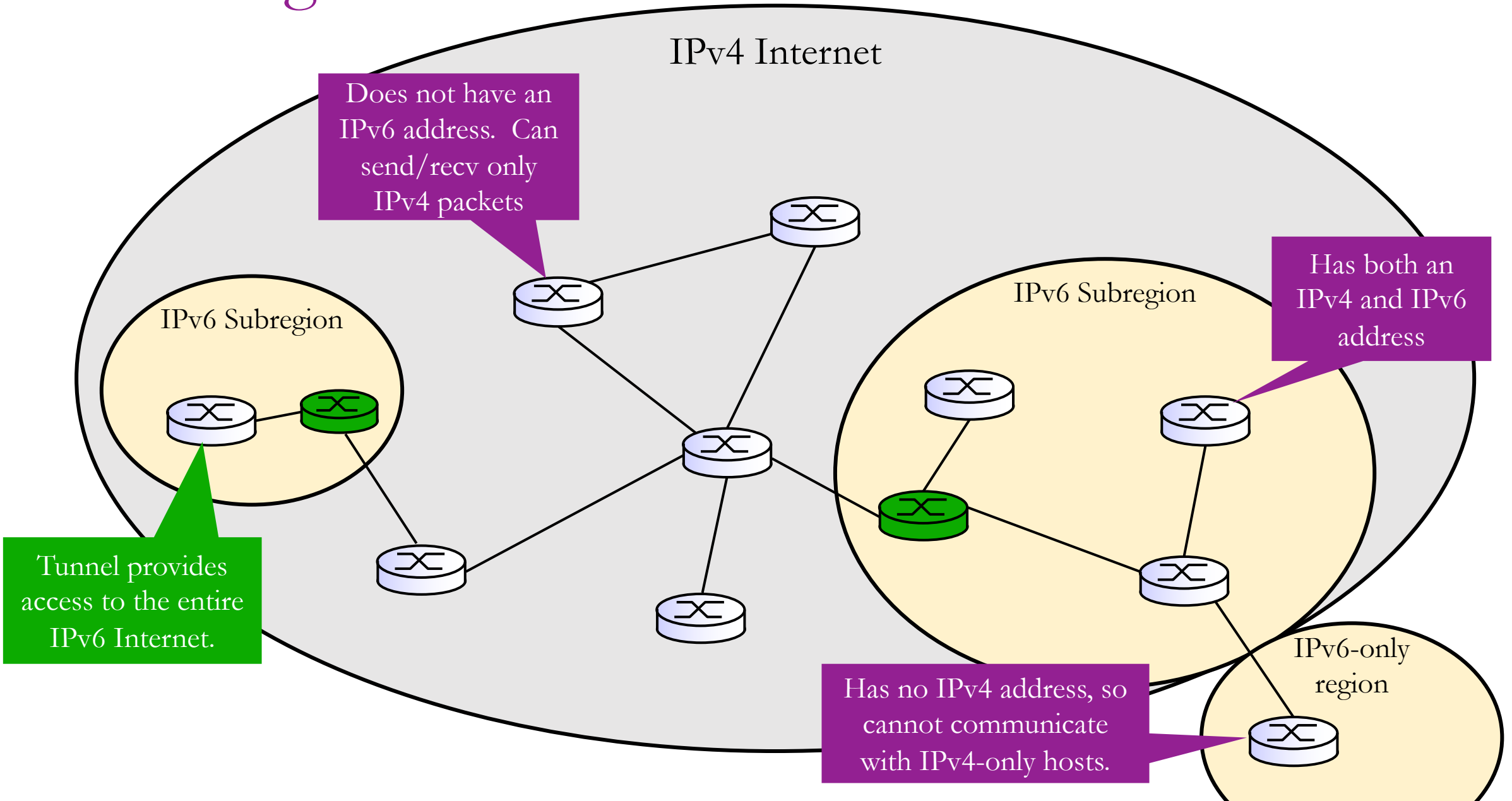
- Internet is too large, important, and disorganized to force everyone to upgrade simultaneously to IPv6.
- Current Internet is a mix of IPv4 and IPv6 routers.
- ***Tunneling*** puts one protocol inside the payload of another:



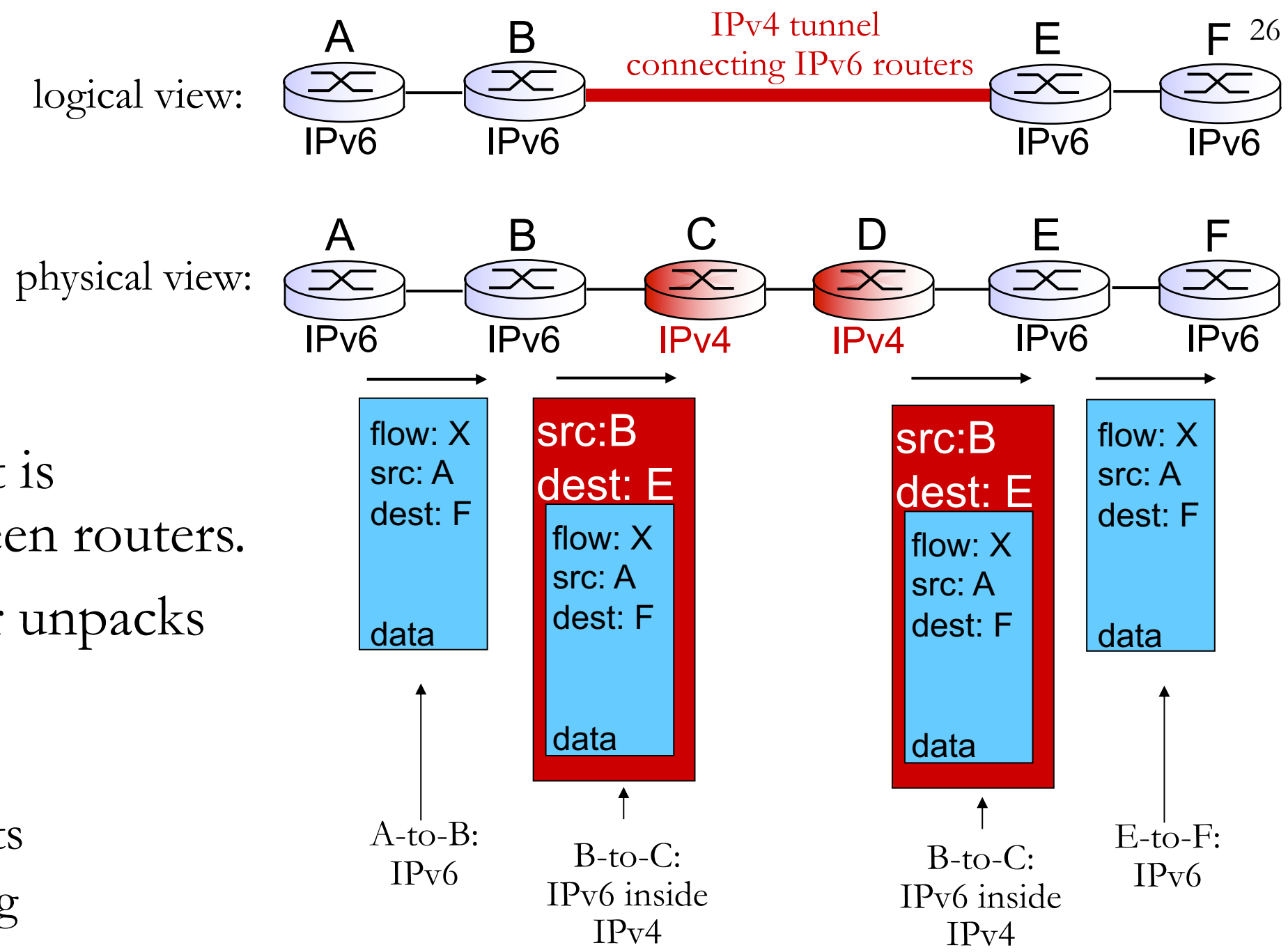
- Either IPv4 inside IPv6 *or* IPv6 inside IPv4

Tunneling illustration

Green routers are tunnel endpoints



Tunneling example



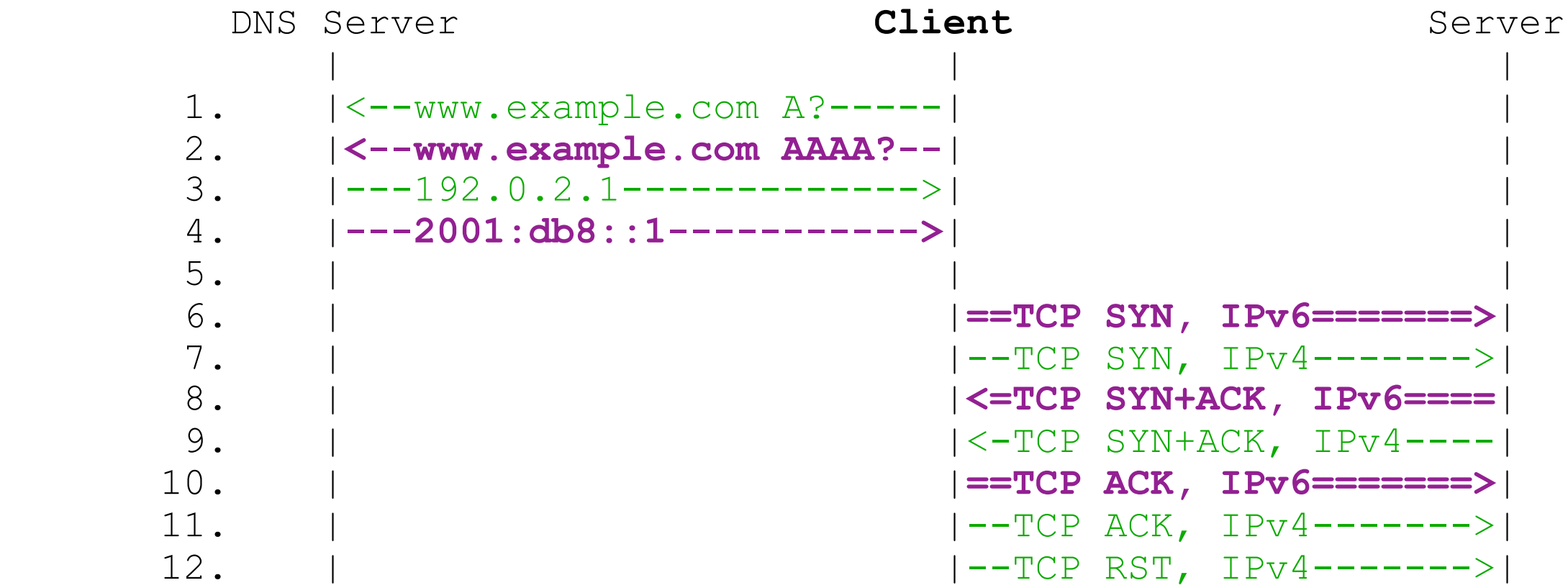
- Tunneled packet is addressed between routers.
- Receiving router unpacks the IPv6 packet.
- Adds overhead:
  - extra header bits
  - extra processing

How do IPv6 hosts talk to IPv4 hosts?

- IPv6 is ***not*** backward compatible with IPv4.
- The "true IPv6" Internet is physically a bunch of "islands"
- *Tunneling* joins those islands with IPv4
- **Dual stack** hosts are configured for both IPv4 and IPv6.
  - These hosts have **two IP addresses**.
  - DHCP request gives IPv4 address, and **DHCPv6** request gives IPv6 address.
  - IPv6 access network (ISP) must be a dual-stack network to give customers full access to the Internet.
- **DNS AAAA** (quad A) records list IPv6 addresses for hosts.
  - When connecting to www.google.com, a dual-stack client will make both DNS A and AAAA requests, to find both an IPv4 and IPv6 addresses (if any).
  - If there is no AAAA record, then fall back to IPv4.

RFC 6555: Happy Eyeballs Dual Stack

- Tunneling can make IPv6 slower than IPv4, so some clients will simultaneously try both IPv4 and IPv6 and use whichever connection completes the TCP handshake first:



- Packet 12 cancels the IPv4 connection because IPv6 ACK'ed first.

IPv6/IPv4 interoperability summary

- **IPv6 → IPv6:** *normal operation*
- **IPv4 → IPv4:** *normal operation*

- **IPv6 → (through IPv4 network) → IPv6:** *tunneling*
- **IPv4 → (through IPv6 network) → IPv4:** *tunneling*

- **IPv4 → IPv6:** *Not directly possible! Use dual-stack config with IPv6*
- **IPv6 → IPv4:** *Not directly possible! Use dual-stack config with IPv4*

- *Further reference:* https://pdfs.semanticscholar.org/34bb/2f946f83b656c6989d42fe043bf5f259b514.pdf

How to find an IPv6 tunnel endpoint?

- Large organizations operating multiple IPv6 islands will set up their own tunnels.
- Organizations may also "peer" with each other by configuring tunnels.
- Smaller organization may use a **tunnel broker** to access the IPv6 Internet:
  - Hurricane Electric (a Tier 1 ISP) operates a free tunnel broker service: https://tunnelbroker.net/
  - This lets you start using IPv6 at home even if your ISP is not IPv6 enabled!
  - But if your IPv4 address changes, then the tunnel must be reconfigured.

Recap

- **Private networks** are isolated from the **public** Internet, but usually connected through a Network Address Translator (**NAT**).
  - **Port mapping** makes multiple machines on the private subnet look like multiple sockets (processes) on one big machine.
  - NAT requires no awareness or cooperation from hosts on either side.
  - NAT is also one way to implement a load balancer.
  - Besides NATs, **middleboxes** include *firewalls* and other security appliances.
- **IPv6** uses 128-bit addresses for practically unlimited public addresses.
  - IPv6 adds 20 bytes of header overhead.
  - Not directly compatible with IPv4. Adopted by ~30% of end hosts.
  - **Dual-stack** hosts have both IPv4 and IPv6 addresses to reach entire Internet.
  - Interoperates with IPv4 via **tunneling:** send IPv6 packet inside IPv4 packet.