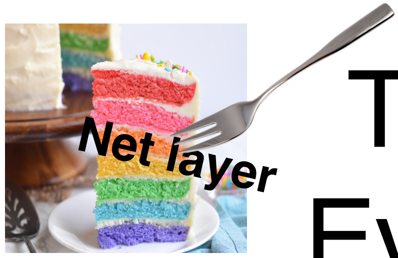
Network Layer: Protocols (Part 3)

Lecture 22

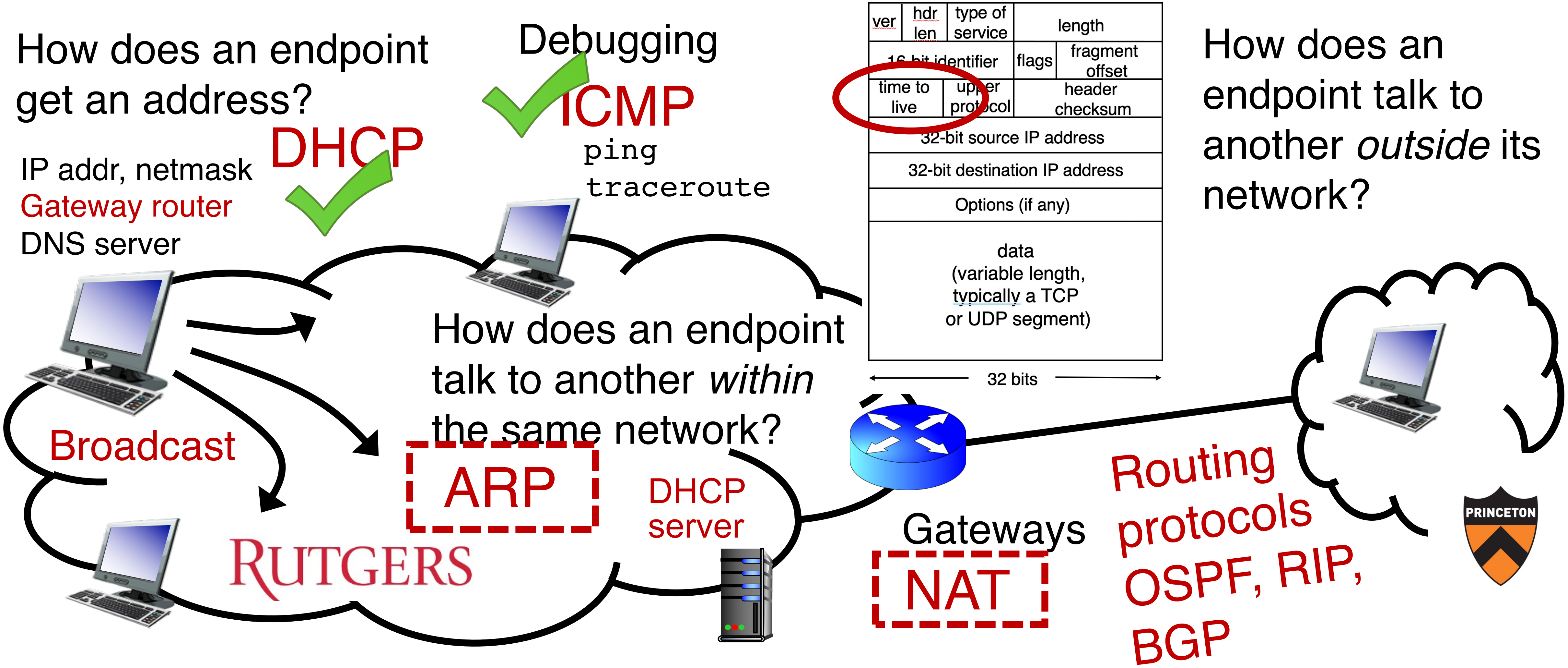
http://www.cs.rutgers.edu/~sn624/352-S22

Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK



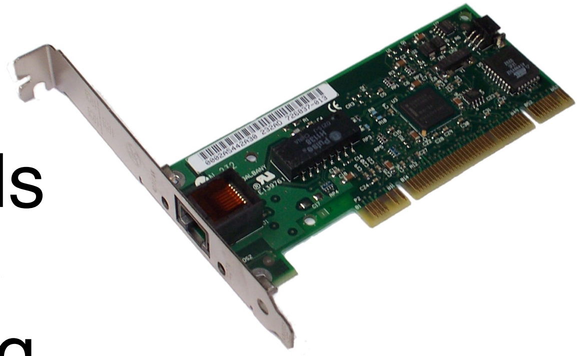
The network layer is all about reachability. Every protocol below solves a sub-problem.



Address Resolution Protocol

Background: Let’s peek into the link layer

- Each network adapter has a hardware address or a MAC address
  - E.g., the Wi-Fi adapter on your laptop has one
- Assigned by the manufacturer, not expected to vary over time
  - Think about it as an identifier for the device
- To communicate over a single link, a sender needs the destination hardware address
- Directory mechanisms like DNS and bootstrapping mechanisms like DHCP provide IP addresses
- Given an IP address, how does an endpoint find the hardware address?



Address Resolution Protocol (ARP)

- ARP solves the following problem. Given an IP, find the machine's hardware address
  - IP → MAC resolution
- All endpoints that are looked up are expected to be within the same network
- Hence, address resolution can use broadcast:
  - We don't need to develop directory mechanisms like DNS
  - Send (ARP) queries to everyone, asking for a MAC given an IP

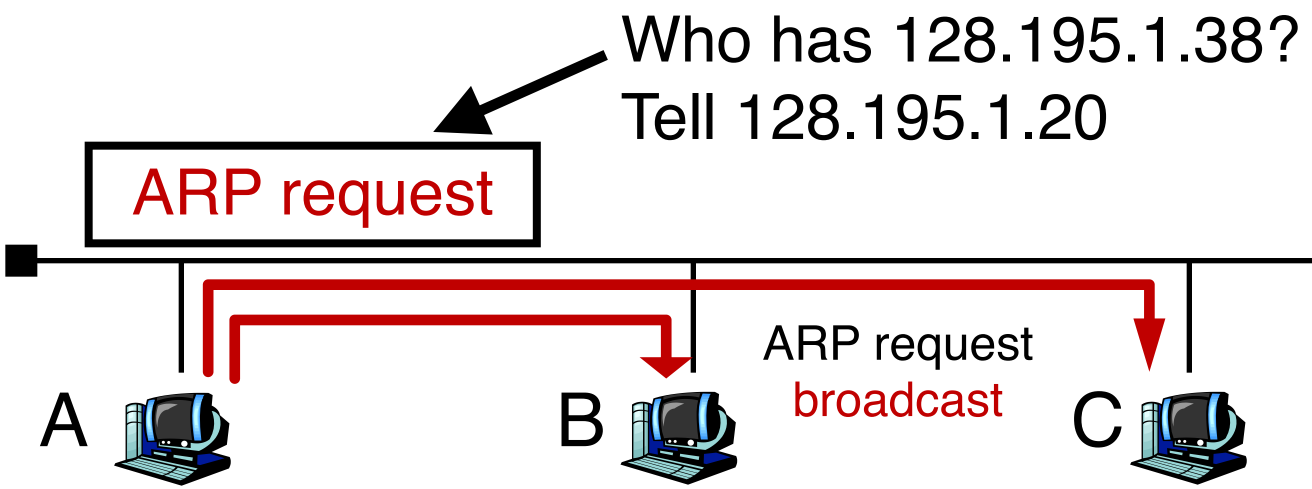
ARP packet format

- Hardware type: link-layer protocol
  - Example: Ethernet (1)
- Hardware address length:
  - Example: Ethernet = 6 bytes
- Protocol Type: network-layer protocol
  - Example: IPv4 (0x0800)
- Protocol address length
  - Example: IPv4 = 4 bytes
- Operation:
  - ARP request: 1, reply: 2
- Sender's addresses
- Address to be resolved (or response)

Internet Protocol (IPv4) over Ethernet ARP packet

Octet offset	0	1
0	Hardware type (HTYPE)	
2	Protocol type (PTYPE)	
4	Hardware address length (HLEN)	Protocol address length (PLEN)
6	Operation (OPER)	
8	Sender hardware address (SHA) (first 2 bytes)	
10	(next 2 bytes)	
12	(last 2 bytes)	
14	Sender protocol address (SPA) (first 2 bytes)	
16	(last 2 bytes)	
18	Target hardware address (THA) (first 2 bytes)	
20	(next 2 bytes)	
22	(last 2 bytes)	
24	Target protocol address (TPA) (first 2 bytes)	
26	(last 2 bytes)	

ARP operation



Ethernet Address:
05:23:f4:3d:e1:04
IP Address:
128.195.1.20

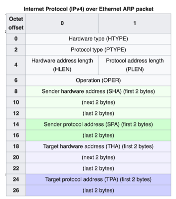
Ethernet Address:
12:04:2c:6e:11:9c
IP Address:
128.195.1.122

Ethernet Address:
98:22:ee:f1:90:1a
IP Address:
128.195.1.38

Wants to transmit
to 128.195.1.38

Different target
IP address:
ignore ARP

Matching target
IP: send reply



Hardware type: Ethernet
Protocol type: IPv4
Hardware addr length: 6
Protocol addr length: 4
Operation: 2 (reply)
Sender hardware addr:
05:23:f4:3d:e1:04
Sender protocol addr:
128.195.1.20
Target HW addr:
98:22:ee:f1:90:1a
Target protocol addr:
128.195.1.38

Communicating outside the local net?

- Suppose endpoint A wants to communicate with endpoint B that is in a different network
- ARP broadcast outside the local network is too expensive
  - How does one limit the scope of the broadcast? Internet-wide?
- Besides, the hardware address format used by B's network might be different from that of A's network!
- ARPs are not meaningful across network boundaries
- Communicating to a network-external endpoint just means sending the packet to the gateway router
  - Host can know that a destination is external using IP addr and netmask
  - Host can talk to the gateway using DHCP (to get IP) and ARP (to get MAC)

Summary of ARP

- A useful mechanism to allow hosts inside a network to communicate:
- ARP protocol helps resolve IP addresses into MAC addresses using a broadcast mechanism
- Communication outside the local network requires ARP-ing for and sending packets to the gateway

Network Address Translation (NAT)

Background: The Internet's growing pains

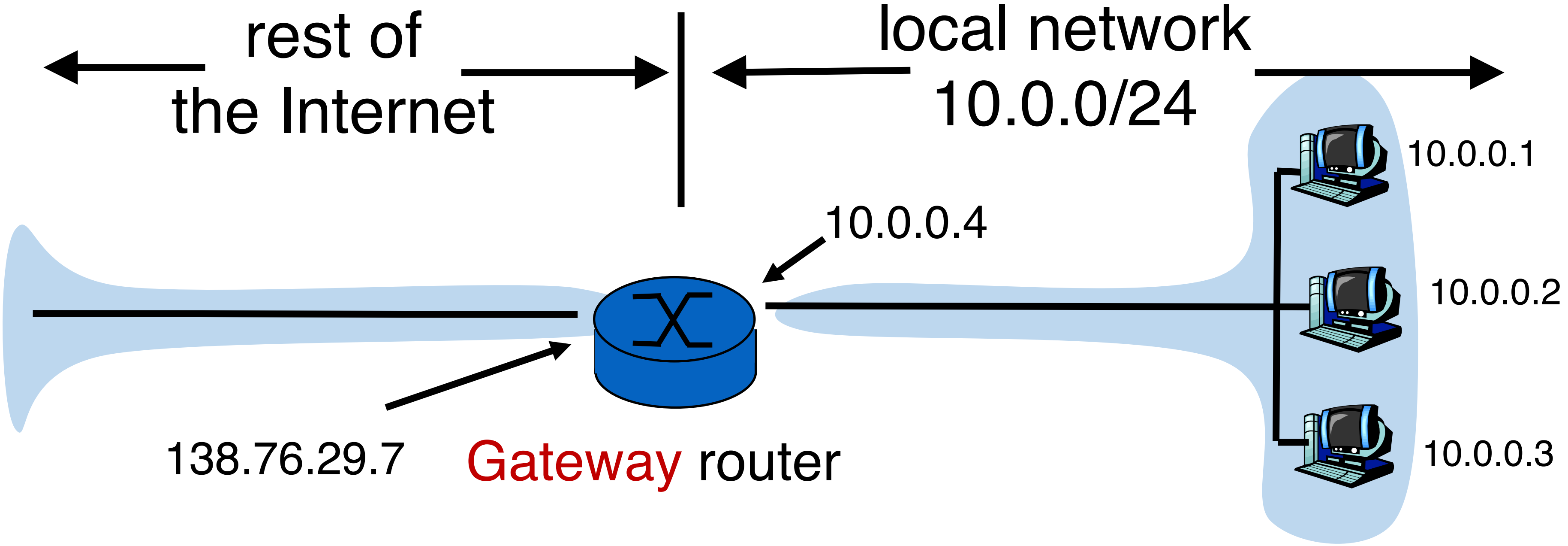
- Networks had incompatible addressing
  - IPv4 versus other network-layer protocols (X.25)
  - Routable address ranges different across networks
- Entire networks were changing their Internet Service Providers
  - ISPs don't want to route directly to internal endpoints, just to the gateway
- IPv4 address exhaustion
  - Insufficient large IP blocks even for large networks
  - Rutgers (AS46) has > 130,000 publicly routable IP addresses
  - IIT Madras (a well-known public university in India, AS141340) has 512

(Source: ipinfo.io)

Network Address Translation

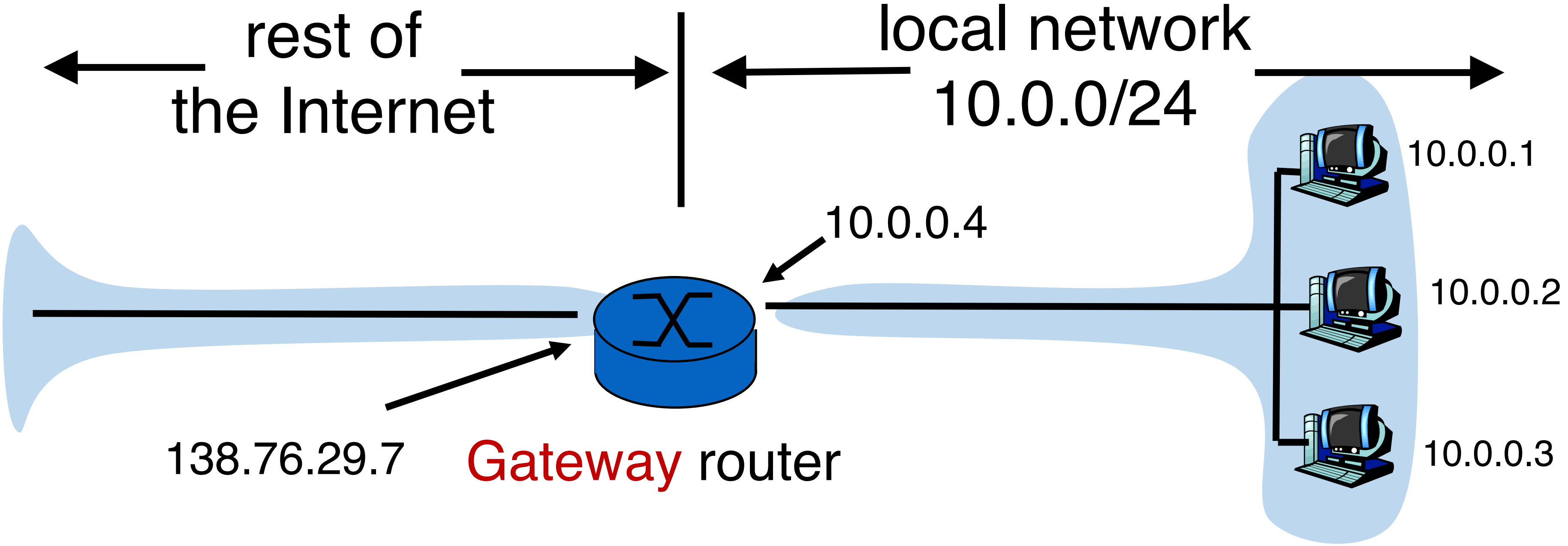
- When a router modifies fields in an IP packet to:
- Enable communication across networks with different (network-layer) addressing formats and address ranges
- Allow a network to change its connectivity to the Internet en masse by modifying the source IP to a (publicly-visible) gateway IP address
- Masquerade as an entire network of endpoints using (say) one publicly visible IP address
  - Effect: use fewer IP addresses for more endpoints!

Typical NAT setup



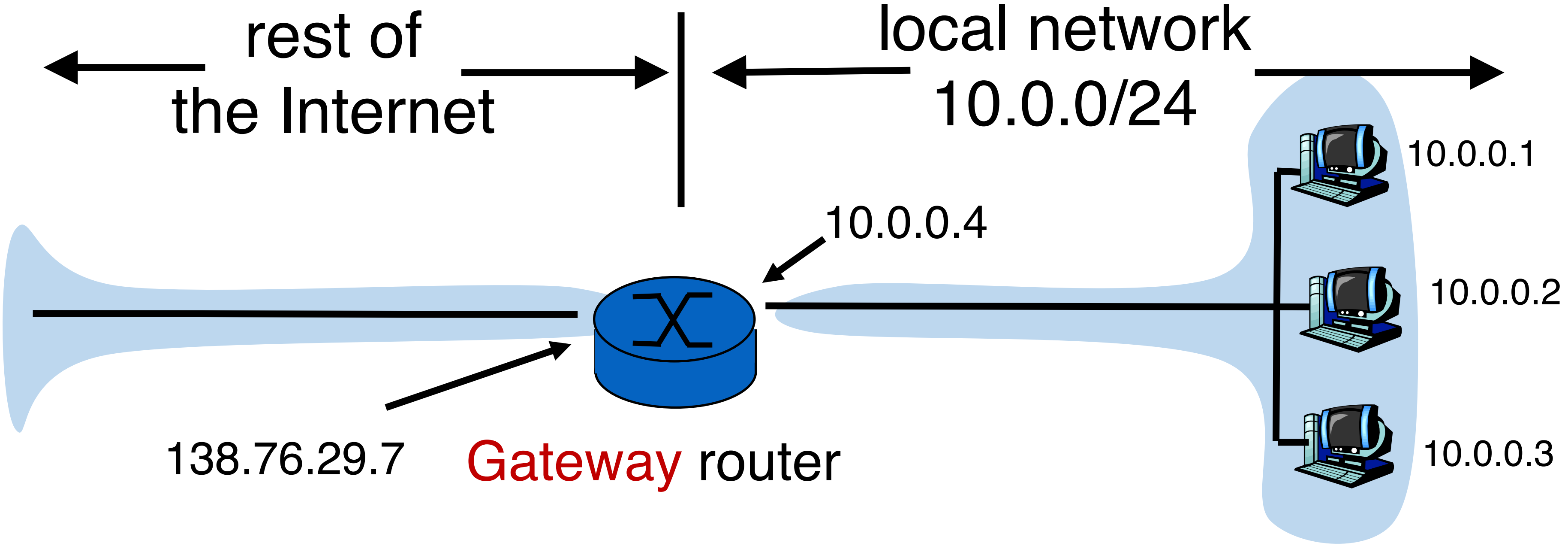
- The gateway's IP, 138.76.29.7 is publicly visible
- The local endpoint IP addresses in 10.0.0/24 are private
- All datagrams leaving local network have the same source IP as the gateway

Typical NAT setup



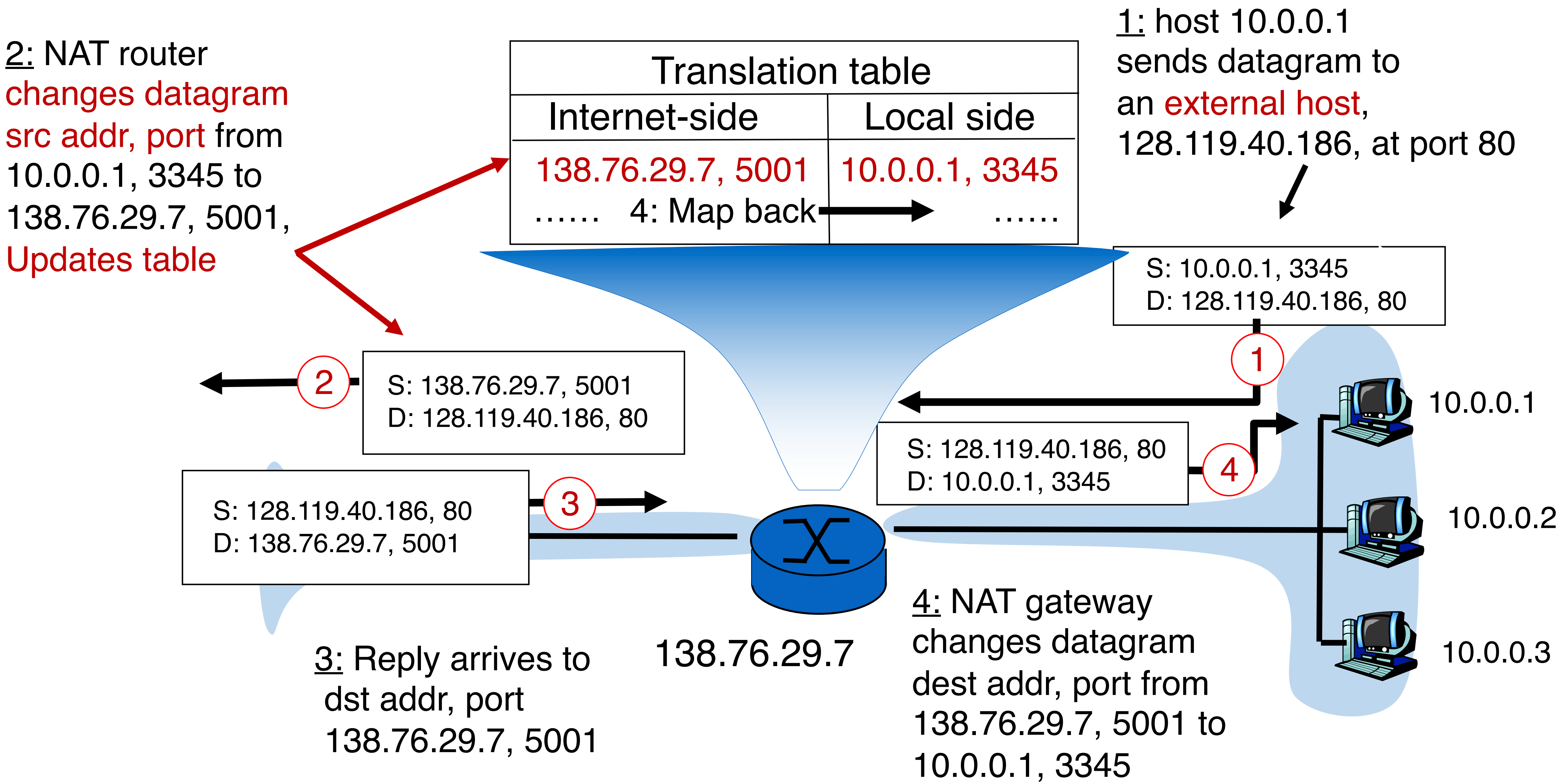
That is, for the rest of the Internet, the gateway masquerades as a single endpoint representing (hiding) all the private endpoints. The entire network just needs one (or a few) public IP addresses.

Typical NAT setup



The NAT gateway router accomplishes this by using a different transport port for each distinct (transport-level) conversation between the local network and the Internet.

Typical NAT setup



Features of IP-masquerading NAT

- Use one or a few public IPs: You don’t need a lot of addresses from your ISP
- Change addresses of devices inside the local network freely, without notifying the rest of the Internet
- Change the public IP address freely independent of network-local endpoints
- Devices inside the local network are not publicly visible, routable, or accessible
- Most IP masquerading NATs block incoming connections originating from the Internet
  - Only way to communicate is if the internal host initiates the conversation

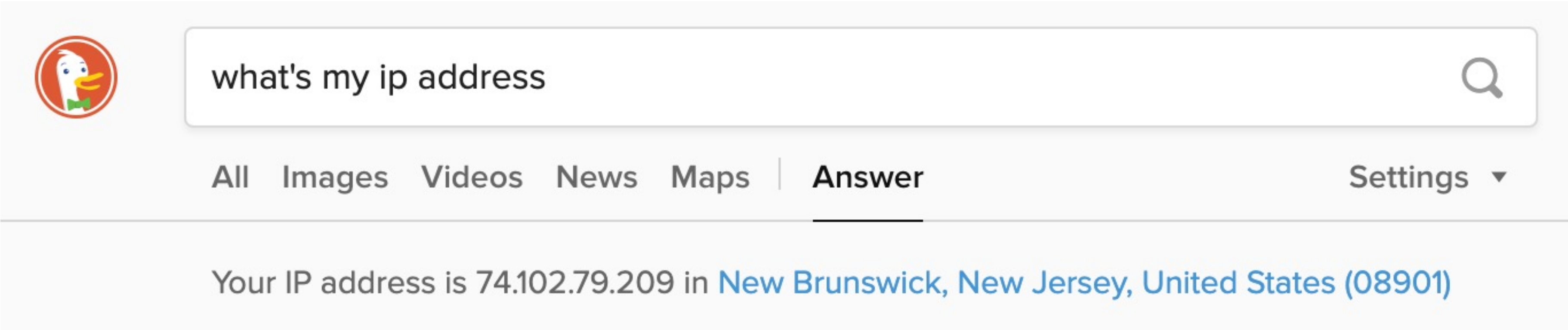
If you’re home, you’re likely behind NAT

- Most access routers (e.g., your home WiFi router) implement network address translation

- You can check this by comparing your local address (visible from `ifconfig`) and your externally-visible IP address (e.g., type “what’s my IP address?” on your browser search bar)

If you're home, you're likely behind NAT

```
[flow:352-S20]$  ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether f0:18:98:1c:fc:36
        inet6 fe80::1036:7dea:82ee:e868%en0 prefixlen 64 secured scopeid 0xa
        inet 192.168.1.151 netmask 0xffffff00 broadcast 192.168.1.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
[flow:352-S20]$
```



Limitations of IP-masquerading NATs

- Connection limit due to 16-bit port-number field
  - ~64K total simultaneous connections with a single public IP address
- NAT can be controversial
  - "Routers should only manipulate headers up to the network layer, not modify headers at the transport layer!"
- Application developers must take NAT into account
  - e.g., peer-to-peer applications like Skype
- Internet "purists": instead, solve address shortage with IPv6
  - 32-bit IP addresses are just not enough
  - Esp. with more devices (your watch, your fridge, …) coming online

Internet Protocol version 6 (IPv6)

IPv6: Main changes from IPv4

- Large address space: 128-bit addresses (16 bytes)
  - Allows up to 3.4 x 10^{38} unique addresses

- Fixed length headers (40 bytes)
  - Improves the speed of packet processing in routers
  - IPv6 options processing happens through a separate mechanism: using the field corresponding to the upper-layer protocol

- New control message protocol: ICMPv6

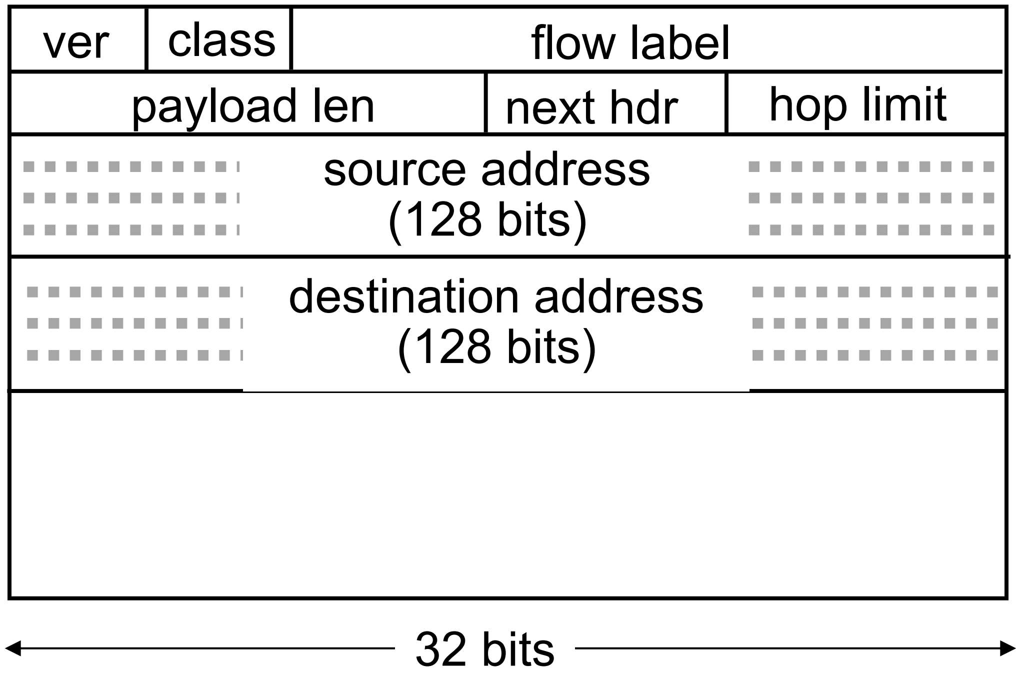
IPv6: Main changes from IPv4

- New quality of service bits: flow label and traffic class
  - Flow label: denotes packets belonging to the same conversation
  - How the field is populated (ie: what exactly belongs to a “flow”) isn’t specified
  - Routers may choose to provide performance guarantees to flows of specific traffic classes (more on this later)

- No IP checksum: remove redundant error detection mechanisms
  - Checksums exist already on common transport (TCP/UDP) and link layer (Ethernet) headers

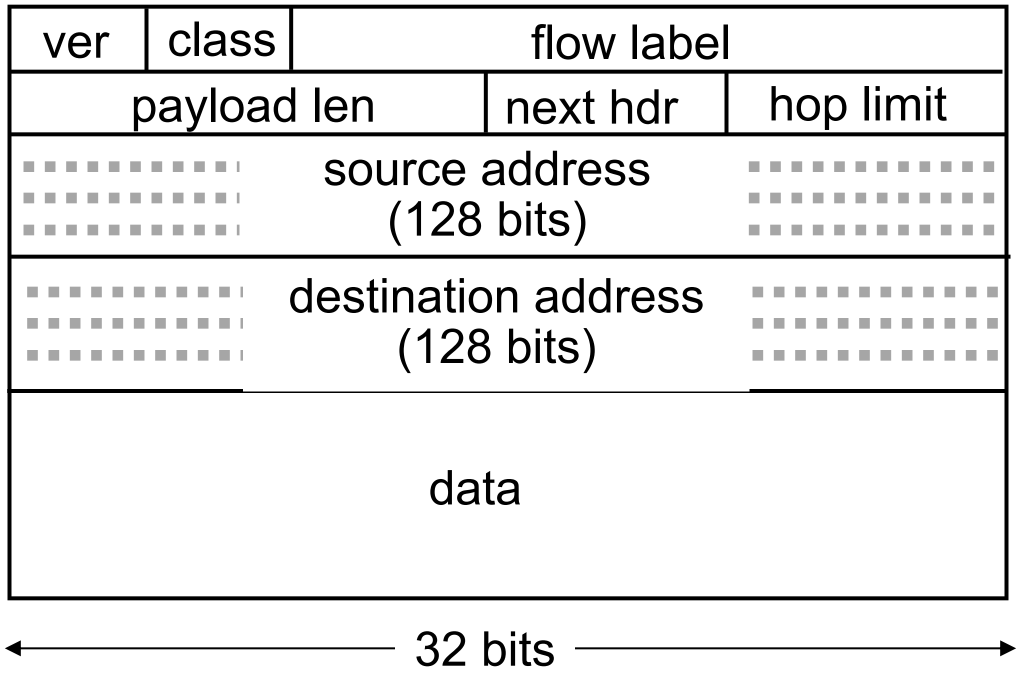
IPv6 datagram format

- Version: 6
- Class and flow label: for traffic differentiation at routers
- Next header: same as the upper-layer protocol in IPv4. Also used to include IPv6 options
- Hop limit: same as TTL in IPv4

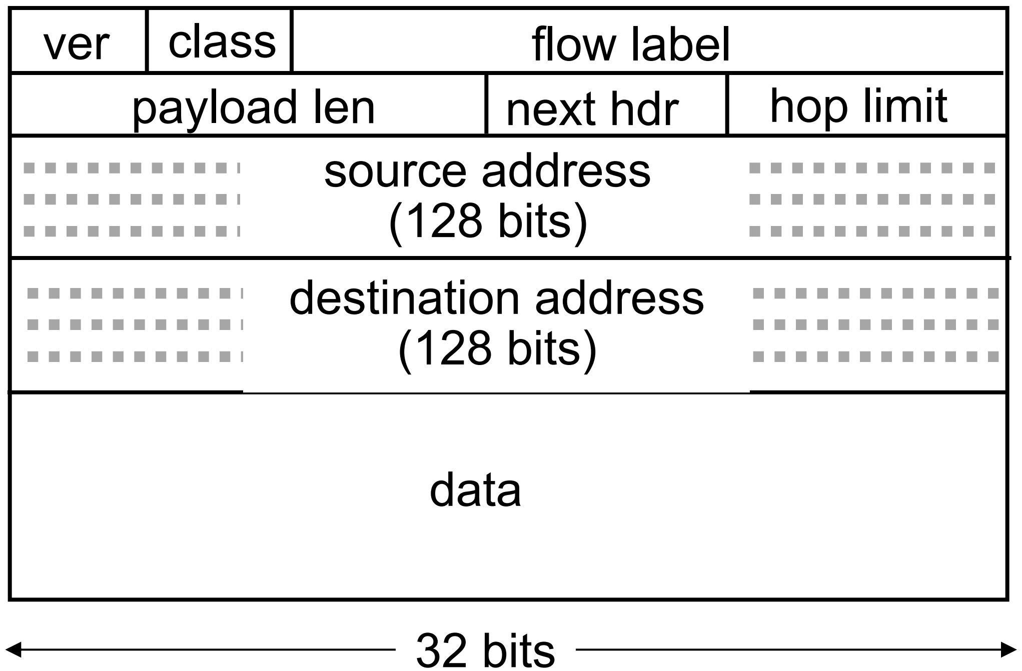


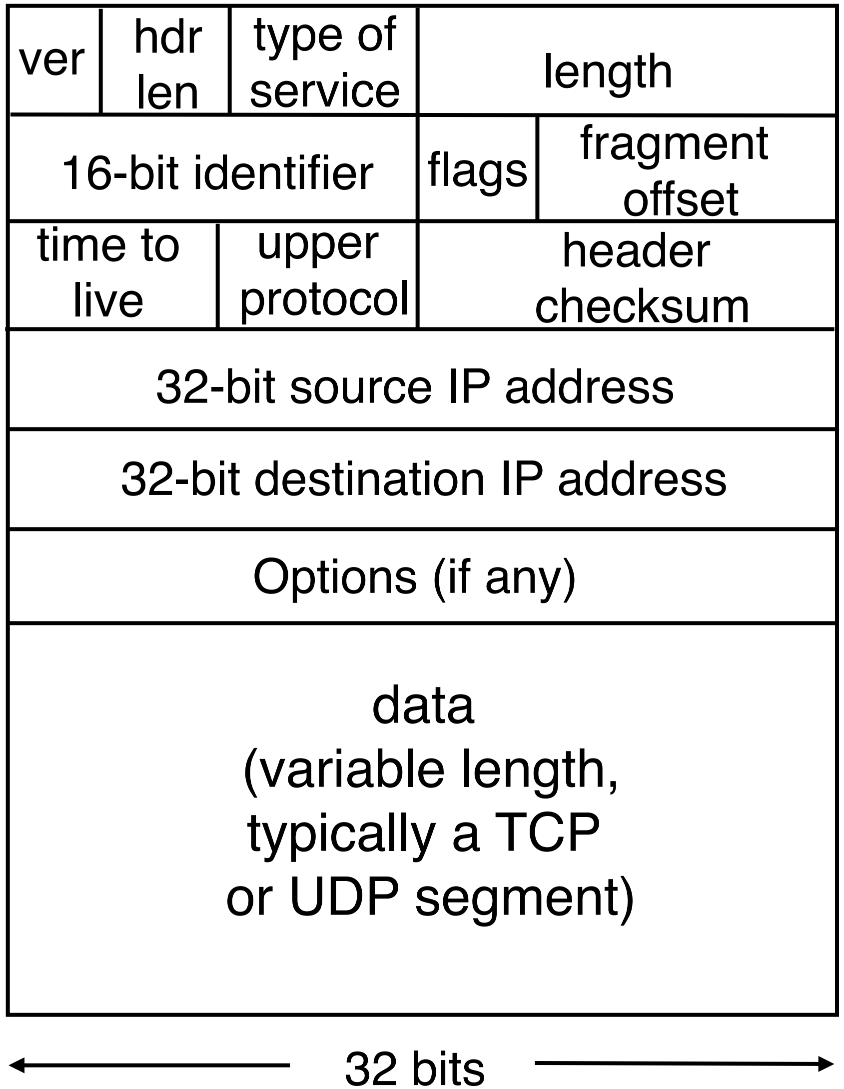
IPv6 datagram format

- Version: 6
- Class and flow label: for traffic differentiation at routers
- Next header: same as the upper-layer protocol in IPv4. Also used to include IPv6 options
- Hop limit: same as TTL in IPv4



Can you spot the differences?





IPv6 addresses

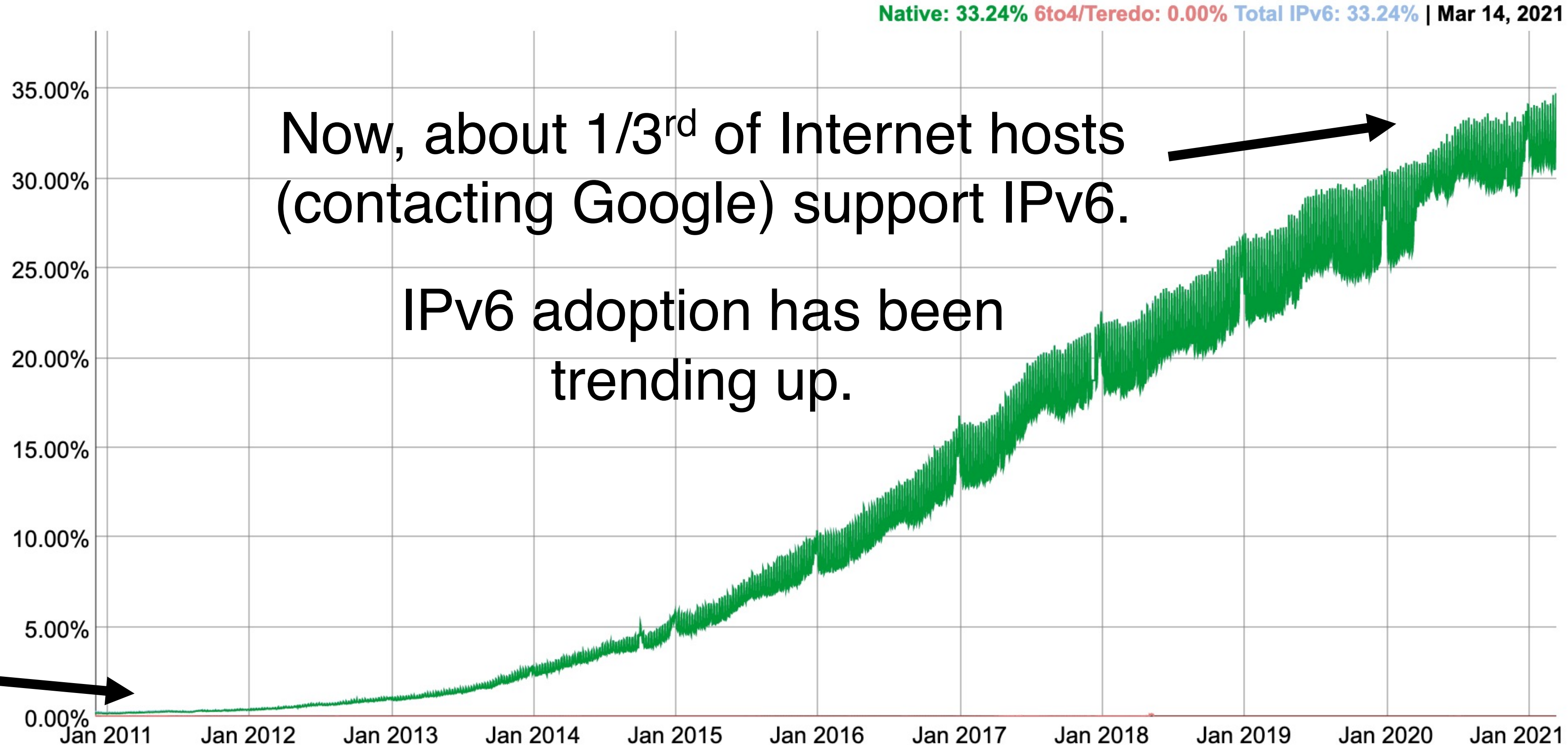
- IPv6 uses IPv4-CIDR-like (classless) addressing
- Notation: xx:xx:xx:xx:xx:xx:xx:xx
  - x = 4-bit hex number
  - Contiguous 0s are compressed: 47CD::A456:0124
- An IPv4-compatible IPv6 address has a prefix of 96 0-bits
  - Example: ::128.64.18.87
- Globally routable unicast addresses must start with bits 001
- CIDR prefixes written the usual way:
  - Example: 2000::/48 can contain 2^{80} endpoints

Source: https://www.google.com/intl/en/ipv6/statistics.html#tab=ipv6-adoption

IPv6 adoption

When IP became a mainstream network-layer protocol, IPv4 was baked into router hardware.

~0% of Internet hosts used IPv6 for a long time (about 30 years)



Now, about 1/3rd of Internet hosts (contacting Google) support IPv6.

IPv6 adoption has been trending up.

In 2012, Google and a bunch of large orgs decided to support IPv6 irrevocably.