IPv4 and IPv6

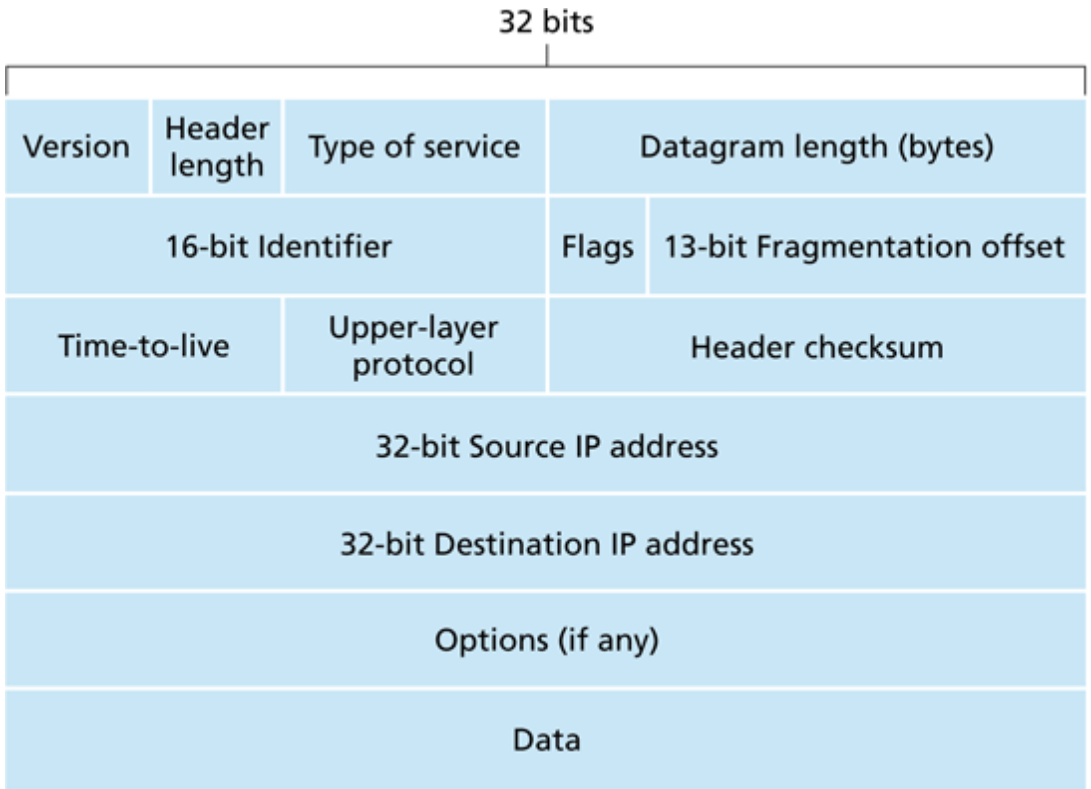
Daniel Zappala

CS 460 Computer Networking
Brigham Young University

Imagine Building IP

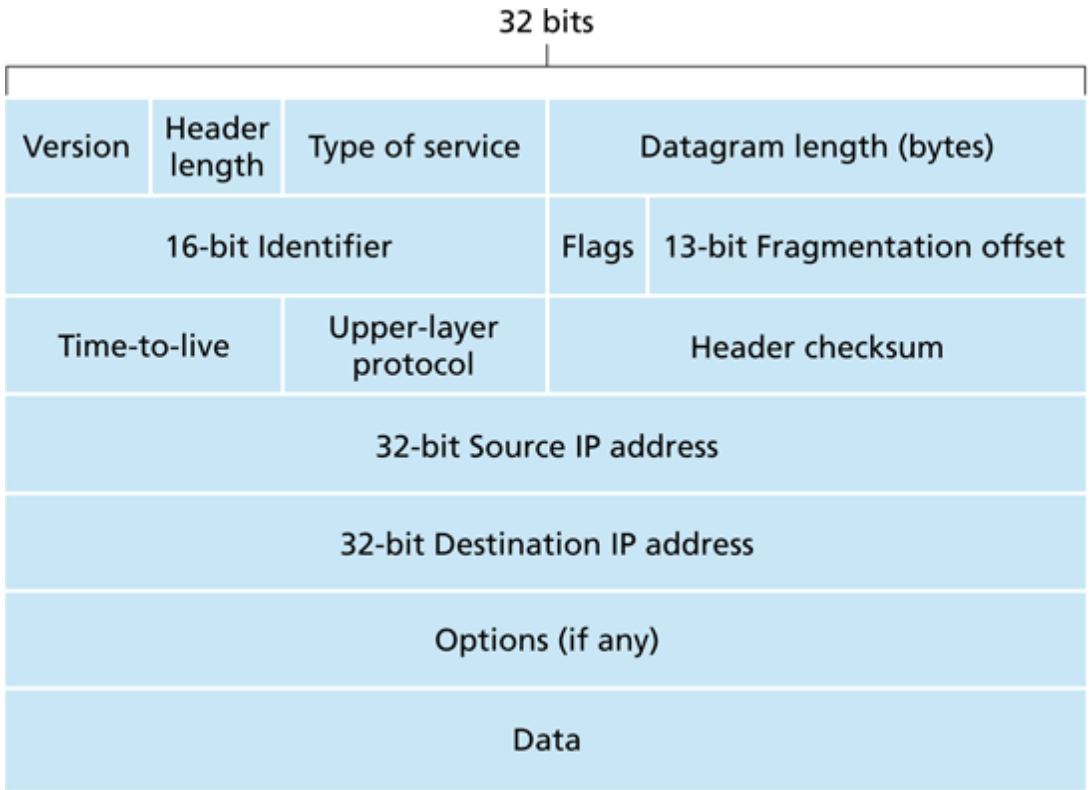
Common protocol for all networks
Must be very simple
Must last 40+ years

IPv4 Header Format



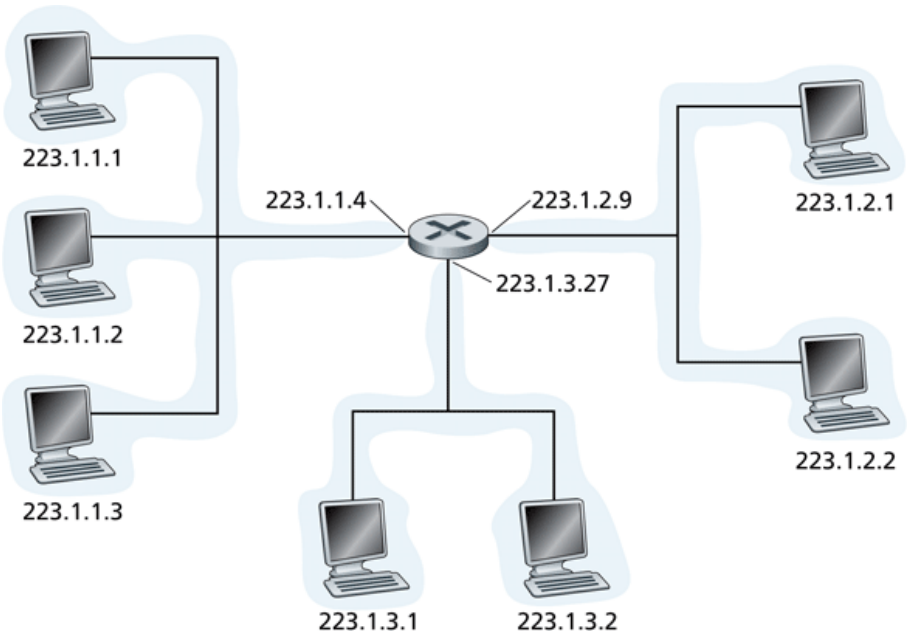
- header length: 20 bytes min.
- ToS: early attempt to route packets along paths with low delay or high bandwidth
- fragmentation: identifier, flags, offset

IPv4 Header Format



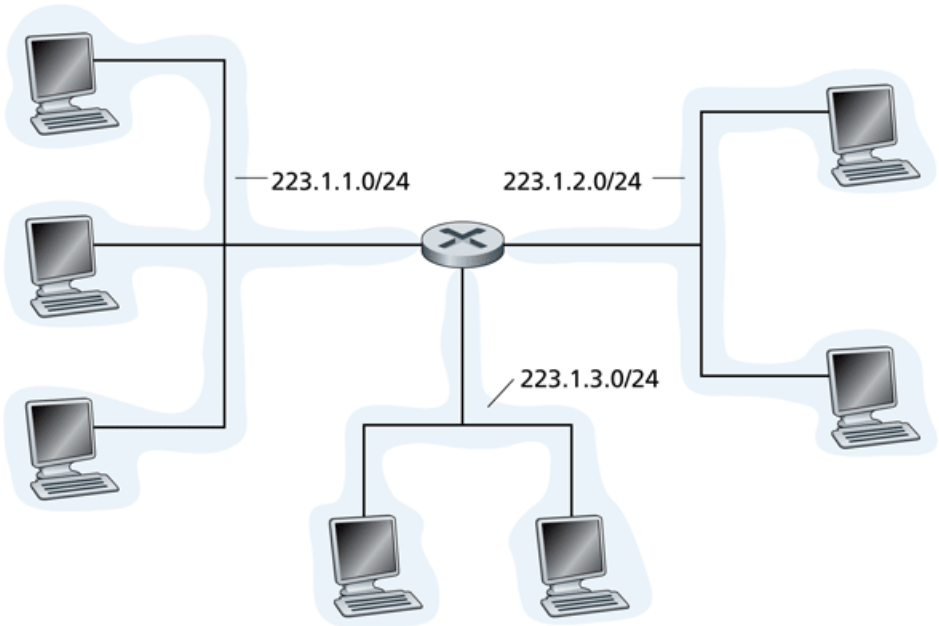
- TTL: used to literally be time (at least one second), now hops
- protocol: deliver to this protocol at destination
- options: includes timestamp, record route, source route

IPv4 Addresses



- 32 bits
- dotted-decimal notation: each part is 8 bits
- identifies an interface/link on a host or router

Subnets



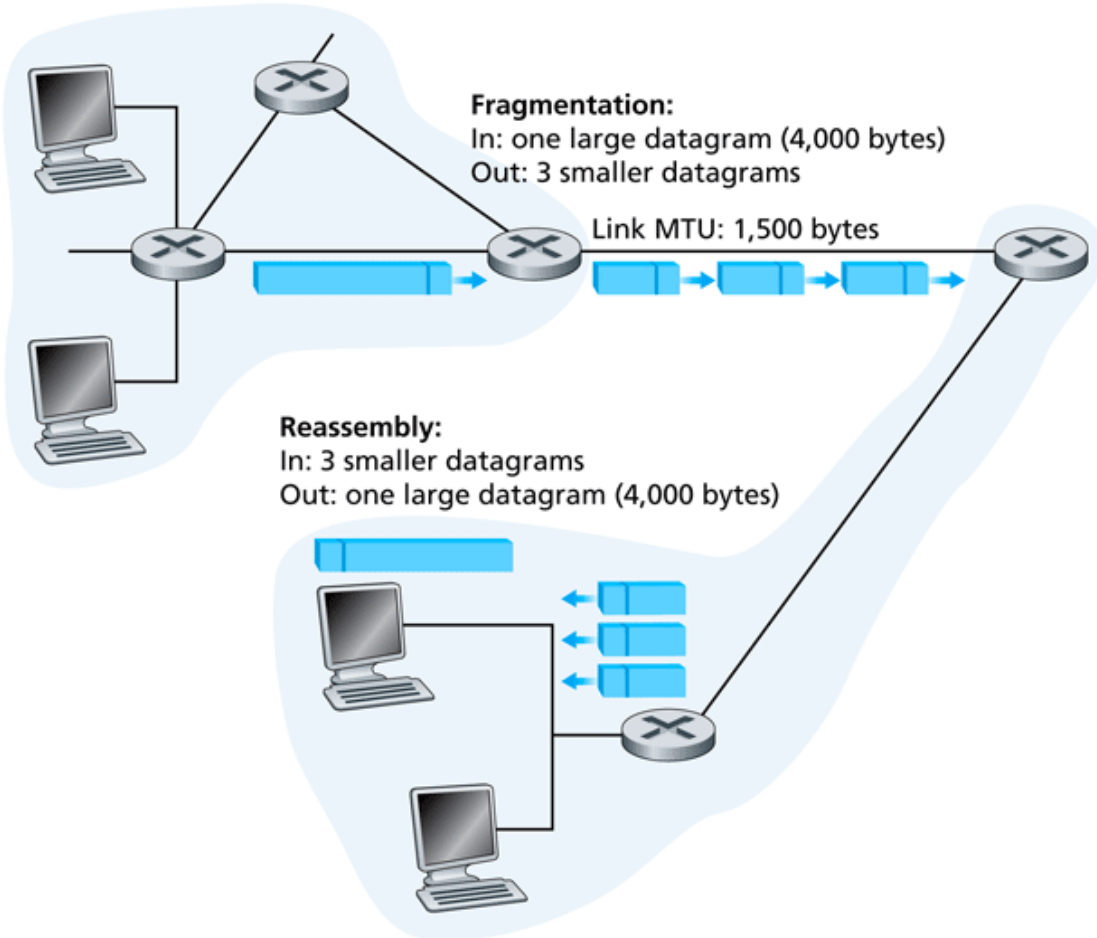
- IP address can be divided into subnet part (high-order bits) and host part (loworder bits)
- prefix notation: 223.1.1.0/24 indicates that the subnet is the high-order 24 bits
- interfaces whose IP addresses are on the same subnet can physically reach each other without a router

Forwarding Process

- check the destination address
  - is this one of my addresses?
    - send to next protocol specified in IP header
  - is this one of my subnets?
    - send to link layer to forward to the destination
  - do I have a route?
    - send to link layer to forward to next IP hop
  - destination unknown!
    - send an ICMP error to the source of the IP packet

IPv4 Fragmentation and Reassembly

- each link has an MTU (maximum transfer unit) defining largest link-layer frame
- IP packets larger than the MTU must be fragmented
- reassembly only occurs at final destination
- uses IP fragmentation fields



IPv4 Fragmentation Example

- 4000 byte datagram, 1500 byte MTU
- how long will the fragmented packets be?
  - need 20 byte header
  - first two packets 1480 bytes
  - last packet is 3980 - 2*1480 + 20 = 1040
- what will the offsets be?
  - byte position in file / 8
- MF (more fragments) flag is set to 1 in all fragments except last
- ID must be unique to sender

Original Packet

length	ID	MF	offset
4000	x	0	0

Fragments

length	ID	MF	offset
1500	x	1	0
1500	x	1	185
1040	x	0	370

ICMP

ICMP: Internet Control Message Protocol

- error reporting, ping
- network layer above IP: ICMP messages carried in IP datagrams
- ICMP message: type, code, checksum, message-specific data (RFC 792)

ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

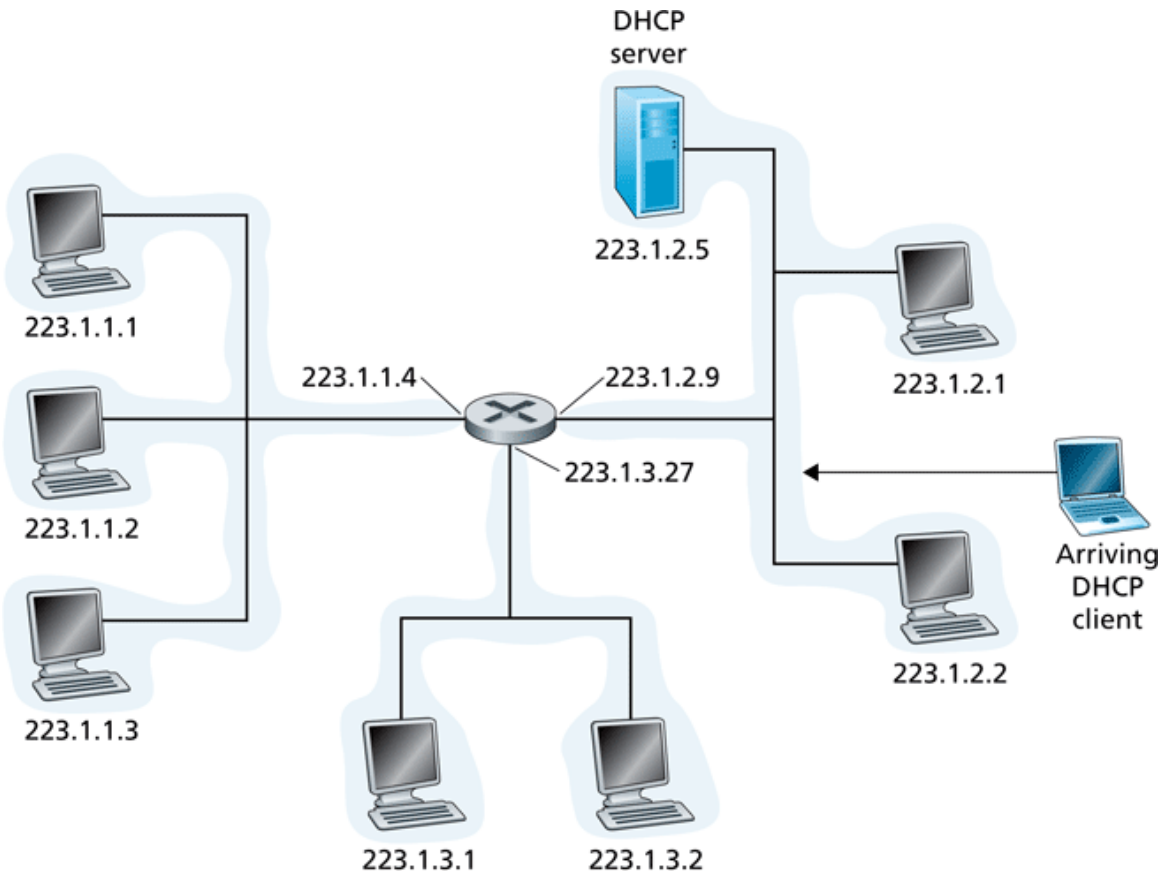
Traceroute and ICMP

- source sends UDP segments to destination
  - start with TTL $= 1$
  - increment TTL by one
  - use unlikely port number
- when nth datagram arrives at nth router
  - discard datagram (TTL expired)
  - send ICMP TTL expired message to source
  - message includes IP header, 64 bits of original datagram
- when ICMP message arrives, source calculates RTT for that hop
- traceroute takes 3 samples for each hop
- stop when ICMP returns a host unreachable packet, code 3 = port unreachable

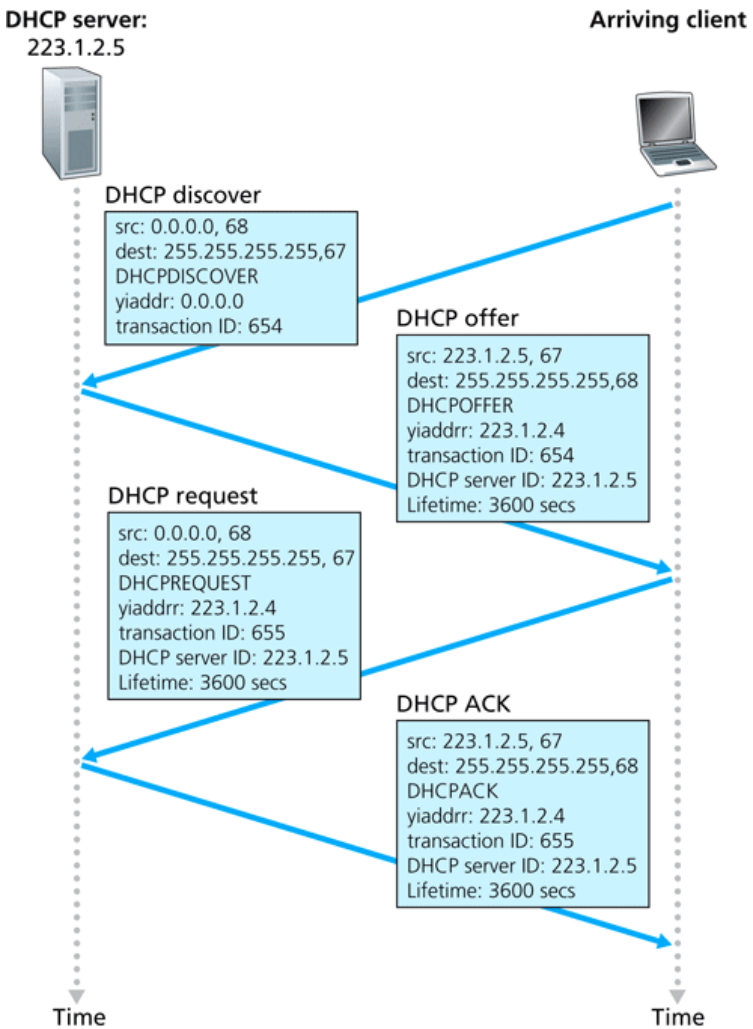
DHCP

DHCP

- IP addresses can be assigned manually
  - hard-coded into a configuration file
  - e.g. Gentoo: /etc/conf.d/net
- DHCP: dynamically get address from server



DHCP Transaction



NAT

NAT: Network Address Translation

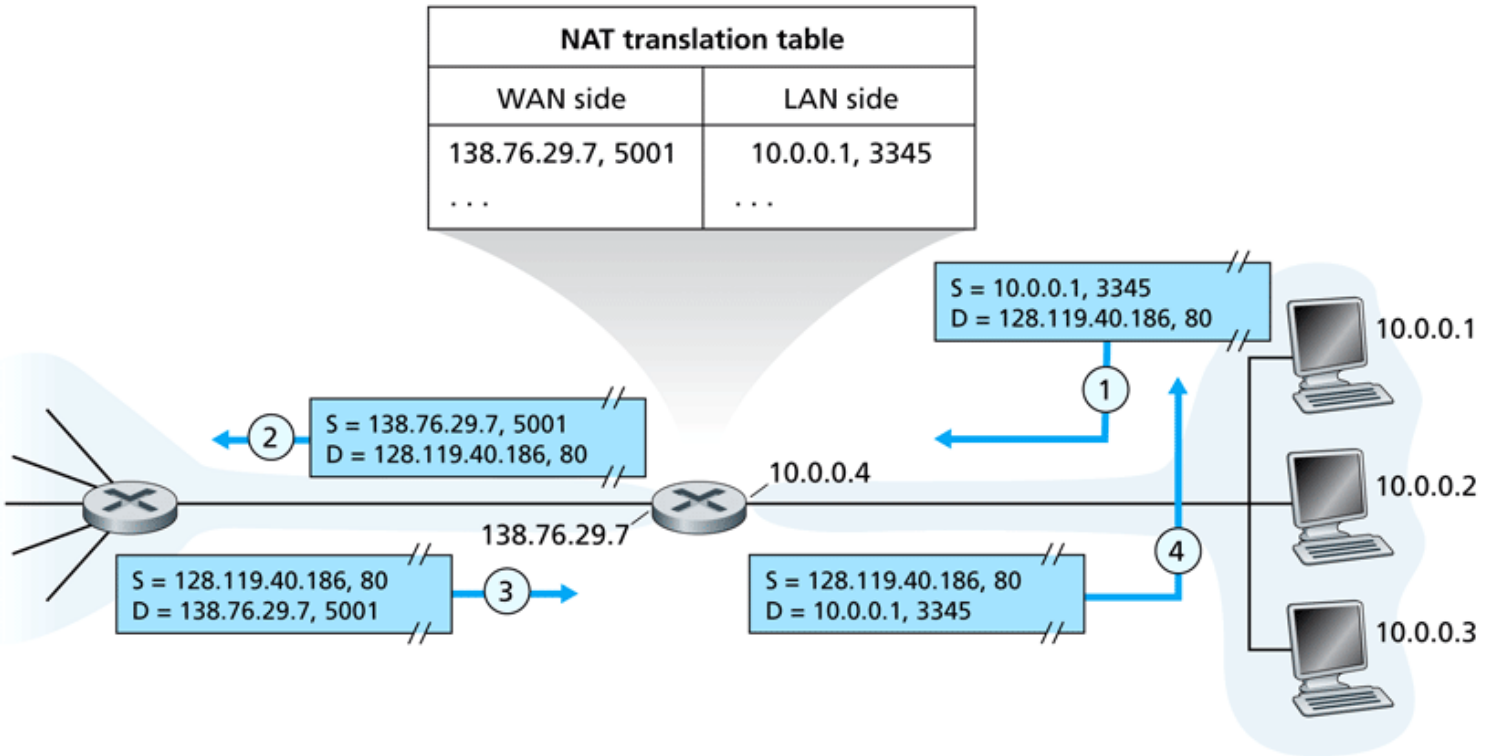


Figure 4.22 ♦ Network address translation

- use public port numbers to map to private connections
- can support 60,000+ connections with a single IP address

NAT: the Good ...

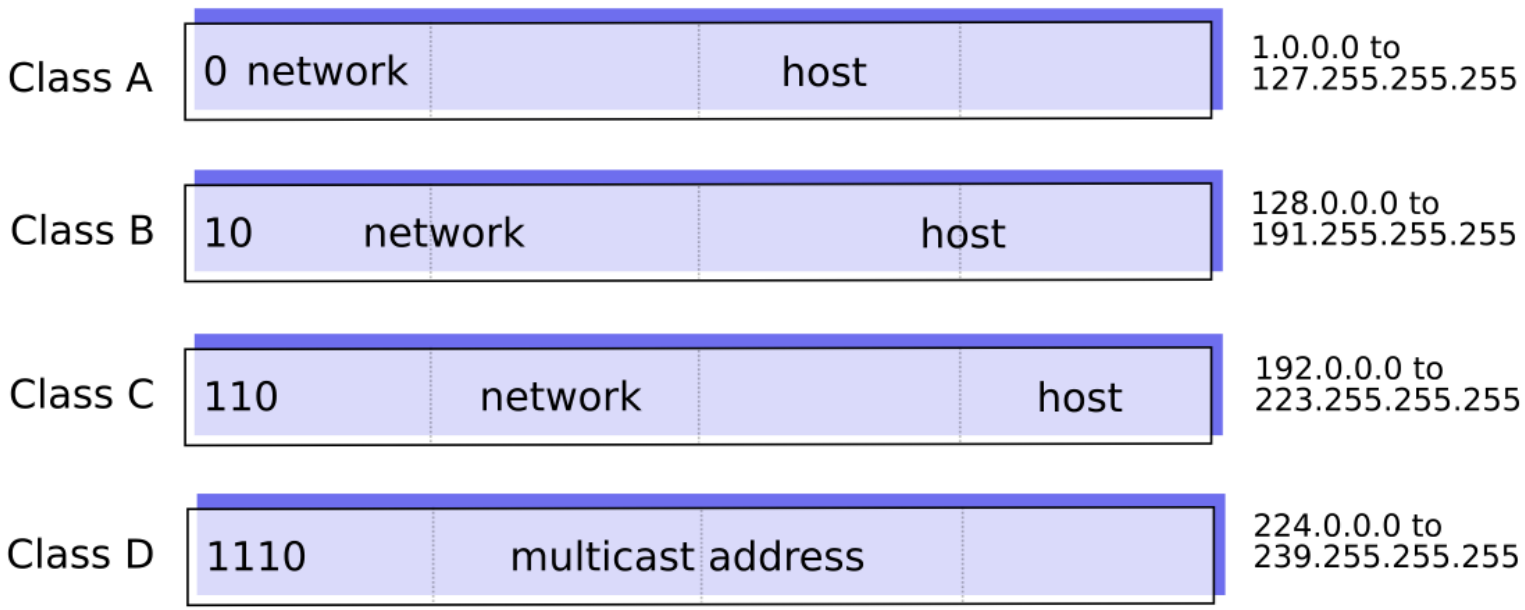
- conserves IP address space: private network only needs one IP address
- can change private IP addresses without notifying rest of Internet (DNS)
- can change ISP without changing IP addresses
- acts as a type of firewall - only reachable ports are those that you open first

...and the Bad

- layer violation: routers should only process IP, ports are in TCP/UDP
- violates end-to-end nature of Internet: any host can open a connection to any other host – makes running local servers and peer-to-peer applications hard
- address shortage should be resolved by IPv6
- individual computers should be made as secure as possible, rather than relying on firewalls or NAT boxes
- prevents many peer-to-peer applications from working
  - note: many emerging hacks and standards, including UPnP, that allow an application to create a mapping for a server running behind the NAT

CIDR

Classful IP Addressing



- used in early days of Internet to assign addresses to organizations
- led to waste: organizations want at least a B (65,000 addresses), even if they have 1000 machines (4 class Cs)
- early Internet users even got a class A (Stanford was 36.0.0.0)
- quickly ran out of addresses

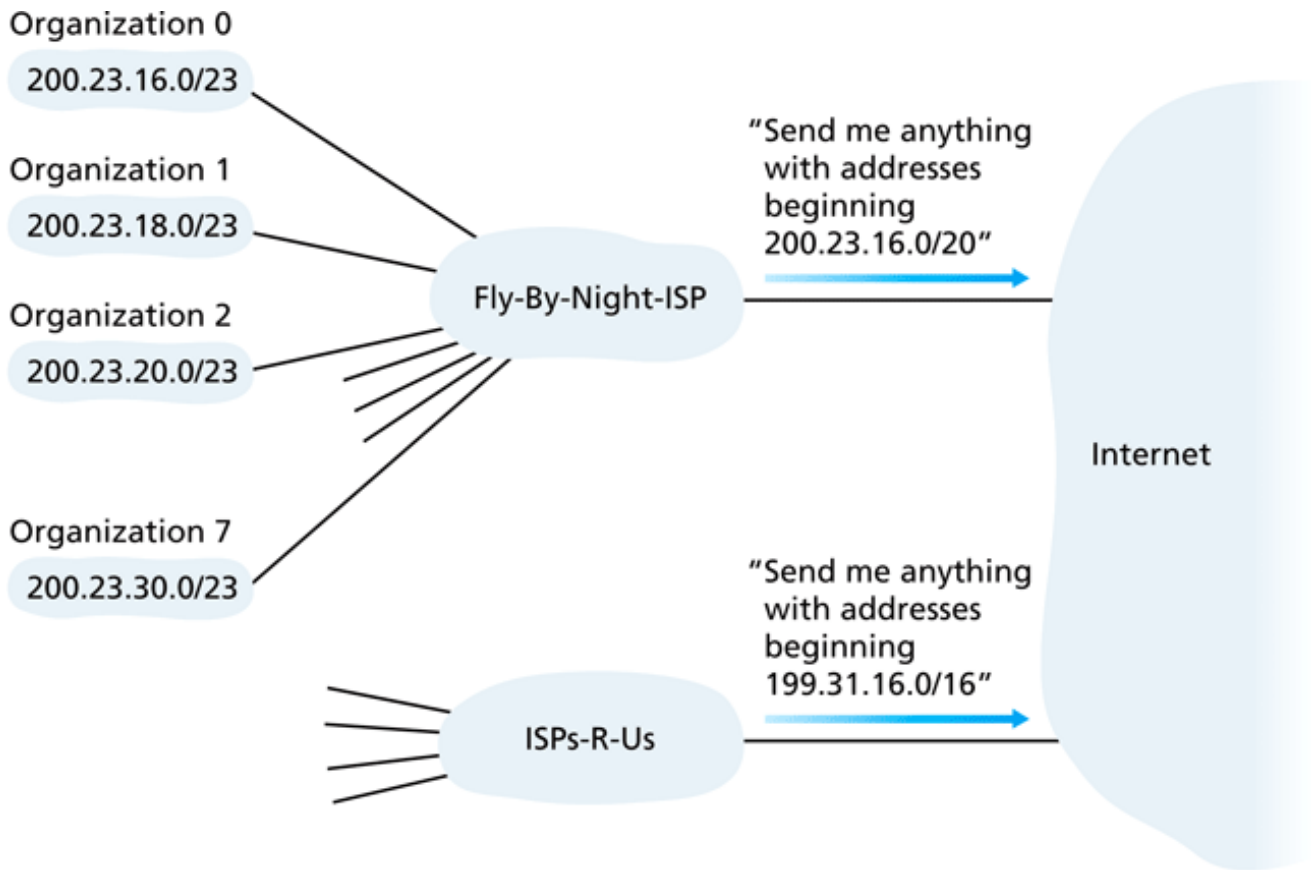
CIDR: Classless InterDomain Routing

- subnet portion of address has an arbitrary length
- address format: a.b.c.d/x, where x is number of bits in subnet portion
- example:
  - 11001000 00010111 00010000 00000000
  - 200.23.16.0/23
- enables conservation of IP address space, efficient routing
- IANA required organizations to return Class A, B addresses and re-number

Using CIDR Addresses

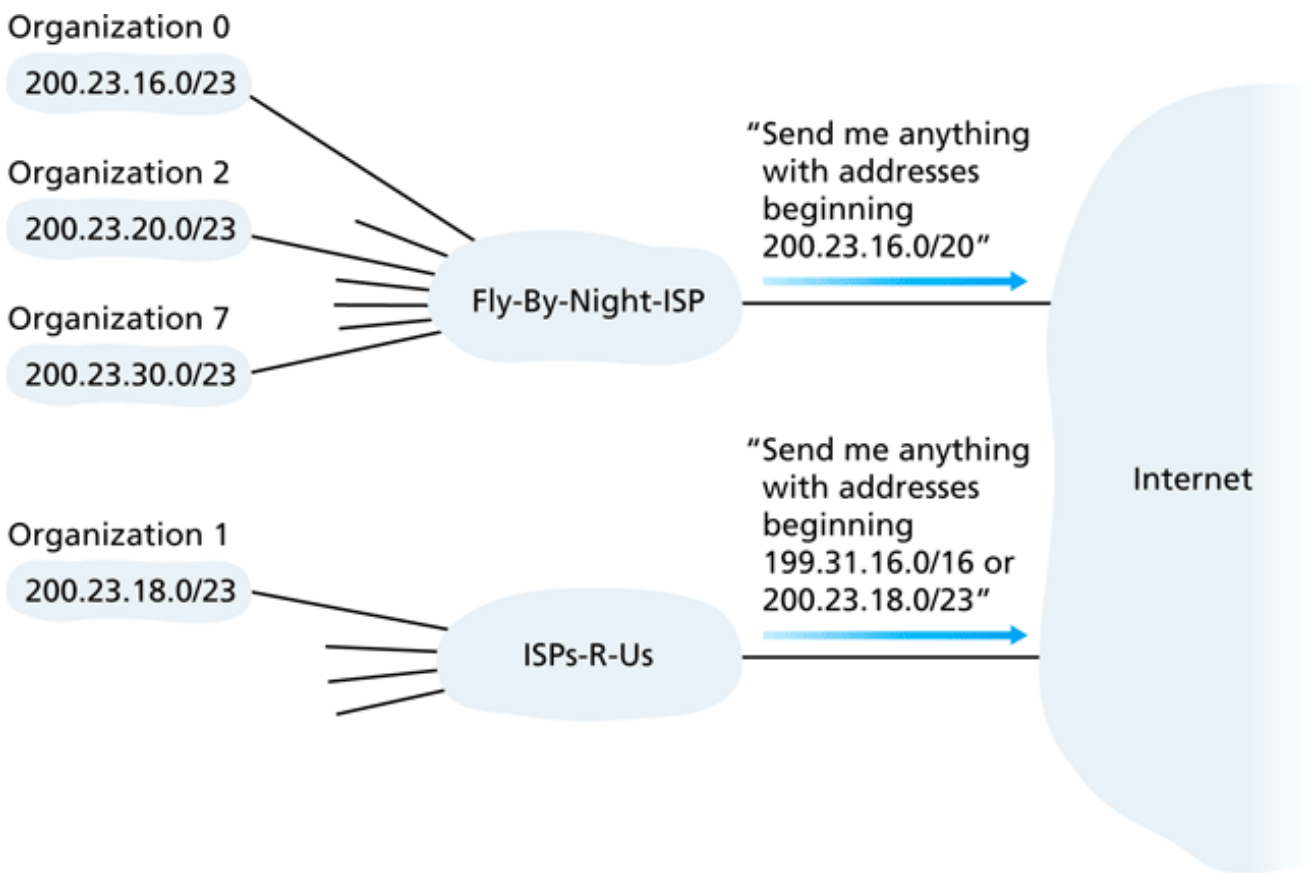
- each ISP has an assigned address space, from ICANN
- e.g 11001000 00010111 00010000 00000000 200.23.16.0/20
- can allocate to its customers
  - 11001000 00010111 00010000 00000000 200.23.16.0/23
  - 11001000 00010111 00010010 00000000 200.23.18.0/23
  - 11001000 00010111 00010100 00000000 200.23.20.0/23
  - ...
  - 11001000 00010111 00011110 00000000 200.23.30.0/23

CIDR and Route Aggregation



- using CIDR allows routes to be aggregated

Breaking Route Aggregation



- changing ISPs (and keeping IP addresses) breaks route aggregation

IPv6

Motivation: 32-bit address space running out

- short-term solutions
  - CIDR, reclaim class A addresses
  - NAT
- IETF coordinated design process, many proposals
  - discussion on big-internet and IPng lists
  - CATNIP - variable length addresses, interoperability among many protocols
  - NIMROD - variable length, hierarchical addresses, separate host identification (naming) from host location (routing)
  - TUBA - use CLNP for network layer, with OSI-specified big addresses
  - SIPP: Simple Internet Protocol Plus (Steve Deering) - 64-bit addresses, remove unneeded functionality
- and the winner is ...

Motivation: 32-bit address space running out

- short-term solutions
  - CIDR, reclaim class A addresses
  - NAT
- IETF coordinated design process, many proposals
  - discussion on big-internet and IPng lists
  - CATNIP - variable length addresses, interoperability among many protocols
  - NIMROD - variable length, hierarchical addresses, separate host identification (naming) from host location (routing)
  - TUBA - use CLNP for network layer, with OSI-specified big addresses
  - SIPP: Simple Internet Protocol Plus (Steve Deering) - 64-bit addresses, remove unneeded functionality
- and the winner is ...
  - SIP $\Rightarrow$ SIPP (SIP + PIP + IPAE) $\Rightarrow$ IPv6

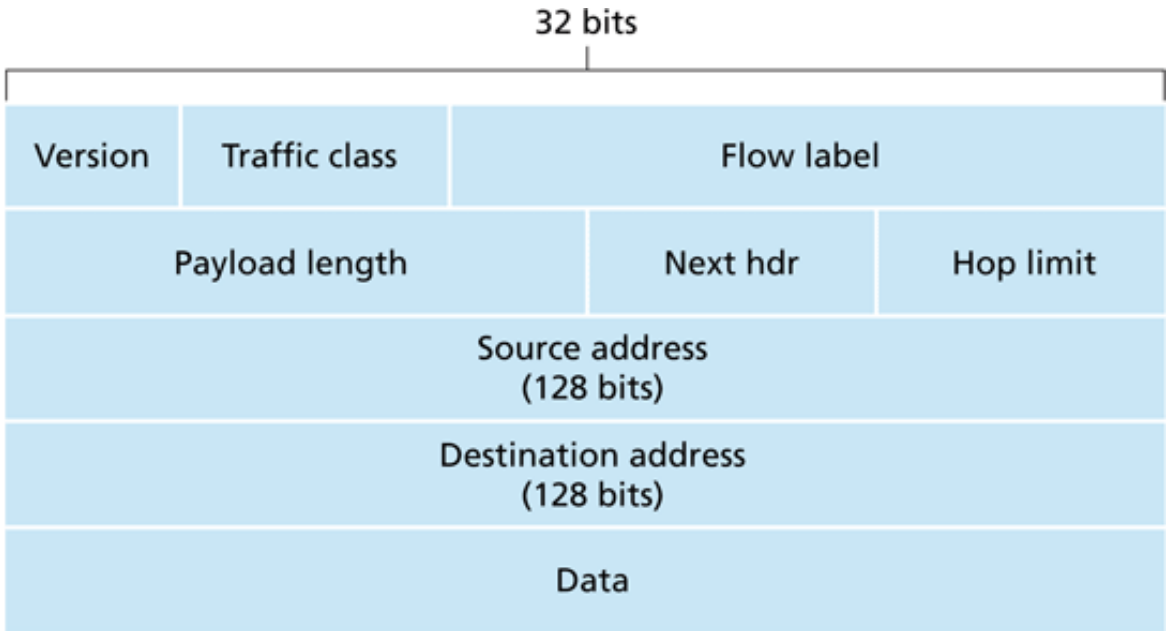
How Big Is Big Enough?

- proposals
  - fixed length, 64 bits
  - variable length, up to 160 bits
  - compromise: 128 bits
- theoretically perfect allocation
  - 128 bits $= 3.4 * 10^{38}$ addresses
  - $7 * 10^{27}$ atoms in your body, so $4.86 * 10^{10}$ addresses per atom
  - 6 billion people in the world
  - 8 billion addresses per atom in your body
- in reality ...
  - prefix (address type): 3 bits
  - registry ID: n bits
  - provider ID: m bits
  - subscriber ID: o bits
  - intra-subscriber ID : $125 - n - m - o$ bits
- address space can always be wasted

As Long As We're Designing a New Version of IP ...

- simplify IP header
  - get rid of functionality not used or needed in IPv4
  - speed processing/forwarding
  - no checksum
  - no fragmentation
  - fixed 40-byte header, no options
- support emerging QoS proposals
  - traffic class, flow label

IPv6 Header



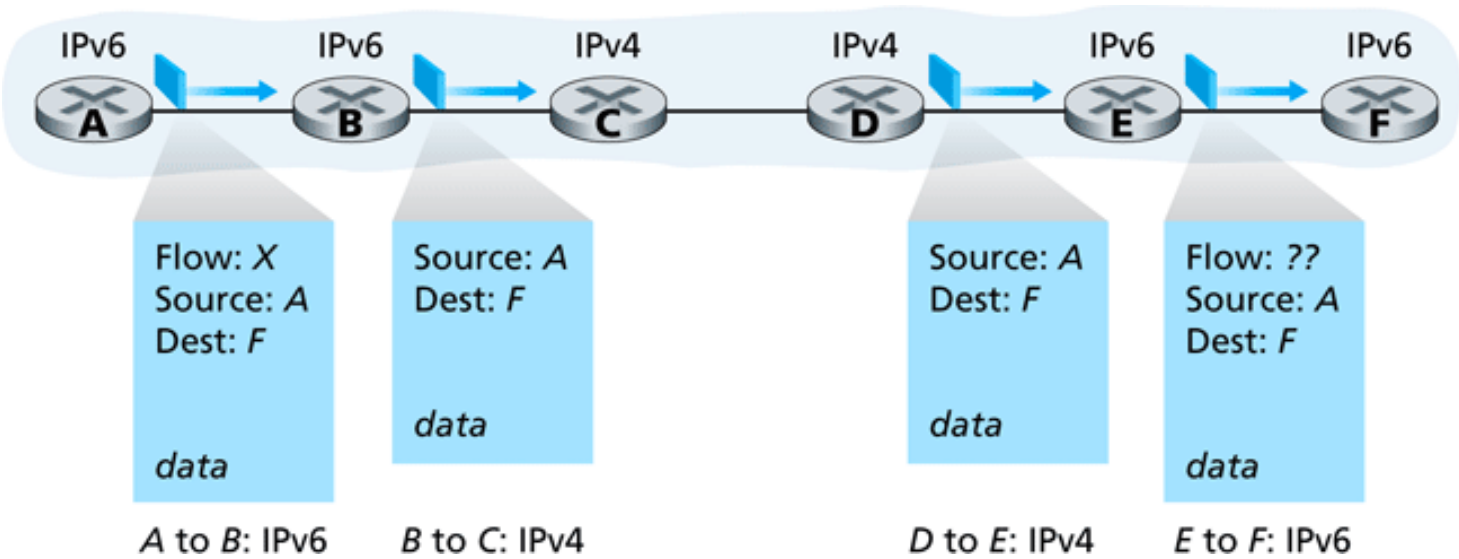
- traffic class: enable routers to map traffic into classes (delay, loss guarantees, etc)
- flow label: uniquely identify all packets for a particular flow/application, used for QoS
- Next Header: upper layer protocol or option

What Happened to IPv5?

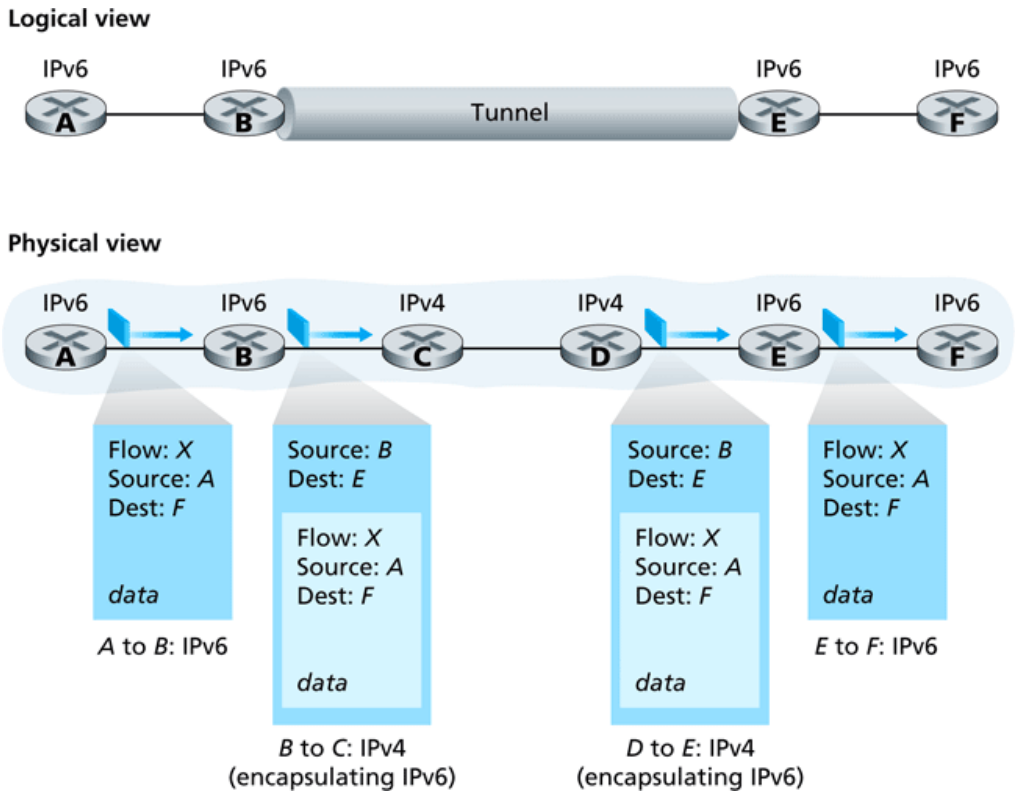
- 0-3 : unassigned
- 4 : IPv4
- 5 : ST (Stream Protocol), not used
- 6 : IPv6 (was SIP, then SIPP)
- 7 : CATNIP
- 8 : PIP
- 9 : TUBA
- 10-15 : not assigned

IPv6 Transition

- can't upgrade all routers at the same time or on the same day
- must interoperate between IPv4 and IPv6
- dual-stack: support both IPv4 and IPv6 in a single host/router
  - can deliver native IPv6 traffic where supported
  - loses IPv6 information when translating to IPv4



IPv6 Tunnels



- tunnels: IPv6 carried as payload in IPv4 packet
  - can carry IPv6 packets end-to-end
  - requires configuration

The End is Near!

- some regional registries have already run out of addresses
- ▸ Geoff Huston
- ▸ Wikipedia Page