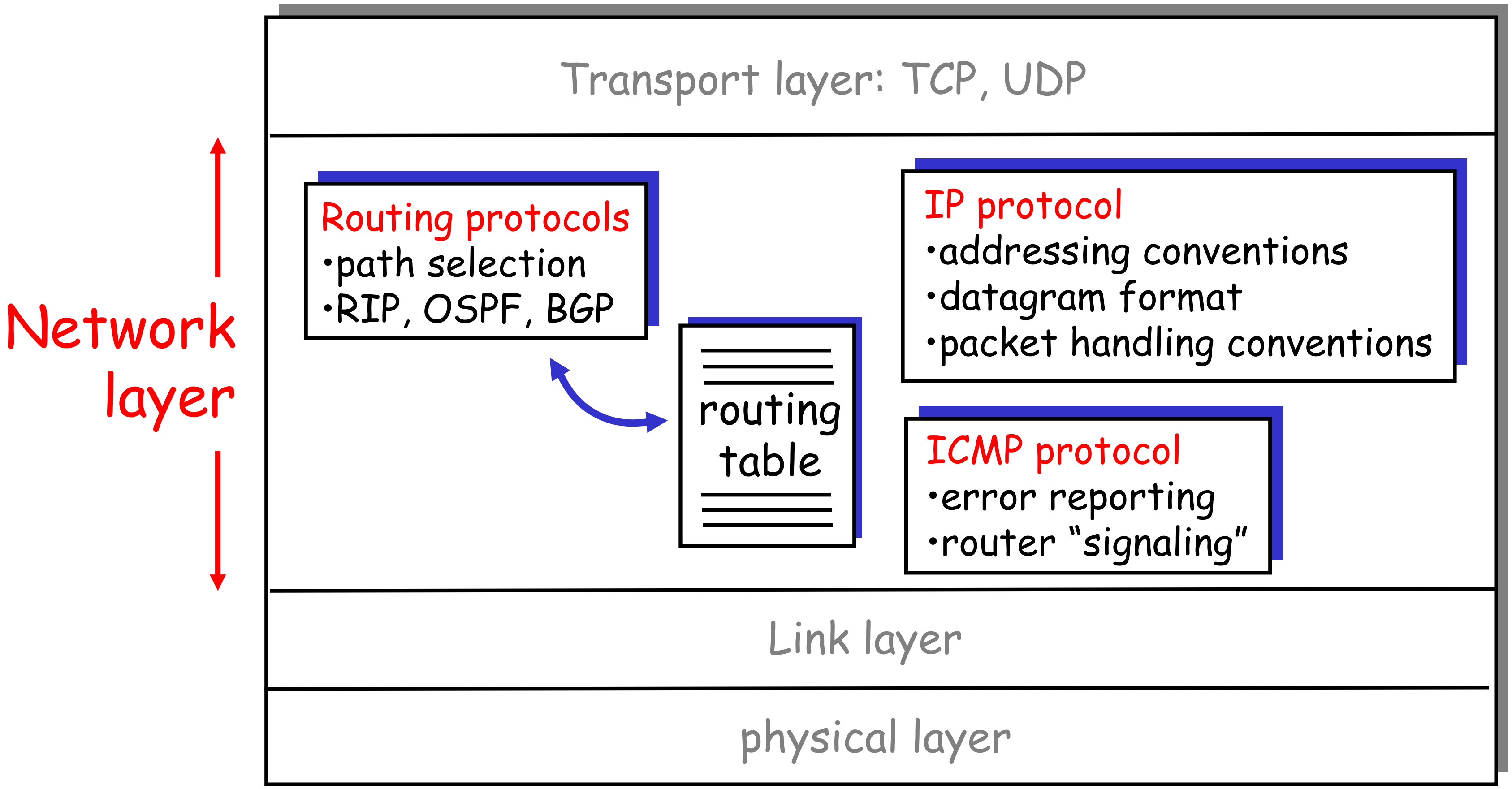
Chapter 5: Network Layer

- 5.1 Introduction
- 5.2 Virtual circuit and datagram networks
- 5.3 What's inside a router?
- 5.4 Routing algorithms:
  - Dijkstra's algorithm
  - Broadcast routing
  - Link state
  - Distance vector
  - Hierarchical routing
- 5.5 Routing in the Internet
- 5.6 IP: Internet Protocol
  - IPv4 Datagram format
  - IP fragment
  - IPv4 addressing
  - NAT
  - ARP
  - ICMP
  - IPv6

The Internet Network layer

Host, router network layer functions:



IP datagram format

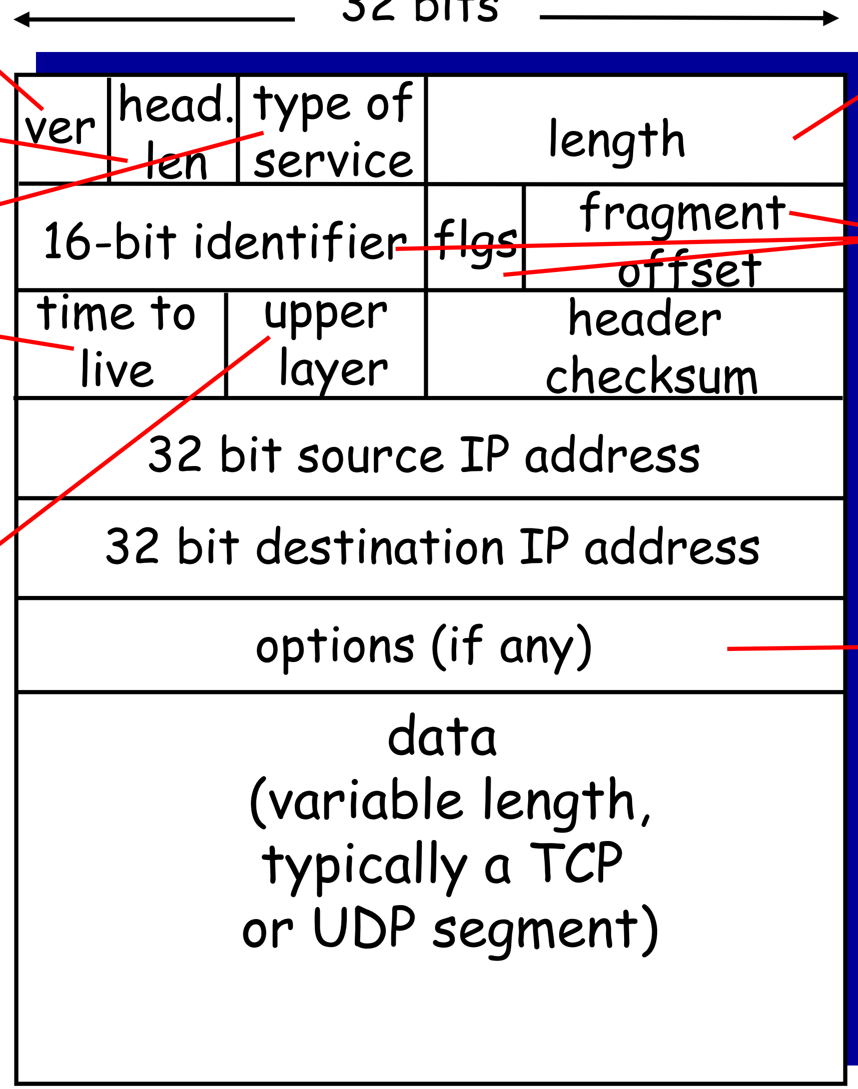
IP protocol version number

header length (in 4 bytes)

"type" of data

max number remaining hops (decremented at each router)

upper layer protocol to deliver payload to



total datagram length (bytes)

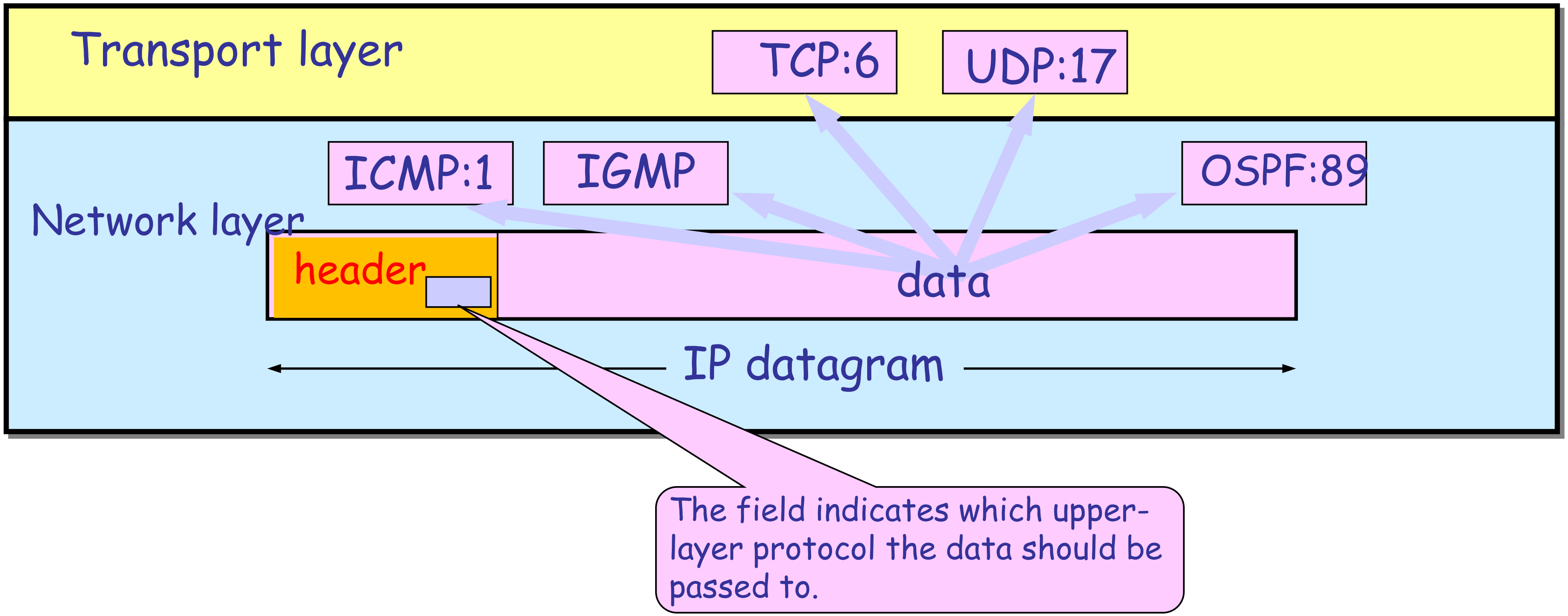
for fragmentation/ reassembly

e.g. timestamp, record route taken, specify list of routers to visit

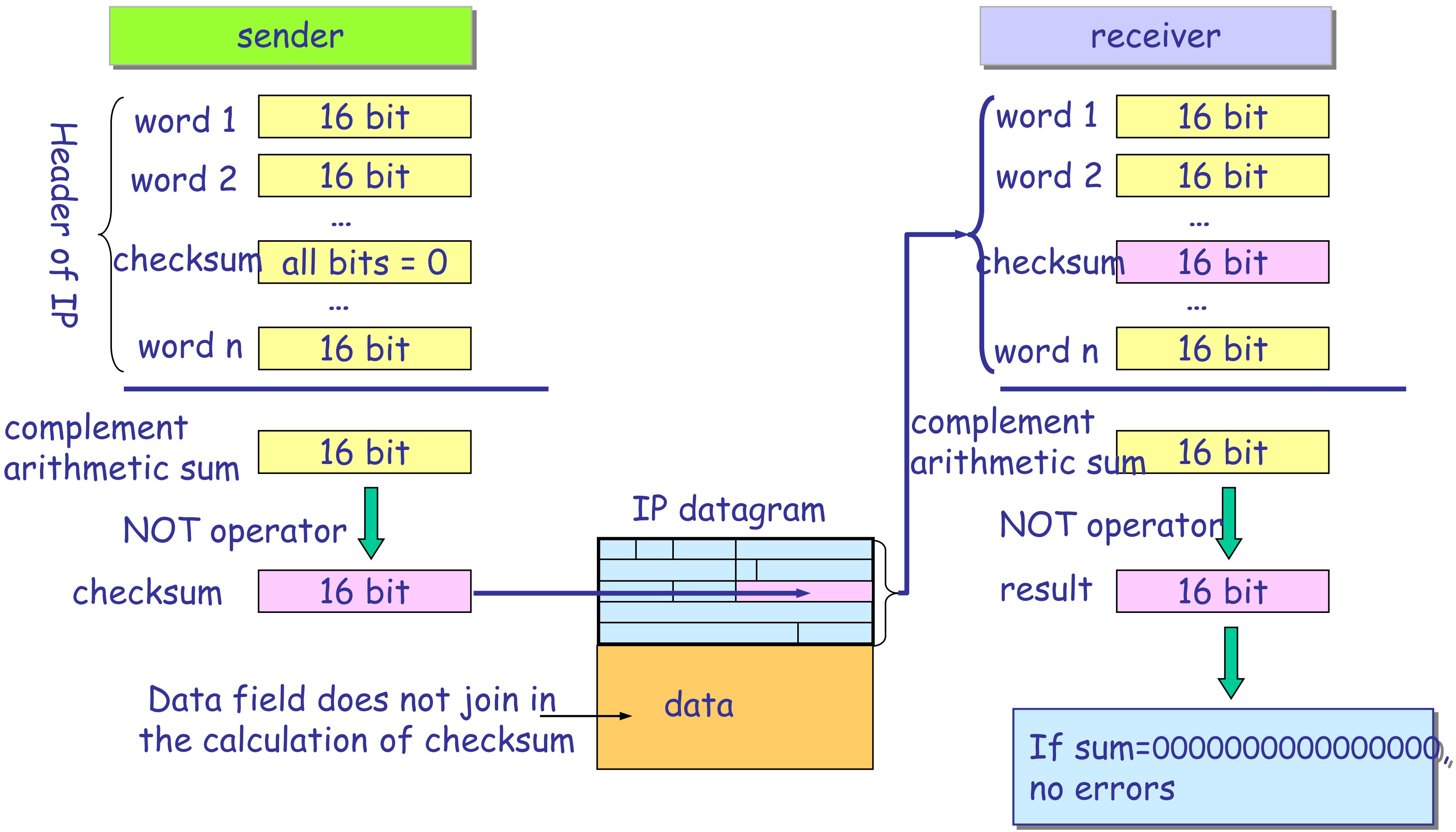
how much overhead?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

Upper-layer protocol field

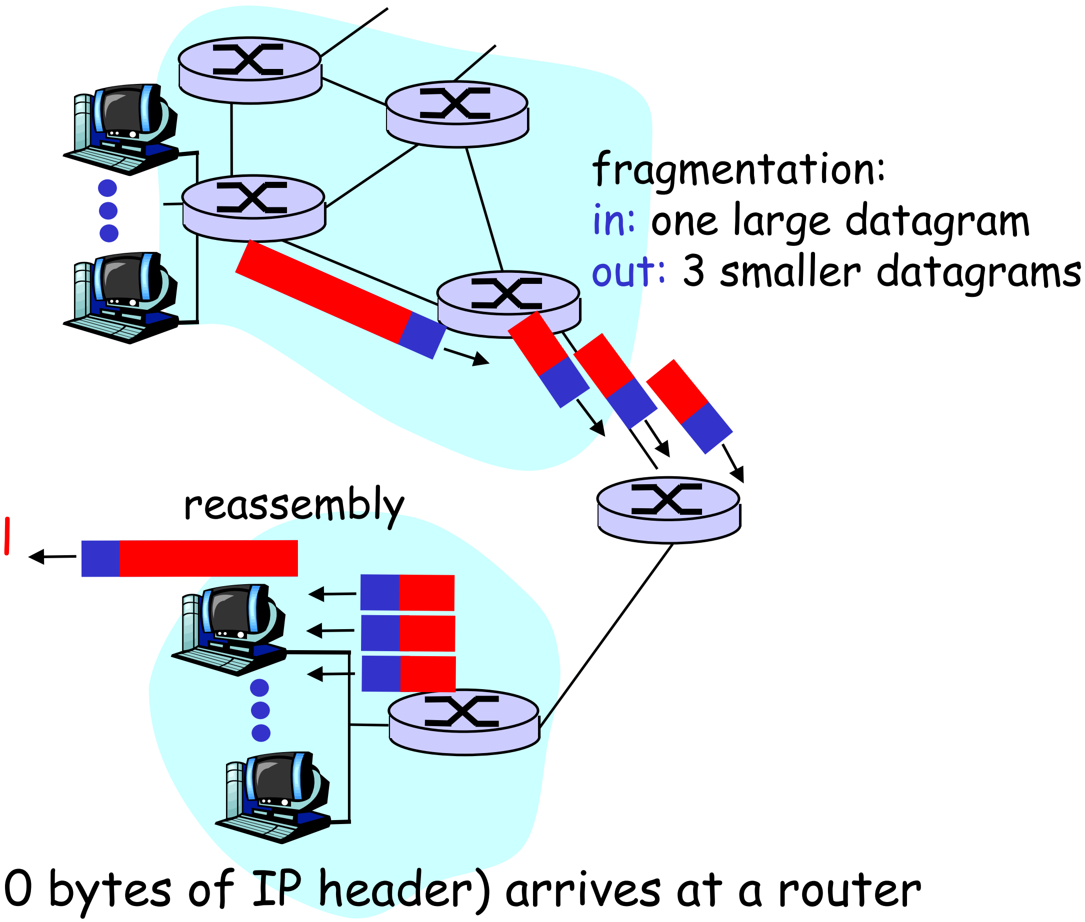


Header checksum



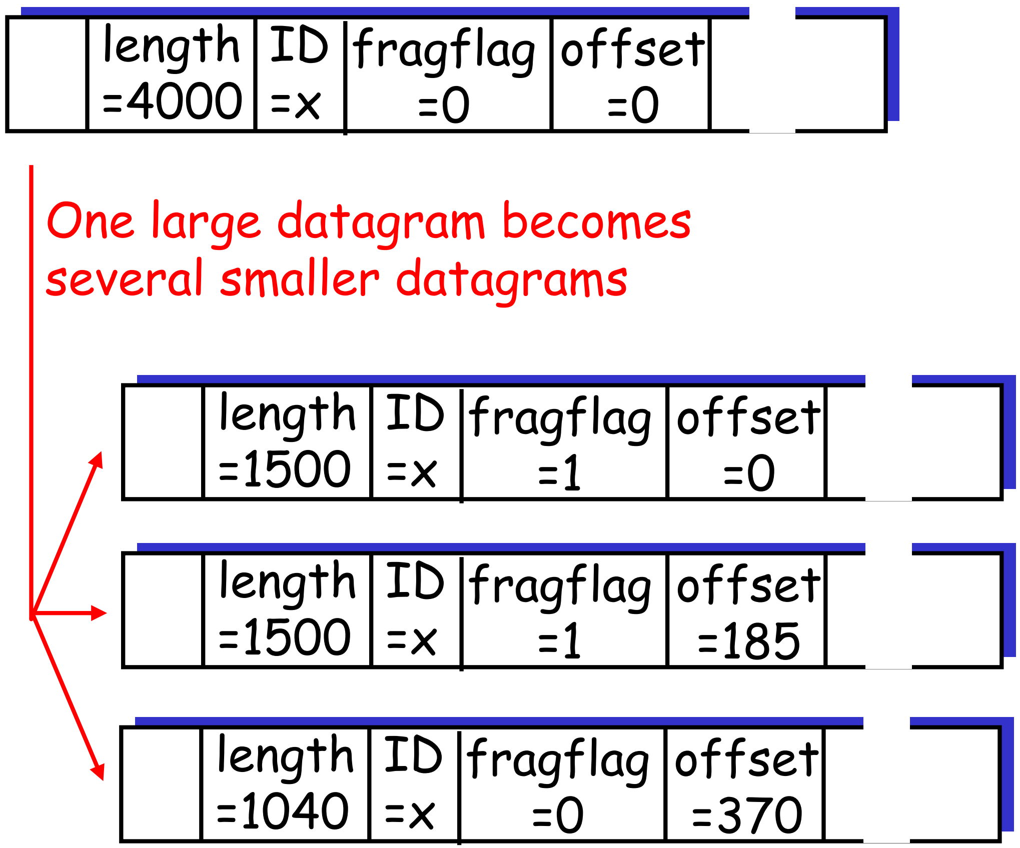
IP Fragmentation & Reassembly

- network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments

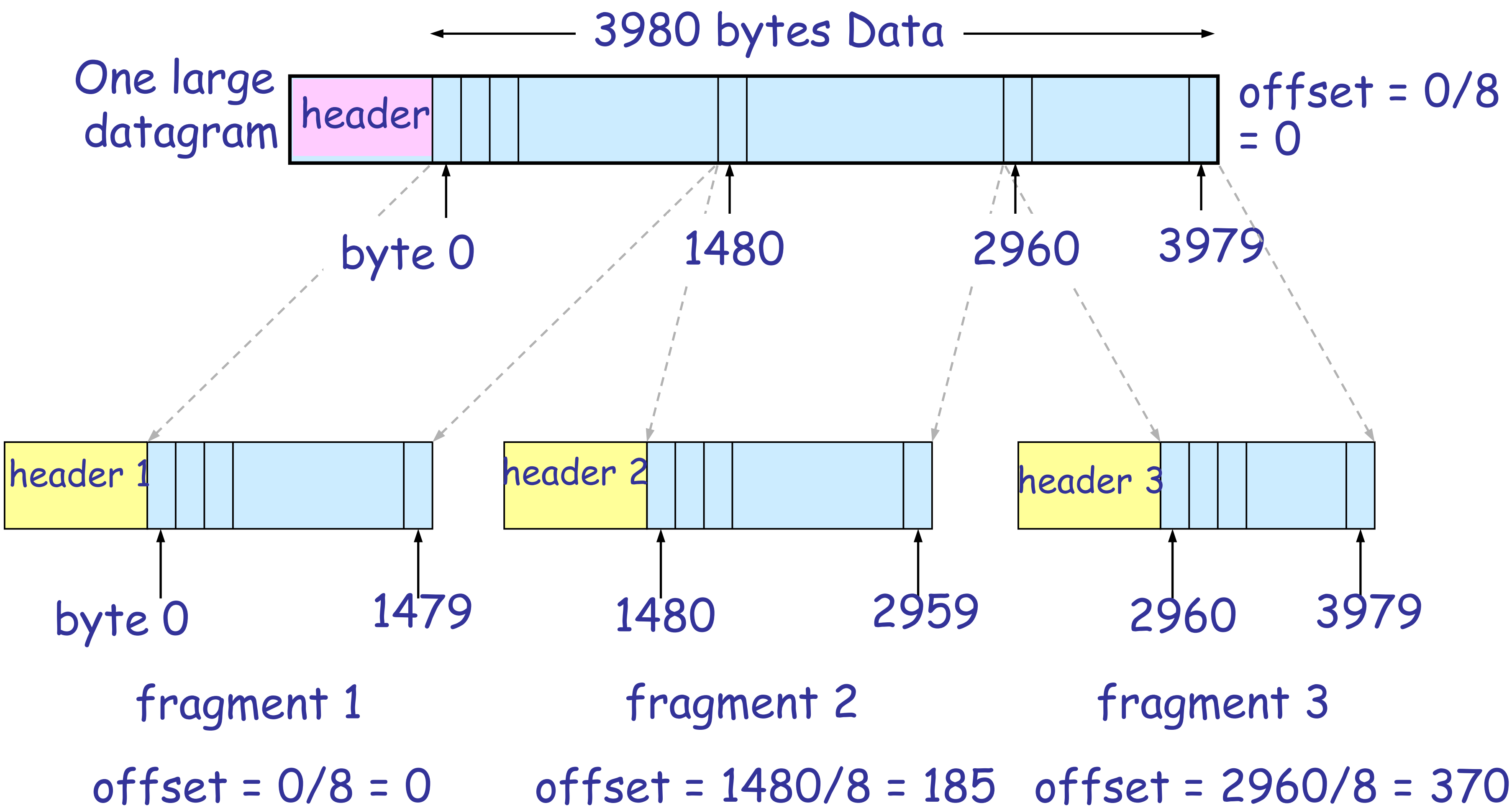


Q: A datagram of 4000 bytes (20 bytes of IP header) arrives at a router and must be forwarded to a link with an MTU of 1500 bytes. How to do?

IP Fragmentation and Reassembly



IP Fragmentation



IP datagram

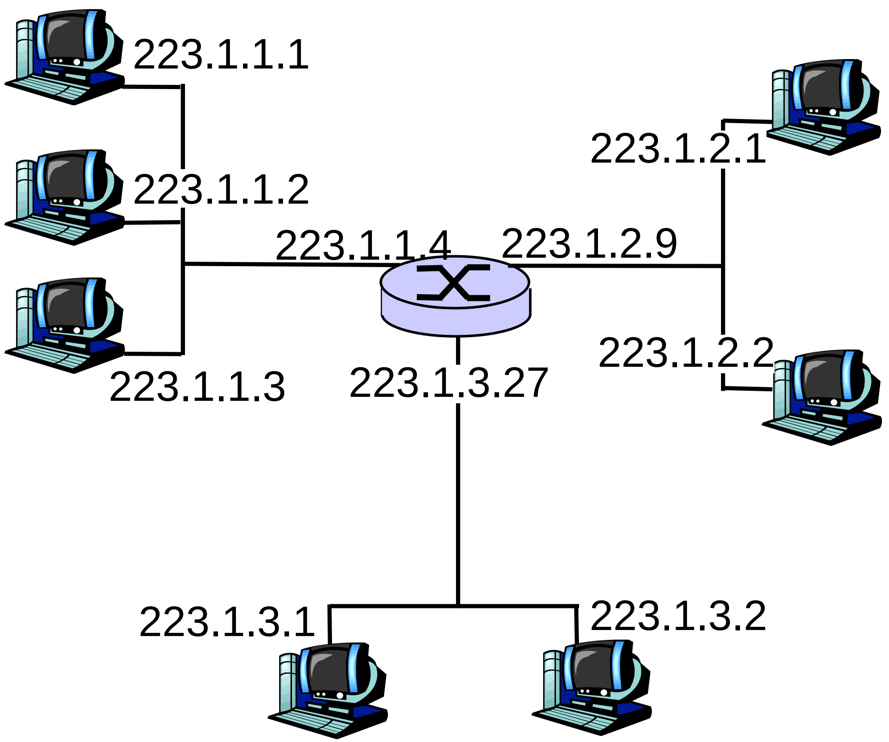
- Now assume a IP datagram is captured. The first 20 bytes are as follows:

  0x45 0x00 0x00 0x3C 0x1A 0x37 0x00
  0x00 0x80 0x01 0x6E 0x31 0xC0 0xA8
  0x01 0xD4 0xD3 0x9B 0x1C 0x41

Please try to analyze the value and meaning of each field in the IP datagram header.

IP Addressing: introduction

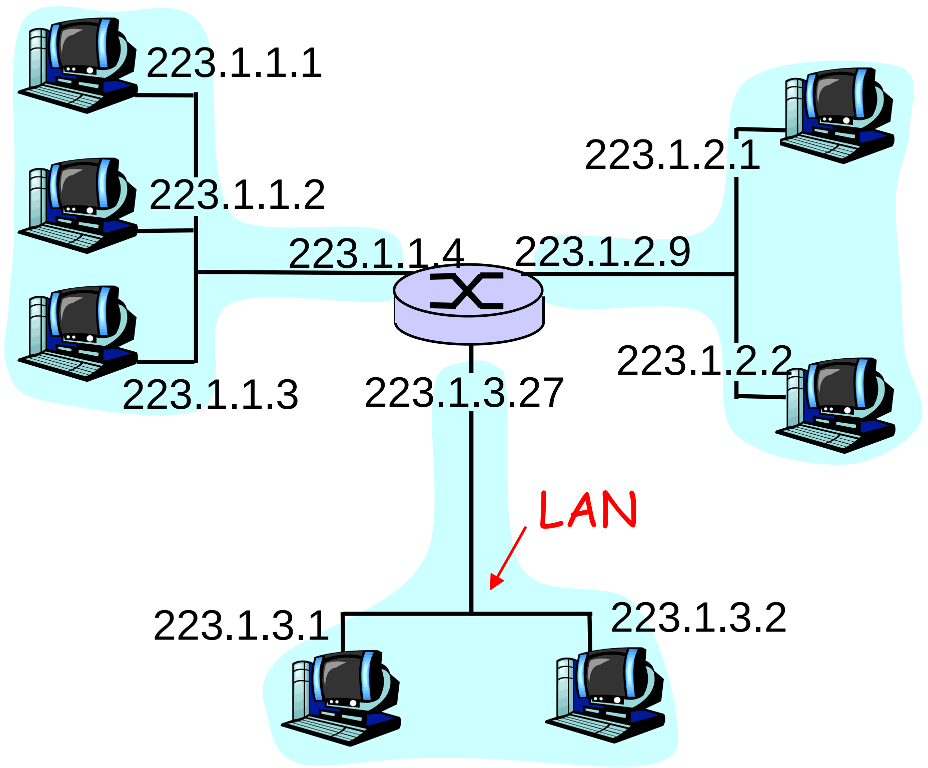
- **IP address:** 32-bit identifier for host, router *interface*
- ***interface:*** connection between host, router and physical link
  - router's typically have multiple interfaces
  - host may have multiple interfaces
  - IP addresses associated with interface, not host, router



$$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$$

IP Addressing

- IP address:
  - network part (high order bits)
  - host part (low order bits)
- *What's a network ?* (from IP address perspective)
  - device interfaces with same network part of IP address
  - can physically reach each other without intervening router

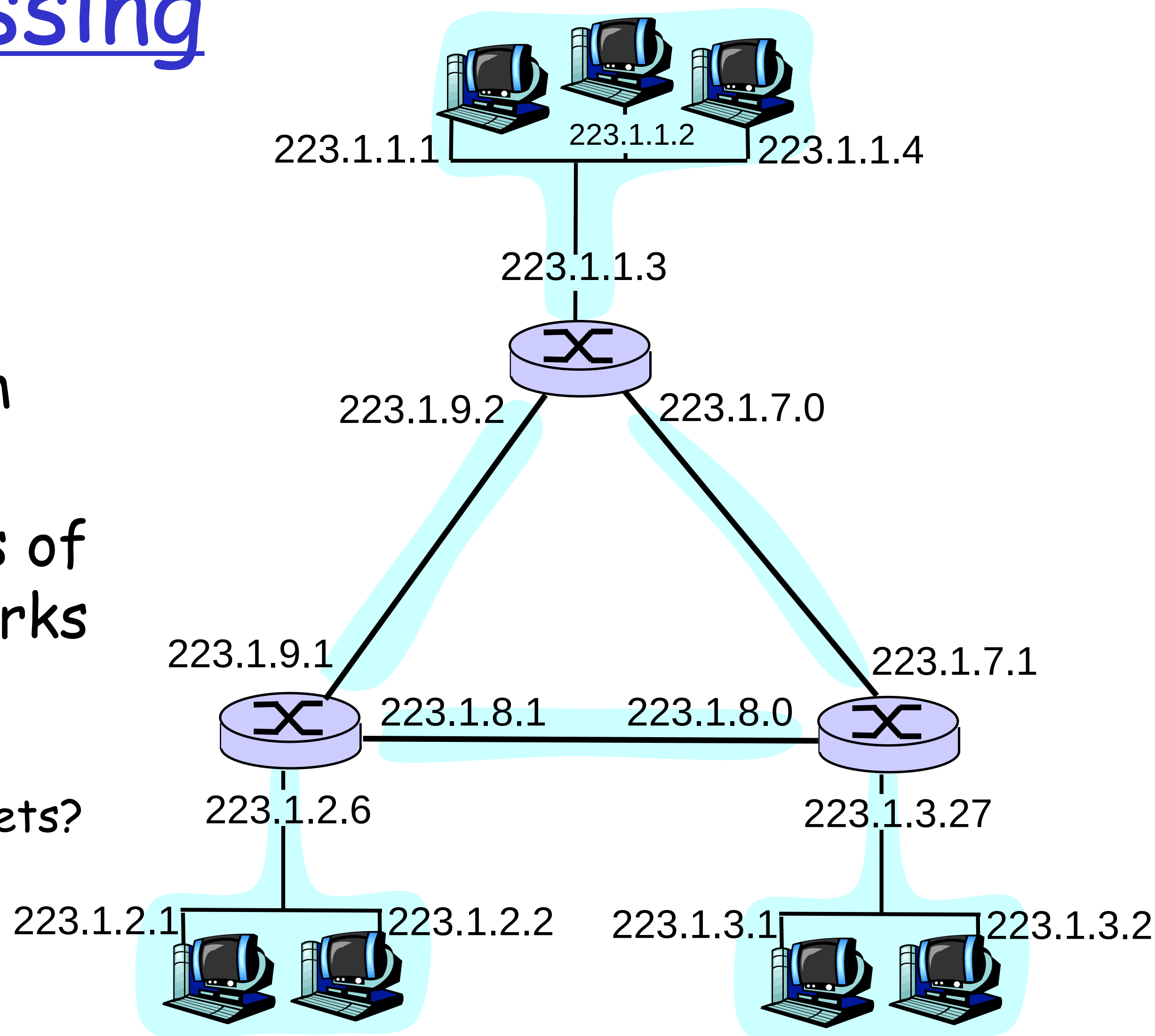


network consisting of 3 IP networks (for IP addresses starting with 223, first 24 bits are network address)

IP Addressing

How to find the networks?

- Detach each interface from router, host
- create "islands of isolated networks

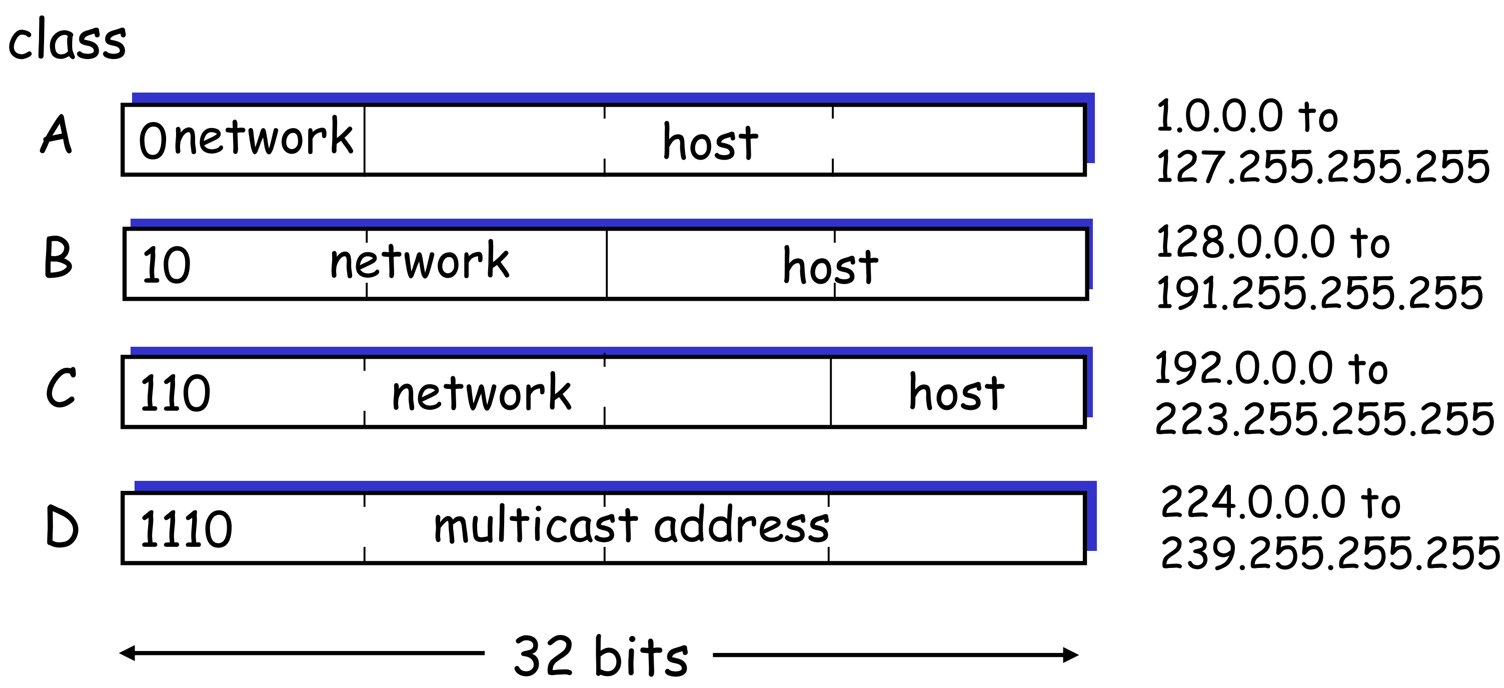


How many networks/subnets?

IP Addresses

given notion of "network", let's re-examine IP addresses:

"class-full" addressing:



IP addressing

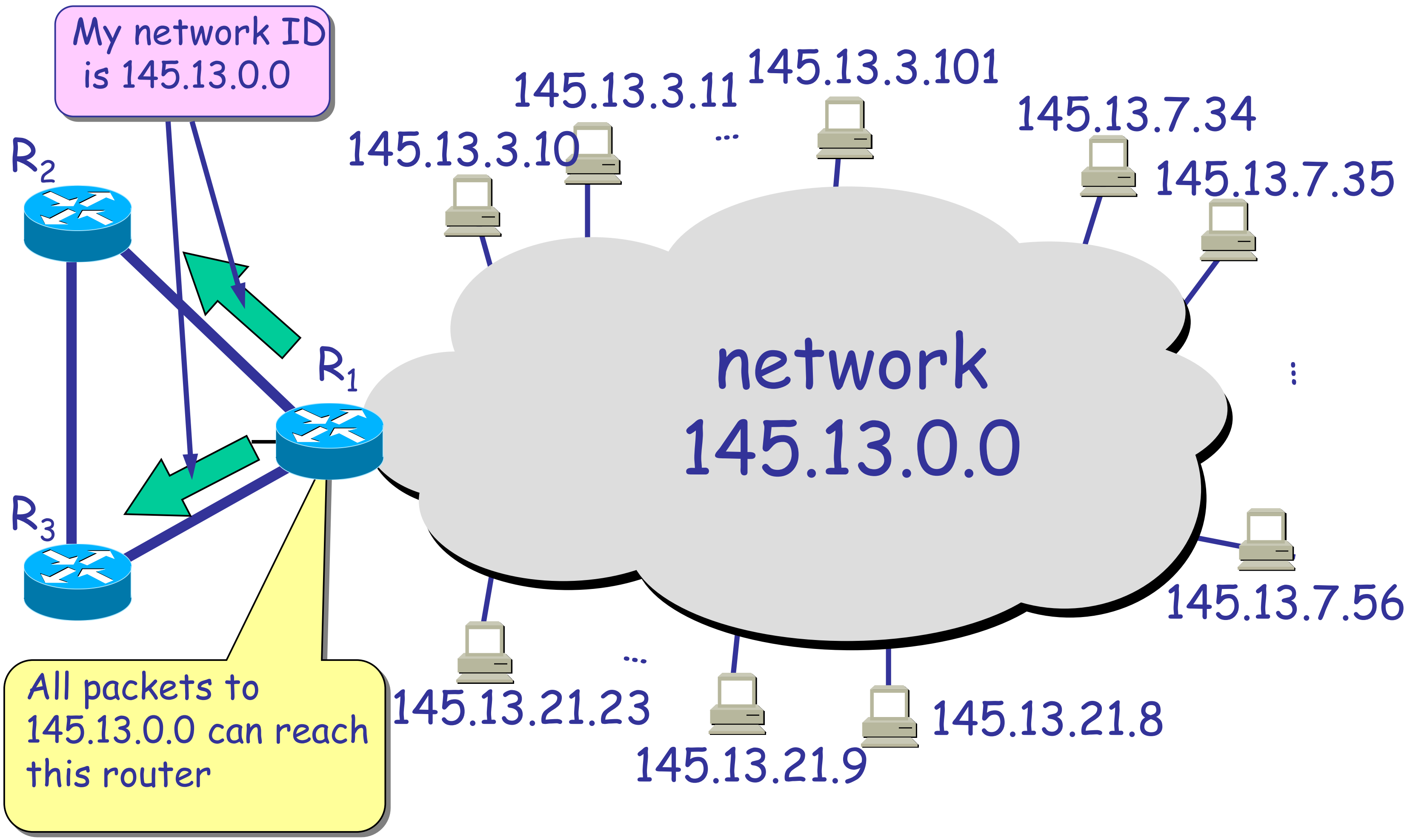
- Dotted-decimal notation: 193.32.216.9
- classful addressing:
  - inefficient use of address space, address space exhaustion
  - e.g., class B net allocated enough addresses for 65K hosts, even if only 2K hosts in that network
- Subnetting and subnet mask

network part | subnet part | host part

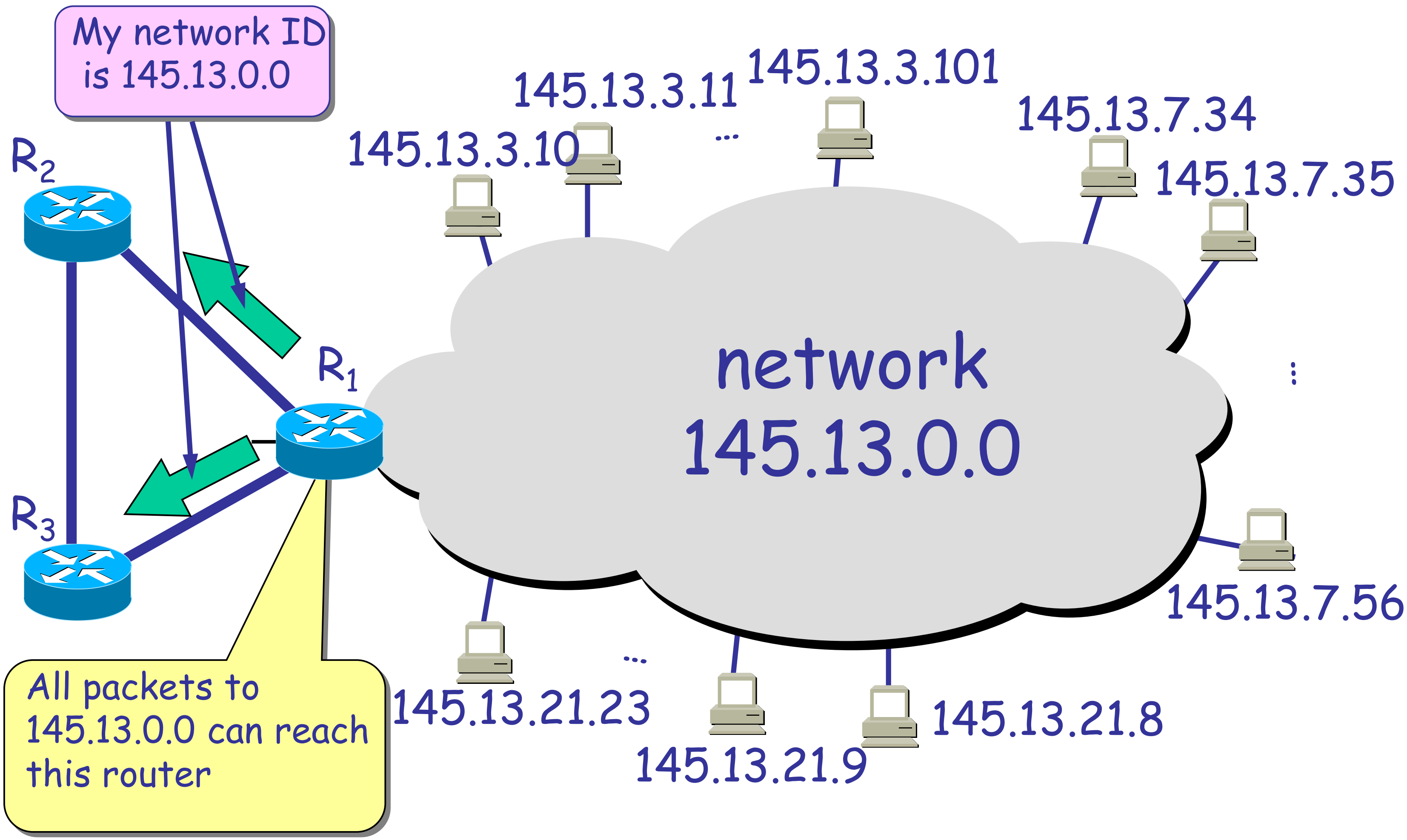
11001000 00010111 00010000 010 00000

200.23.16.64/27

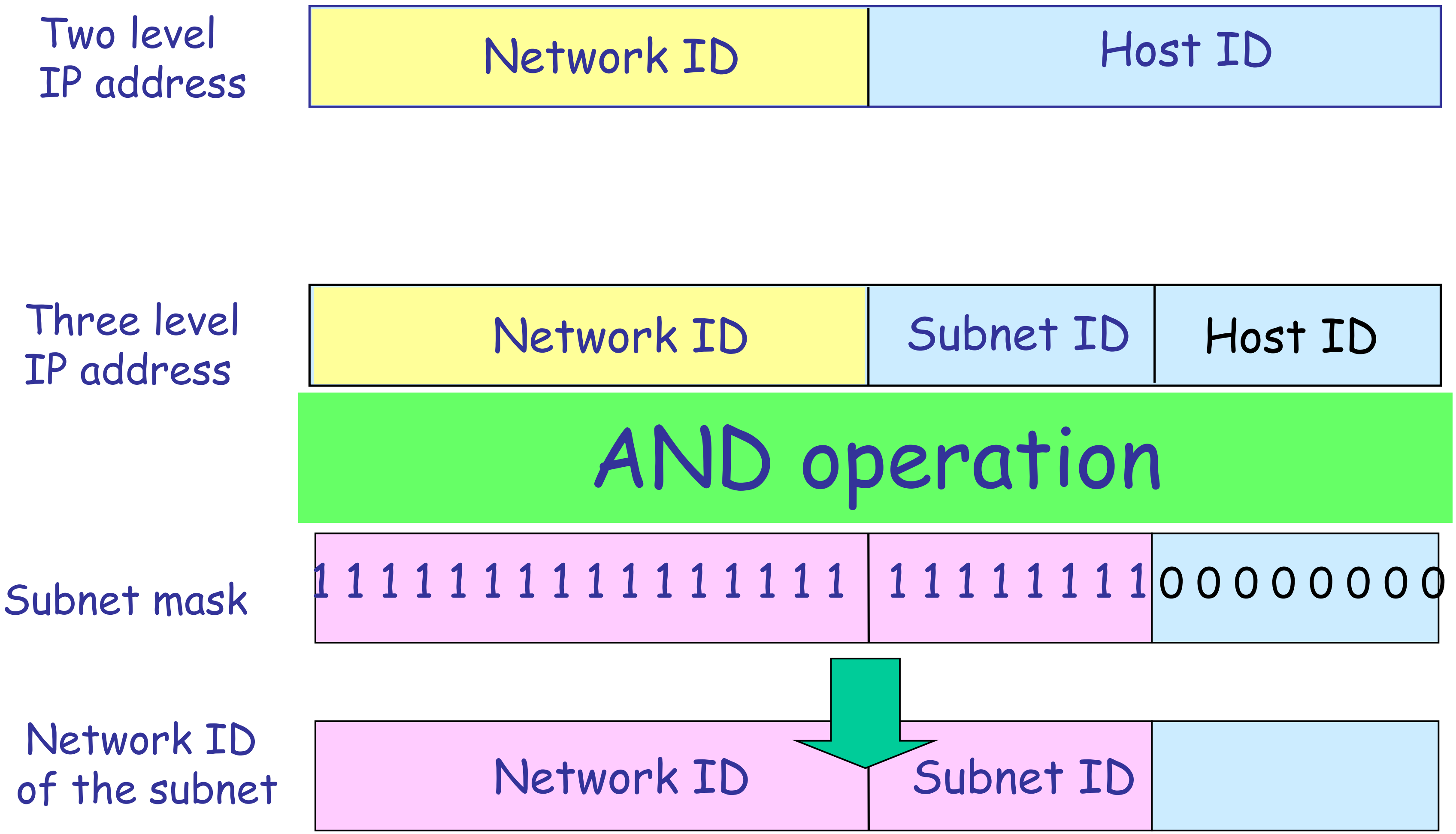
A class B network without subnet: 145.13.0.0



A network for others, the class B with 3 subnet



Subnet mask



Two level IP address
Network ID
Host ID
Three level IP address
Network ID
Subnet ID
Host ID
AND operation
Subnet mask
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0
Network ID of the subnet
Network ID
Subnet ID

IP addressing: CIDR

- CIDR: Classless InterDomain Routing
  - Two level address: only two parts
  - network portion of address of arbitrary length
  - address format: a.b.c.d/x, where x is # bits in network portion of address
  - Network portion is often called prefix

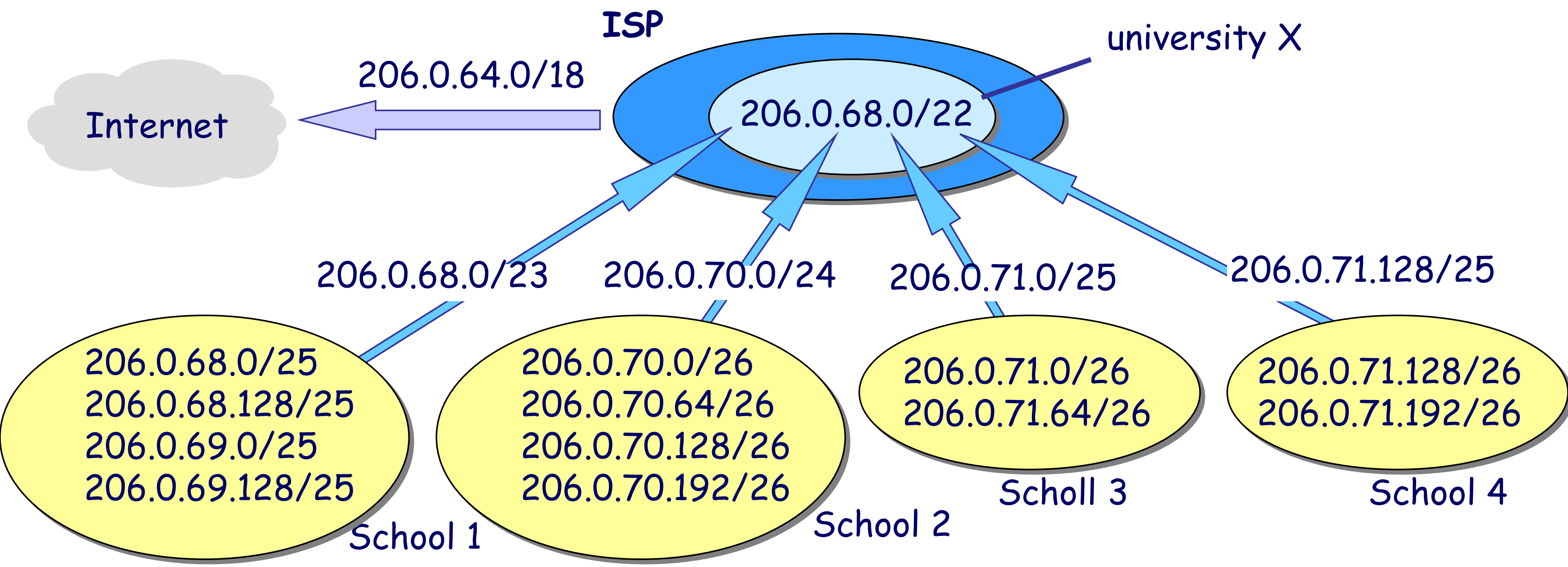
network part

host part

11001000 00010111 00010000 00000000

200.23.16.0/23

CIDR example



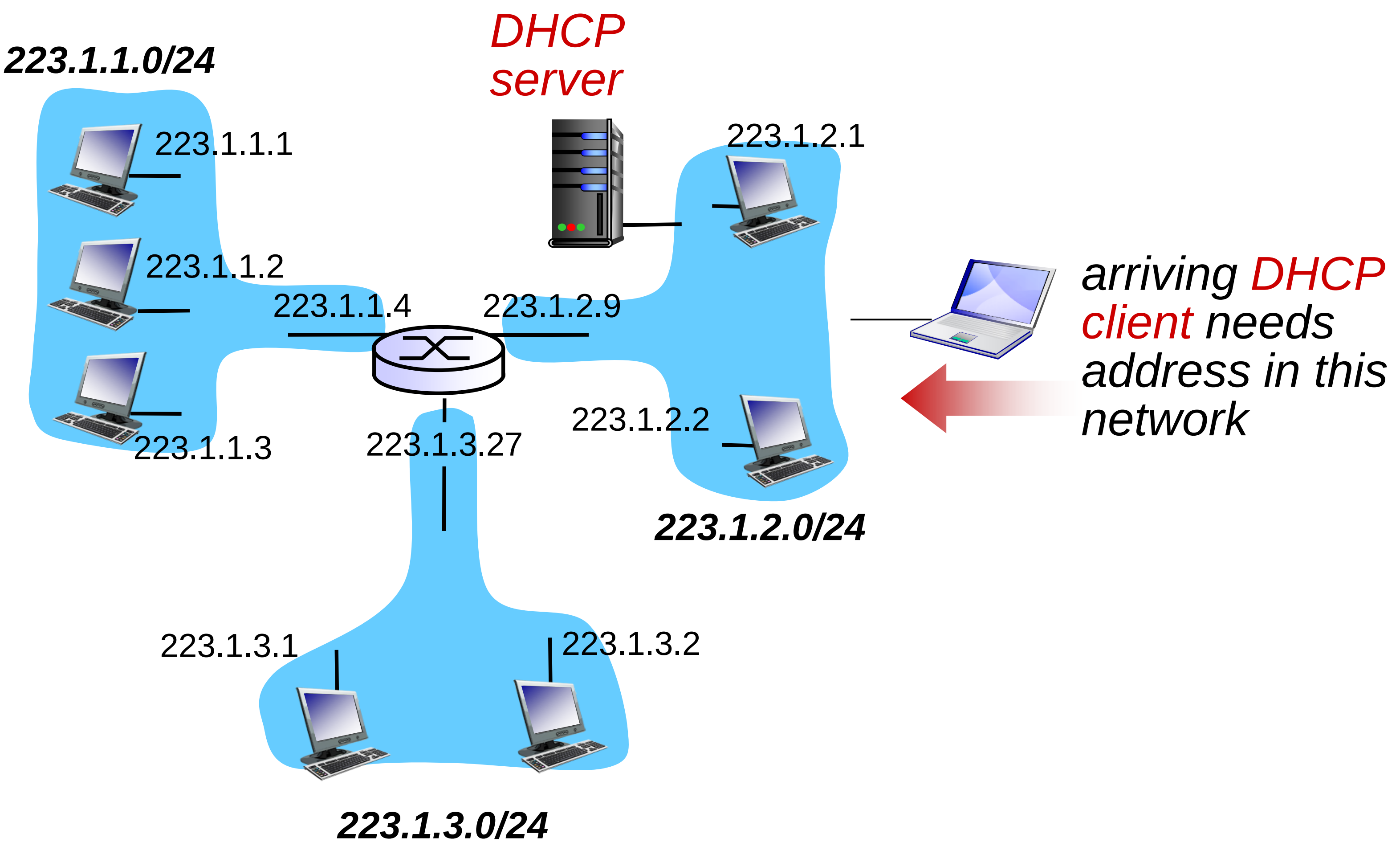
organizatio	address block	binary	number
ISP	206.0.64.0/18	11001110.00000000.01*	16384
Uni. X	206.0.68.0/22	11001110.00000000.010001*	1024
sch. 1	206.0.68.0/23	11001110.00000000.0100010*	512
Sch. 2	206.0.70.0/24	11001110.00000000.01000110.*	256
Sch.3	206.0.71.0/25	11001110.00000000.01000111.0*	128
Sch. 4	206.0.71.128/25	11001110.00000000.01000111.1*	128

IP addresses: how to get one?

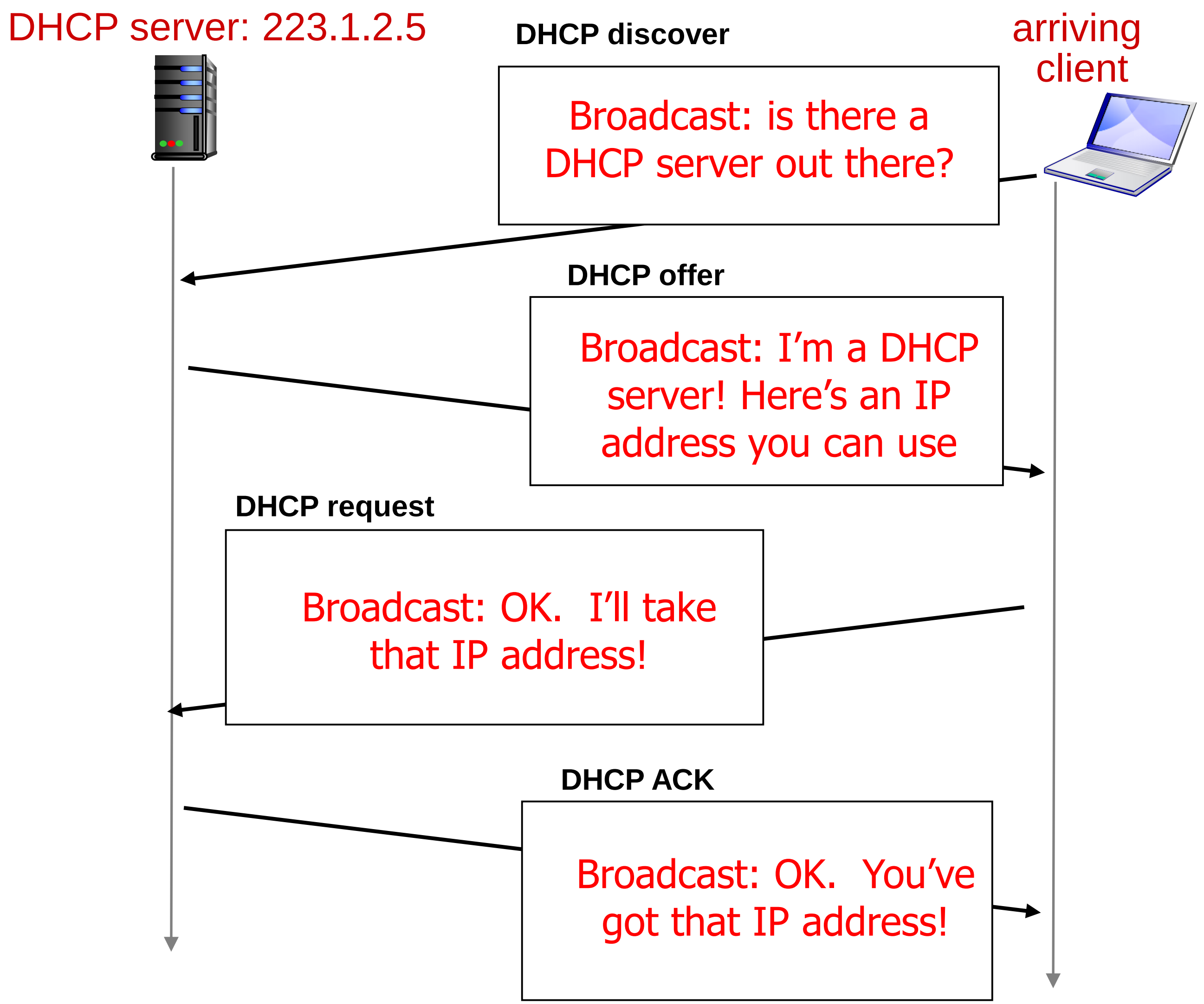
Hosts:

- hard-coded by system admin in a file
- DHCP (Dynamic Host Configuration Protocol): allows host to dynamically obtain its IP address from network server when it joins network
  - "plug-and-play"
  - can renew its lease on address in use
  - allows reuse of addresses (only hold address while connected/"on")
  - support for mobile users who want to join network

DHCP client-server scenario



DHCP client-server scenario



IP addresses: how to get one?

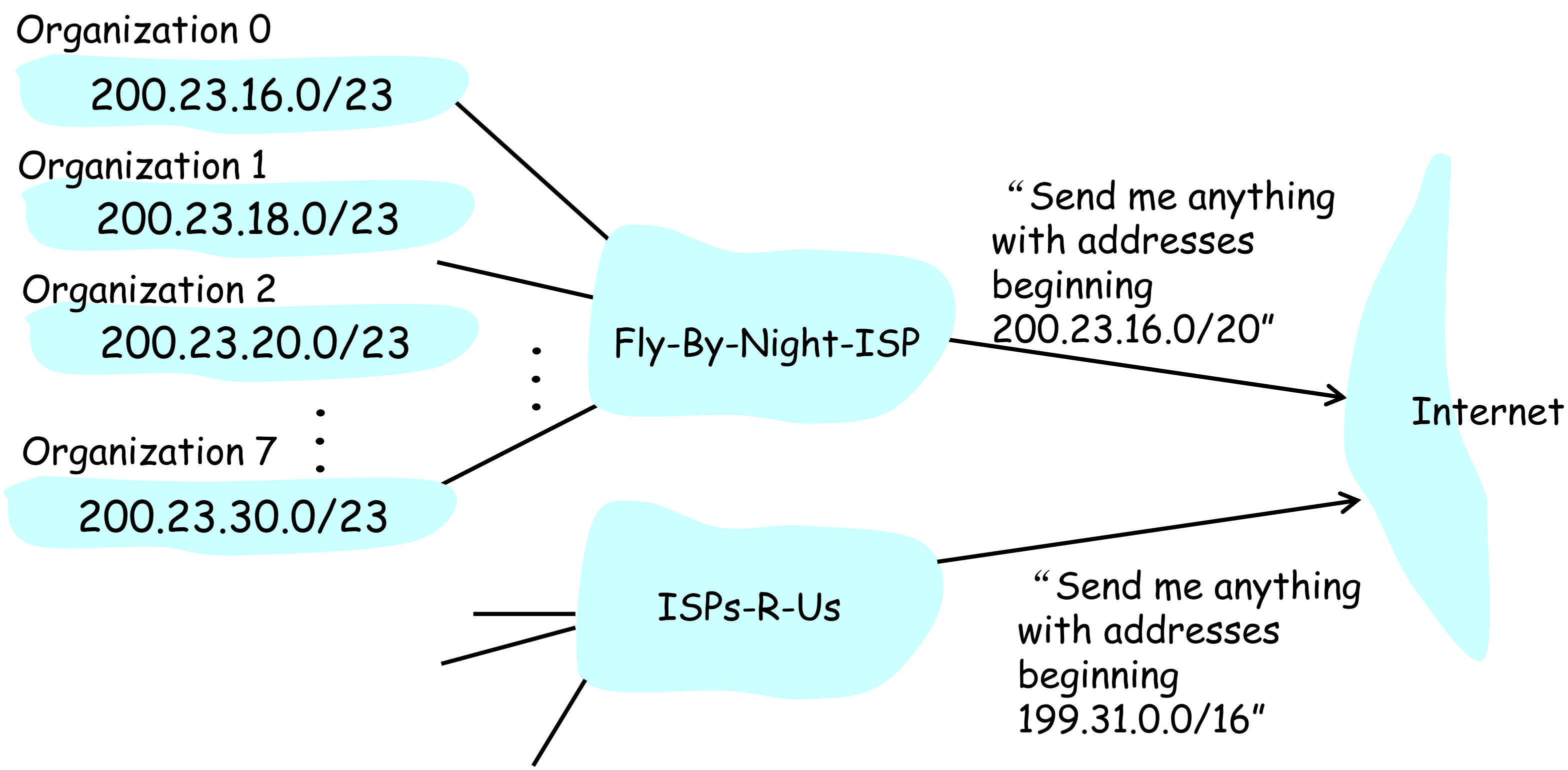
Network:

- get allocated portion of ISP's address space:

ISP's block	11001000 00010111 00010000 00000000		200.23.16.0/20
Organization 0	11001000 00010111 00010000 00000000		200.23.16.0/23
Organization 1	11001000 00010111 00010010 00000000		200.23.18.0/23
Organization 2	11001000 00010111 00010100 00000000		200.23.20.0/23
...	.....	....	....
Organization 7	11001000 00010111 00011110 00000000		200.23.30.0/23

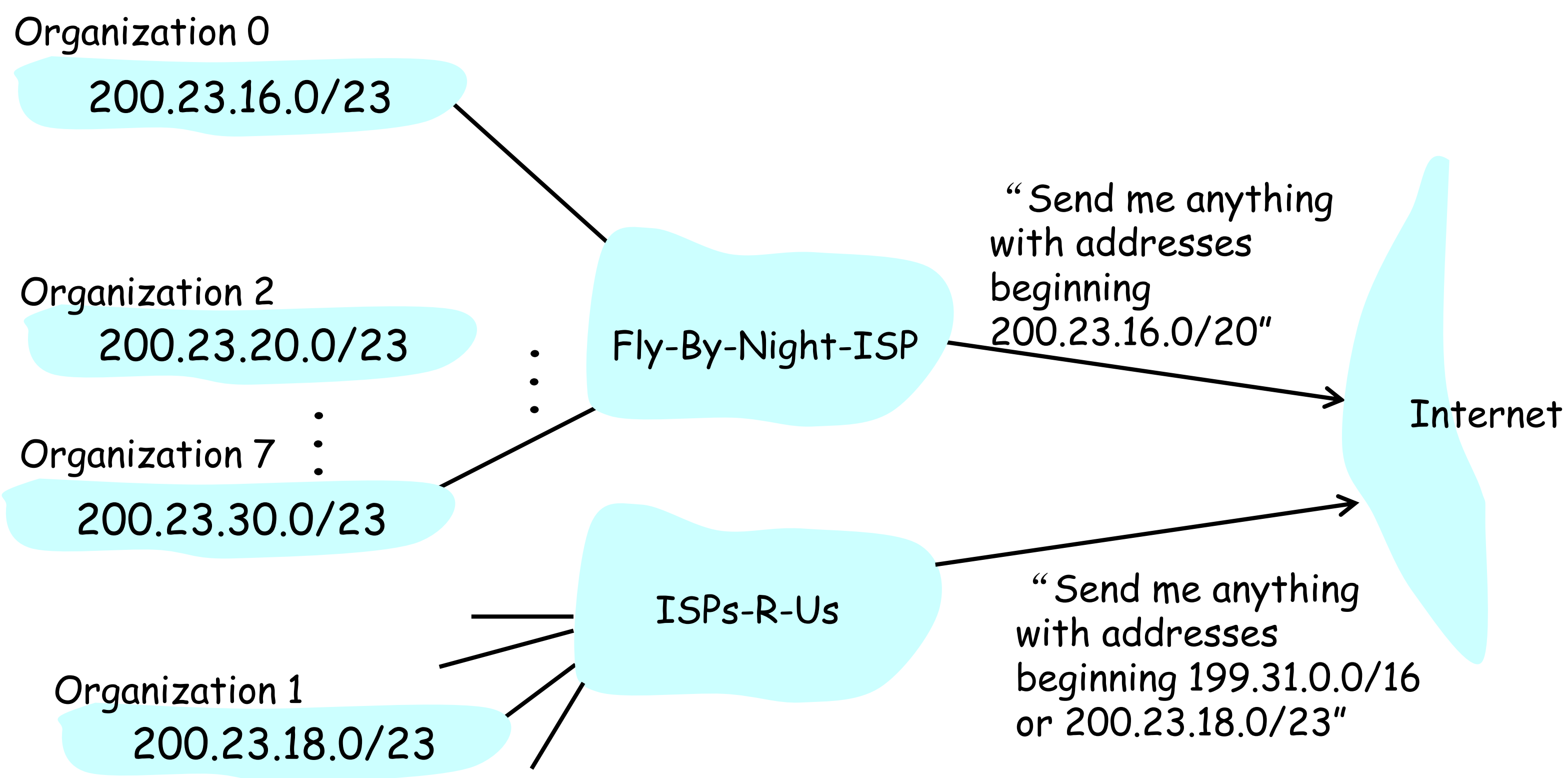
Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:



Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1



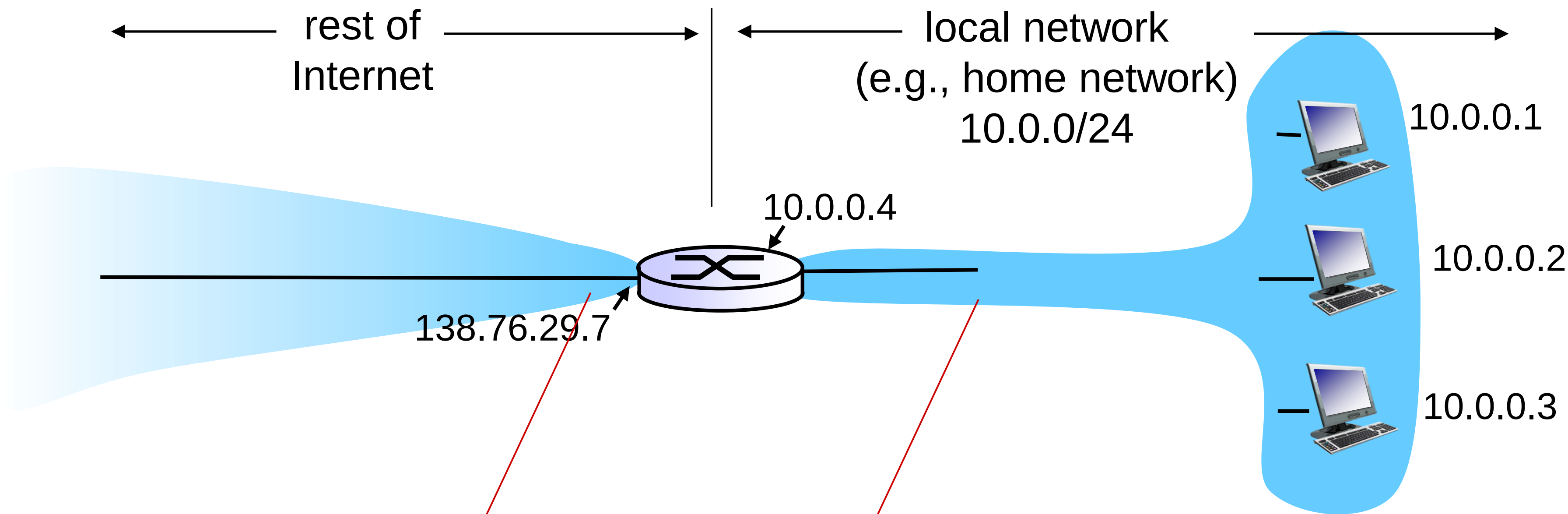
IP addressing: the last word...

Q: How does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers

- allocates addresses
- manages DNS
- assigns domain names, resolves disputes

NAT: Network Address Translation



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

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: Network Address Translation

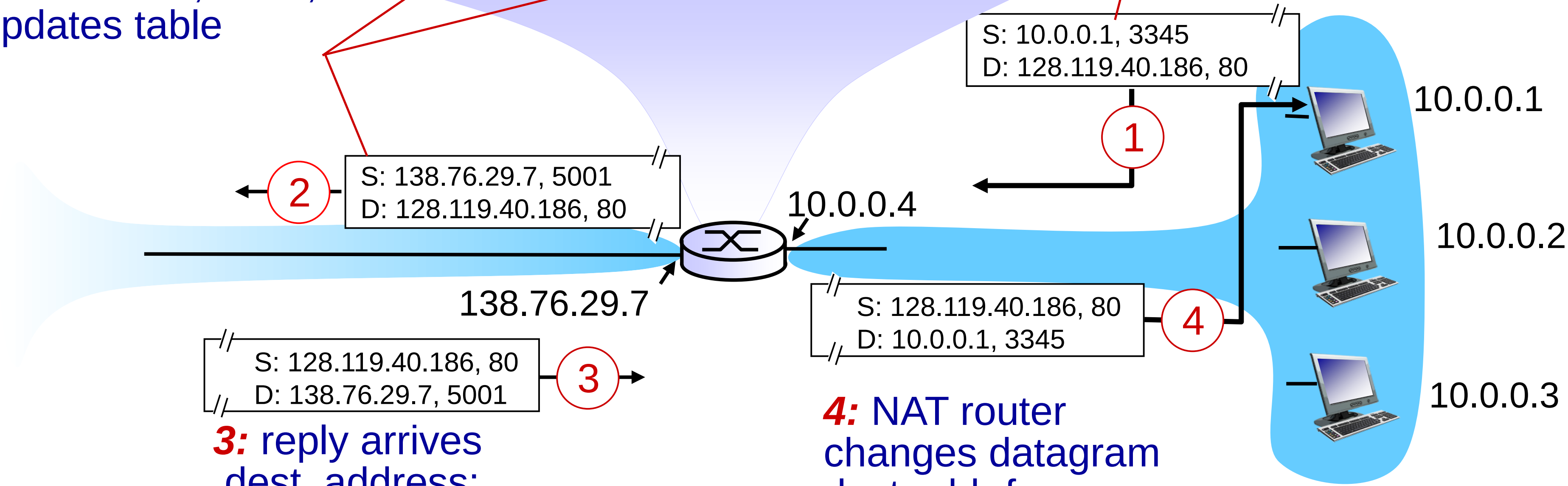
- Motivation: local network uses just one IP address as far as outside world is concerned
- Implementation: NAT router must:
  - Outgoing datagrams: replace (source IP address, port #) to (NAT IP address, new port #)
  - Remember (in NAT translation table) every translation pair
  - Incoming datagrams: replace (NAT IP address, new port #) with correspongding (source IP address, port #) stored in NAT table

NAT: Network Address Translation

NAT translation table	
WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
……	……

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

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

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

NAT: Network Address Translation

- Outside node cannot initiate the communication

- Reserved addresses:
  - 10.0.0.0 – 10.255.255.255
  - 172.16.0.0 – 172.31.255.255
  - 192.168.0.0 – 192.168.255.255

NAT: network address translation

- 16-bit port-number field:
  - 60,000+ simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - routers should only process up to layer 3
  - violates end-to-end argument
    - NAT possibility must be taken into account by app designers, e.g., P2P applications
  - address shortage should be solved by IPv6
  - NAT traversal: what if client wants to connect to server behind NAT?

LAN addresses and ARP

32-bit IP address:

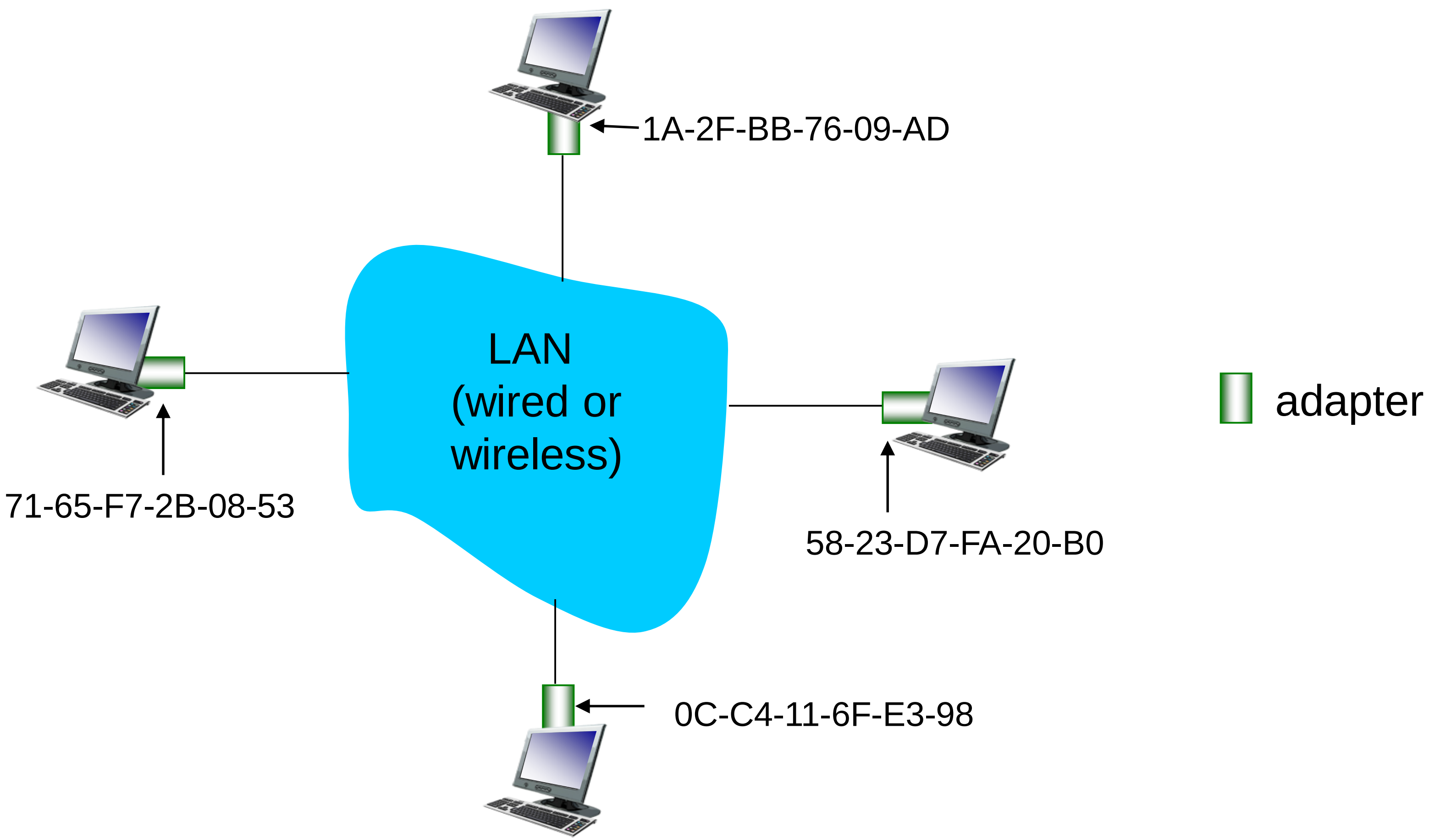
- *network-layer* address
- used to get datagram to destination network (recall IP network definition)

LAN (or MAC or physical) address:

- used to get datagram from one interface to another physically-connected interface (same network)
- 48 bit MAC address (for most LANs) burned in the adapter ROM
- e.g.: 1A-2F-BB-76-09-AD

LAN addresses and ARP

Each adapter on LAN has unique LAN address



LAN Address (more)

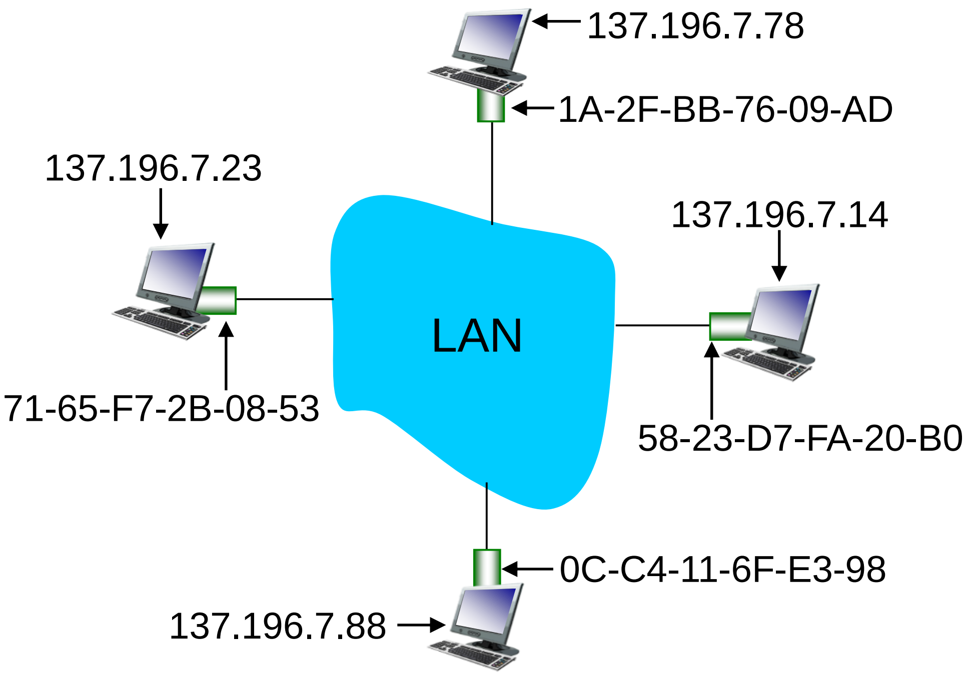
- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:

  (a) MAC address: like Social Security Number

  (b) IP address: like postal address
- MAC flat address => portability
  - can move LAN card from one LAN to another
- IP hierarchical address NOT portable
  - depends on IP subnet to which node is attached

ARP: Address Resolution Protocol

Question: how to determine MAC address of B given B's IP address?



- Each IP node (Host, Router) on LAN has ARP module, table
- ARP Table: IP/MAC address mappings for some LAN nodes

  < IP address; MAC address; TTL>

  < .... >

  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

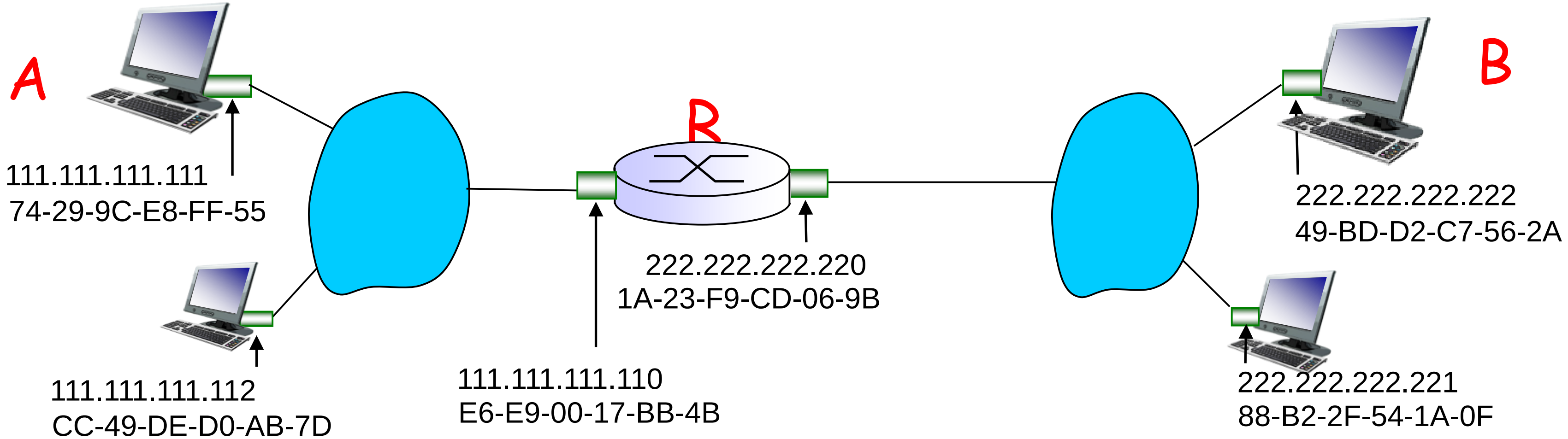
ARP protocol: same LAN

- A wants to send datagram to B
  - B's MAC address not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - destination MAC address = FF-FF-FF-FF-FF-FF
  - all nodes on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast)
- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ARP is "plug-and-play":
  - nodes create their ARP tables *without intervention from net administrator*

Routing to another LAN

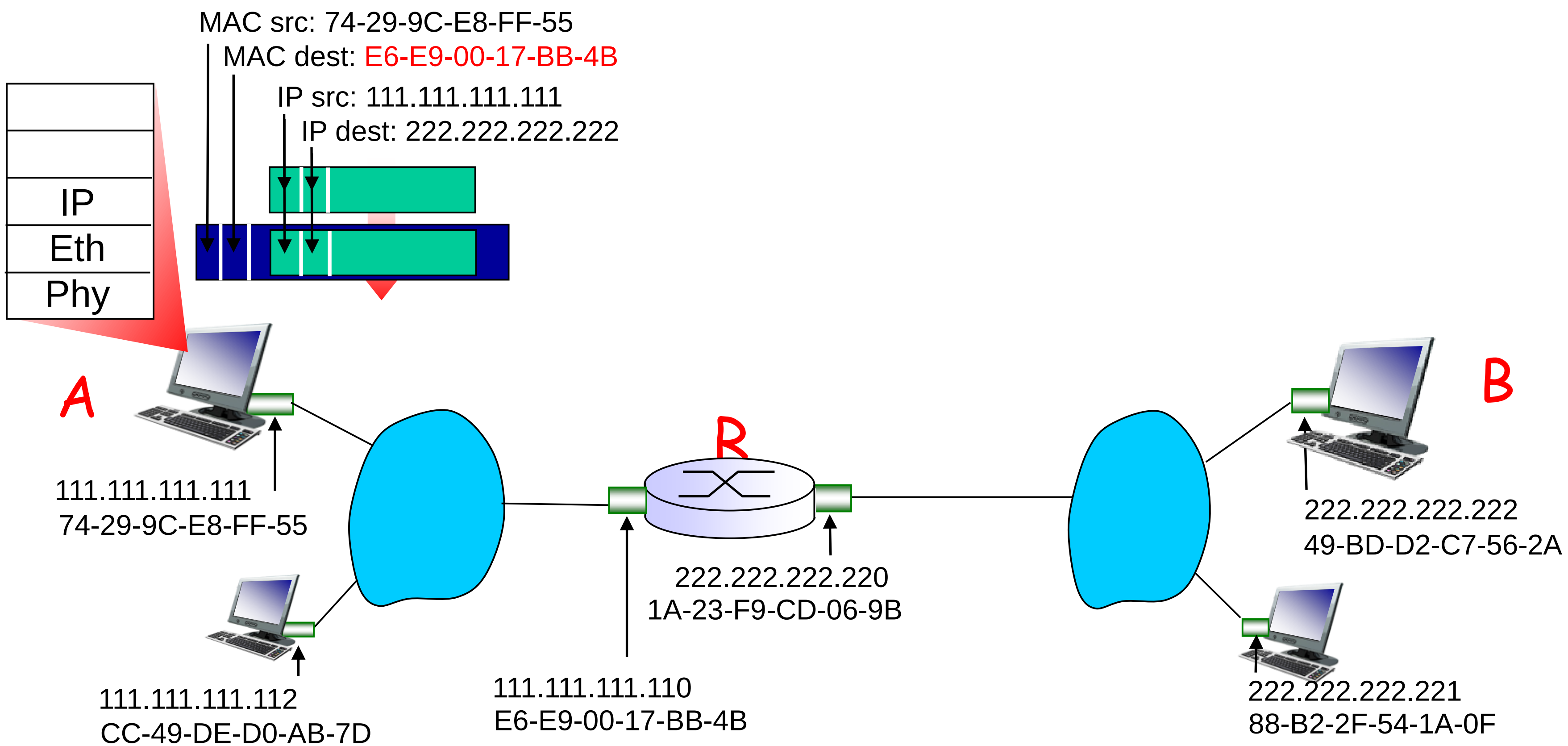
walkthrough: send datagram from A to B via R

- focus on addressing – at IP (datagram) and MAC layer (frame)
- assume A knows B's IP address
- assume A knows IP address of first hop router, R (how?)
- assume A knows R's MAC address (how?)



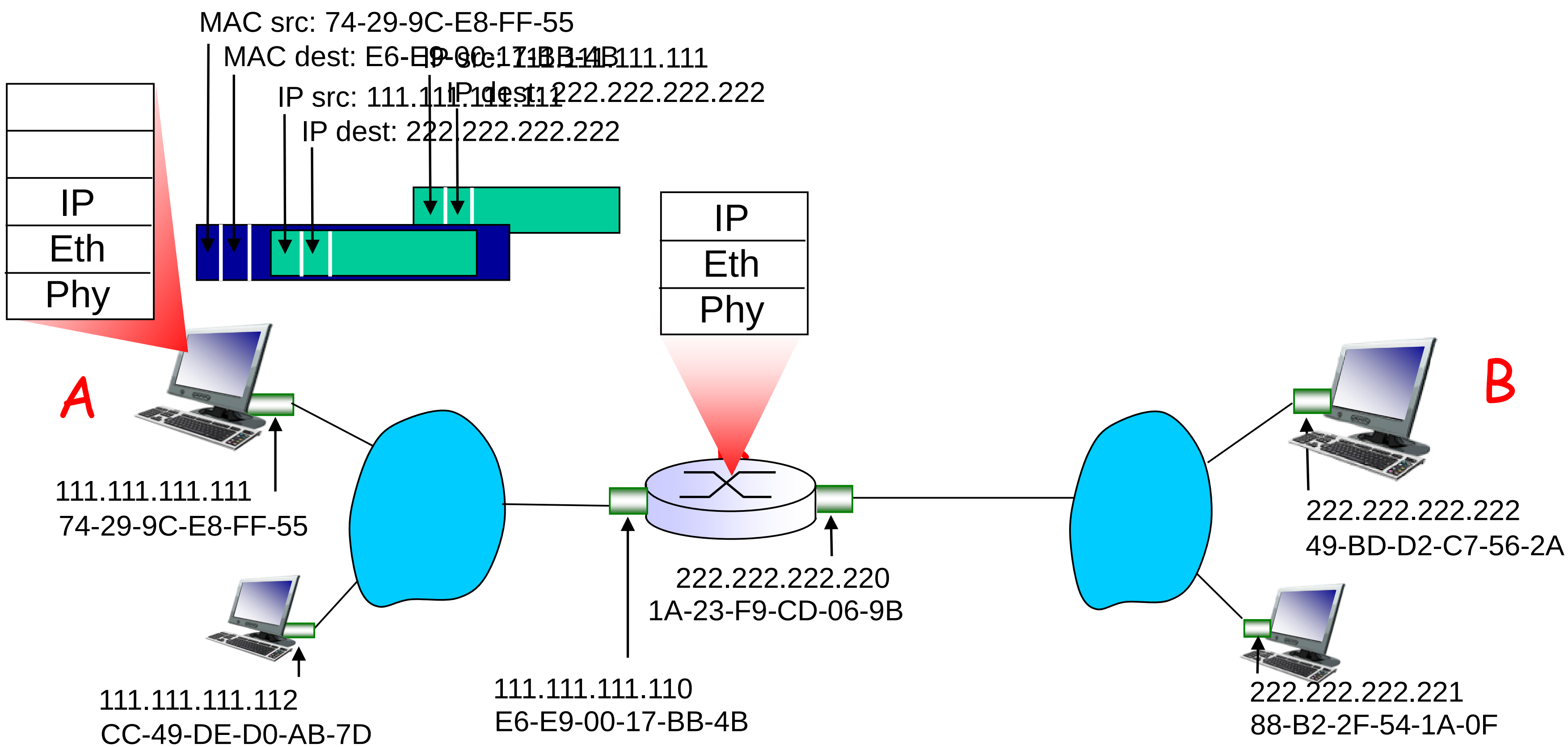
Routing to another LAN

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as destination address, frame contains A-to-B IP datagram



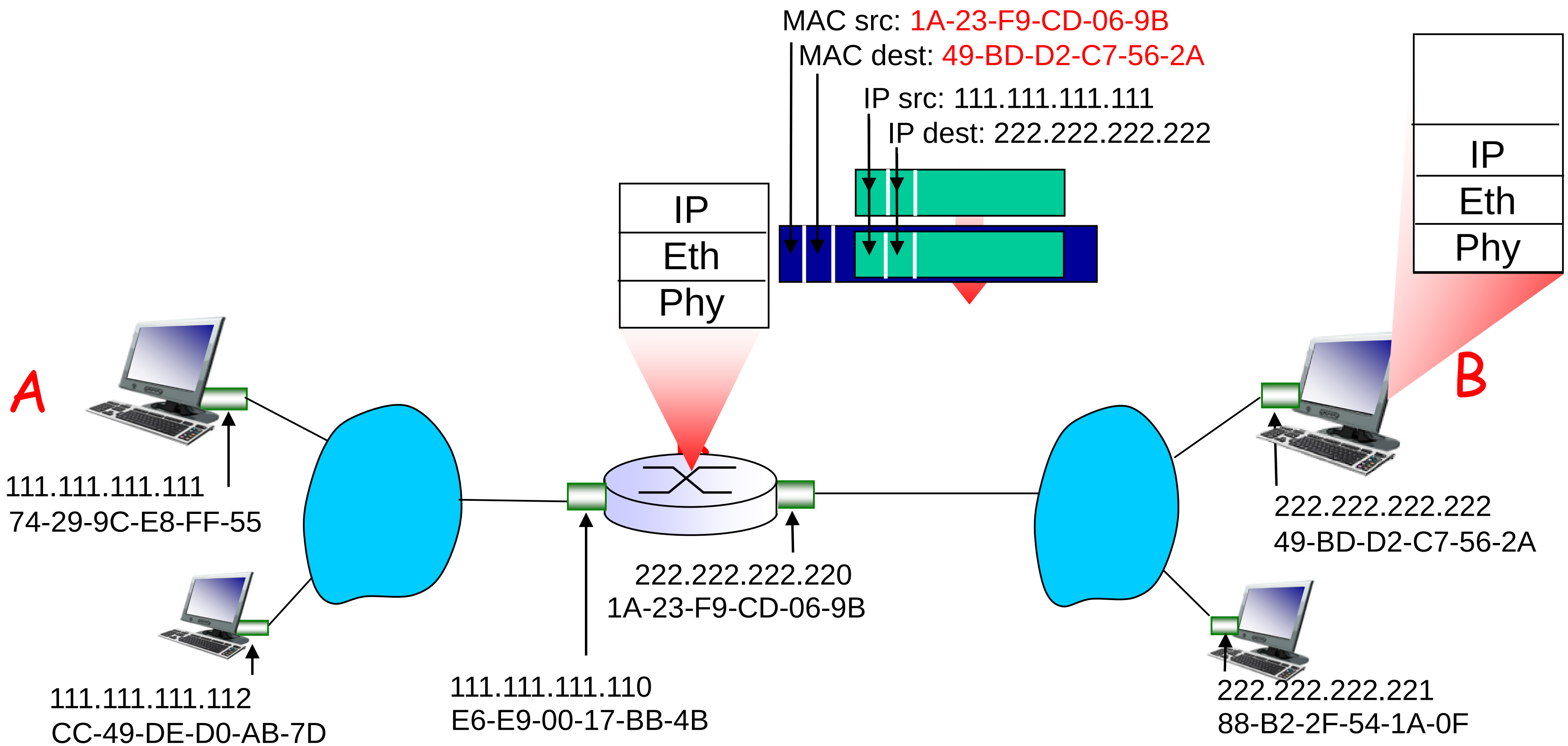
Routing to another LAN

- frame sent from A to R
- frame received at R, datagram removed, passed up to IP



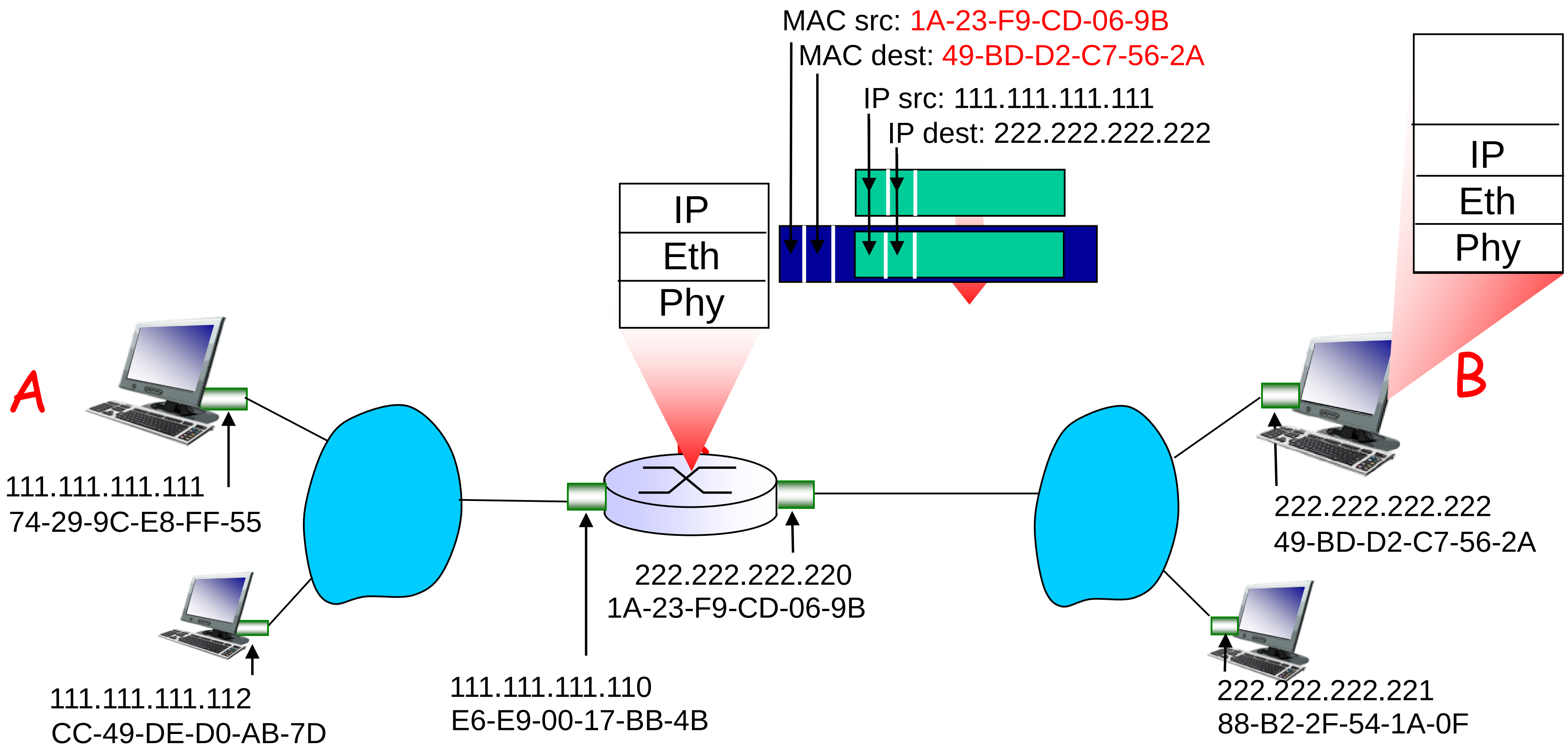
Routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as destination address, frame contains A-to-B IP datagram



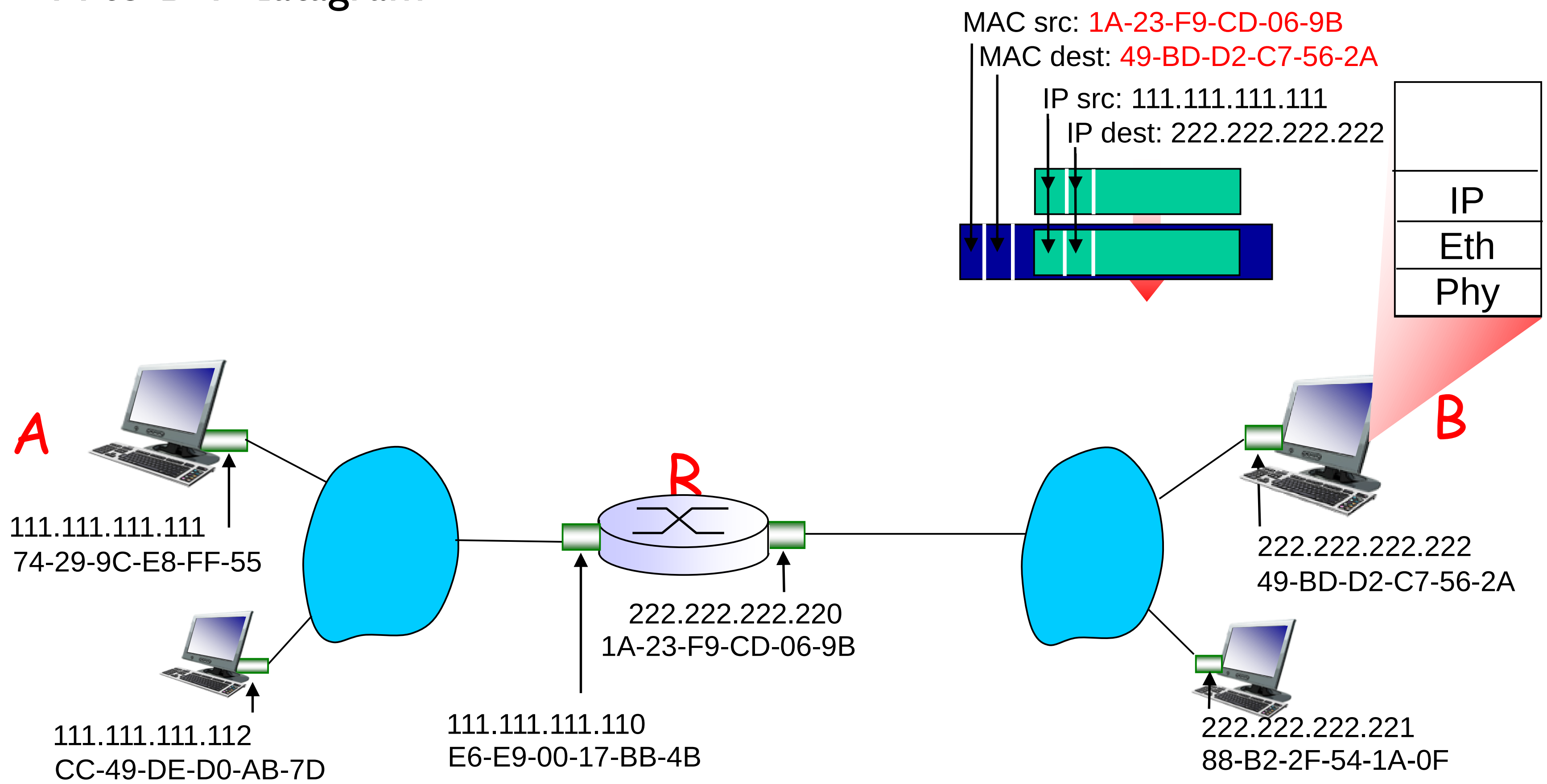
Routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as destination address, frame contains A-to-B IP datagram



Routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

ICMP: Internet Control Message Protocol

- used by hosts, routers, gateways to communication network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

ICMP:brief summary

- ICMP is the control sibling of IP
- ICMP is used by IP and uses IP as network layer protocol
- ICMP is used for ping, traceroute, and path MTU discovery
  - Ping: uses ICMP Echo request/reply messages
  - Path MTU discovery
    - Send a large IP datagram with "No fragment" bit set
    - Reduce size until success (No ICMP message received)

ping

```
C:\Documents and Settings\XXR>ping mail.sina.com.cn

Pinging mail.sina.com.cn [202.108.43.230] with 32 bytes of data:

Reply from 202.108.43.230: bytes=32 time=368ms TTL=242
Reply from 202.108.43.230: bytes=32 time=374ms TTL=242
Request timed out.
Reply from 202.108.43.230: bytes=32 time=374ms TTL=242

Ping statistics for 202.108.43.230:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 368ms, Maximum = 374ms, Average = 372ms
```

Traceroute and ICMP

- Source sends series of UDP segments to dest.
  - First has TTL=1
  - Second has TTL=2, etc.
  - Unlikely port number
- When nth datagram arrives to nth router:
  - Router discards datagram
  - And sends to source an ICMP message (type 11, code 0)
  - Message includes name of router & IP address
- When ICMP message arrives, source calculates RTT
- Traceroute does this 3 times
- Stopping criterion
- UDP segment eventually arrives at destination host
- Destination returns ICMP "port unreachable" packet (type 3, code 3)
- When source gets this ICMP, stops

traceroute

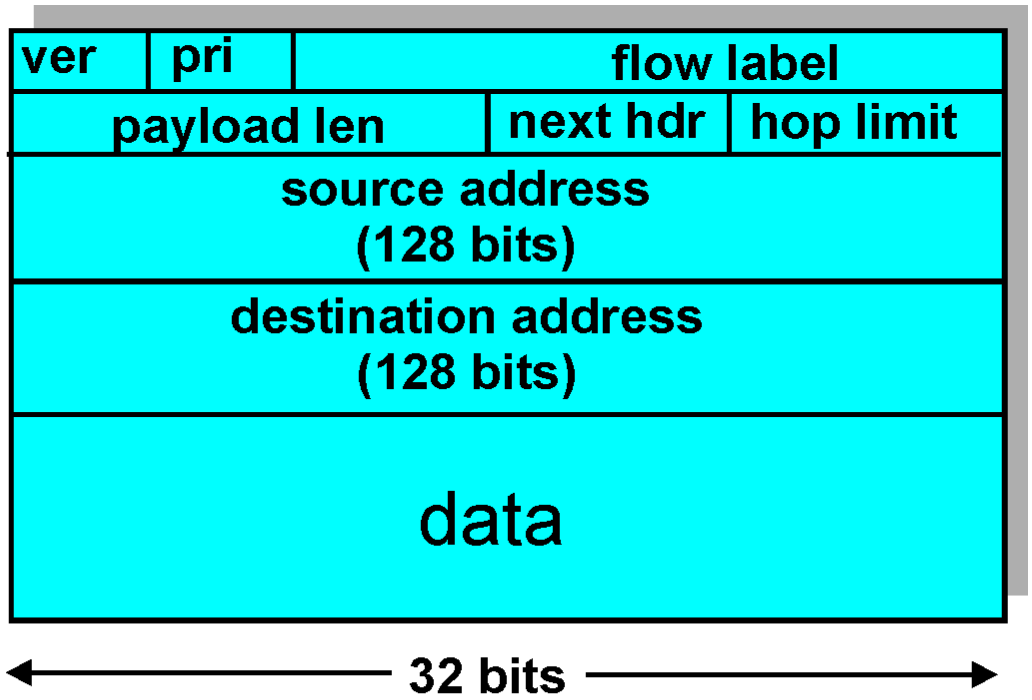
```
C:\Documents and Settings\XXR>tracert mail.sina.com.cn

Tracing route to mail.sina.com.cn [202.108.43.230]
over a maximum of 30 hops:

  1    24 ms    24 ms    23 ms  222.95.172.1
  2    23 ms    24 ms    22 ms  221.231.204.129
  3    23 ms    22 ms    23 ms  221.231.206.9
  4    24 ms    23 ms    24 ms  202.97.27.37
  5    22 ms    23 ms    24 ms  202.97.41.226
  6    28 ms    28 ms    28 ms  202.97.35.25
  7    50 ms    50 ms    51 ms  202.97.36.86
  8   308 ms   311 ms   310 ms  219.158.32.1
  9   307 ms   305 ms   305 ms  219.158.13.17
 10   164 ms   164 ms   165 ms  202.96.12.154
 11   322 ms   320 ms  2988 ms  61.135.148.50
 12   321 ms   322 ms   320 ms  freemail43-230.sina.com [202.108.43.230]

Trace complete.
```

IPv6

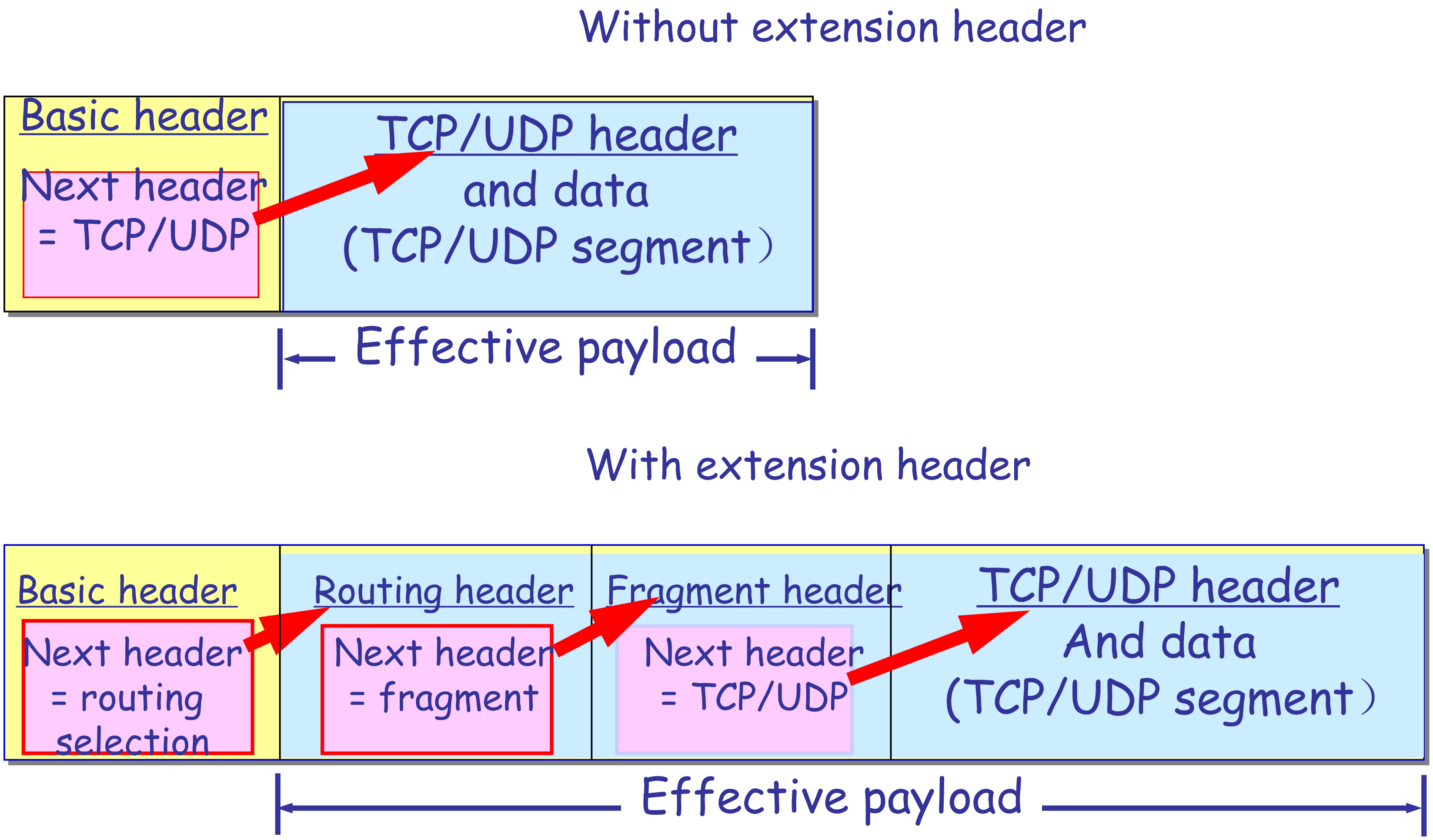
- Initial motivation: 32-bit address space soon to be completely allocated.
- Additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS
  - new "anycast" address: route to "best" of several replicated servers
- IPv6 datagram format:
  - fixed-length 40 byte header
  - no fragmentation allowed

IPv6 Header (Cont)

Priority: identify priority among datagrams in flow
Flow Label: identify datagrams in same "flow."
(concept of "flow" not well defined).
Next header: identify upper layer protocol for data



Next header



Without extension header
Basic header
Next header = TCP/UDP
TCP/UDP header
and data
(TCP/UDP segment)
Effective payload
With extension header
Basic header
Next header = routing selection
Routing header
Next header = fragment
Fragment header
Next header = TCP/UDP
TCP/UDP header
And data
(TCP/UDP segment)
Effective payload

Other Changes from IPv4

- *Checksum*: removed entirely to reduce processing time at each hop
- *Options:* allowed, but outside of header, indicated by "Next Header" field
- *ICMPv6:* new version of ICMP
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions

IPv6 address

- Three types: unicast, multicast, anycast
- Colon hexadecimal notation:

68E6:8C64:FFFF:FFFF:0:1180:960A:FFFF

- Zero compression:

FF05:0:0:0:0:0:0:B3 == FF05::B3 ;

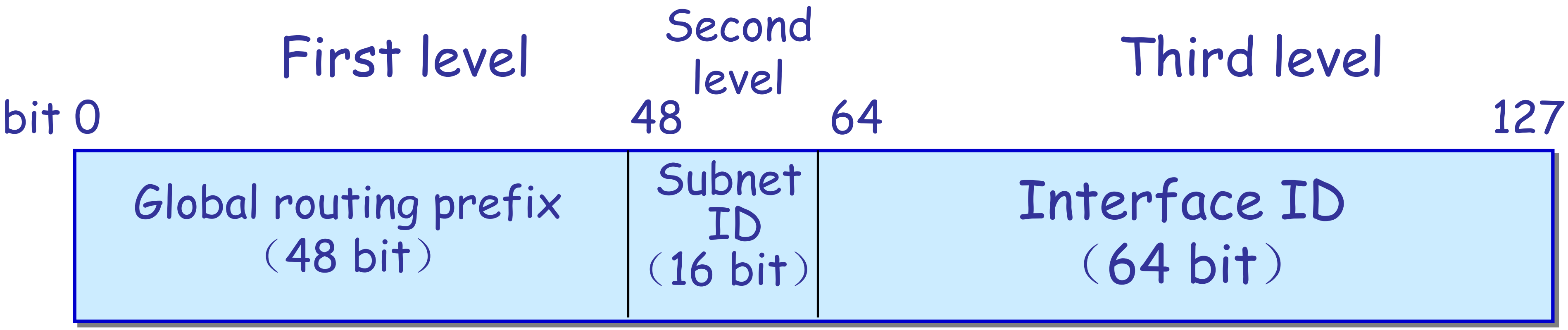
0:0:0:0:0:0:128.10.2.1 == ::128.10.2.1

12AB:0000:0000:CD30:0000:0000:0000:0000/60

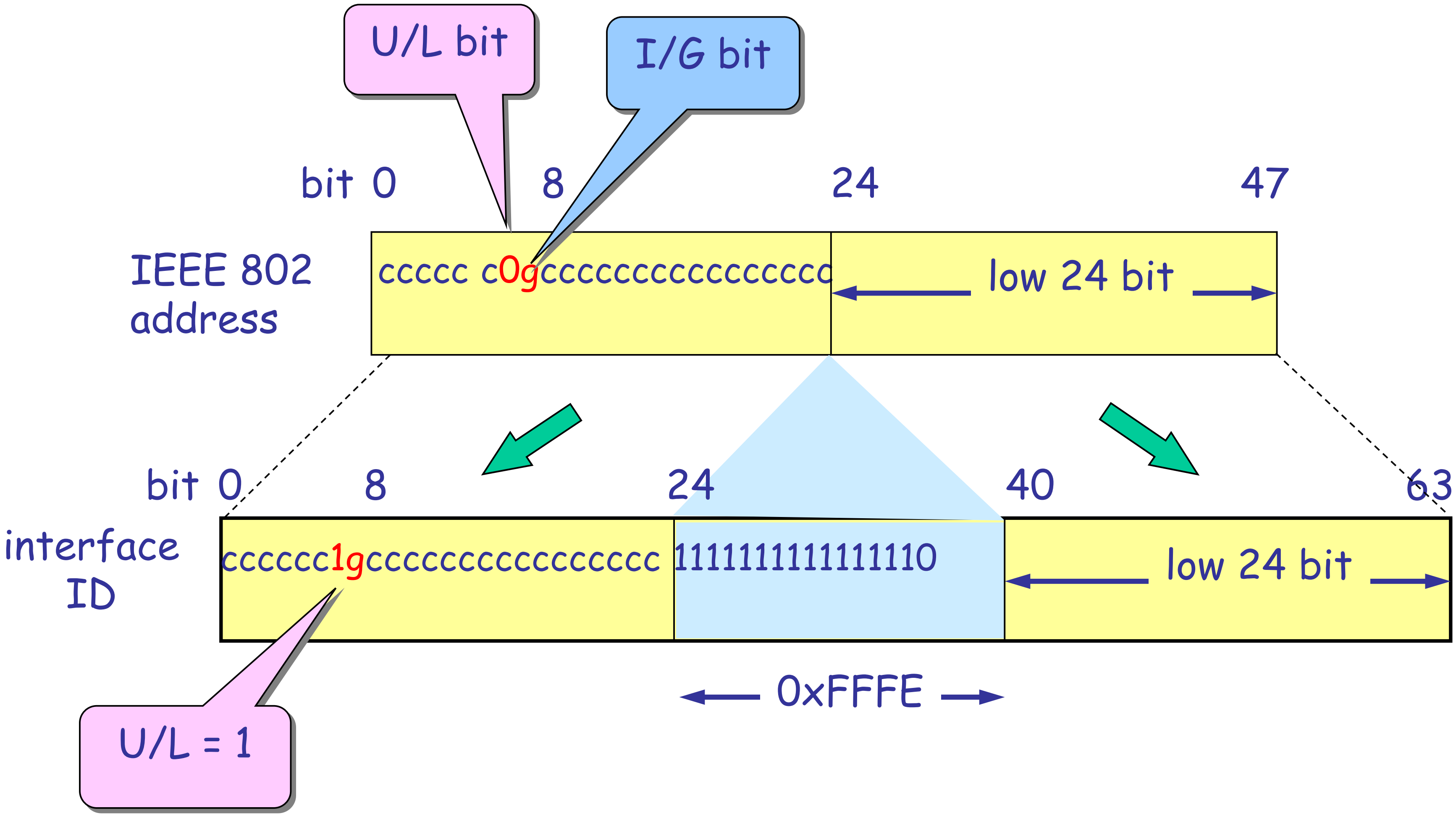
== 12AB::CD30:0:0:0:0/60

== 12AB:0:0:CD30::/60

Unicast address



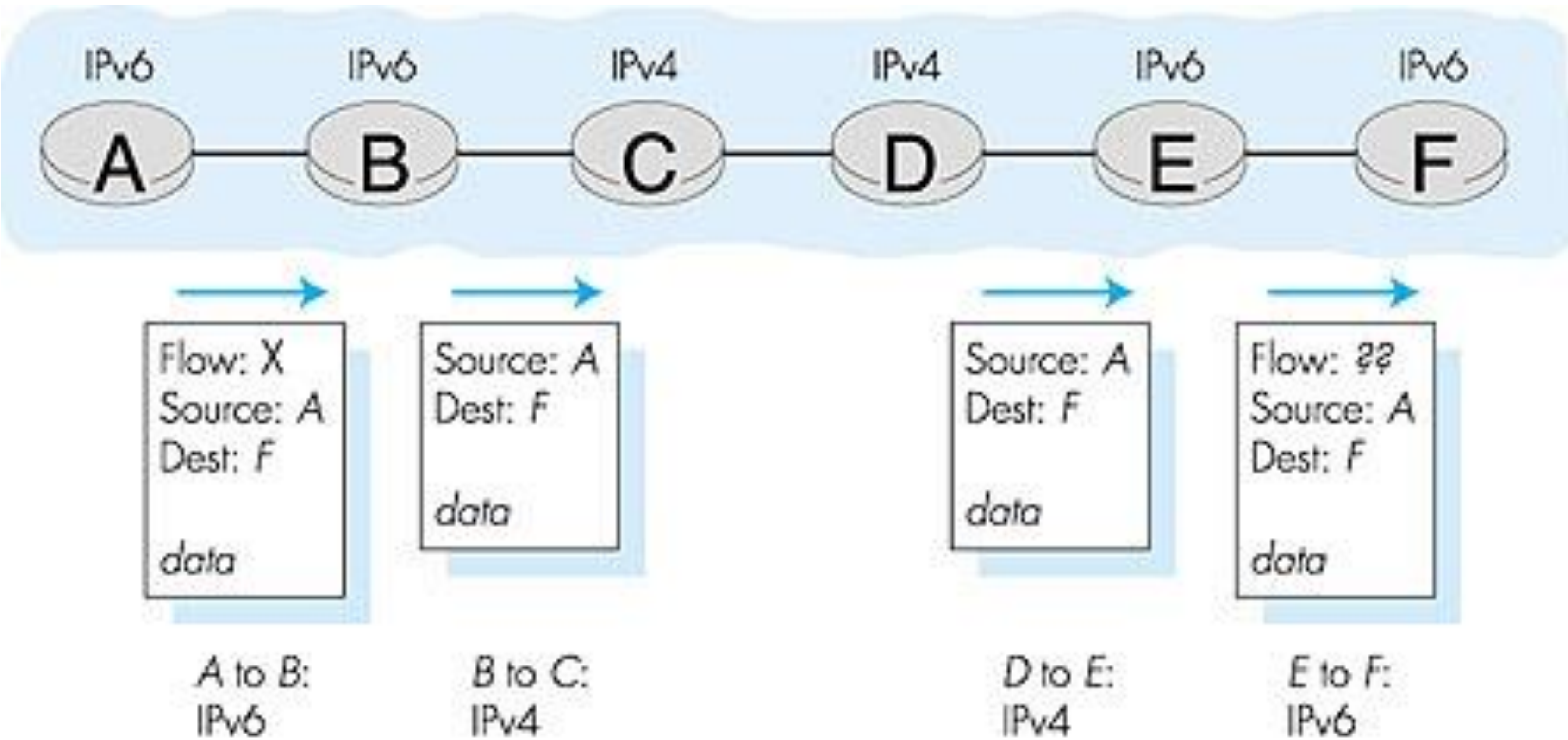
EUI-64


U/L bit
I/G bit
bit 0
8
24
47
IEEE 802 address
ccccc c0gcccccccccccccccc
low 24 bit
bit 0
8
24
40
63
interface ID
cccccc1gcccccccccccccccc
1111111111111110
low 24 bit
0xFFFE
U/L = 1

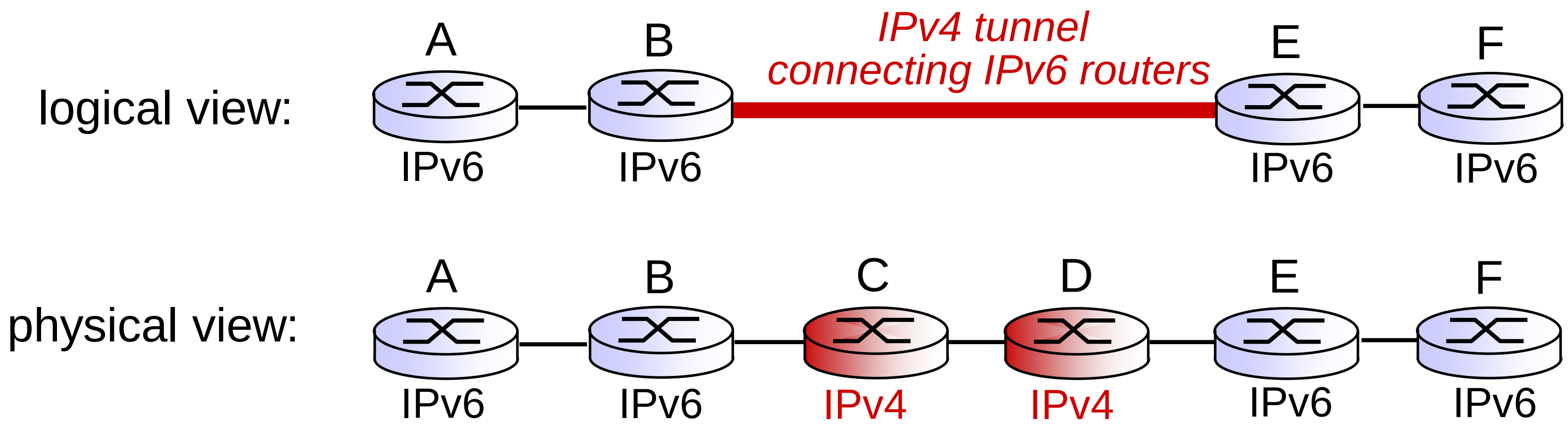
Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneous
  - no "flag days"
  - How will the network operate with mixed IPv4 and IPv6 routers?
- Two proposed approaches:
  - *Dual Stack*: some routers with dual stack (v6, v4) can "translate" between formats
  - *Tunneling:* IPv6 carried as payload n IPv4 datagram among IPv4 routers

Dual Stack Approach

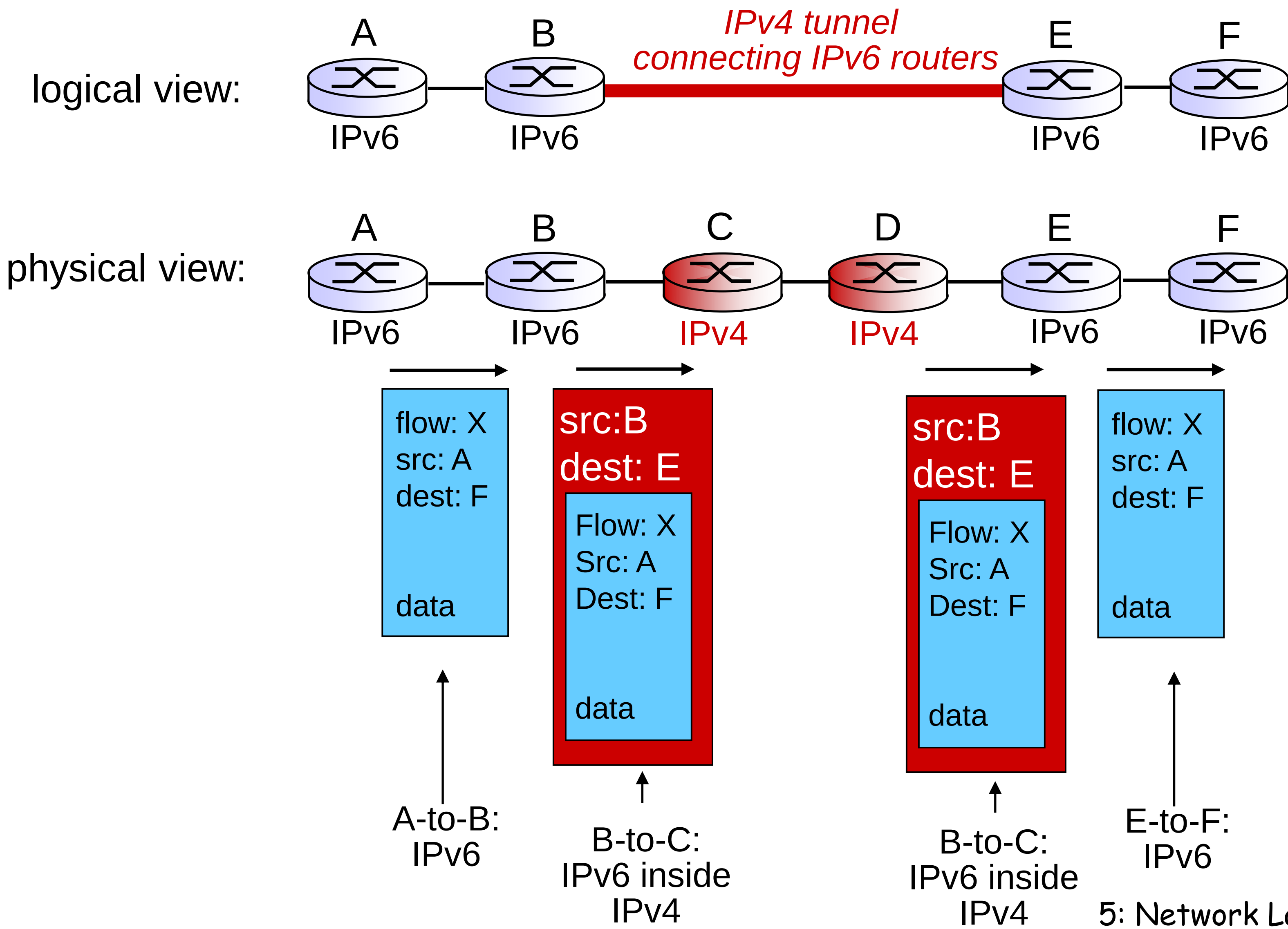


Tunneling



logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6
IPv6
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6

Tunneling



logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6
IPv6
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
flow: X
src: A
dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
flow: X
src: A
dest: F
data
A-to-B:
IPv6
B-to-C:
IPv6 inside
IPv4
B-to-C:
IPv6 inside
IPv4
E-to-F:
IPv6

IPv6: adoption

- Google: 8% of clients access services via IPv6
- NIST: 1/3 of all US government domains are IPv6 capable
- *Long (long!) time for deployment, use*
  - 20 years and counting!
  - think of application-layer changes in last 20 years: WWW, Facebook, streaming media, Skype, …
  - *Why?*

Summary

- Virtual circuit and datagram networks
- Routing algorithms:
  - Dijkstra's algorithm
  - Broadcast routing
  - Link state
  - Distance vector ("count to infinity" problem)
- Routing in the Internet (RIP, OSPF, BGP)
- IP: Internet Protocol
  - IPv4 Datagram format
  - IP fragment
  - IPv4 addressing
  - NAT
  - ARP
  - ICMP
  - IPv6
  - From IPv4 to IPv6