
# COMPUTER NETWORKS

**Ashwini M Joshi.**

Department of Computer Science and Engineering

**ashwinimjoshi@pes.edu**

## Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

4.5 Introduction to Network Layer Protocols

- DHCP
- ICMP

4.6 IPv6 Addressing

4.7 Introduction to Routing Algorithms

## Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

4.5 Introduction to Network Layer Protocols

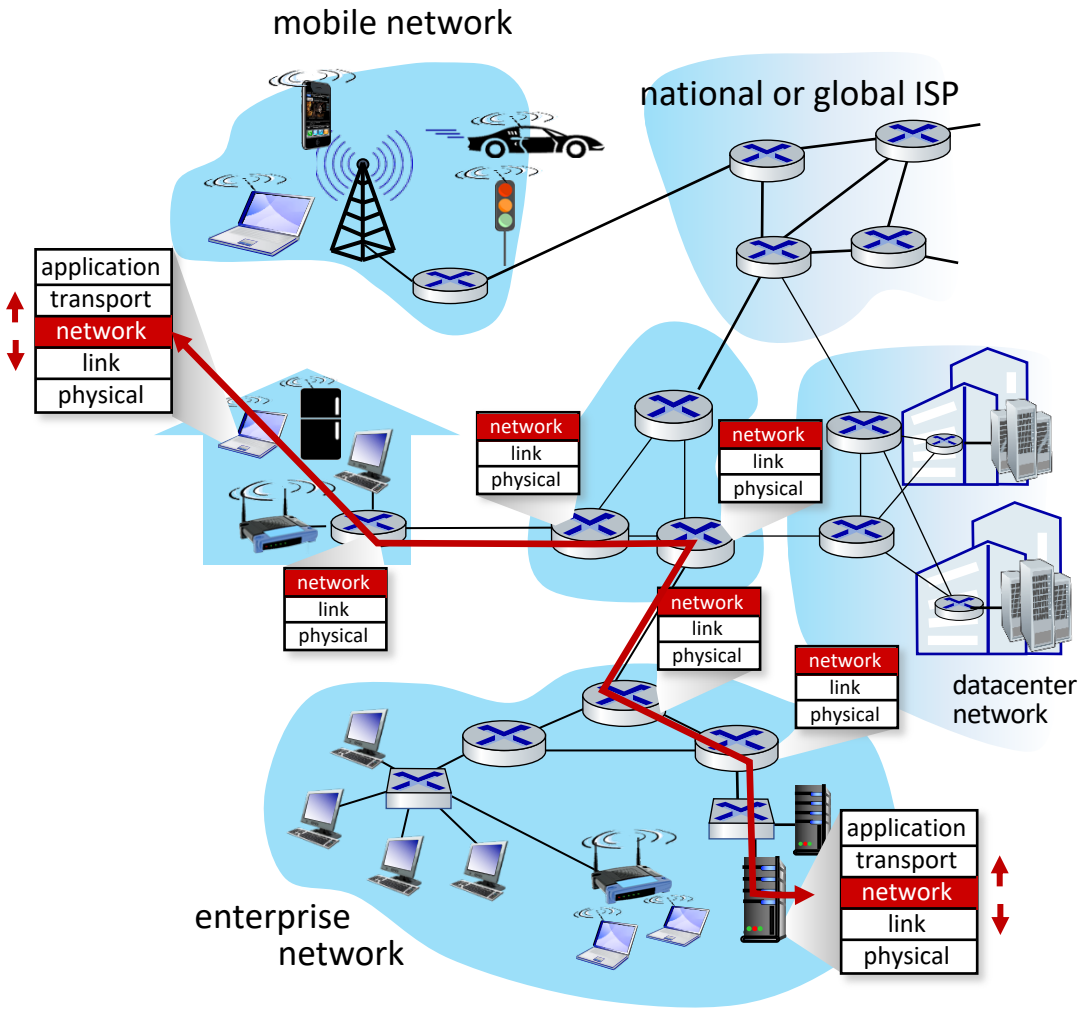
- DHCP
- ICMP

4.6 IPv6 Addressing

4.7 Introduction to Routing Algorithms

# Network-layer Services and Protocol

- transport segment from sending to receiving host
  - sender: encapsulates segments into datagrams, passes to link layer
  - receiver: delivers segments to transport layer protocol
- network layer protocols in *every Internet device*: hosts, routers
- routers:
  - examines header fields in all IP datagrams passing through it
  - moves datagrams from input ports to output ports to transfer datagrams along end-end path



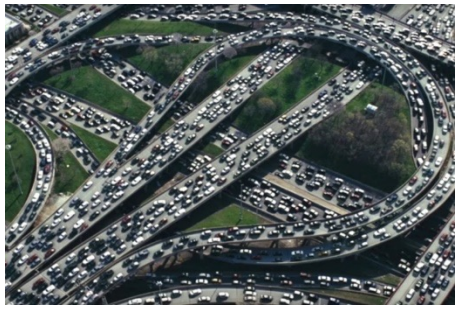
# Two Key Network-layer Functions

## network-layer functions:

- *forwarding:* move packets from a router's input link to appropriate router output link
- *routing:* determine route taken by packets from source to destination
  - *routing algorithms*

## analogy: taking a trip

- *forwarding:* process of getting through single interchange
- *routing:* process of planning trip from source to destination

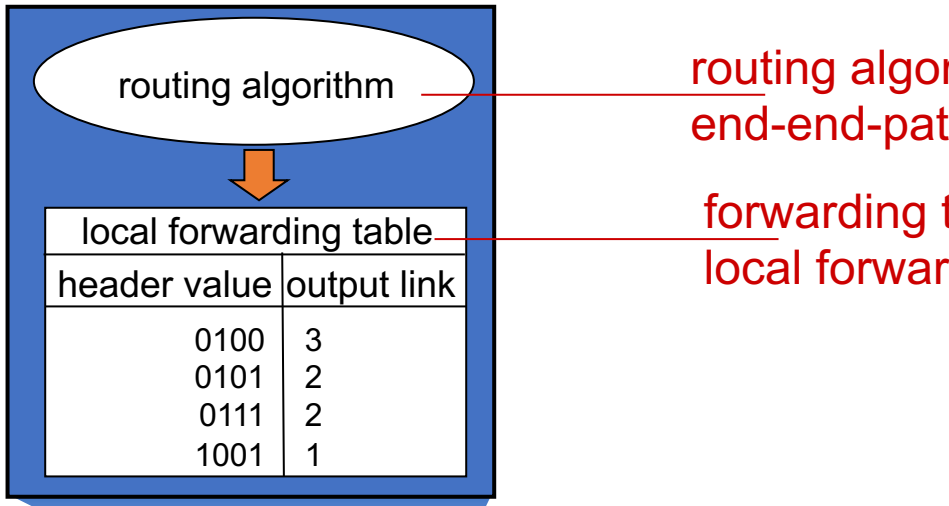


forwarding

routing

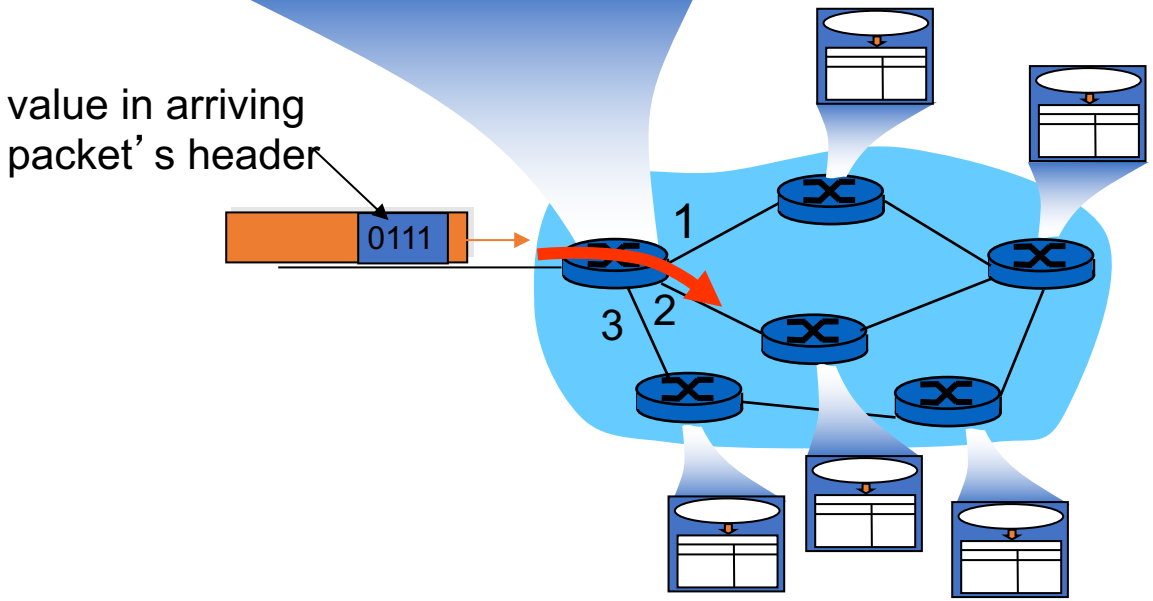
# COMPUTER NETWORKS

## Interplay between Routing and Forwarding




routing algorithm determines end-end-path through network

forwarding table determines local forwarding at this router



## Connection Setup

- 3rd important function in *some* network architectures:
  - ATM, frame relay, X.25
- before datagrams flow, two end hosts *and* intervening routers establish virtual connection
  - routers get involved
- network vs transport layer connection service:
  - *network:* between two hosts (may also involve intervening routers in case of VCs)
  - *transport:* between two processes

## Network Service Model

*Q:* What *service model* for "channel" transporting datagrams from sender to receiver?

example services for *individual* datagrams:

- guaranteed delivery
- guaranteed delivery with less than 40 msec delay

example services for a *flow* of datagrams:

- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet spacing

## Network-layer Service Model

| Network Architecture | Service Model | Quality of Service (QoS) Guarantees? | | | |
|---|---|---|---|---|---|
| | | Bandwidth | Loss | Order | Timing |
| Internet | best effort | none | no | no | no |

Internet "best effort" service model

*No* guarantees on:

i. successful datagram delivery to destination
ii. timing or order of delivery
iii. bandwidth available to end-end flow

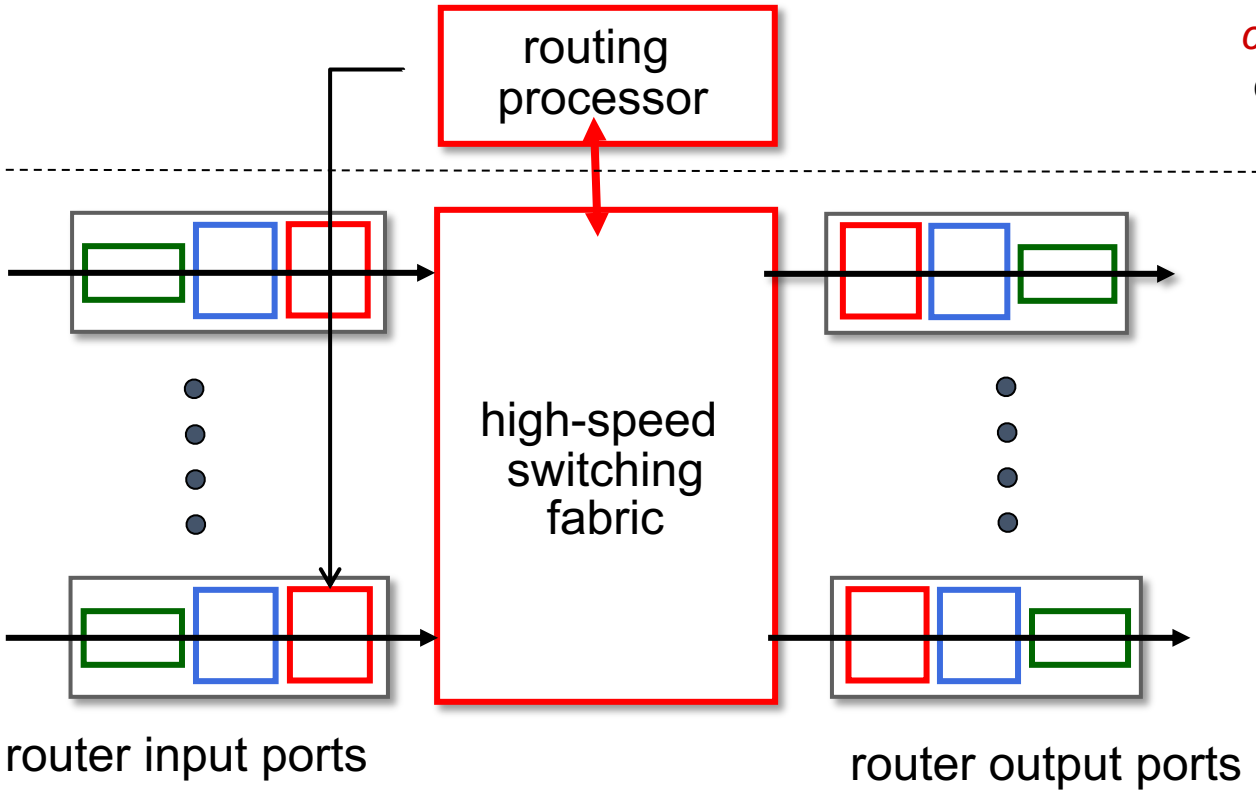
# Network-layer Service Model

| Network Architecture | Service Model | Quality of Service (QoS) Guarantees ? | | | |
|---|---|---|---|---|---|
| | | Bandwidth | Loss | Order | Timing |
| Internet | best effort | none | no | no | no |
| ATM | Constant Bit Rate | Constant rate | yes | yes | yes |
| ATM | Available Bit Rate | Guaranteed min | no | yes | no |
| Internet | Intserv Guaranteed (RFC 1633) | yes | yes | yes | yes |
| Internet | Diffserv (RFC 2475) | possible | possibly | possibly | no |

## Unit – 4 Network Layer and Internet Protocol

## Router Architecture Overview

# high-level view of generic router architecture:



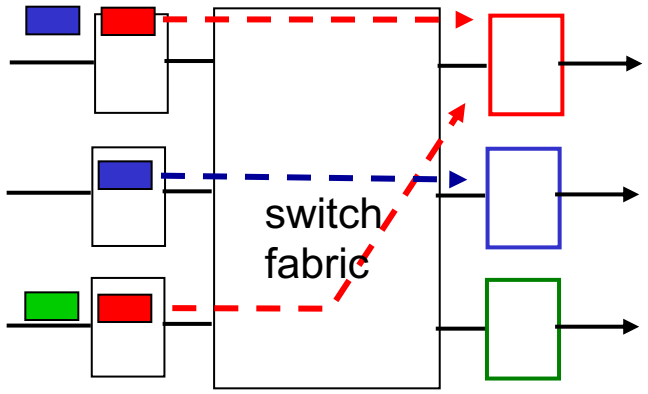

*routing, management control plane* (software) operates in millisecond time frame

*forwarding data plane* (hardware) operates in nanosecond timeframe

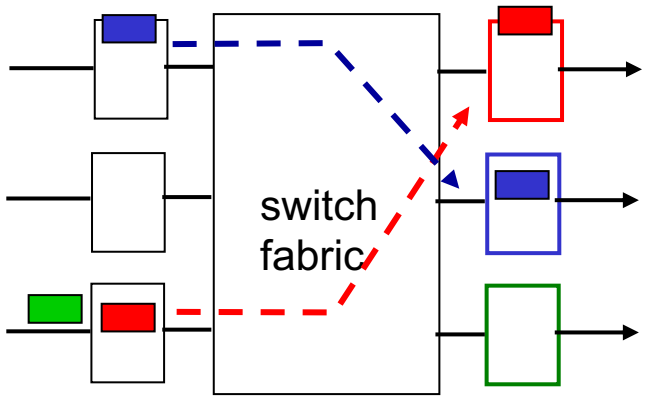
## Input port queuing

- If switch fabric slower than input ports combined -> queueing may occur at input queues
  - queueing delay and loss due to input buffer overflow!

- Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward



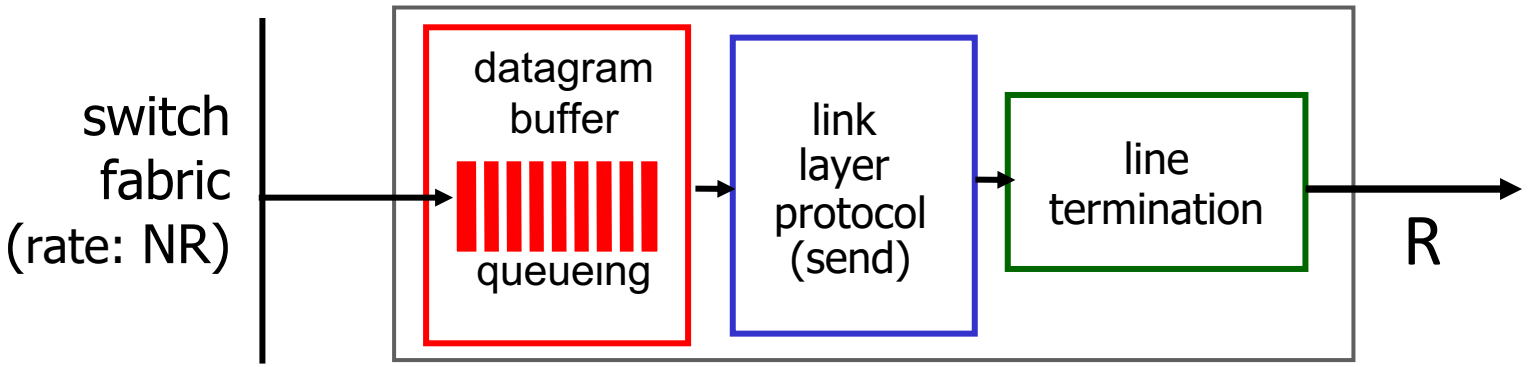

output port contention: only one red datagram can be transferred. lower red packet is *blocked*




one packet time later: green packet experiences HOL blocking

# Output port queuing

This is a really important slide



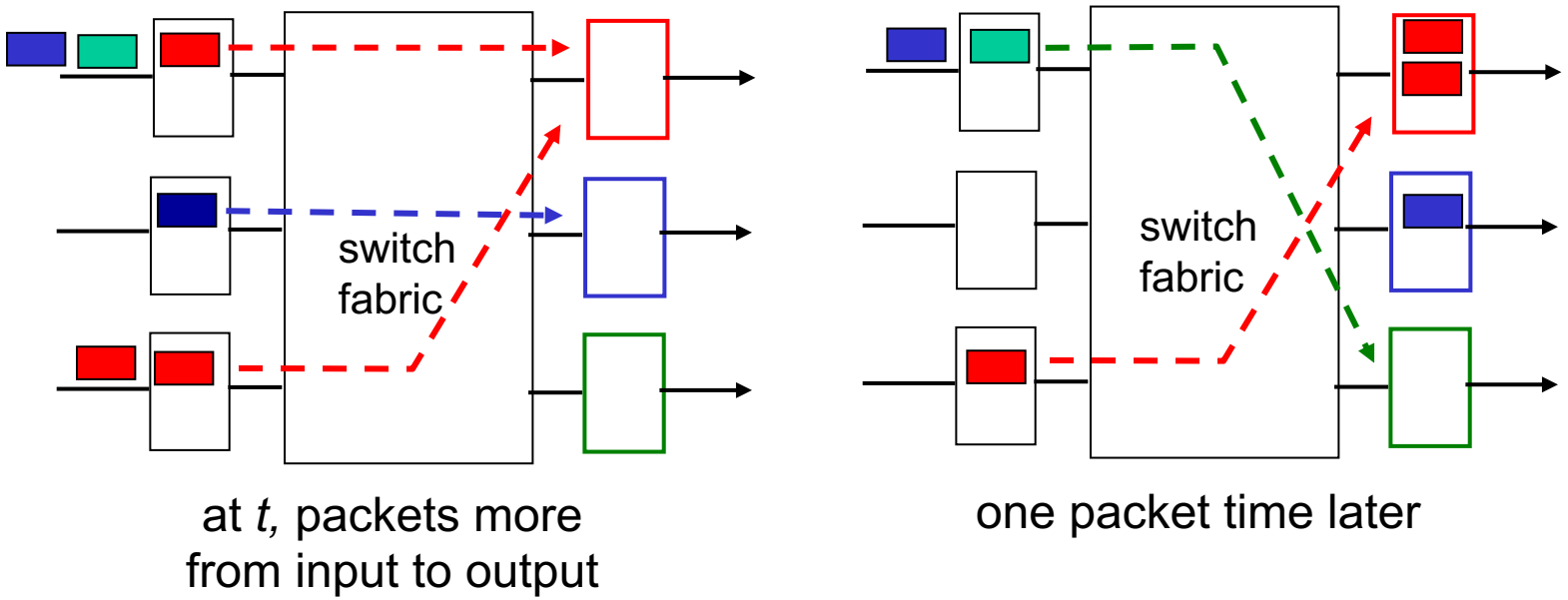

- *Buffering* required when datagrams arrive from fabric faster than link transmission rate. *Drop policy:* which datagrams to drop if no free buffers?

→ Datagrams can be lost due to congestion, lack of buffers

- *Scheduling discipline* chooses among queued datagrams for transmission

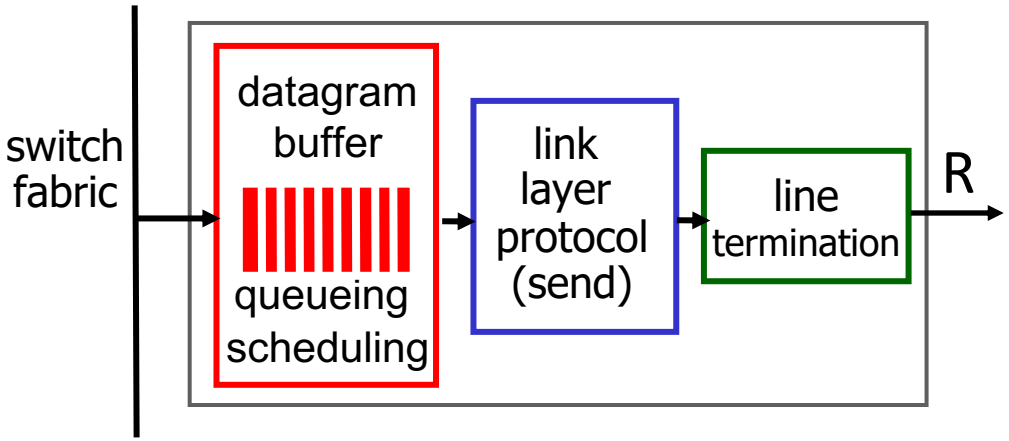
→ Priority scheduling – who gets best performance, network neutrality

# Output port queuing




- buffering when arrival rate via switch exceeds output line speed
- *queueing (delay) and loss due to output port buffer overflow!*

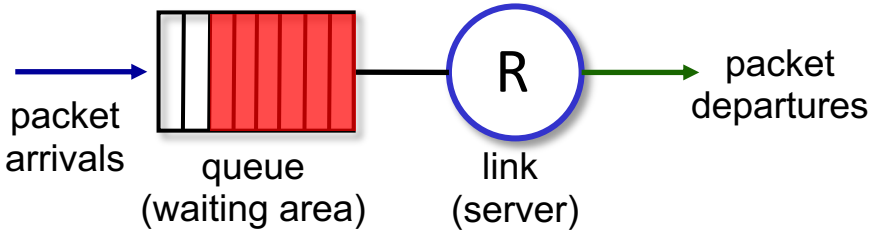
# Buffer Management




## buffer management:

- drop: which packet to add, drop when buffers are full
  - tail drop: drop arriving packet
  - priority: drop/remove on priority basis

## Abstraction: queue




- marking: which packets to mark to signal congestion (ECN, RED)

## Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

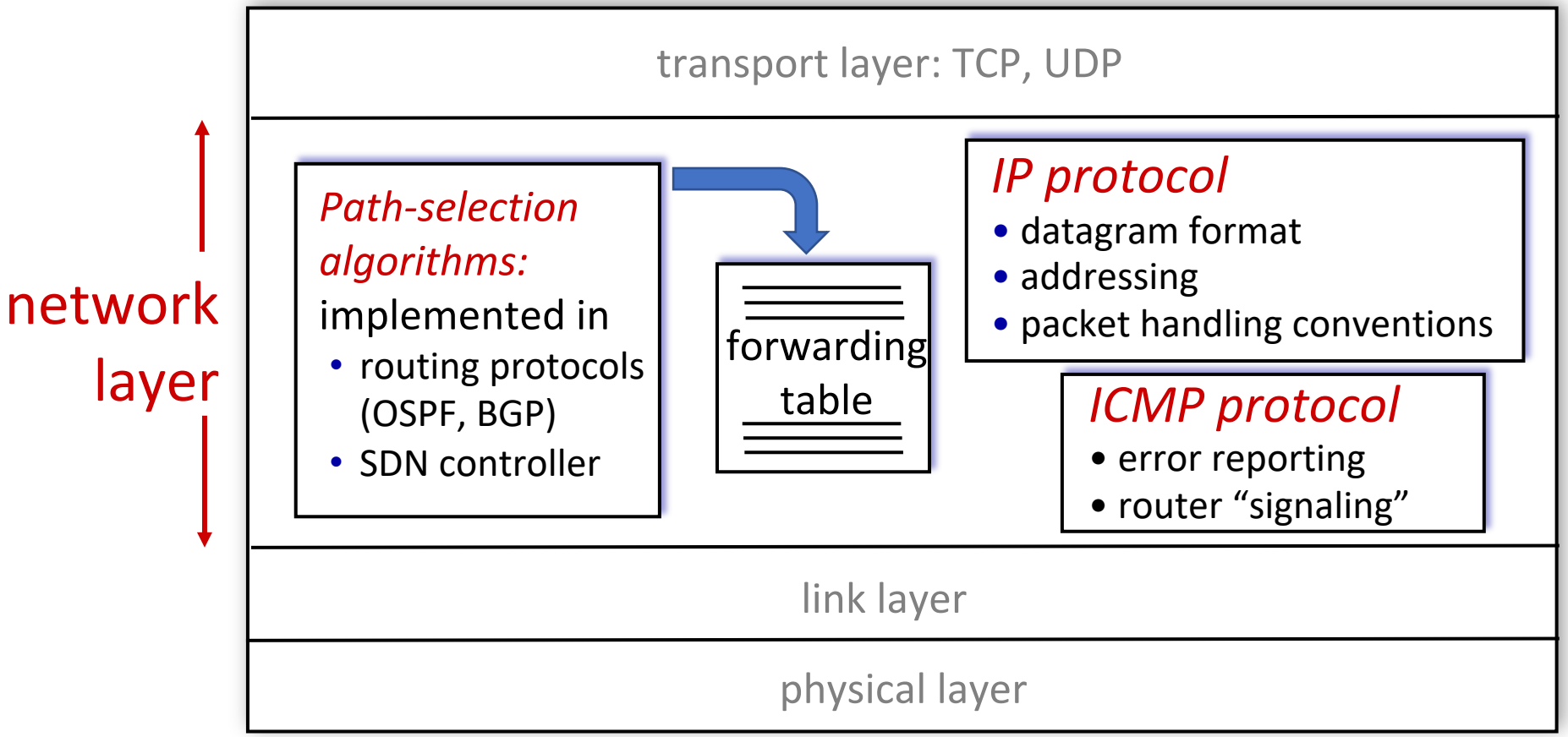
4.5 Introduction to Network Layer Protocols

- DHCP
- ICMP

4.6 IPv6 Addressing

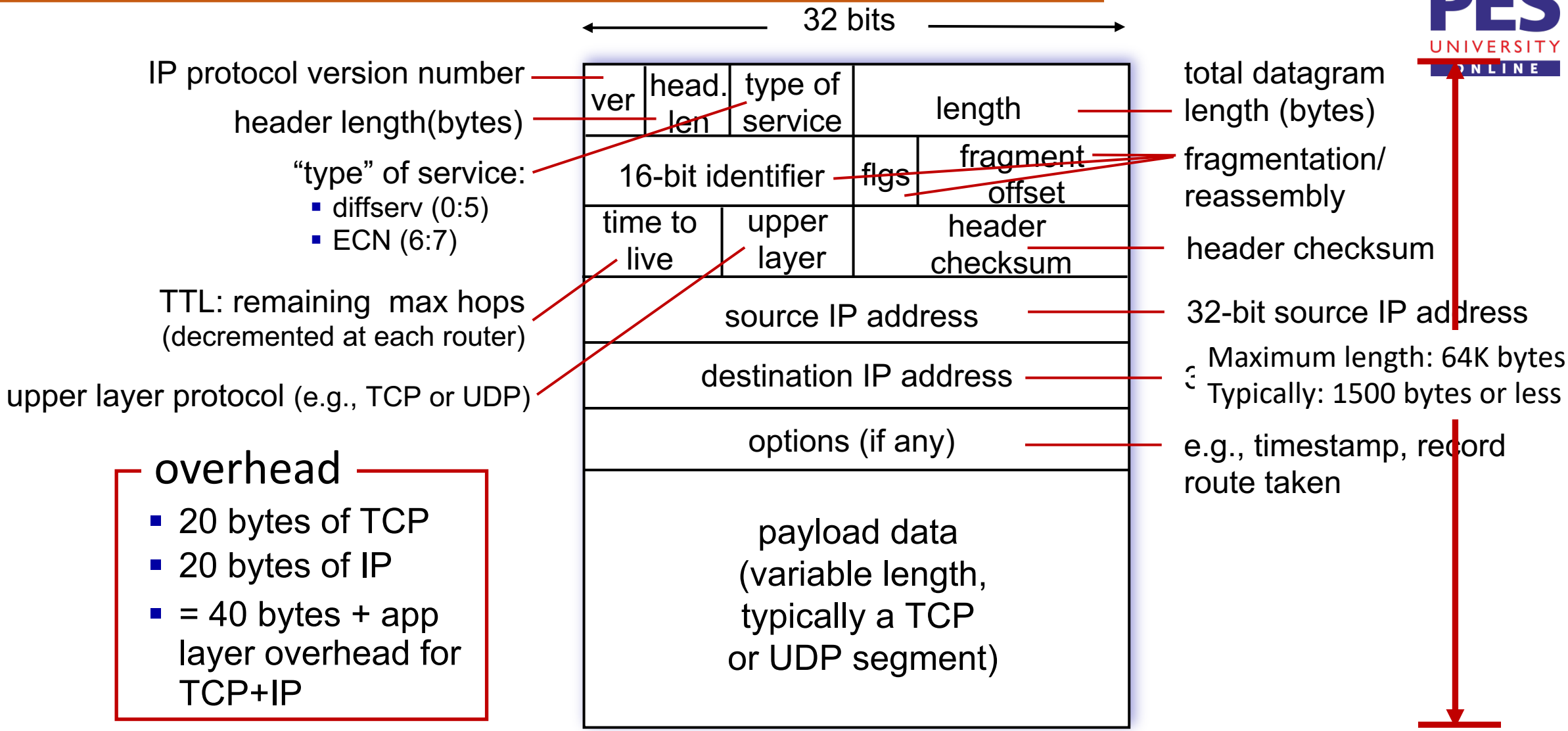
4.7 Introduction to Routing Algorithms

host, router network layer functions:



# COMPUTER NETWORKS

## IP Datagram Format




**overhead**

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

## Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

4.5 Introduction to Network Layer Protocols

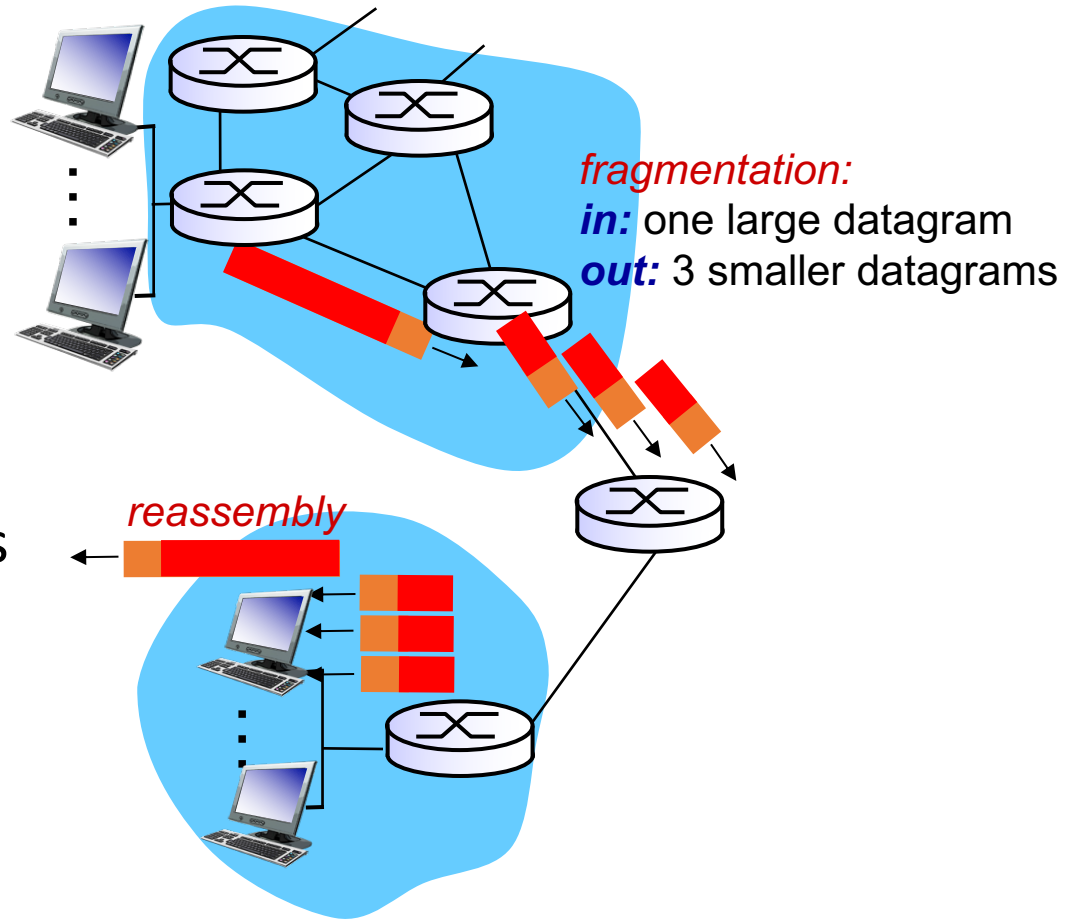
- DHCP
- ICMP

4.6 IPv6 Addressing

4.7 Introduction to Routing Algorithms

# IP fragmentation, reassembly

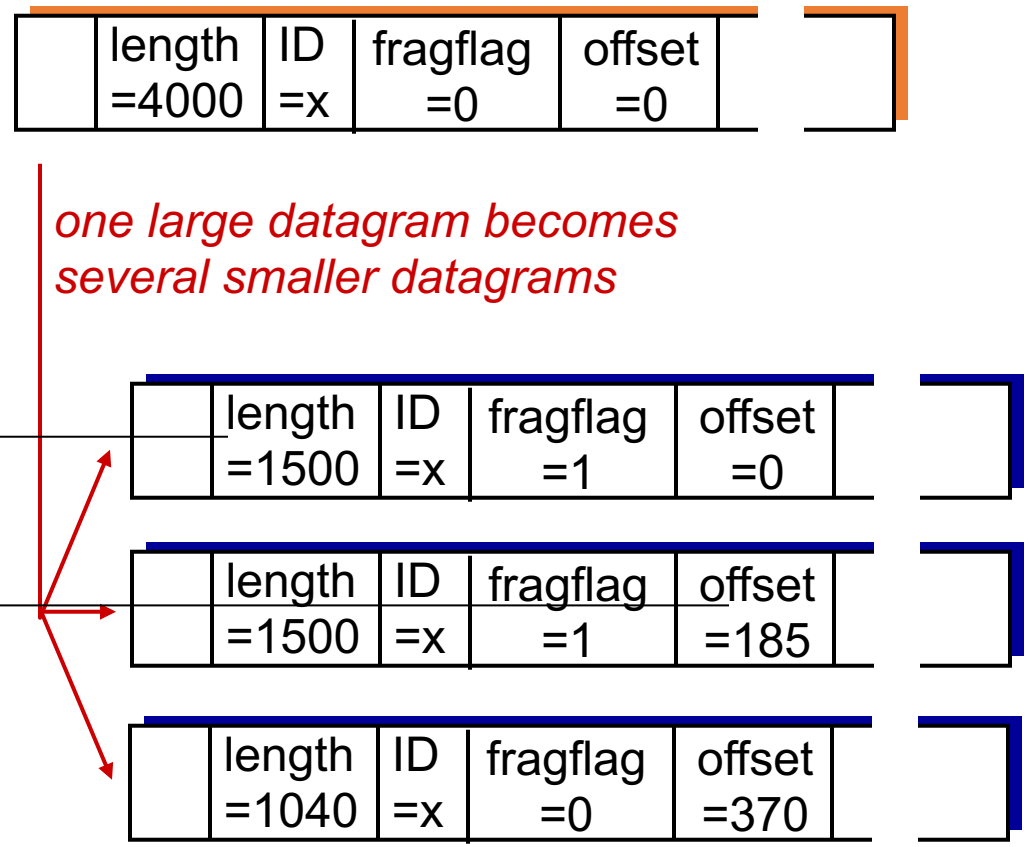
- network links have MTU (max.transfer size) - largest possible link-level frame
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order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



# IP fragmentation, reassembly

*example:*

- 4000 byte datagram
- MTU = 1500 bytes



## Fragmentation – Example


### Original IP Datagram

| Sequence | Identifier | Total Length | DF May / Don't | MF Last / More | Fragment Offset |
|---|---|---|---|---|---|
| 0 | 345 | 5140 | 0 | 0 | 0 |

# Fragmentation – Example

## Original IP Datagram

| Sequence | Identifier | Total Length | DF May / Don't | MF Last / More | Fragment Offset |
|---|---|---|---|---|---|
| 0 | 345 | 5140 | 0 | 0 | 0 |

## IP Fragments (Ethernet)

| Sequence | Identifier | Total Length | DF May / Don't | MF Last / More | Fragment Offset |
|---|---|---|---|---|---|
| 0-0 | 345 | 1500 | 0 | 1 | 0 |
| 0-1 | 345 | 1500 | 0 | 1 | 185 |
| 0-2 | 345 | 1500 | 0 | 1 | 370 |
| 0-3 | 345 | 700 | 0 | 0 | 555 |

## Fragmentation – Numerical Example

- An IP router with a Maximum Transmission Unit (MTU) of 200 bytes has received an IP packet of size 520 bytes with an IP header of length 20 bytes. The values of the relevant fields in the IP header.

## Fragmentation – Numerical Example

- An IP router with a Maximum Transmission Unit (MTU) of 200 bytes has received an IP packet of size 520 bytes with an IP header of length 20 bytes. The values of the relevant fields in the IP header.

| | | | |
|---|---|---|---|
| | 20 \| 176 | 20 \| 176 | 20 \| 148 |
| Fragment Offset | 0 | 22 | 44 |
| MF | 1 | 1 | 0 |
| Header length | 5 | 5 | 5 |
| Total length | 196 | 196 | 168 |

## Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

4.5 Introduction to Network Layer Protocols

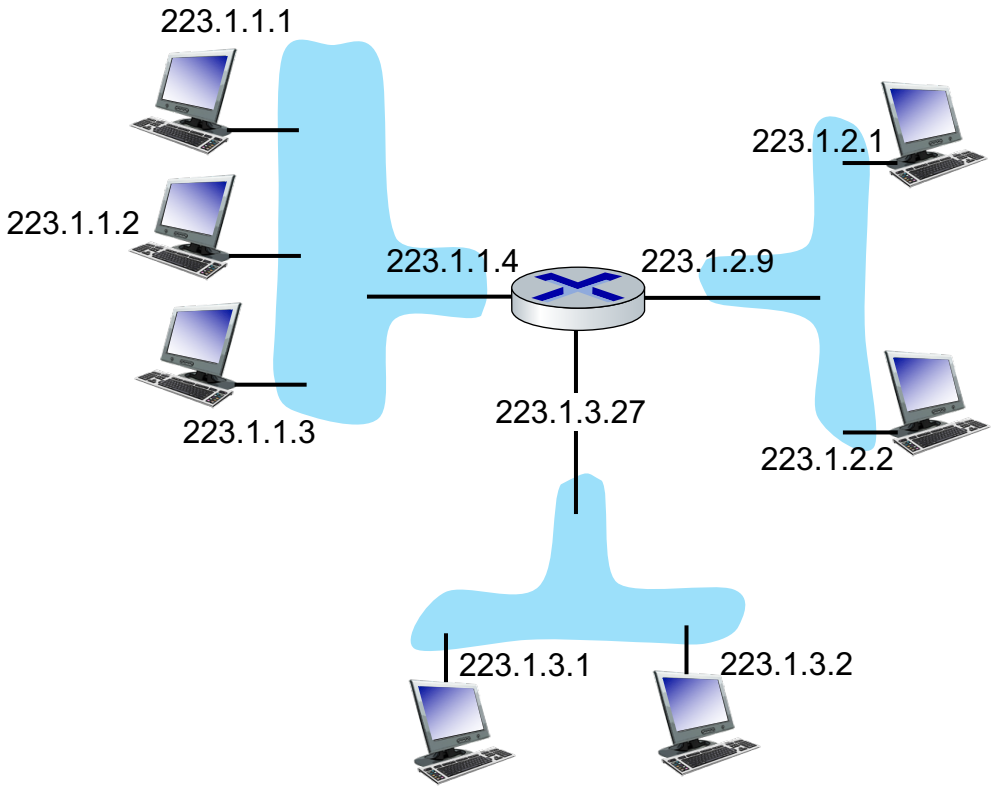
- DHCP
- ICMP

4.6 IPv6 Addressing

4.7 Introduction to Routing Algorithms

# IP addressing: Introduction

- IP address: 32-bit unique ID associated with each host or router *interface*
- interface: connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)




dotted-decimal IP address notation:

223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 11111111

**Solution:**

We replace each group of 8 bits with its equivalent decimal number and add dots for separation.

a. 129.11.11.239
b. 193.131.27.255

Change the following IPv4 addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78
b. 221.34.7.82

**Solution:**

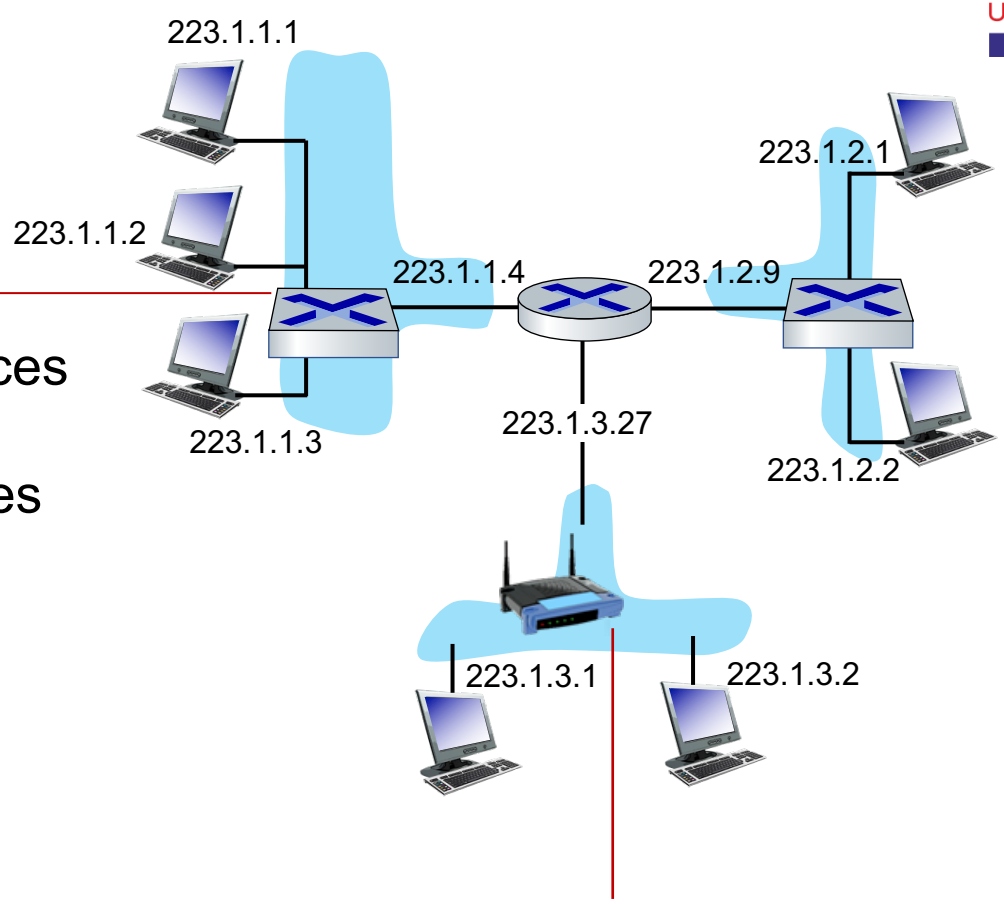
We replace each decimal number with its binary equivalent.

a. 01101111 00111000 00101101 01001110
b. 11011101 00100010 00000111 01010010

# IP addressing: Introduction

Q: how are interfaces actually connected?

A: we'll learn about that in chapters 6, 7



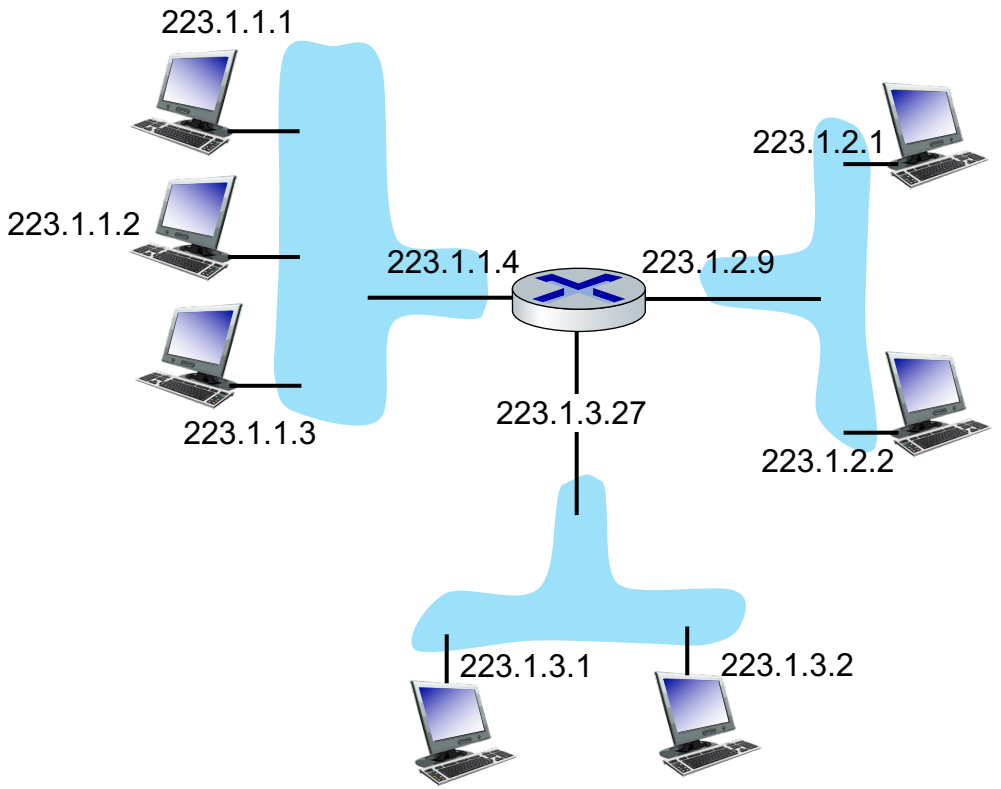

*A:* wired Ethernet interfaces connected by Ethernet switches

*A:* wireless WiFi interfaces connected by WiFi base station

*For now:* don't need to worry about how one interface is connected to another (with no intervening router)

## Subnets

- *What's a subnet ?*
  - device interfaces that can physically reach each other without passing through an intervening router




network consisting of 3 subnets

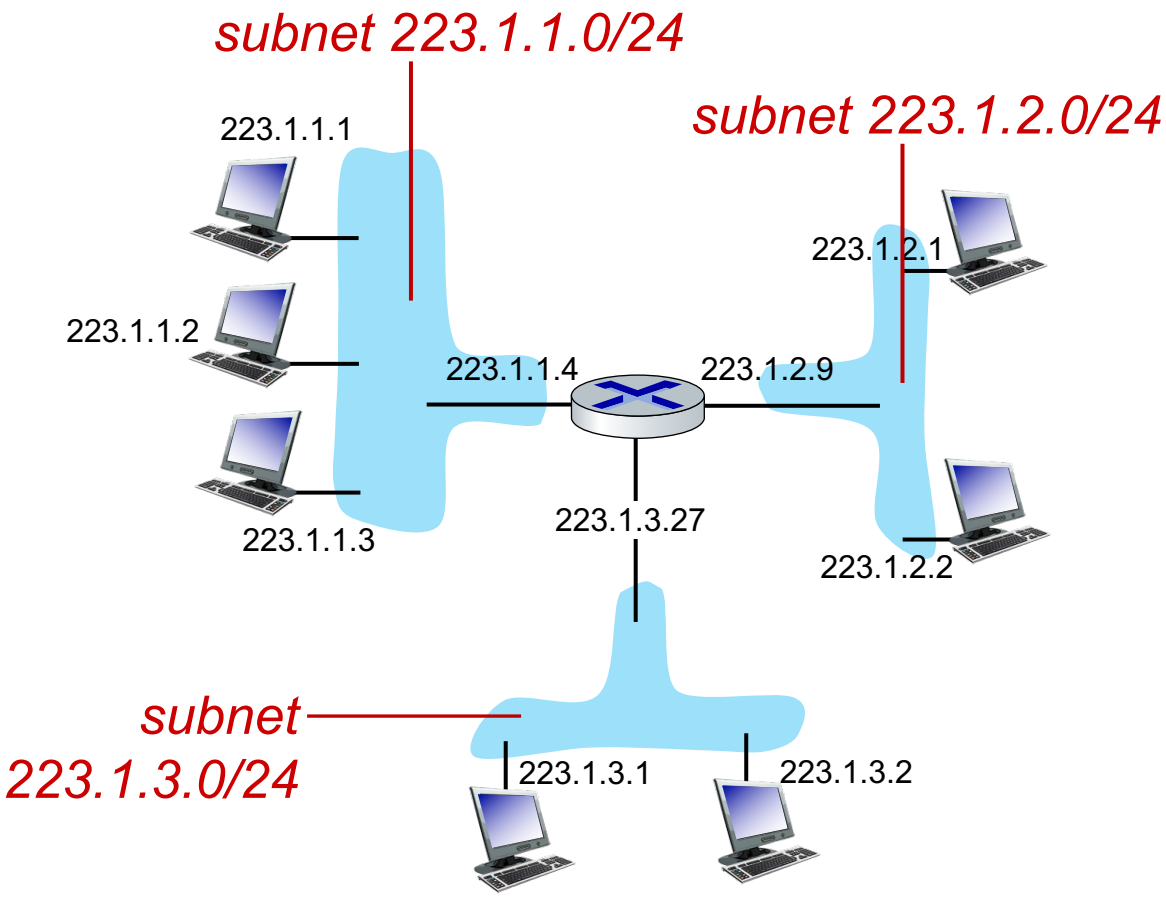
- IP addresses have structure:
  - subnet part: devices in same subnet have common high order bits
  - host part: remaining low order bits

**A portion of a network that shares a particular subnet address.**

# Subnets

*Recipe for defining subnets:*

- detach each interface from its host or router, creating "islands" of isolated networks
- each isolated network is called a *subnet*

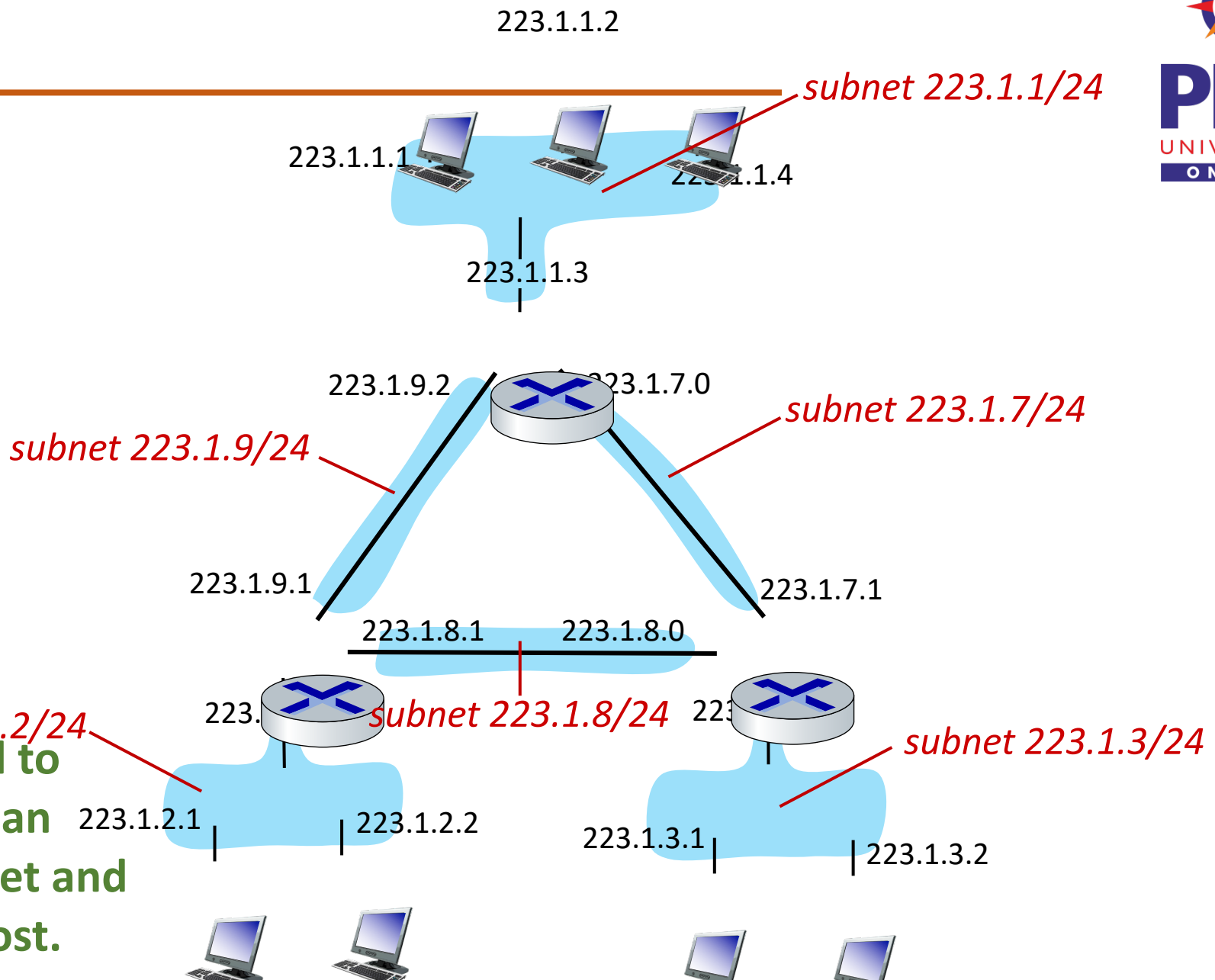



subnet mask: /24

(high-order 24 bits: subnet part of IP address)

## Subnets

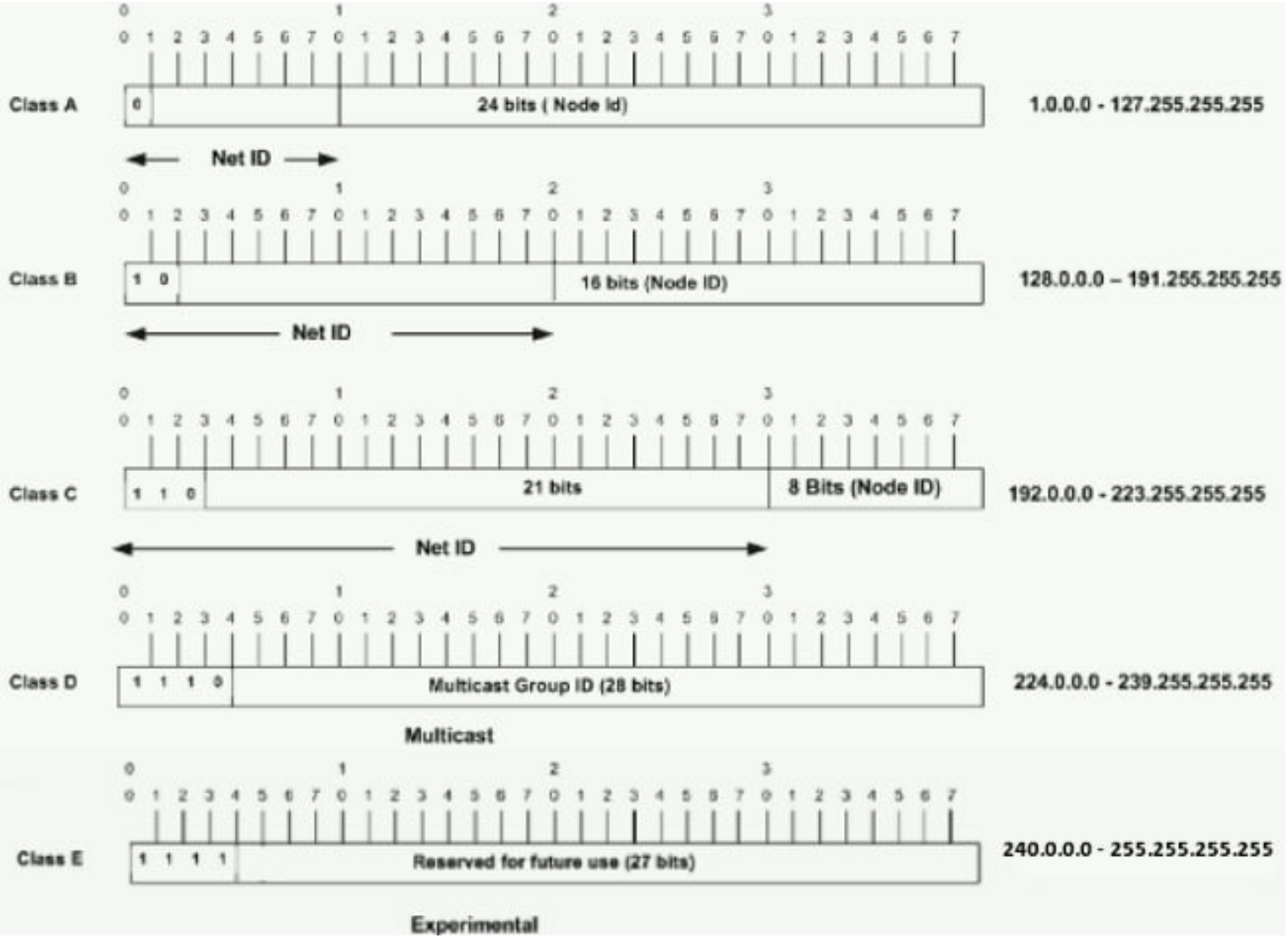
- where are the subnets?
- what are the /24 subnet addresses?




**A 32-bit combination used to describe which portion of an address refers to the subnet and which part refers to the host.**

# COMPUTER NETWORKS

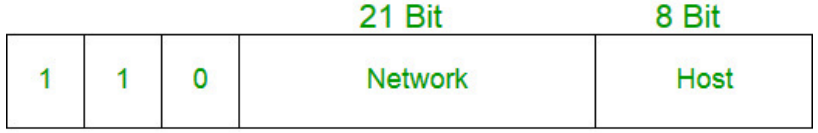
## IP addressing: Classful Addressing



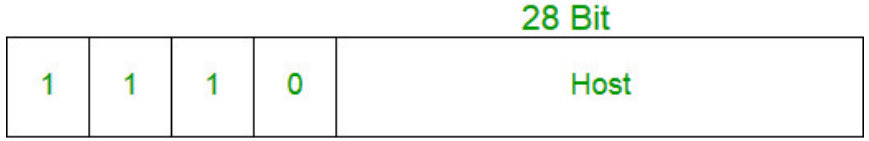


**Class A**

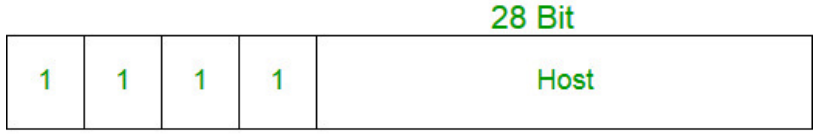

**Class B**




**Class C**




**Class D**




**Class E**

# IP addressing: Classful Addressing

| Class | HOB | NET ID Bits | Host ID Bits | No of Networks | Host Per Network | Start Address | End Address |
|---|---|---|---|---|---|---|---|
| Class A | 0 | 8 | 24 | $2^7$=128 | $2^{24}$=16,777,216 | 0.0.0.0 | 127.255.255.255 |
| Class B | 10 | 16 | 16 | $2^{14}$=16,384 | $2^{16}$=65,536 | 128.0.0.0 | 191.255.255.255 |
| Class C | 110 | 24 | 8 | $2^{21}$=2,097,152 | $2^8$=256 | 192.0.0.0 | 223.255.255.255 |
| Class D | 1110 | - | - | - | - | 224.0.0.0 | 239.255.255.255 |
| Class E | 1111 | - | - | - | - | 240.0.0.0 | 255.255.255.255 |

**Class D: multicast, Class E: reserved (experimental)**

**Broadcast Address: 255.255.255.255**

**Network Masks**

Class A: 255.0.0.0
Class B: 255.255.0.0
Class C: 255.255.255.0

# IP addressing: Classful Subnet Masks

| Class | Binary | Dotted-Decimal | CIDR Notation |
|---|---|---|---|
| A | 11111111 00000000 00000000 00000000 | 255.0.0.0 | /8 |
| B | 11111111 11111111 00000000 00000000 | 255.255.0.0 | /16 |
| C | 11111111 11111111 11111111 00000000 | 255.255.255.0 | /24 |

**Find the class of each address.**

a. 00000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 11111111
c. 14.23.120.8
d. 252.5.15.111

**Solution**

a. The first bit is 0. This is a class A address.
b. The first 2 bits are 1; the third bit is 0. This is a class C address.
c. The first byte is 14; the class is A.
d. The first byte is 252; the class is E.

## IP addressing: Why not Classful Addressing?

- Network portion - 8, 16, or 24 bits in length – known as **Class A, B** and **C** networks respectively
- A class **C (/24)** - accommodate only up to $2^8$ - 2 = **254 hosts** (two of the $2^8$ = 256 addresses are reserved for special use)
- A class **B (/16)** subnet, supports up to **65,634 hosts**
- A class **A (/8)** subnet, up to **16,777,214 hosts**

**too small for many organizations**

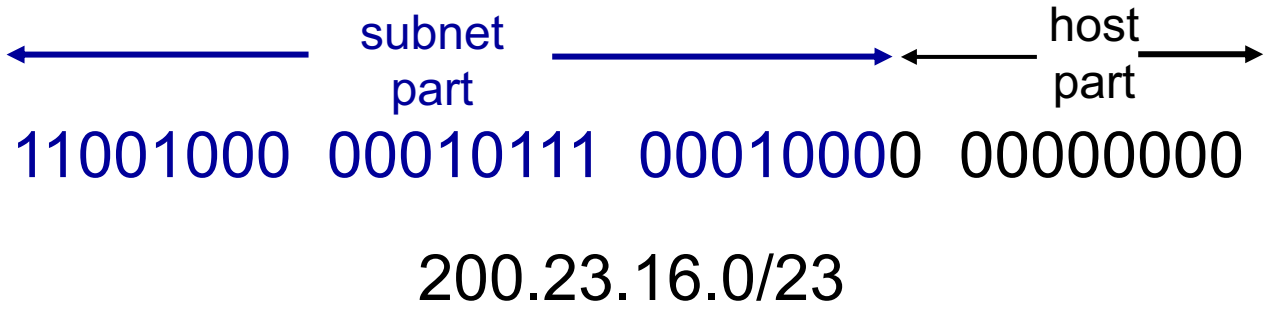
**too large, poor utilization**

1. $2^n$=Number of Network
2. $2^h$-2 = Number of Host

**Classful addressing, which is almost obsolete, is replaced with classless addressing.**

# CIDR: Classless InterDomain Routing (pronounced "cider")

- subnet portion of address of arbitrary length
- address format: a.b.c.d/x, where x is # bits in subnet portion of address



# COMPUTER NETWORKS

## IP addressing: CIDR

| CIDR Block Prefix | # of Host Addresses (inclusive of network and broadcast-id) |
|---|---|
| /30 | 4 hosts (valid host would be 2) |
| /29 | 8 hosts (valid host would be 6) |
| /28 | 16 hosts (valid host would be 14) |
| /27 | 32 hosts |
| /26 | 64 hosts |
| /25 | 128 hosts |
| /24 | 256 hosts |
| /23 | 512 hosts |
| /22 | 1,024 hosts |
| /21 | 2,048 hosts |
| /20 | 4,096 hosts |
| /19 | 8,192 hosts |
| /18 | 16,384 hosts |
| /17 | 32,768 hosts |
| /16 | 65,536 hosts |
| /15 | 131,072 hosts |
| /14 | 262,144 hosts |
| /13 | 524,288 hosts |
| /12 | 1,048,576 hosts and so on..... |

## IP addressing: CIDR – Prefix Lengths

| /n | Mask | /n | Mask | /n | Mask | /n | Mask |
|---|---|---|---|---|---|---|---|
| /1 | 128.0.0.0 | /9 | 255.128.0.0 | /17 | 255.255.128.0 | /25 | 255.255.255.128 |
| /2 | 192.0.0.0 | /10 | 255.192.0.0 | /18 | 255.255.192.0 | /26 | 255.255.255.192 |
| /3 | 224.0.0.0 | /11 | 255.224.0.0 | /19 | 255.255.224.0 | /27 | 255.255.255.224 |
| /4 | 240.0.0.0 | /12 | 255.240.0.0 | /20 | 255.255.240.0 | /28 | 255.255.255.240 |
| /5 | 248.0.0.0 | /13 | 255.248.0.0 | /21 | 255.255.248.0 | /29 | 255.255.255.248 |
| /6 | 252.0.0.0 | /14 | 255.252.0.0 | /22 | 255.255.252.0 | /30 | 255.255.255.252 |
| /7 | 254.0.0.0 | /15 | 255.254.0.0 | /23 | 255.255.254.0 | /31 | 255.255.255.254 |
| /8 | 255.0.0.0 | /16 | 255.255.0.0 | /24 | 255.255.255.0 | /32 | 255.255.255.255 |

**The addresses in color are the default masks for classes A, B, and C.**
**Thus, classful addressing is a special case of classless addressing.**

## IP addresses: how to get one?

*Q:* how does *network* get subnet part of IP address?

*A:* gets allocated portion of its provider ISP's address space

| | | |
|---|---|---|
| ISP's block | 11001000 00010111 00010000 00000000 | 200.23.16.0/20 |

ISP can then allocate out its address space in 8 blocks:

| | | | |
|---|---|---|---|
| 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 |

- IP address of one host is 25.34.12.56/16
- Find the first addresses in this block.
- Find the last addresses in this block.

- IP address of one host is 182.44.82.16/26
- Find the first addresses in this block.
- Find the last addresses in this block.

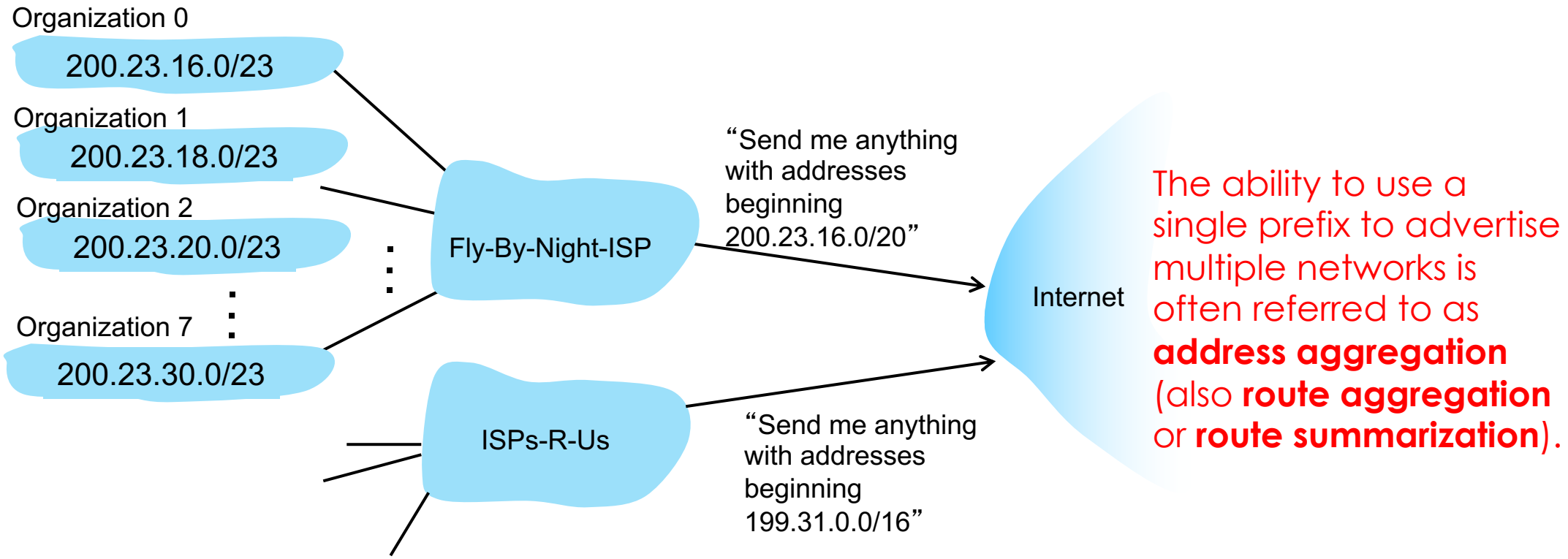
217.83.58.0/26 — organization's block.
↳ the admin — 4 subnets

- An organization is granted the block 214.17.160.0/24. The administrator wants to create 8 subnets.
- Find the subnet mask.
- Find the last addresses in first subnet.
- Find the first addresses in last subnet.

1) What is the subnet mask of each subnet
2) What is the block of 1st subnet - addr
3) 1st addr in 1st subnet
4) last addr in 1st subnet
5) What is the addr space per subnet
6) How many hosts can be connected per subnet?

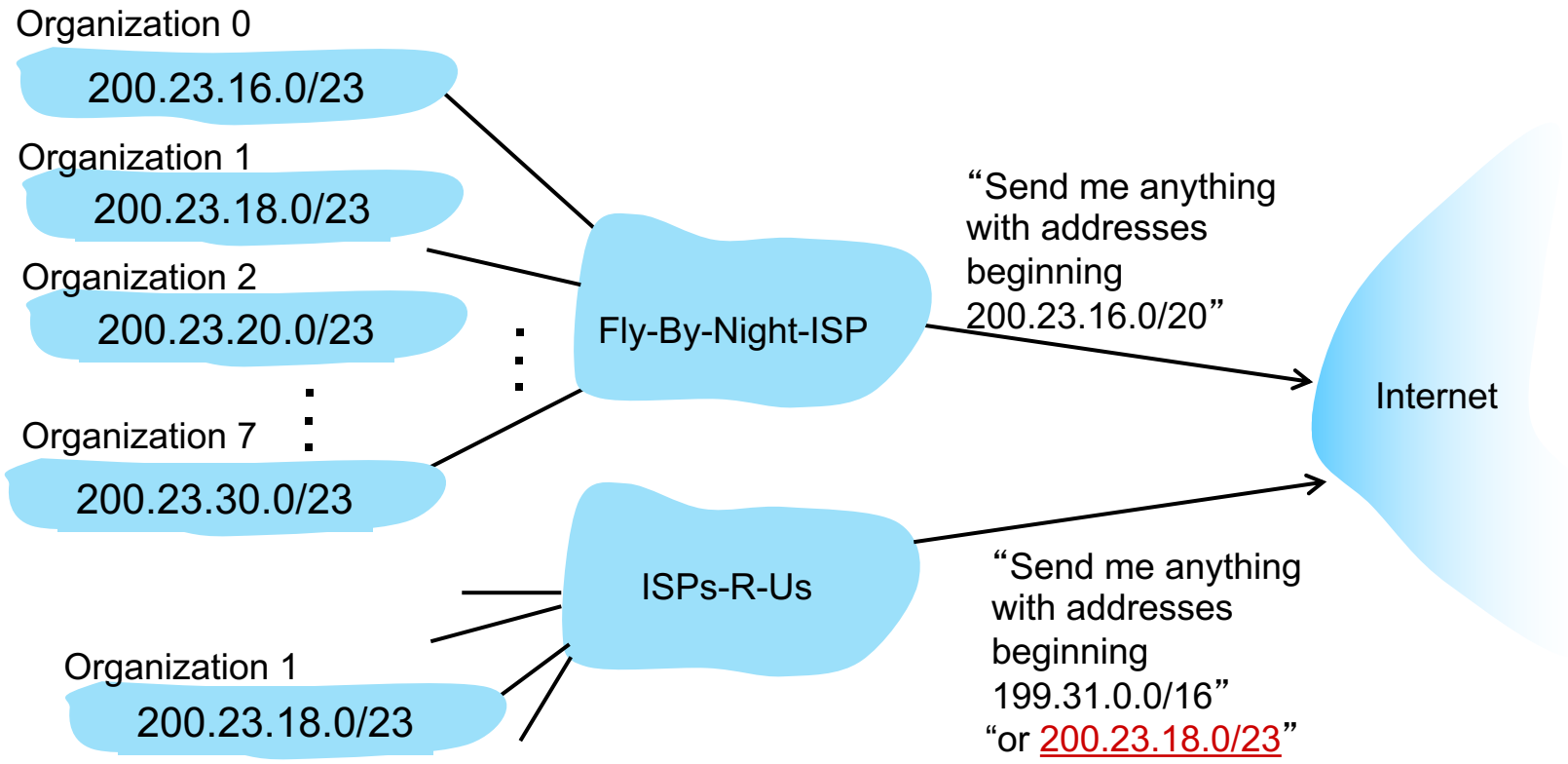
# Hierarchical addressing: route aggregation

hierarchical addressing allows efficient advertisement of routing information:



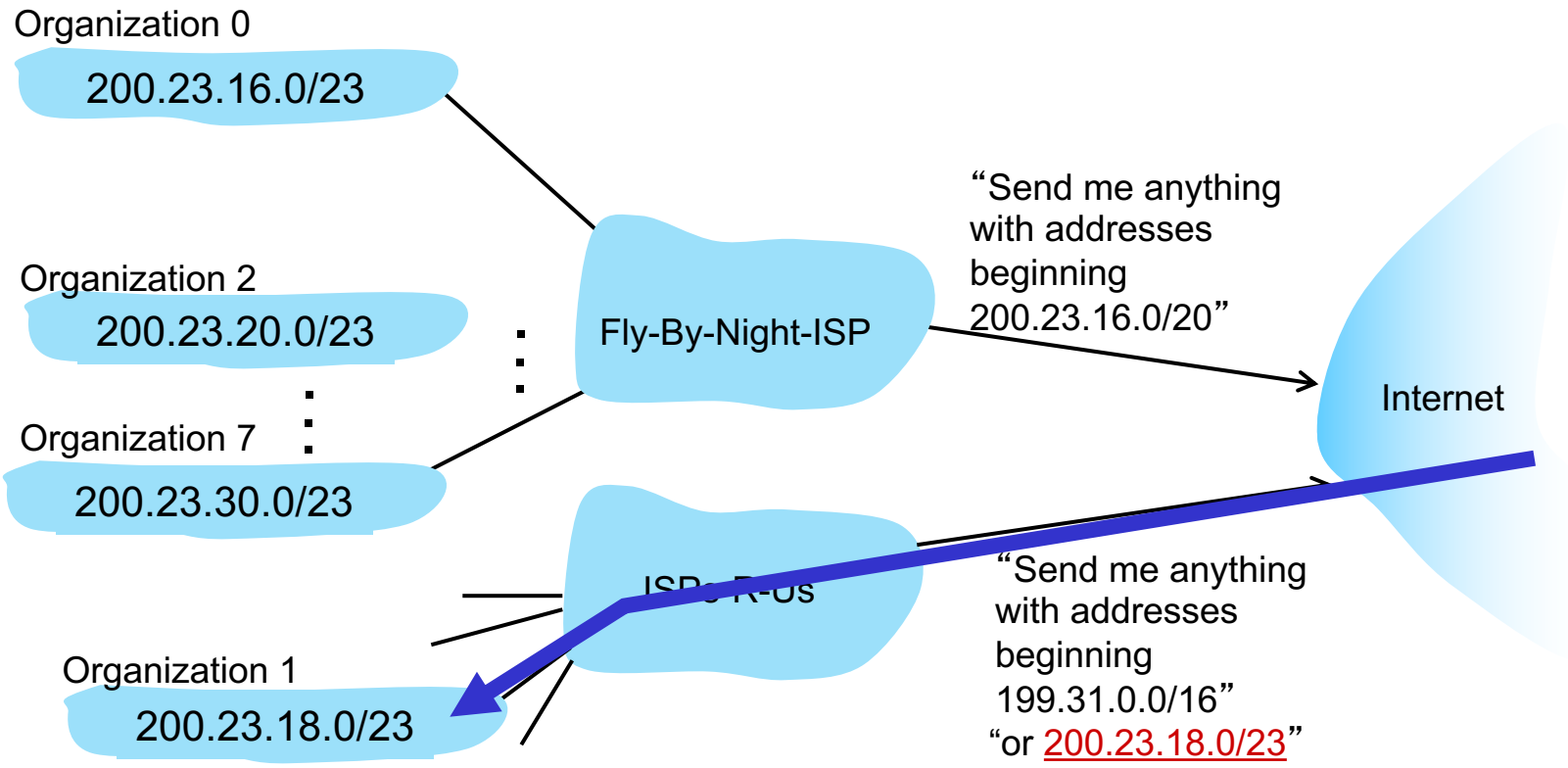
# Hierarchical addressing: more specific routes

- Organization 1 moves from Fly-By-Night-ISP to ISPs-R-Us
- ISPs-R-Us now advertises a more specific route to Organization 1



# Hierarchical addressing: more specific routes

- Organization 1 moves from Fly-By-Night-ISP to ISPs-R-Us
- ISPs-R-Us now advertises a more specific route to Organization 1



# IP addresses: how to get one?

That's actually two questions:

1. Q: How does a *host* get IP address within its network (host part of address)?
2. Q: How does a *network* get IP address for itself (network part of address)

How does *host* get IP address?

- hard-coded by sysadmin in config file (e.g., /etc/rc.config in UNIX)
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - "plug-and-play"

## Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

4.5 Introduction to Network Layer Protocols

- DHCP
- ICMP

4.6 IPv6 Addressing

4.7 Introduction to Routing Algorithms

# DHCP: Dynamic Host Configuration Protocol

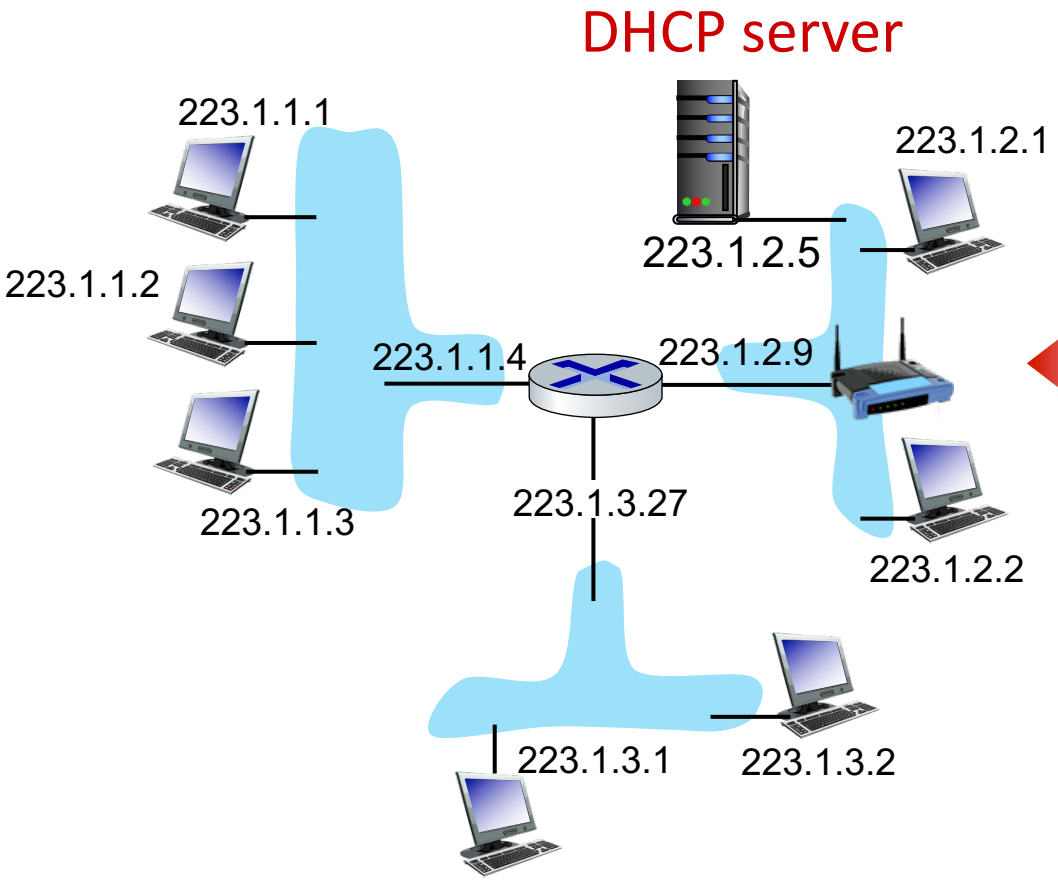
goal: host *dynamically* obtains IP address from network server when it "joins" network

- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/on)
- support for mobile users who join/leave network

## DHCP overview:

- host broadcasts DHCP discover msg [optional]
- DHCP server responds with DHCP offer msg [optional]
- host requests IP address: DHCP request msg
- DHCP server sends address: DHCP ack msg

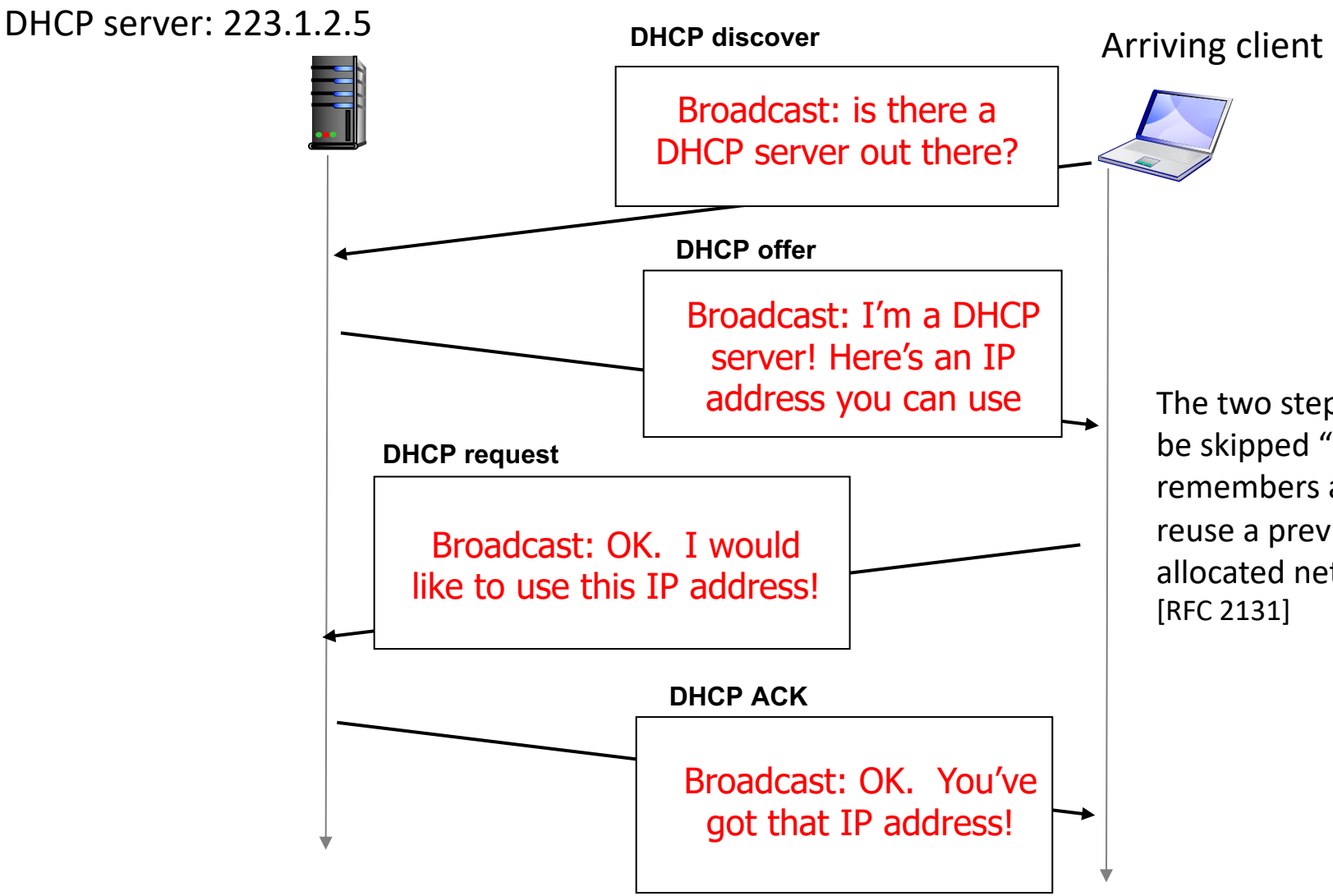
# DHCP client-server scenario




Typically, DHCP server will be co-located in router, serving all subnets to which router is attached

arriving DHCP client needs address in this network

# DHCP client-server scenario



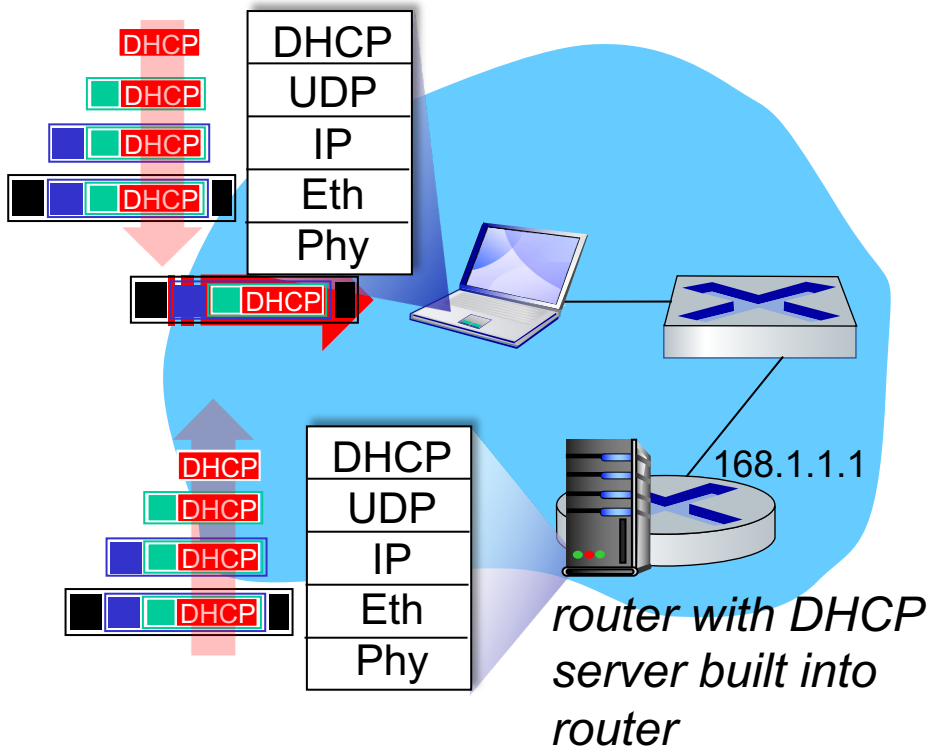

The two steps above can be skipped "if a client remembers and wishes to reuse a previously allocated network address" [RFC 2131]

## DHCP: more than IP addresses

DHCP can return more than just allocated IP address on subnet:

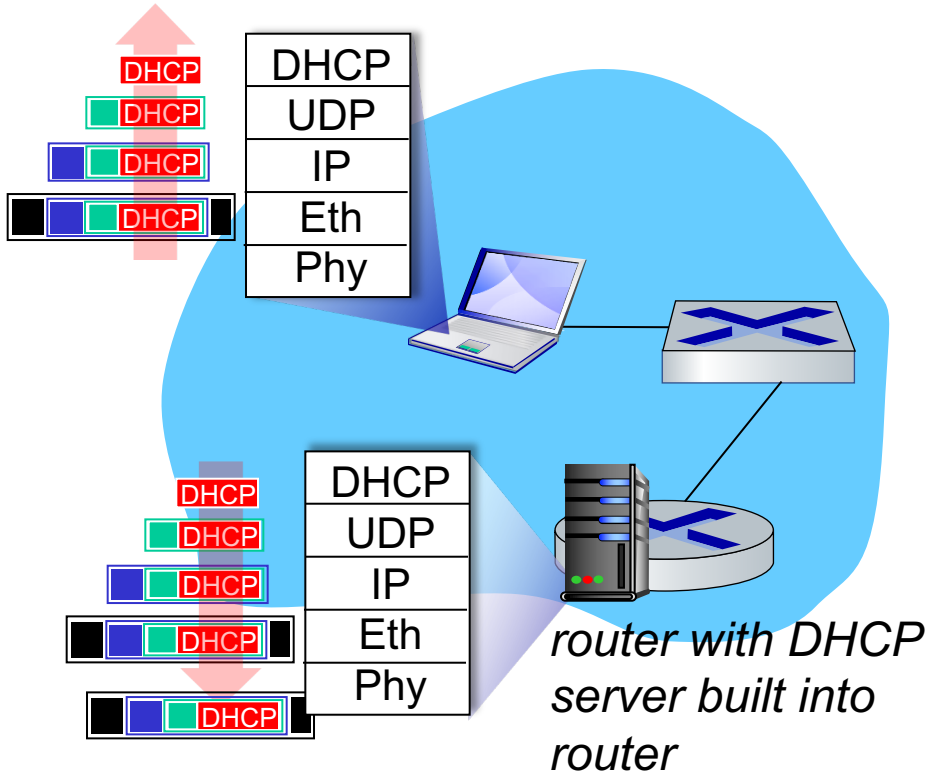
- address of first-hop router for client
- name and IP address of DNS sever
- network mask (indicating network versus host portion of address)

# DHCP: example




- Connecting laptop will use DHCP to get IP address, address of first-hop router, address of DNS server.
- DHCP REQUEST message encapsulated in UDP, encapsulated in IP, encapsulated in Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet demux'ed to IP demux'ed, UDP demux'ed to DHCP

# DHCP: example




- DCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
- encapsulated DHCP server reply forwarded to client, demuxing up to DHCP at client
- client now knows its IP address, name and IP address of DNS server, IP address of its first-hop router

# COMPUTER NETWORKS

## IP addressing: last words…

*Q:* how does an ISP get block of addresses?

*A:* ICANN: Internet Corporation for Assigned Names and Numbers
http://www.icann.org/

- allocates IP addresses, through 5 regional registries (RRs) (who may then allocate to local registries)
- manages DNS root zone, including delegation of individual TLD (.com, .edu, …) management

*Q:* are there enough 32-bit IP addresses?

- ICANN allocated last chunk of IPv4 addresses to RRs in 2011
- NAT (next) helps IPv4 address space exhaustion
- IPv6 has 128-bit address space

"Who the hell knew how much address space we needed?" Vint Cerf (reflecting on decision to make IPv4 address 32 bits long)

## Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

4.5 Introduction to Network Layer Protocols

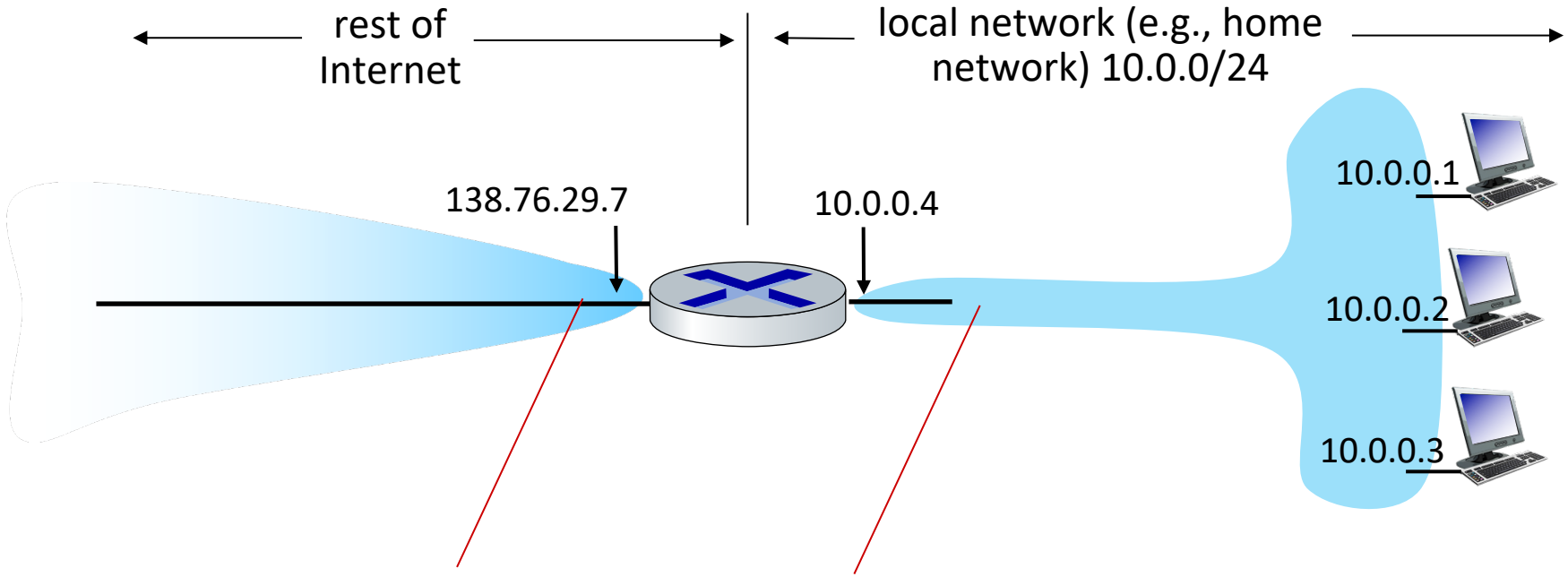
- DHCP
- ICMP

4.6 IPv6 Addressing

4.7 Introduction to Routing Algorithms

# NAT: Network Address Translation

NAT: all devices in local network share just one IPv4 address as far as outside world is concerned




*all* datagrams *leaving* local network have *same* source NAT IP address: 138.76.29.7, but *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

- all devices in local network have 32-bit addresses in a “private” IP address space (10/8, 172.16/12, 192.168/16 prefixes) that can only be used in local network
- advantages:
  - just one IP address needed from provider ISP for *all* devices
  - can change addresses of host in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - security: devices inside local net not directly addressable, visible by outside world

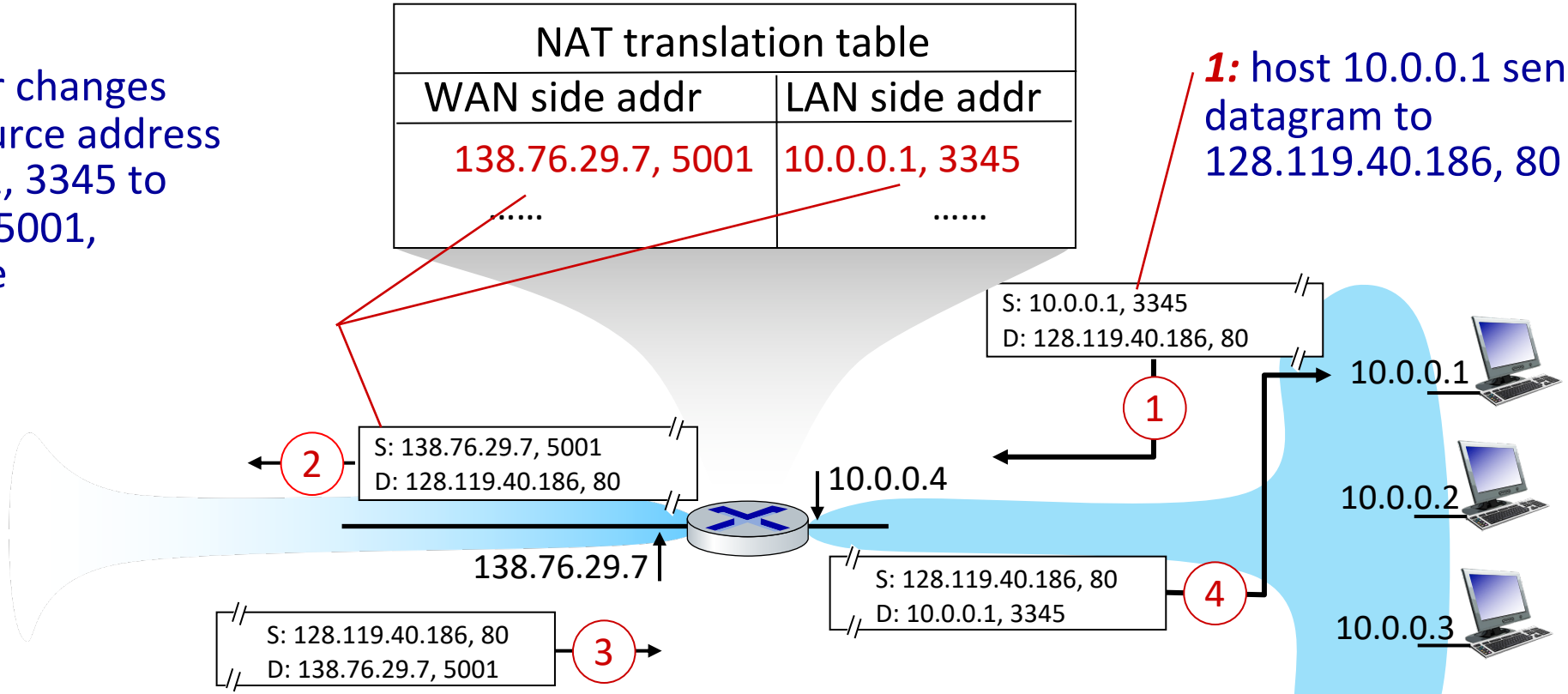
## NAT: Network Address Translation

implementation: NAT router must (transparently):

- outgoing datagrams: replace (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  - remote clients/servers will respond using (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 destination fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

# NAT: Network Address Translation

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

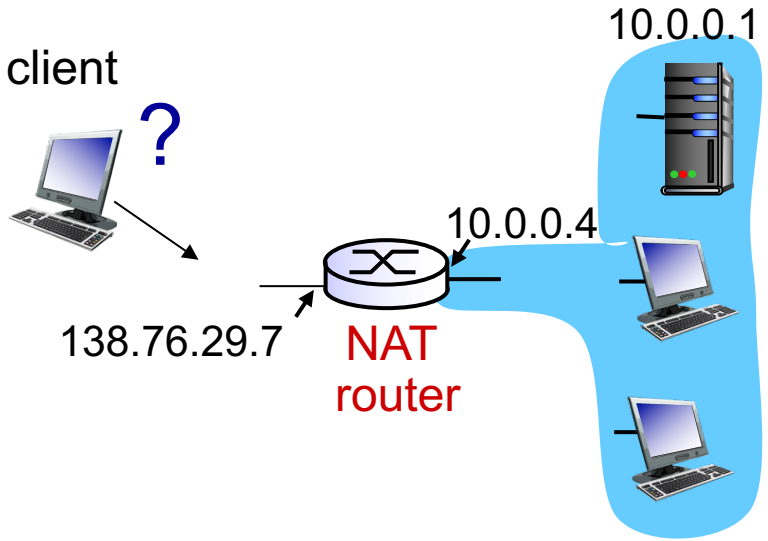



***3:*** reply arrives, destination address: 138.76.29.7, 5001

## NAT: Network Address Translation

- NAT has been controversial:
  - routers “should” only process up to layer 3
  - address “shortage” should be solved by IPv6
  - violates end-to-end argument (port # manipulation by network-layer device)
  - NAT traversal: what if client wants to connect to server behind NAT?
- but NAT is here to stay:
  - extensively used in home and institutional nets, 4G/5G cellular nets

# NAT: Network Address Translation




*Solution:* statically configure NAT to forward incoming connection requests at given port to server

- e.g., (123.76.29.7, port 25000) always forwarded to 10.0.0.1 port 25000

## Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

4.5 Introduction to Network Layer Protocols

- DHCP
- ICMP

4.6 IPv6 Addressing

4.7 IPv6 Protocol

4.8 Introduction to Routing Algorithms

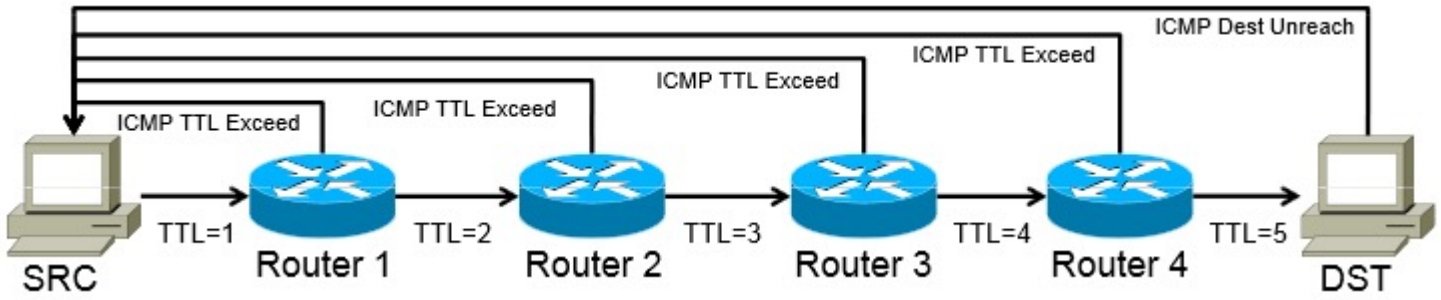
# ICMP: Internet Control Message Protocol

- Used by hosts and routers to communicate network-layer information.
- Typical use of ICMP is for error reporting.
- Eg: “Destination network unreachable”
- ICMP messages – carried as IP payloads.
- ICMP is often considered part of IP, but architecturally it lies just above IP.
- ICMP messages have a type and a code field.

| 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: Internet Control Message Protocol

- Perform – Network diagnostics
- Utilities traceroute and ping – use ICMP.
- Other examples of ICMP messages:
  - Source quench message (congestion)
  - Parameter problem (bad IP header)
  - Time exceeded message (TTL)
  - Destination unreachable
  - Redirection message



## Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

4.5 Introduction to Network Layer Protocols

- DHCP
- ICMP

4.6 IPv6 Addressing

4.7 IPv6 Protocol

4.8 Introduction to Routing Algorithms

# Introduction to IPv6

- 128 bits address
- 340 undecillion address (340 + 33 digits)
- IPv6 address apace is $2^{96}$ times of the IPv4 address.
- Made up of eight 16-bit blocks (octet)
- Separated by ':'
- Hexadecimal, each hexadecimal character – 4 bits

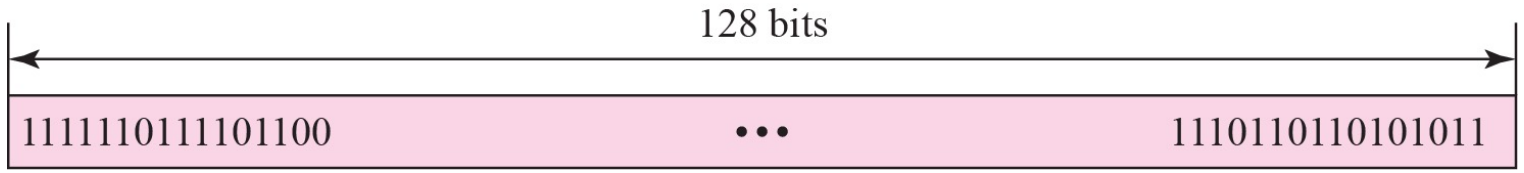
**340,282,366,920,938,463,374,607,431,768,211,456**

**Rules:**

- Discard leading Zero(es)
- If two of more blocks contain consecutive zeroes, omit them all and replace with double colon sign ::

| Decimal | Binary | Hexadecimal |
|---|---|---|
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| 10 | 1010 | A |
| 11 | 1011 | B |
| 12 | 1100 | C |
| 13 | 1101 | D |
| 14 | 1110 | E |
| 15 | 1111 | F |

# Introduction to IPv6




*Colon hexadecimal notation*

FDEC : BA98 : 7654 : 3210 : ADBF : BBFF : 2922 : FFFF

*Zero compression*

FDEC : 0 : 0 : 0 : 0 : BBFF : 0 : FFFF → FDEC :: BBFF : 0 : FFFF

Original address | Zero compressed

*CIDR address*

FDEC :: BBFF : 0 : FFFF/60

*Rule 1*

2001:0000:3238:DFE1:0063:0000:0000:FEFB

2001:0000:3238:DFE1:63:0000:0000:FEFB

*Rule 2*

2001:0000:3238:DFE1:0063:0000:0000:FEFB

2001:0000:3238:DFE1:63::FEFB

2001:0:3238:DFE1:63::FEFB

Show the unabbreviated colon hex notation for the following IPv6 addresses:

a. An address with 64 0s followed by 64 1s.
b. An address with 128 0s.
c. An address with 128 1s.
d. An address with 128 alternative 1s and 0s.

*Solution*

a. 0000:0000:0000:0000:FFFF:FFFF:FFFF:FFFF
b. 0000:0000:0000:0000:0000:0000:0000:0000
c. FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF
d. AAAA:AAAA:AAAA:AAAA:AAAA:AAAA:AAAA:AAAA

The following shows the zero contraction version of addresses in previous example.

a. :: FFFF:FFFF:FFFF:FFFF
b. ::
c. FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF
d. AAAA:AAAA:AAAA:AAAA:AAAA:AAAA:AAAA:AAAA

## IPv6 Addressing

Show abbreviations for the following addresses:

a. 0000:0000:FFFF:0000:0000:0000:0000:0000

b. 1234:2346:0000:0000:0000:0000:0000:1111

c. 0000:0001:0000:0000:0000:0000:1200:1000

d. 0000:0000:0000:0000:0000:FFFF:24.123.12.6

**Solution**

a. 0:0:FFFF::

b. 1234:2346::1111

c. 0:1::1200:1000

d. ::FFFF:24.123.12.6

Decompress the following addresses and show the complete unabbreviated IPv6 address:

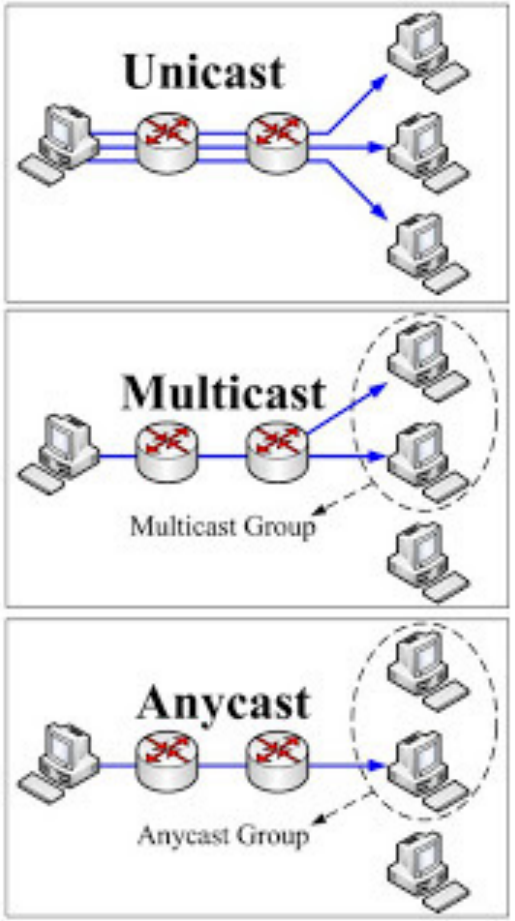
a. 1111::2222
b. ::
c. 0:1::
d. AAAA:A:AA::1234

**Solution**

a. 1111:0000:0000:0000:0000:0000:0000:2222
b. 0000:0000:0000:0000:0000:0000:0000:0000
c. 0000:0001:0000:0000:0000:0000:0000:0000
d. AAAA:000A:00AA:0000:0000:0000:0000:1234

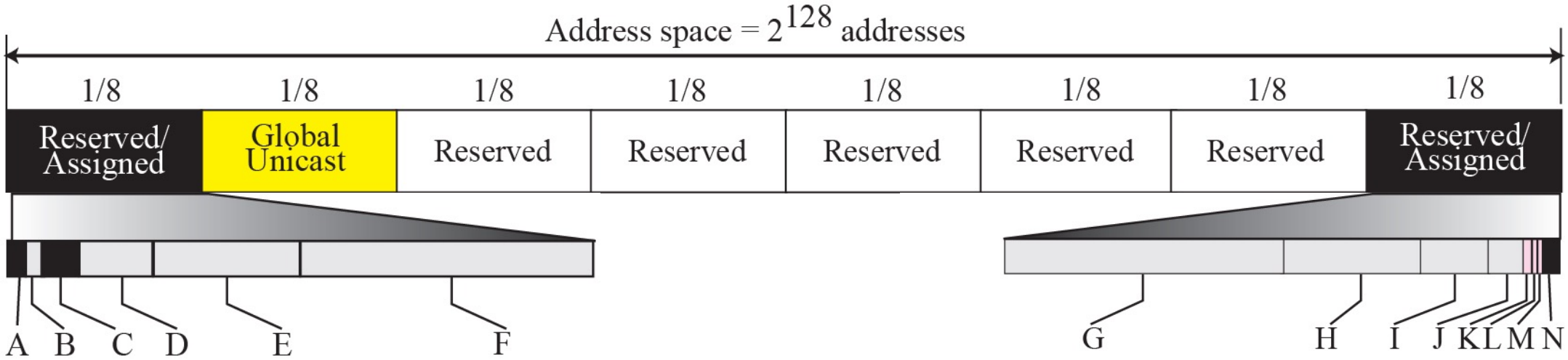
## Three Address Types

- In IPv6, a destination address belong to:
  - **Unicast Address (one-to-one)** – defines a single interface (computer or router)
  - **Multicast Address (one-to-many)** – also defines a group of computers, each member of the group receives a copy.
  - **Anycast Address (one-to-nearest)** – defines a group of computers that all share a single address.
  - **Broadcast Address (one-to-all)** – ×



# COMPUTER NETWORKS

## Address Space Allocation




**A: 1/256 IPv4 Compatible**
B: 1/256 Reserverd
C: 1/128 Reserved
D: 1/64 Reserved
E: 1/32 Reserved
F: 1/16 Reserved
G: 1/16 Reserved
H: 1/32 Reserved
I: 1/64 Reserved
**J: 1/128 Unique Local Unicast**
K: 1/512 Reserved
**L: 1/1024 Link Local**
M: 1/256 Reserved
**N: 1/256 Multicast**

# Address Space Allocation

| | Block Prefix | CIDR | Block Assignment | Fraction |
|---|---|---|---|---|
| 1 | 0000 0000 | 0000::/8 | Reserved (IPv4 compatible) | 1/256 |
| | 0000 0001 | 0100::/8 | Reserved | 1/256 |
| | 0000 001 | 0200::/7 | Reserved | 1/128 |
| | 0000 01 | 0400::/6 | Reserved | 1/64 |
| | 0000 1 | 0800::/5 | Reserved | 1/32 |
| | 0001 | 1000::/4 | Reserved | 1/16 |
| 2 | 001 | 2000::/3 | Global unicast | 1/8 |
| 3 | 010 | 4000::/3 | Reserved | 1/8 |
| 4 | 011 | 6000::/3 | Reserved | 1/8 |
| 5 | 100 | 8000::/3 | Reserved | 1/8 |
| 6 | 101 | A000::/3 | Reserved | 1/8 |
| 7 | 110 | C000::/3 | Reserved | 1/8 |
| 8 | 1110 | E000::/4 | Reserved | 1/16 |
| | 1111 0 | F000::/5 | Reserved | 1/32 |
| | 1111 10 | F800::/6 | Reserved | 1/64 |
| | 1111 110 | FC00::/7 | Unique local unicast | 1/128 |
| | 1111 1110 0 | FE00::/9 | Reserved | 1/512 |
| | 1111 1110 10 | FE80::/10 | Link local addresses | 1/1024 |
| | 1111 1110 11 | FEC0::/10 | Reserved | 1/1024 |
| | 1111 1111 | FF00::/8 | Multicast addresses | 1/256 |

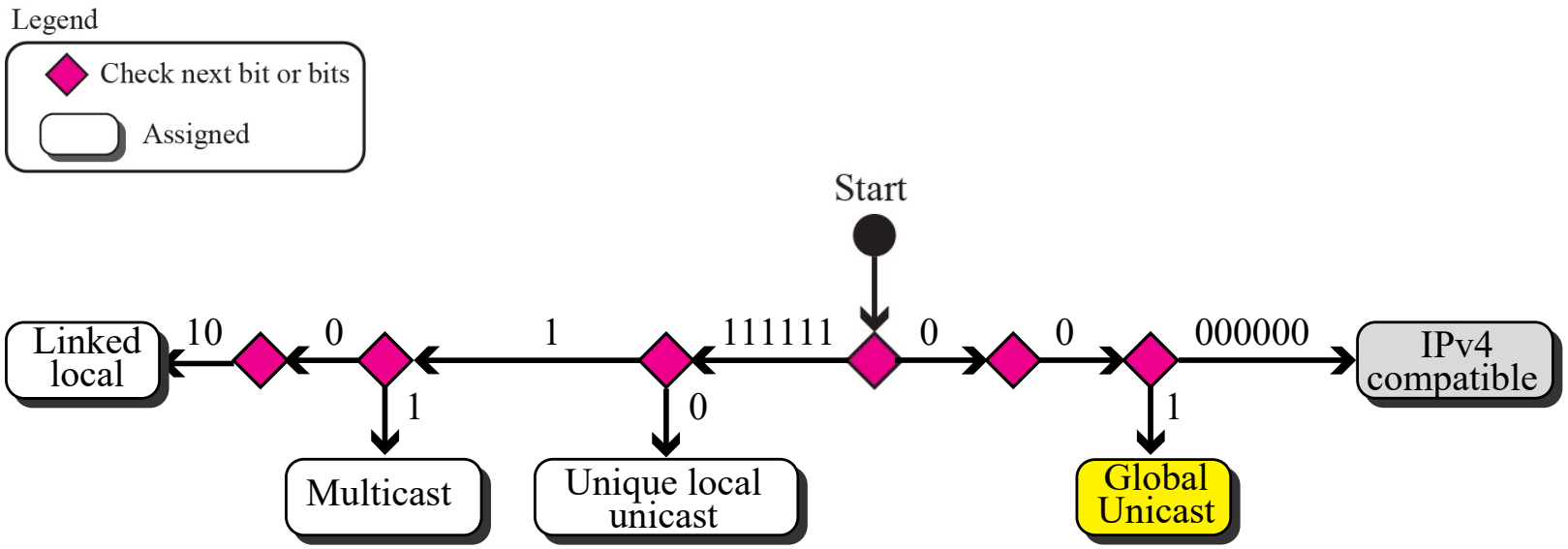
Figure shows that only a portion of the address space can be used for global unicast communication. How many addresses are in this block?

**Solution**

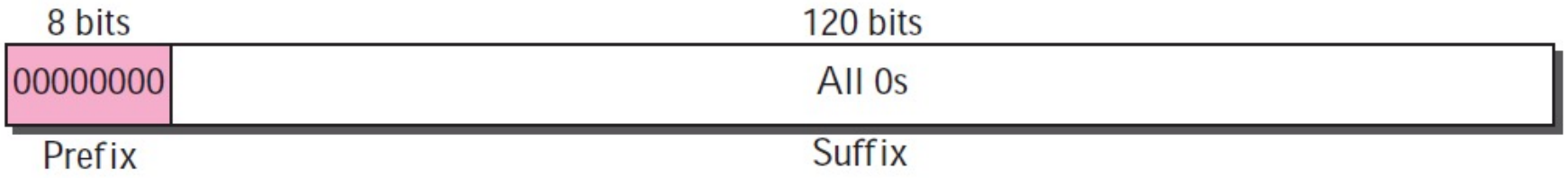
This block occupies only one-eighth of the address spaces. To find the number of addresses, we can divide the total address space by 8 or $2^3$ . The result is $(2^{128})/(2^3) = 2^{125}$ —> a huge block.

# COMPUTER NETWORKS

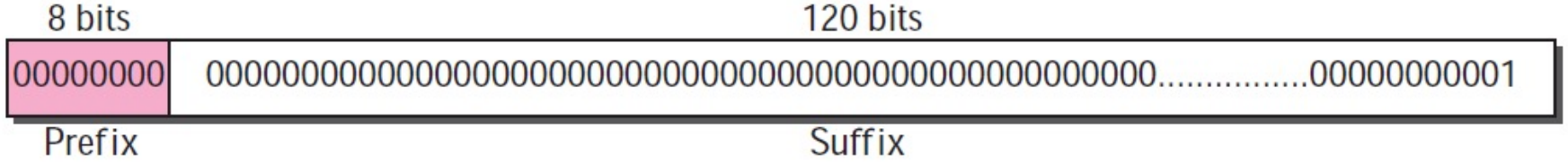
## Algorithm for finding the allocated blocks



## Assigned and Reserved Blocks

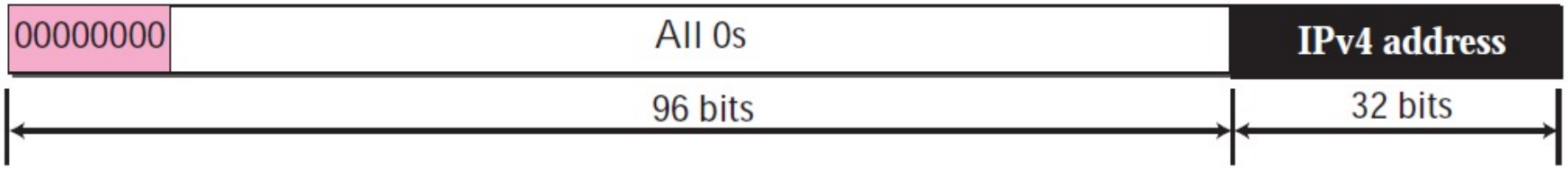



**The unspecified address in IPv6 is ::/128. It should never be used as a destination address.**




**The loopback address in IPv6 is ::1/128. It should never be used as a destination address.**

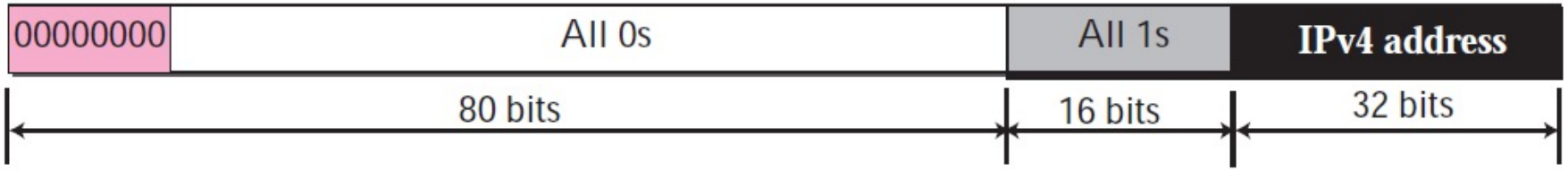
## Assigned and Reserved Blocks




**Embedded IPv4 Addresses – Compatible Address.**

**CIDR notation is ::/96.**

***(Ipv6 -> IPv6)***

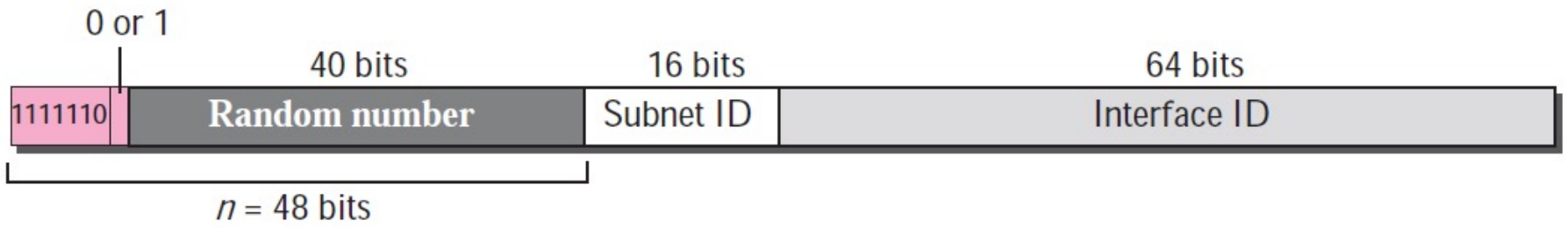



**Embedded IPv4 Addresses – Mapped Address.**

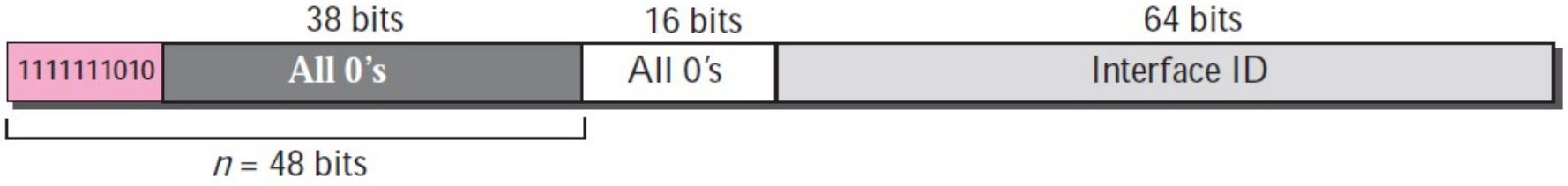
***(Ipv6 -> IPv4)***

## Assigned and Reserved Blocks

### Private Addressing in IPv6 – Site level & Link level

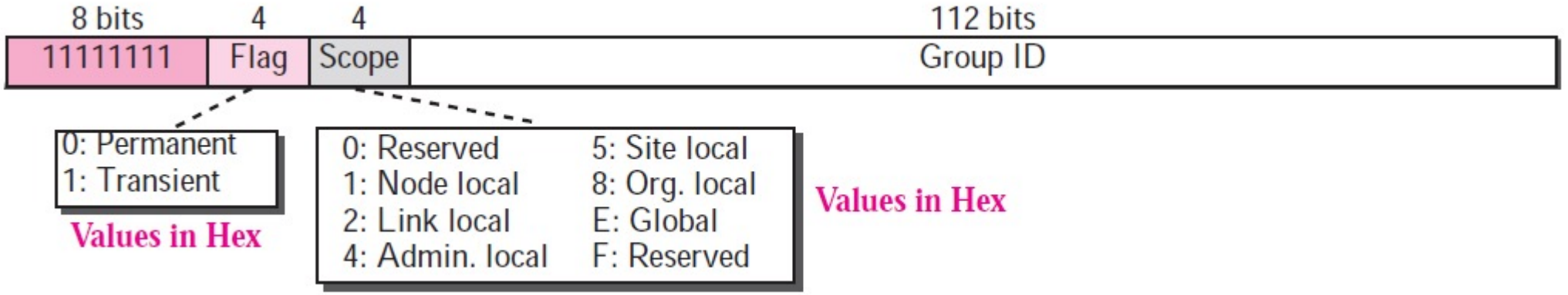



**Unique local unicast block**




**Link local block**

## Assigned and Reserved Blocks

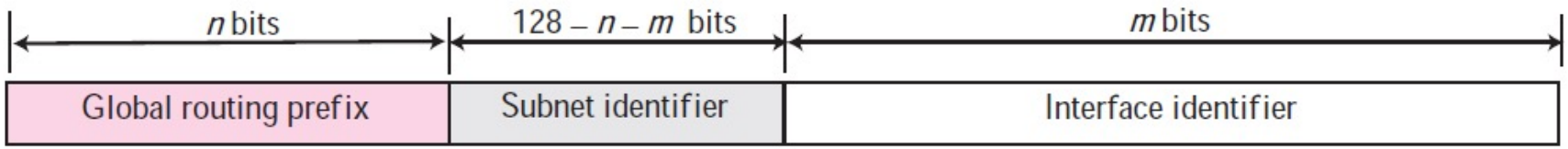



**Multicast address - used to define a group of hosts instead of just one.**

# Global Unicast Addresses

- Used for unicast (one-to-one) communication between two hosts in the Internet.
- CIDR notation for the block is **2000::/3** - three leftmost bits are the same for all addresses (001).
- Three Levels of Hierarchy:
  - Global Routing Prefix
  - Subnet identifier
  - Interface identifier

**Global unicast address**



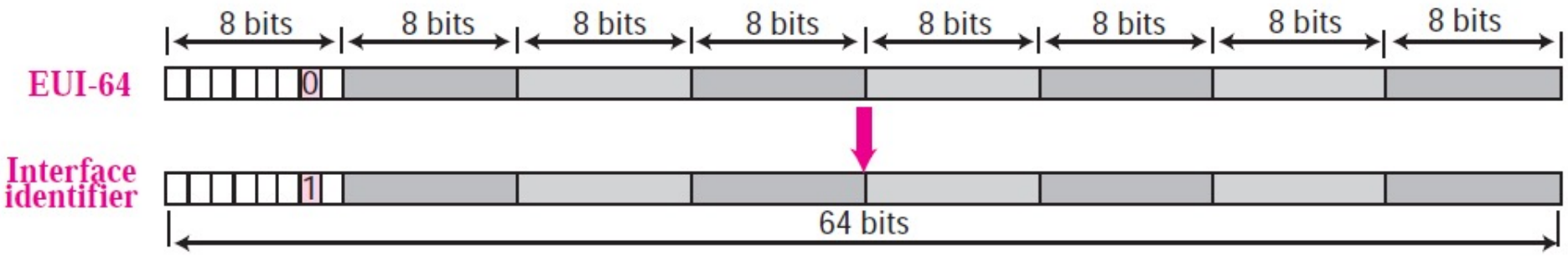

| *Block Assignment* | *Length* |
|---|---|
| Global routing prefix ($n$) | 48 bits |
| Subnet identifier ($128 - n - m$) | 16 bits |
| Interface identifier ($m$) | 64 bits |

***Recommended Length of Different Parts in Unicast Addressing***

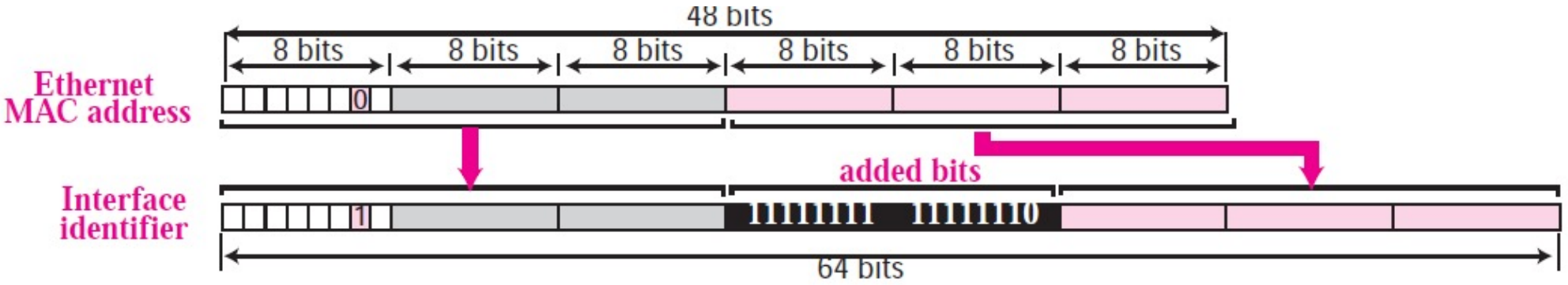
## Global Unicast Addresses

- **Global Routing Prefix** –> 3 bits (001) + 45 bits = $2^{45}$ sites (ISP/private org)
- **Subnet identifier** –> $2^{16}$ = 6553 subnets
- **Interface identifier** –> similar to Host ID in IPv4 (IP level & physical level are different),
  - Physical address is 48 bits while the hostid is less than 32 bits
  - Embed physical address (<64 bits) into Interface ID – whole/part
  - Two physical addressing scheme :
    - 64-bit extended unique identifier (EUI-64) defined by IEEE &
    - 48-bit physical address defined by Ethernet.

# Global Unicast Addresses




**Mapping for EUI-64**




**Mapping for Ethernet MAC (48 bits)**

## Global Unicast Addresses

**1. Find the interface identifier if the physical address in the EUI is $(F5\text{-}A9\text{-}23\text{-}EF\text{-}07\text{-}14\text{-}7A\text{-}D2)_{16}$ using the format we defined for Ethernet addresses.**

***Solution***

We only need to change the seventh bit of the first octet from 0 to 1 and change the format to colon hex notation.
The result is F7A9:23EF:0714:7AD2.

**2. Find the interface identifier if the Ethernet physical address is $(F5\text{-}A9\text{-}23\text{-}14\text{-}7A\text{-}D2)_{16}$ using the format we defined for Ethernet addresses.**

***Solution***

We only need to change the seventh bit of the first octet from 0 to 1, insert two octet FFFE16 and change the format to colon hex notation.
The result is F7A9:23FF:FE14:7AD2 in colon hex.

**An organization is assigned the block 2000:1456:2474/48. What is the CIDR notation for the blocks in the first and second subnets in this organization?**

***Solution***

Theoretically, the first and second subnets should use the block with subnet identifier $0001_{16}$ and $0002_{16}$. This means that the blocks are 2000:1456:2474:0000/64 and 2000:1456:2474:0001/64.

**An organization is assigned the block 2000:1456:2474/48. What is the IPv6 address of an interface in the third subnet if the IEEE physical address of the computer is (F5-A9-23-14-7A-D2)$_{16}$?**

***Solution***

The interface identifier is F7A9:23FF:FE14:7AD2. If we add this identifier to the global prefix and the subnet identifier, we get: 2000:1456:2474:0002:F7A9:23FF:FE14:7AD2/128

## Autoconfiguration

- In IPv6, a host can also configure itself (DHCP can still be used to allocate an IPv6 address to a host.)
  1. The host first creates a **link local address** for itself.
     - 10-bit link local prefix (1111 1110 10) + 54 zeros + 64-bit interface identifier.
  2. The host then tests to see if this link local address is unique.
     - 64-bit interface identifier is supposed to be unique (high probability)
     - To confirm, sends a *neighbor solicitation message*, and waits for *neighbor advertisement message*. Fails -> DHCP.
  3. Pass -> host stores this address as its link-local address (for private communication), prepare its global unicast address.
     - Sends a *router solicitation message*, receives a *router advertisement message*

**Assume a host with Ethernet address $(F5\text{-}A9\text{-}23\text{-}11\text{-}9B\text{-}E2)_{16}$ has joined the network. What would be its global unicast address if the global unicast prefix of the organization is 3A21:1216:2165 and the subnet identifier is A245?**

## Solution

The host first creates its interface identifier as F7A9:23FF:FE11:9BE2
using the Ethernet address read from its card, the host then creates its link-local address as: FE80::F7A9:23FF:FE11:9BE2

- Assuming that this address is unique, the host sends a router solicitation message and receives the router advertisement message that announces the combination of global unicast prefix and the subnet identifier as 3A21:1216:2165:A245:1232.
- The host then appends its interface identifier to this prefix to find and store its global unicast address as: A21:1216:2165:A245:F7A9:23FF:FE11:9BE2

## Renumbering

- Renumbering of the address prefix (n) is built in IPv6 - to allow sites to change the service provider.
- Each site – prefix by service provider.
- If the site changes provider, address prefix need to be changed.
- Router – assign new prefix, use the existing prefix for a short time before disabling it (two prefixes – during transition time).
- Next Generation DNS – under study.

## Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

4.5 Introduction to Network Layer Protocols

- DHCP
- ICMP

4.6 IPv6 Addressing

4.7 IPv6 Protocol

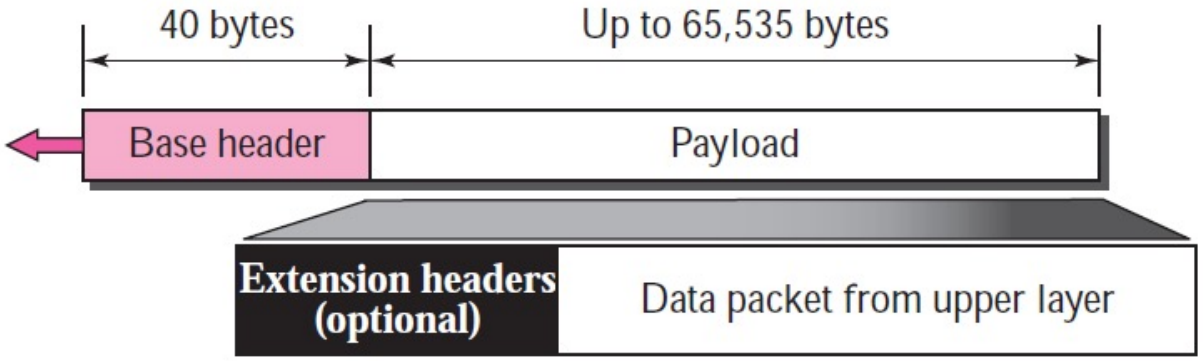
4.8 Introduction to Routing Algorithms

# IPv6 Protocol: Rationale for Change

- Larger address space
- Better header format
- New options
- Allowance for extension
- Support for resource allocation
- Support for more security

*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
fragmentation/reassembly: no

# IPv6 Datagram Format



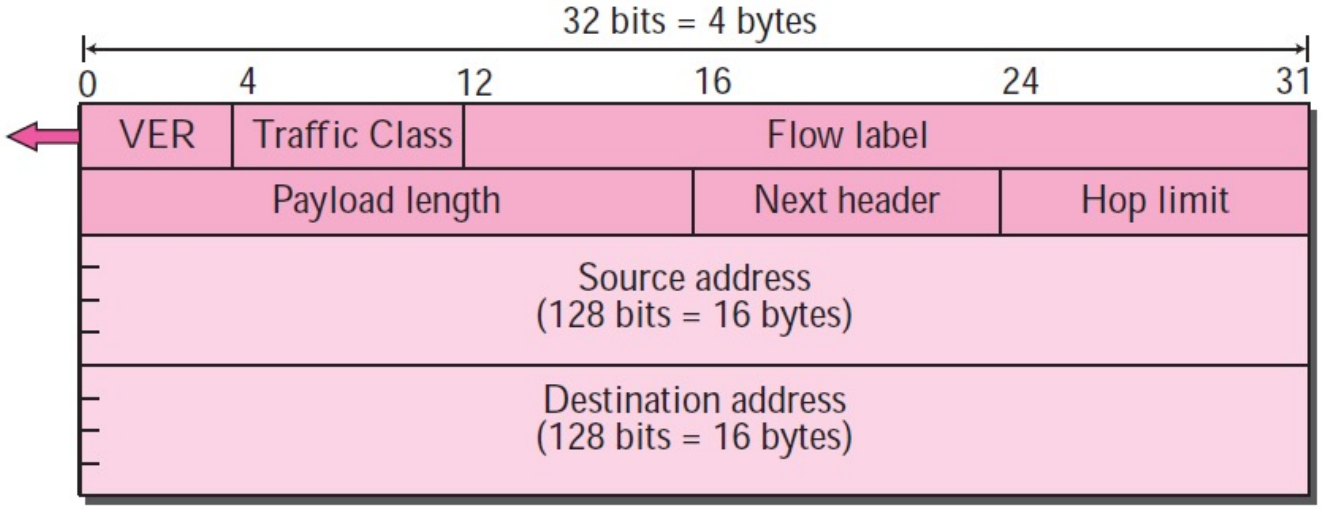




**Table 27.1** *Next Header Codes*

| *Code* | *Next Header* | *Code* | *Next Header* |
|---|---|---|---|
| 0 | Hop-by-hop option | 44 | Fragmentation |
| 2 | ICMP | 50 | Encrypted security payload |
| 6 | TCP | 51 | Authentication |
| 17 | UDP | 59 | Null (No next header) |
| 43 | Source routing | 60 | Destination option |

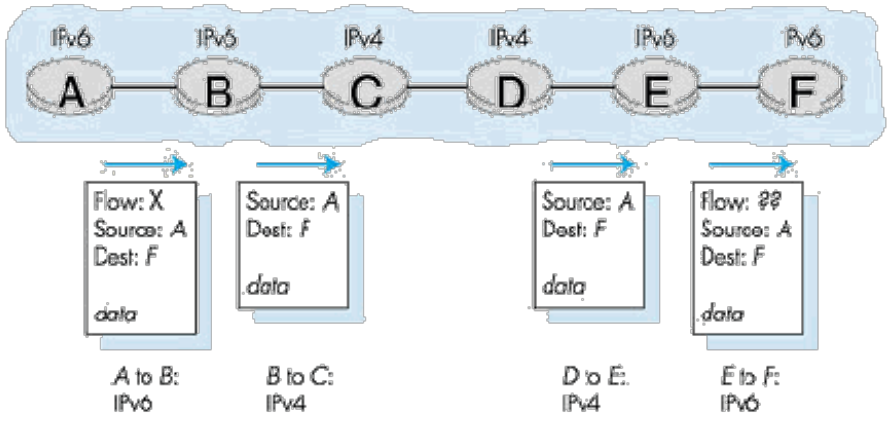
**Format of the base header**

# COMPUTER NETWORKS

## Transition from IPv4 to IPv6

- not all routers can be upgraded simultaneously
  - no “flag days”
  - how will network operate with mixed IPv4 and IPv6 routers?

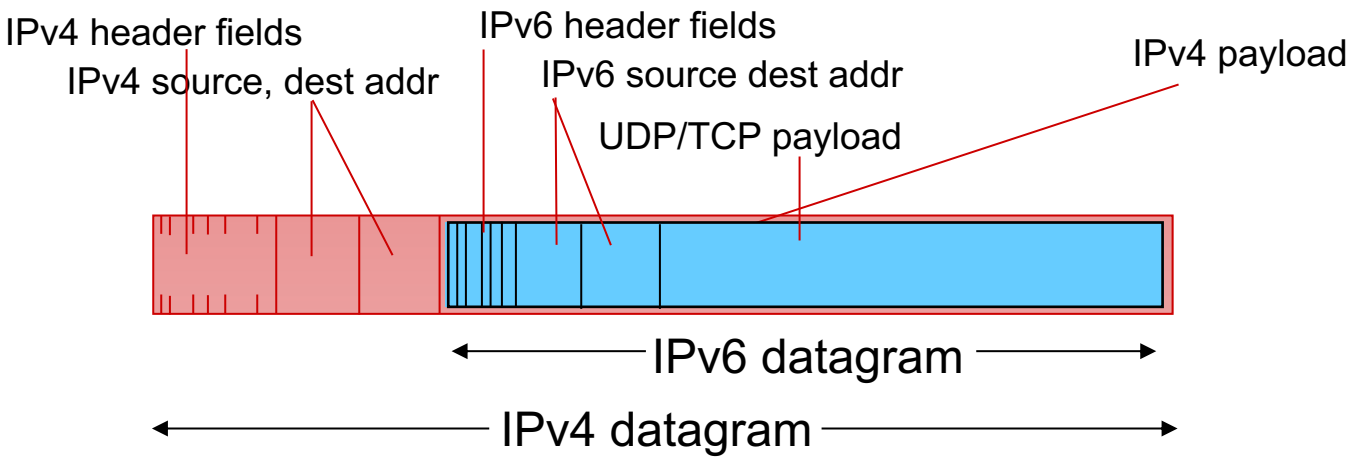
### Dual Stack Approach



# Transition from IPv4 to IPv6

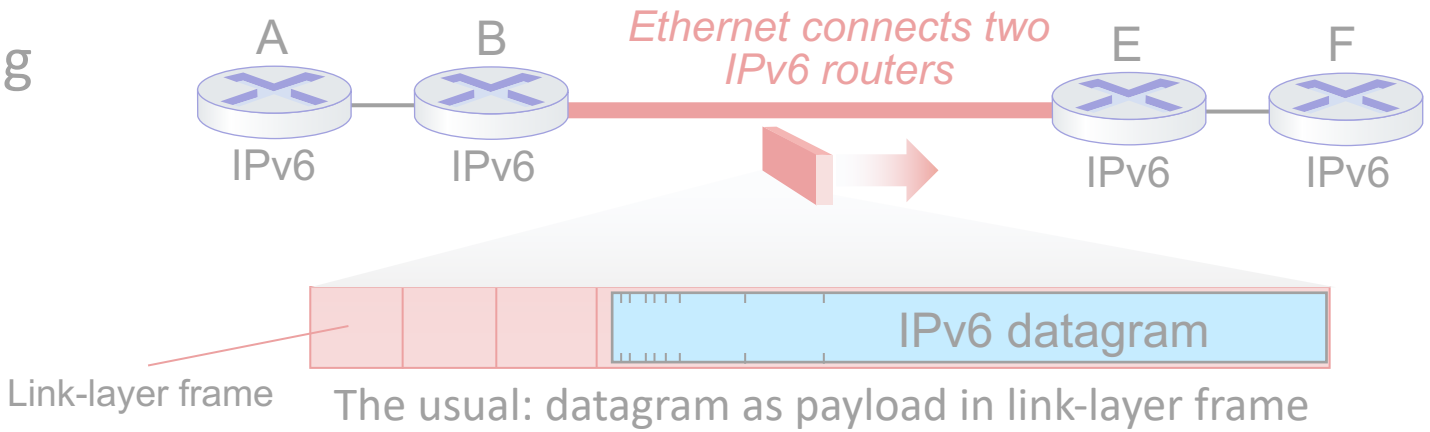
- not all routers can be upgraded simultaneously
  - no “flag days”
  - how will network operate with mixed IPv4 and IPv6 routers?

- tunneling: IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers (“packet within a packet”)
  - tunneling used extensively in other contexts (4G/5G)

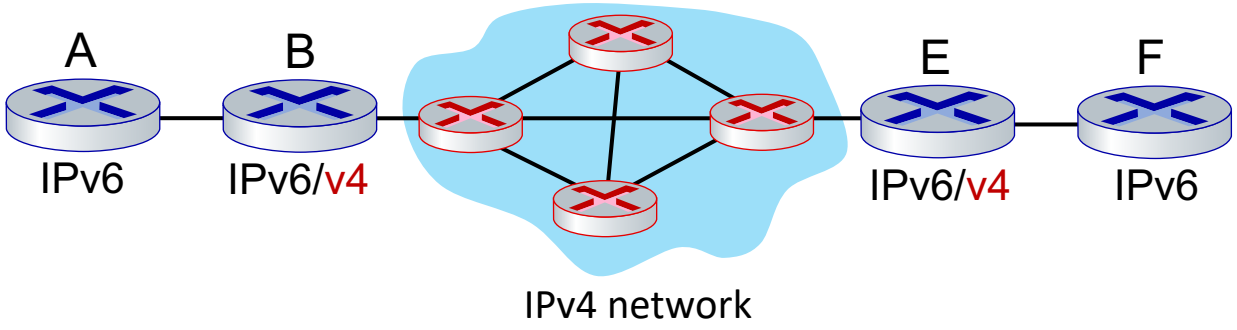


# Tunneling and Encapsulation

Ethernet connecting two IPv6 routers:

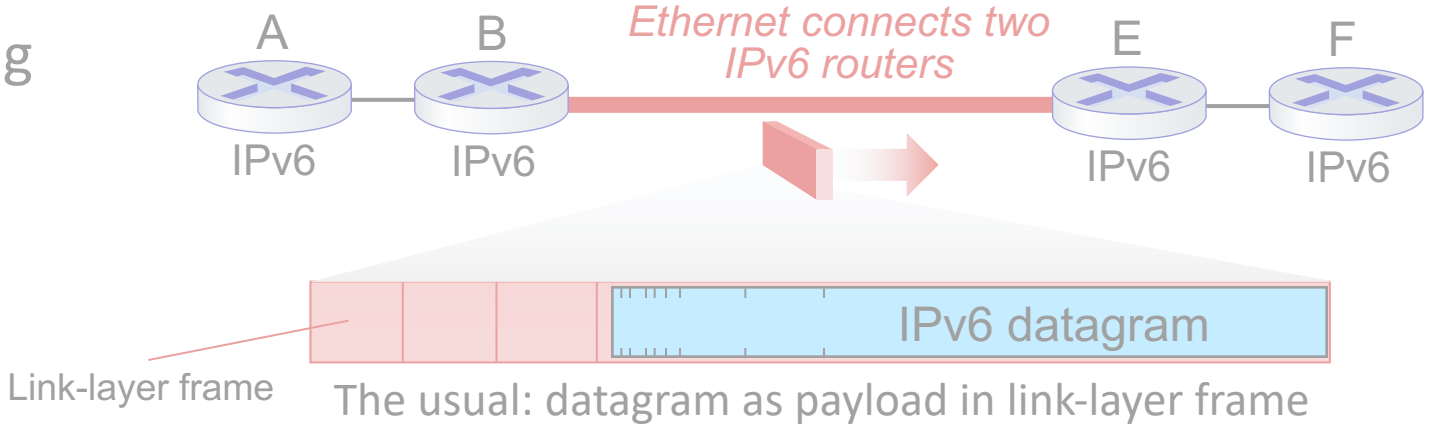



IPv4 network connecting two IPv6 routers

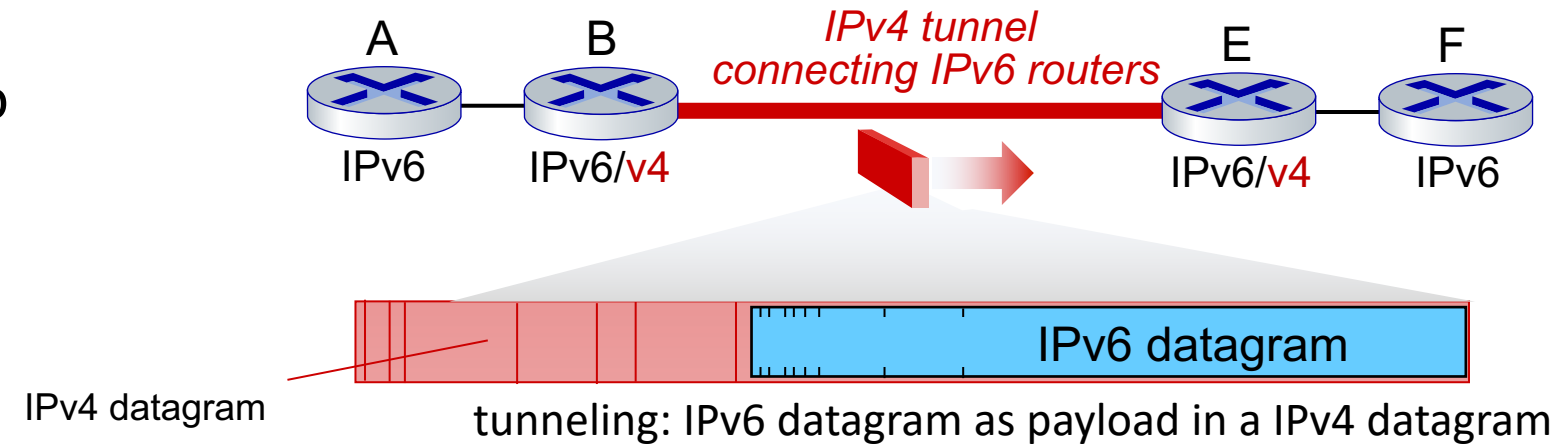


# Tunneling and Encapsulation

Ethernet connecting two IPv6 routers:

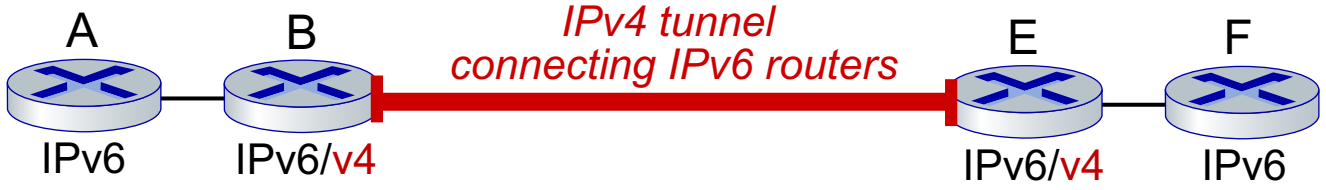



IPv4 tunnel connecting two IPv6 routers

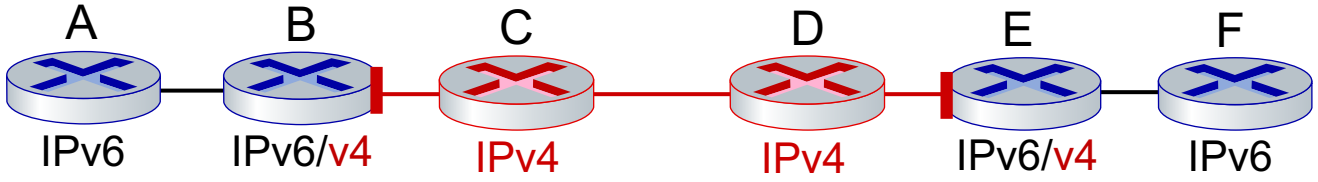


# Tunneling

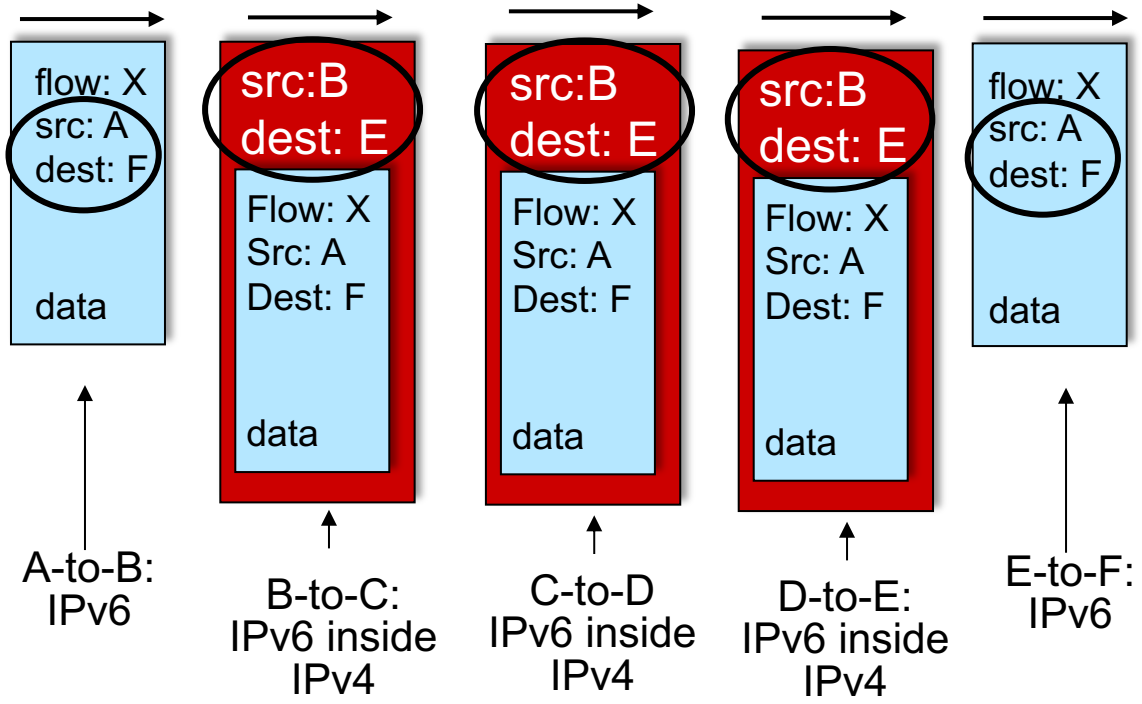
logical view:




physical view:




Note source and destination addresses!



# IPv6 Adoption

https://www.akamai.com/us/en/resources/our-thinking/state-of-the-internet-report/state-of-the-internet-ipv6-adoption-visualization.jsp

## Unit – 4 Network Layer and Internet Protocol

4.1 Overview of Network Layer

4.2 What's Inside a Router?

4.3 Switching

4.4 The Internet Protocol (IP)

- Datagram format
- Fragmentation
- IPv4 addressing
- NAT

4.5 Introduction to Network Layer Protocols

- DHCP
- ICMP

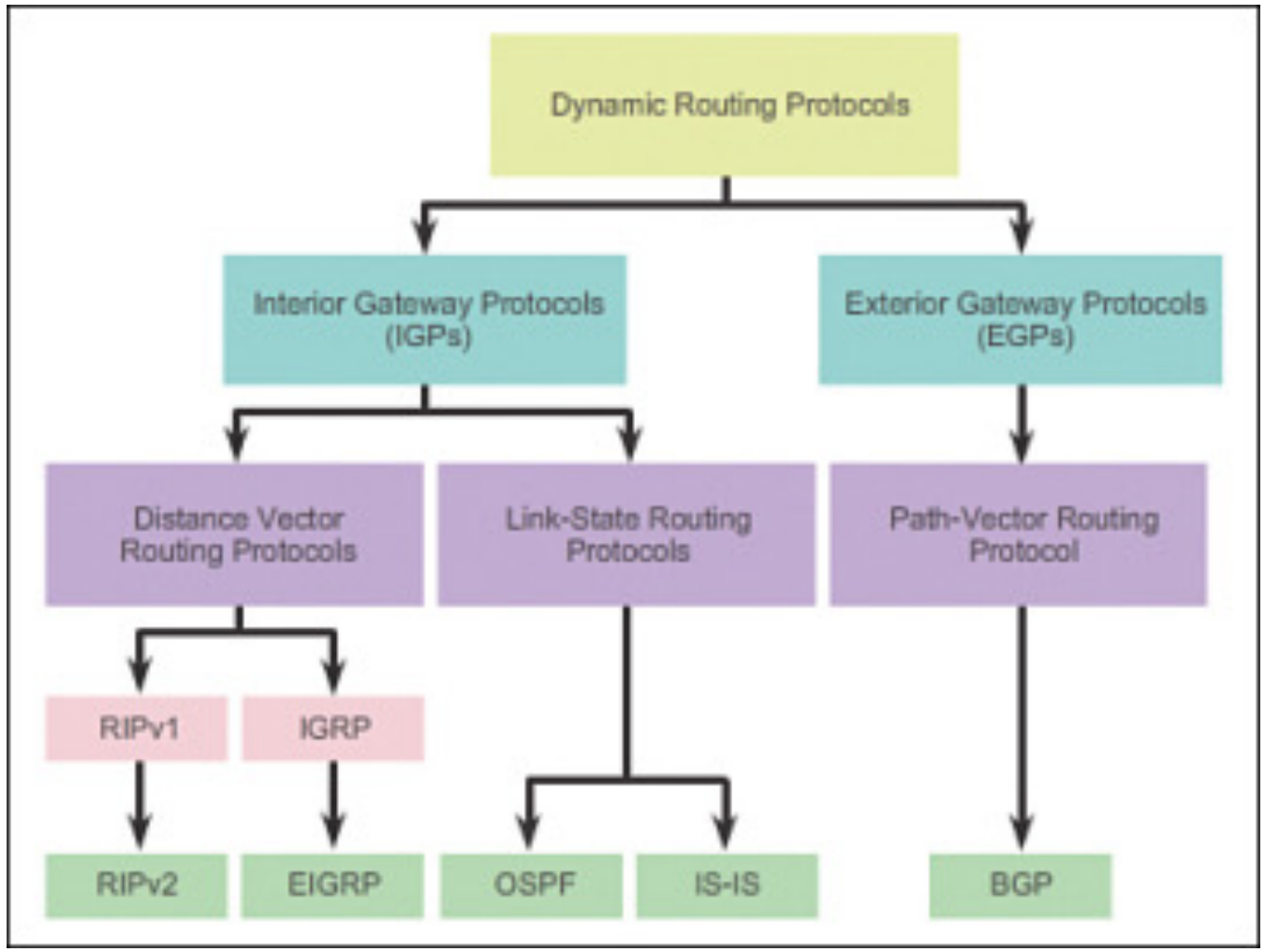
4.6 IPv6 Addressing

4.7 IPv6 Protocol

4.8 Introduction to Routing Algorithms

# COMPUTER NETWORKS

## Introduction to Routing Algorithms



# Distance Vector vs Link State

| | Distance Vector | Link State |
|---|---|---|
| **Primary principle** | Send entire routing table to its neighbors | Only provides link state information |
| **Learning about network** | Learn about network only from neighbors | Learn about network from all routers |
| **Building the routing table** | Based on inputs from only neighbors | Based on complete database collected from all routers |
| **Advertisement of updates** | Sends periodic updates every 30-90 seconds – **Broadcasts updates** | Use triggered updates, only when there is a change – **Multicasts updates** |
| **Routing loops** | Vulnerable | Less prone to routing loops |

# Distance Vector vs Link State

| | Distance vector | Link State |
|---|---|---|
| Convergence (stabilization) | Slow | Fast |
| Resources | Less CPU power and memory | More CPU power and memory required |
| Cost | | More than Distance vector |
| Scalability | | More scalable than distance vector |
| Examples | RIP, IGRP | OSPF, IS-IS |


# THANK YOU

**Ashwini M Joshi**
Department of Computer Science and Engineering

**ashwinimjoshi@pes.edu**