

# CS2105

## An $\text{\AA}\omega\epsilon\sigma\omicron\mu\epsilon$ Introduction to Computer Networks

Lectures 6&7: The Network Layer



Department of Computer Science  
School of Computing

# Lectures 6&7: The Network Layer

*After this class, you are expected to understand:*

- ❖ the basic services network layer provides.
- ❖ the purpose of DHCP and how it works.
- ❖ IP address, subnet, subnet mask and address allocation.
- ❖ how longest prefix forwarding in a router works.
- ❖ the purpose of routing protocols on the Internet.
- ❖ the principle of Bellman-Ford equation.
- ❖ the workings of distance vector algorithm.
- ❖ the purpose of NAT and how it works.
- ❖ the Internet Protocol (IP) and how datagram fragmentation works.

# Lectures 6&7: Roadmap

## 4.1 Overview of Network Layer

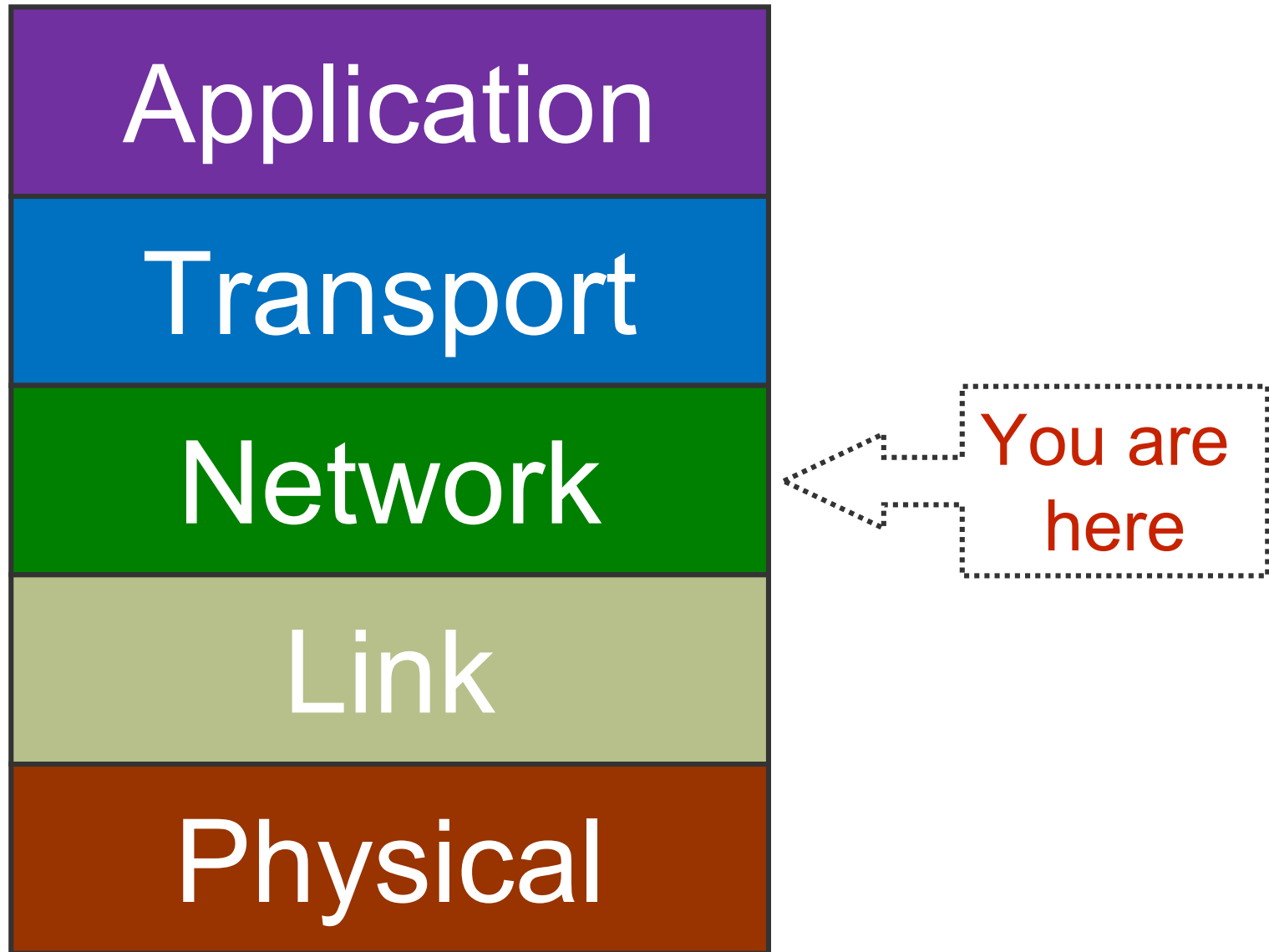
## 4.2 What's Inside a Router

## 4.3 The Internet Protocol (IP)

## 5.2 Routing Algorithms

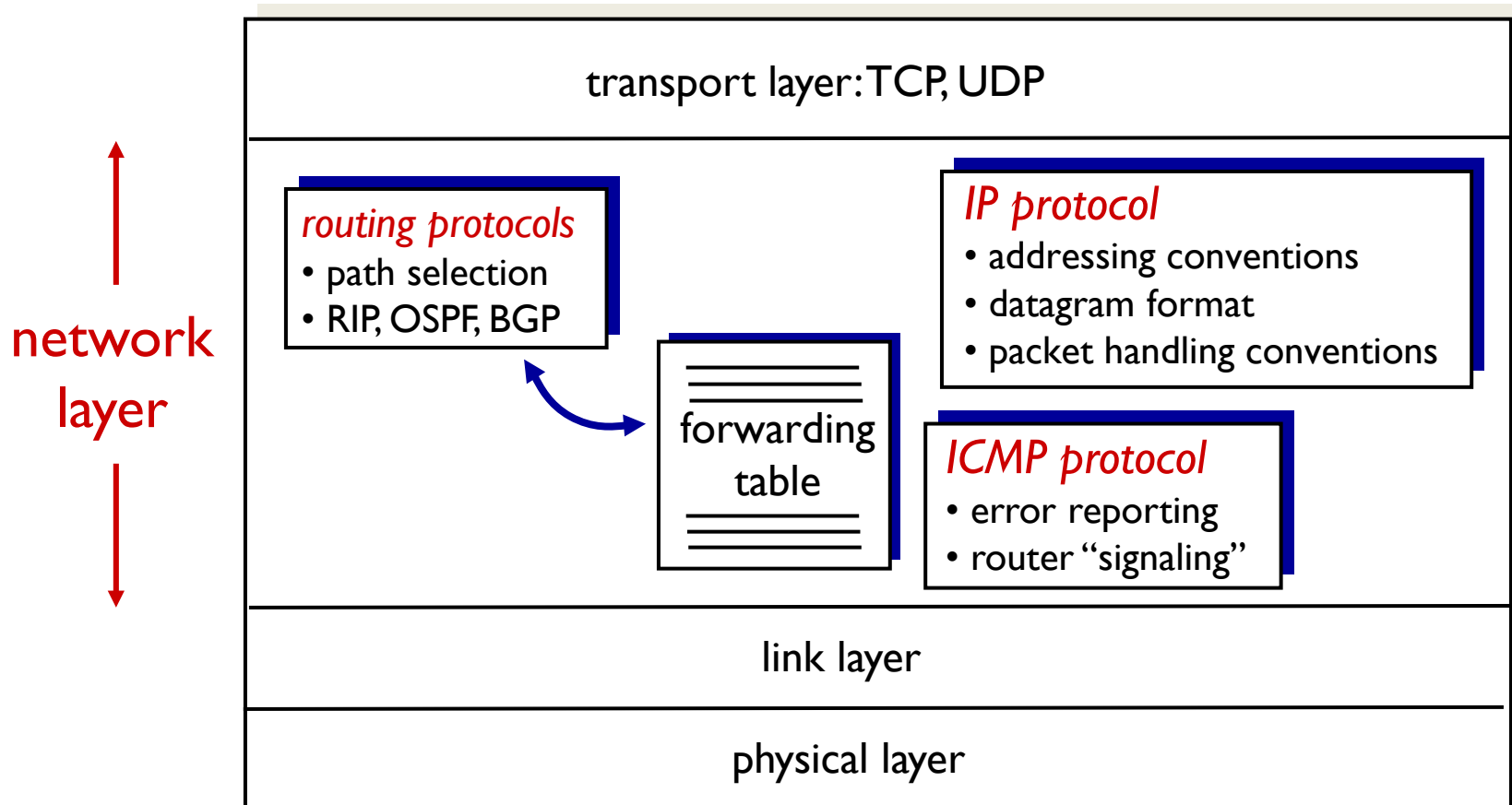
## 5.6 ICMP

Kurose Textbook, Chapters 4&5  
(Some slides are taken from the book)



# Network Layer Services

- ❖ Network layer delivers packets to receiving hosts.
  - Routers examine header fields of IP datagrams passing it.



# Lectures 6&7: Roadmap

4.1 Overview of Network Layer

4.2 What's Inside a Router

- 4.2.1 Destination-Based Forwarding

4.3 The Internet Protocol (IP)





- 4.3.3 IPv4 Addressing

5.2 Routing Algorithms

5.6 ICMP

# IP Address

- ❖ **IP address** is used to identify a host (or a router).
  - A 32-bit integer expressed in either binary or decimal

Binary:	00000001	00000010	00000011	10000001
				
Decimal:	1	2	3	129

IP address use dotted decimal notation

- ❖ How does a host get an IP address?
  - manually configured by system administrator, or
  - automatically assigned by a **DHCP (Dynamic Host Configuration Protocol)** server.

# Dynamic Host Configuration Protocol

- ❖ **DHCP** allows a host to dynamically obtain its IP address from DHCP server when it joins network.
  - IP address is **renewable**
  - allow reuse of addresses (only hold address while connected)
  - support mobile users who want to join network.
- ❖ **DHCP**: 4-step process:
  - 1) Host broadcasts **“DHCP discover”** message
  - 2) DHCP server responds with **“DHCP offer”** message
  - 3) Host requests IP address: **“DHCP request”** message
  - 4) DHCP server sends address: **“DHCP ACK”** message



Arriving  
client



DHCP server  
223.1.2.5

all the messages will be broadcasted on the network -  
since there might be many DHCP servers, thus need to  
broadcast which offer is chosen so that the other DHCP  
servers can retract their offer and save that IP address

**DHCP discover**

special IP

src : 0.0.0.0, 68  
dest: 255.255.255.255, 67  
yiaddr: 0.0.0.0  
transaction ID: 654

broadcast  
address

**DHCP offer**

src: 223.1.2.5, 67  
dest: 255.255.255.255, 68  
yiaddr: 223.1.2.4  
transaction ID: 654  
lifetime: 3600 secs

Your IP  
address

**DHCP request**

src: 0.0.0.0, 68  
dest: 255.255.255.255, 67  
yiaddr: 223.1.2.4  
transaction ID: 655  
lifetime: 3600 secs

**DHCP ACK**



src: 223.1.2.5, 67  
dest: 255.255.255.255, 68  
yiaddr: 223.1.2.4  
transaction ID: 655  
lifetime: 3600 secs

transaction ID is so that you know which reply from server  
is for your request

# More on DHCP

- ❖ In addition to host IP address assignment, DHCP may also provide a host additional network information:
  - IP address of first-hop router
  - IP address of local DNS server
  - Network mask (indicating network prefix versus host ID of an IP address)
  
- ❖ DHCP runs over UDP
  - DHCP server port number: 67
  - DHCP client port number: 68

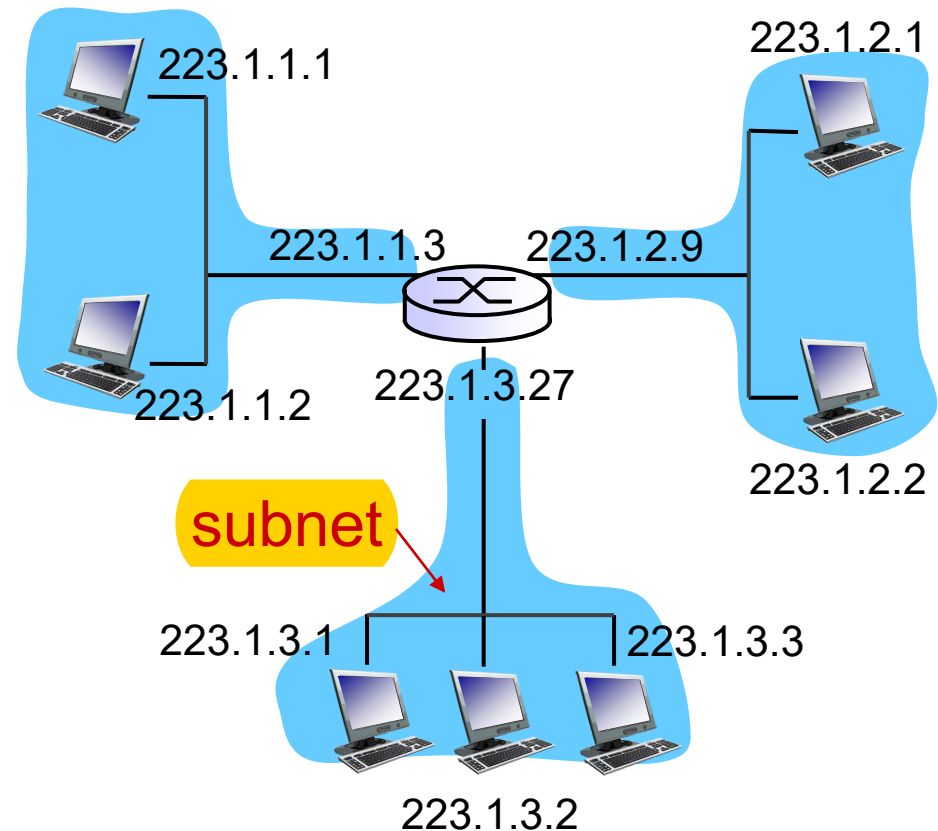
# Some Special IP Addresses

Special Addresses	Present Use
0.0.0.0/8	Non-routable meta-address for special use
 127.0.0.0/8 172.16.0.0 10.0.0.0 172.25.1.1 1010 1100 0001 1100 0000 0001 0000 0001	 oopback address. A datagram sent to an address within this block loops back inside the host. This is ordinarily implemented using only 127.0.0.1/32.
10.0.0.0/8 172.16.0.0/12 192.168.0.0/16	Private addresses, can be used without any coordination with IANA or an Internet registry.
255.255.255.255/32	Broadcast address. All hosts on the same subnet receive a datagram with such a destination address.

The full list of special IP addresses can be found in RFC5735:  
<https://tools.ietf.org/rfc/rfc5735.txt>

# IP Address and Network Interface

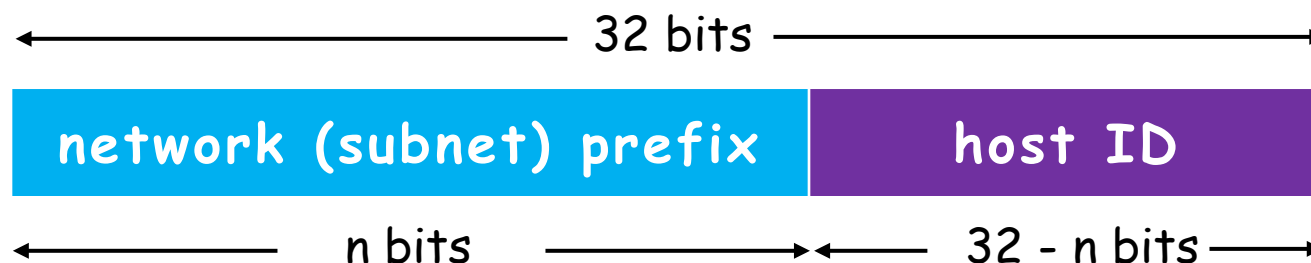
- ❖ An IP address is associated with a **network interface**.
  - A host usually has one or two network interfaces (e.g. wired Ethernet and WiFi).
  - A router typically has multiple interfaces.





A network consisting of 3 subnets  
(first 24 bits of IP addr. are network prefix)

# IP Address and Subnet

- ❖ An IP address logically comprises two parts:



- ❖ **Subnet** is a network formed by a group of “directly” interconnected hosts.
  - Hosts in the same subnet have the same network prefix of IP address.
  - Hosts in the same subnet can physically reach each other without intervening router.   $\rightarrow$  1<sup>st</sup> hop router  $\leftarrow$  
  - They connect to the outside world through a router.

# IP Address: CIDR

- ❖ The Internet's IP address assignment strategy is known as **Classless Inter-domain Routing (CIDR)**.
  - Subnet prefix of IP address is of arbitrary length.
  - Address format: **a.b.c.d/x**, where **x** is the number of bits in subnet prefix of IP address.

← subnet prefix → ← host ID →  
 11001000 00010111 00010000 00101010

this subnet contains  $2^9$  IP addresses  
 subnet prefix: 200.23.16.42/23

/23 indicates the no. of  
 bits of subnet prefix

# Subnet Mask

- ❖ **Subnet mask** is used to determine which subnet an IP address belongs to.
  - made by setting all subnet prefix bits to "1"s and host ID bits to "0"s.
- ❖ Example: for IP address **200.23.16.42/23**:

	<div style="display: flex; align-items: center; justify-content: space-around;"> <span>←</span> <span>subnet prefix</span> <span>→</span> <span>←</span> <span>host ID</span> <span>→</span> </div>			
IP address in binary	11001000	00010111	00010000	00101010
Subnet mask	11111111	11111111	11111110	00000000
Subnet mask in decimal	255.255.254.0			

# Quiz

subnet prefix allows only 6 bits for host id - thus it should end with 10

❖ For the following 4 IP addresses, which one is in a different subnet from the rest 3?

a. 172.26.185.128/26

b. 172.26.185.130/26

c. 172.26.185.160/26

d. 172.26.185.192/26



# IP Address Allocation

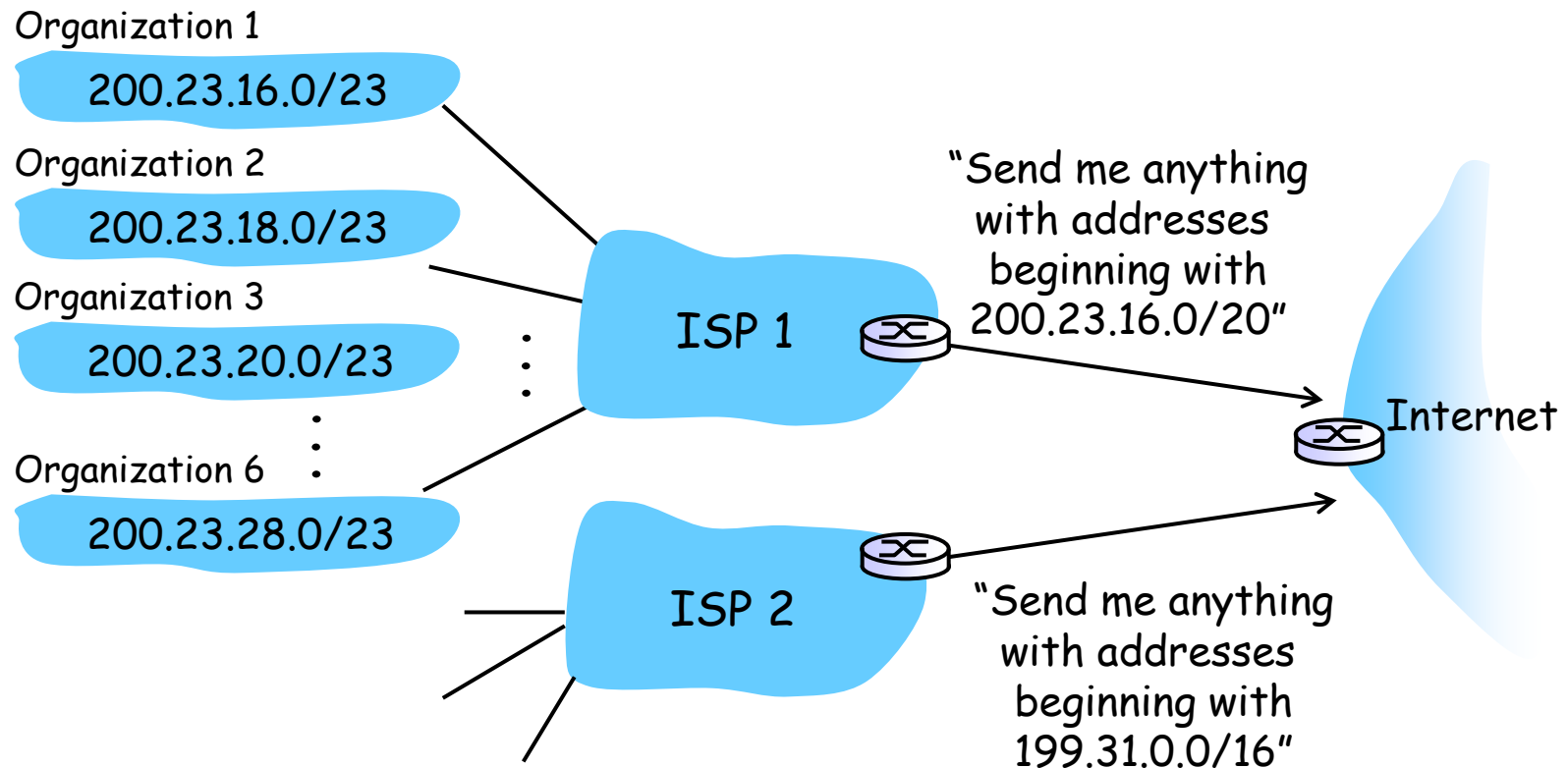
- ❖ **Q:** How does an organization obtain a block of IP addresses?
- ❖ **A:** Buy from registry or rent from ISP's address space.

	Binary Address	Decimal Address
ISP's block	11001000 00010111 0001 000 0 00000000	200.23.16.0/20
Organization 1	11001000 00010111 0001 000 0 00000000	200.23.16.0/23
Organization 2	11001000 00010111 0001 001 0 00000000	200.23.18.0/23
Organization 3	11001000 00010111 0001 010 0 00000000	200.23.20.0/23
...	...	...
Organization 6	11001000 00010111 0001 101 0 00000000	200.23.28.0/23

use 3 more bits to differentiate  
6 organizations

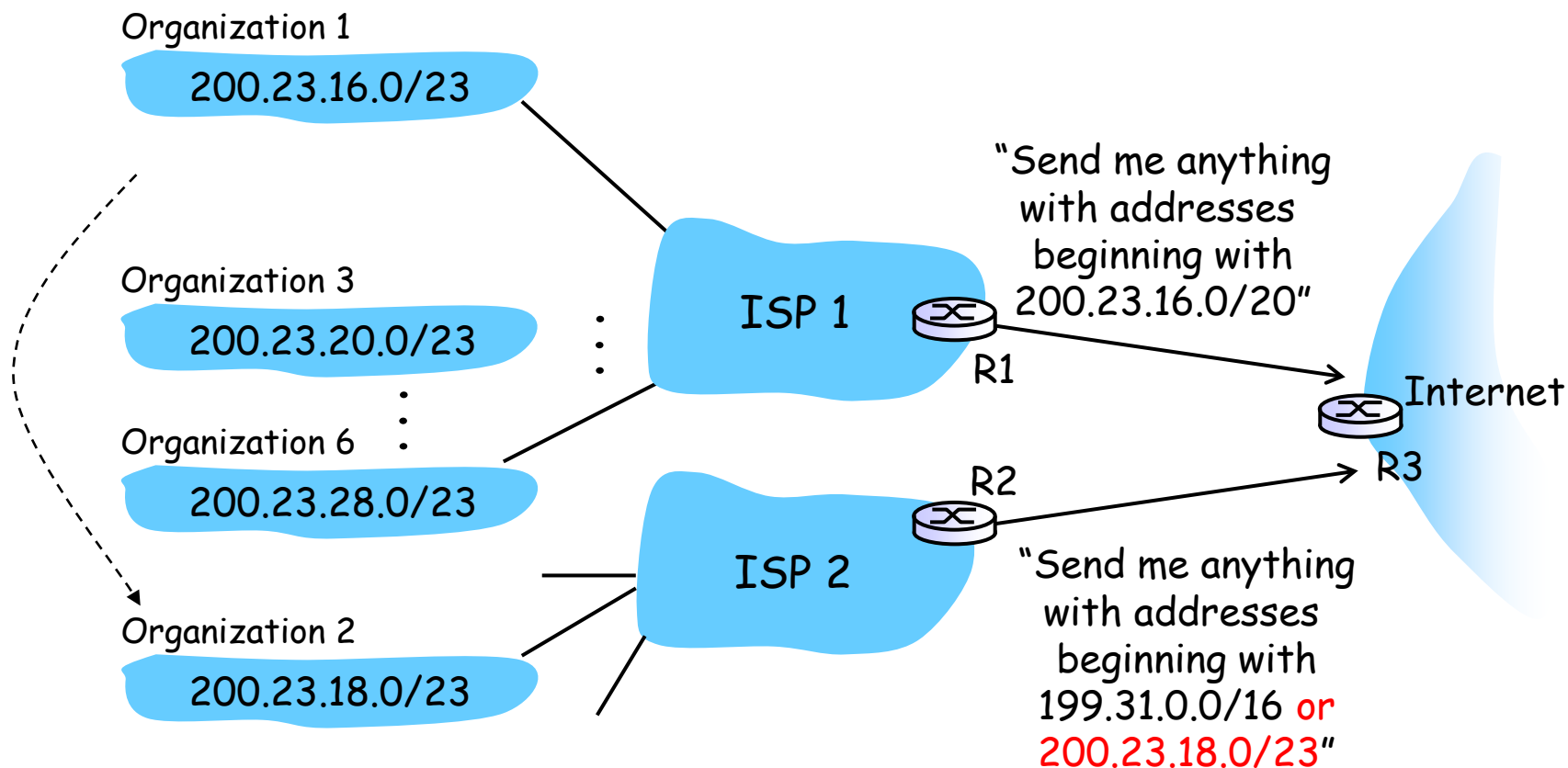
# Hierarchical Addressing

Hierarchical addressing allows efficient advertisement of routing information:



# Hierarchical Addressing

Suppose Organization 2 now switches to ISP 2, but doesn't want to renumber all of its routers and hosts.

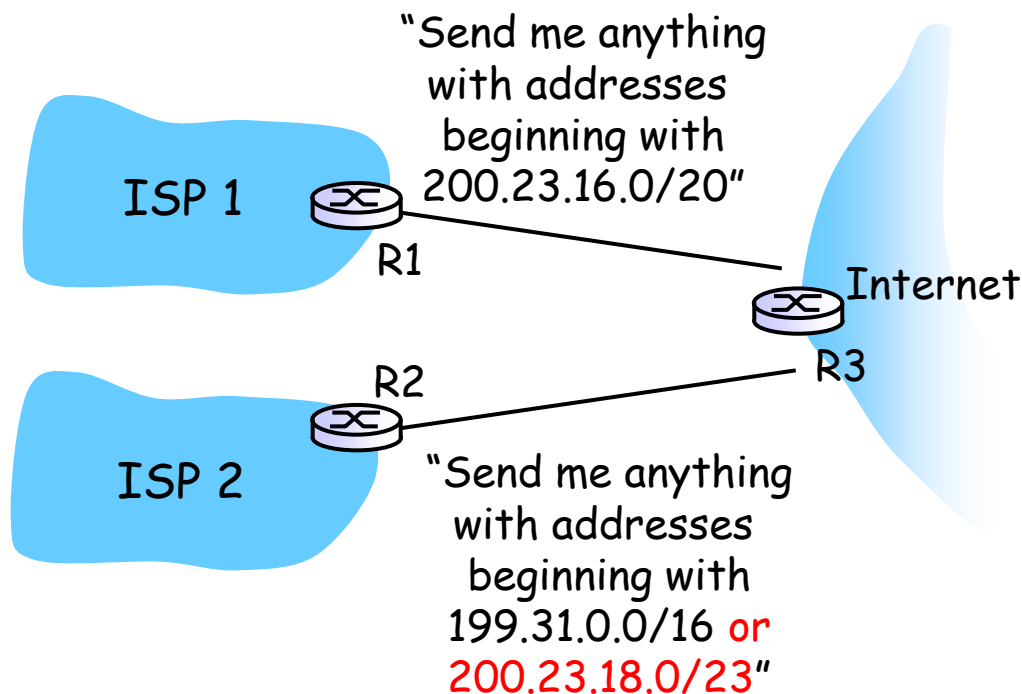


# Longest Prefix Match (1/2)

- ❖ **Question:** which router to deliver to,
- if a packet has destination IP **200.23.20.2**?
  - if a packet has destination IP **200.23.19.3**?

Forwarding Table at R3

Net mask	Next hop
200.23.16.0/20	R1
200.23.18.0/23	R2
199.31.0.0/16	R2
...	...



# Longest Prefix Match (2/2)

- ❖ Packet with destination IP **200.23.20.2** ➡ R1
  - (Binary: **11001000** **00010111** **00010100** 00000010)
- ❖ Packet with destination IP **200.23.19.3** ➡ R2
  - (Binary: **11001000** **00010111** **00010011** 00000011)

Forwarding Table at R3

Net mask	Net mask in binary	Next hop
200.23.16.0/20	<b>11001000</b> <b>00010111</b> <b>00010000</b> 00000000	R1
200.23.18.0/23	<b>11001000</b> <b>00010111</b> <b>00010010</b> 00000000	R2
199.31.0.0/16	<b>11000111</b> <b>00011111</b> 00000000 00000000	R2
...		...

match the  
longest prefix

# More on IP Address Allocation

- ❖ **Q1:** How does an organization obtain a block of IP addresses?
- ❖ **A1:** Buy from registry or rent from ISP's address space.
- ❖ **Q2:** How does an ISP get a block of addresses?
- ❖ **A2:** ICANN: Internet Corporation for Assigned Names and Numbers
  - Allocates addresses
  - Manages DNS
  - Assigns domain names, resolves disputes

# Lectures 6&7: Roadmap

4.1 Overview of Network Layer

4.2 What's Inside a Router

4.3 The Internet Protocol (IP)

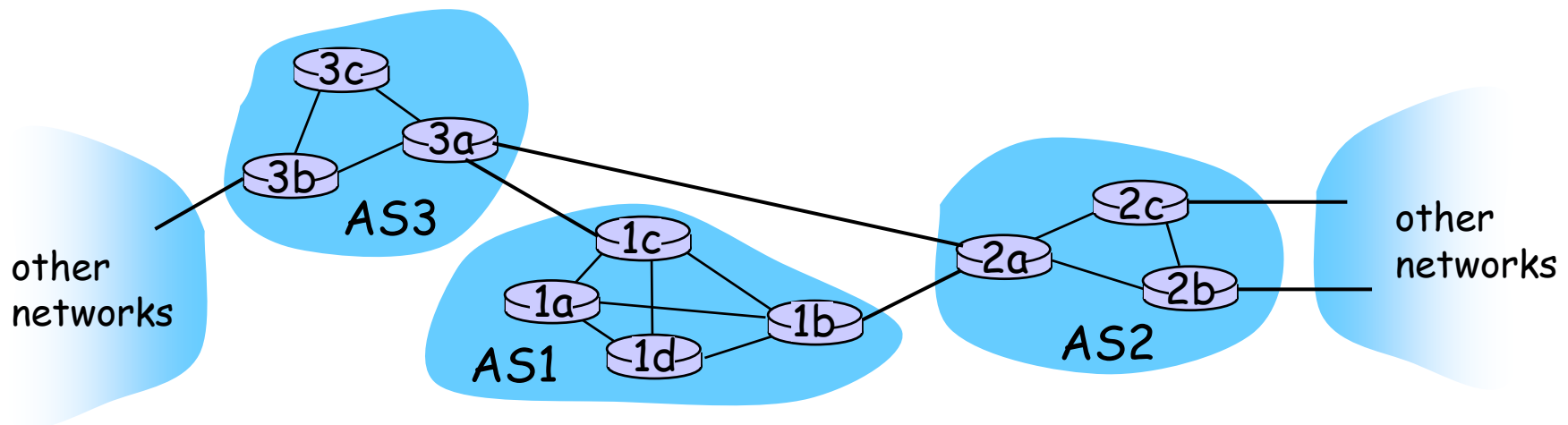
5.2 Routing Algorithms

- 5.2.2 The Distance Vector Routing Algorithm

5.6 ICMP

# Internet: Network of Networks

- ❖ The Internet is a “network-of-networks”.
  - A hierarchy of Autonomous Systems (AS), e.g., ISPs, each owns routers and links.
- ❖ Due to the size of the Internet and the decentralized administration of the Internet, routing on the Internet is done hierarchically.





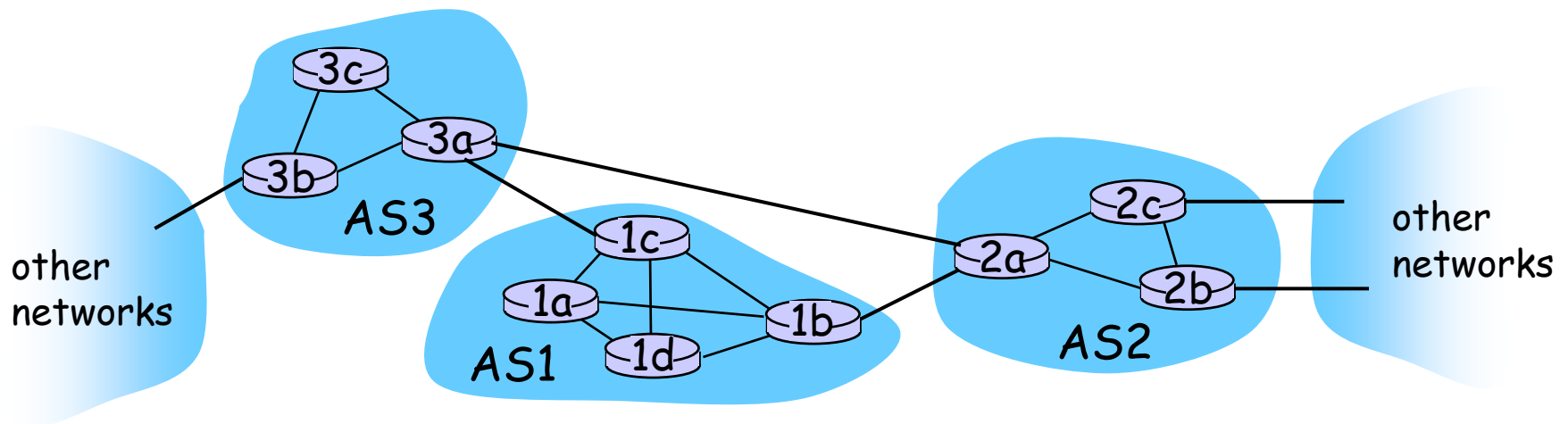
# Routing in The Internet

## ❖ Intra-AS routing

- Finds a good path between two routers within an AS.
- Commonly used protocols: **RIP**, **OSPF**

## ❖ Inter-AS routing (not covered)

- Handles the interfaces between ASs.
- The de facto standard protocol: **BGP**



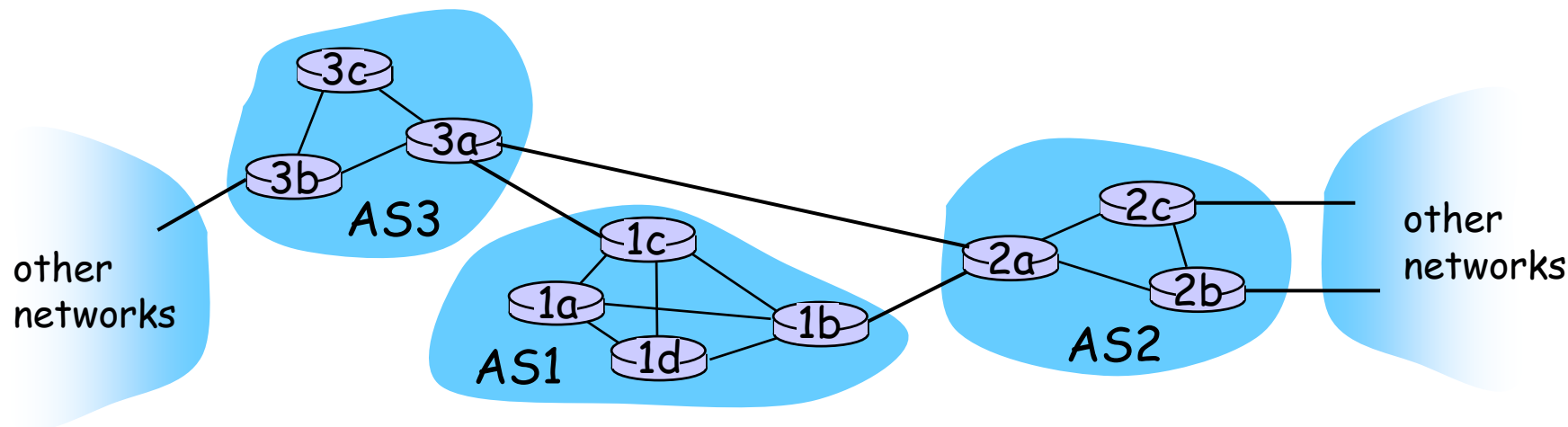
# Routing in The Internet

## ❖ Intra-AS routing

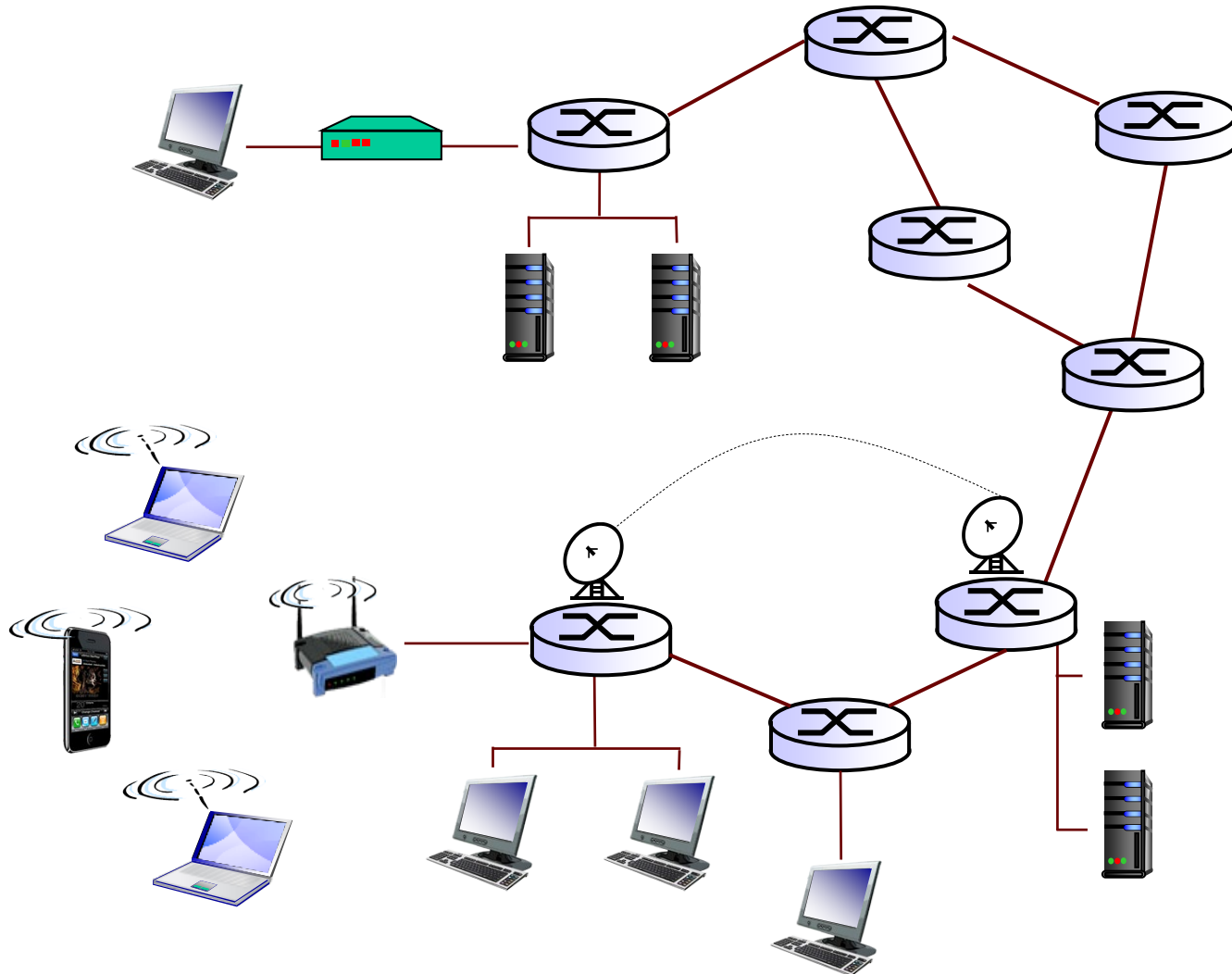
- Single admin, so no policy decisions are needed.
- Routing mostly focus on performance.

## ❖ Inter-AS routing (not covered)

- Admin often wants to control over how its traffic is routed, who routes through its net, etc.
- Policy may dominate over performance.

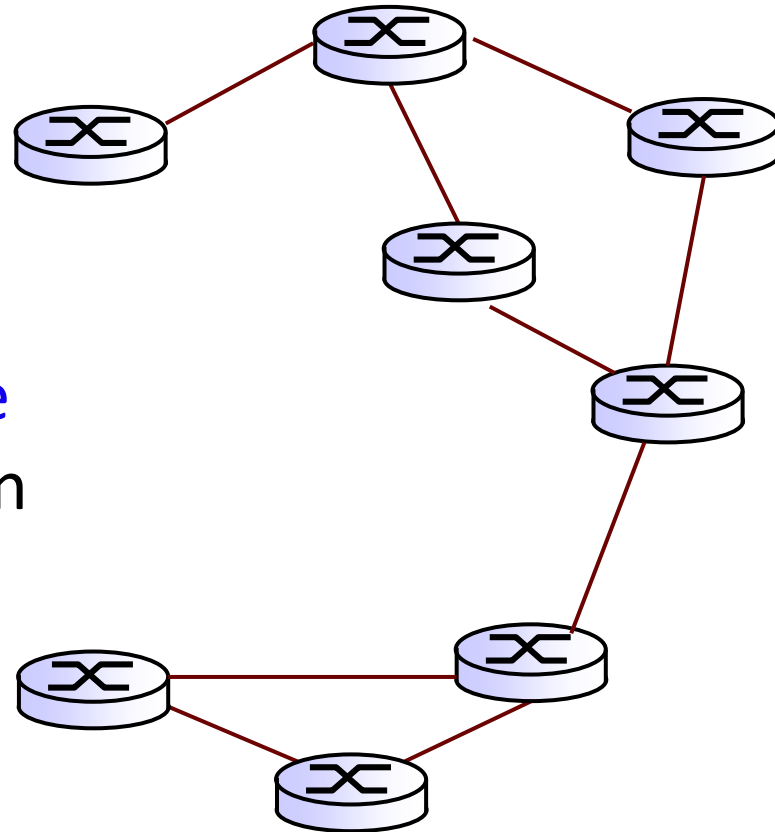


# Abstract View of Intra-AS Routing



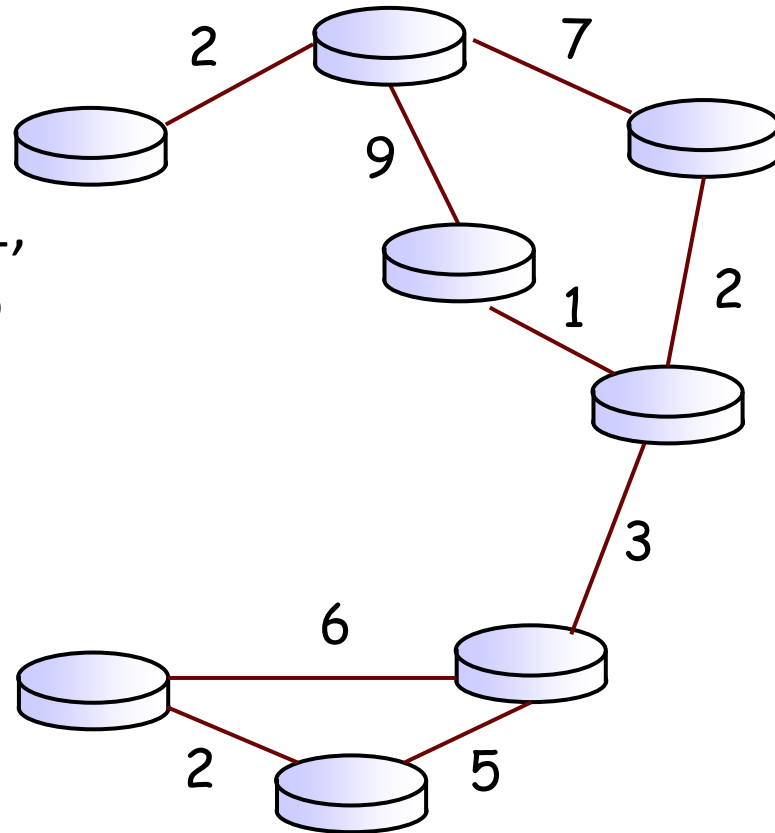
# Abstract View of Intra-AS Routing

- ❖ We can abstractly view a network of routers as a **graph**, where **vertices are routers** and **edges are physical links** between routers.



# Abstract View of Intra-AS Routing

- ❖ We can associate a **cost** to each link.
  - cost could always be 1, or inversely related to bandwidth, or related to congestion.

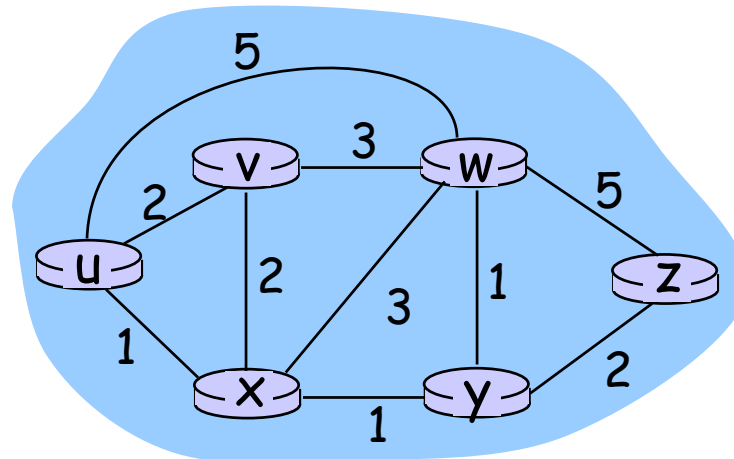


**Routing: finding a least cost path between two vertices in a graph**

# Routing Algorithms Classification

## *“link state” algorithms*

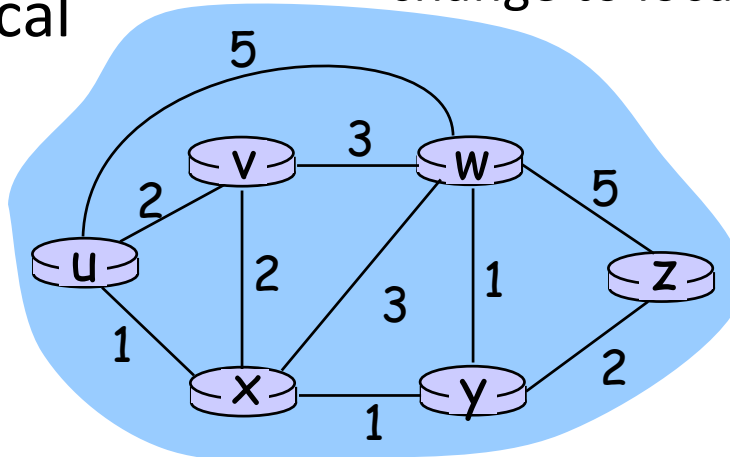
- ❖ All routers have the complete knowledge of network topology and link cost.
  - Routers periodically broadcast link costs to each other.
- ❖ Use Dijkstra algorithm to compute least cost path locally (using global map).
- ❖ Non-examinable 😊



# Routing Algorithms Classification

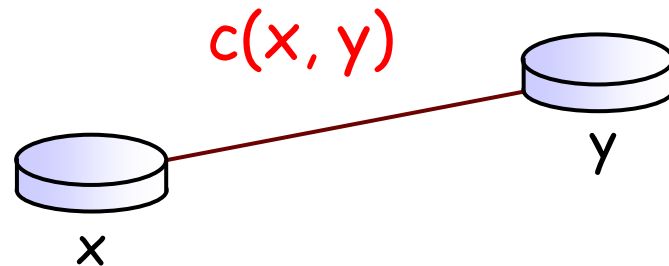
## *“distance vector” algorithms*

- ❖ Routers know physically-connected neighbors and link costs to neighbors.
- ❖ Routers exchange “local views” with neighbors and update own “local views” (based on neighbors’ view).
- ❖ Iterative process of computation
  1. Swap local view with direct neighbours.
  2. Update own’s local view.
  3. Repeat 1 - 2 till no more change to local view.

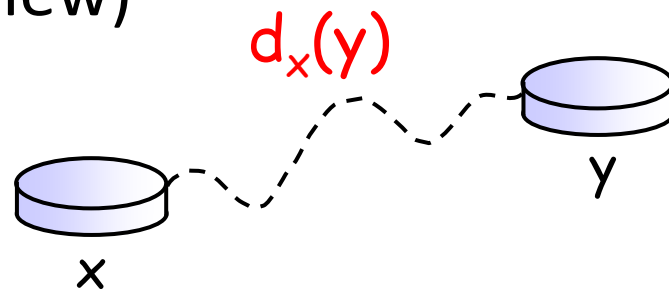


# Some Graph Notations

- ❖  $c(x, y)$ : the cost of link between routers  $x$  and  $y$ 
  - $= \infty$  if  $x$  and  $y$  are not direct neighbours



- ❖  $d_x(y)$ : the cost of the least-cost path from  $x$  to  $y$   
(from  $x$ 's view)

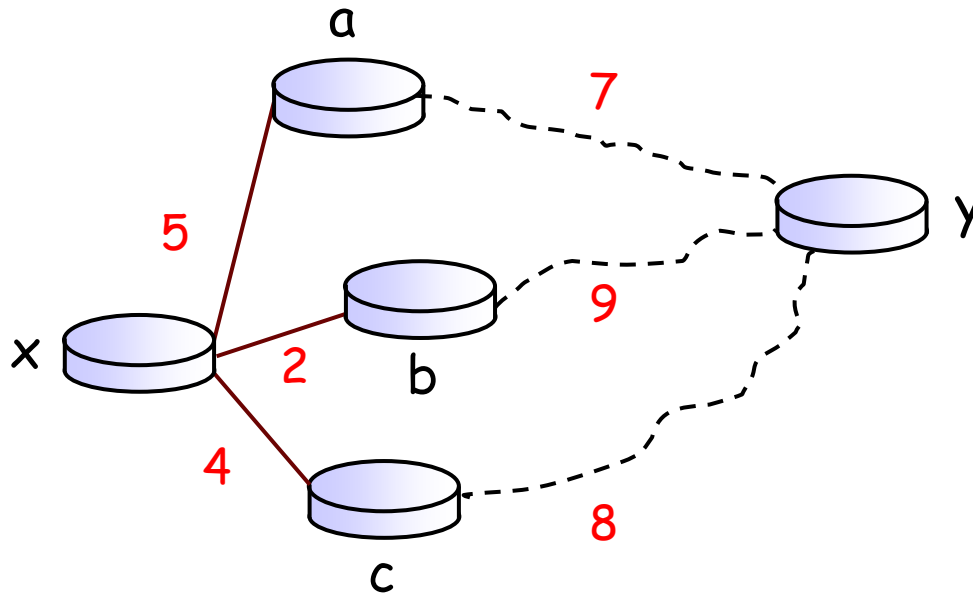




# Bellman-Ford Equation

$$d_x(y) = \min_v \{c(x, v) + d_v(y)\}$$

where min is taken over all direct neighbors  $v$  of  $x$



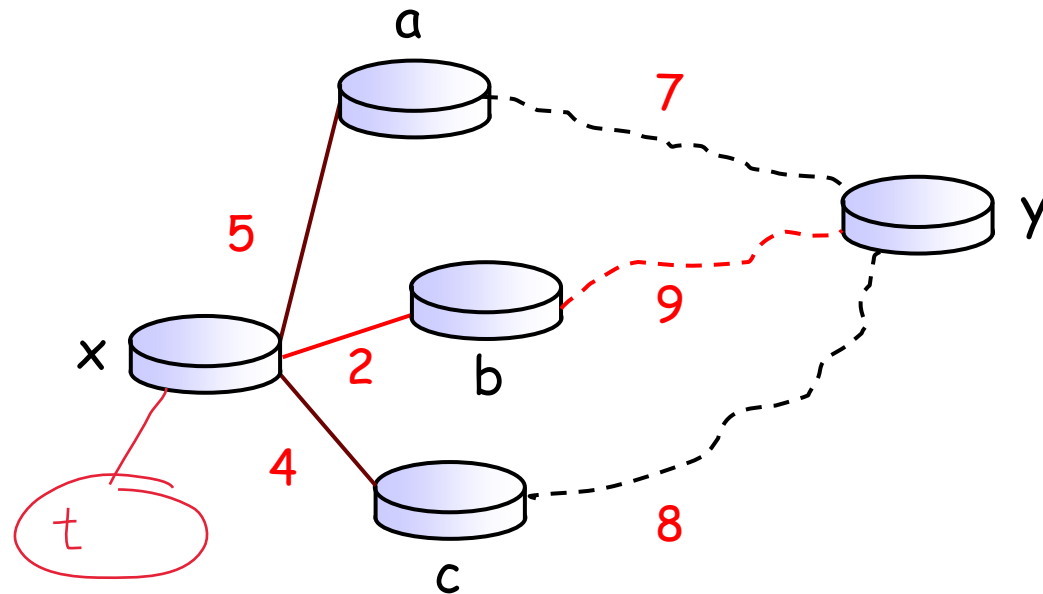
$$d_x(y) = \min_v \{ c(x, a) + d_a(y), \\ c(x, b) + d_b(y), \\ c(x, c) + d_c(y) \}$$

$$= \min \{12, 11, 12\} = 11$$

# Bellman-Ford Equation

dynamic programming through the recursive process

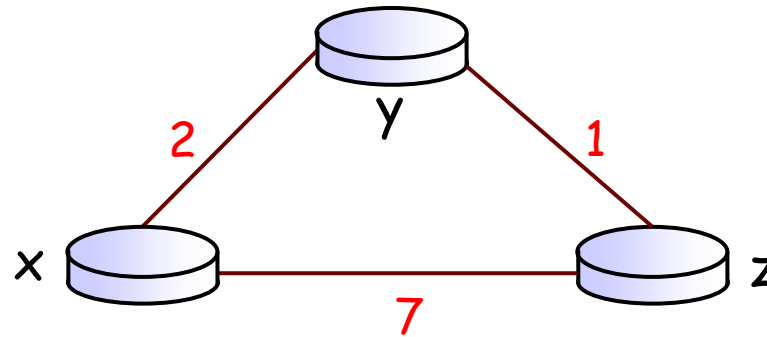
- ❖ To find the least cost path,  $x$  needs to know the cost from each of its direct neighbour to  $y$ .
- ❖ Each neighbour  $v$  sends its **distance vector**  $(y, k)$  to  $x$ , telling  $x$  that the cost from  $v$  to  $y$  is  $k$ .



Now  $x$  knows, to reach  $y$ , packet should be forward to  $b$  and the total cost would be 11.

# Bellman-Ford Example

$$d_x(y) = \min_v \{c(x, v) + d_v(y)\}$$



cost to

	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

x' view

cost to

	x	y	z
x			
y	2	0	1
z			

y' view

cost to

	x	y	z
x			
y			
z	7	1	0

z' view

# Distance Vector Algorithm

- ❖ Every router,  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{z}$ , sends its distance vectors to its directly connected neighbors.
- ❖ When  $\mathbf{x}$  finds out that  $\mathbf{y}$  is advertising a path to  $\mathbf{z}$  that is cheaper than  $\mathbf{x}$  currently knows,
  - $\mathbf{x}$  will update its distance vector to  $\mathbf{z}$  accordingly.
  - In addition,  $\mathbf{x}$  will note down that all packets for  $\mathbf{z}$  should be sent to  $\mathbf{y}$ . This info will be used to create forwarding table of  $\mathbf{x}$ .
- ❖ After every router has exchanged several rounds of updates with its direct neighbors, all routers will know the least-cost paths to all the other routers.

# RIP

- ❖ **RIP (Routing Information Protocol)** implements the DV algorithm. It uses **hop count** as the cost metric (i.e., **insensitive to network congestion**).
- ❖ Exchange routing table every 30 seconds over **UDP** port 520.
- ❖ “Self-repair”: if no update from a neighbour router for 3 minutes, assume neighbour has failed.

# Lectures 6&7: Roadmap

4.1 Overview of Network Layer

4.2 What's Inside a Router

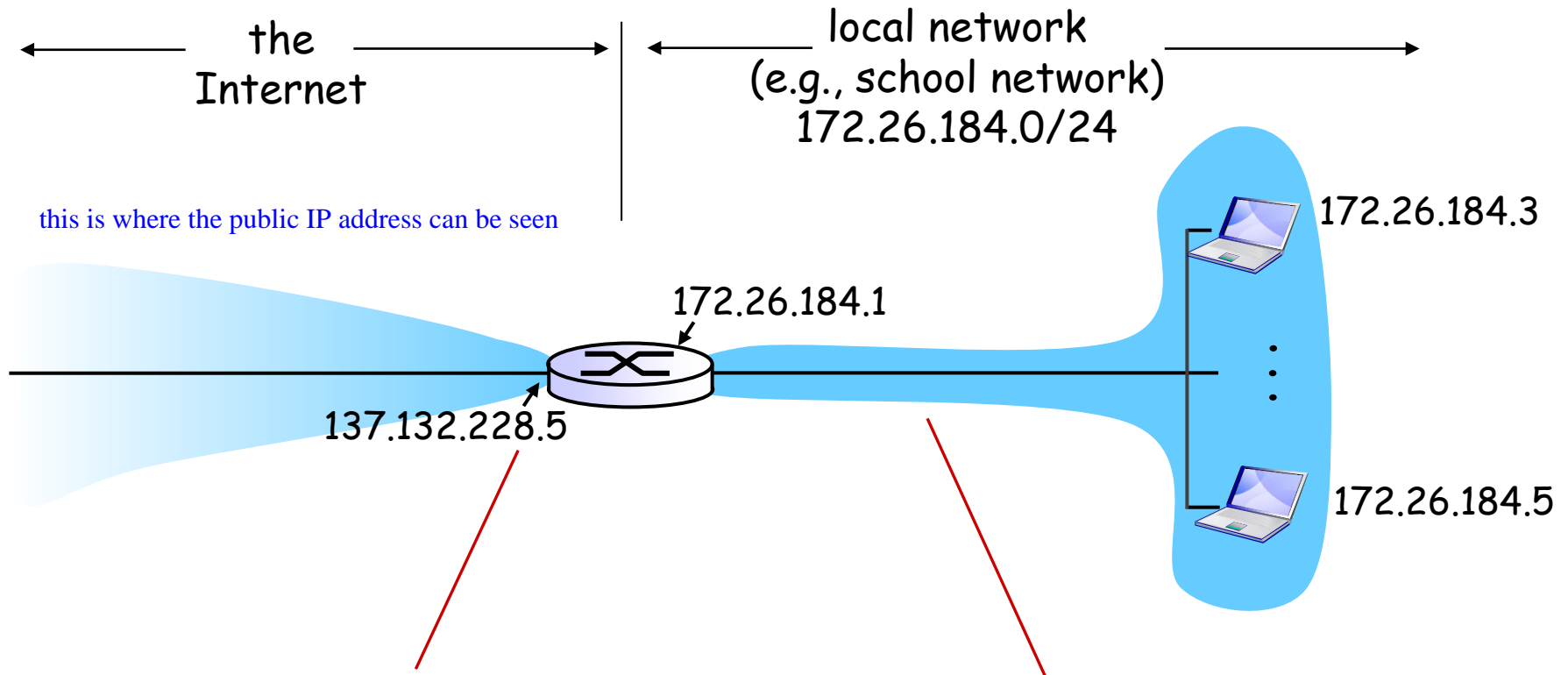
4.3 The Internet Protocol (IP)

- 4.3.4 Network Address Translation

5.2 Routing Algorithms

5.6 ICMP

# NAT: Network Address Translation



*all* datagrams *leaving* local network have the *same* source NAT IP address: 137.132.228.5

Within local network, hosts use private IP addresses 172.26.184.\* for communication

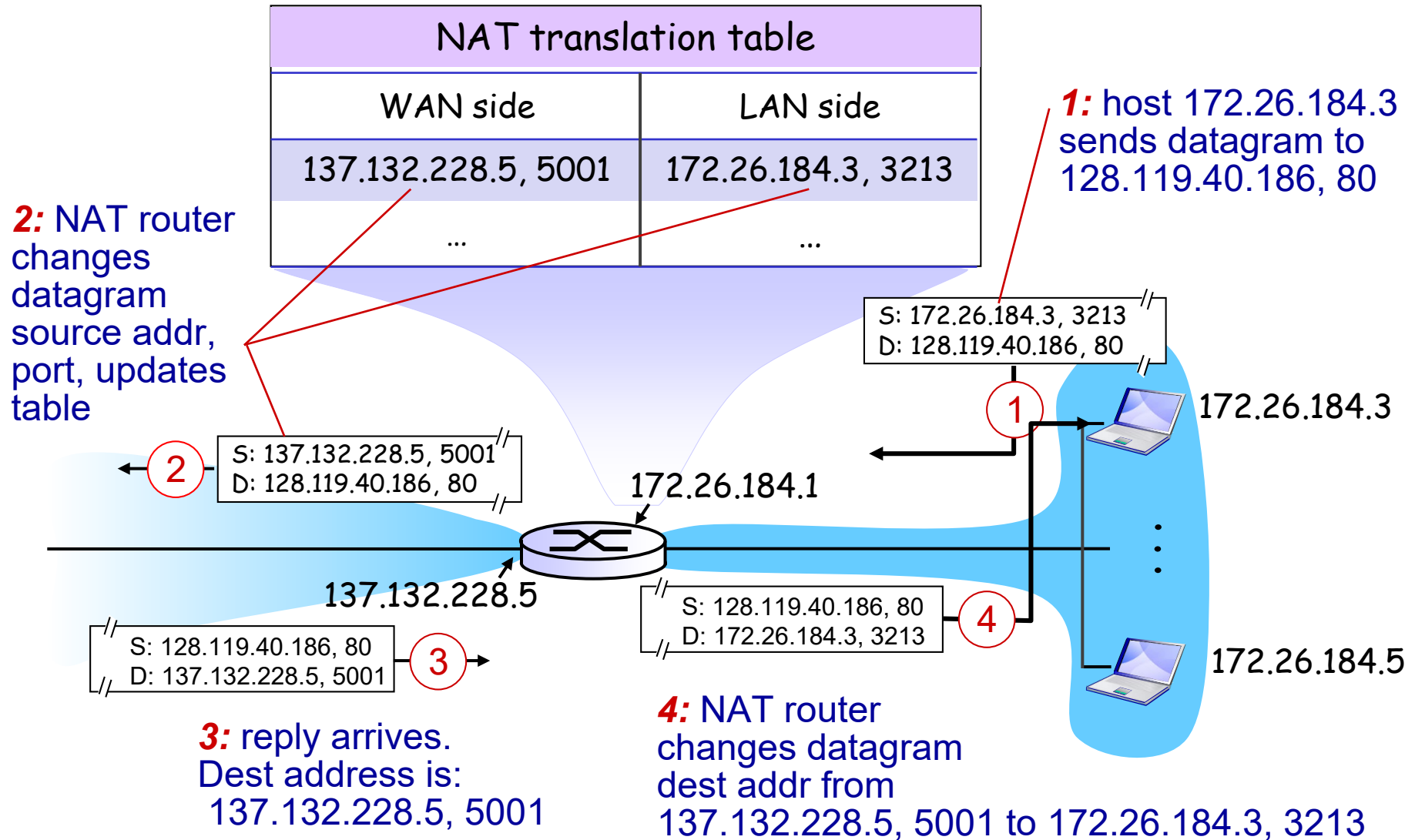
# NAT: Implementation

## ❖ NAT routers must:

- **Replace** (source IP address, port #) of every **outgoing datagram** to (NAT IP address, **new port #**).
- **Remember** (in NAT translation table) the mapping from (source IP address, port #) to (NAT IP address, new port #).
- **Replace** (NAT IP address, new port #) in destination fields of every **incoming datagram** with corresponding (source IP address, port #) stored in NAT translation table.



# NAT: Illustration



# NAT: Motivation and Benefits

- ❖ No need to rent a range of public IP addresses from ISP: just one public IP for the NAT router.
- ❖ All hosts use private IP addresses. Can change addresses of hosts in local network without notifying the outside world.
- ❖ Can change ISP without changing addresses of hosts in local network.
- ❖ Hosts inside local network are not explicitly addressable and visible by outside world (a security plus).

# Lectures 6&7: Roadmap

4.1 Overview of Network Layer

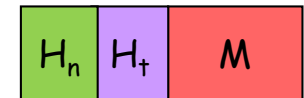
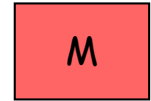
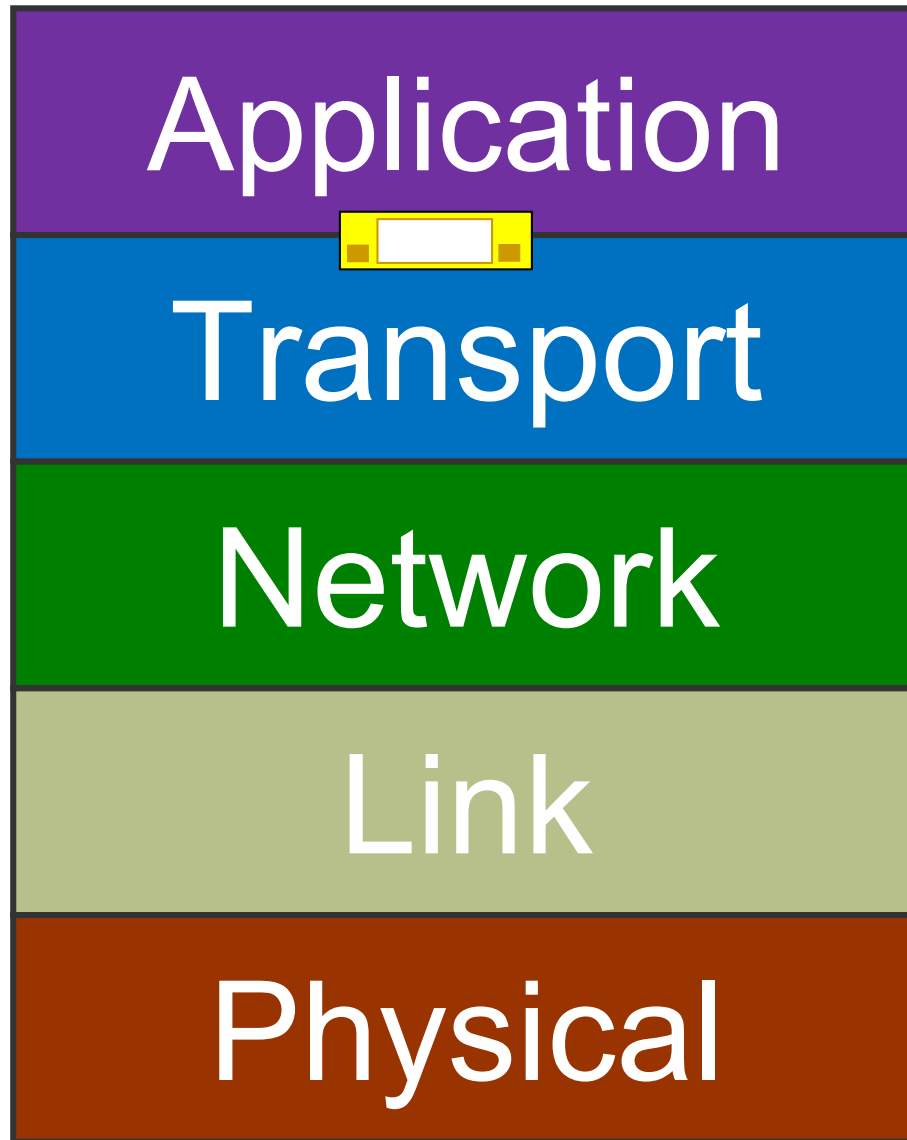
4.2 What's Inside a Router

4.3 The Internet Protocol (IP)

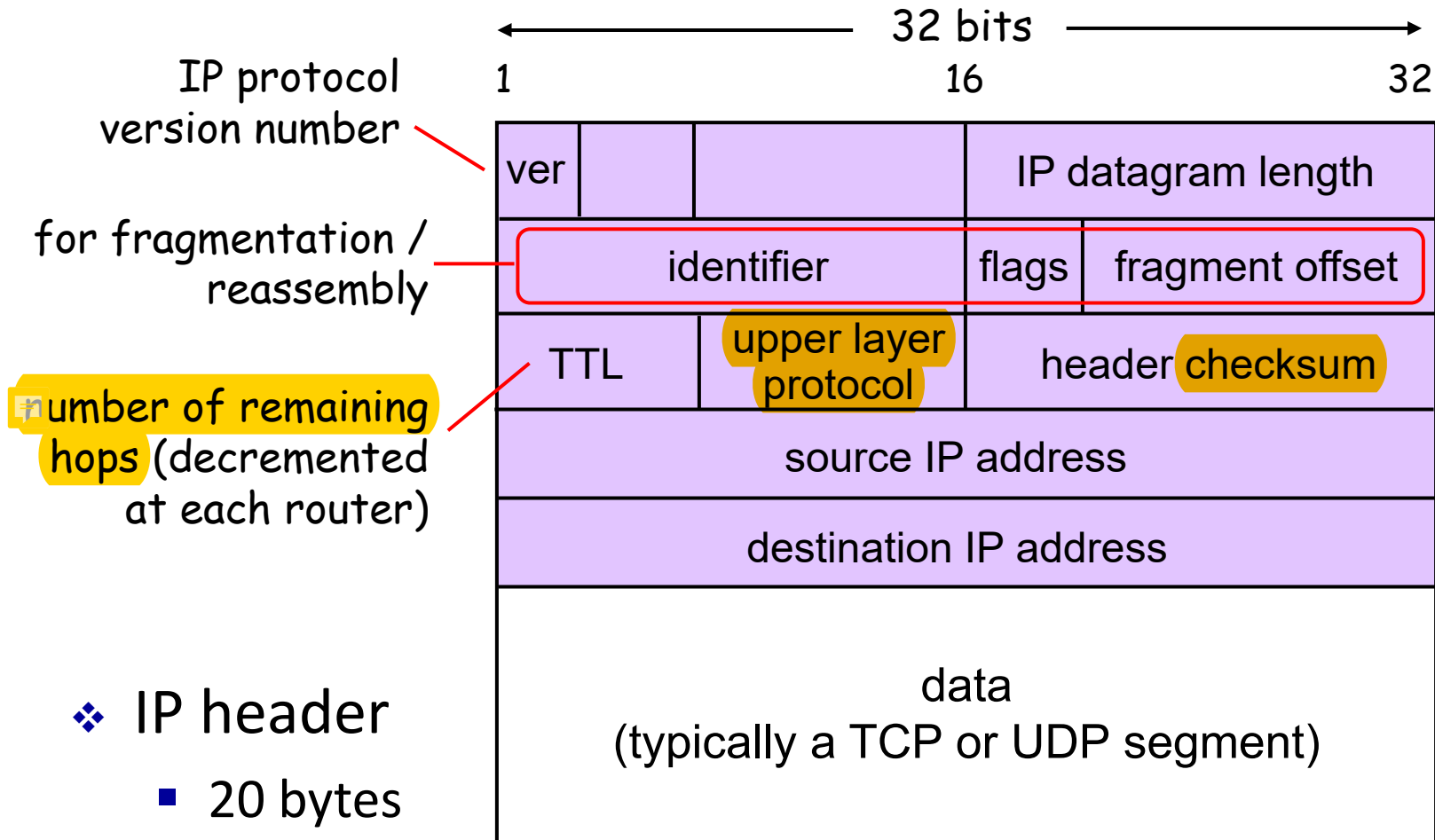
- 4.3.1 IPv4 Datagram Format
- 4.3.2 IPv4 Datagram Fragmentation
- 4.3.5 IPv6 (non-examinable)

5.2 Routing Algorithms

5.6 ICMP



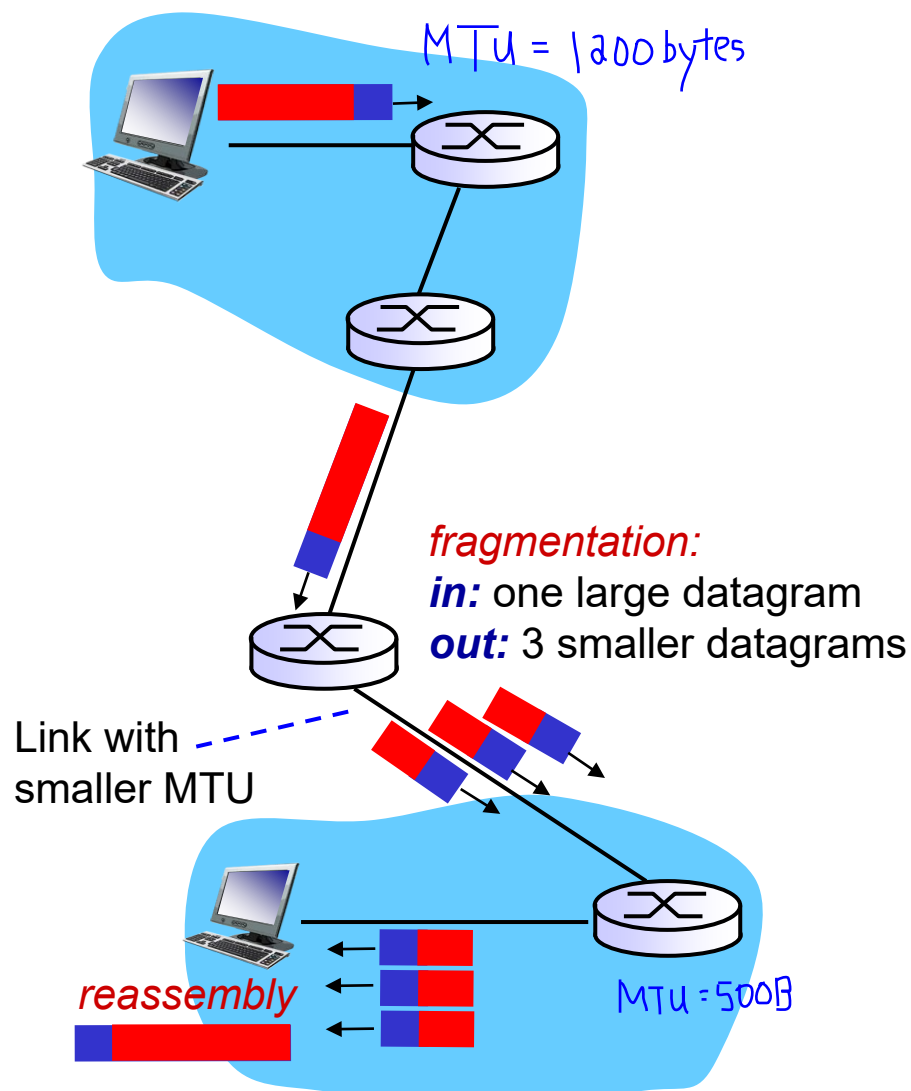
# IPv4 Datagram Format



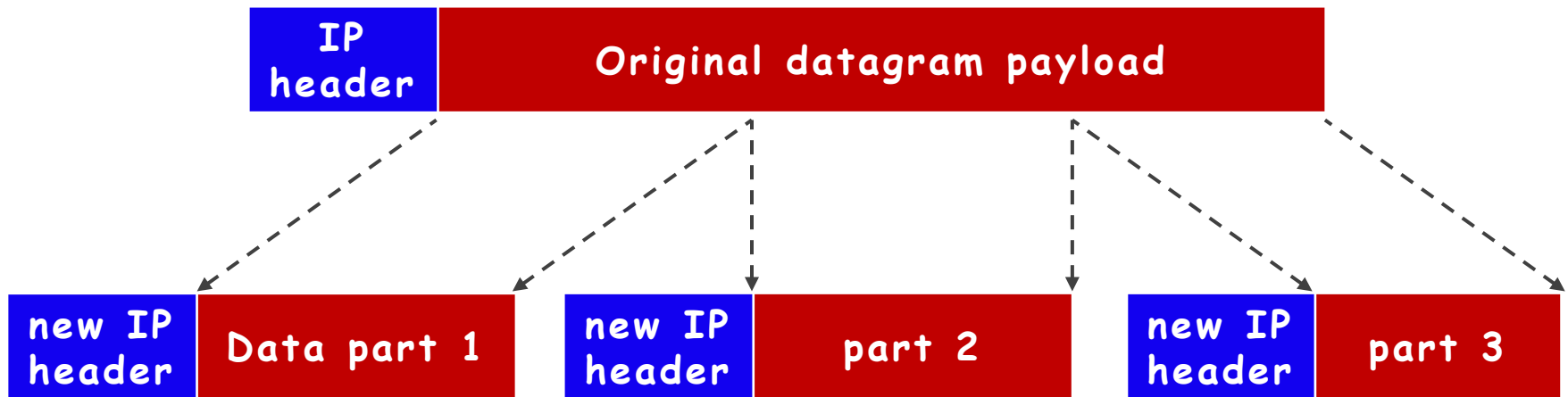
(some fields are not shown)

# IP Fragmentation & Reassembly

- ❖ Different links may have different **MTU (Max Transfer Unit)** – the maximum amount of data a link-level frame can carry.
- ❖ “Too large” IP datagrams may be fragmented by routers.



# IP Fragmentation Illustration



- ❖ Destination host will reassemble the packet.
- ❖ IP header fields are used to identify fragments and their relative order.

# IP Fragmentation

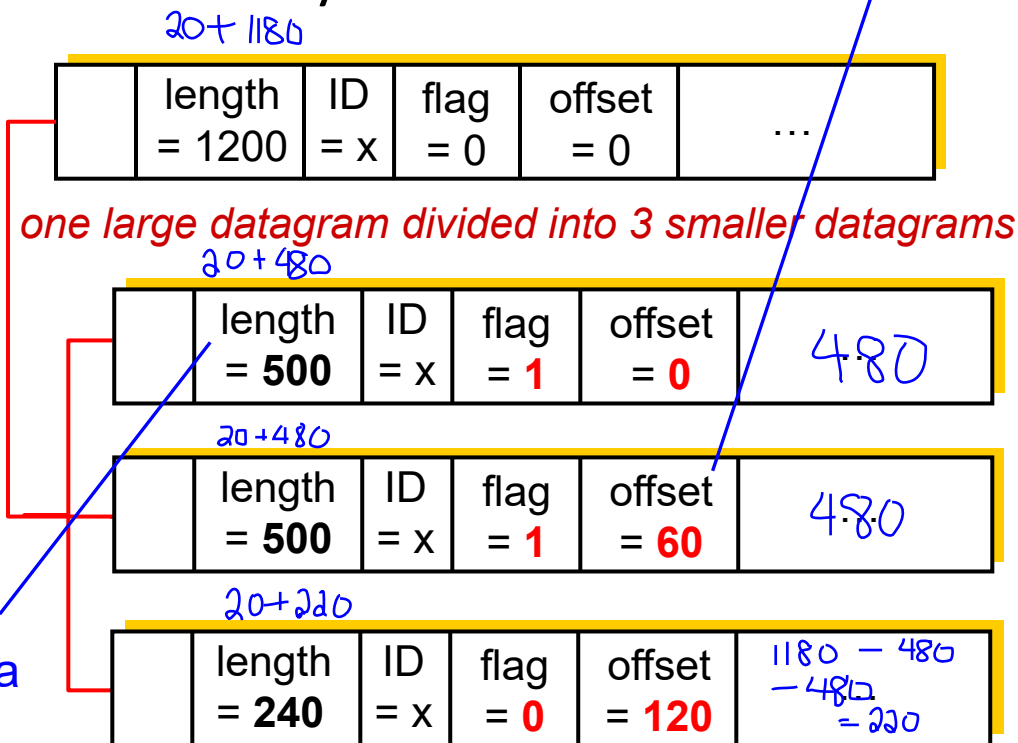
			length
identifier	flags	offset	
source IP address			
destination IP address			

- ❖ Flag (frag flag) is set to
  - **1** if there is next fragment from the same segment.
  - **0** if this is the last fragment.
- ❖ Offset is expressed in unit of 8-bytes.

## ❖ Example

- 20 bytes of IP header
- 1,200 byte IP datagram
- MTU = 500 bytes

carry 480  
bytes of data

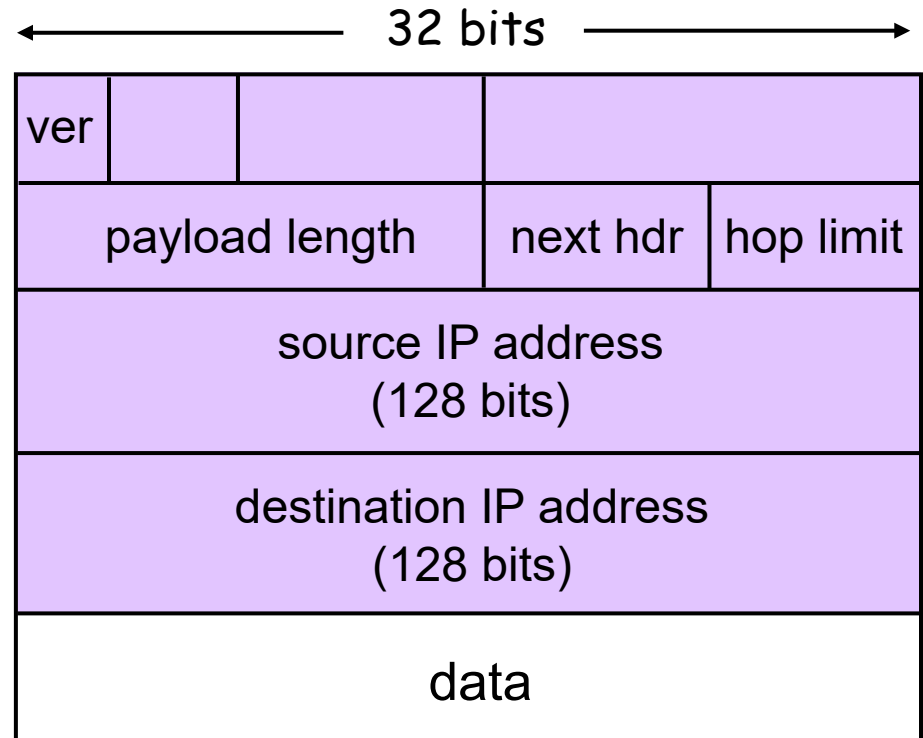




# IPv6

**Non-  
examinable**

- ❖ IPv6 is designed to replace IPv4.
- ❖ Primary motivation: 32-bit IPv4 address space is soon to be completely allocated.
- ❖ IPv6 datagram:
  - 40 byte header



(some fields are not shown)

Example IPv6 address (in hexadecimal):  
2001:0db8:85a3:0042:1000:8a2e:0370:7334

# Lectures 6&7: Roadmap

4.1 Overview of Network Layer

4.2 What's Inside a Router

4.3 The Internet Protocol (IP)

5.2 Routing Algorithms

5.6 ICMP

# ICMP

- ❖ **ICMP (Internet Control Message Protocol)** is used by hosts & routers to communicate network-level information.
  - Error reporting: unreachable host / network / port / protocol
  - Echo request/reply (used by ping)
- ❖ ICMP messages are carried in IP datagrams.
  - ICMP header starts after IP header.

# ICMP Type and Code

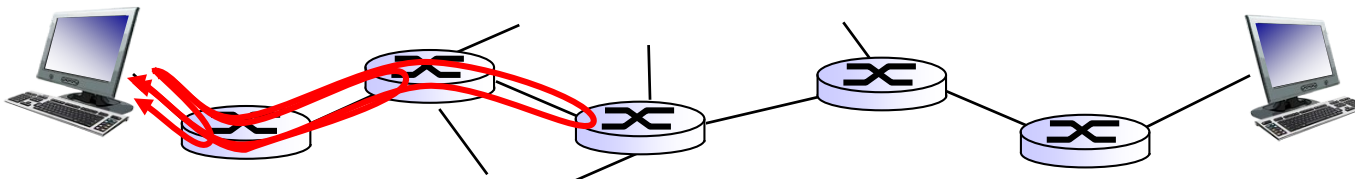
❖ ICMP header: Type + Code + Checksum + others.

Type	Code	Description
8	0	echo request (ping)
0	0	echo reply (ping)
3	1	dest host unreachable
3	3	dest port unreachable
11	0	TTL expired
12	0	bad IP header

Selected ICMP Type and subtype (Code)

# Examples: *ping* and *traceroute*

- ❖ The command **ping** sees if a remote host will respond to us – do we have a connection?
- ❖ The command **traceroute** sends a series of small packets across a network, and attempts to display the route (or path) that the messages would take to get to a remote host.



# Lectures 6&7: Summary

- ❖ An IP address is associated with a network interface. A device may have multiple network interfaces, thus multiple IP addresses.
- ❖ DHCP automates the assignment of IP addresses in an organization's network.
- ❖ On TCP/IP networks, subnets are defined as all devices whose IP addresses have the same network (subnet) prefix.
- ❖ Subnet mask is useful in checking if two hosts are on the same subnet.

# Lectures 6&7: Summary

- ❖ Routing is the process of selecting best paths in a network.
- ❖ **NAT** maps one IP addresses space into another.
  - Commonly used to hide an entire private IP address space behind a single public IP address.
  - NAT router uses stateful translation tables to remember the mapping.
- ❖ **ICMP** is used by routers to send error messages.
  - E.g. when TTL is 0, a packet is discarded and an ICMP error message is sent to the datagram's source address.