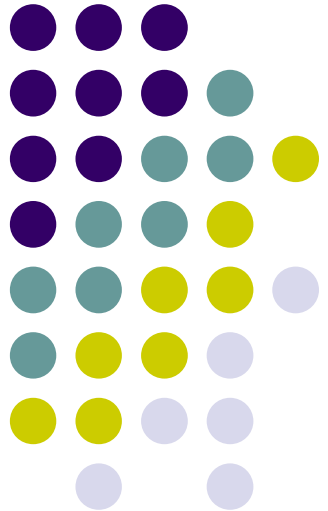
# Lecture 6: Internet Layer

Reading 5.1. and 5.6 in Computer Networks, Tanenbaum



# Contents

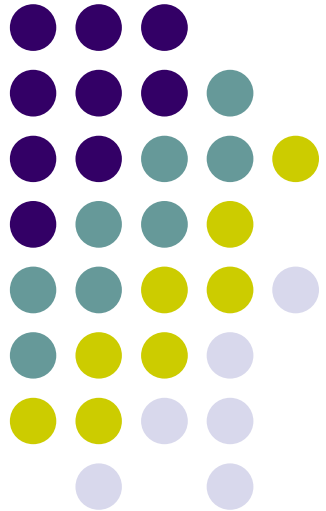
- Internet Protocol
- IP address and IP packet format
- ICMP- Protocol for control message

# Introduction about IP

Concepts

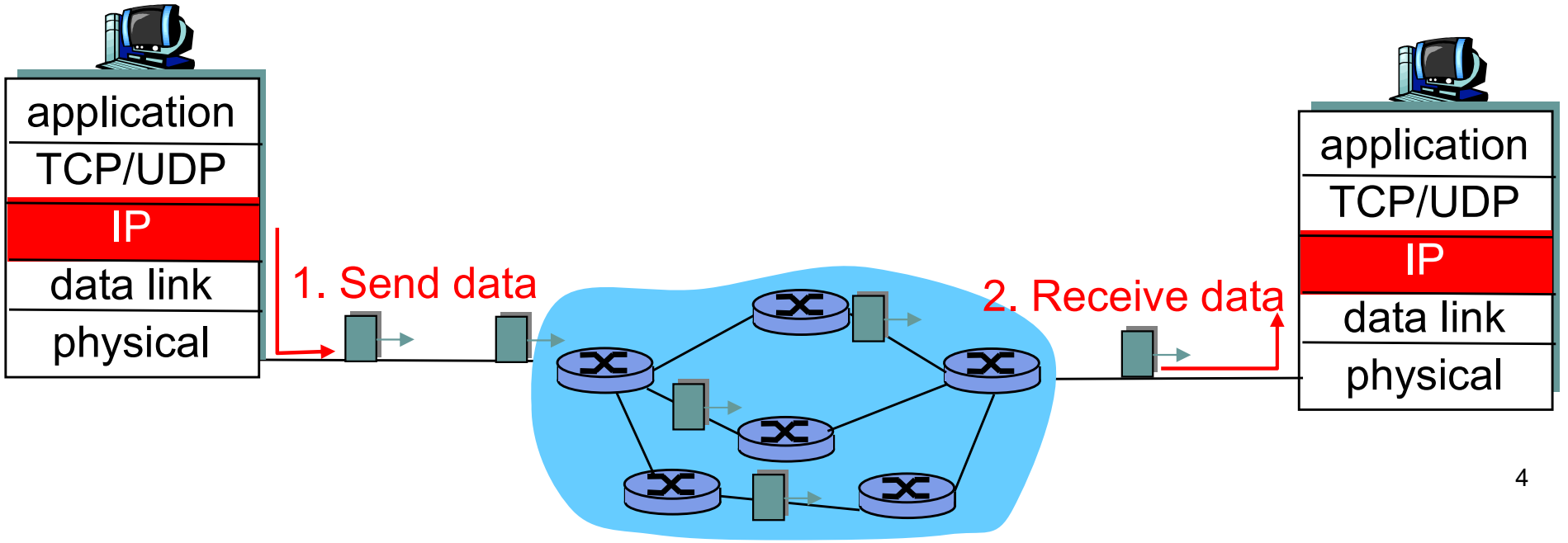
Store and forward principles

Characteristic of IP



# Network layer and Internet protocol

- Role of network layer: Transferring data between distant nodes
- Two main functionalities of Network layer
  - *Routing*: Determine the path for transferring data from the source to the destination nodes → Role of routing protocol.
  - *Forwarding*: Transferring data from the an incoming port to an outgoing port of a node (router) according to the path defined above ➔ Role of routed protocol: Internet Protocol (IP)



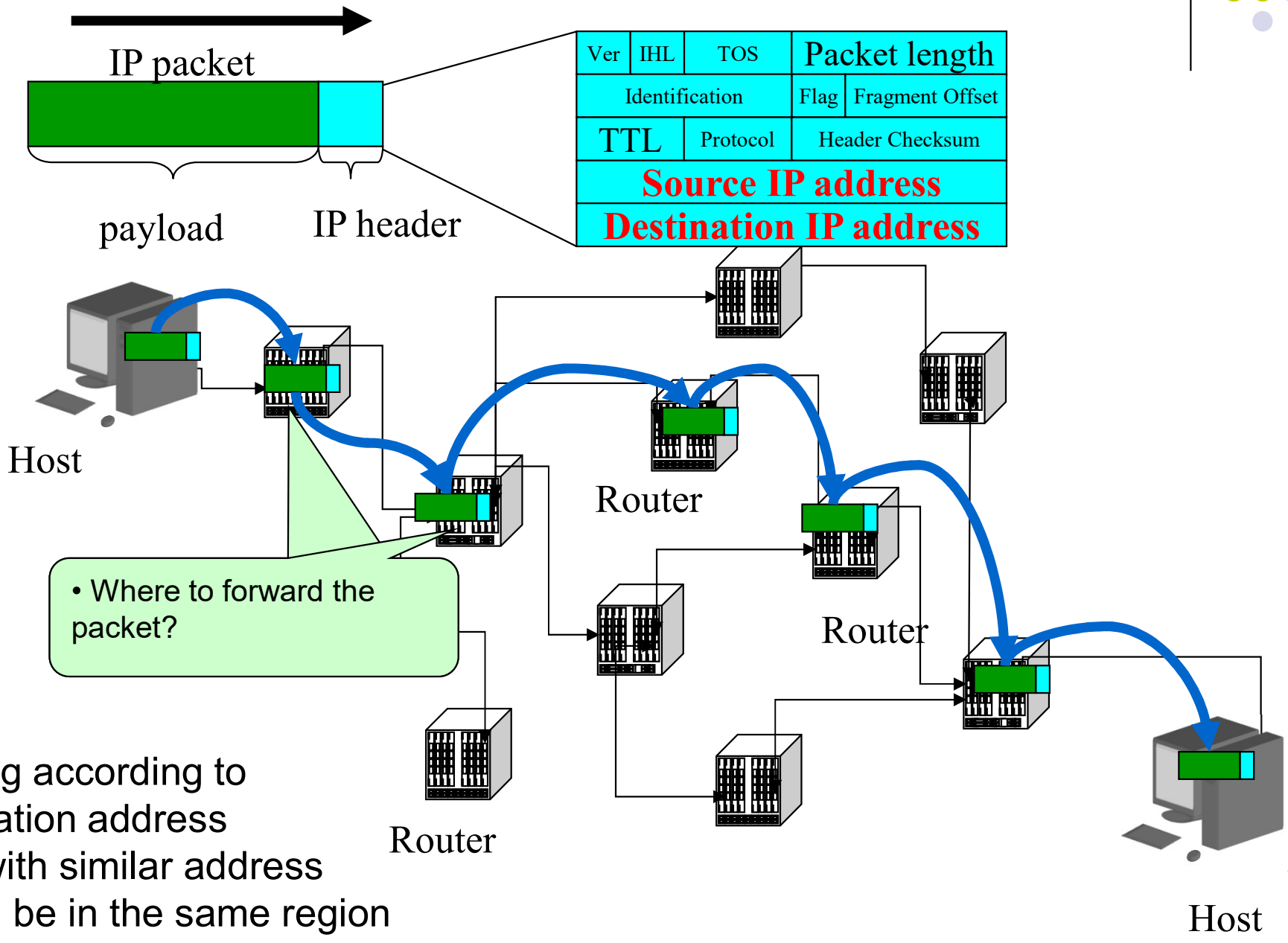
# Network layer and Internet protocol

- Layer 2 devices allow to connect limited number of close hosts
- When hosts are far from each other, intermediates nodes with forwarding and path finding functionality is needed → Router
  - Finding routes
  - Forwarding data according to destination Network layer address

# IP principles

- Elements
  - **host** = end system;
  - **subnetwork** = a collection of hosts that are connected by layer-2 devices
    - Host of the same subnetwork have similar addresses: a common prefix
  - Routers: intermediate nodes interconnect subnetworks:
- Packet forwarding
  - **direct**: inside a subnetwork hosts communicate directly without routers, layer-2 device (switch) delivers packets to hosts
  - **indirect**: between subnetworks one or several routers forward packets based on
    - structured address space
    - routing tables: aggregation of entries

# IP Routing and forwarding




- Routing according to destination address
- Host with similar address should be in the same region

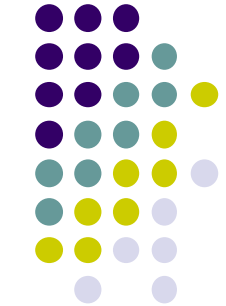
# IP forwarding

- Routing table
  - Each router have a routing table telling where to forward a packet

| Network | Next-hop |
|---|---|
| 10.0.0.0/24 | A |
| 172.16.0.0/24 | C |
| 192.168.0.0/24 | Direct |

- Rule for sending packets (hosts, routers)
  - § if the destination IP address has the same prefix as one of my interfaces, send directly to that interface
  - § otherwise send to a router as given by the IP routing table

# IP characteristics



- Not reliable / fast
  - Sending data in "*best effort*" manner
  - No mechanism to recover error data at the receiver
  - When necessary, leave the upper layer (TCP) to ensure the data reliability.
- Packets are processed independently one of the other.

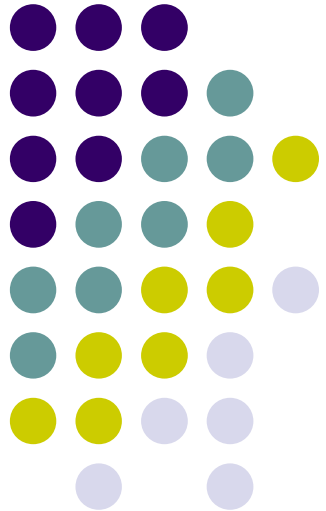
# IP address

IP address classes

CIDR – Classless Inter-Domain routing

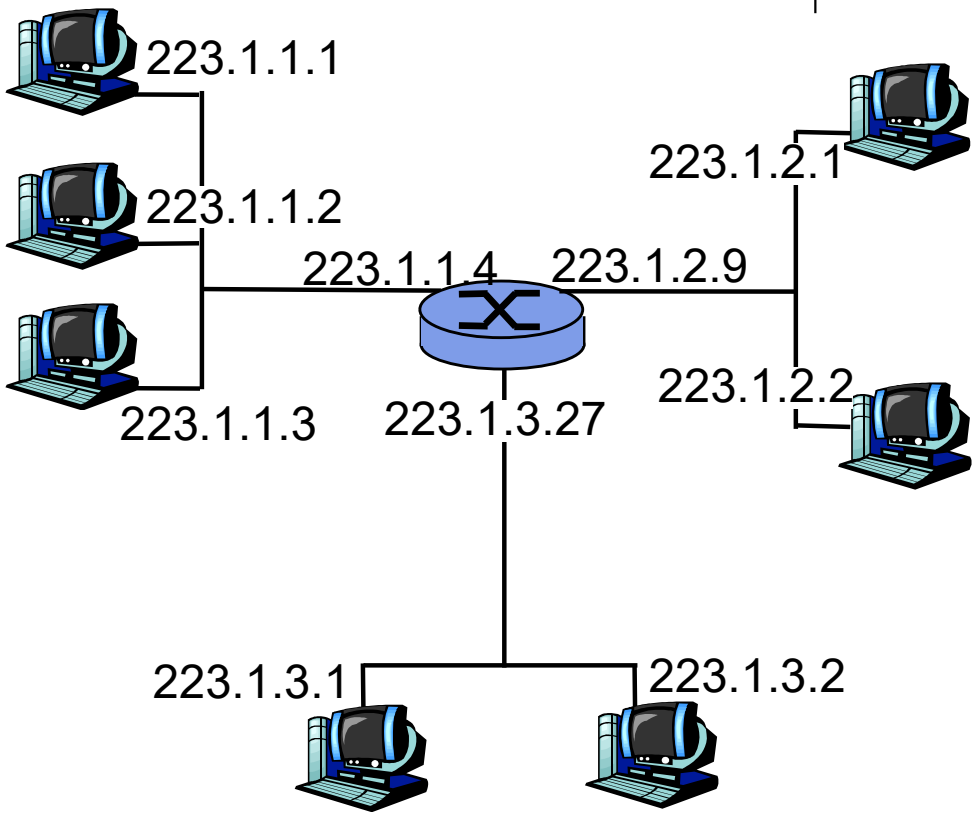
Subnet and netmask

Special IP addresses



# IP address (IPv4)

- IP address: A 32-bit number identifying uniquely a network interface
- Interface:
  - router's typically have multiple interfaces
  - host may have multiple interfaces
  - IP addresses associated with interface, not host, router

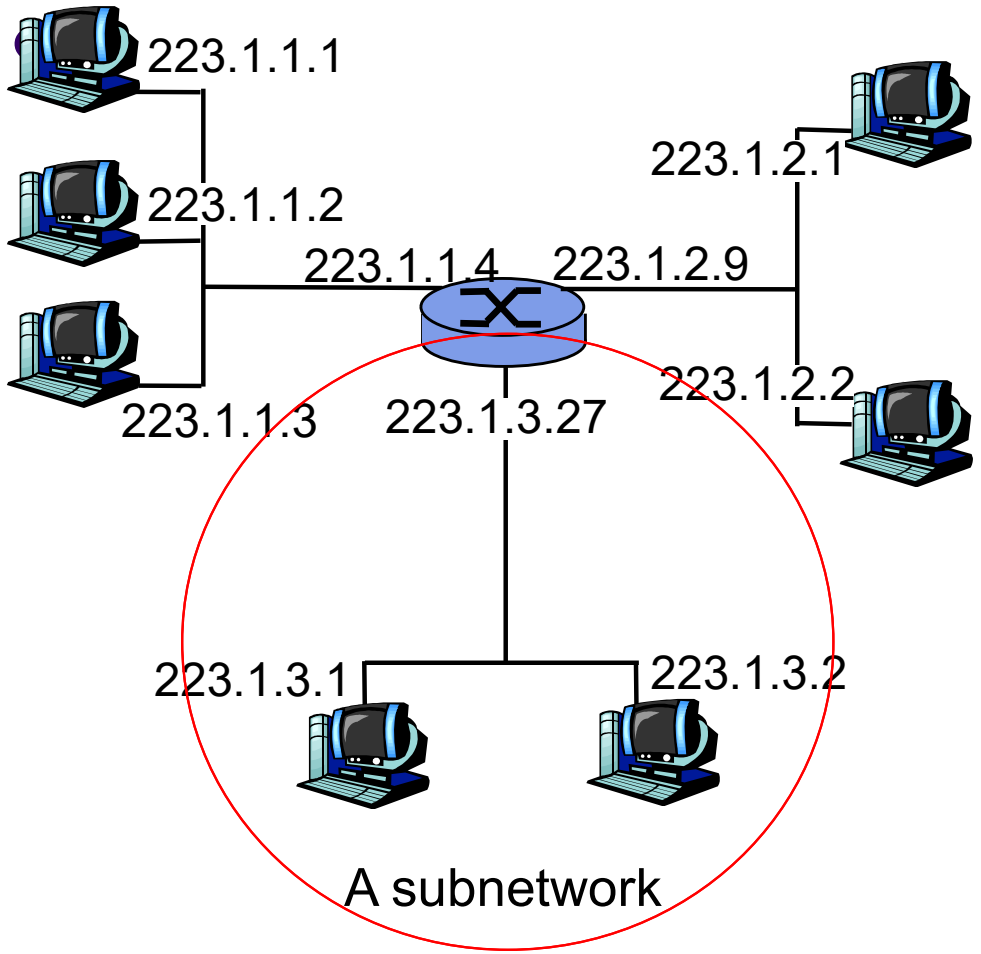



223.1.1.1 = 11011111 00000001 00000001 00000001

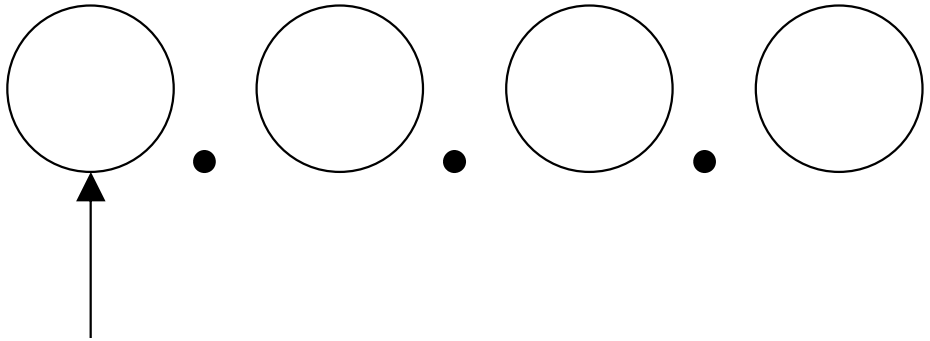
223 1 1 1

# IP address (IPv4)

- For routing purpose, IP address of interfaces in the same subnetwork have the same prefix.
- What's a subnetwork?
  (from IP address perspective)
  - device interfaces with same prefix
  - can physically reach each other without intervening router (using layer 2 technology only)



# Dot notation



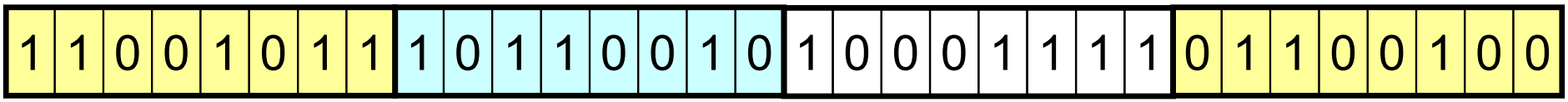
8 bits
0 – 255 integer

Example:

| | |
|---|---|
| 203.178.136.63 | o |
| 259.12.49.192 | x |
| 133.27.4.27 | o |

Use 4 x 8 bits describing a 32 bits address

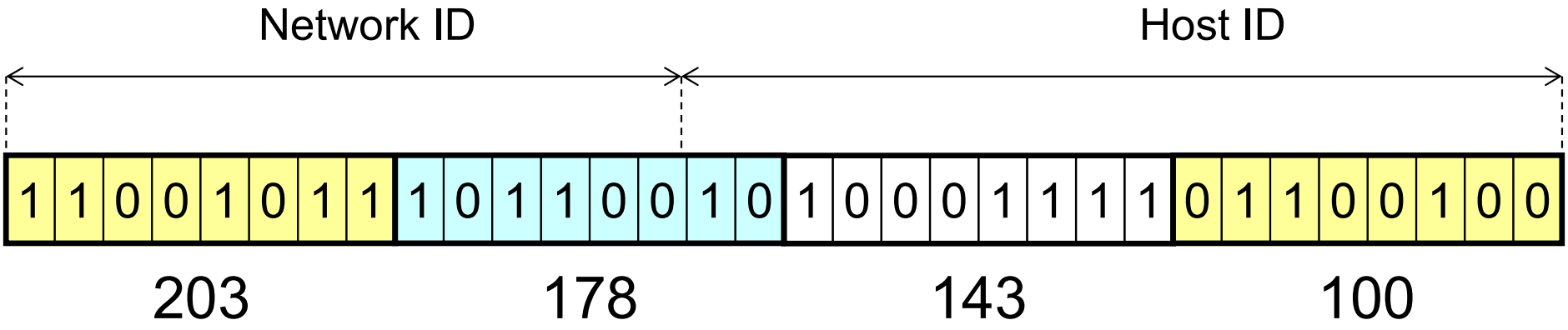
3417476964




203 178 143 100

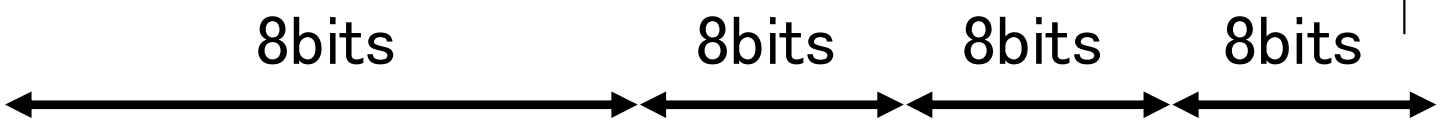
# Host address, network address

- IP address contains two parts
  - Host ID – identify a host in a network
  - Network ID – identify a network




- How to know which bits belong to network ID or host ID parts?
  - Use classful IP address
  - Use classless IP address– CIDR

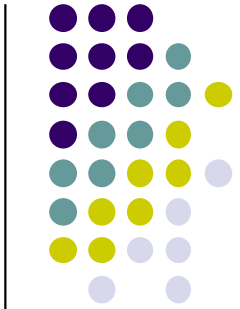
# Classify IP addresses




| | | | | | |
|---|---|---|---|---|---|
| Class A | 0 | 7bit | H | H | H |
| Class B | 1 0 | 6bit | N | H | H |
| Class C | 1 1 0 | 5bit | N | N | H |
| Class D | 1 1 1 0 | Multicast | | | |
| Class E | 1 1 1 1 | Reserve for future use | | | |

| | # of network | # of hosts |
|---|---|---|
| Class A | 128 | 2^24 |
| Class B | 16384 | 65536 |
| Class C | 2^21 | 256 |

# Exercise



- Determine which classes do these IP addresses belong to:
  - 10.10.10.9
  - 192.168.70.5
  - 129.60.4.7

# Limitation of classful IP address

- Inefficient use of addressing space
  - Hard classification of addressing space into classes (A, B, C, D, E) makes it is difficult to use all the address space

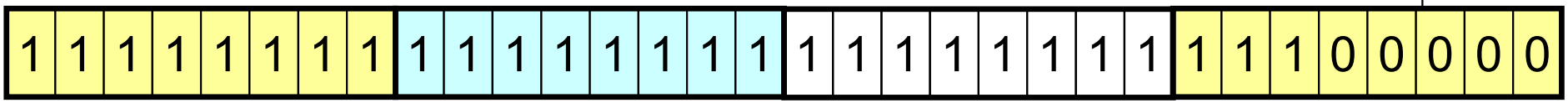
## Solution…

- CIDR: Classless Inter Domain Routing
  - Network ID part will have variable length.
  - Length of Network ID part is specified in Network mask
  - Address notation: a.b.c.d/x, where x (mask) the number of bit of Network ID part.

# Network mask

- Network mask divides the IP address into two parts
  - Part corresponding to Host ID
  - Part corresponding to Network ID
- IP addresses are assigned to hosts so that all hosts in the same network have the same Network ID part.
- Based on Network mask, it is possible to
  - Identify the network where an IP address belongs to
  - Calculate how many IP addresses available in the network associated with the mask.

# Presentation of network mask

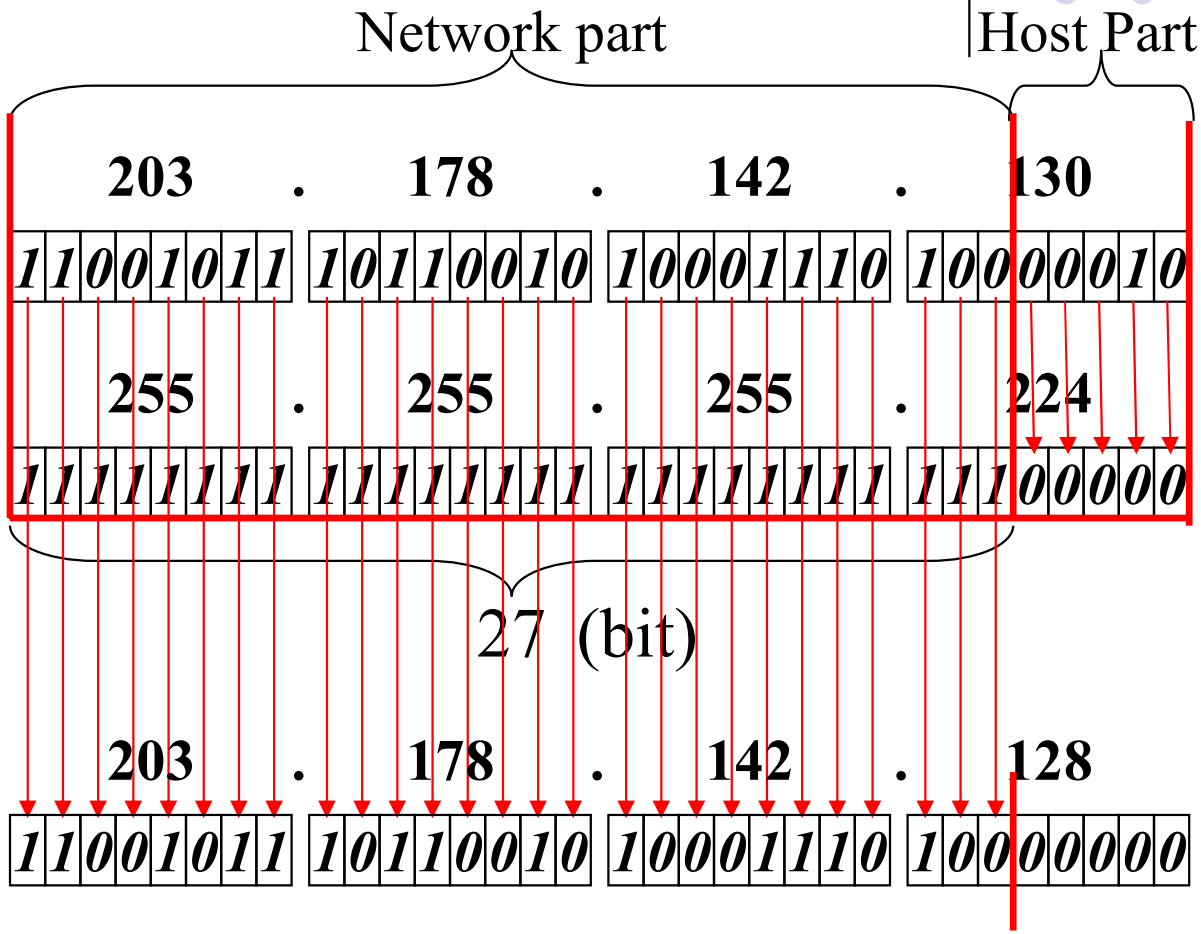



- 255.255.255.224
- /27
- 0xFFFFFFe0

- Last byte may be:

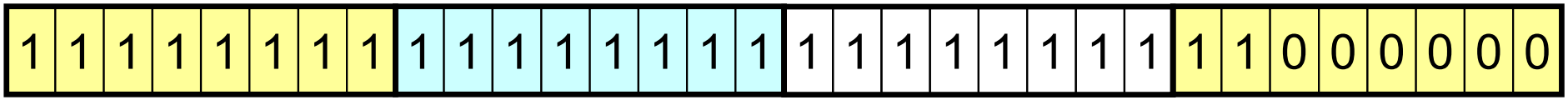
0 248
128 252
192 254
224 255
240

# Calculation of network address




**Network address** **203.178.142.128/27**

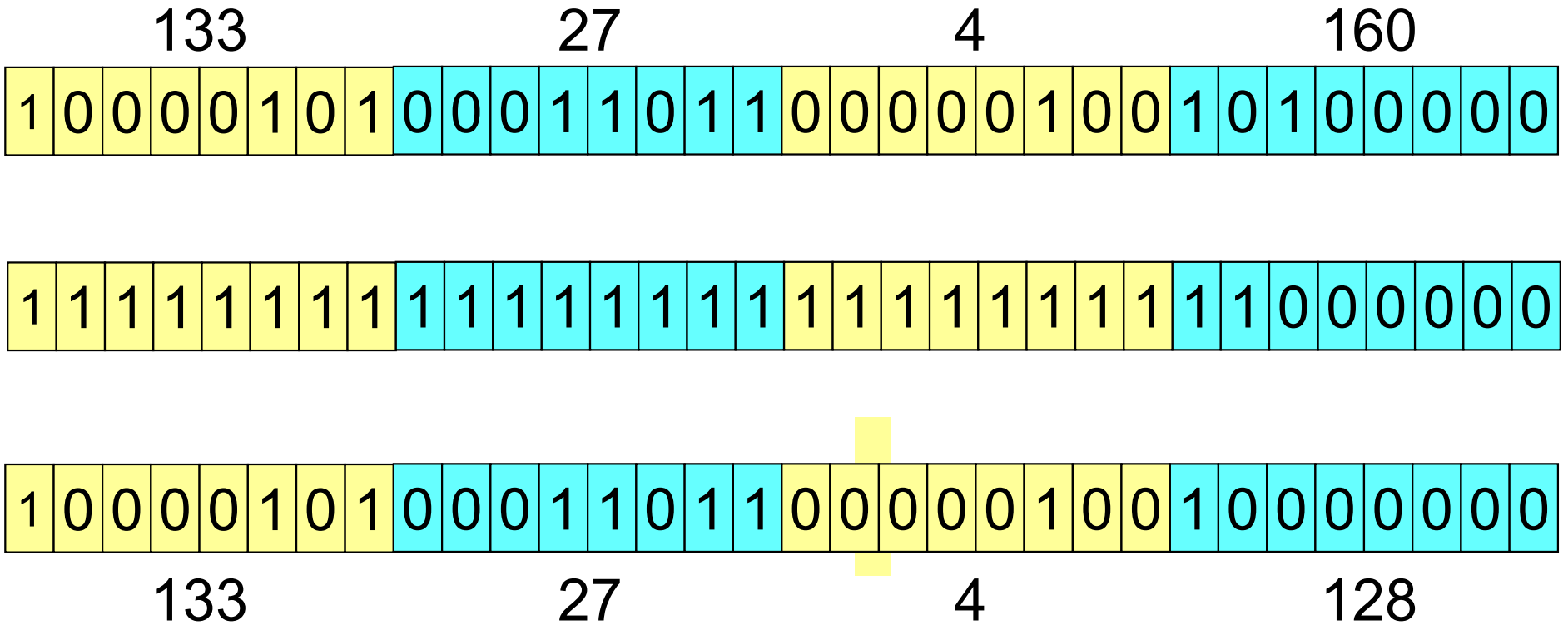
# Calculation of network size



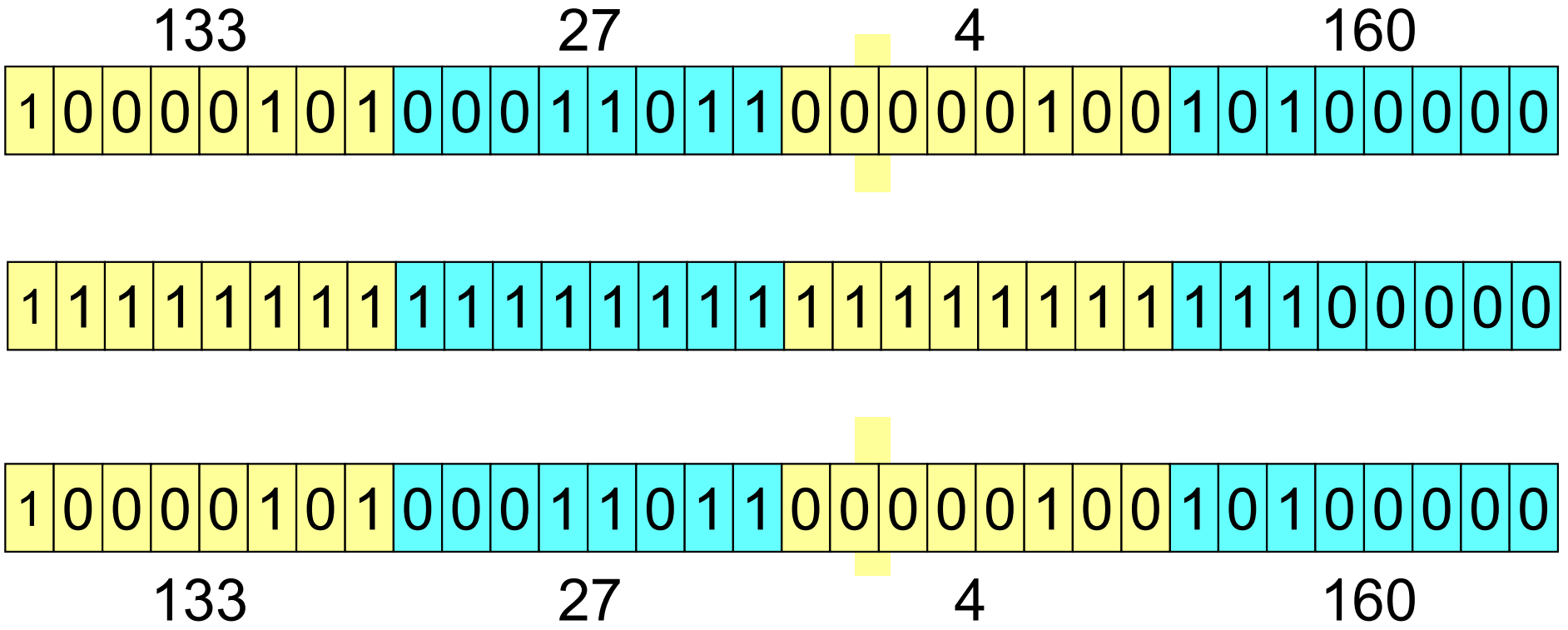

- Network size
  - Power of 2
- RFC1878

- In case of mask /26
  - Bits for Host ID = 6 bits
  - $2^6$=64 possible address:
    - 0 - 63
    - 64 - 127
    - 128 - 191
    - 192 – 255
  - Including network address and broadcast address

# Network address or host address (1)



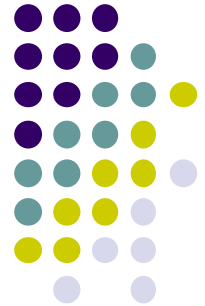
# Network address or host address (2)



# Different significations of IP address

- Network address
  - IP address assigned to a network
  - hostID contains all 0
- Host address
  - IP address assigned to a network card
- Broadcast address
  - Address used for sending data to all hosts in a network
  - All bit 1 in HostID part.

# Exercice: IP address and network mask



- Which of the following IP addresses are host address, network address, broadcast address?

(1) 203.178.142.128 /25

(2) 203.178.142.128 /24

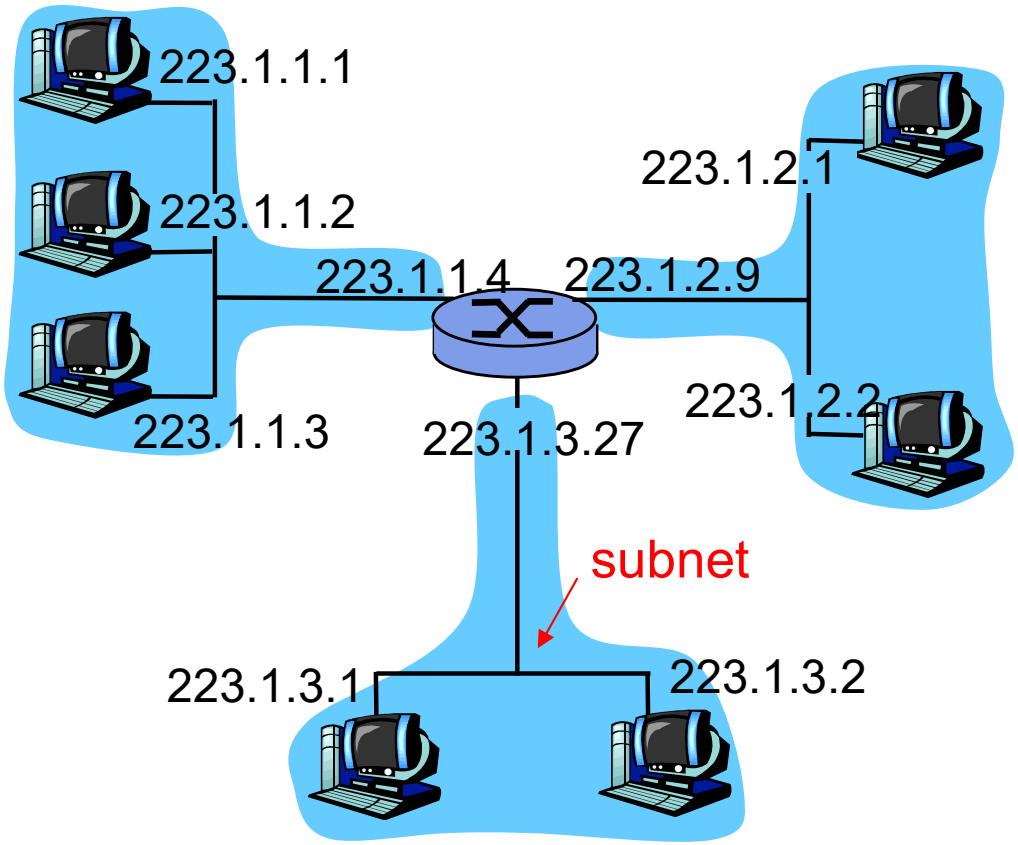
(3) 203.178.142.127 /25

(4) 203.178.142.127 /24

- Attn: With CIDR addressing, IP address should always coming with a network mask

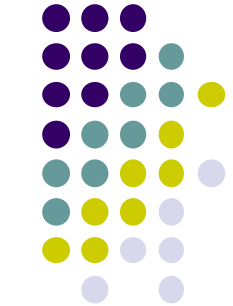
# Subnet

- Subnet is a part of a network
  - Hosts of a subnet communicate directly without reaching to layer 3.
  - Usually is one department of an organization
- Design question: How to assign addresses of a network to subnets
  - Use a longer netmask

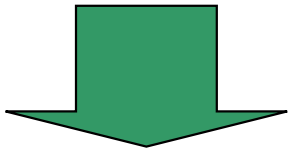



A network with 3 subnets.

# Example: Divide into 2 subnets



11001000 00010111 00010000 00000000
200. 23. 16. 0 /24

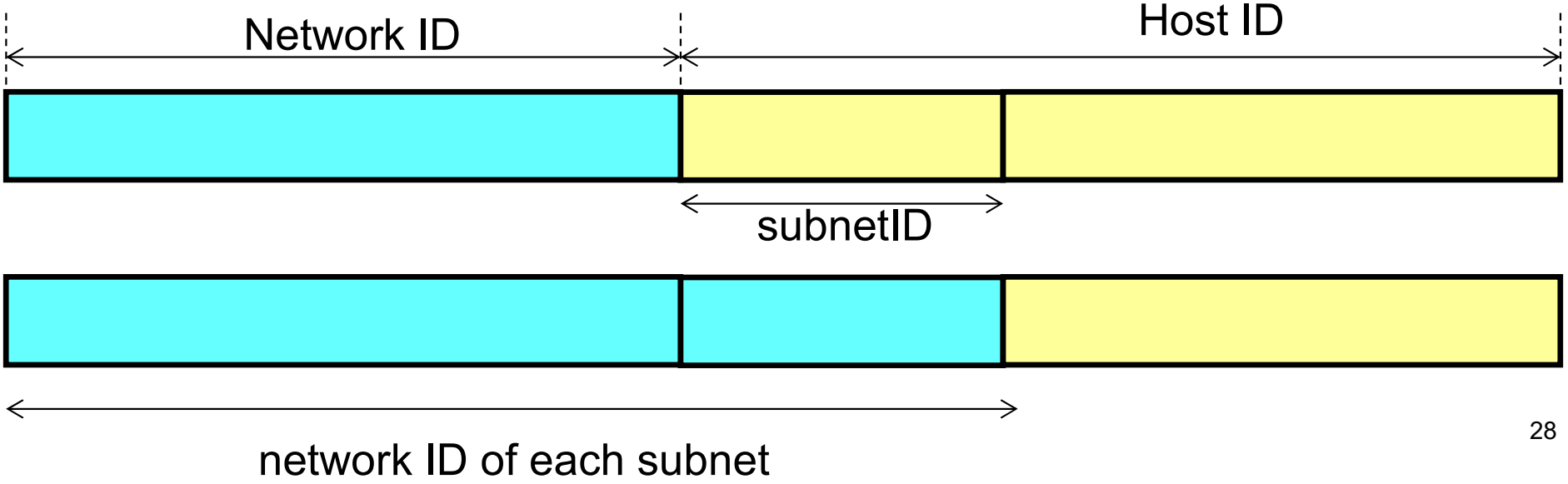


11001000 00010111 00010000 00000000
200. 23. 16. 0 /25

11001000 00010111 00010000 10000000
200. 23. 16. 128 /25

# Principle

- Divide a IP range into sub-ranges of equal size
- Take some bits from HostID part to distinguish subnets
  - each subnet contains IP addresses with a fixed values of subnet ID.



# Exercise: Dividing into subnets

- Given IP addresses in the range 200.23.16.0/24

1) Need to organize into 8 subnets

- Address of each subnetwork? Mask? Number of hosts/network
- 200.23.16.0 /27

2) General question: Need to create N subnets. Network address? Mask?

- Each network contains 14 hosts → /28
- Each network contains 30 hosts → /27
- Each network contains 31 hosts → /26
- Each network contains 70 hosts → /25

# Answers

- 200.23.16.0 /27 → **000**0 0000
- 200.23.16.32 /27 → **001**0 0000
- 200.23.16.64 /27 → **010**0 0000
- 200.23.16.96 /27 → **011**0 0000
- 200.23.16.128 /27 → **100**0 0000
- 200.23.16.160 /27 → **101**0 0000
- 200.23.16.192 /27 → **110**0 0000
- 200.23.16.224 /27 → **111**0 0000

# Addressing space of IPv4

- In theory
  - All between 0.0.0.0 ～ 255.255.255.255
  - Some special IP address (RFC1918)

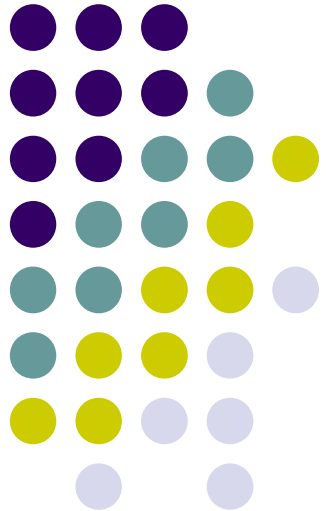
| | |
|---|---|
| Private address | 10.0.0.0/8<br>172.16.0.0/12<br>192.168.0.0/16 |
| Loopback address | 127.0.0.0 |
| Multicast address | 224.0.0.0<br>～239.255.255.255 |

- Self assigned IP address: 169.254.0.0/16

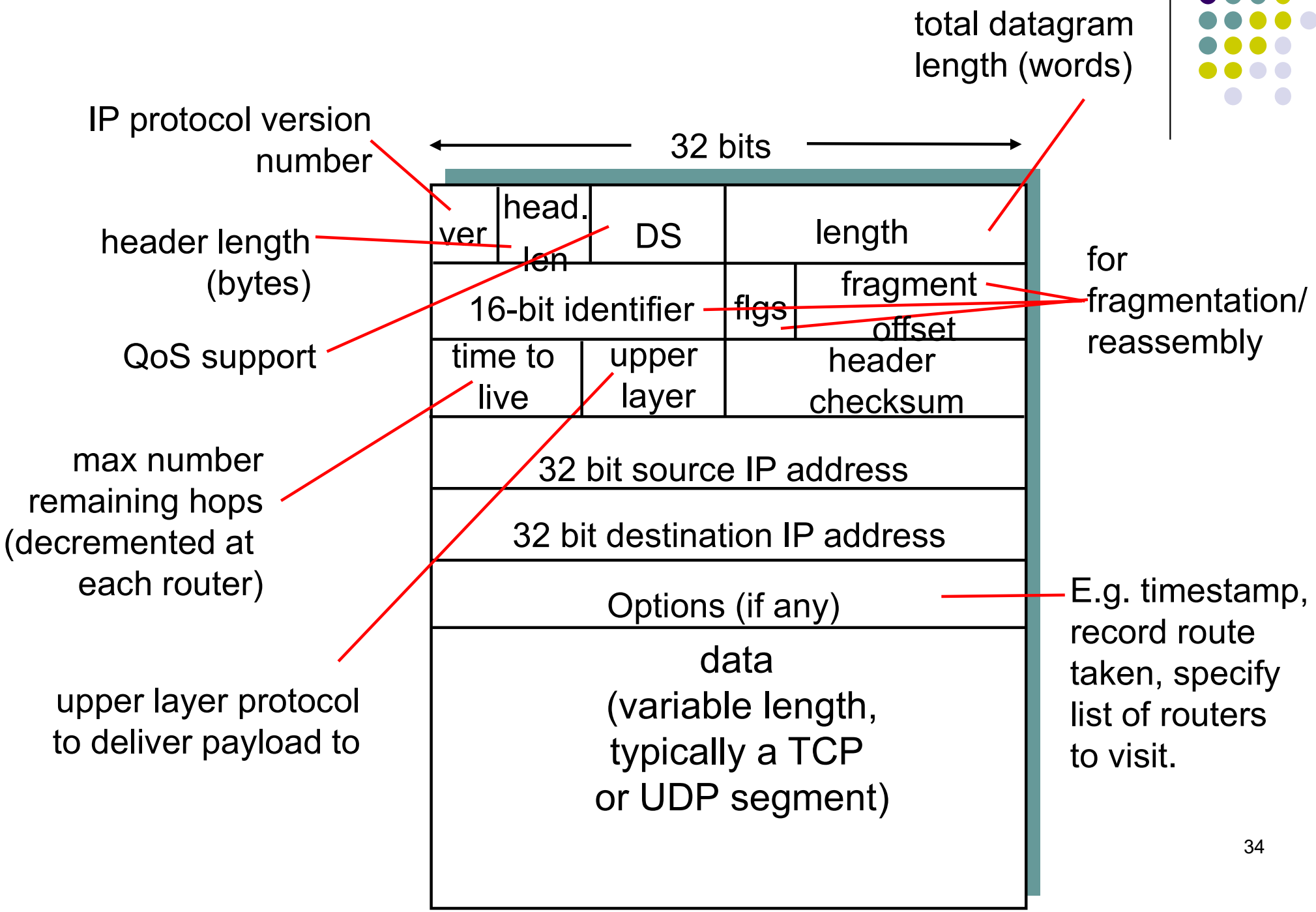
# Attention about IP

- Currently IPv4: 32 bits
  - 133.113.215.10 (IPv4)
- IPv6 is also widely used: 128bits
  - 2001:200:0:8803::53 (IPv6)
  - Fix 64 first bit for subnet ID, 64 last bit belongs to interface ID.
  - Security feature is integrated

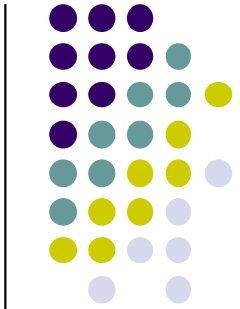
# IP package



# Header of IP

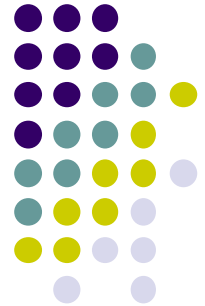

total datagram length (words)
IP protocol version number
32 bits
header length (bytes)
QoS support
ver
head. len
DS
length
16-bit identifier
flgs
fragment offset
for fragmentation/ reassembly
time to live
upper layer
header checksum
max number remaining hops (decremented at each router)
32 bit source IP address
32 bit destination IP address
Options (if any)
E.g. timestamp, record route taken, specify list of routers to visit.
data (variable length, typically a TCP or UDP segment)
upper layer protocol to deliver payload to

# IP header (1)



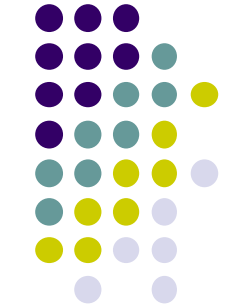
- Version (4 bits)
  - IPv4
  - IPv6
- Header length: 4bits
  - In word unit (4 bytes)
  - Min:  5
  - Max: 60

# IP header (2)



- DS (Differentiated Service : 8bits)
  - Old name: Type of Service
  - Used for QoS management by some router
  - Diffserv

# IP header (3)



- Length: total length including header (16 bits)
  - In bytes unit
  - Max: 65536
- 16 bits Identifier– ID of the packet
  - Used for identifying all fragments of the same packet when it is fragmented
  - Flag
  - Fragmentation offset – offset of the first byte of the fragment in its original packet

# IP header (4)

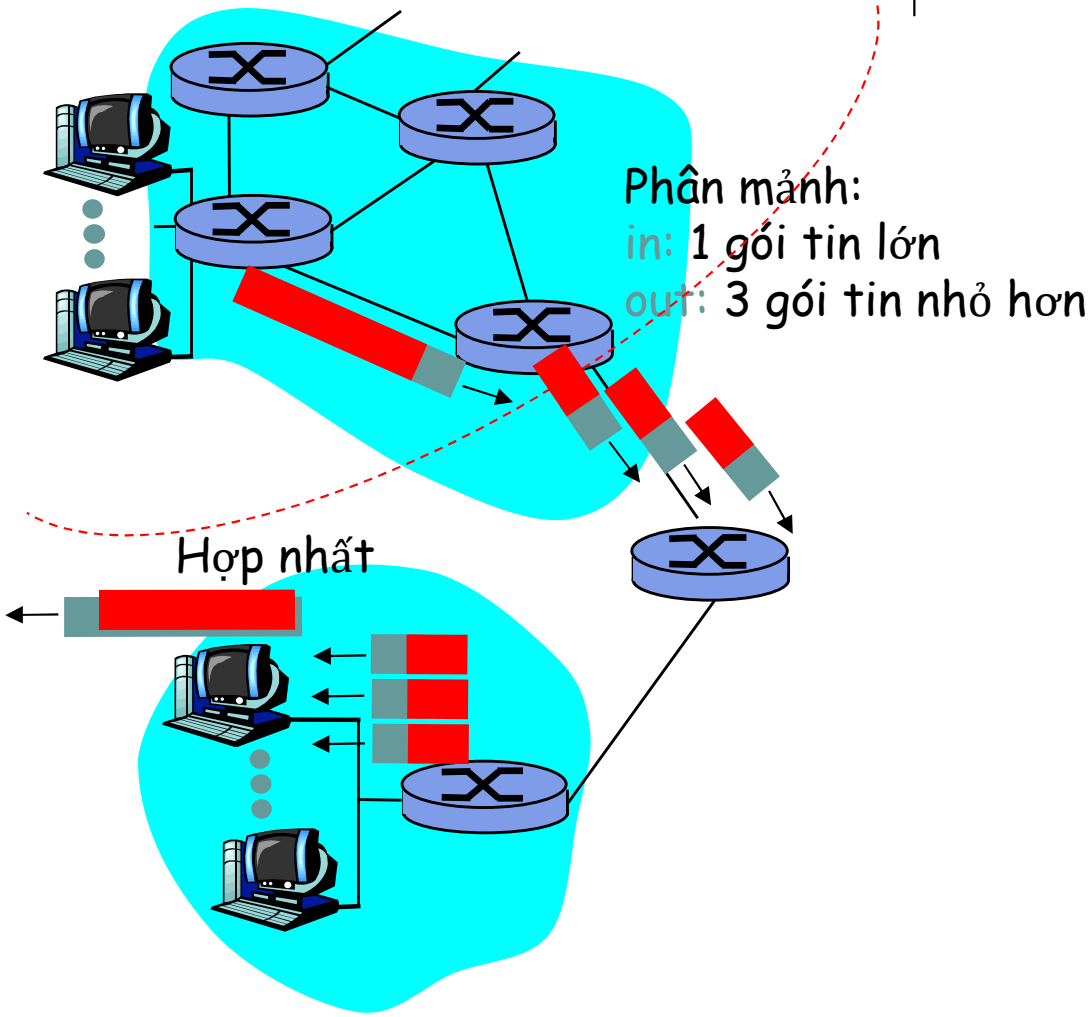
- TTL, 8 bits – Time to live
  - Maximum number of hops (router) the packet is allowed to travel
  - Max: 255
  - Router decreases TTL 1 unit when processing a packet
  - The packet will be destroyed when TTL reaches to 0
- Protocol – upper layer protocol
  - Transport protocol (TCP, UDP,…)
  - Other network layer protocols that are encapsulated in IP packet (ICMP, IGMP, OSPF )

# IP header (4)

- Checksum: to detect corruption in the header of IPv4 data packets
- Source IP address
  - 32 bit, address of the sender
- Destination IP address
  - 32 bit, address of the receiver.

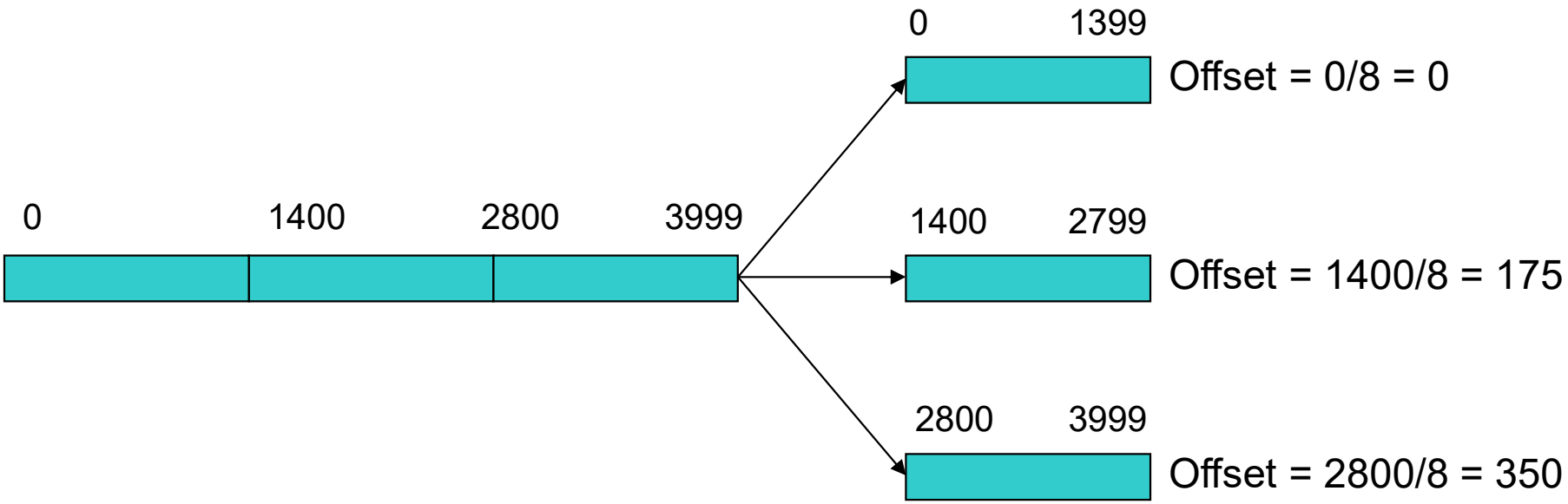
# Packet fragmentation (1)

- Each link has a fixed MTU (Maximum transfering unit)
- Different media have different MTU
- If IP packet > MTU, it should be
  - Divided into small fragments
  - Gathered at the destination



# Packet fragmentation (2)

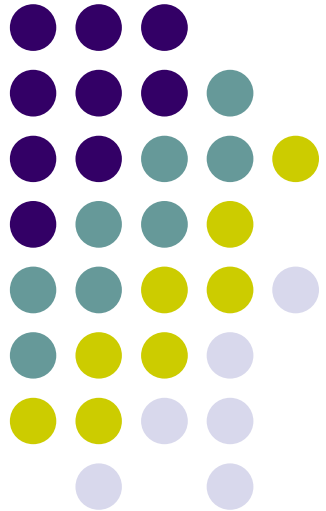
- Offset
  - Position of the fragment in the original packet
  - In 8 bytes units



# Internet Control Message Protocol

Packet format

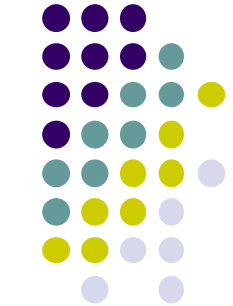
Ping and Traceroute



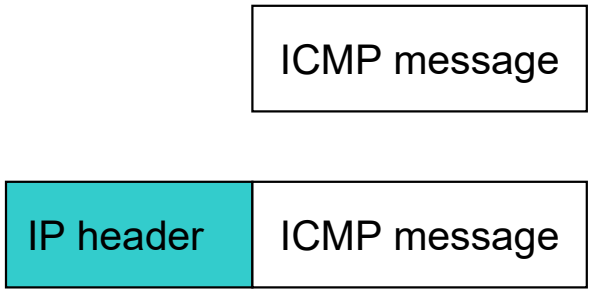
# Idea of ICMP (1)

- IP is unreliable, connectionless
  - Lack of supporting and error control mechanism
- ICMP is used in network layer for providing information exchange between sender and receivers
  - Error information: inform that a packet cannot reach a host, a network or a port.

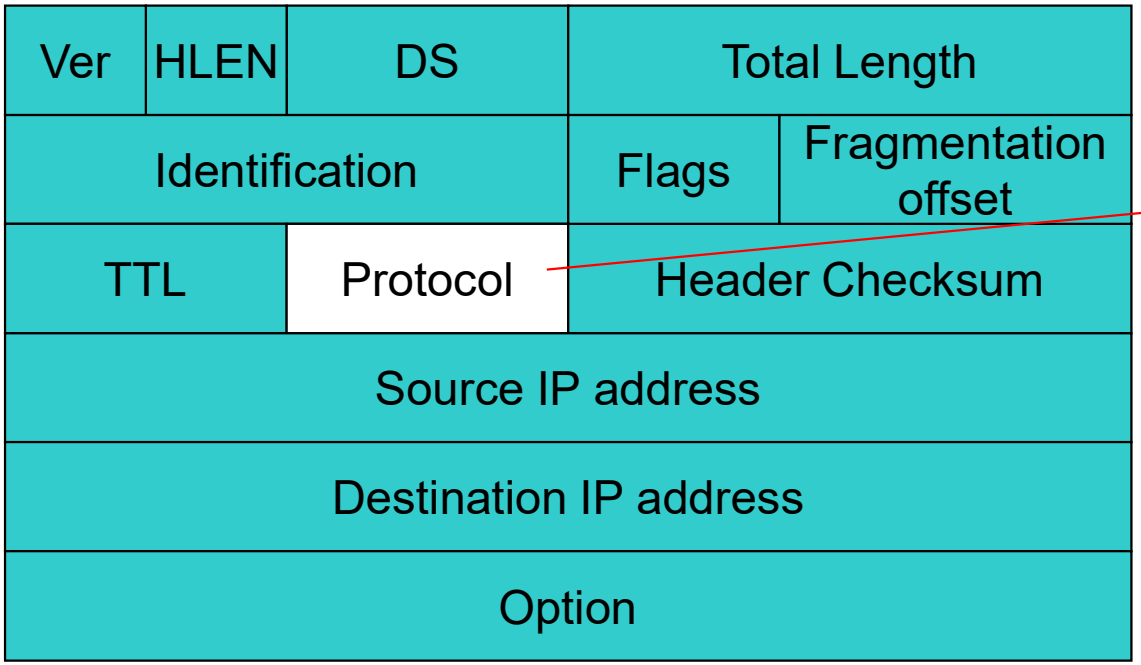
# Idea of ICMP (2)



- Also in network layer but is “above” IP
  - ICMP message is encapsulated in IP
- ICMP message: Type, Code, with 8 first bytes of the error IP message



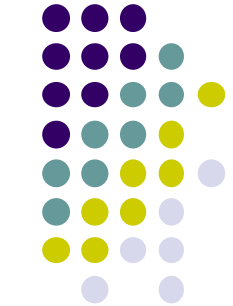
# IP header and Protocol field



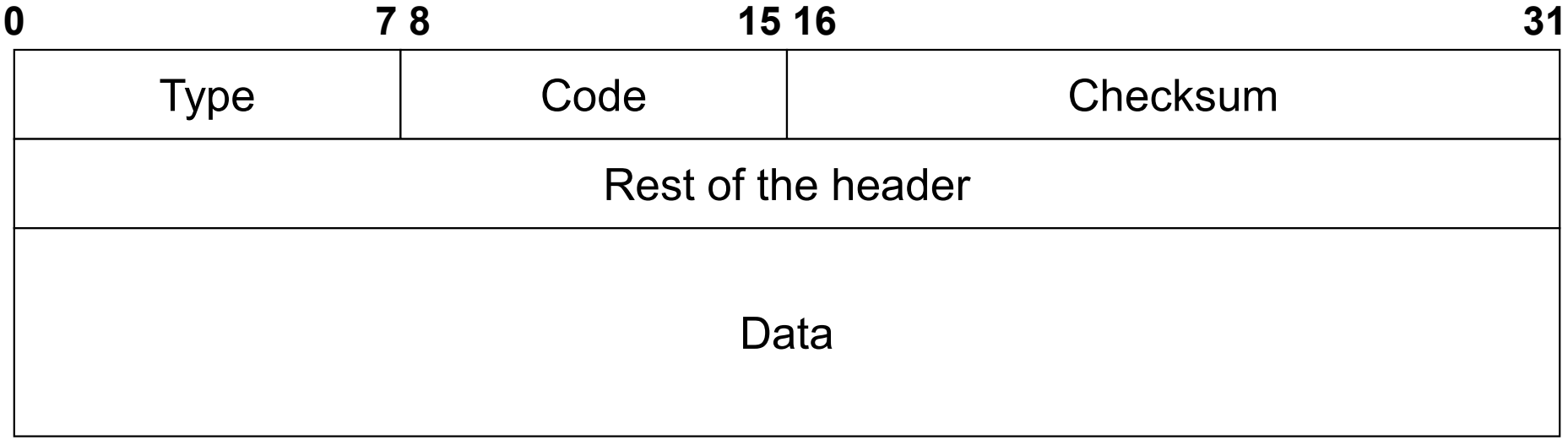

Protocol:

- 1: ICMP
- 2: IGMP
- 6: TCP
- 17: UDP
- 89: OSPF

# ICMP message format



- Type: type of ICMP message
- Code:  cause of error
- Checksum
- Rest of header varies according on type



# Some ICMP message types

| | | | |
|---|---|---|---|
| ICMP Message Type | Error-reporting messages | 3 | Destination Unreachable |
| | | 4 | Source quench (nguồn giảm tốc độ) |
| | | 5 | Redirection |
| | | 11 | Time exceeded |
| | | 12 | Parameter problem |
| | Query messages | 8 or 0 | Echo reply or request |
| | | 13 or 14 | Time stamp request or reply |
| | | 17 or 18 | Address mask request or reply |
| | | 9 or 10 | Router advertisement or solicitation |

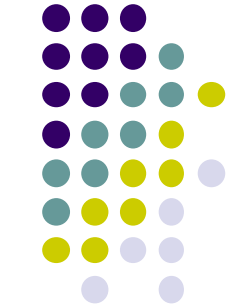
# ICMP and debuging tools

- ICMP always works transparently for users
- Users can use ICMP by using some debuging tools
  - ping
  - traceroute

# Ping and ICMP

- ping
  - Test a connection
  - Sender sends packet “ICMP echo request”
  - Receiver responses with “ICMP echo reply”
- Data field contains the time stamp when the packet is sent
  - For calculating RTT (round-trip time)

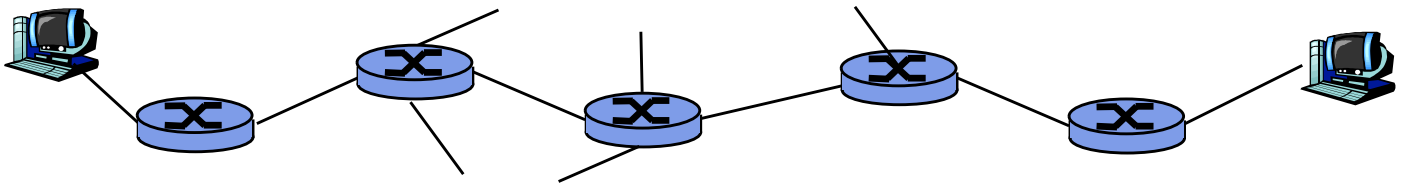
# Ping: Example



```
C:\Documents and Settings\hongson>ping www.yahoo.co.uk

Pinging www.euro.yahoo-eu1.akadns.net [217.12.3.11] with 32 bytes of data:

Reply from 217.12.3.11: bytes=32 time=600ms TTL=237
Reply from 217.12.3.11: bytes=32 time=564ms TTL=237
Reply from 217.12.3.11: bytes=32 time=529ms TTL=237
Reply from 217.12.3.11: bytes=32 time=534ms TTL=237

Ping statistics for 217.12.3.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 529ms, Maximum = 600ms, Average = 556ms
```

# Traceroute and ICMP

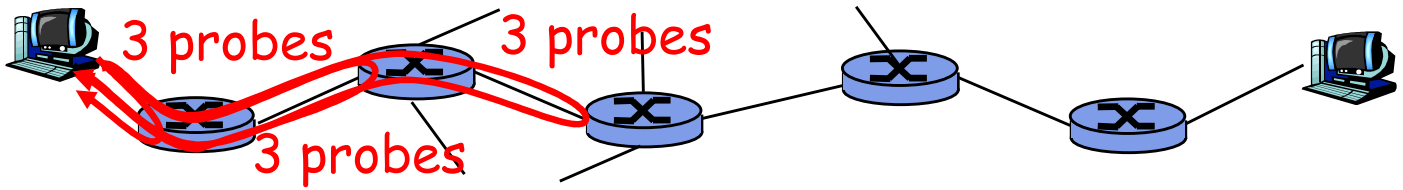
- Sender send many packets to receiver
  - First packet has TTL =1
  - Second packet has TTL=2, …
- When packet number n arrives to $n^{th}$ router:
  - Router destroys the packer
  - Router send back an ICMP packet (type 11, code 0) containing IP address of the router
- Based on the reply message, the sender can calculate RTT



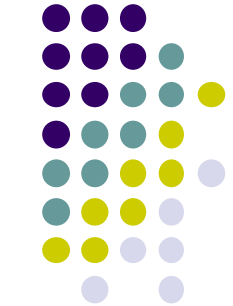
# Traceroute and ICMP

Termination condition

- When ICMP echo packet arrive to the destination
- When source receives ICMP “host unreachable” (type 3, code 3)

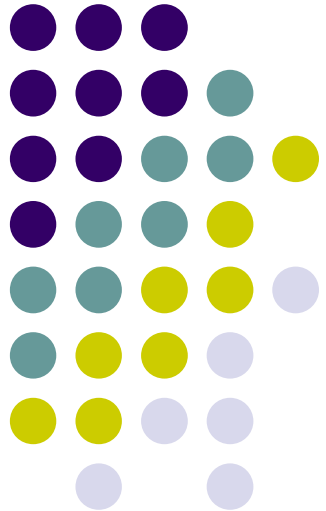


# Traceroute: Example



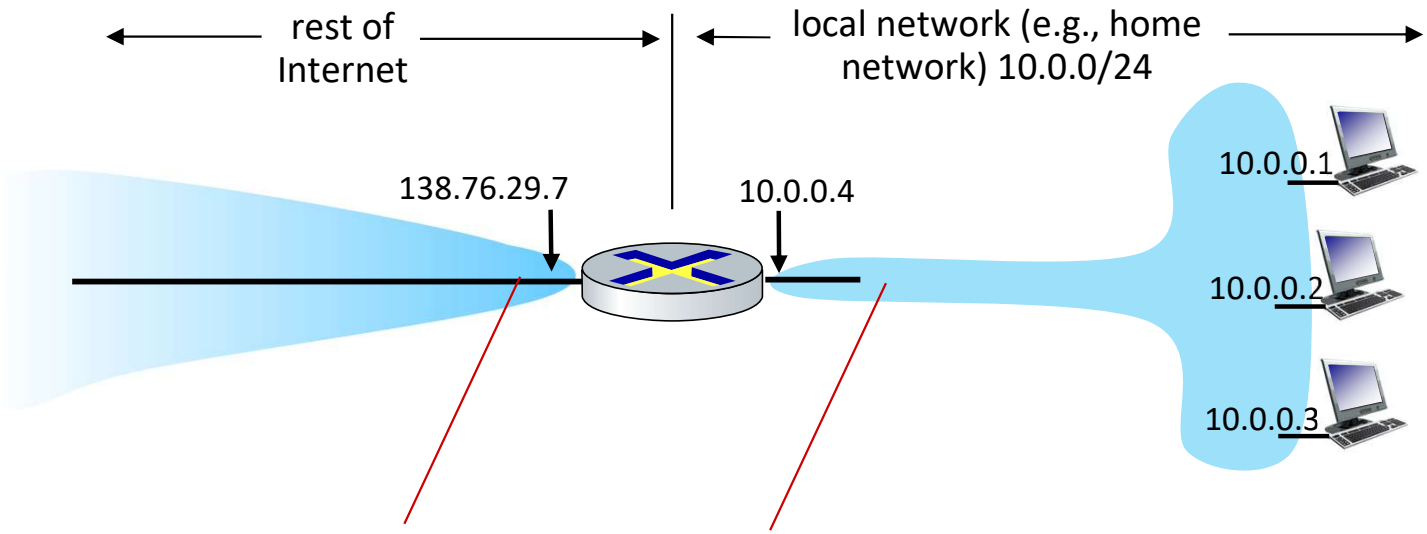
```
C:\Documents and Settings\hongson>tracert www.jaist.ac.jp

Tracing route to www.jaist.ac.jp [150.65.5.208]
over a maximum of 30 hops:

  1    1 ms    <1 ms    <1 ms  192.168.1.1
  2   15 ms    14 ms    13 ms  210.245.0.42
  3   13 ms    13 ms    13 ms  210.245.0.97
  4   14 ms    13 ms    14 ms  210.245.1.1
  5  207 ms   230 ms    94 ms  pos8-2.br01.hkg04.pccwbtn.net [63.218.115.45]
  6    *      403 ms   393 ms  0.so-0-1-0.XT1.SCL2.ALTER.NET [152.63.57.50]
  7  338 ms   393 ms   370 ms  0.so-7-0-0.XL1.SJC1.ALTER.NET [152.63.55.106]
  8  402 ms   404 ms   329 ms  POS1-0.XR1.SJC1.ALTER.NET [152.63.55.113]
  9  272 ms   288 ms   310 ms  193.ATM7-0.GW3.SJC1.ALTER.NET [152.63.49.29]
 10   205 ms   206 ms   204 ms  wide-mae-gw.customer.alter.net [157.130.206.42]
 11   427 ms   403 ms   370 ms  ve-13.foundry2.otemachi.wide.ad.jp [192.50.36.62]
 12   395 ms   399 ms   417 ms  ve-4.foundry3.nezu.wide.ad.jp [203.178.138.244]
 13   355 ms   356 ms   378 ms  ve-3705.cisco2.komatsu.wide.ad.jp [203.178.136.193]
 14   388 ms   398 ms   414 ms  c76.jaist.ac.jp [203.178.138.174]
 15   438 ms   377 ms   435 ms  www.jaist.ac.jp [150.65.5.208]

Trace complete.
```

# Network address translation



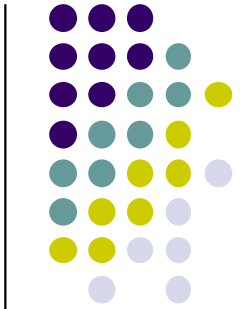
# NAT: network address translation

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



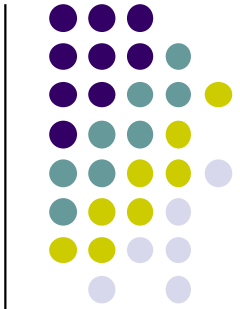

*all* datagrams *leaving* local network have *same* source NAT IP address: 138.76.29.7, but *different* source port numbers

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

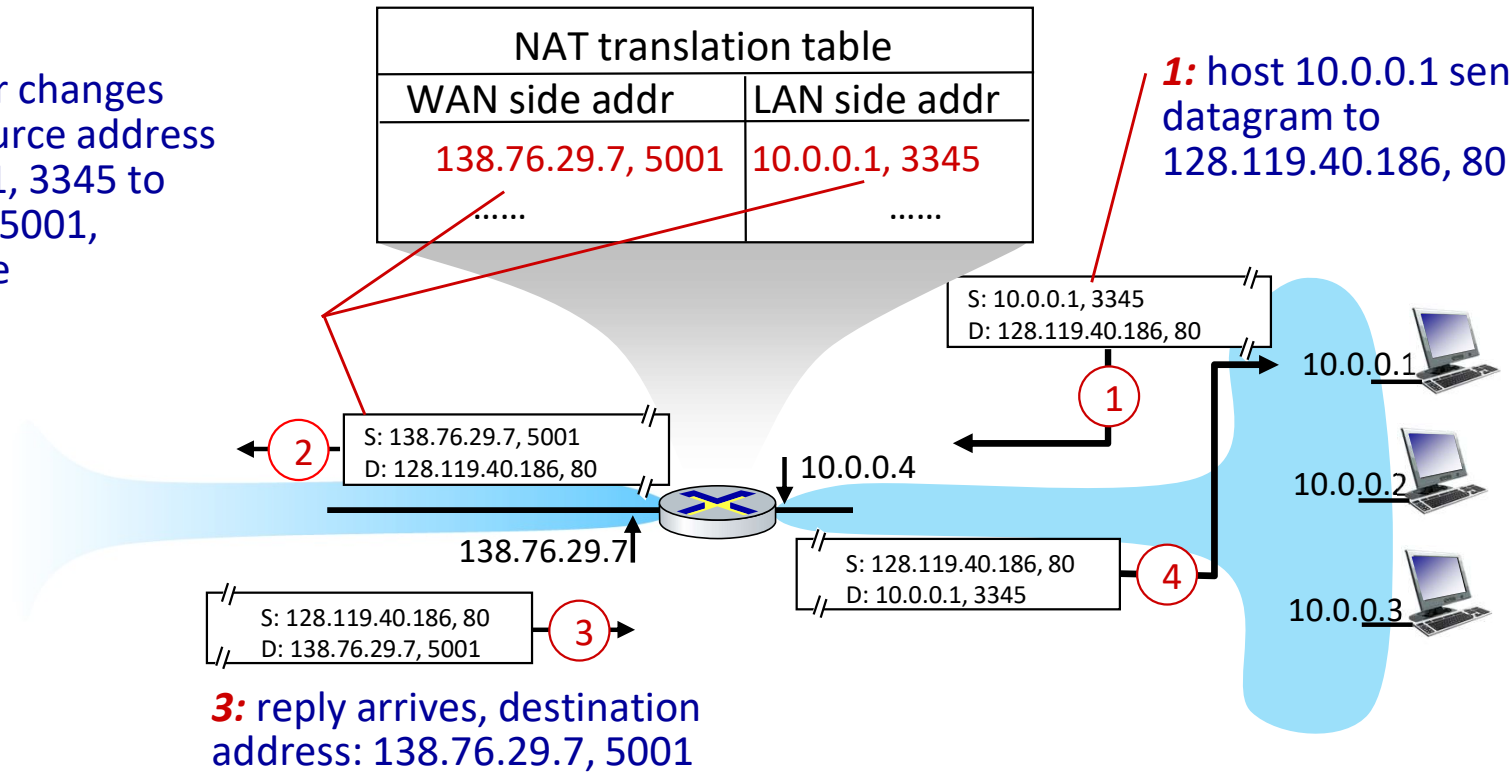


# NAT: network address translation

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

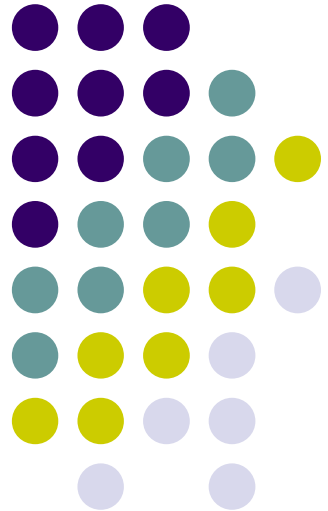
# NAT: network address translation




Static vs dynamic NAT

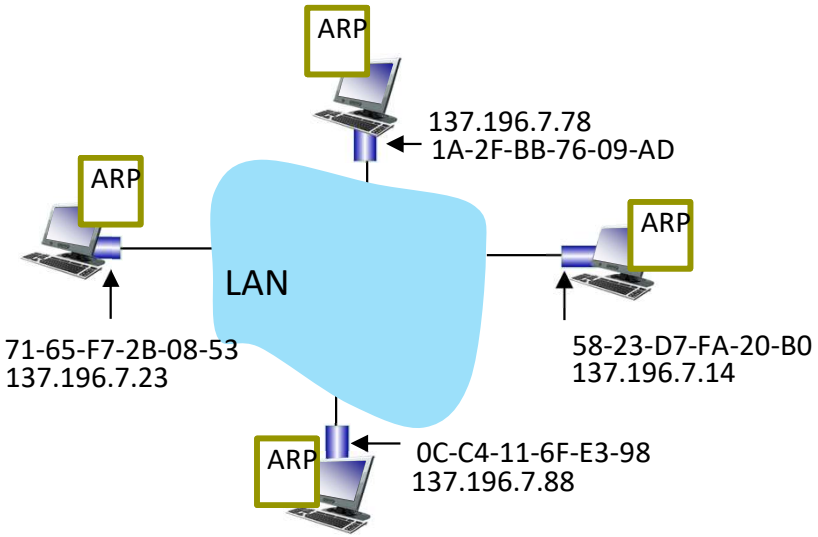
- Simple NAT: One private IP for one public IP, fixed
- Dynamic NAT: an available public IP will be assigned for a private IP dynamically

# Address resolution protocol



# ARP: address resolution protocol

*Question:* how to determine interface's MAC address, knowing its IP address?




ARP table: each IP node (host, router) on LAN has 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 in action

example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

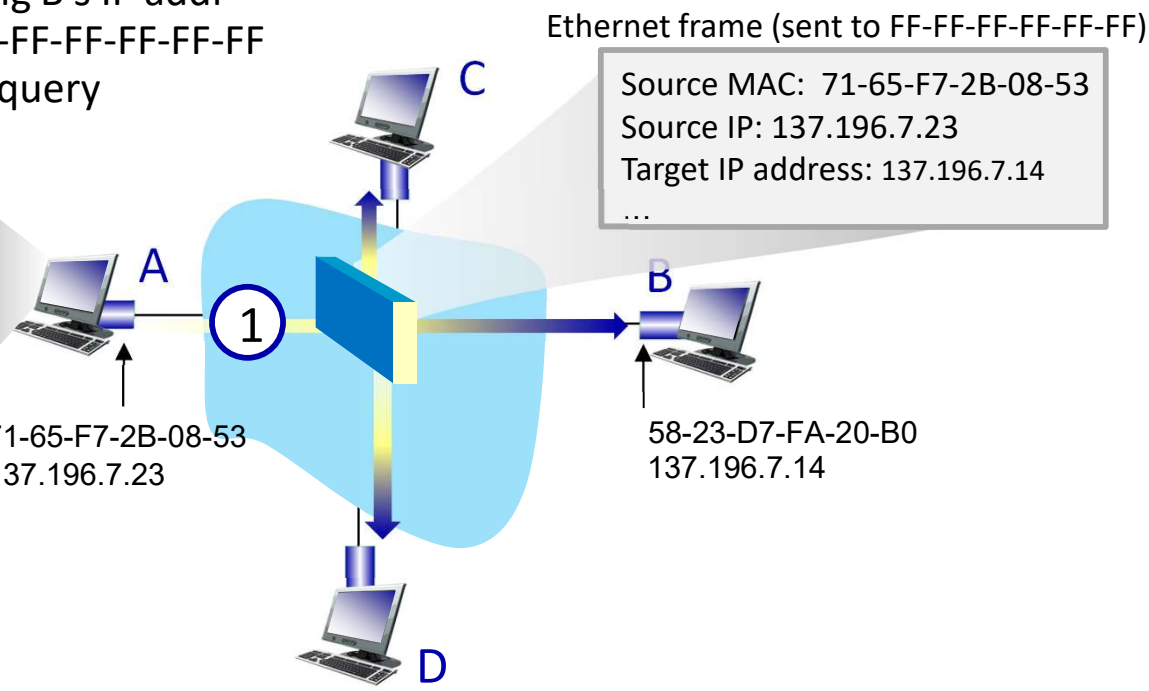
A broadcasts ARP query, containing B's IP addr

1

- destination MAC address = FF-FF-FF-FF-FF-FF
- all nodes on LAN receive ARP query

ARP table in A

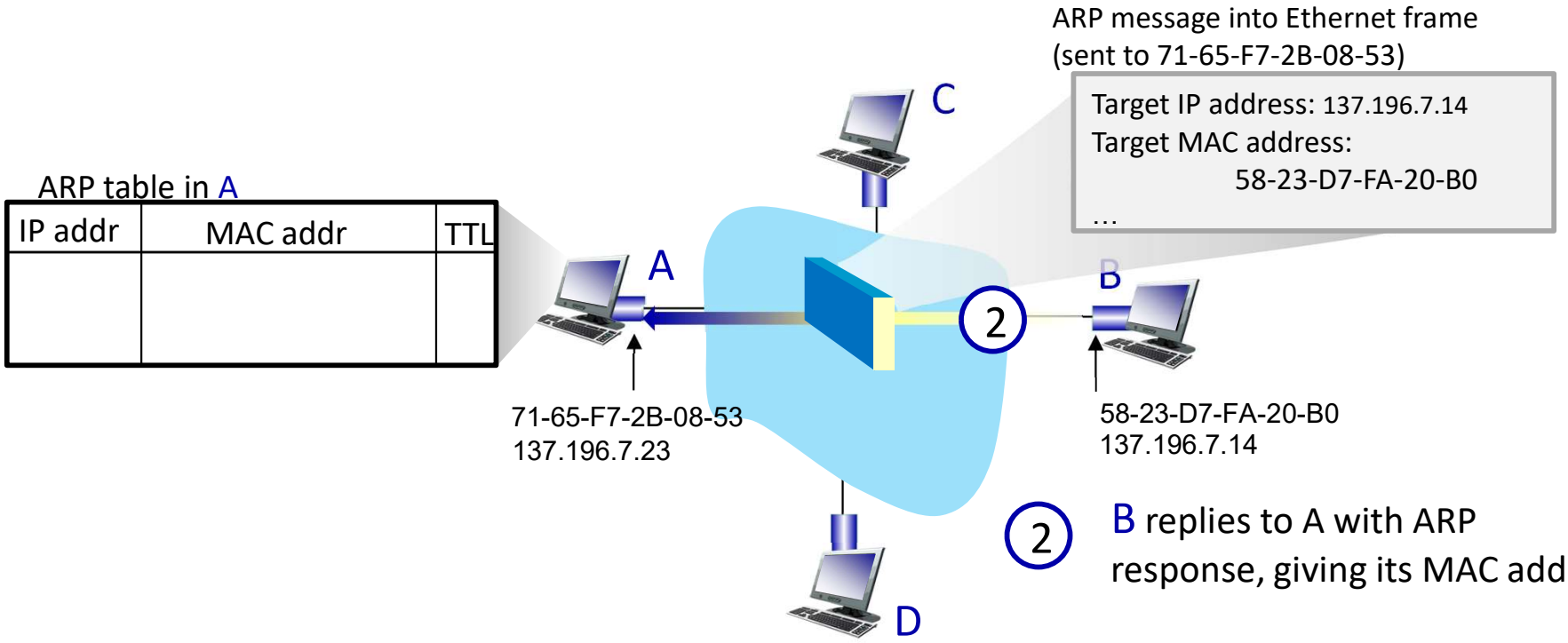
| IP addr | MAC addr | TTL |
|---|---|---|
| | | |



# ARP protocol in action

example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address



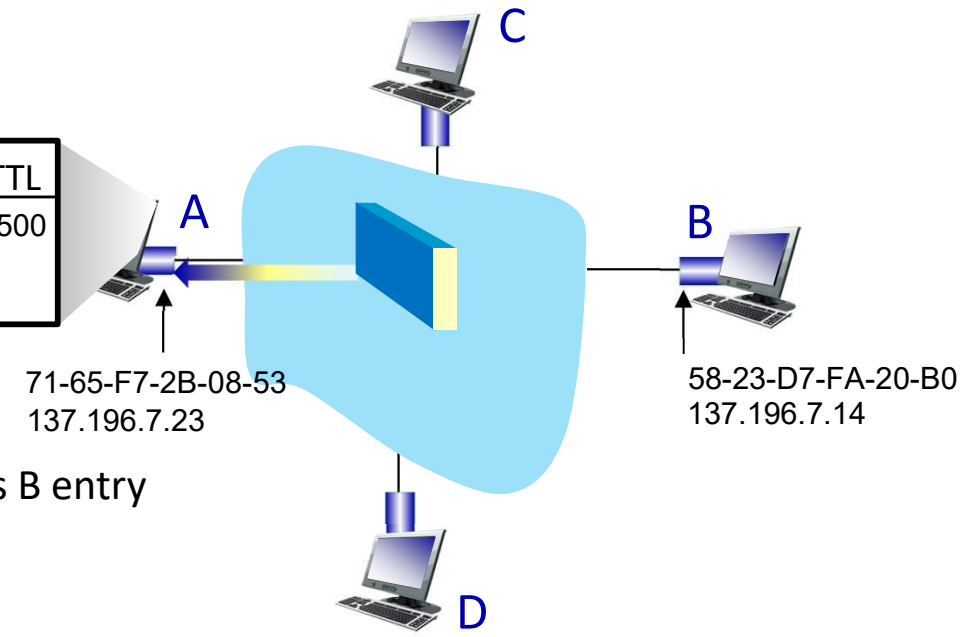
# ARP protocol in action

example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

ARP table in A

| IP addr | MAC addr | TTL |
|---|---|---|
| 137.196.7.14 | 58-23-D7-FA-20-B0 | 500 |

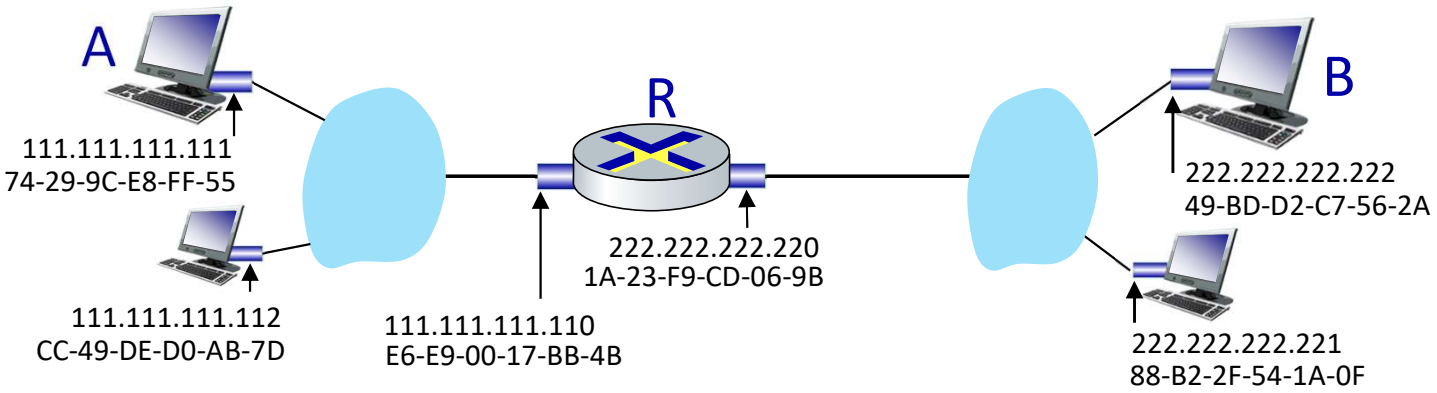



3 A receives B's reply, adds B entry into its local ARP table

# Routing to another subnet: addressing

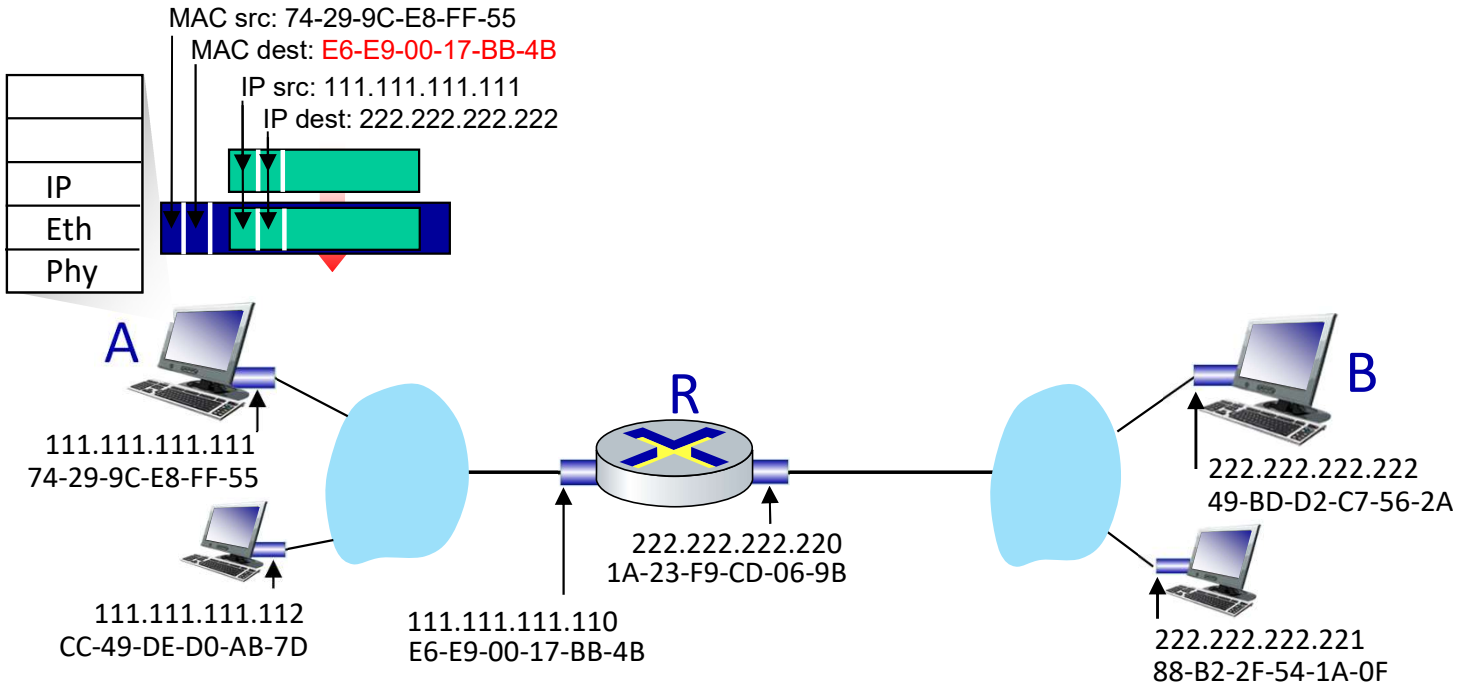
walkthrough: sending a datagram from *A* to *B* via *R*

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



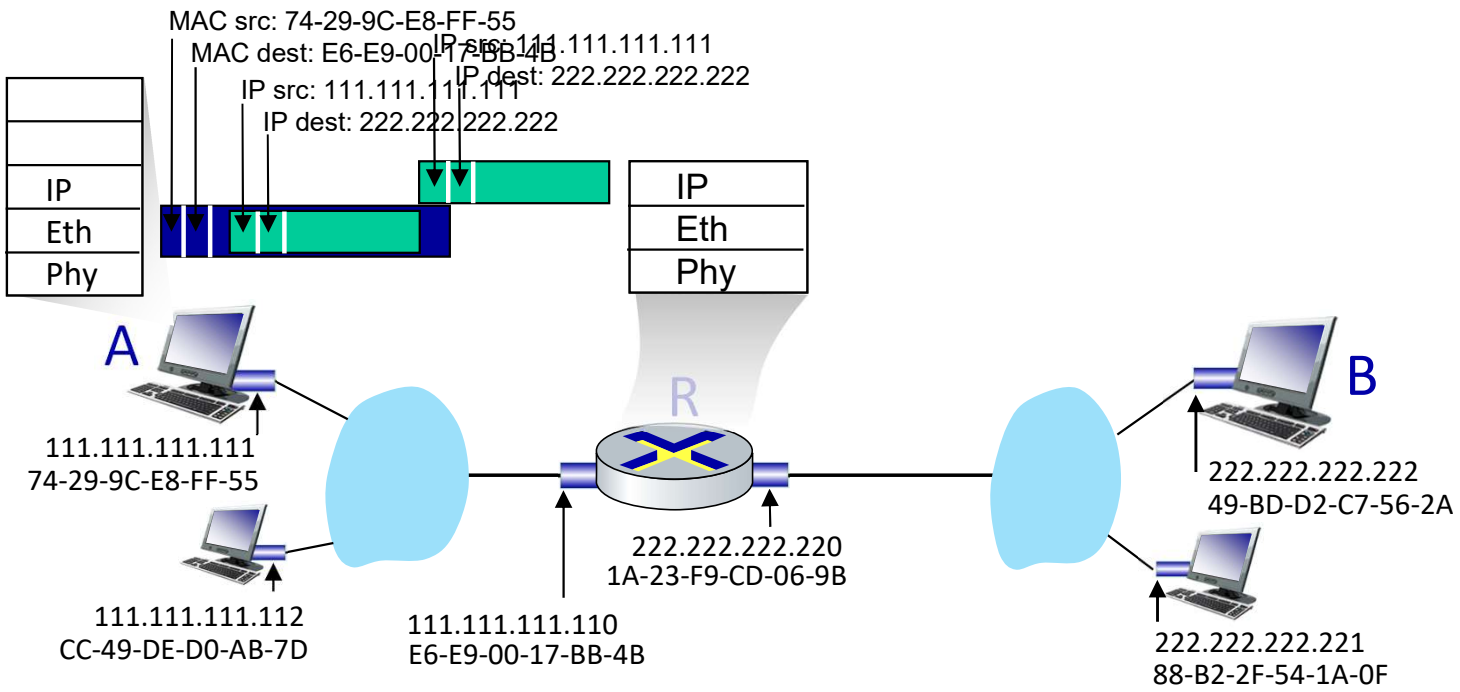
# Routing to another subnet: addressing

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



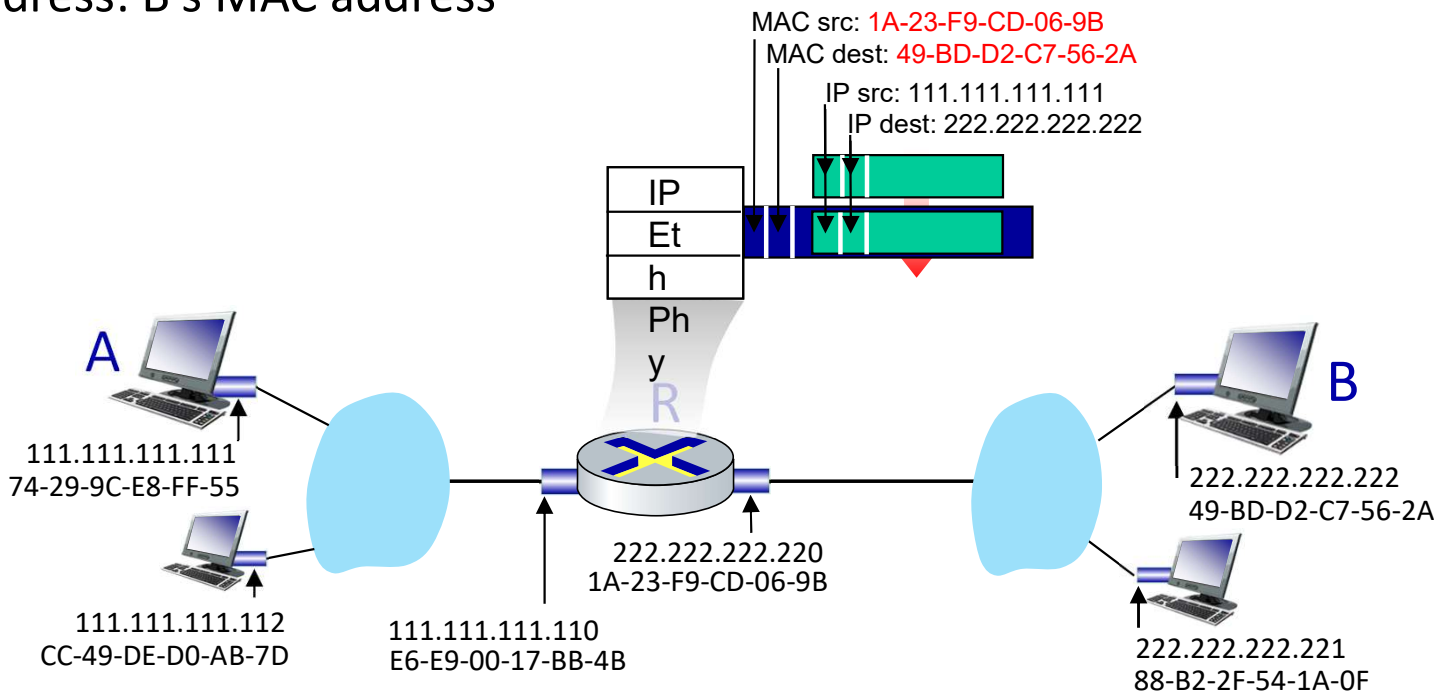
# Routing to another subnet: addressing

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



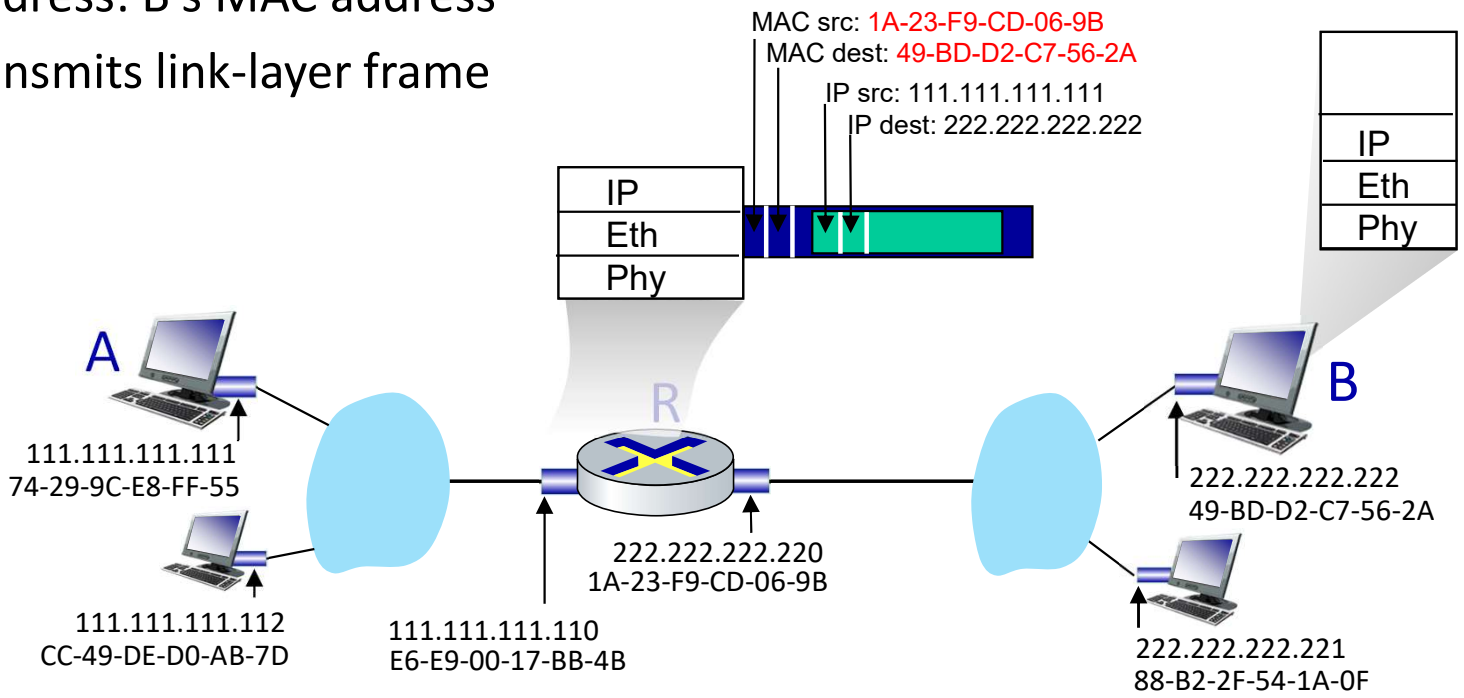
# Routing to another subnet: addressing

- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. Frame destination address: B's MAC address



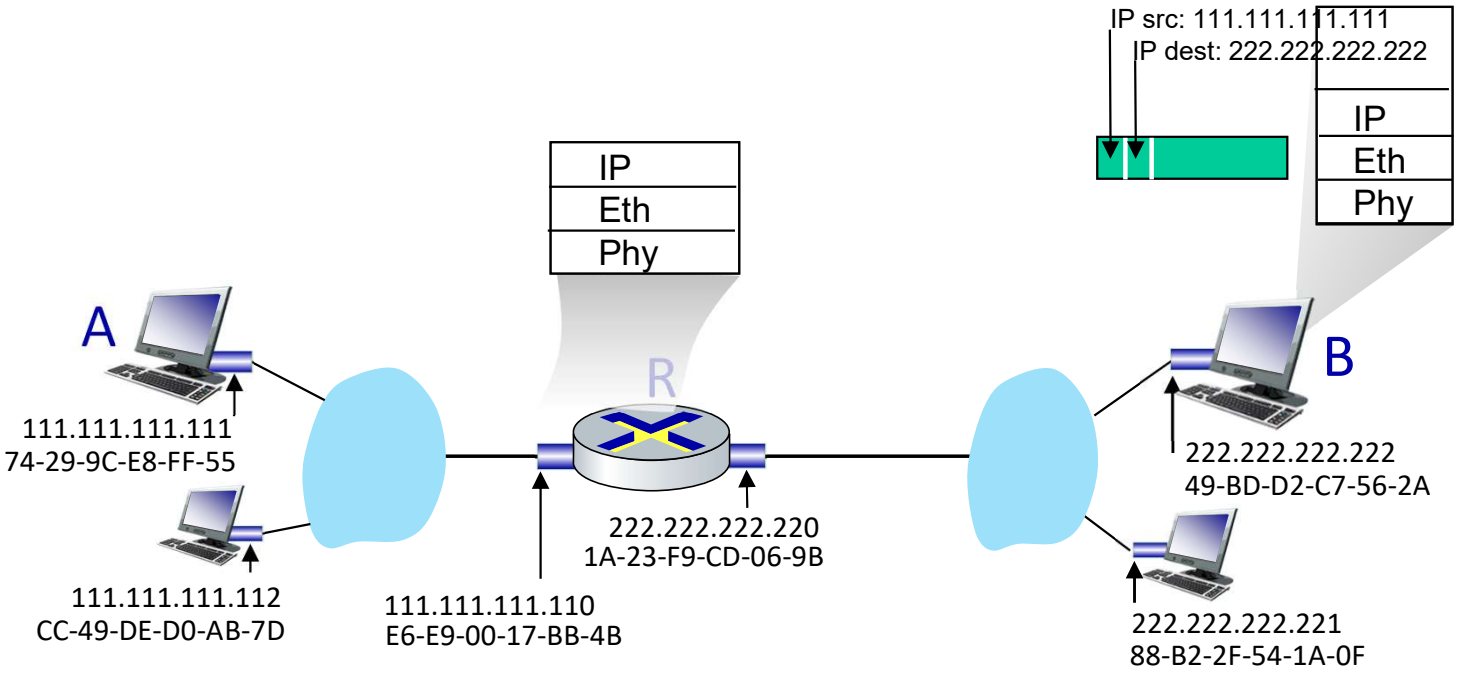
# Routing to another subnet: addressing

- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. Frame destination address: B's MAC address
- transmits link-layer frame

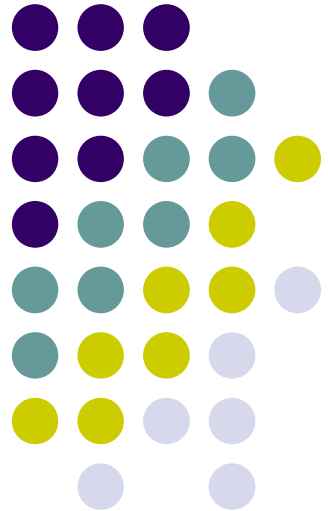


# Routing to another subnet: addressing

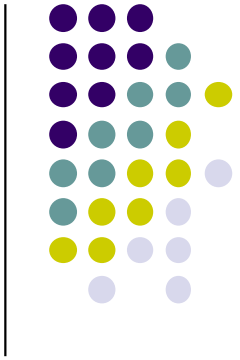
- B receives frame, extracts IP datagram destination B
- B passes datagram up protocol stack to IP



# Dynamic Host Configuration Protocol



# DHCP: Dynamic Host Configuration Protocol



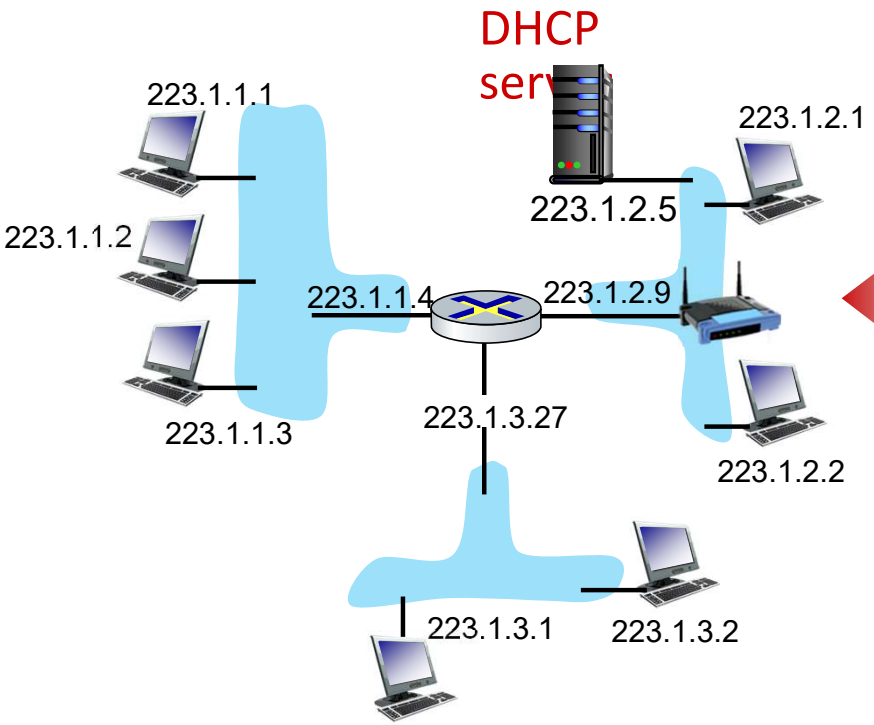
goal: host *dynamically* obtains IP address from network server when it "joins" network

- 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

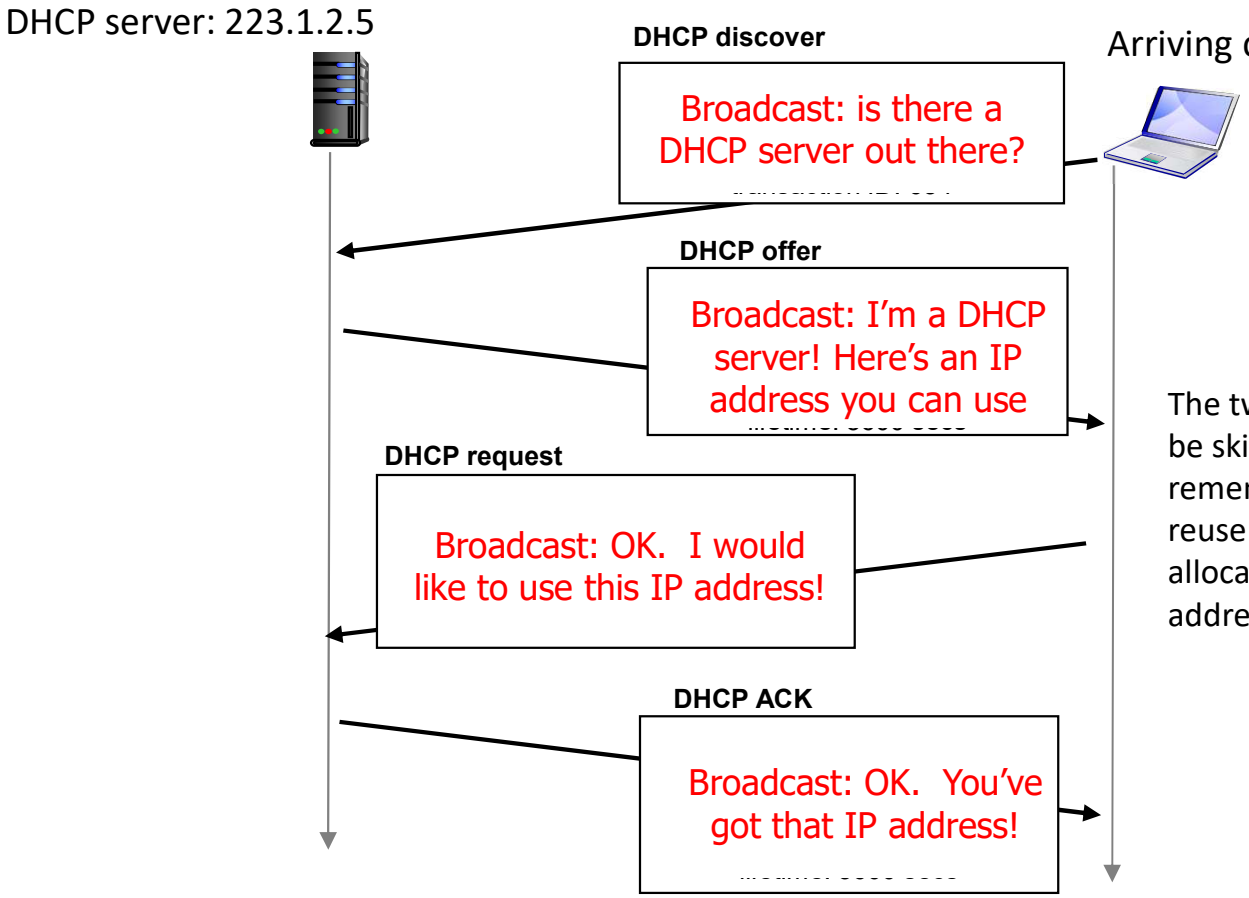
# DHCP client-server scenario



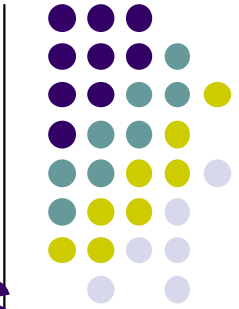

Typically, DHCP server will be co-located in router, serving all subnets to which router is attached

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]



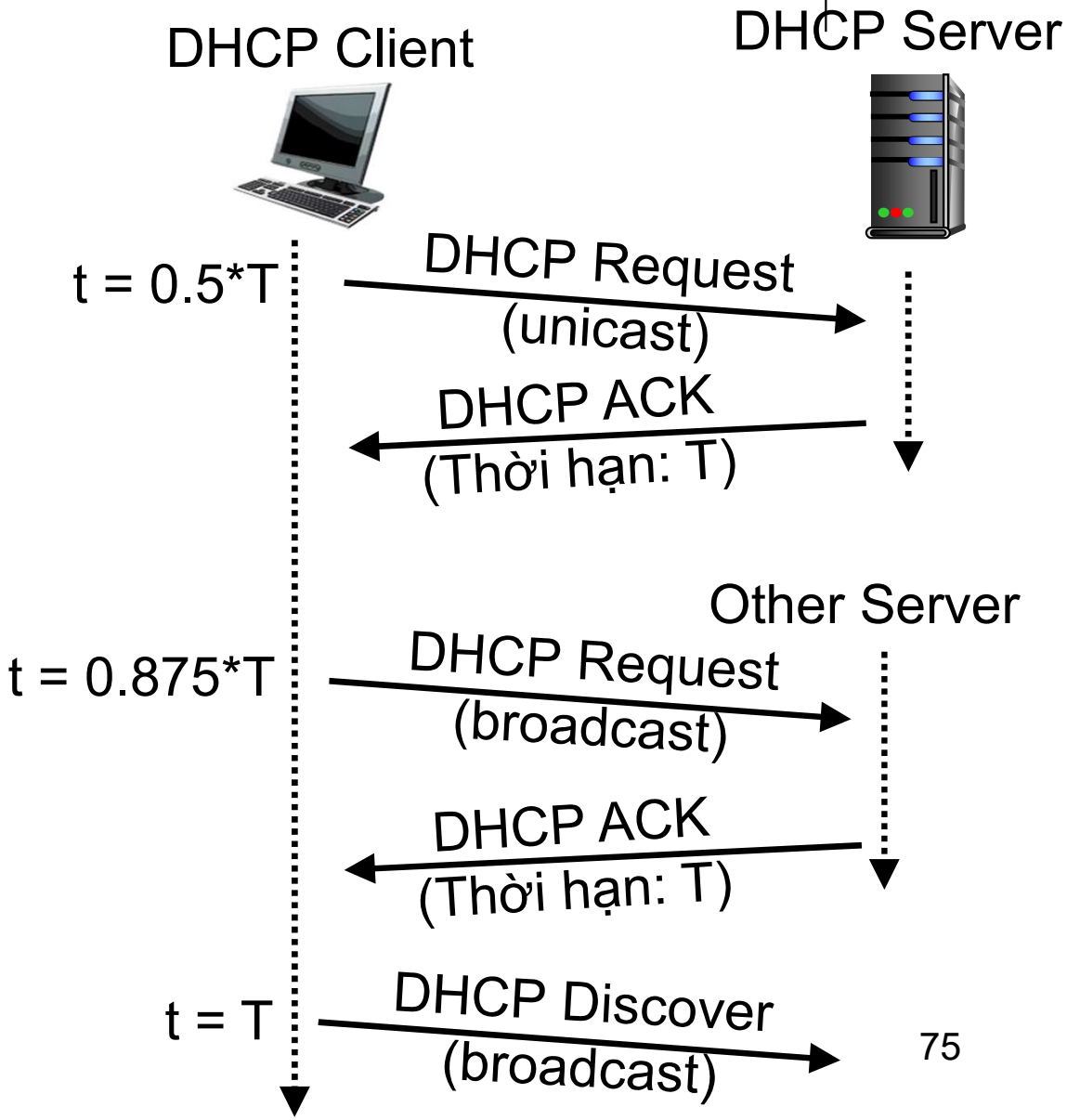
# DHCP: more than IP addresses

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)

# Extend using

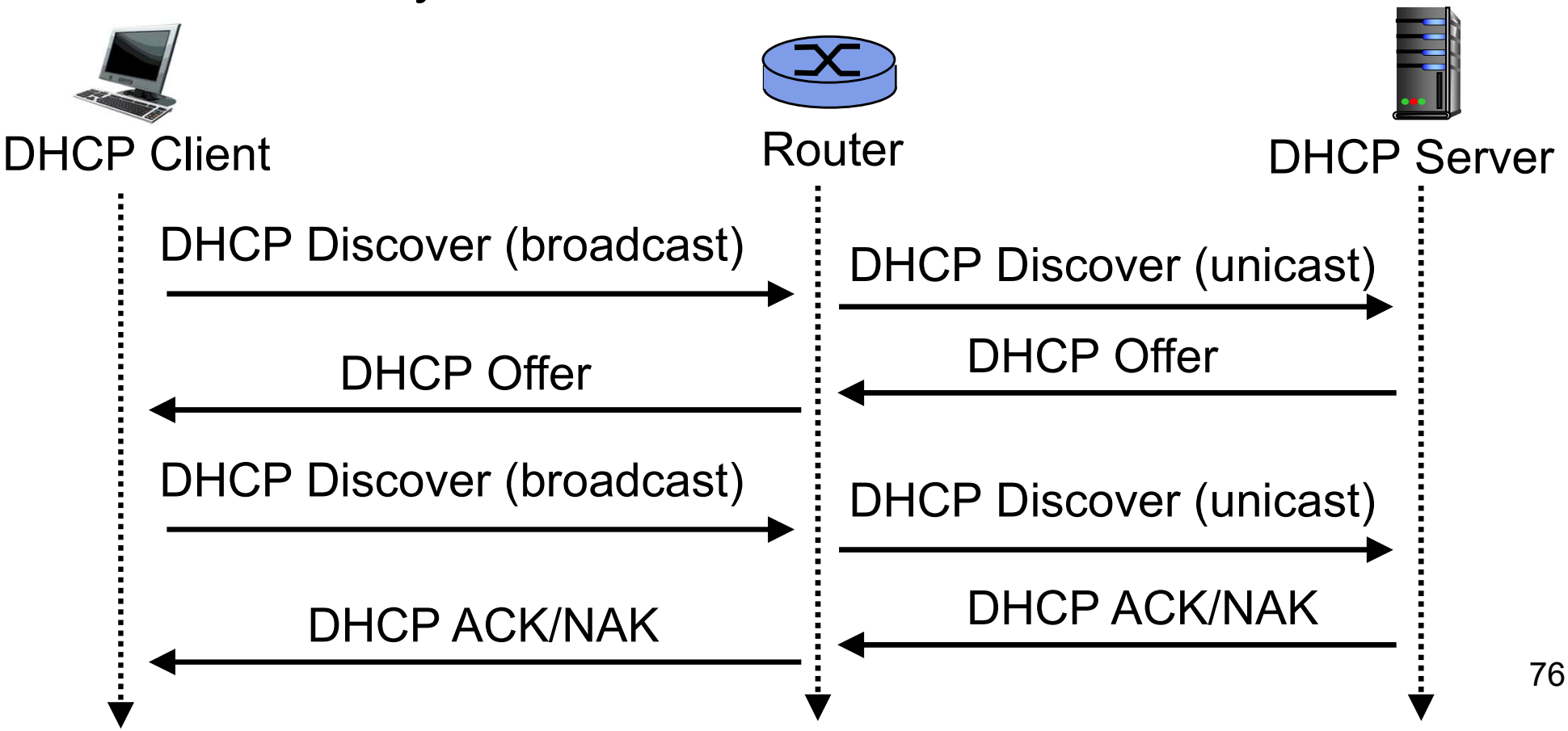
- Limit time → extend
- t = 0.5*T, client sends DHCP Request to DHCP Server to request extension
- No DHCP ACK, then t = 0.875*T, client sends the broadcast DHCP Request
- No DHCP ACK, while t = T, client sends DHCP Discover



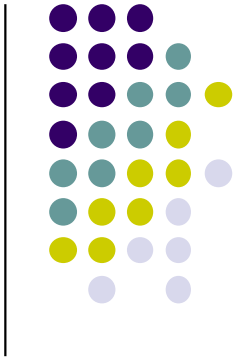
# DHCP Relay

- DHCP Server stays on another subnet → broadcast packets will be forwared by routers

→ DHCP Relay on routers



# Summary



- More on Network Layer
- Internet protocol
- IP address and IP packet format
- ICMP
  - Ping
  - Traceroute
- DHCP
- NAT
- ARP