

# I226

# Computer Networks

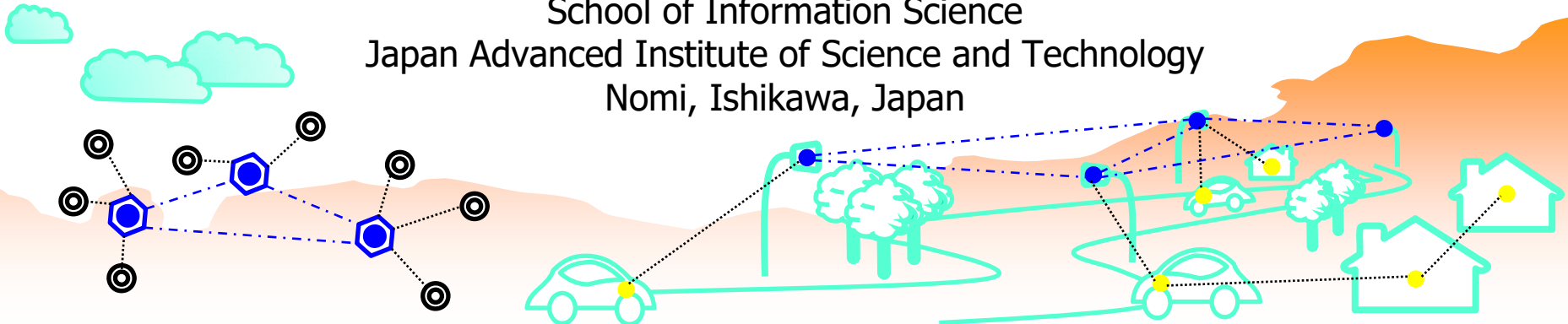
## Chapter 4

## Network Layer I

---

**Assoc. Prof. Yuto Lim**

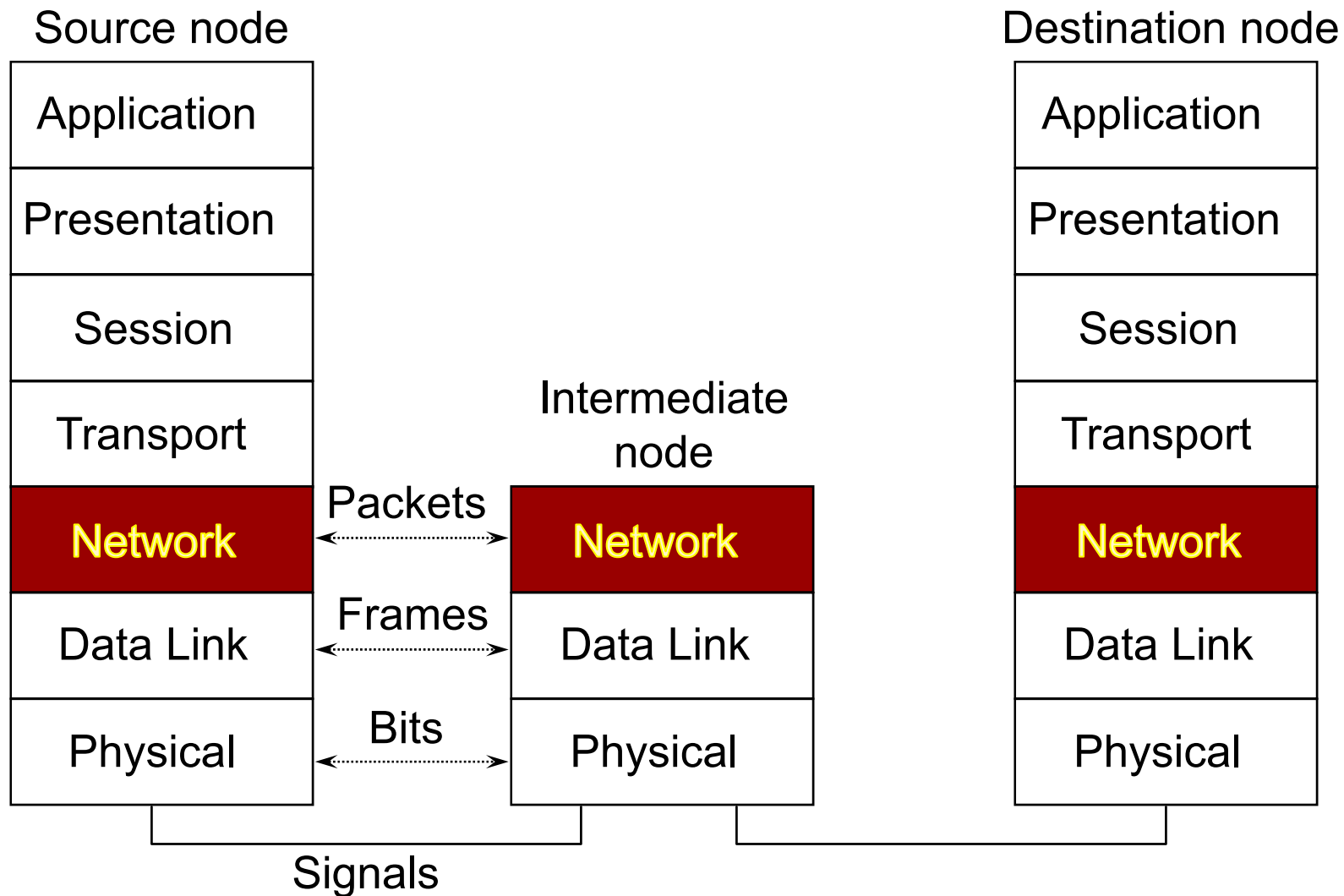
School of Information Science  
Japan Advanced Institute of Science and Technology  
Nomi, Ishikawa, Japan



# **Objectives of this Chapter**

- Provide an understanding what are the five functions of network layer
- Explain in-depth knowledge on addressing issue and its network design
- Give an explanation of Internet control protocols for address resolution and multicast

# Network Layer

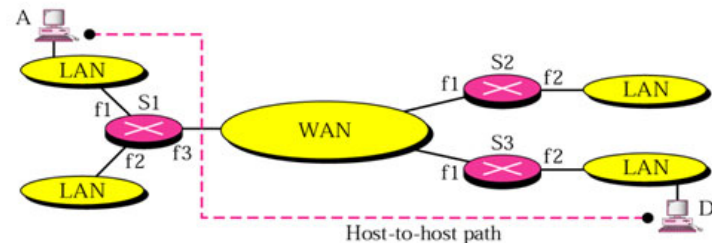


# Outline

- Introduction to Network Layer
- Packetizing
- Fragmentation and Re-assembly
- Addressing: IP Address
  - Classless InterDomain Routing (CIDR)
- Routing
- Internet Control Protocols
  - Internet Control Message Protocol (ICMP)
  - Address Resolution Protocol (ARP)
  - Internet Group Management Protocol (IGMP)

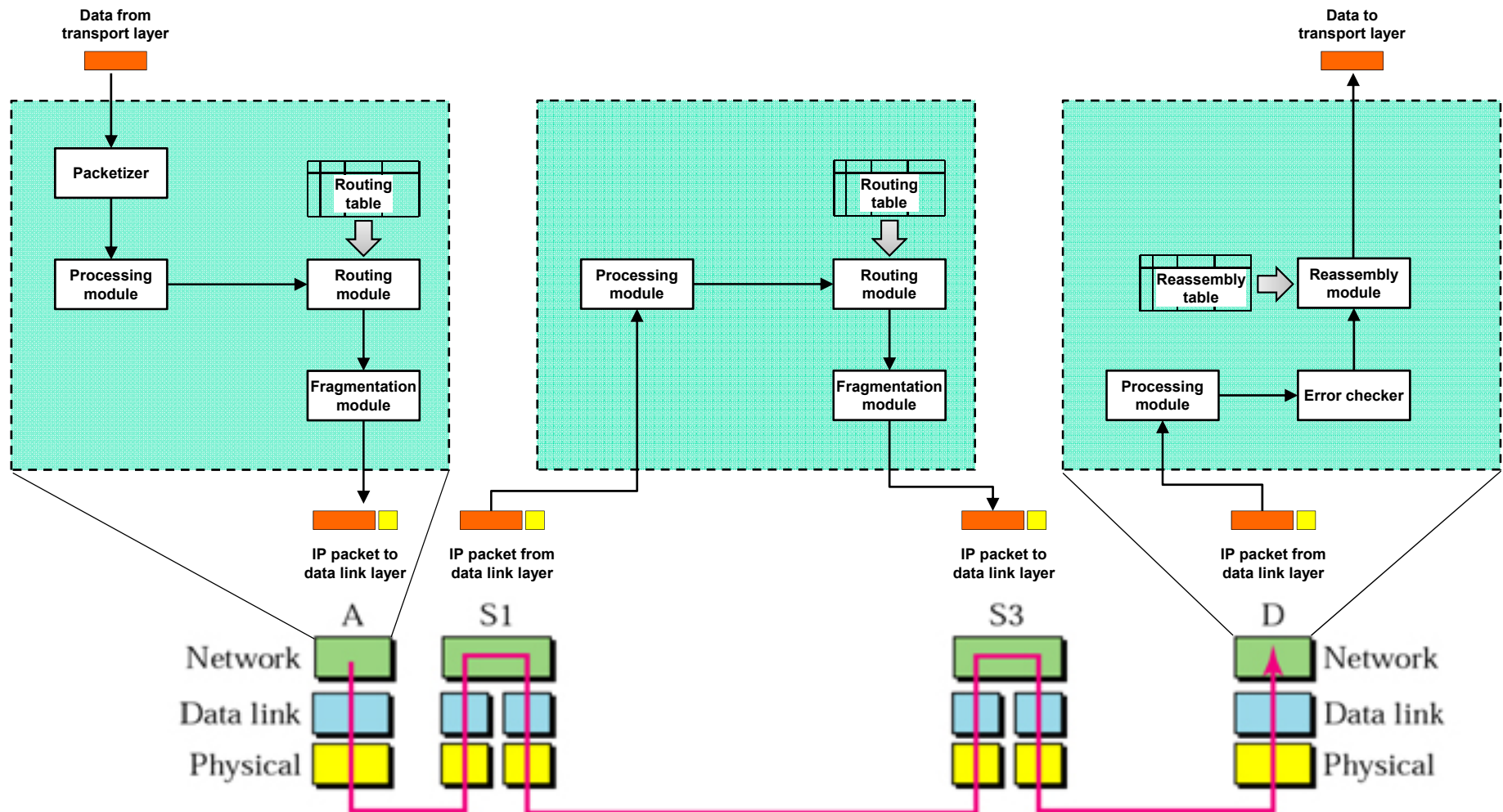
# Functions of Network Layer

- Main task of the network layer is to move packets from the **source** host to the **destination** host



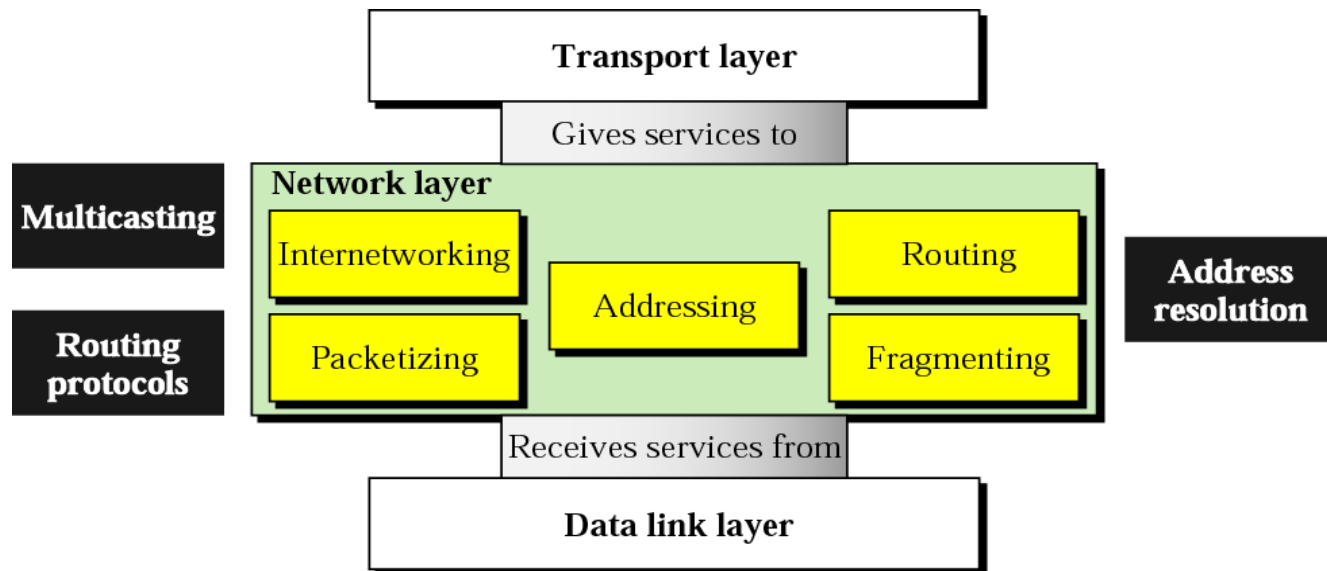
- ① Packetizing
- ② Fragmentation and reassembling
- ③ Addressing: provides structured addresses to name networks
  - ▣ Uses an addressing scheme, e.g., IP address
- ④ Internetworking (**Chapter 5**)
- ⑤ Routing algorithm: performs network routing (**Chapter 8**)
  - ▣ How to forward packets?

# Functions of Network Layer



# Where is the Network Layer?

- From OSI, the **Network Layer** rests between the upper layer (Transport layer) and the lower layer (Data Link Layer)
- From the TCP/IP model, the network layer is called the **Internet Layer** and it rests between the upper Transport Layer and the lower Host to Network Layer



# A Connecting Network

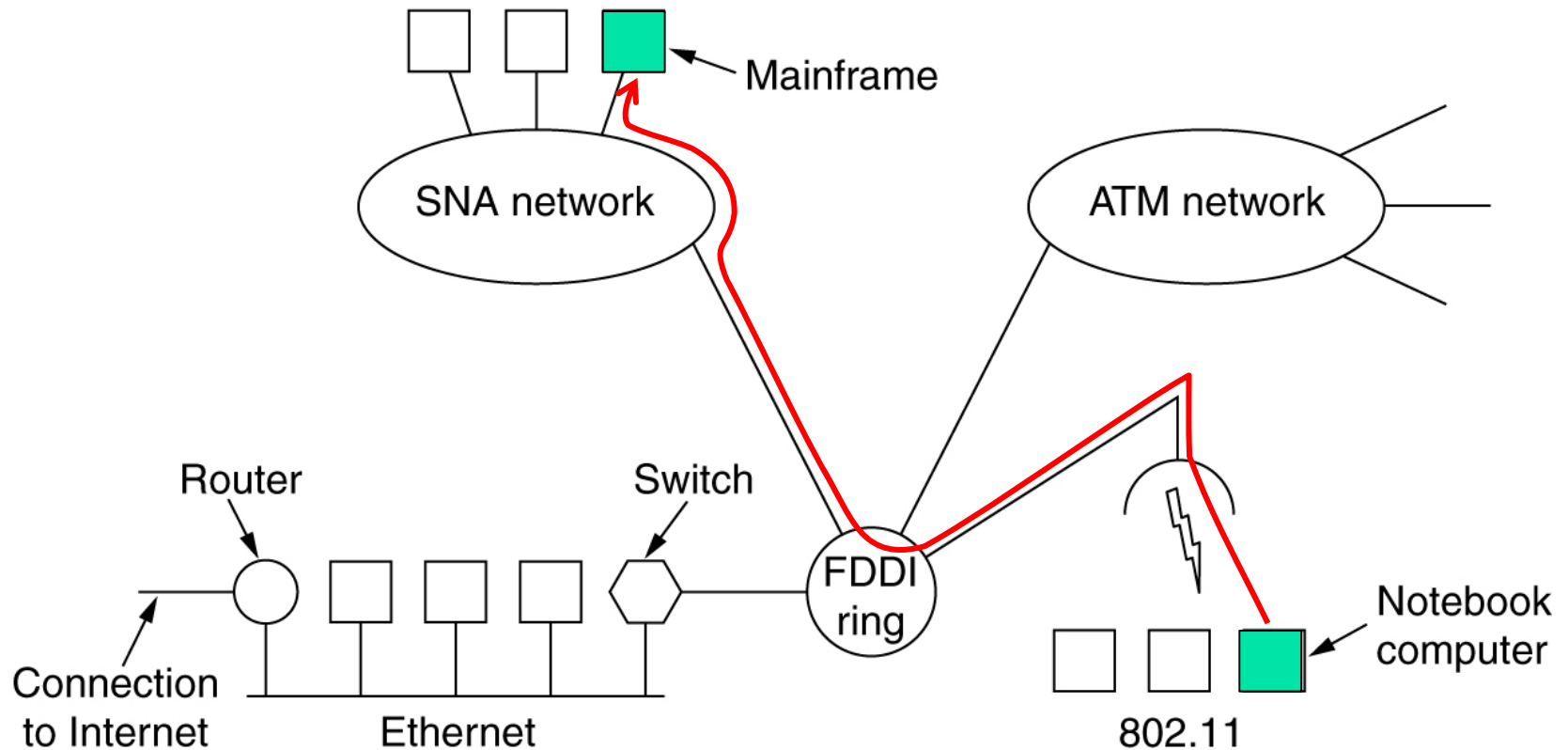
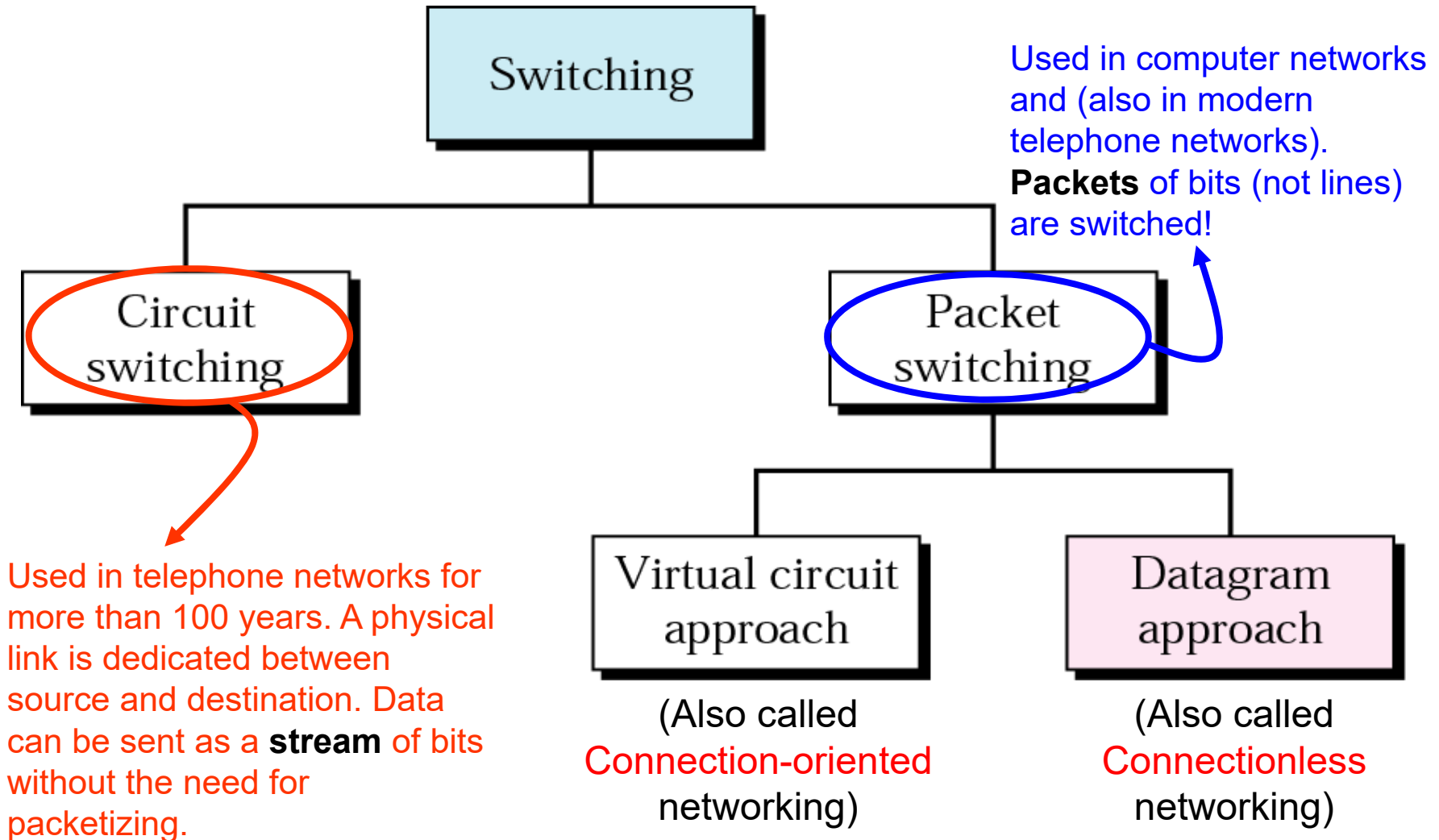


Figure: A collection of interconnected networks



# Switching Technique



# Packet Switching Technique

## ■ 2 packet switching techniques:

Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Our Internet today

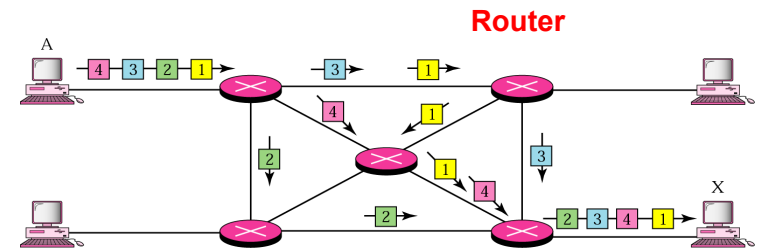


Figure: Datagram

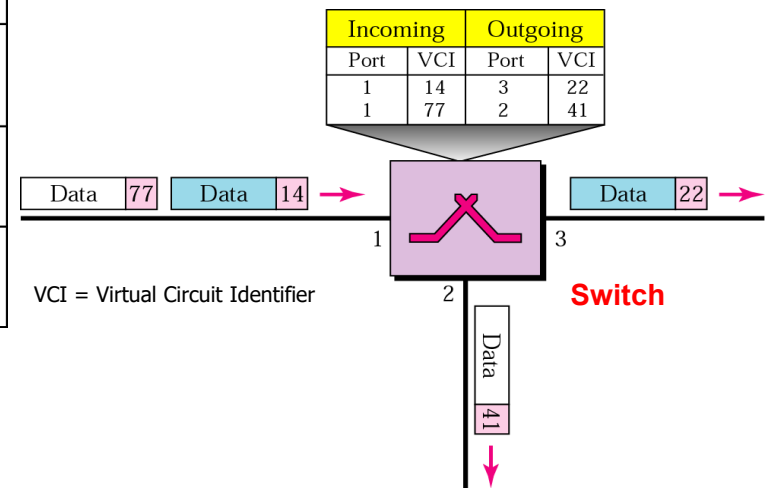
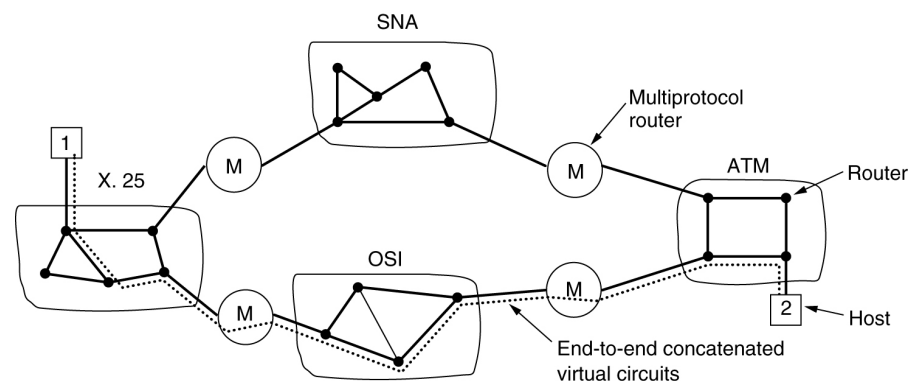
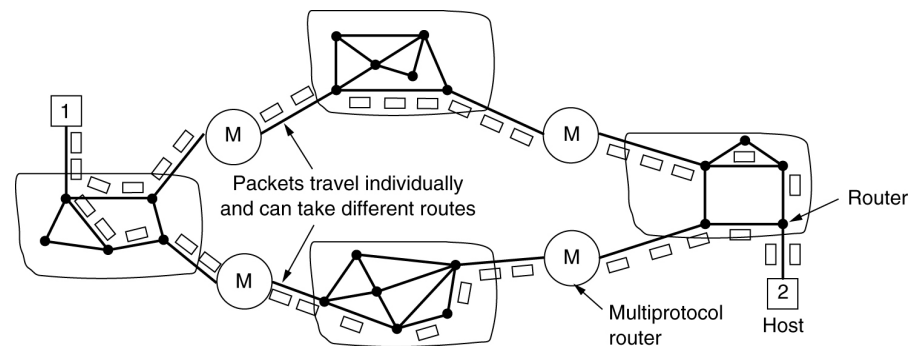


Figure: Virtual-circuit

# Services

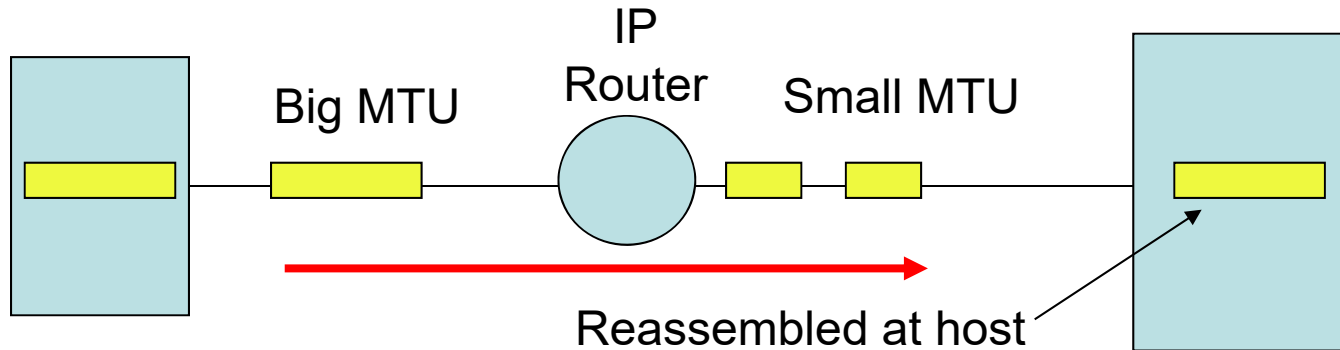
- Transport layer provides 2 services:
  - **Connectionless** service
    - No set up is needed
    - Each packet contains information which allows the packet to be individually routed hop-by-hop through the network
  - **Connection-oriented** service
    - A setup stage is used to determine the end-to-end path before a connection is established
    - Data flow streams are identified by some type of connection indicator (e.g.,: OSI, X.25, SNA)



# Packetizing

- Objective: to encapsulate packets received from upper-layer protocols
- At sender
  - Break segment into packets
  - Assign packet numbers
- At final destination
  - Check if all packets have arrived
  - Reassemble packets into segment

# Fragmentation and Re-assembly



- To handle different **maximum transmission unit** (MTU) size, the network layer must be able to fragment transport layer MTU into smaller MTU units, so that they can be transferred over various data link layer technologies, e.g.,
  - ▣ Ethernet is 1492 bytes
  - ▣ Token Ring is 4500 bytes
- Disadvantages of fragmentation
  - ▣ Smaller MTU block, larger overhead
  - ▣ Smaller MTU block, more interrupts
  - ▣ More MTUs, more time processing
- When to re-assemble
  - ▣ At destination – this results in packets getting smaller as data traverses Internet
  - ▣ At router – need large buffers at routers, buffers may fill with fragments, all fragments must go through same router

# IP Fragmentation

- IP re-assembles at destination only
- Uses fields in header
  - **Data unit Identifier (ID)**
    - Identifies end system originated datagram
    - Source and destination address
    - Protocol layer generating data (e.g., TCP)
    - Identification supplied by that layer
  - **Data length**
  - **Offset (13-bit length)**
    - Position of fragment of user data in original datagram
    - In multiples of 8 octets
  - **More flag**
    - Indicates that this is not the last fragment

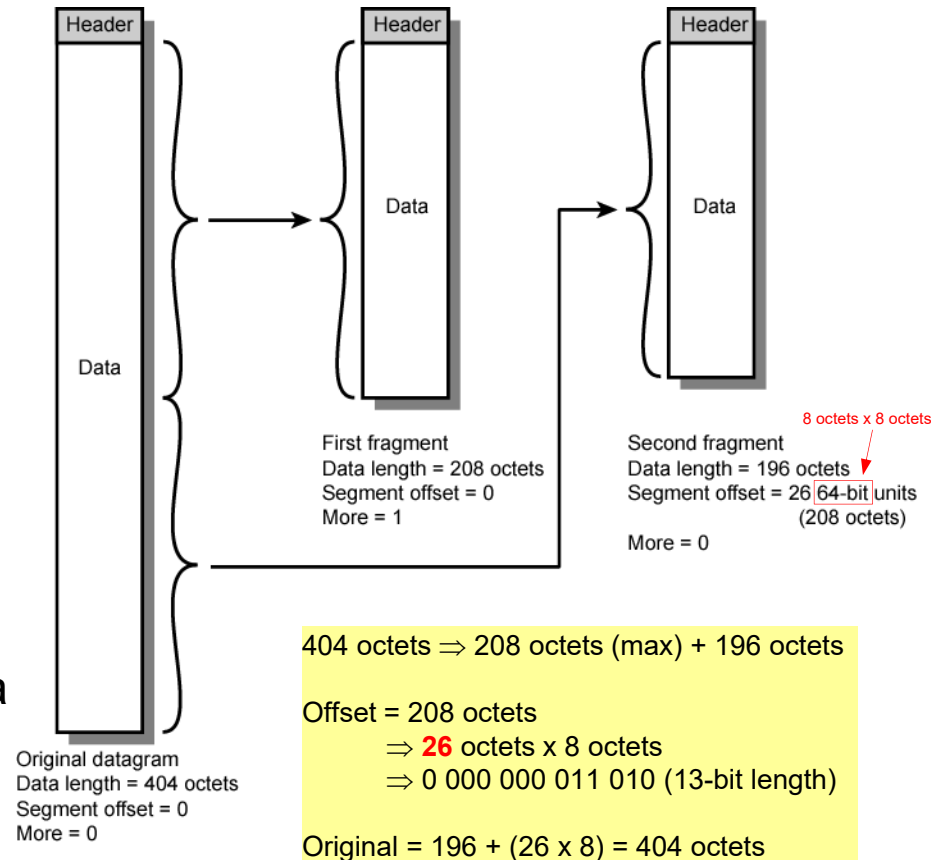


Figure: Fragmentation example

# Internet Names and Addresses

- Network names, generally like:
  - garfield.cs.hiram.edu
  - groups.google.com
- Essentially hierarchical in “domains”
  - “garfield” in “computer science” at “hiram college” in the “education” domain
  - “groups” of the “google” company in the “company” domain
- Translated to IP addresses by Domain Name Servers (**DNS**)
- IPv4 has 4 sets of 8 bits (0-255), e.g.,: 143.206.149.21

# Getting an Address

- Organization gets a block of addresses from an Internet Service Provider (**ISP**)
- ISP gets its addresses from a bigger ISP or **ICANN** (also manages DNS names)
- Individual hosts gets address within organization's block
  - Manual: system administrator gives host a fixed IP (needed for externally available servers)
  - DHCP: protocol to request an available address for a finite time (it also gets first-hop router and DNS information)
    - DHCP addresses can be reused by different subscribers if all subscribers are not online all the times

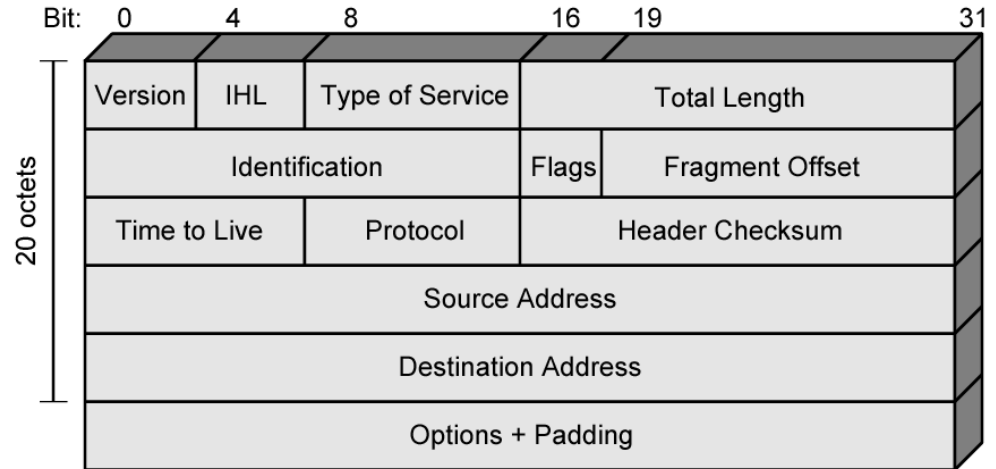


# Addressing

- **Unique** addresses are required to define each host/machine/device/user in the network
- We cannot use the data link layer addresses! Because these addresses depend on the data-link technology
- Internet Address – IP Address
  - Classless addressing – mid 1990s
  - Classful addressing – original architecture, Class A, B, C, D, and E
  - IPv4
    - **32-bit** binary number, total IP address size  $2^{32}$  ( $4.3 \times 10^9$ )
    - **Dotted-Decimal** Notation  
E.g., 128.11.3.31, 255.255.255.0
  - IPv6
    - **128-bit** binary number, total IP address size is  $2^{128}$  ( $3.4 \times 10^{38}$ )
    - **Colon-Hexadecimal** Notation  
E.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334

# IPv4 Header

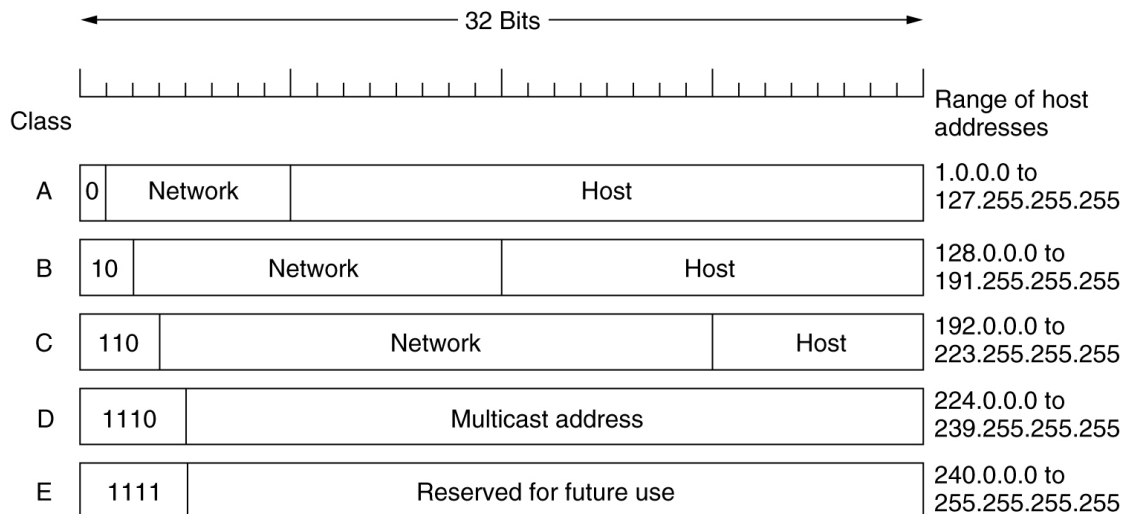
- Version – e.g., 4
- Internet Header Length – In 32-bit words, including options
- Type of Service
- Total Length – datagram length
- Identification – sequence number, which is used with addresses and user protocol to identify datagram uniquely
- Flags – more bit, don't fragment
- Fragmentation Offset
- Time to Live
- Protocol – TCP, UDP, assigned numbers are on [www.iana.org](http://www.iana.org)



- Header checksum
  - Re-verified at each router
  - 16-bit ones complement sum of all 16-bit words in header
  - Set to zero during calculation
- Source address
- Destination address
- Option
- Padding – to fill to multiple of 32 bits long

# Addressing using IPv4

- IP address do not identify hosts in general. They identify a host **on a network**
- If a computer is connected to more than one network, it has more than one IP address (e.g.,: routers, multihomed hosts)
  - Class A – 128 blocks (1st byte), 16,777,216 hosts
  - Class B – 16,384 blocks (1st & 2nd byte), 65536 hosts
  - Class C – 2,097,152 blocks (1st, 2nd, 3rd byte), 256 hosts
  - Class D – 1 block, Multicasting, 268,435,456 hosts
  - Class E – reserved for future use, 268,435,456 hosts



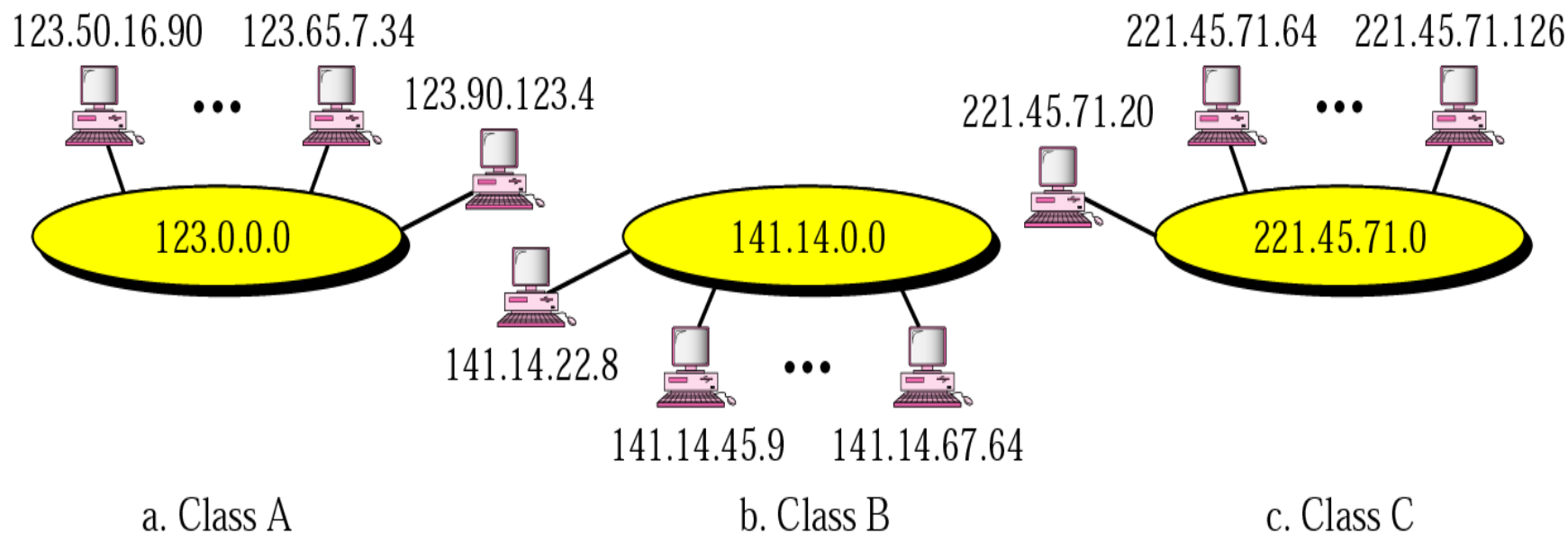
## Special IP Address

0 0	This host
0 0      ...      0 0      Host	A host on this network
1 1	Broadcast on the local network
Network      1 1 1 1      ...      1 1 1 1	Broadcast on a distant network
127      (Anything)	Loopback

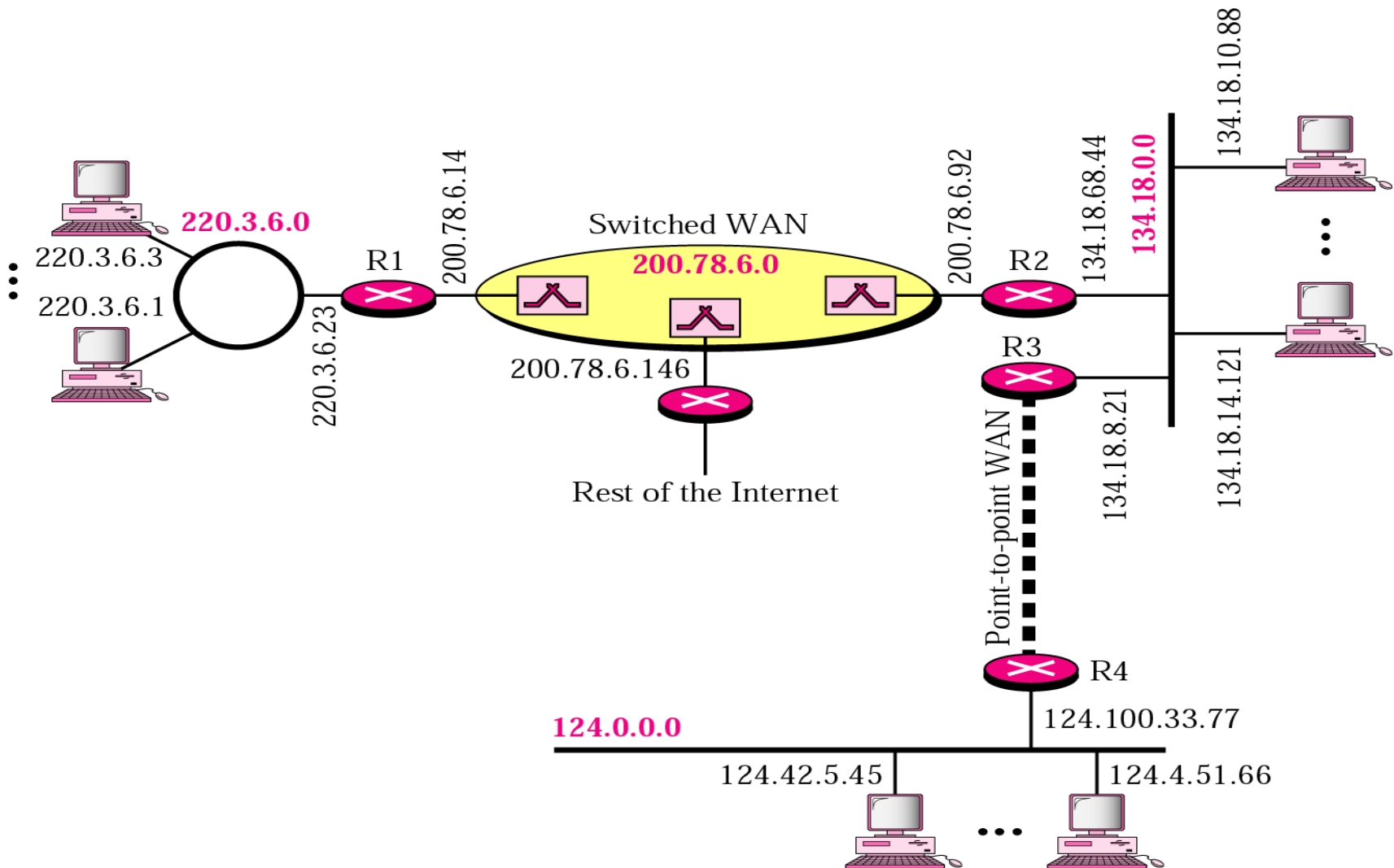
# Network Address

- An address defines a network with all host-id = 0

Netid	Hostid
Specific	All 0s



# Sample Internet

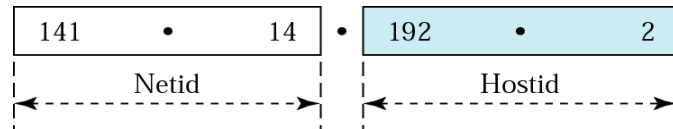


# Subnets and Subnet Masks

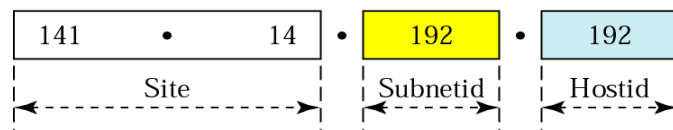
- Allow arbitrary complexity of internetworked LANs within organization
- Insulate overall internet from growth of network numbers and routing complexity
- Site looks to rest of internet like single network
- Each LAN assigned **subnet** number
- Host portion of address partitioned into subnet number and host number
- Local routers route within subnetted network
- **Subnet mask** indicates the split between network + subnet number and host number

# Subnet Addressing

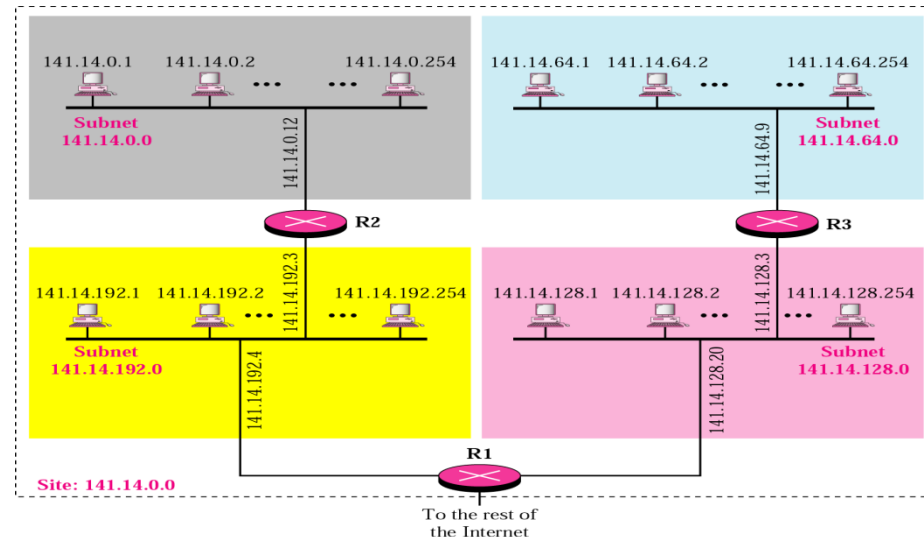
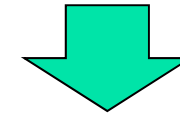
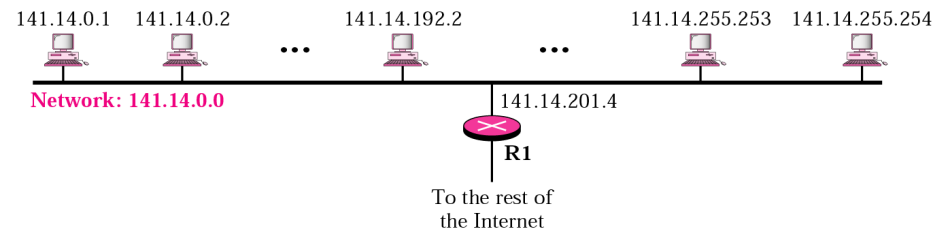
- Class A & B networks too big
  - ▣ Very few LANs have close to 64K hosts
  - ▣ For electrical/LAN limitations, performance or administrative reasons
- Need simple way to get multiple “networks”
  - ▣ Use bridging, multiple IP networks or split up single network address ranges (subnet)



a. Without subnetting



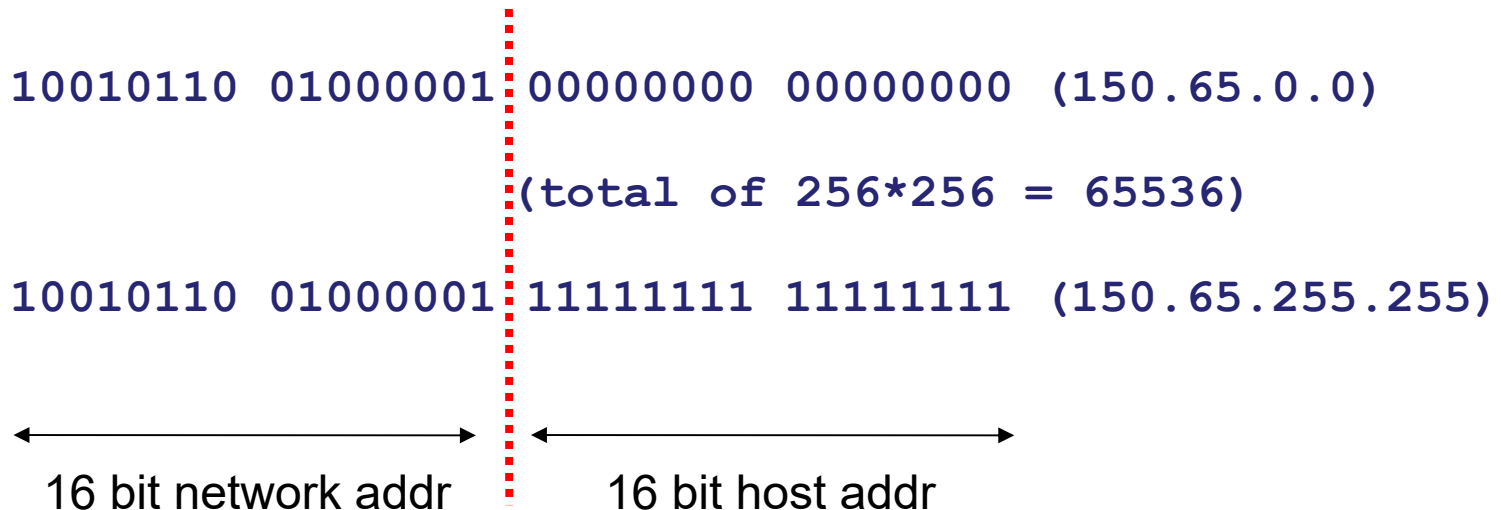
b. With subnetting





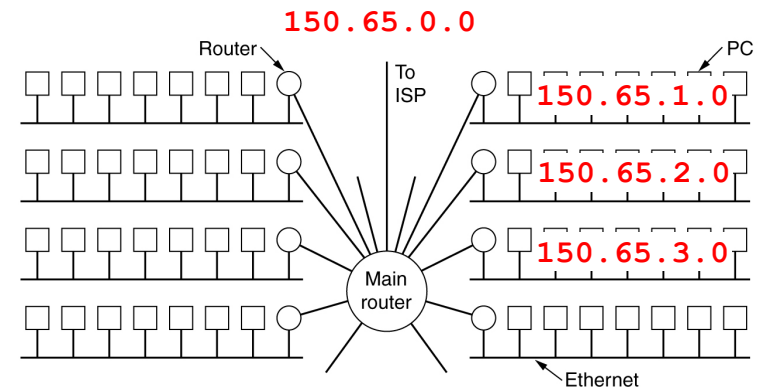
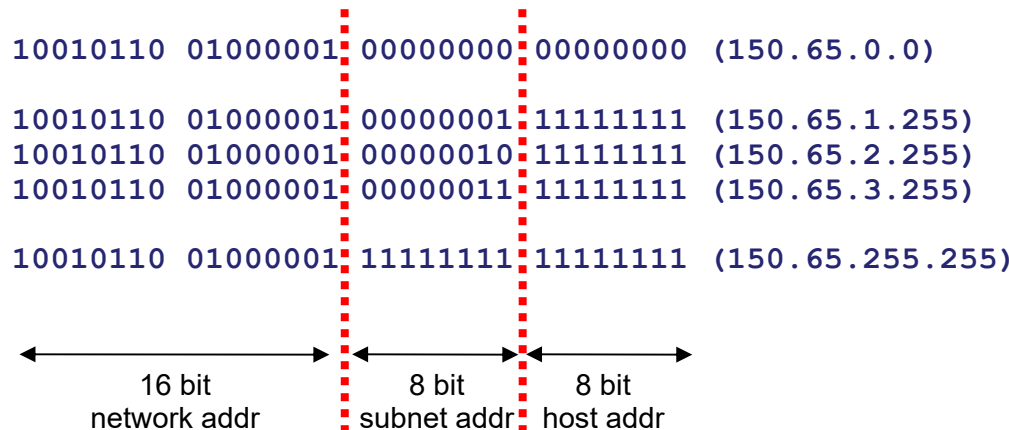
# Example of Network Design

- JAIST has been assigned the 150.65.x.x range of addresses (this is a **class B** address)
  - ▣ We could have had one big network (with up to 65536 hosts) for the whole university attached to a single router
  - ▣ But that would be an administrative nightmare: troubleshooting, traffic locality, and address allocation
  - ▣ So we create smaller **subnets**



# Example of Network Design (cont.)

- Inside JAIST's network, we subnet our Class B allocation
- 150.65.0.0 into 256 subnets, by “stealing” 8 bits from the host bits
- Now one or more subnets can be assigned to each department and each department (subnet) can have a router
- So, a typical JAIST IP address is written as 150.65.170.1/24 where the /24 denotes the **netmask**



# Storing/Exchanging Address

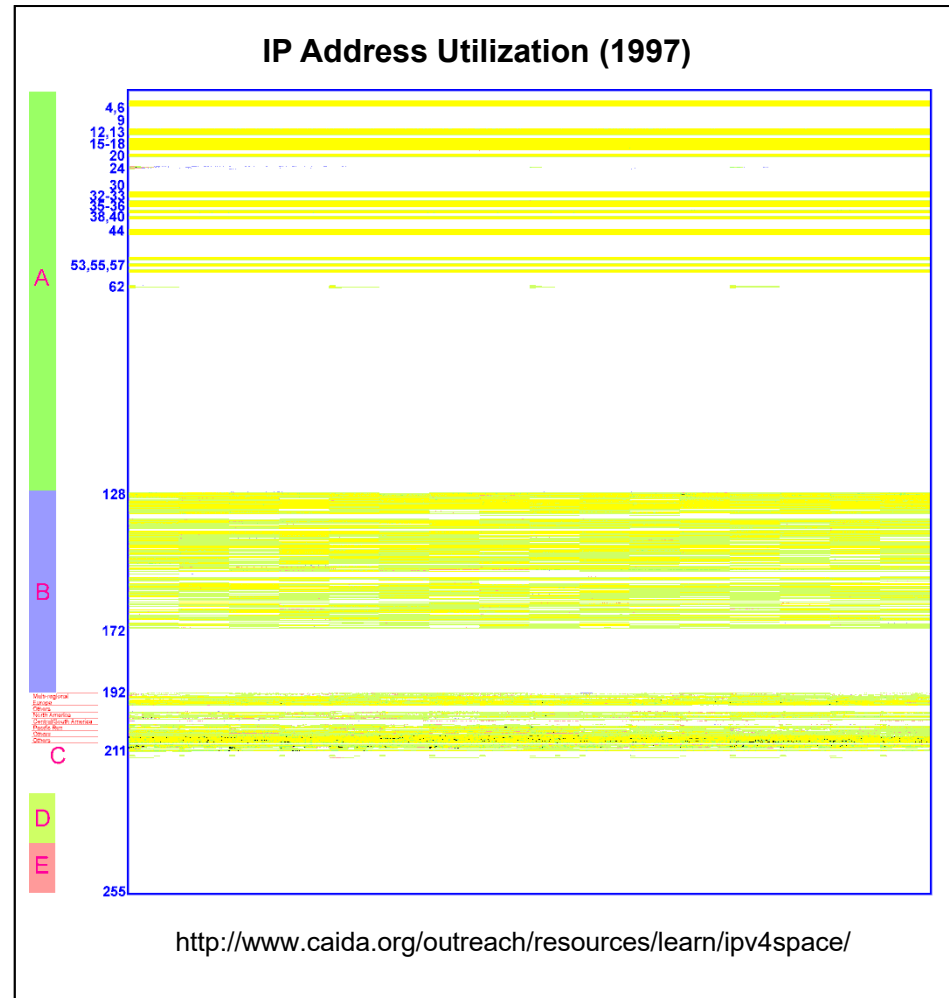
- Traditional IP scheme the netmask is implicit in the address
- Let see the entries that would be stores in a typical router:

<u>Network</u>				<u>Next Hop</u>
208.12.16/24	11010000	00001100	00010000*	x.x.x.x
...				
208.12.21/24	11010000	00001100	00010101*	x.x.x.x
...				
...				
208.12.31/24	11010000	00001100	00011111*	x.x.x.x

- If we use classful addressing we must list 15 entries in the routing table
- **Scaling issue:** A few decades back, Internet was growing dramatically, service providers faced 2 problems
  - ▣ Growth of routing table entries
  - ▣ Depletion of addresses space

# IP Address Problem (1997)


- Address space depletion
  - In danger of running out of classes A and B
  - Why?
    - Class C too small for most domains
    - Very few class A – very careful about giving them out
    - Class B – greatest problem
- Class B sparsely populated
  - But people refuse to give it back
- Large forwarding tables
  - 2 million possible class C groups



# Classless InterDomain Routing (CIDR)

- Since the first 20 bits are identical for all addresses, these entries could be aggregated as

<u>Network</u>		<u>Next Hop</u>
208.12.16/24	11010000 00001100 00010000*	x.x.x.x
...		
208.12.21/24	11010000 00001100 00010101*	x.x.x.x
...		
208.12.31/24	11010000 00001100 00011111*	x.x.x.x

208.12.16/20	11010000 00001100 0001*	x.x.x.x
--------------	-------------------------	---------

- This reduces the no. of entries in the routing table significantly. However, there might be exception that break entries what could have been aggregated. This introduces a set of issues resolved using the **longest-prefix-match algorithms**

## CIDR (cont.)

- CIDR only works well if next hop of all the aggregated entries are the same. Suppose

<u>Network</u>				<u>Next Hop</u>
208.12.16/24	11010000	00001100	00010000*	x.x.x.x
...				
208.12.21/24	11010000	00001100	00010101*	y.y.y.y
208.12.22/24	11010000	00001100	00010110*	x.x.x.x
...				
208.12.31/24	11010000	00001100	00011111*	x.x.x.x

- Now not all hosts with first 20 bits common have the same next hop, so what do we do? We can either not aggregating or create exceptions such as

208.12.16/20	11010000	00001100	0001*	x.x.x.x
208.12.21/24	11010000	00001100	00010101*	y.y.y.y

- But, 208.12.21.5 will match both the first and second entry, so which one do we choose? Now we use the longest-prefix-match and use the second entry

## CIDR (cont.)

- Set of IP address assignments
- Dropping the classes makes forwarding more complicated
- The routing table is scanned sequentially
- The entries can be aggregated, e.g., the 3 entries to 194.24.0.0/19
- If multiple entries with different subnet mask lengths match, the longest mask is used

University	First address	Last address	How many	Written as
Cambridge	194.24.0.0	194.24.7.255	2048	194.24.0.0/21
Edinburgh	194.24.8.0	194.24.11.255	1024	194.24.8.0/22
(Available)	194.24.12.0	194.24.15.255	1024	194.24.12/22
Oxford	194.24.16.0	194.24.31.255	4096	194.24.16.0/20

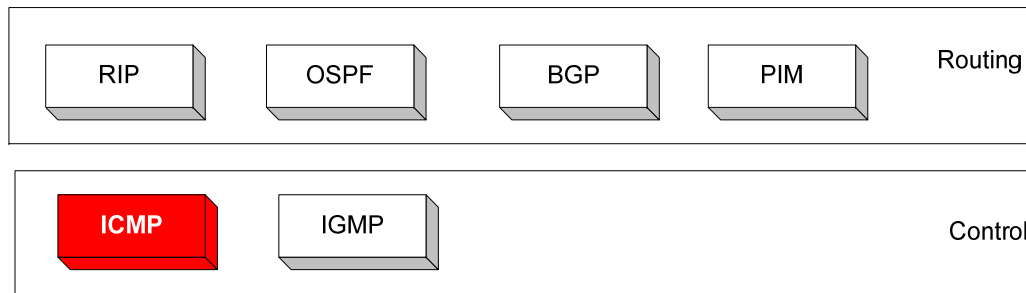
# Routing

- Where do you send the packets?
- Objective: is to calculate 'good' routes
- End systems and routers maintain routing tables
  - Indicate next router to which datagram should be sent
  - Static – may contain alternative routes
  - Dynamic – flexible response to congestion and errors
- Source routing
  - Source specifies route as sequential list of routers to be followed
  - Security
  - Priority
- Route recording
- Routing algorithms are discussed on Chapter 8

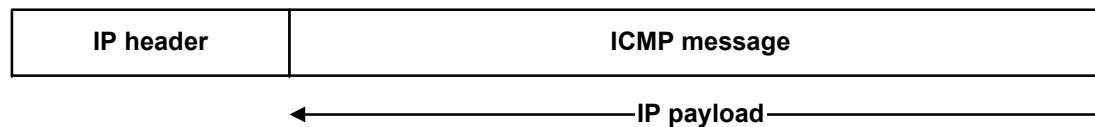


# Internet Control Message Protocol (ICMP)

- IP (Internet Protocol) relies on several other protocols to perform necessary control and routing functions
  - ▣ Control functions (ICMP)
  - ▣ Multicast signaling (IGMP)
  - ▣ Setting up routing tables (RIP, OSPF, BGP)

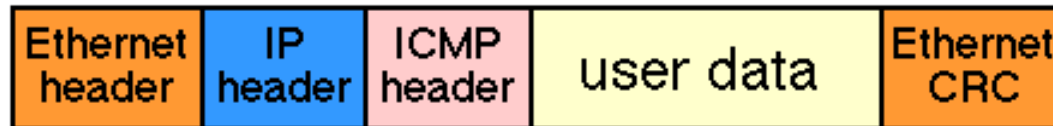


- ICMP is a helper protocol that supports IP with facility for
  - ▣ Error reporting
  - ▣ Simple queries
- ICMP messages are encapsulated as IP datagrams

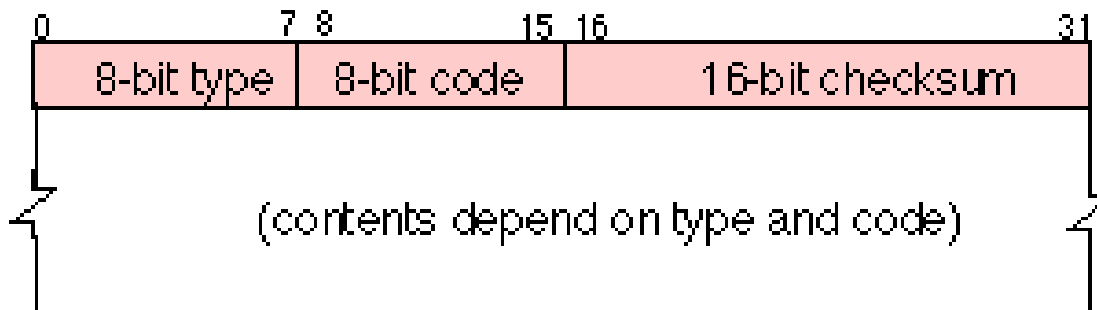


# ICMP (cont.)

- Purpose: to provide **feedback** about problems in the IP network environment
- Delivered in IP packets



- ICMP message format
  - ▣ 4 bytes of ICMP header and optional message

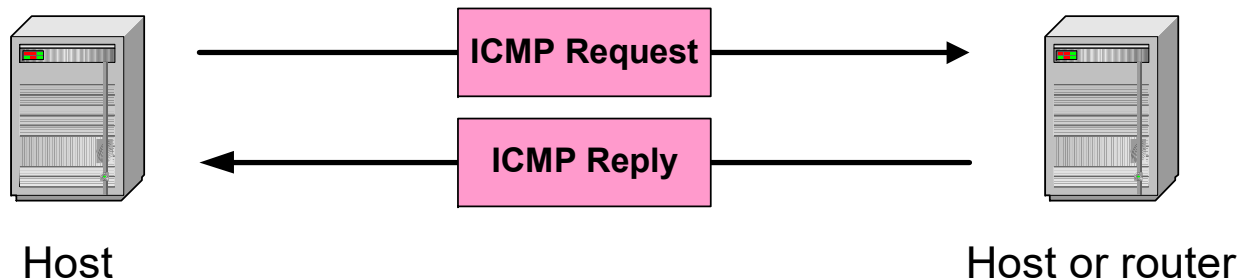


# ICMP (cont.)

- ICMP is used to exchange messages between **routers**
- To announce network errors
  - If a network, host, port is unreachable, ICMP Destination Unreachable Message is sent to the source host
- To announce network congestion
  - When a router runs out of buffer queue space, ICMP Source Quench Message is sent to the source host
- To assist troubleshooting
  - ICMP Echo Message is sent to a host to test if it is alive – used by **ping**
- To announce timeouts
  - If a packet's TTL field drops to zero, ICMP Time Exceeded Message is sent to the source host – used by **traceroute**

# ICMP Query Message

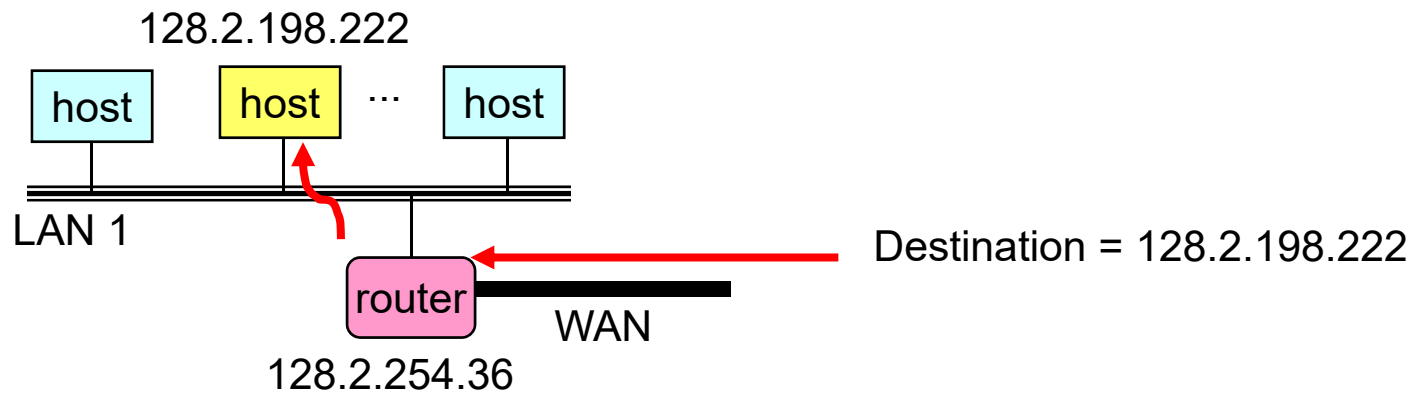
- ICMP query
  - Request sent by host to a router or host
  - Reply sent back to querying host



Type/Code	Description
8/0	Echo Request
0/0	Echo Reply
13/0	Timestamp Request
14/0	Timestamp Reply
10/0	Router Solicitation
9/0	Router Advertisement

} The **ping** command uses Echo Request/Echo Reply

# Finding a Local Machine



- Routing gets packet to correct local network
  - Based on IP address
  - Router sees that destination address is of local machine
- Still need to get packet to host
  - Using link-layer protocol
  - Need to know hardware address
- Same issue for any local communication
  - Find local machine, given its IP address

# Address Resolution Protocol (ARP)

- RFC826, protocol to know the MAC address from the IP address of a host
- Using broadcast to get a response from a requested host
- Once the information obtained, it creates ARP table
  - Erase the entry when time is elapsed, and refresh the flow with the same questions
  - “arp” command is used in Windows/UNIX

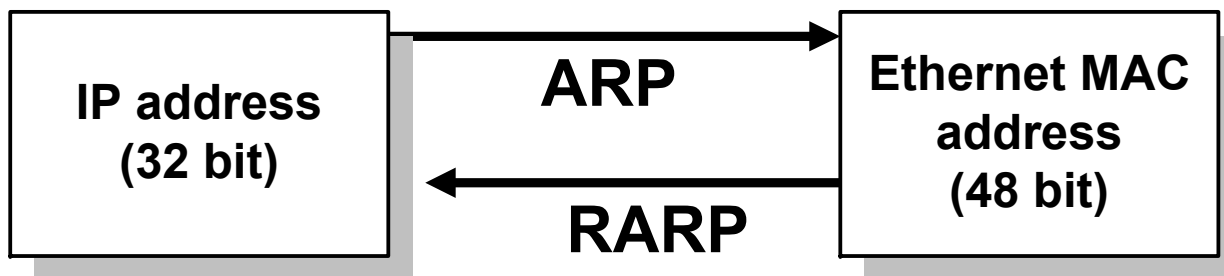
```
[ lss2-is13 ] % /usr/sbin/arp -a
```

Net to Media Table: IPv4

Device	IP Address	Mask	Flags	Phys Addr
fjgi0	ap-is1-is13e1	255.255.255.255		00:01:30:2c:94:00
fjgi0	lss-is13e1	255.255.255.255		00:03:ba:27:a5:98
fjgi0	lss2-is13	255.255.255.255	SP	00:e0:00:c5:0f:70
fjgi0	ss49-is13e1	255.255.255.255		00:30:1e:69:04:80
hme0	192.168.0.1	255.255.255.255	SP	00:e0:00:c4:8f:70
hme0	BASE-ADDRESS. MCAST. NET	240.0.0.0	SM	01:00:5e:00:00:00
fjgi0	BASE-ADDRESS. MCAST. NET	240.0.0.0	SM	01:00:5e:00:00:00

# Reverse ARP

- RFC903, protocol to do the reverse operation of ARP
- A protocol used by a host to request its IP address from an administrative host, when it has available its MAC address
- Requires one or more server hosts to maintain a database of mappings of MAC addresses to their respective IP addresses
- Now, this role is done by DHCP and BOOTP



# Address Resolution Protocol (ARP)

- Diagrammed for Ethernet (6-byte MAC addresses)

- Operation

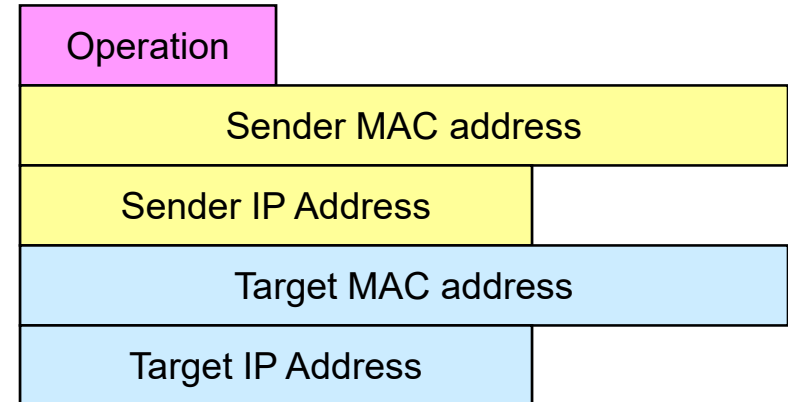
- 1: request
    - 2: reply

- Sender

- Host sending ARP message

- Target

- Intended receiver of message



- Low-level Protocol

- Operates only within local network

- Determines mapping from IP address to hardware (MAC) address

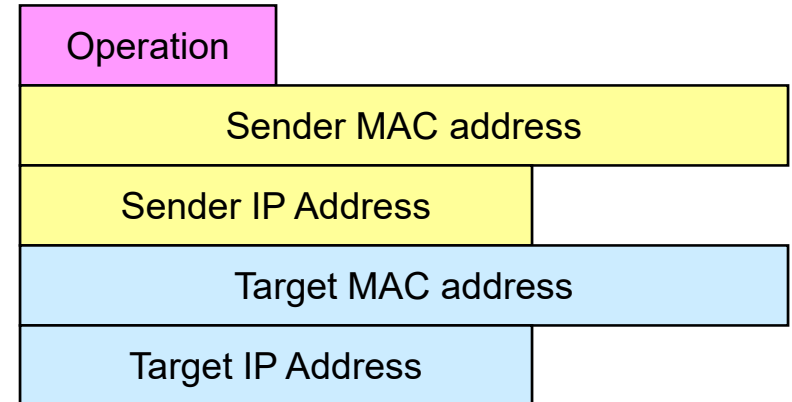
- Mapping determined dynamically

- No need to statically configure tables
    - Only requirement is that each host know its own IP address



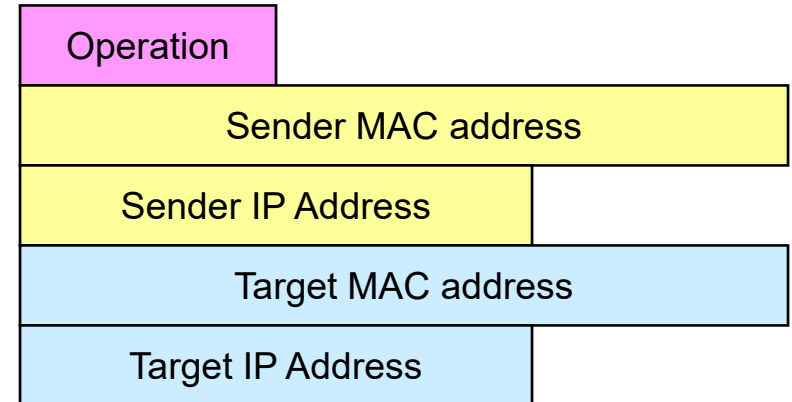
# ARP Request

- ARP request
  - Operation
    - 1: request
  - Sender
    - Host that wants to determine MAC address of another machine
  - Target
    - Other machine
- Requestor
  - Fills in own IP and MAC address as “sender”
- Mapping
  - Fills desired host IP address in target IP address
- Requestor
  - Send to MAC address `ff:ff:ff:ff:ff:ff`, Ethernet broadcast

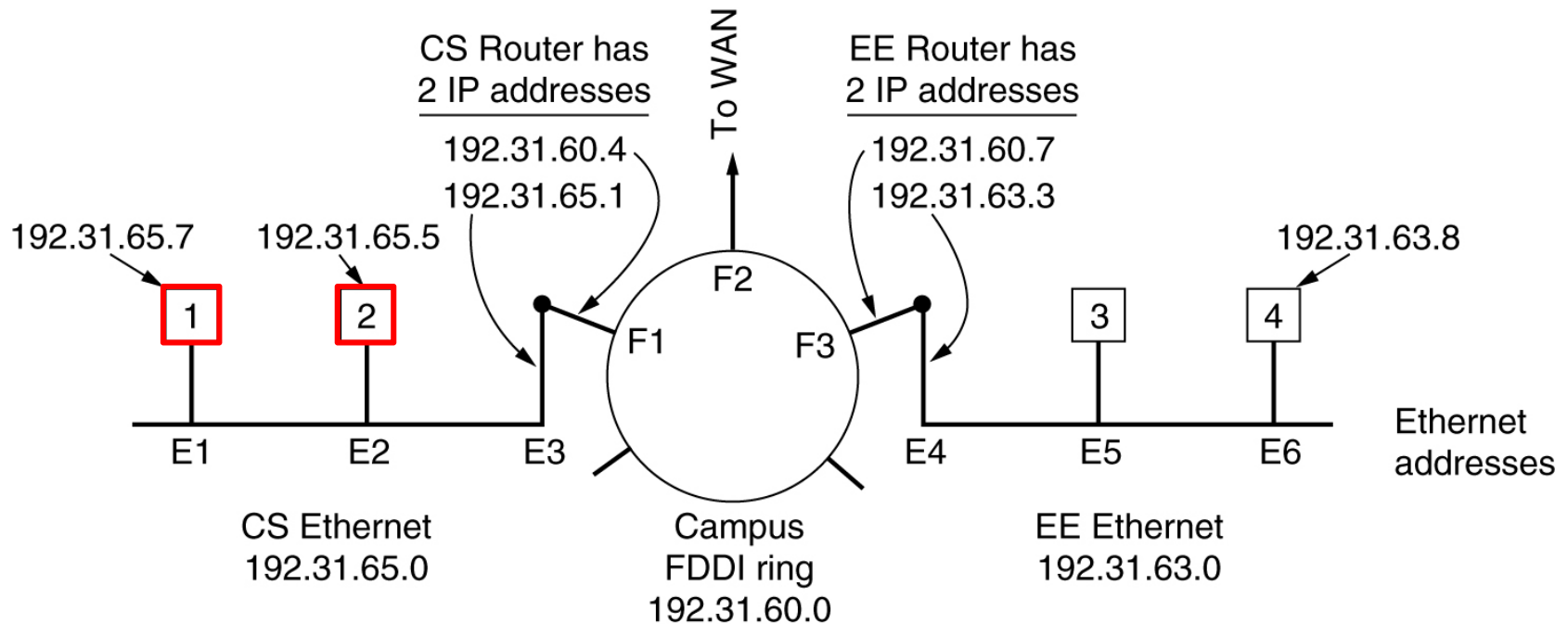


# ARP Reply

- ARP reply
  - Operation
    - 2: reply
  - Sender
    - Host with desired IP address
  - Target
    - Original requestor
  
- Responder becomes “sender”
  - Fill in own IP and MAC address
  - Set requestor as target
  - Send to requestor’s MAC address



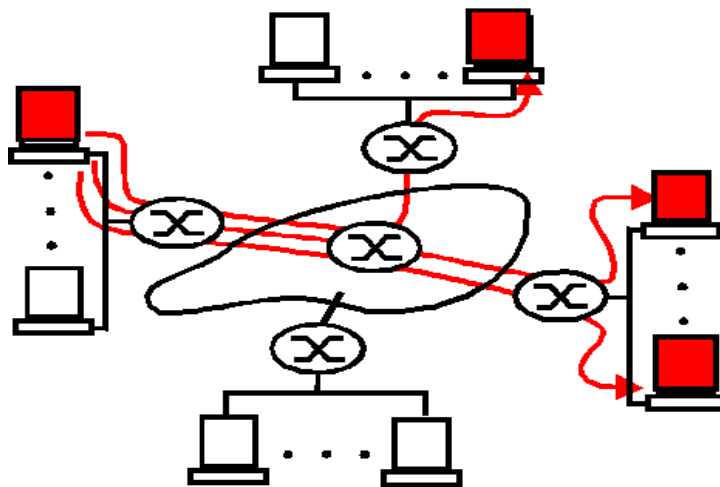
# ARP Example



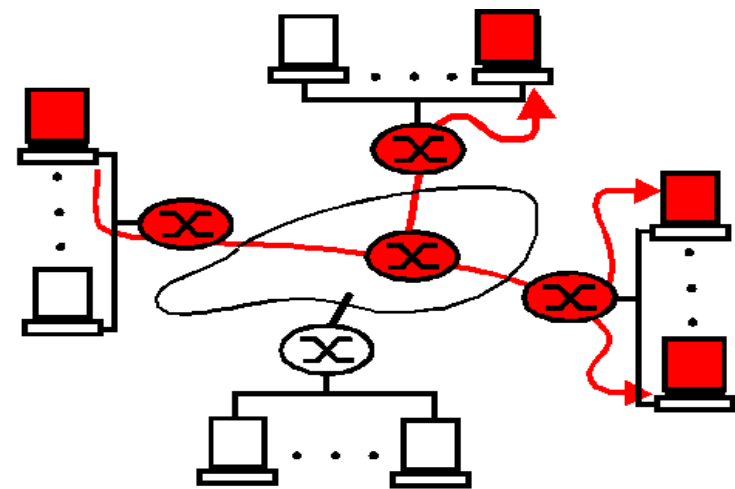
- 3 interconnected /24 networks: 2 Ethernets and 1 FDDI ring
- Interface between data link layer and network layer
- Mapping between IP addresses and MAC Ethernet addresses
- Host 1 want to send a packet to host 2. It broadcasts on his LAN: Who has IP addr. 192.31.65.5? Host 2 will respond with his MAC address E2

# Multicast Routing

- **Multicast**: delivery of a packet to a group of receivers
- Multicasting is becoming increasingly popular in the Internet (video on demand; whiteboard; interactive games)
- Multiple unicast vs. multicast



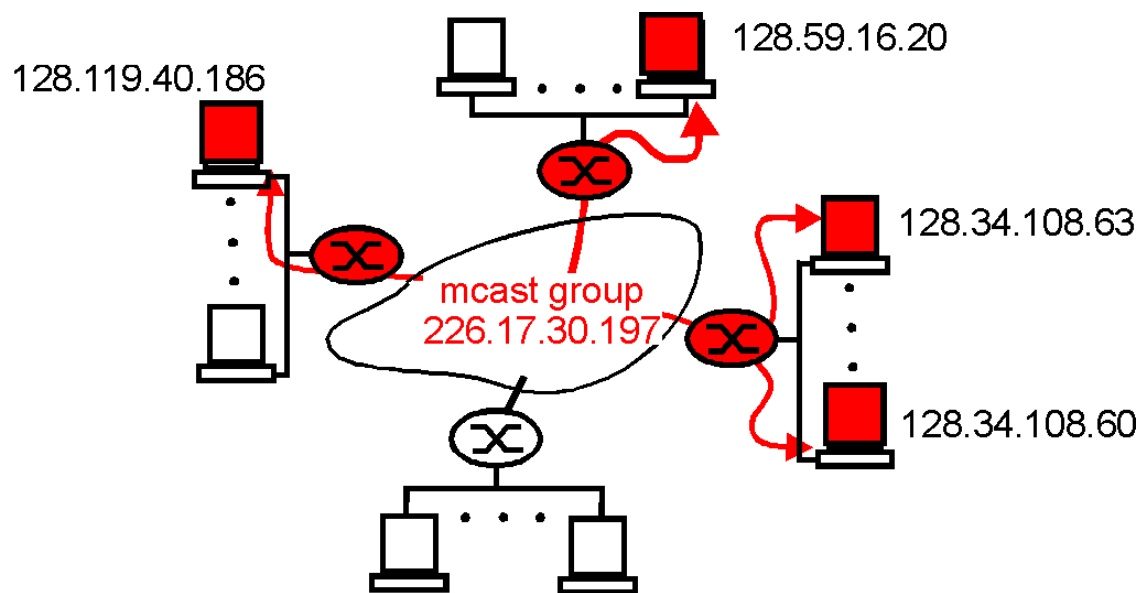
multicast via unicast



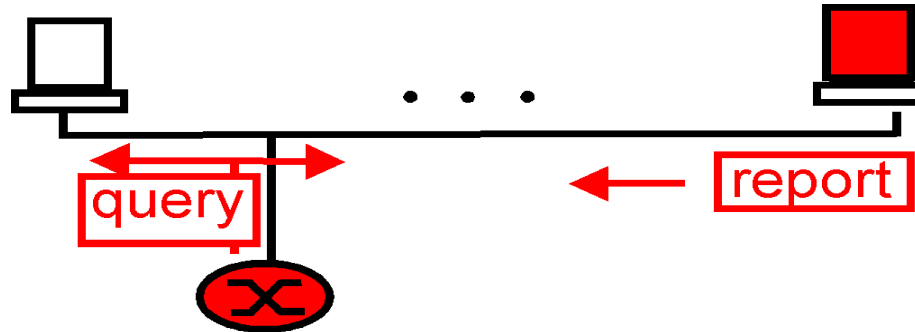
network multicast

# Multicast Group Address

- Multicast group address “delivered” to all receivers in the group
- Internet uses **Class D** for multicast
- Multicast address distribution is managed by IGMP protocol



# IGMP



- **Internet Group Management Protocol** (IGMP) operates between router and local hosts, typically on a LAN
- Router queries the local hosts for multicast group membership information
- Router “connects” active hosts to multicast tree via multicast protocol
- Hosts respond with membership reports: actually, the first host which responds (at random) speaks for all
- Host issues “leave-group” message to leave; this is optional since router periodically polls anyway (soft state concept)

# **Announcement**

- Next is Chapter 5 Network Layer II
- 10:50 ~ 12:30 on 26 October (Wednesday)