1902EC504 - COMPUTER NETWORKS

Prof S.Praveen Kumar M.E,(Ph.D), EMCAA, MISTE, IAENG.,
Assistant Professor/CSE
E.G.S Pillay Engineering College, Nagapattinam

Course Outcomes

- At the end of this course students can able to

UNIT 3 - NETWORK LAYER

- **Network Layer**
  - Functionality of network layer
  - Network addressing
- **Network routing**
  - Routing algorithms
- **Internetworking**
  - Network layer protocols
  - Switching concepts – Circuit switching and Packet switching
  - Quality of service
  - Network layer design issues.

FUNCTIONALITIES OF NETWORK LAYER

- The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links).
- The network layer ensures that each packet gets from its point of origin to its final destination.
- If two systems are connected to the same link, there is usually no need for a network layer.
- However, if the two systems are attached to different networks (links) with connecting devices between the networks (links), there is often a need for the network layer to accomplish source-to-destination delivery.

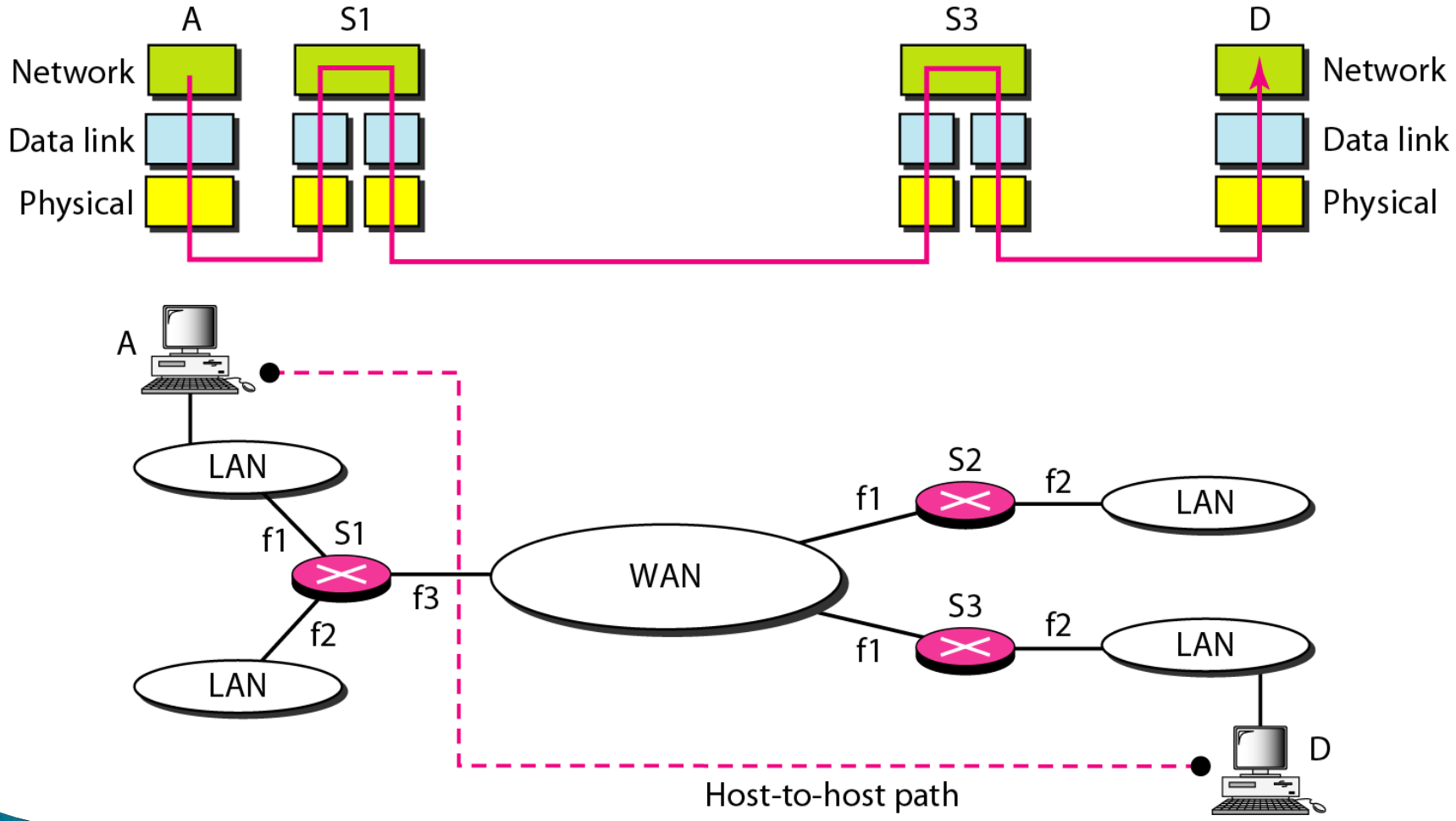
- **Functions of Network Layer:**
- **1.Logical addressing:** The physical addressing implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems.
- The network layer adds a header to the packet coming from the upper layer that, includes the logical addresses of the sender and receiver.
- **2.Routing:** When independent networks or links are connected to create *internetworks*(network of networks) or a large network, the connecting devices (called *routers* or *switches) route or switch the packets to their final destination.*

INTERNETWORKING

- The LANs and switched data sub networks are interconnected to allow the users to communicate across several sub networks.
- This extended network so formed is called as Internetwork.
- The network layer at the source is responsible for creating a packet from the data coming from another protocol (such as a transport layer protocol).
- The header of the packet contains, the logical addresses of the source and destination.
- The network layer is responsible for checking its routing table to find the routing information such as the outgoing interface of the packet or the physical address of the next node.
- If the packet is too large, the packet is fragmented.

- The network layer at the switch or router is responsible for forwarding or routing the packet.
- When a packet arrives, the router consults its routing table and finds the interface from which the packet must be sent.
- The packet, after some changes in the header, with the routing information is passed to the data link layer again.
- The network layer at the destination is responsible for address verification.
- It makes sure that the destination address on the packet is the same as the address of the host.
- If the packet is a fragment, the network layer waits until all fragments have arrived, and then reassembles them and delivers the reassembled packet to the transport layer.

- The network layer is responsible for host-to-host delivery and for routing the packets through the routers or switches.



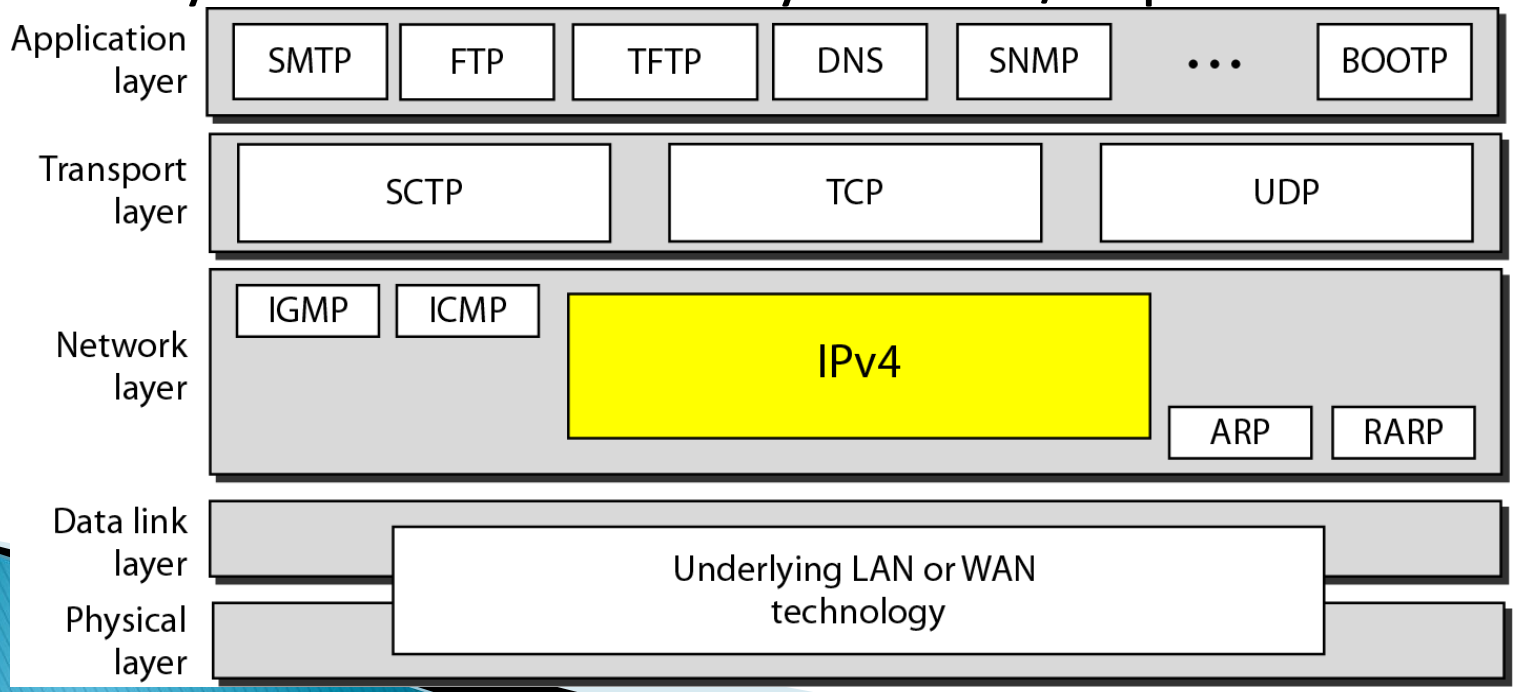
Network layer in an internetwork

- **Internet as a Datagram Network:** The Internet, at the network layer, is a packet-switched network.
- Packet switching uses either the virtual circuit approach or the datagram approach.
- Switching at the network layer in the Internet uses the datagram approach to packet switching.
- It uses the universal addresses defined in the network layer to route packets from the source to the destination.

- **Internet as a Connectionless Network:** Communication at the network layer in the Internet is connectionless.
- In connectionless service, the network layer protocol treats each packet independently, with each packet having no relationship to any other packet.
- The packets in a message may or may not travel the same path to their destination.
- This type of service is used in the datagram approach to packet switching.
- The reason for this decision is that the Internet is made of so many heterogeneous networks that it is almost impossible to create a connection from the source to the destination without knowing the nature of the networks in advance.

INTERNET PROTOCOL(IP)

- IP is a key tool used to build scalable, heterogeneous internetworks.
- IP corresponds to network layer in OSI reference model and provides a connectionless best effort delivery service to the transport layer.
- **IPv4:** The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols

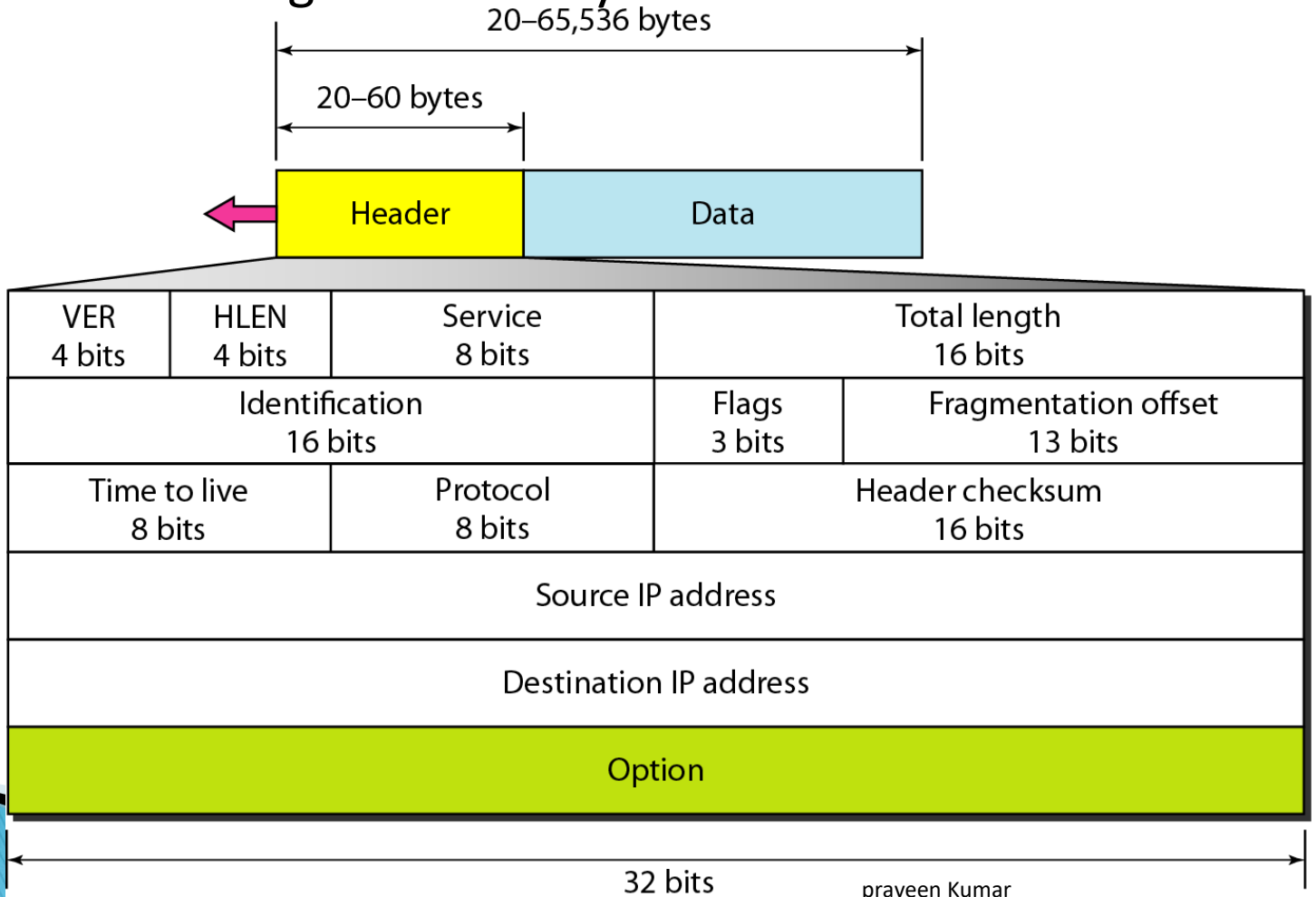


- IPv4 is an unreliable and connectionless datagram protocol-a best-effort delivery service.
- The term *best-effort means that IPv4 provides no error control or flow control* except for error detection on the header.
- IPv4 assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees.
- If reliability is important, IPv4 must be paired with a reliable protocol such as TCP.

- IPv4 is also a connectionless protocol for a packet-switching network that uses the datagram approach.
- This means that each datagram is handled independently, and each datagram can follow a different route to the destination.
- This implies that datagrams sent by the same source to the same destination could arrive out of order.
- Also, some could be lost or corrupted during transmission.
- To handle these problems, IPv4 relies on a higher-level protocol(TCP) to take care of all these problems.

- **Datagram:** Packets in the IPv4 layer are called datagrams
- A datagram is a variable-length packet consisting of two parts: header and data.
- The header is 20 to 60 bytes in length and contains information essential to routing and delivery.

IPv4 datagram format



- **Version (VER):** This 4-bit field defines the version of the IPv4 protocol.
- **Header length (HLEN):** This 4-bit field defines the total length of the datagram header in 4-byte words.
- This field is needed because the length of the header is variable between 20 and 60 bytes.
- **Total length:** This is a I6-bit field that defines the total length (header plus data) of the IPv4 datagram in bytes.
- To find the length of the data coming from the upper layer, subtract the header length from the total length.
- The header length can be found by multiplying the value in the HLEN field by 4.
- Length of data =total length - header length

- **Services:** This field, is used to specify differentiated services.

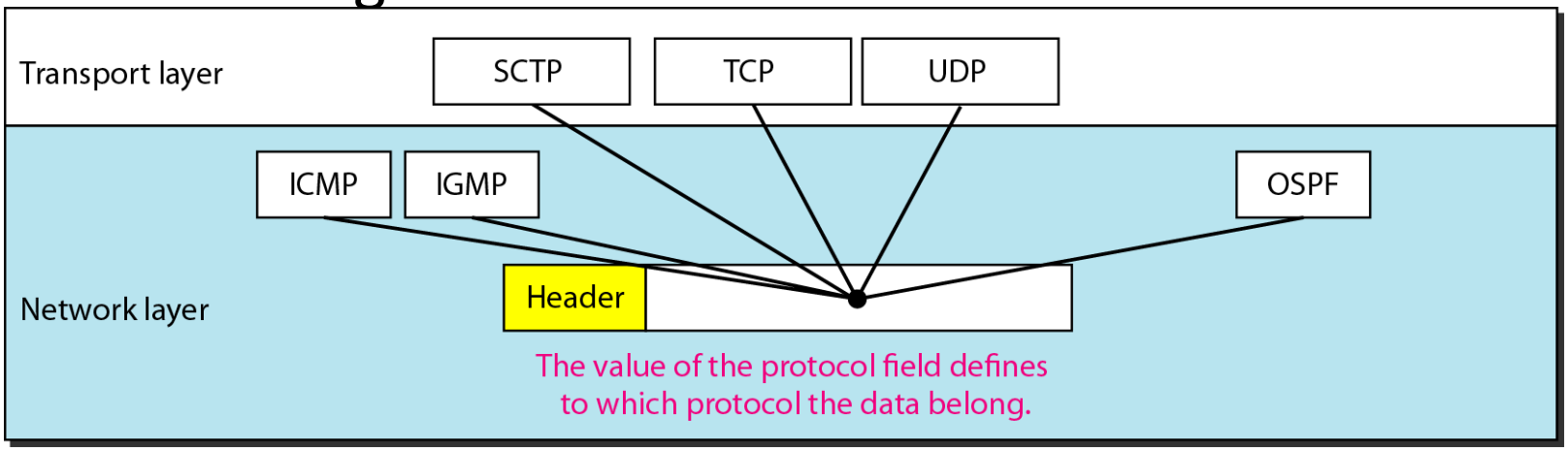
Default types of service

Protocol	*TOS Bits*	*Description*
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

- **Identification:** This field is used in fragmentation.
- **Flags:** This field is used in fragmentation.
- **Fragmentation offset:** This field is used in fragmentation
- **Time to live:** A datagram has a limited lifetime in its travel through an internet.
- This field was originally designed to hold a timestamp, which was decremented by each visited router.
- The datagram was discarded when the value became zero.
- However, all the machines must have synchronized clocks and must know how long it takes for a datagram to go from one machine to another.

Protocol field and encapsulated data

- **Protocol:** This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer.
- An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP.
- This field specifies the final destination protocol to which the IPv4 datagram is delivered.

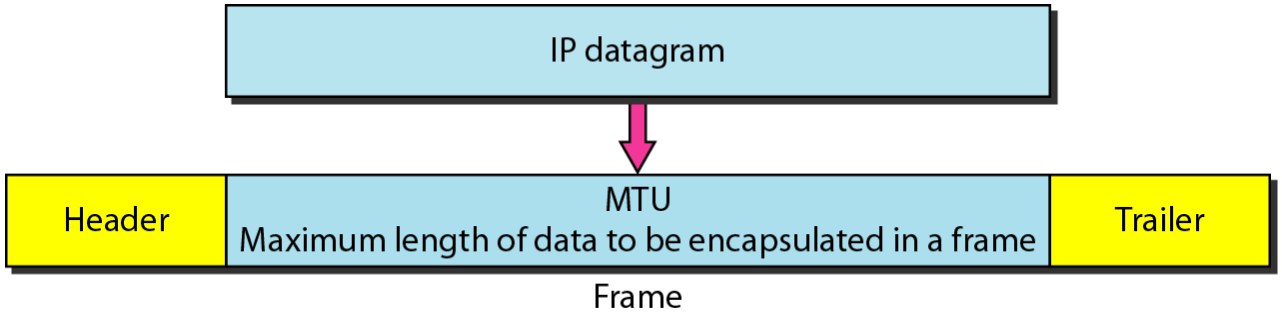


Protocol values

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

- **Source address:** This 32-bit field defines the IPv4 address of the source.
- This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.
- **Destination address:** This 32-bit field defines the IPv4 address of the destination.
- This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

- **Fragmentation:** we must divide the datagram to make it possible to pass through physical networks. This is called fragmentation.
- ***Maximum Transfer Unit (MTU):***
- ***MTU*** is the largest size of packet that can be sent in a packet switched network.
- TCP uses the MTU to determine the maximum size of each packet in any transmission.
- The value of the MTU depends on the physical network protocol.



Maximum transfer unit (MTU)

Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

MTUs for some networks

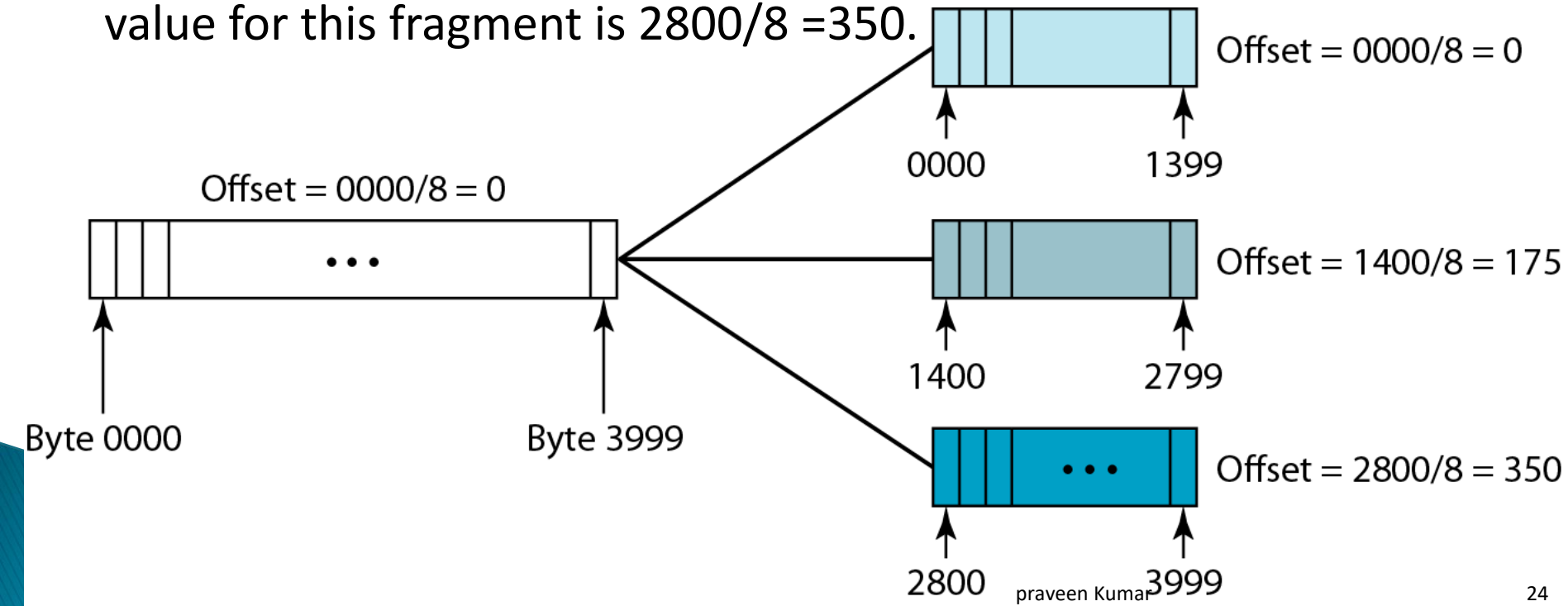
- The source usually does not fragment the IPv4 packet.
- The transport layer will segment the data into a size that can be accommodated by IPv4 and the data link layer.
- The reassembly of the datagram is done only by the destination host because each fragment becomes an independent datagram.
- ***Fields Related to Fragmentation:***
- **Identification:** This 16-bit field identifies a datagram originating from the source host.
- The identification number helps the destination in reassembling the datagram.

- **Flags:** This is a 3-bit field. The first bit is reserved.
- The second bit is called the *donotfragment bit(D).*
- *If its value is 1, the machine must not fragment the datagram.*
- If its value is 0, the datagram can be fragmented if necessary.
- The third bit is called the *more fragment bit(M).*
- *If its value is 1, it means the datagram is not the last fragment, and* there are more fragments after this one.
- If its value is 0, it means this is the last or only fragment.

Flags used in fragmentation

- **Fragmentation offset:** This 13-bit field shows the relative position of this fragment with respect to the whole datagram.
- **Steps followed in Fragmentation:**
- 1. The first fragment has an offset field value of zero.
- 2. Divide the length of the first fragment by 8.
- The second fragment has an offset value equal to that result.
- 3. Divide the total length of the first and second fragments by 8.
- The third fragment has an offset value equal to that result.
- 4. Continue the process.
- The last fragment has a *more bit value of O.*

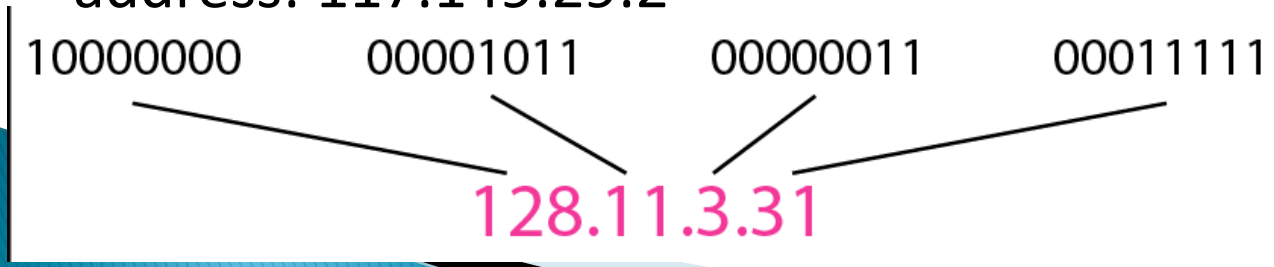
- Consider a datagram with a data size of 4000 bytes fragmented into three fragments.
- The bytes in the original datagram are numbered 0 to 3999.
- The first fragment carries bytes 0 to 1399. The offset for this datagram is 0/8 =O.
- The second fragment carries bytes 1400 to 2799. The offset value for this fragment is 1400/8 = 175.
- Finally, the third fragment carries bytes 2800 to 3999. The offset value for this fragment is 2800/8 =350.



IPv4 ADDRESSES

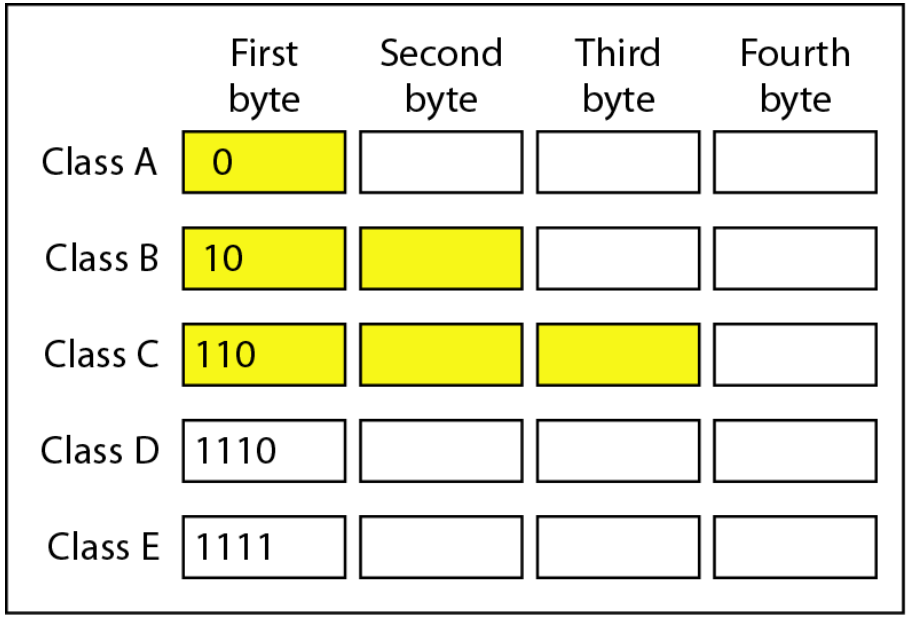
- An **IPv4 address is a 32-bit address that *uniquely and universally defines the connection*** of a device (for example, a computer or a router) to the Internet.
- They are unique address defines one, and only one, connection to the Internet.
- Two devices on the Internet can never have the same address at the same time.
- **Address Space:** IPv4 uses 32-bit addresses, which means that the address space is 4,294,967,296 (more than 4 billion devices).

- **Notations:**
- There are two notations to show an IPv4 address:
- Binary notation and dotted decimal notation.
- ***Binary Notation:*** In binary notation, the IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte.
- The following is an example of an IPv4 address in binary notation: 01110101 10010101 00011101 00000010
- ***Dotted-Decimal Notation:*** To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes.
- The following is the dotted decimal notation of the above address: 117.149.29.2

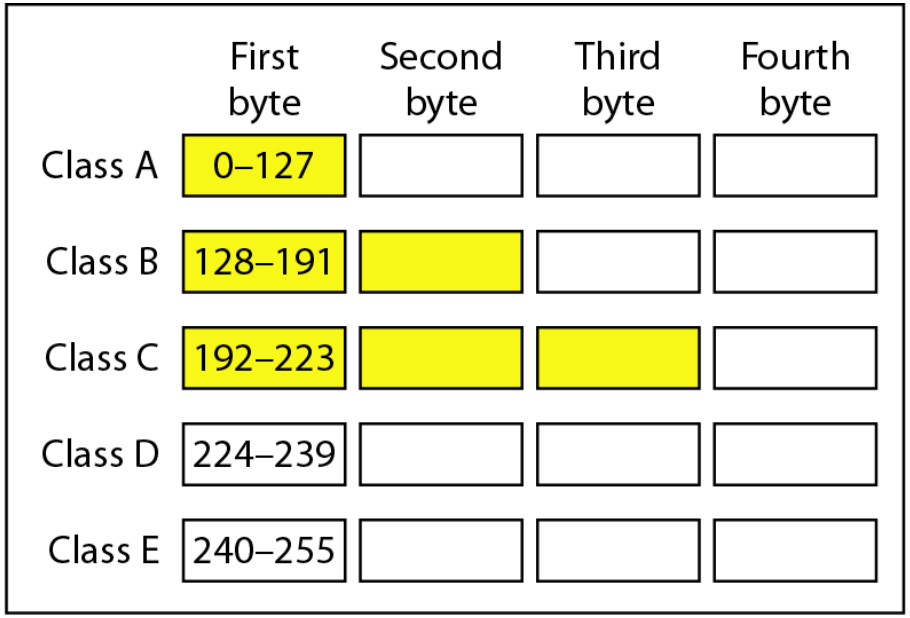


Dotted-decimal notation and binary notation for an IPv4 address

- **Classful Addressing:** In classful addressing, the address space is divided into five classes: A, B, C, D, and E.
- If the address is given in binary notation, the first few bits are used to find the class of the address.
- If the address is given in dotted-decimal notation, the first byte defines the class.



a. Binary notation



b. Dotted-decimal notation

Finding the classes in binary and dotted-decimal notation

- ***Example:***
- 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 O. 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 (between 0 and 127). The class is A.
- d. The first byte is 252 (between 240 and 255).
- The class is E.

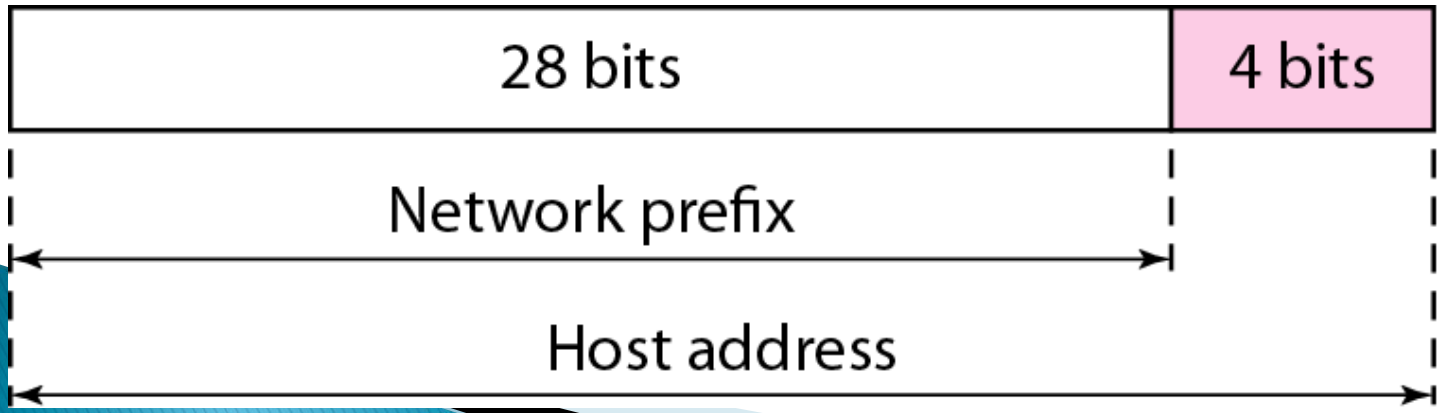
- ***Classes and Blocks:*** Class A addresses were designed for large organizations with a large number of attached hosts or routers.
- Class B addresses were designed for midsize organizations with tens of thousands of attached hosts or routers.
- Class C addresses were designed for small organizations with a small number of attached hosts or routers.

Class	*Number of Blocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

Number of blocks and block size in classful IPv4 addressing

- **Subnetting:** Subnetting is a process of dividing large network into the smaller networks based on IP address.
- **Subnet mask:** It is used to identify network address of an IP address.
- It is a 32 bit number that masks an IP address and divides the IP address into network address and host address.
- ***Network Addresses:***
- The first address in the class, is called the network address and defines the organization network.

- ***Subnetting:***
- ***Hierarchy:***
- *Two-Level Hierarchy: No Subnetting*
- *Three-Level Hierarchy: Subnetting*
- ***Two-Level Hierarchy: No Subnetting:*** An IP address can define only two levels of hierarchy when not subnetted.
- The two common terms are prefix and suffix.
- The part of the address that defines the network is called the prefix.
- The part that defines the host is called the suffix.



Two levels of hierarchy in a IPv4 address

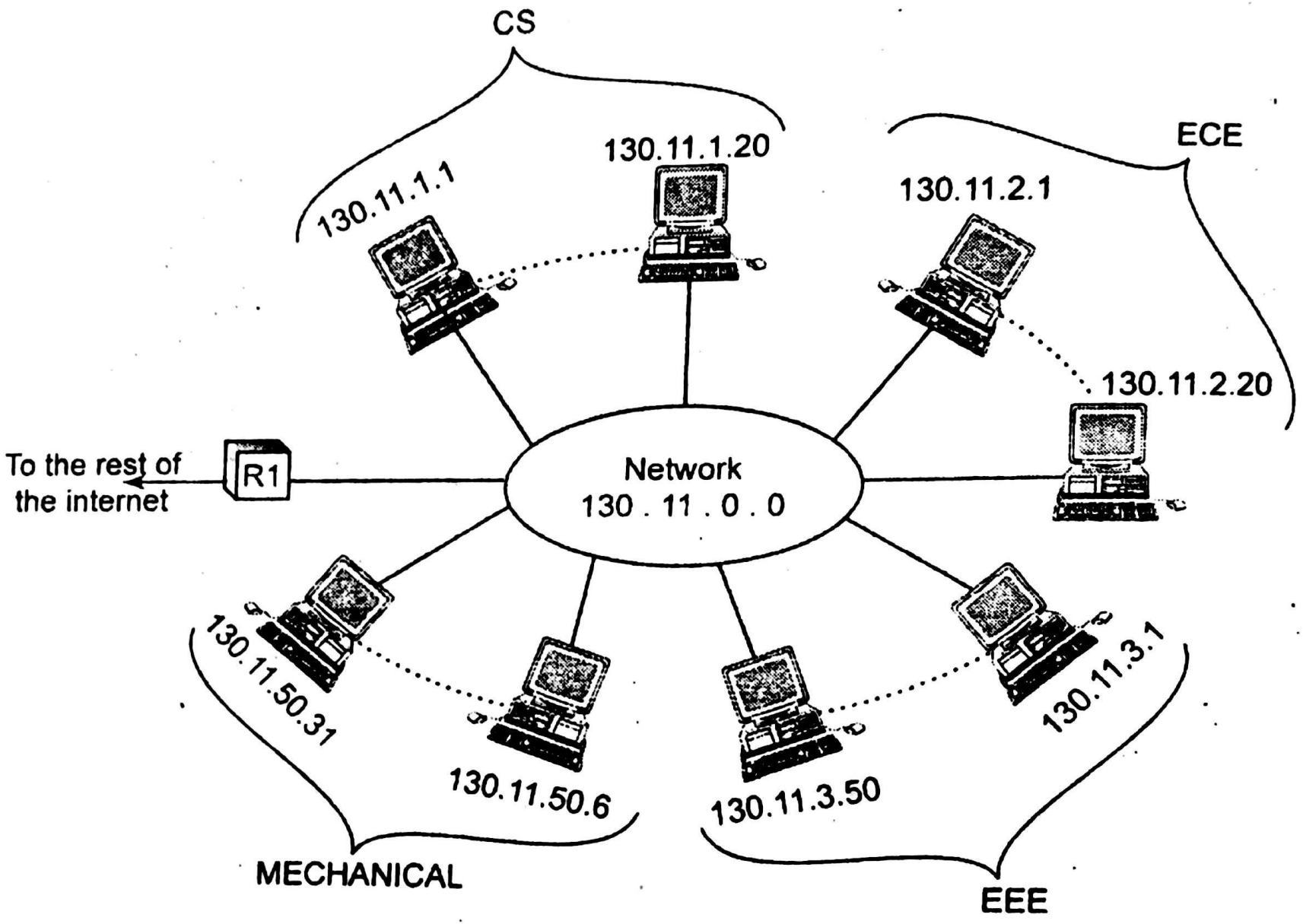
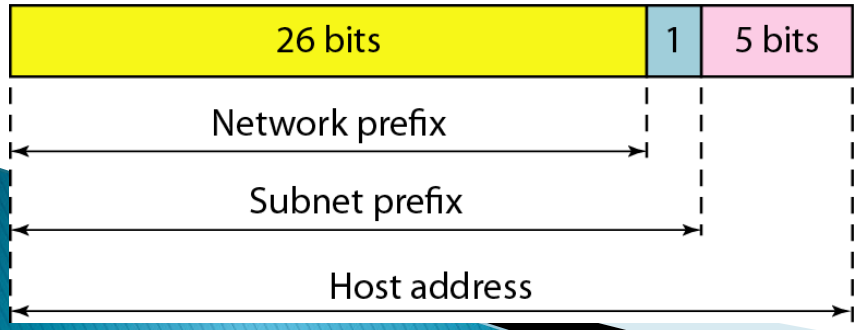
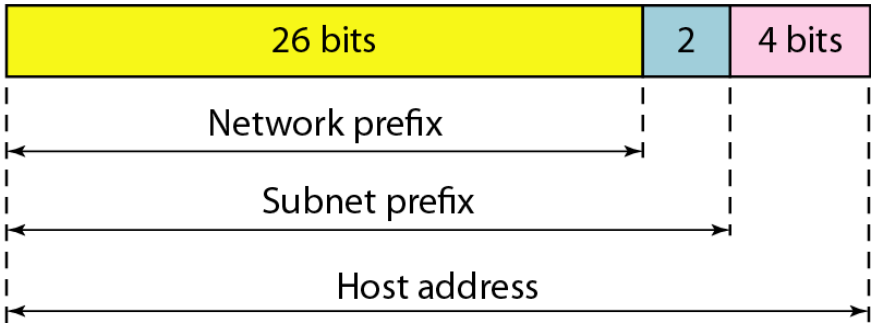


Fig.11.5. A network with 2 levels of hierarchy (not subnetted)

- ***Three-Level Hierarchy: Subnetting:***
- An organization that is granted a large block of addresses may want to create clusters of networks called subnets and divide the addresses between the different subnets.
- The rest of the world still sees the organization as one entity.
- But, internally there are several subnets.
- All messages are sent to the router address that connects the organization to the rest of the Internet
- The router routes the message to the appropriate subnets. The organization create small sub blocks of addresses, each assigned to specific subnets.

Two levels of hierarchy in a IPv4 address





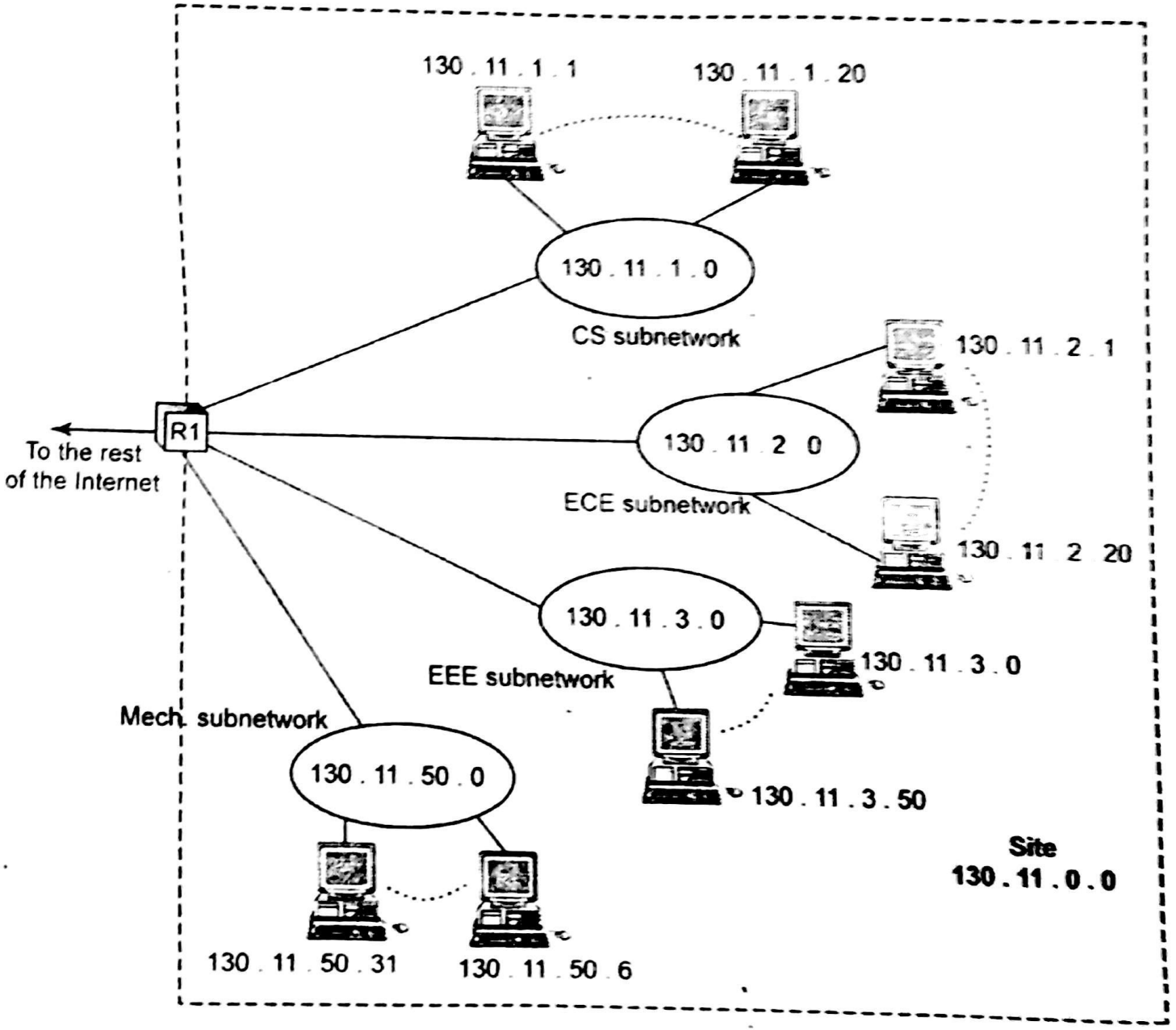


Fig.11.6. A network with 3 levels of hierarchy (subnetted)

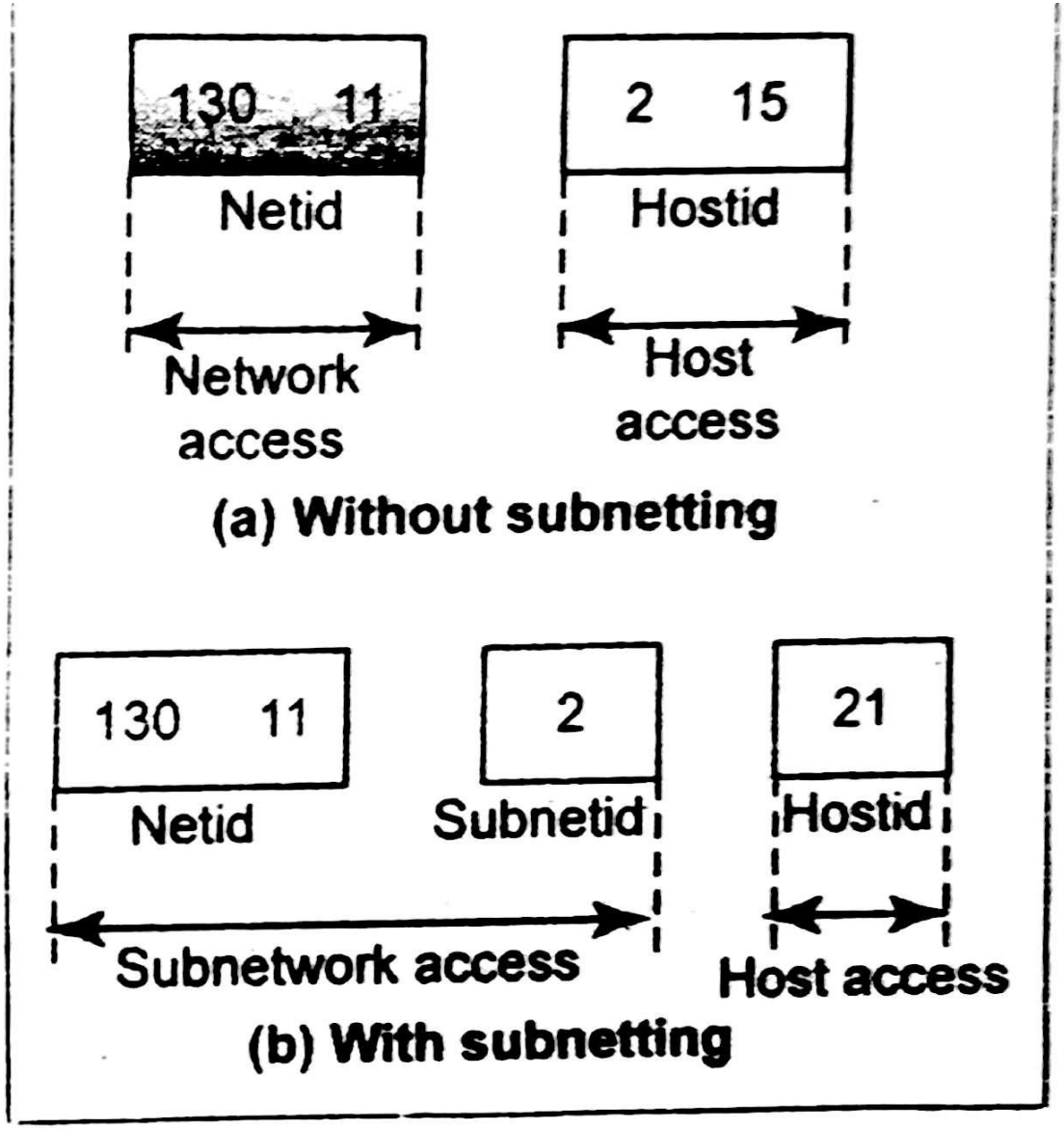


Fig.11.7. Address in a network with and without subnetting.

Masking types:

Boundary Level Masking	Non Boundary Level Masking
In this mask number is either 0 or 255.	In this mask number is greater than 0 or less than 255.
If the mask number is 255, then IP address is repeated in the subnetwork address.	Perform bitwise AND operation between IP address and mask number.
If the mask number is 0, then 0 is repeated in the subnetwork address.	

9. ***Find the subnetwork address for the following:***

a) IP address : 125.34.12.56

mask : 255.255.0.0

b) IP address : 141.181.14.16

Mask : 255.255.224.0

a) IP address : 125 . 34 . 12 . 56

mask : 255 . 255 . 0 . 0

---

subnetwork address : 125 . 34 . 0 . 0

---

b) IP address : 141 . 181 . 14 . 16

mask : 255 . 255 . 224 . 0

---

subnetwork address : 141 . 181 . 0 . 0

	1	1	1	0	0	0	0	0	(224)
(AND)	0	0	0	0	1	1	1	0	(14)
	0	0	0	0	0	0	0	0	

2	14		
2	7	-	0
2	3	-	1
	1	-	1

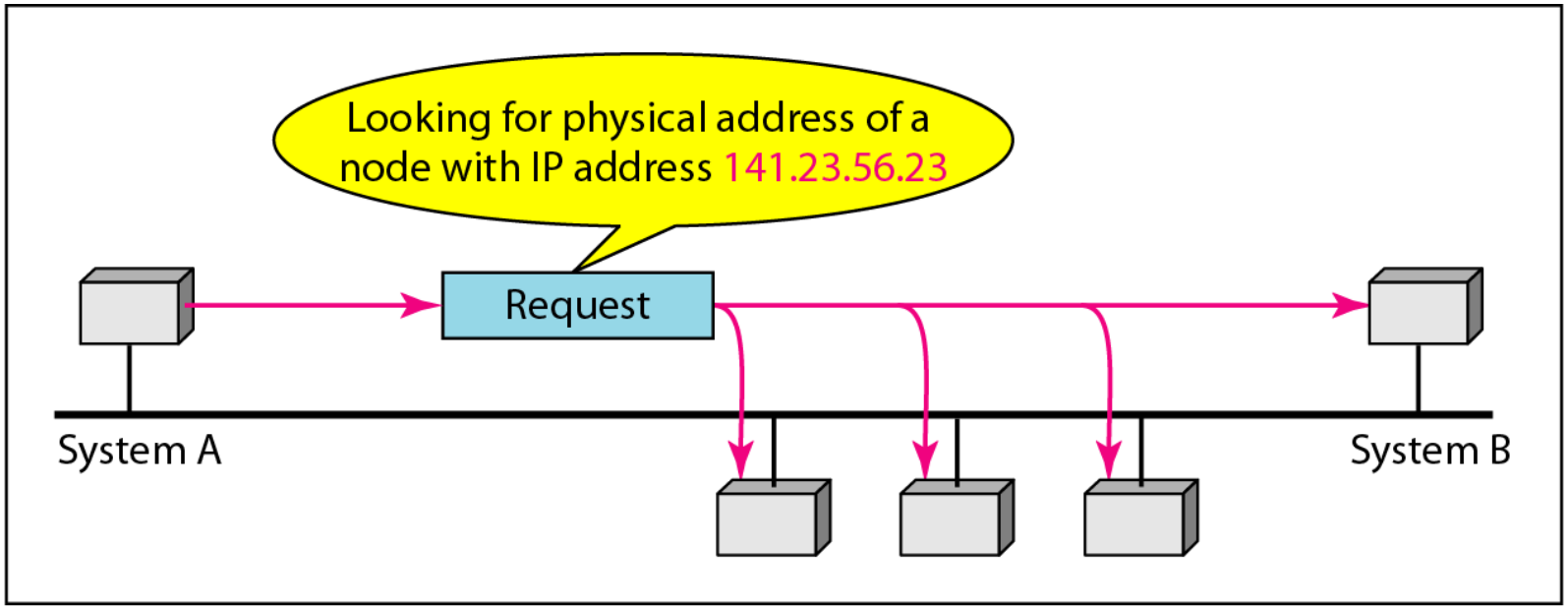
2	224		
2	112	-	0
2	56	-	0
2	28	-	0
2	14	-	0
2	7	-	0
2	3	-	1
	1	-	1

- **CIDR: Classless Interdomain Routing Notation**
- Using CIDR notation, a prefix of 205.100.0.0 of length is written as 205.100.0.0/22.
- The /22 notation indicated that the network mask is 22 bits.

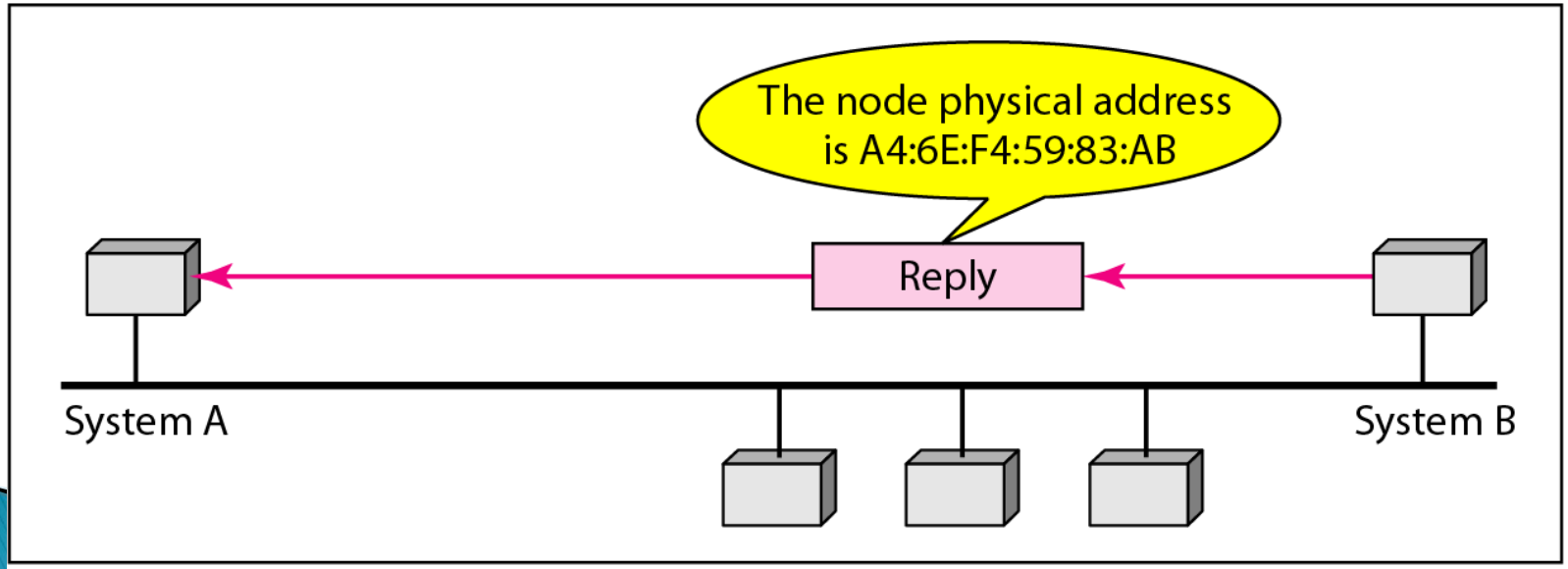
- **NETWORK LAYER PROTOCOLS:**
- **Mapping Logical to Physical Address:**
- **ARP - Address Resolution Protocol:**
- Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver.
- The logical (IP) address is obtained from the DNS, if the sender is the host or it is found in a routing table if the sender is a router.
- But the IP datagram must be encapsulated in a frame to be able to pass through the physical network.
- This means that the sender needs the physical address of the receiver.

- ***ARP operation:***
- The host or the router sends an ARP query packet.
- The packet includes the physical and IP addresses of the sender and the IP address of the receiver.
- Because the sender does not know the physical address of the receiver, the query is broadcasted over the network.
- Every host or router on the network receives and processes the ARP query packet.
- But only the intended recipient recognizes its IP address and sends back an ARP response packet.
- The response packet contains the recipient's IP and physical addresses.
- The packet is unicast directly to the inquirer by using the physical address received in the query packet.

ARP operation



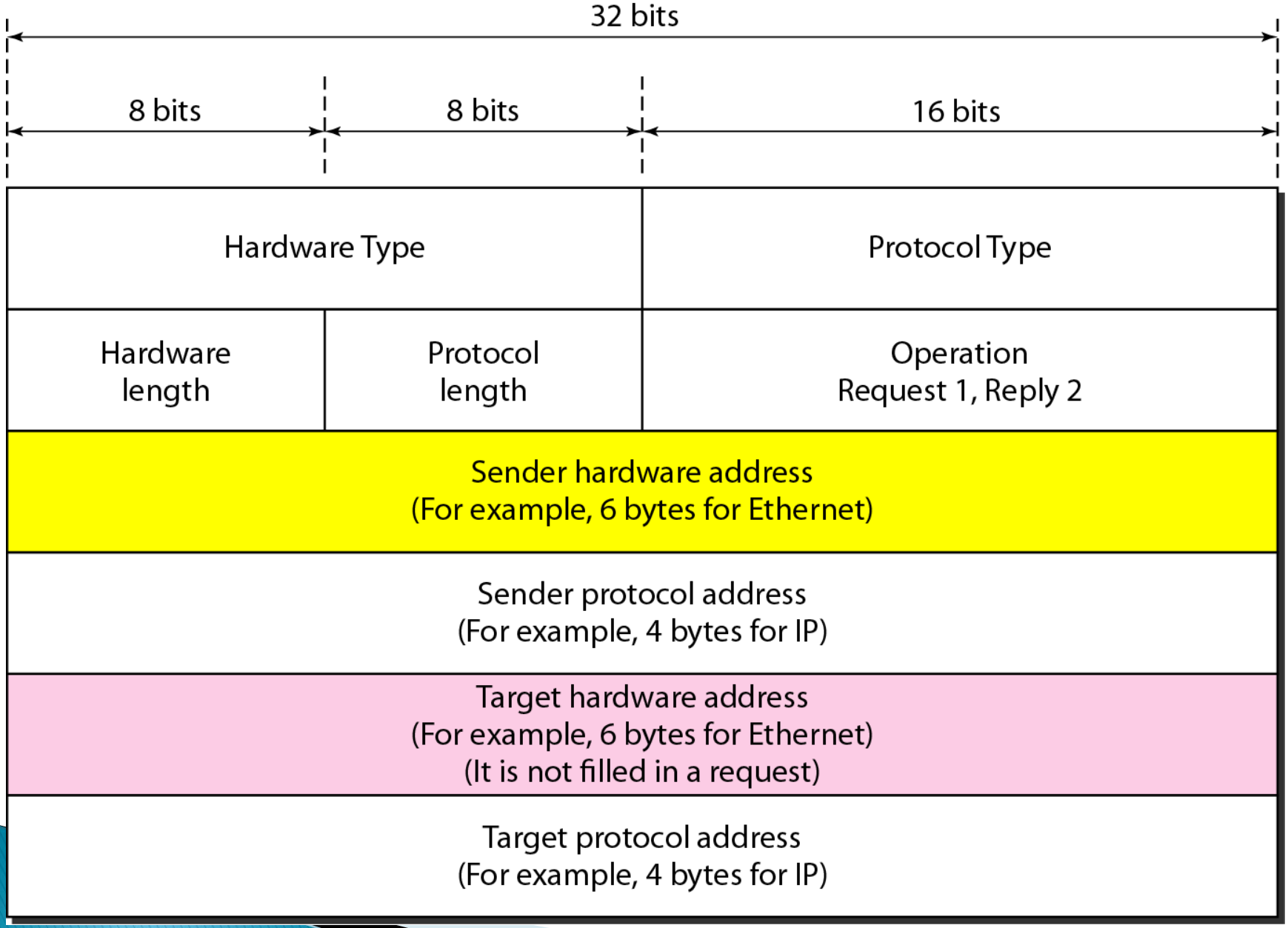
a. ARP request is broadcast



b. ARP reply is unicast

- The system on the left (A) has a packet that needs to be delivered to another system (B) with IP address 141.23.56.23.
- System A needs to pass the packet to its data link layer for the actual delivery, but it does not know the physical address of the recipient(B).
- It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IP address of 141.23.56.23.
- This packet is received by every system on the physical network, but only system B will answer it.
- System B sends an ARP reply packet that includes its physical address.
- Now system A can send all the packets it has for this destination by using the physical address it received.

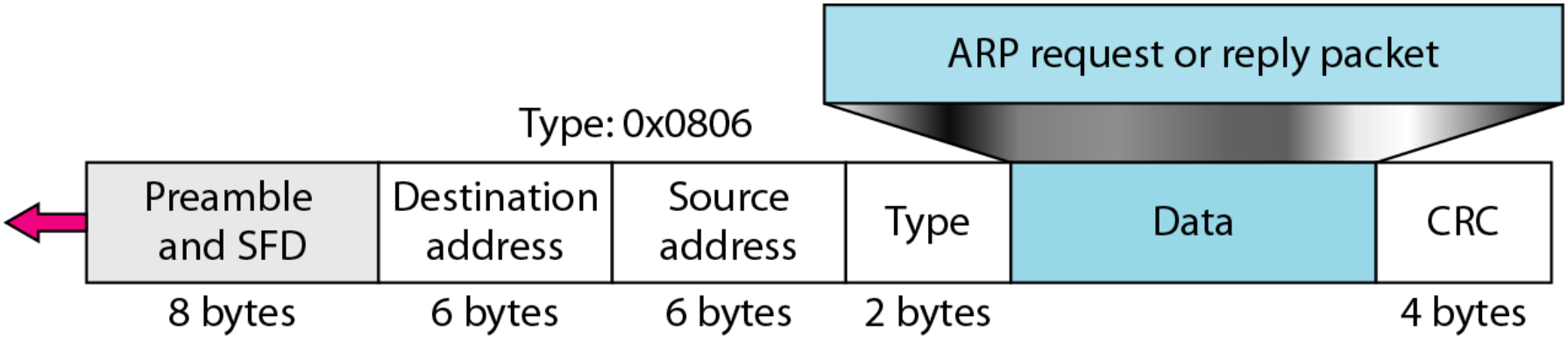
ARP packet format



- The fields of ARP packet are as follows:
- **Hardware type:** This is a 16-bit field defining the type of the network on which ARP is running.
- For example, Ethernet is given type 1.
- ARP can be used on any physical network.
- **Protocol type:** This is a 16-bit field defining the protocol. For example, the value of this field for the IPv4 protocol is 080016.
- **Hardware length:** This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.
- **Protocol length:** This is an 8-bit field defining the length of the logical address in bytes. For example, for the IPv4 protocol the value is 4.

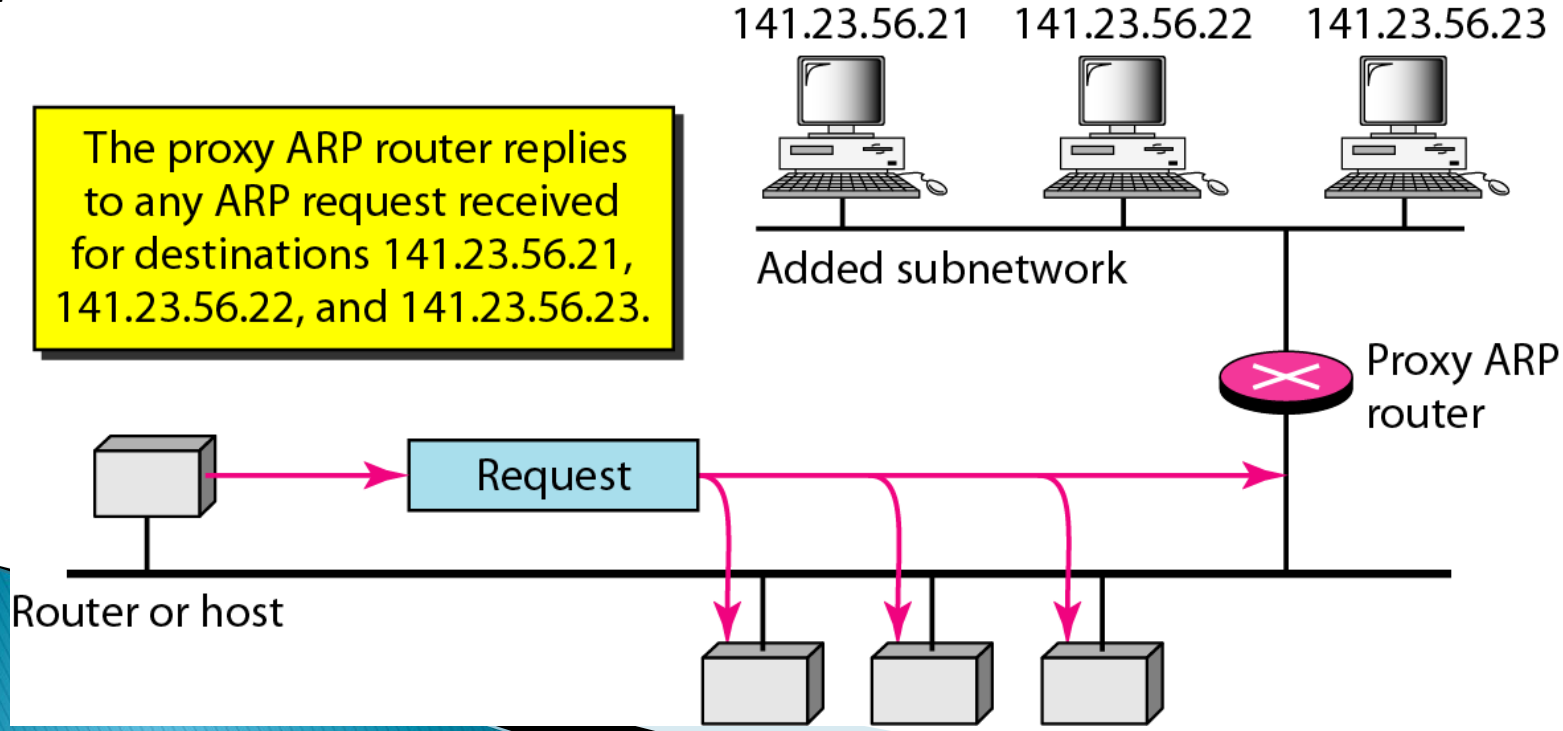
- **Operation:** This is a 16-bit field defining the type of packet.
- Two packet types are defined: ARP request (1) and ARP reply (2).
- **Sender hardware address:** This is a variable-length field defining the physical address of the sender. For example, for Ethernet this field is 6 bytes long.
- **Sender protocol address:** This is a variable-length field defining the logical address of the sender. For the IP protocol, this field is 4 bytes long.
- **Target hardware address:** This is a variable-length field defining the physical address of the target. For example, for Ethernet this field is 6 bytes long.
- For an ARP request message, this field is all 0s because the sender does not know the physical address of the target.
- **Target protocol address:** This is a variable-length field defining the logical address of the target. For the IPv4 protocol, this field is 4 bytes long.

- ***Encapsulation:***
- An ARP packet is encapsulated directly into a data link frame.
- For example an ARP packet is encapsulated in an Ethernet frame.
- The type field indicates that the data carried by the frame are an ARP packet.



Encapsulation of ARP packet

- ***Proxy ARP:***
- *A proxy ARP is an* ARP that acts on behalf of a set of hosts.
- Whenever a router running a proxy ARP, receives an ARP request looking for the IP address of one of these hosts, the router sends an ARP reply announcing its own hardware (physical) address.
- After the router receives the actual IP packet, it sends the packet to the appropriate host or router.

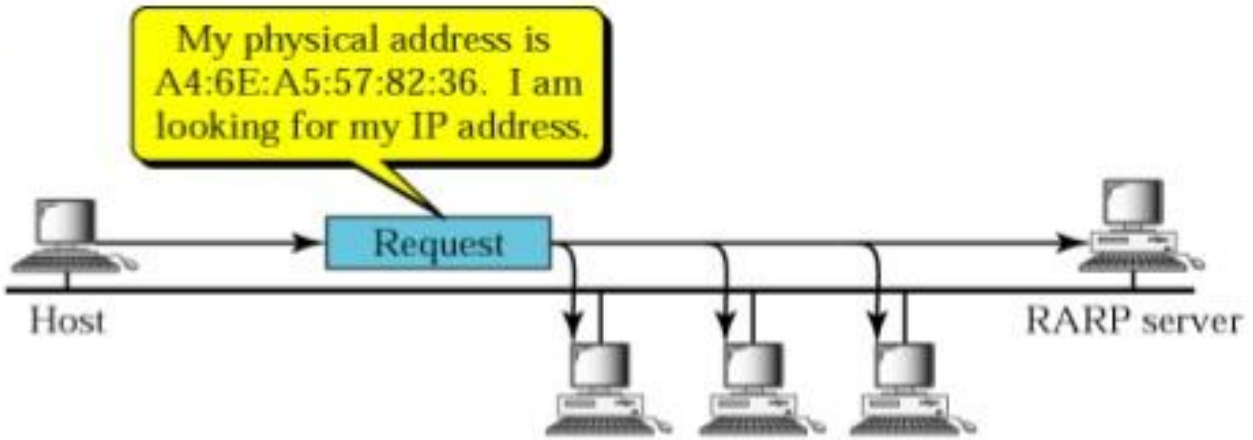


- **Mapping Physical to Logical Address:**
- RARP – Reverse Address Resolution Protocol
- DHCP – Dynamic Host Configuration Protocol
- There are occasions in which a host knows its physical address, but needs to know its logical address.
- This may happen in two cases:
- 1. A diskless station is just booted. The station can find its physical address by checking its interface, but it does not know its IP address.
- 2. An organization does not have enough IP addresses to assign to each station and it needs to assign IP addresses on demand.
- The station can send its physical address and ask for a short time lease.

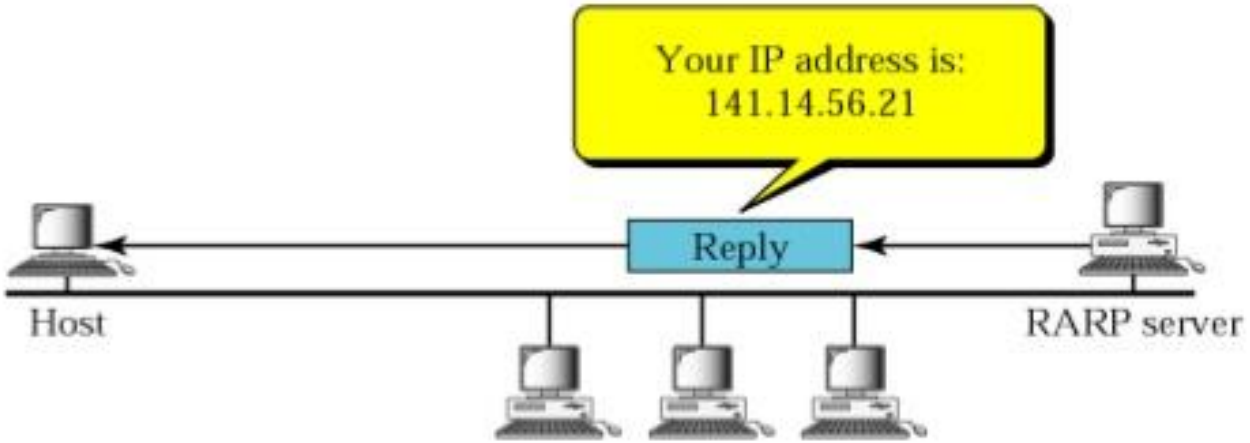
- ***RARP:* Reverse Address Resolution Protocol (RARP)** finds the logical address for a machine that knows only its physical address.
- To create an IP datagram, a host or a router needs to know its own IP address.
- The IP address of a machine is usually read from its configuration file stored on a disk file.
- However, a diskless machine is usually booted from ROM, which has minimum booting information.
- The ROM is installed by the manufacturer.
- It cannot include the IP address because the IP addresses on a network are assigned by the network administrator.

- The machine can get its physical address by reading its NIC.
- It can then use the physical address to get the logical address by using the RARP protocol.
- ***RARP operation:***
- A RARP request is created and broadcast on the local network.
- Another machine on the local network that knows the IP address will respond with a RARP reply.
- The requesting machine must be running a RARP client program.
- The responding machine must be running a RARP server program.

RARP operation



a. RARP request is broadcast



b. RARP reply is unicast

Packet Format

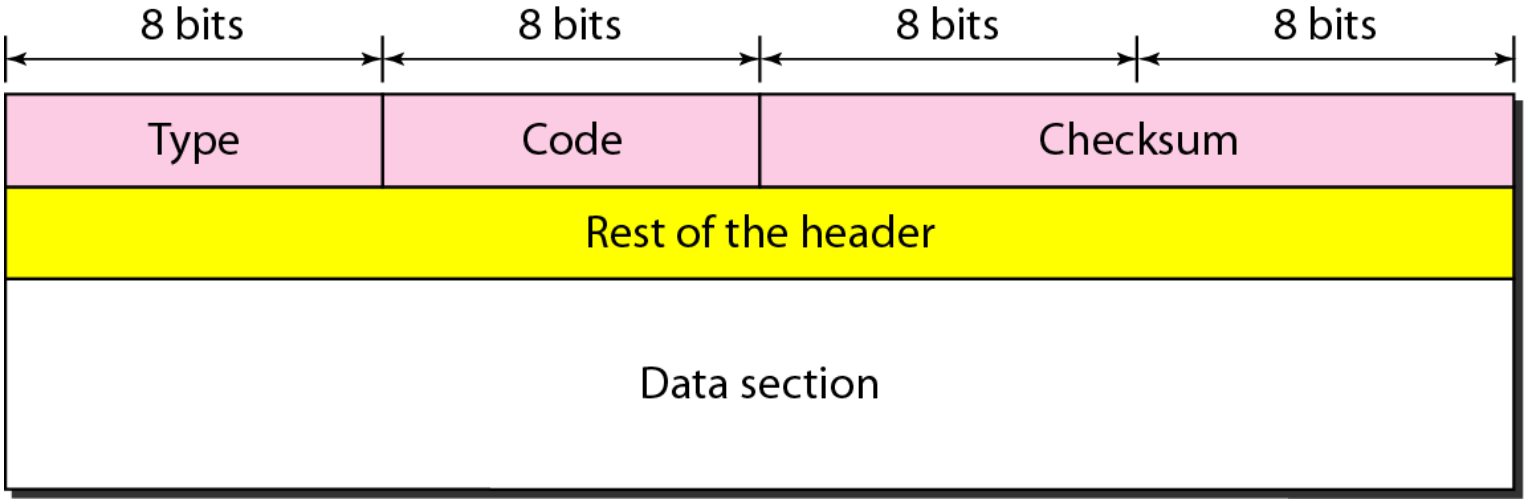
Hardware type		Protocol type
Hardware length	Protocol length	Operation Request 3, Reply 4
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP) (It is not filled for request)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled for request)		
Target protocol address (For example, 4 bytes for IP) (It is not filled for request)		

ICMP – Internet Control Message Protocol

- The IP protocol has no error-reporting or error-correcting mechanism.
- What happens if a router must discard a datagram because it cannot find a router to the final destination, or because the time-to-live field has a zero value?
- What happens if the final destination host must discard all fragments of a datagram because it has not received all fragments within a predetermined time limit?
- These are examples of situations where an error has occurred and the IP protocol has no built-in mechanism to notify the original host.

- The IP protocol also lacks a mechanism for host and management queries.
- A host sometimes needs to determine if a router or another host is alive.
- And sometimes a network administrator needs information from another host or router.
- The Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies. It is a companion to the IP protocol.

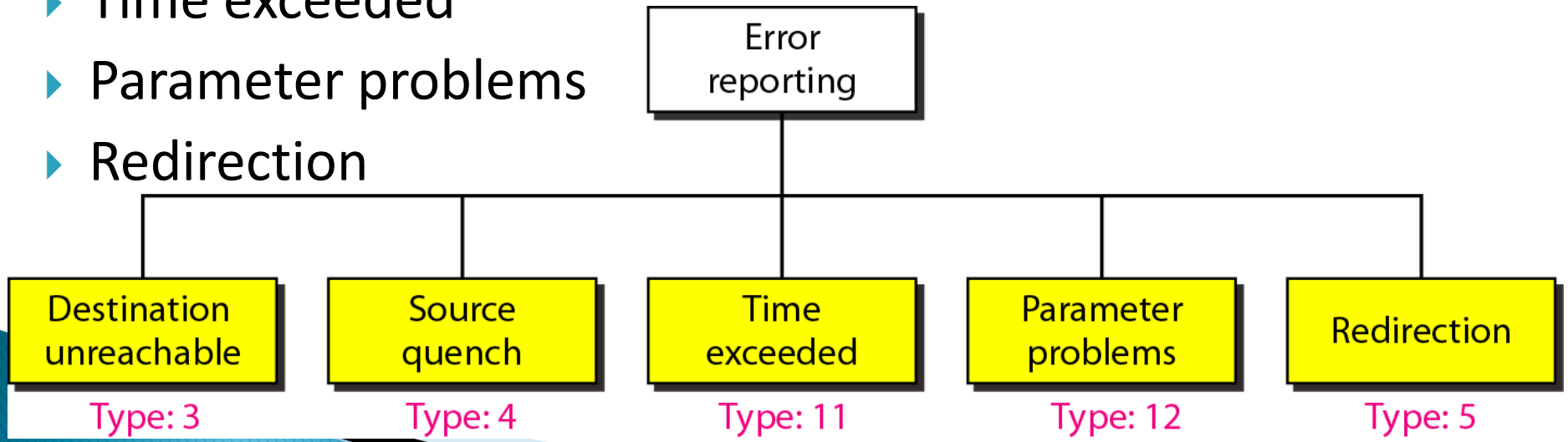
- **Message Format:** An ICMP message has an 8-byte header and a variable-size data section.



General format of ICMP messages

- **The type field,** defines the type of the message.
- **The code field** specifies the reason for the particular message type.
- **The checksum field** is used to detect errors in the ICMP message.

- **Error Reporting:** One of the main responsibilities of ICMP is to report errors.
- ICMP always reports error messages to the original source.
- ICMP uses the source IP address to send the error message to the source (originator) of the datagram.
- **Five types of errors are handled:**
- Destination unreachable
- Source quench
- Time exceeded
- Parameter problems
- Redirection

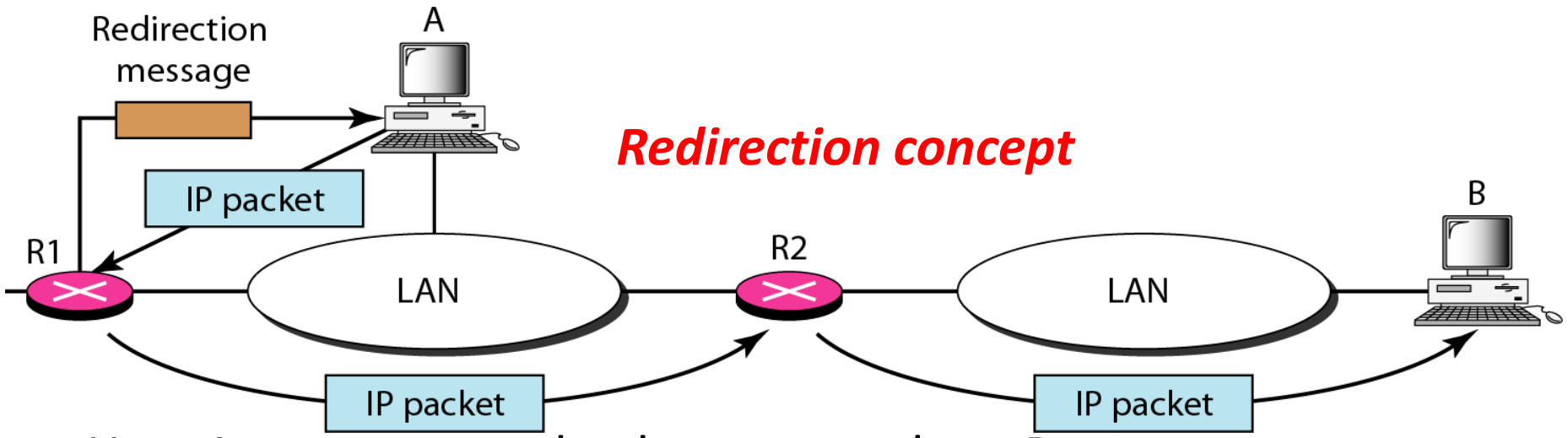


- ***Destination Unreachable:*** When a router cannot route a datagram or a host cannot deliver a datagram, the datagram is discarded and the router or the host sends a destination-unreachable message back to the source host that initiated the datagram.
- The destination-unreachable messages can be created by either a router or the destination host.
- **Source quench:** The lack of flow control can create congestion in routers or the destination host.
- A router or a host has a limited-size queue (buffer) for incoming datagrams waiting to be forwarded in the case of a router or to be processed in the case of a host.
- If the datagrams are received much faster than they can be forwarded or processed, the queue may overflow.

- In this case, the router or the host has no choice but to discard some of the datagrams.
- **The source-quench message** in ICMP was designed to add a kind of flow control to the IP.
- When a router or host discards a datagram due to congestion, it sends a source-quench message to the sender of the datagram.
- This message has two purposes.
- First, it informs the source that the datagram has been discarded.
- Second, it warns the source that there is congestion somewhere in the path and that the source should slow down (quench) the sending process.

- **Time exceeded:** The time-exceeded message is generated in **two cases:**
- First **when a datagram visits a router**, the value of this field is decremented by 1.
- When the time-to-live value reaches 0, after decrementing, the router discards the datagram.
- when the datagram is discarded, a time-exceeded message must be sent by the router to the original source.
- Second, a time-exceeded message is also generated **when not all fragments that make up a message** arrive at the destination host within a certain time limit.

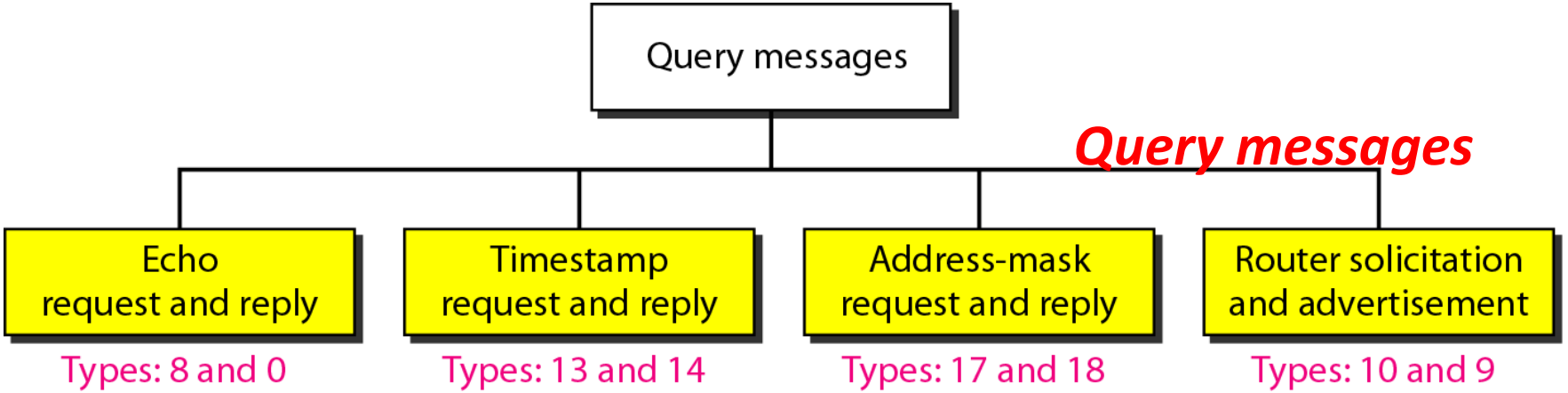
- ***Parameter Problem:*** If a router or the destination host discovers an ambiguous or missing value in any field of the datagram, it discards the datagram and sends a parameter-problem message back to the source.
- ***Redirection:*** When a host comes up, its routing table has a limited number of entries.
- It usually knows the IP address of the default router.
- For this reason, the host may send a datagram, which is destined for another network, to the wrong router.
- In this case, the router that receives the datagram will forward the datagram to the correct router.
- To update the routing table of the host, it sends a redirection message to the host.



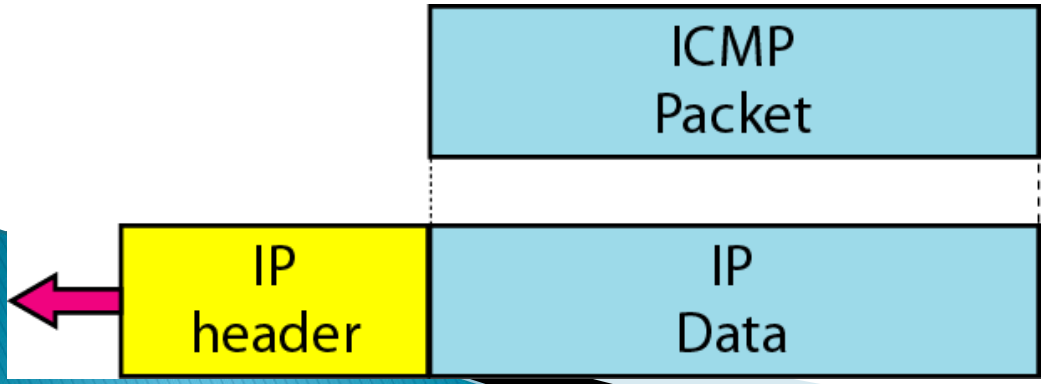
Redirection concept

- Host A wants to send a datagram to host B.
- Router R2 is the most efficient routing choice.
- But host A did not choose router R2.
- The datagram goes to R1 instead.
- Router R1, after consulting its table, finds that the packet should have gone to R2.
- It sends the packet to R2 and, at the same time, sends a redirection message to host A.
- Host A's routing table can now be updated.

- **Query:** In addition to error reporting, ICMP can diagnose some network problems.
- This is accomplished through the query messages, a group of four different pairs of messages.
- A query message is encapsulated in an IP packet, which in turn is encapsulated in a data link layer frame.



Query messages



Encapsulation of ICMP query messages in IP packet

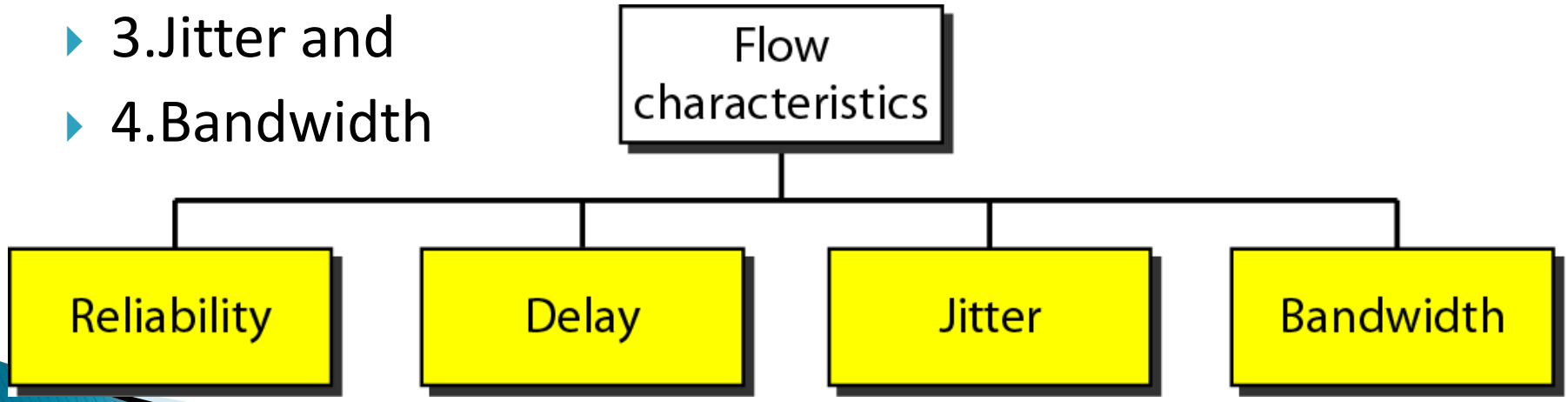
- ***Echo Request and Reply:*** The combination of echo-request and echo-reply messages determines whether two systems (hosts or routers) can communicate with each other.
- The echo-request and echo-reply messages can be used to determine if there is communication at the IP level.
- Because ICMP messages are encapsulated in IP datagrams, the receipt of an echo-reply message by the machine that sent the echo request is proof that the IP protocols in the sender and receiver are communicating with each other using the IP datagram.
- Also, it is proof that the intermediate routers are receiving, processing, and forwarding IP datagrams.
- Most systems provide a version of the *ping command that can create a series* of echo-request and echo-reply messages.

- ***Timestamp Request and Reply:*** Two machines (hosts or routers) can use the timestamp request and timestamp reply messages to determine the round-trip time needed for an IP datagram to travel between them.
- It can also be used to synchronize the clocks in two machines.
- ***Address-Mask Request and Reply:*** A host may know its IP address, but it may not know the corresponding mask.
- For example, a host may know its IP address as 159.31.17.24, but it may not know that the corresponding mask is /24.
- To obtain its mask, a host sends an **address-mask-request** message to a router on the LAN.
- The router receiving the address-mask-request message responds with an **address-mask-reply** message, providing the necessary mask for the host.
- This can be applied to its IP address to get its subnet address.

- ***Router Solicitation and Advertisement:*** *A* host that wants to send data to a host on another network needs to know the address of routers connected to its own network.
- Also, the host must know if the routers are alive and functioning.
- A host can broadcast (or multicast) a **router-solicitation message.**
- The router or routers that receive the solicitation message broadcast their routing information using the **router-advertisement message**.
- when a router sends out an advertisement, it announces not only its own presence but also the presence of all routers on the network of which it is aware.

QoS-QUALITY OF SERVICE

- It refers to the capability of a network to provide better service to network traffic.
- Flow Characteristics:
- Traditionally, four types of characteristics are attributed to a flow:
- 1.Reliability,
- 2.Delay,
- 3.Jitter and
- 4.Bandwidth



- ***1.Reliability:*** Reliability is a characteristic that a flow needs. Lack of reliability means losing a packet or acknowledgment, which entails retransmission.
- The sensitivity of application programs to reliability is not the same.
- For example, it is more important that electronic mail, file transfer, and Internet access have reliable transmissions than telephony or audio conferencing.
- ***2.Delay:***
- Source-to-destination delay is another flow characteristic.
- Applications can tolerate delay in different degrees. In this case, telephony, audio conferencing and video conferencing need minimum delay, while delay in file transfer or e-mail is less important.

- ***3.Jitter:*** Jitter is the variation in delay for packets belonging to the same flow.
- High jitter means the difference between delays is large; low jitter means the variation is small.
- For example, if four packets depart at times **0, 1, 2, 3** and arrive at **20, 21, 22, 23**, all have the same delay, 20 units of time.
- On the other hand, if the above four packets arrive at **21, 23, 21, and 28,** they will have different delays: **21,22, 19, and 24.**
- For applications such as audio and video, the first case is completely acceptable, but the second case is not acceptable.

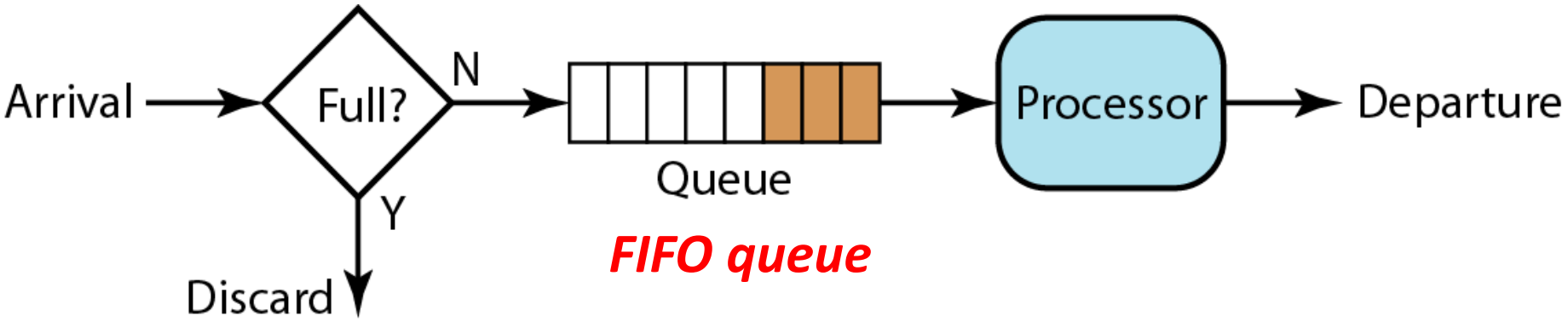
- ***4. Bandwidth:***
- Different applications need different bandwidths.
- In video conferencing we need to send millions of bits per second to refresh a color screen while the total number of bits in an e-mail may not reach even a million.

TECHNIQUES TO IMPROVE QoS:

- 1.Scheduling
- 2.Traffic Shaping
- 3.Resource Reservation
- 4.Admission Control

- **1.Scheduling:** Packets from different flows arrive at a router for processing.
- A good scheduling technique treats the different flows in a fair and appropriate manner.
- Several scheduling techniques are designed to improve the quality of service.
- We discuss three of them here:
- (i)FIFO queuing
- (ii)Priority queuing and
- (iii)weighted fair queuing.

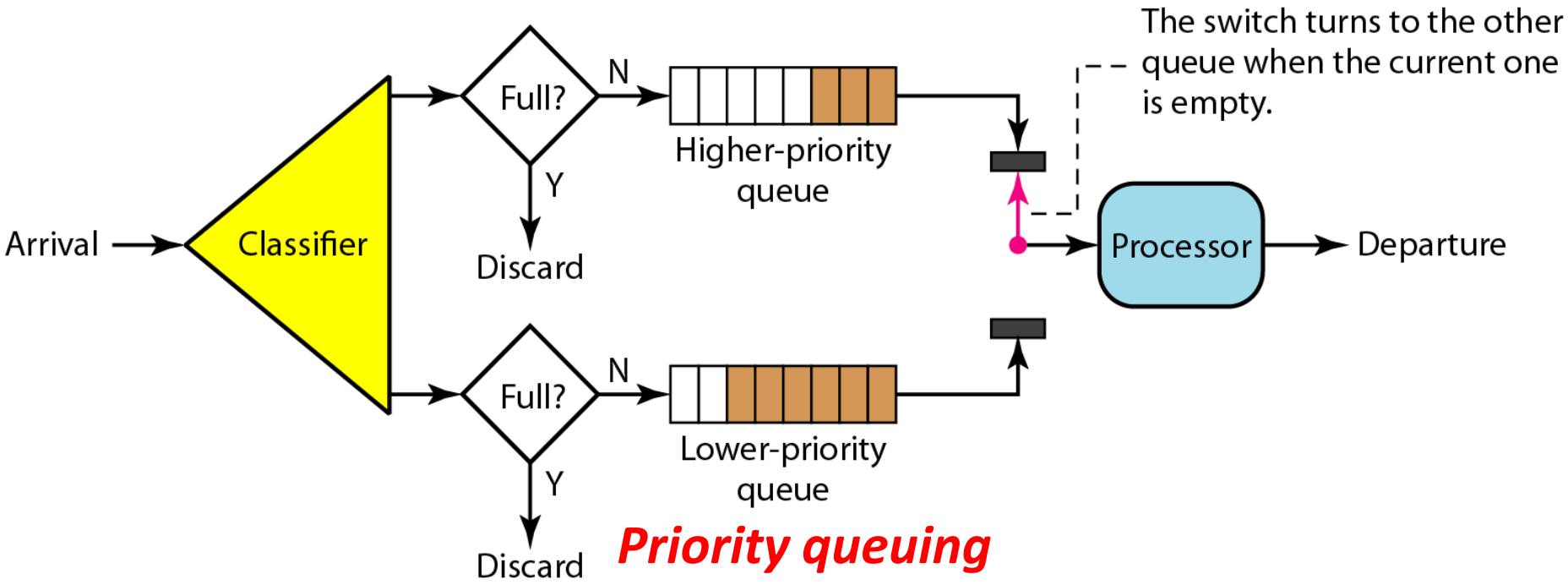
- ***(i)FIFO Queuing:***
- In first-in, first-out (FIFO) queuing, packets wait in a buffer (queue) until the router is ready to process them.
- If the average arrival rate is higher than the average processing rate, the queue will fill up and new packets will be discarded.



FIFO queue

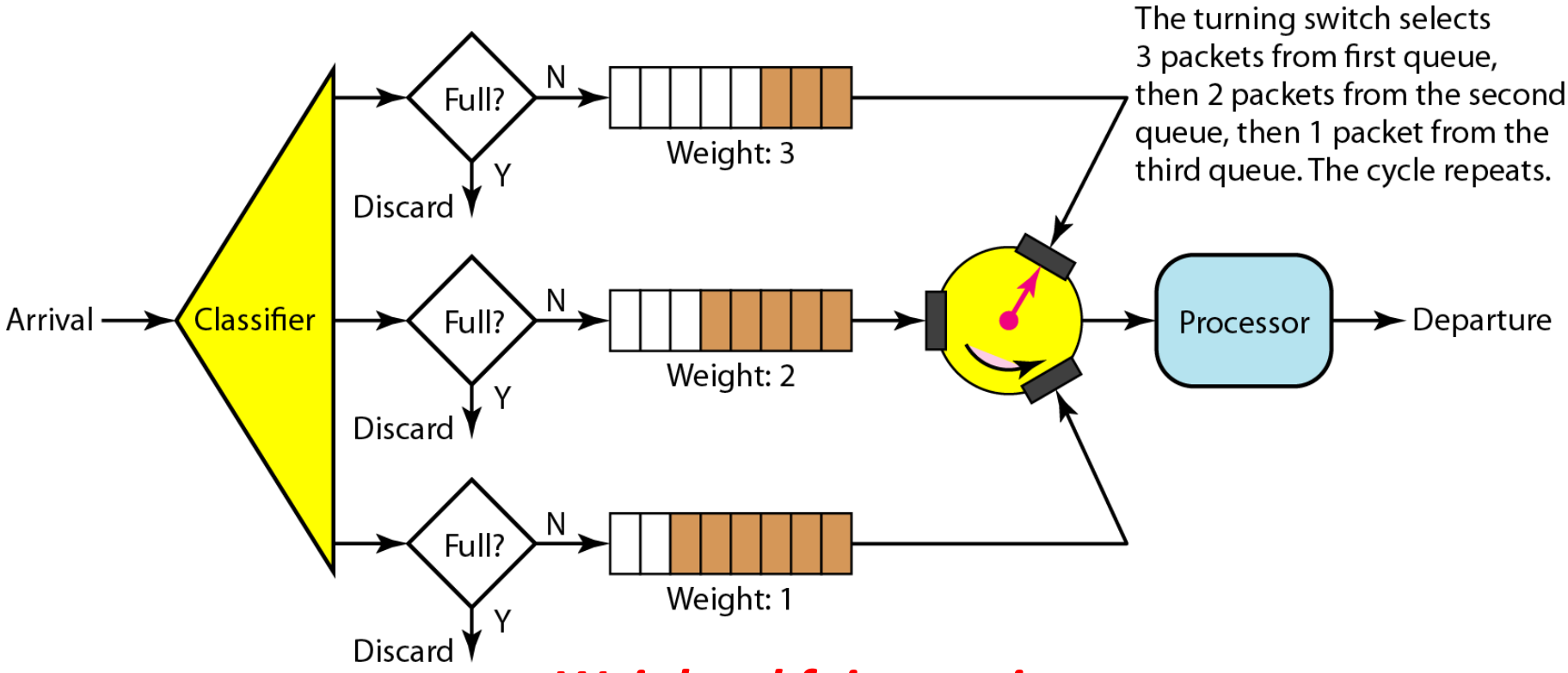
- ***(ii)Priority Queuing:***
- In priority queuing, packets are first assigned to a priority class.
- Each priority class has its own queue.
- The packets in the highest-priority queue are processed first.
- Packets in the lowest-priority queue are processed last.
- A priority queue can provide better QoS than the FIFO queue because higher priority traffic, such as multimedia, can reach the destination with less delay.

- However, there is a potential drawback. If there is a continuous flow in a high-priority queue, the packets in the lower-priority queues will never have a chance to be processed.
- This is a condition called *starvation.*



- ***(iii)Weighted Fair Queuing:*** A better scheduling method is weighted fair queuing.
- In this technique, the packets are still assigned to different classes and admitted to different queues.
- The queues are weighted based on the priority of the queues
- Higher priority means a higher weight.
- The system processes packets in each queue in a round-robin fashion with the number of packets selected from each queue based on the corresponding weight.

- For example, if the weights are 3, 2, and 1, three packets are processed from the first queue, two from the second queue, and one from the third queue.

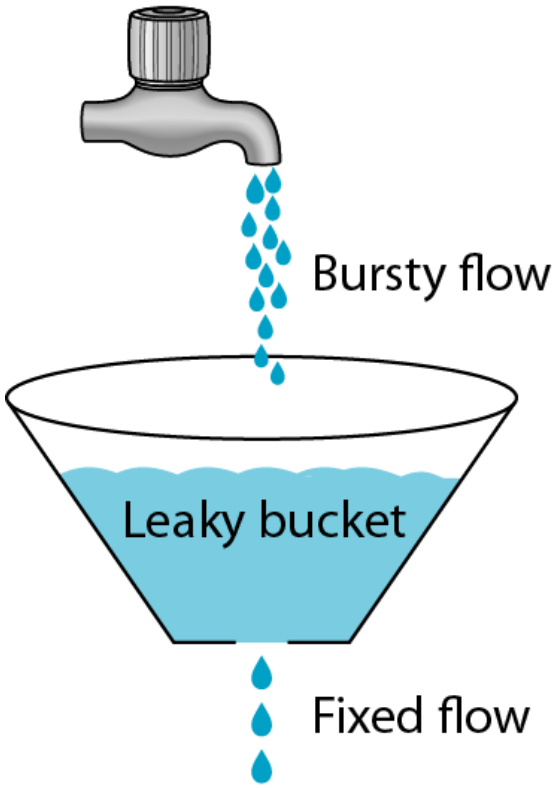


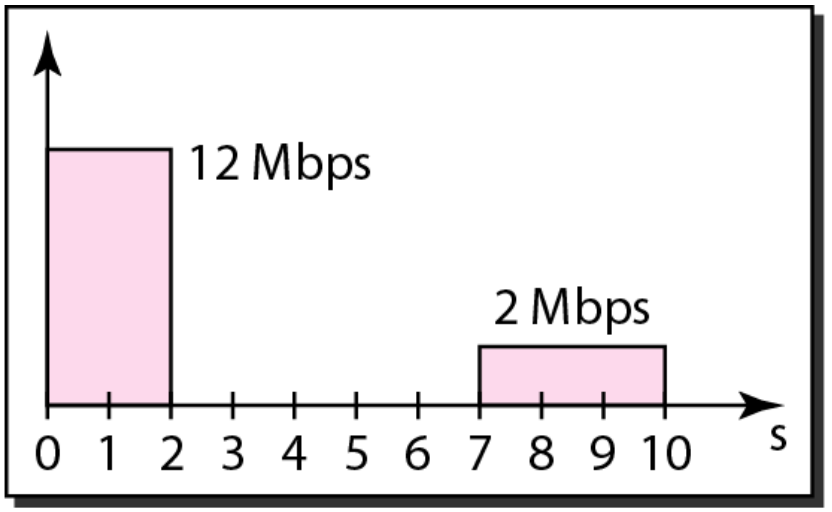
Weighted fair queuing

- **2.Traffic Shaping:** Traffic shaping is a mechanism to control the amount and the rate of the traffic sent to the network.
- Two techniques can shape traffic:
- (i)Leaky bucket algorithm and
- (ii)Token bucket algorithm.
- **(i)Leaky Bucket Algorithm:** If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket.
- The rate at which the water leaks does not depend on the rate at which the water is input to the bucket unless the bucket is empty.
- The input rate can vary, but the output rate remains constant.

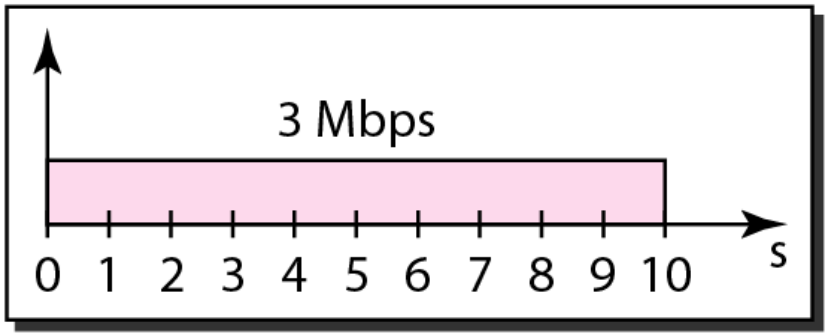
- Similarly, in networking, a technique called **leaky bucket** can smooth out bursty traffic.
- Bursty chunks are stored in the bucket and sent out at an average rate.

Leaky Bucket Algorithm





Bursty data



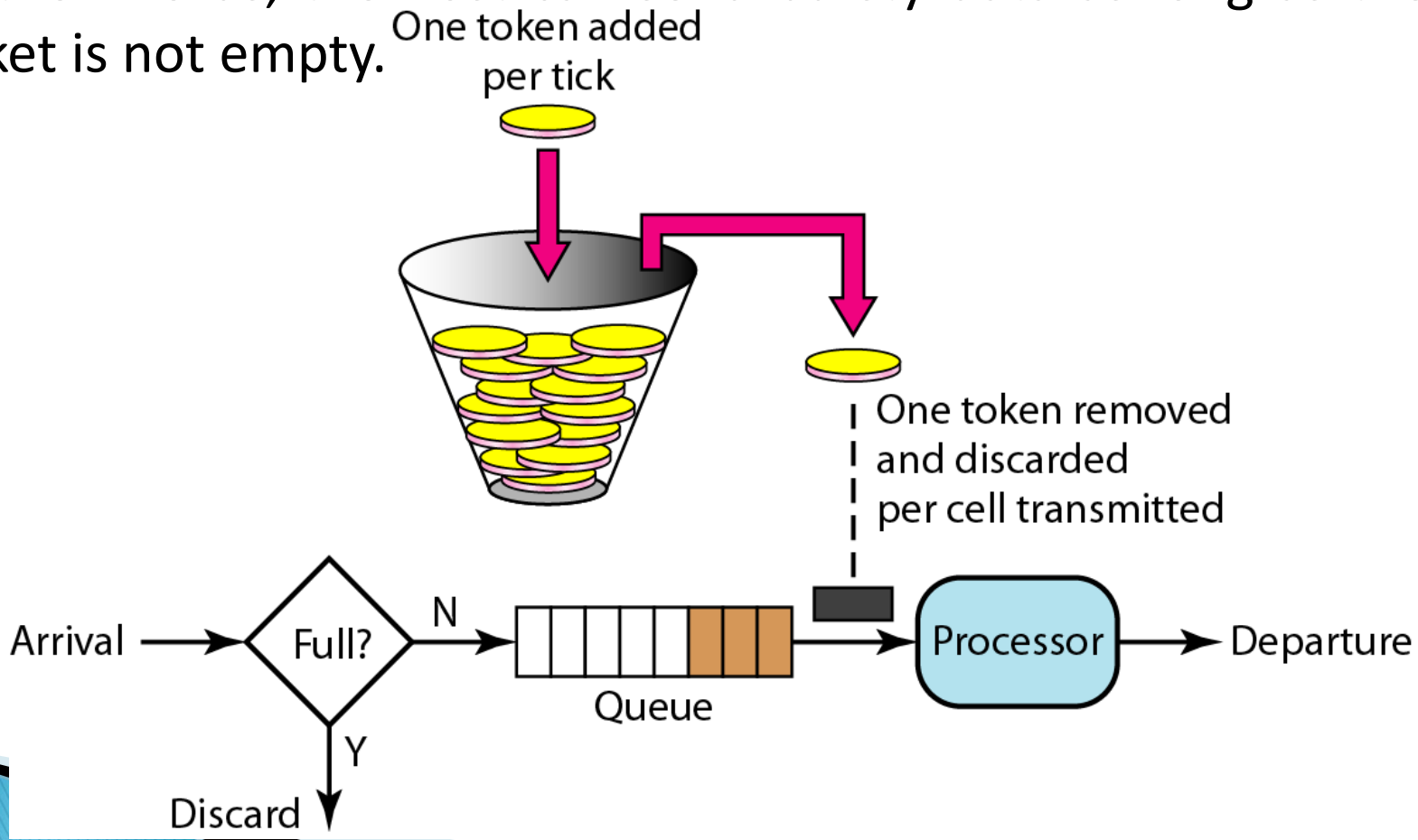
Fixed-rate data

- In this leaky bucket implementation there is a FIFO queue to holds the packet.
- If the traffic consists of fixed-size packets the process removes a fixed number of packets from the queue at each tick of the clock.
- If the traffic consists of variable-length packets, the fixed output rate must be based on the number of bytes or bits.
- If the rule is 1024 bytes per tick, a single 1024 byte packet can be admitted on a tick, two 512 byte packets and so on.

- **(ii)Token bucket algorithm:** The token bucket allows bursty traffic at a regulated maximum rate.
- The token bucket algorithm allows idle hosts to accumulate credit for the future in the form of tokens.
- For each tick of the clock, the system sends *n tokens to the bucket.*
- *The system removes one token for* every cell (or bytes) of data sent.

- For example, if *n is 100 and the host is idle for 100 ticks,* the bucket collects 10,000 tokens.
- Now the host can consume all these tokens in one tick with 10,000 cells, or the host takes 1000 ticks with 10 cells per tick.
- In other words, the host can send bursty data as long as the bucket is not empty.

Token Bucket Algorithm



- **3.Resource Reservation:**
- A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on.
- The quality of service is improved if these resources are reserved beforehand.
- **4.Admission Control:**
- Admission control refers to the mechanism used by a router to accept or reject a flow based on predefined parameters called flow specifications.
- Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow.

- **Flow Specification:** When a source makes a reservation, it needs to define a flow specification.
- A flow specification has two parts: Rspec (resource specification) and Tspec (traffic specification).
- Rspec defines the resource that the flow needs to reserve (buffer, bandwidth, etc.).
- Tspec defines the traffic characterization of the flow.
- **Admission:** After a router receives the flow specification from an application, it decides to admit or deny the service. The decision is based on the previous commitments of the router and the current availability of the resource.

- **NETWORK LAYER DESIGN ISSUES:**
- 1. The network layer is concerned with getting packets from the source all the way to the destination.
- 2. Getting to the destination may require making many hops at intermediate routers along the way.
- 3. To achieve its goals, the network layer must know about the topology of the network (i.e., the set of all routers and links) and choose appropriate paths through it, even for large networks.
- 4. It must also take care when choosing routes to avoid overloading some of the communication lines and routers while leaving others idle.