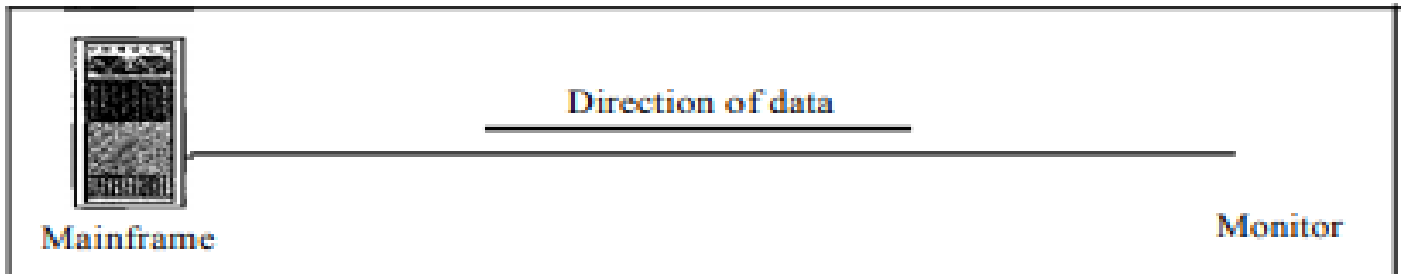


## Basics of Computer Networks

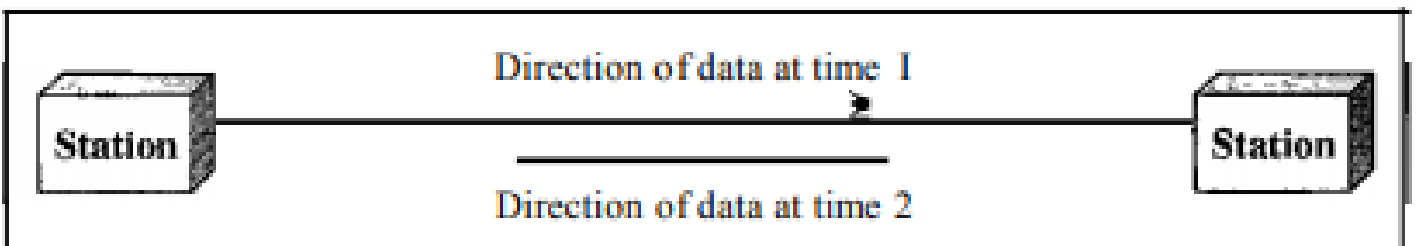
- A **computer network** is a telecommunications network, which allows digital devices(nodes) to exchange data between each other using either wired or wireless connections to share resources (h/w or s/w) e.g. internet. A collection of autonomous computers interconnected by a single technology.
- Networks come in many sizes, shapes and forms. They are usually connected together to make larger networks, with the Internet being the most well-known example of a network of networks.
- Data communications are the exchange of data between two devices via some transmission medium. A data communication system has five components
  - **Message**- information (data) to be communicated e.g. text, audio, video.
  - **Sender**- device how sends the message (computer, phone, camera etc.)
  - **Receiver**- device how receives the message (computer, phone, television etc.)
  - **Transmission medium** – is the physical path by which a message travels from sender to receiver.
  - **Protocol** – the set of rules that governs the data communication.
- Effectiveness of the data communications system depends on four fundamental characteristics
  - **Delivery**- must deliver the data to correct destination.
  - **Accuracy**- must be delivered accurately without any error
  - **Timeliness**- must deliver the data in a timely manner, sometime time in real time applications data delivered after time is useless.
  - **Jitter**- Refers to variation in the packet arrival time i.e. the uneven delay between the packets (mismatch in audio and picture in a video)

- Data flow between two systems can be categorised into three types –
- **Simplex** – the communication is unidirectional, as a one-way street. one device always sends can always send other can always receive. E.g. radio, mouse.
  - The simplex mode can use the entire capacity of the channel to send data in one direction.
  -



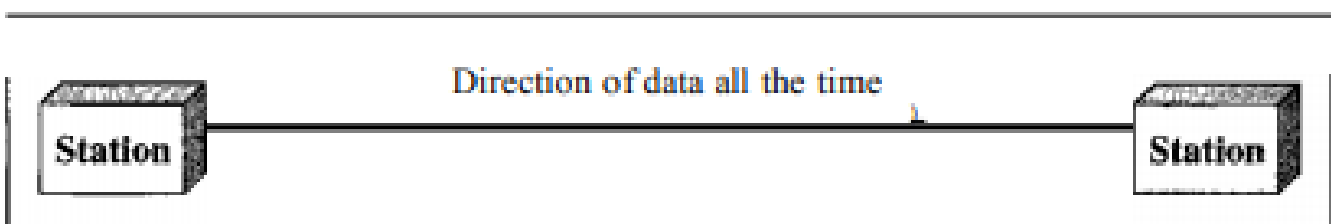
#### a. Simplex

- **Half duplex** – each station can both transmit and receive, but not at the same time. E.g. like a one lane road, walkie-talkie etc.
  - When one device is sending, the other can only receive, and vice versa.
  - In a half-duplex transmission, the entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time.
  - Walkie-talkies are both half-duplex systems.



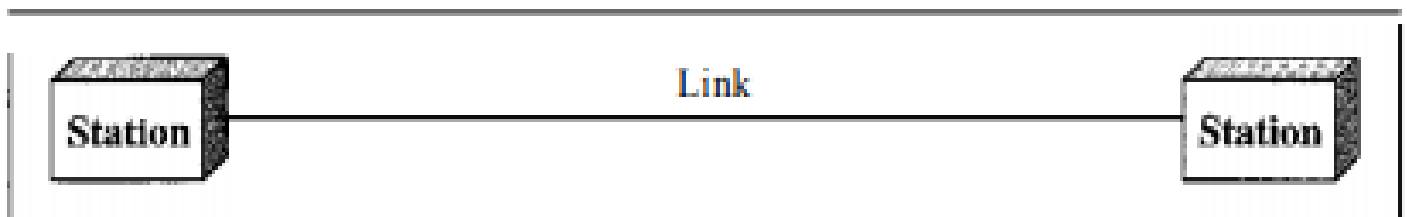
#### b. Half-duplex

- **Full duplex** – both stations can transmit and receive at the same time. Actually, it is two half duplex connections.
  - Telephone network is an example of full-duplex mode, when two people are communicating by a telephone line, both can talk and listen at the same time.
  - The capacity of the channel, must be divided between the two directions.



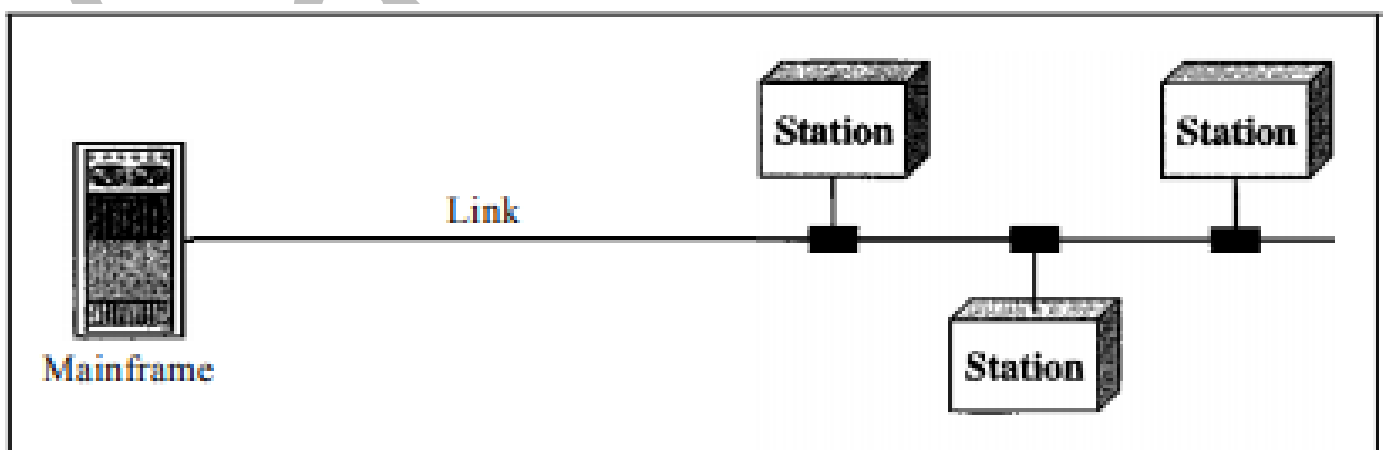
#### c. Full-duplex

- **Network criteria-** a network must be able to meet a certain number of criteria. The most important of these are performance, reliability and security.
  - **Performance** – can be measured in many ways including transit time, response time, number of users, type of transmission medium, capabilities of connected hardware's and efficiency of software.
  - **Reliability** – is a measure of frequency of failure and the time taken to resolve from the failure.
  - **Security** – includes protecting data from unauthorised access, protecting data from damage and development.
- **Physical structure**
- **Types of connection-**
  - **Point to point-** A point-to-point connection provides a dedicated link between two devices. Most point-to-point connections use an actual length of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible.



a. Point-to-point

- **Multipoint** - A multipoint (also called multidrop) connection is one in which more than two specific devices share a single link.



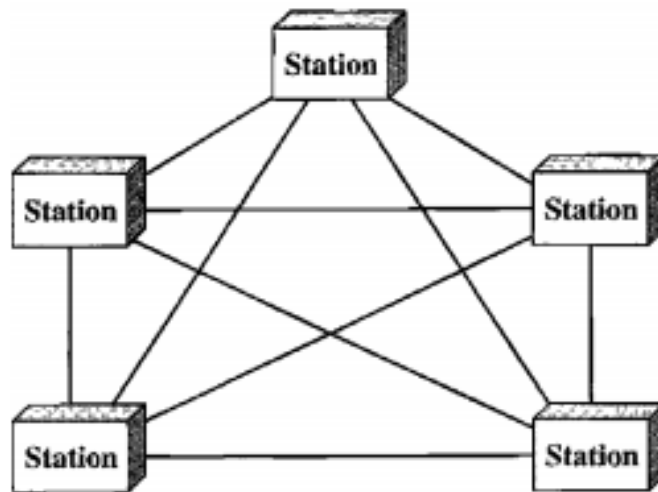
b. Multipoint

# Topology

- Physical topology-refers to the way in which a network is laid out physically. topology of a network is the geometric representation of the relationship of all the links and linking devices to one another.

## Mesh Topology

- In a mesh topology, every device has a dedicated point-to-point link to every other device.
- In mesh topology, we need  $n(n-1)/2$ , duplex-mode links, where n is number of nodes.



### Advantages

- The use of dedicated links guarantees that each connection can carry its own data load, thus eliminating the traffic problems that can occur when links must be shared by multiple devices.
- A mesh topology is robust. If one link becomes unusable, it does not incapacitate the entire system.
- There is the advantage of privacy or security. When every message travel along a dedicated line, only the intended recipient sees it. Physical boundaries prevent other users from gaining access to messages.
- Point-to-point links make fault identification and fault isolation easy. Traffic can be routed to avoid links with suspected problems.

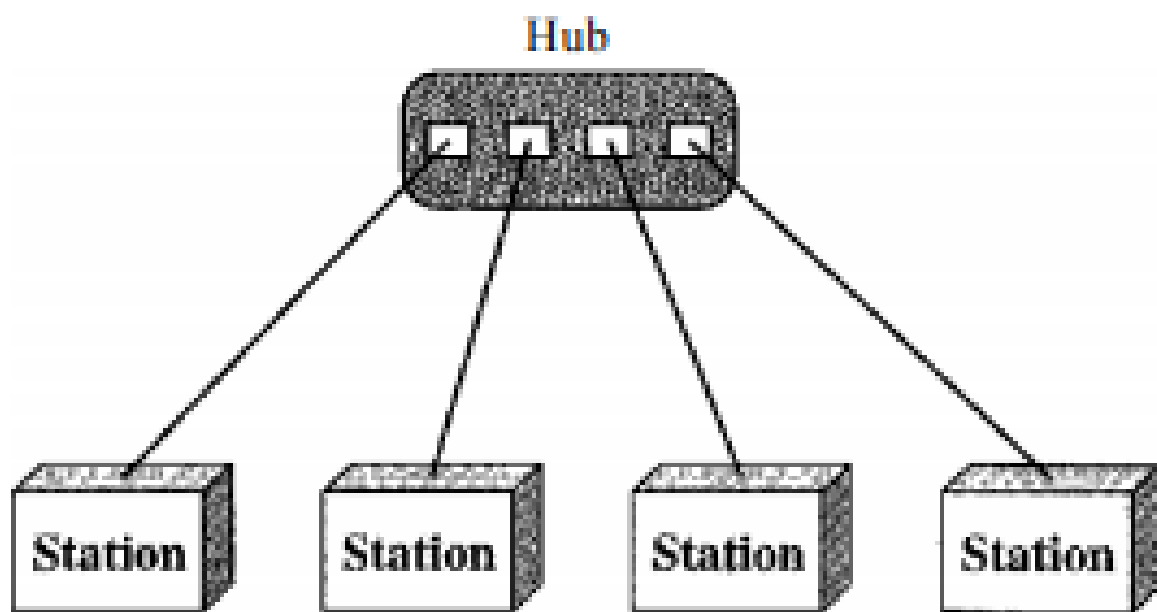
### Disadvantage

- Since every device must be connected to every other device, installation and reconnection are difficult.

- The sheer bulk of the wiring can be greater than the available space (in walls, ceilings, or floors) can accommodate.
- The hardware required to connect each link (I/O ports and cable) can be prohibitively expensive.

## Star Topology

- In a star topology, each device has a dedicated point-to-point link only to a central controller, usually called a hub. The devices are not directly linked to one another.
- The controller acts as an exchange: If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device.



### Advantages

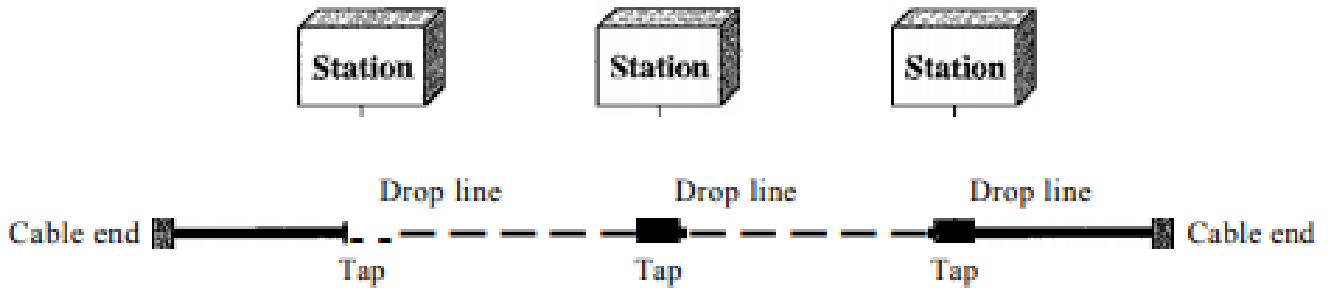
- A star topology is less expensive than a mesh topology.
- It is easy to install and reconfigure and less costly.
- It is robust. If one link fails, only that link is affected.
- Easy fault identification and fault isolation.

### Disadvantage

- Dependency of the whole topology on one single point, the hub. If the hub goes down, the whole system is dead.
- Often more cabling is required in a star than in some other topologies.

## Bus Topology

- A bus topology, is multipoint. One long cable acts as a backbone to link all the devices in a network.



- Nodes are connected to the bus cable by drop lines and taps.
- A drop line is a connection running between the device and the main cable.
- A tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a contact with the metallic core.

### Advantages

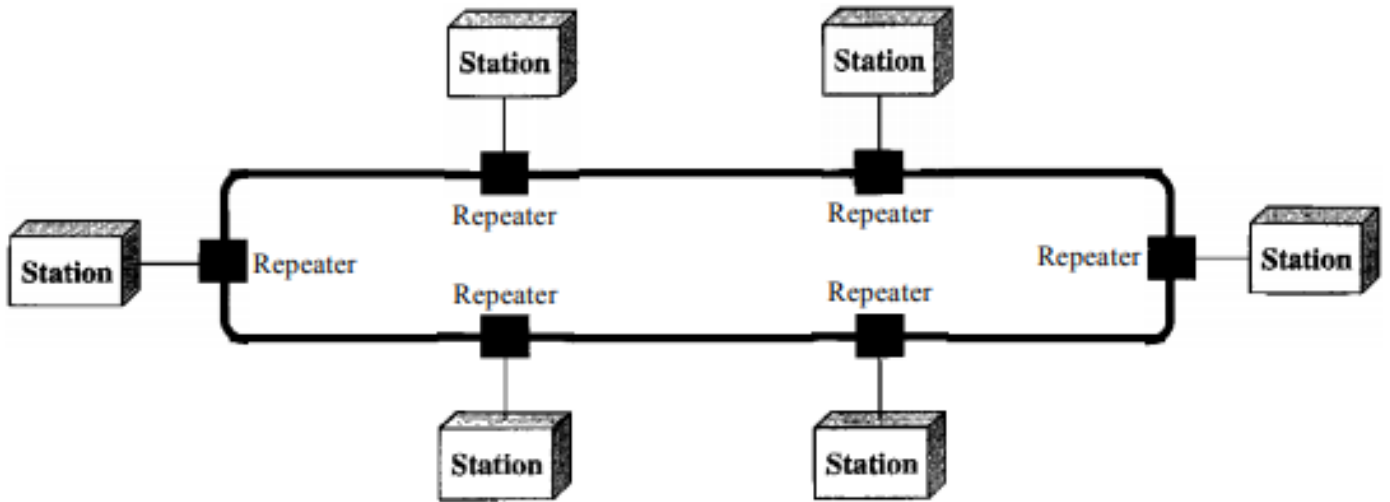
- Advantages of a bus topology include ease of installation.
- Uses less cabling than mesh or star topologies.

### Disadvantage

- Disadvantages include difficult reconnection and fault isolation.
- Difficult to add new devices to network.
- A fault or break in the bus cable stops all transmission.

## Ring Topology

- In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it.
- A signal is passed along the ring in one direction, from device to device, until it reaches its destination. Each device in the ring incorporates a repeater.
- When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along.



### Advantages

- A ring is relatively easy to install and reconfigure.
- Fault isolation is simplified.

### Disadvantages

- A break in the ring (such as a disabled station) can disable the entire network.

### • Categories of networks-

- LAN
- WAN
- MAN

### • Interconnection of network- The Internet

### • Protocols and Standards-

- Syntax
- Semantics
- Timing
- De facto, De jure

## Network Models

- Layered Task- **International standard organization (ISO)** - open system interconnection (OSI) model- allows two system to communicate regardless of their architecture. - which has seven layers with following duties
- An open system is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture.
- The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software.
- The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable.
- It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network.

---

**Application**

---

---

**Presentation**

---

---

**Session**

---

---

**Transport**

---

---

**Network**

---

---

**Data link**

---

---

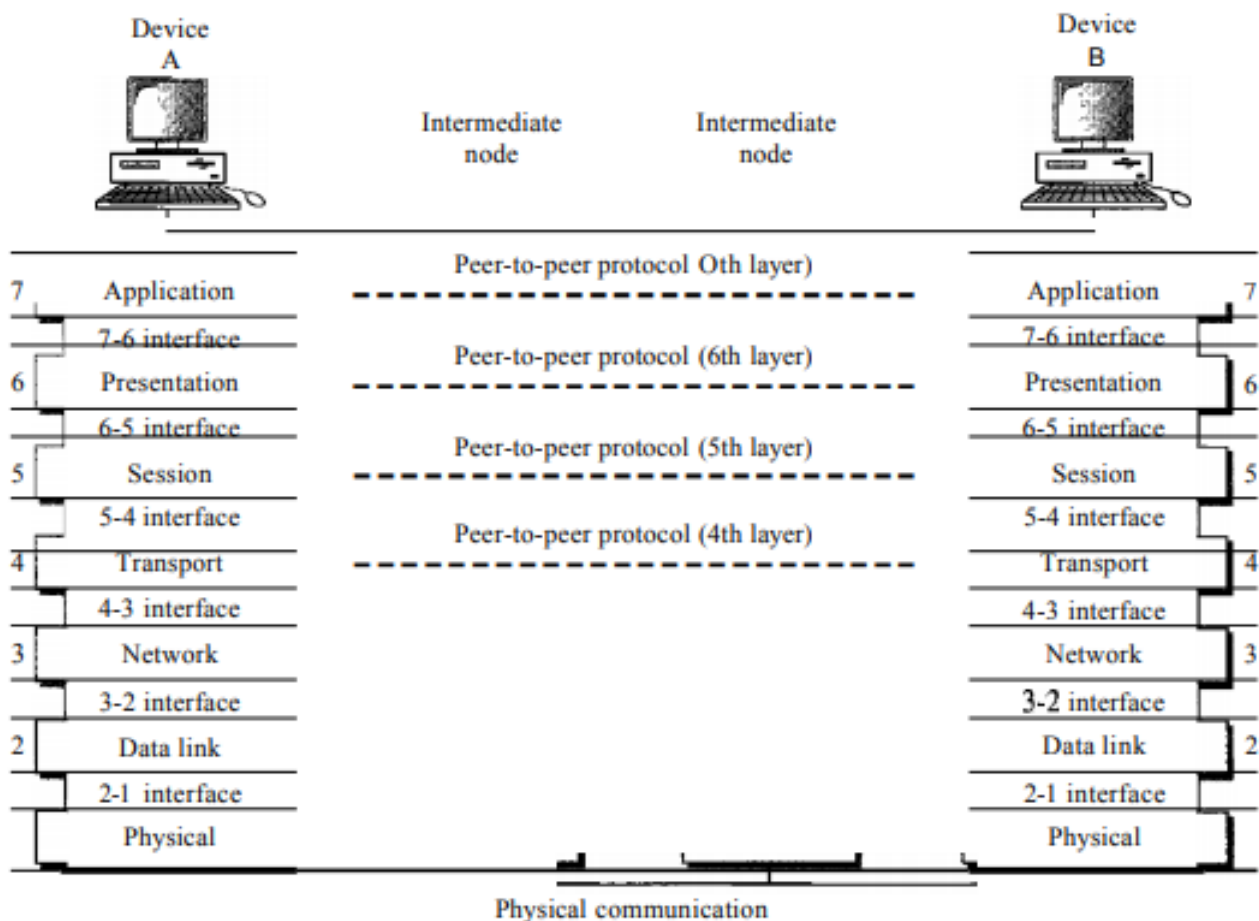
**Physical**

---

---

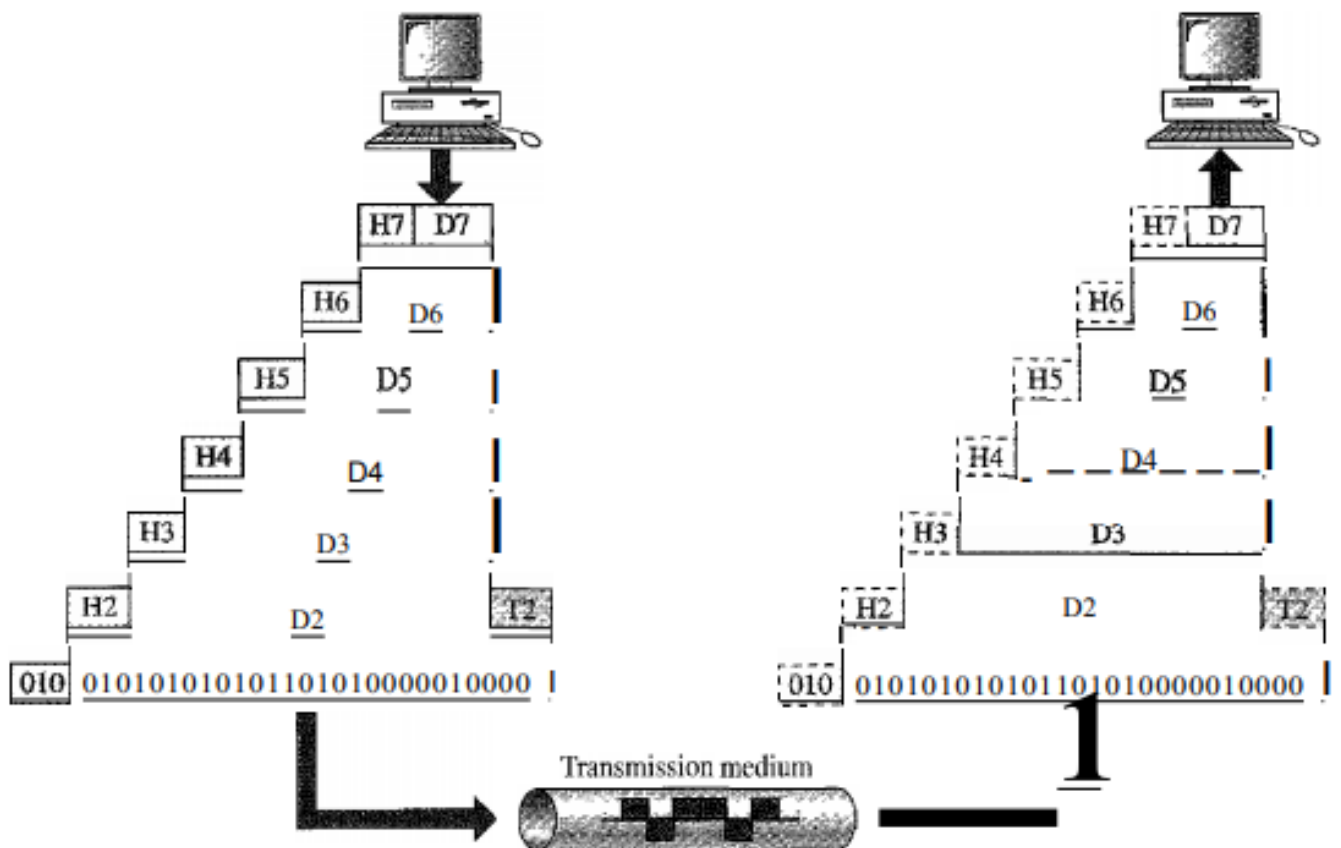
## Layered Architecture

- The OSI model is composed of seven ordered layers: physical (layer 1), data link (layer 2), network (layer 3), transport (layer 4), session (layer 5), presentation (layer 6), and application (layer 7).
- Within a single machine, each layer calls upon the services of the layer just below it. Layer 3, for example, uses the services provided by layer 2 and provides services for layer 4.
- Between machines, layer x on one machine communicates with layer x on another machine.
- This communication is governed by an agreed-upon series of rules and conventions called protocols.
- The processes on each machine that communicate at a given layer are called peer-to-peer processes.



## Peer-to-Peer Processes

- At the physical layer, communication is direct: device A sends a stream of bits to device B (through intermediate nodes).
- At the higher layers, communication must move down through the layers on device A, over to device B, and then back up through the layers.
- Each layer in the sending device adds its own information to the message it receives from the layer just above it and passes the whole package to the layer just below it.
- At layer 1 the entire package is converted to a form that can be transmitted to the receiving device.
- At the receiving machine, the message is unwrapped layer by layer, with each process receiving and removing the data meant for it.



## Physical layer

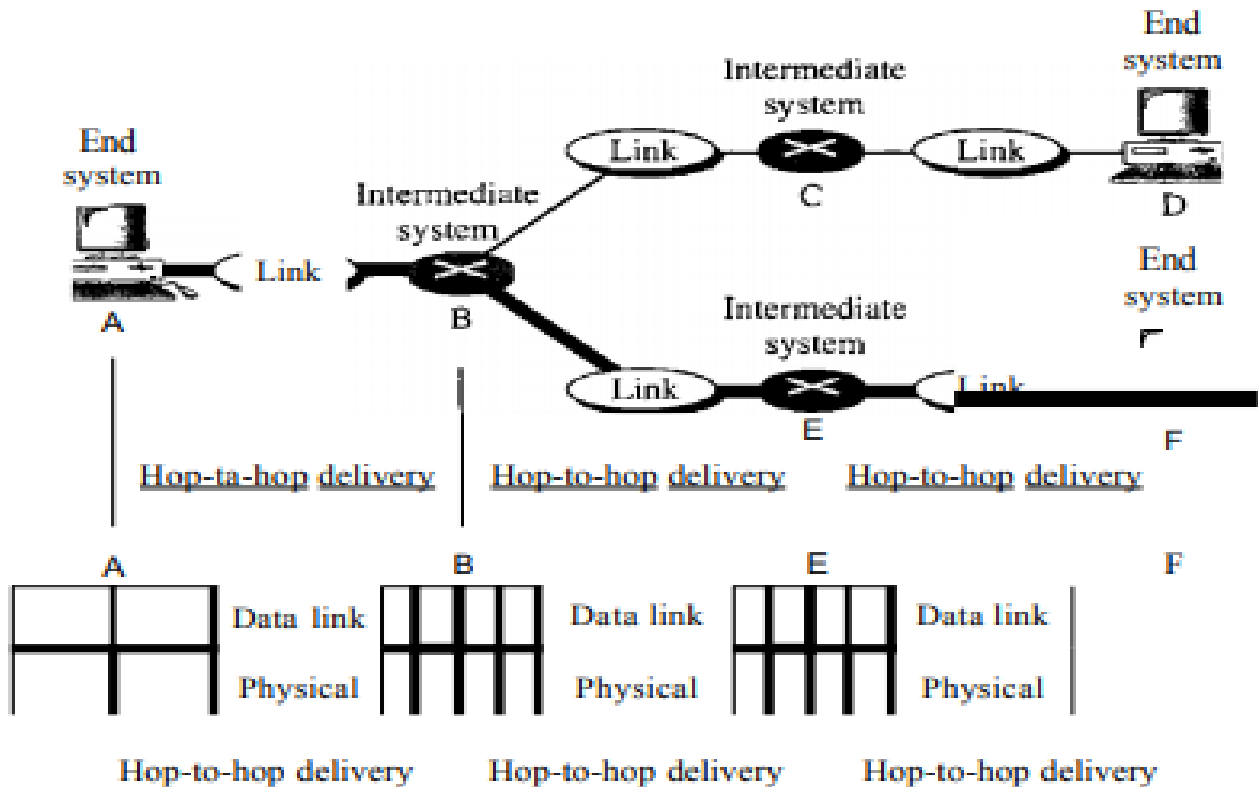
- The physical layer is also concerned with the following:
  - The physical layer defines the **characteristics of the interface** between the devices and the transmission medium.
  - **Representation of bits.** The physical layer data consists of a stream of bits (sequence of 0s or 1s) with no interpretation. To be transmitted, bits must be encoded into signals- electrical or optical.
  - **Data rate.** The transmission rate-the number of bits sent each second-is also defined by the physical layer.
  - **Line configuration.** The physical layer is concerned with the connection of devices to the media.
  - **Physical topology.** The physical topology defines how devices are connected to make a network.
  - **Transmission mode.** The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex.

## Data link layer

- The data link layer transforms the physical layer, a raw transmission facility, to a reliable link.
- It makes the physical layer appear error-free to the upper layer (network layer)

### Main Responsibilities

- **Framing.** The data link layer divides the stream of bits received from the network layer into manageable data units called frames.
- **Physical addressing.** If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame.
- **Flow control.** If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.
- **Error control.** The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.



- **Access control.** When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

## Network layer

- The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links).
- **Logical addressing.** 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, among other things, includes the logical addresses of the sender and receiver.
- **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. One of the functions of the network layer is to provide this mechanism.

## Transport layer

- The transport layer is responsible for process-to-process delivery of the entire message.
- A process is an application program running on a host. Whereas the network layer oversees source-to-destination delivery of individual packets, it does not recognize any relationship between those packets.

### Other Responsibilities

- **Service-point addressing.** The transport layer header must include a type of address called a service-point address (or port address). The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process on that computer.
- **Segmentation and reassembly.** A message is divided into transmittable segments, with each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in transmission.
- **Connection control.** The transport layer can be either connectionless or connection oriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection-oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated.
- **Flow control.** Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.
- **Error control.** Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed process-to process rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication). Error correction is usually achieved through retransmission.

## Session layer

- The session layer is the network dialog controller. It establishes, maintains, and synchronizes the interaction among communicating systems.
- The session layer is responsible for dialog control and synchronization.
- **Dialog control.** The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either half duplex (one way at a time) or full-duplex (two ways at a time) mode.
- **Synchronization.** The session layer allows a process to add checkpoints, or synchronization points, to a stream of data.

## Presentation layer

- The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems.
- **Translation.** The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on. The information must be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver-dependent format.
- **Encryption.** To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.
- **Compression.** Data compression reduces the number of bits contained in the information. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video.

## Application layer

- The application layer enables the user, whether human or software, to access the network.
- It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management, and other types of distributed information services.

### Services

- **Network virtual terminal.** A network virtual terminal is a software version of a physical terminal, and it allows a user to log on to a remote host. To do so, the application creates a software emulation of a terminal at the remote host. The user's computer talks to the software terminal which, in turn, talks to the host, and vice versa. The remote host believes it is communicating with one of its own terminals and allows the user to log on.
- **File transfer, access, and management.** This application allows a user to access files in a remote host (to make changes or read data), to retrieve files from a remote computer for use in the local computer, and to manage or control files in a remote computer locally.
- **Mail services.** This application provides the basis for e-mail forwarding and storage.
- **Directory services.** This application provides distributed database sources and access for global information about various objects and services.

## TCP/IP Protocol Suite

- For physical and data link layer TCP/IP do not define any specific protocol.
- In network layer – IP, ICMP, IGMP, RARP, ARP
- In transport layer – SCTP, TCP, UDP
- In application layer – SMTP, FTP, HTTP, DNS, SNMP, TELNET

**Q** In the following pairs of OSI protocol layer/sub-layer and its functionality, the **INCORRECT** pair is **(Gate-2014) (1 Marks)**

**(a)** Network layer and Routing

**(b)** Data Link Layer and Bit synchronization

**(c)** Transport layer and End-to-end process communication

**(d)** Medium Access Control sub-layer and Channel sharing

**ANSWER B**

**Q** A multiplexer combines four 100-Kbps channels using a time slot of 2 bits. What is the bit rate? **(NET-JULY-2016)**

**(a)** 100 Kbps

**(b)** 200 Kbps

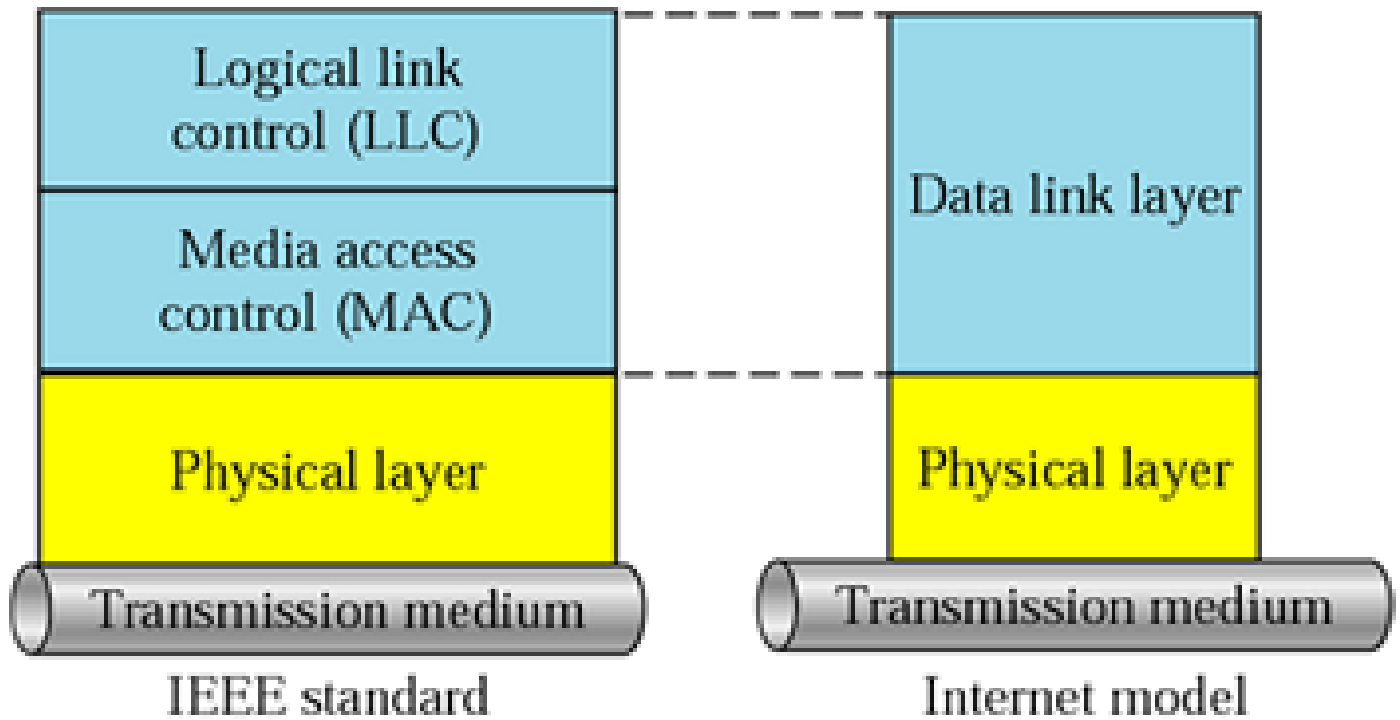
**(c)** 400 Kbps

**(d)** 1000 Kbps

**Ans: c**

## The main services provided by Data Link Layer

- **Error Control**
  - The error needs first to be detected. After detection, it needs to be either corrected at the receiver node or discarded and retransmitted by the sending node.
- **Flow Control**
  - Deals with controlling the flow of frames from sender to receiver.
  - If the rate of produced frames is higher than the rate of consumed frames, frames at the receiving end need to be buffered while waiting to be consumed (processed).
- **Access control**
  - When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.
- **Framing**
  - The data-link layer at each node needs to encapsulate the datagram (packet received from the network layer) in a frame before sending it to the next node.
  - A packet at the data-link layer is normally called a frame.
- **Physical addressing**
  - If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame.



### Two Sublayers

- The IEEE has subdivided the data-link layer into two sublayers: **logical link control (LLC)** (TOP) and **media access control (MAC)** (BOTTOM).
- **Media Access Control (MAC):** It defines the specific access method for each LAN. For example, it defines CSMA/CD as the media access method for Ethernet LANs.
- Flow control, error control, and part of the framing duties are collected into one sublayer called the *logical link control* (LLC).
- Framing is handled in both the LLC sublayer and the MAC sublayer.

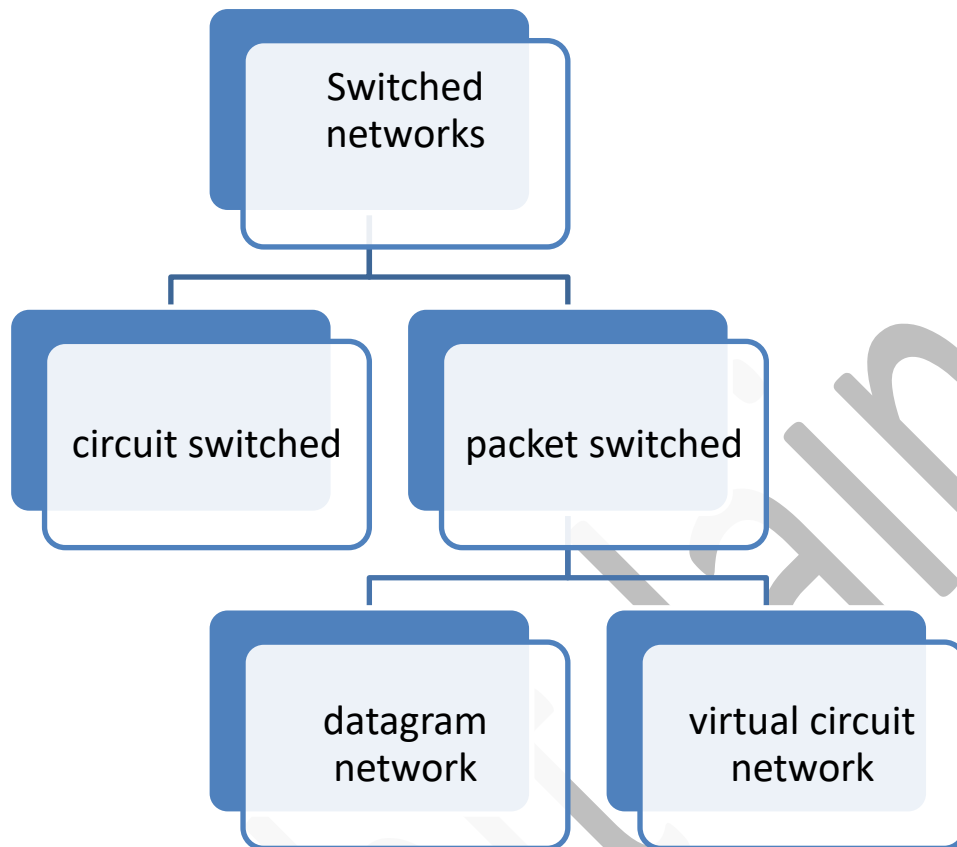
## Physical ADDRESSING (MAC address, Link-Layer address)

- In a connectionless internetwork such as the Internet we cannot make a datagram reach its destination using only IP addresses.
- The reason is that each datagram in the Internet, from the same source host to the same destination host, may take a different path. The source and destination IP addresses define the two ends but cannot define which links the datagram should pass through. Therefore, we need another addressing mechanism, i.e. the link-layer addresses of the two nodes.
- When a datagram passes from the network layer to the data-link layer, the datagram is encapsulated in a frame and two data-link addresses are added to the frame header. These two addresses are changed every time the frame moves from one link to another.

## Connectionless and Connection-Oriented

- A DLC protocol can be either connectionless or connection-oriented.
- **Connectionless Protocol:** Frames are sent from one node to the next without any relationship between the frames; each frame is independent.
- Connectionless means that there is no connection between frames, it does not imply that there is no physical link between nodes.
- **Connection-Oriented Protocol:** A logical connection should first be established between the two nodes (setup phase). After all frames that are somehow related to each other are transmitted (transfer phase), the logical connection is terminated (teardown phase).
- The frames are numbered and sent in order. If they are not received in order, the receiver needs to wait until all frames belonging to the same set are received and then deliver them in order to the network layer.

# Switching



A switched network consists of a series of interlinked nodes, called switches. Switches are the devices capable of creating temporary connections between two or more devices linked to the switch.

1. **Circuit switching** - consist of set switches connected by physical link. A connection between two stations is a dedicated path made of one or more links. A link is divided into  $n$  channels. There are three phases in order setup phase, data transfer phase, tear down phase.
2. **Datagram network** - In data communications, we need to send messages from one end system to another. If the message is going to pass through a packet-switched network, it needs to be divided into packets of fixed or variable size. The size of the packet is determined by the network and the governing protocol. In packet switching, there is no resource allocation for a packet. This means that there is no reserved bandwidth on the links, and there is no scheduled processing time for each packet. Resources are allocated on demand. The allocation is done on a first come, first-served basis. When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed. As with other systems in our daily life, this lack of reservation may create delay. For example, if we do not have a reservation at a

restaurant, we might have to wait. In a datagram network, each packet is treated independently of all others. Even if a packet is part of a multipacket transmission, the network treats it as though it existed alone. Packets in this approach are referred to as datagrams. Packets may also be lost or dropped because of a lack of resources. In most protocols, it is the responsibility of an upper-layer protocol to reorder the datagrams or ask for lost datagrams before passing them on to the application.

3. **Virtual network** - A virtual-circuit network is a cross between a circuit-switched network and a datagram network. It has some characteristics of both.

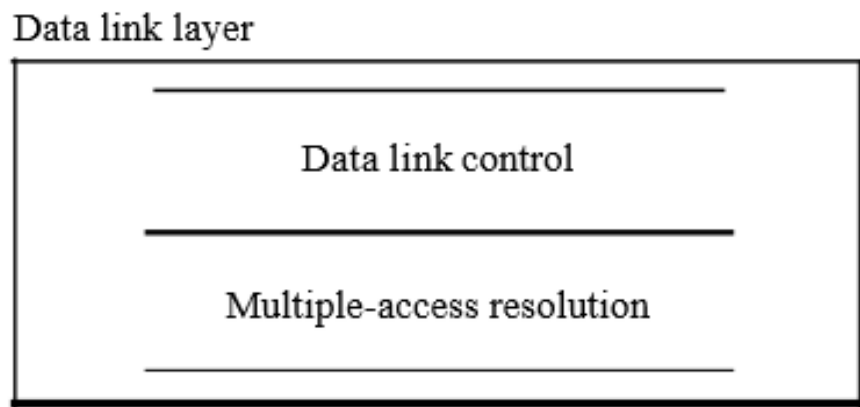
Sanchit Jain

## Multiple access control

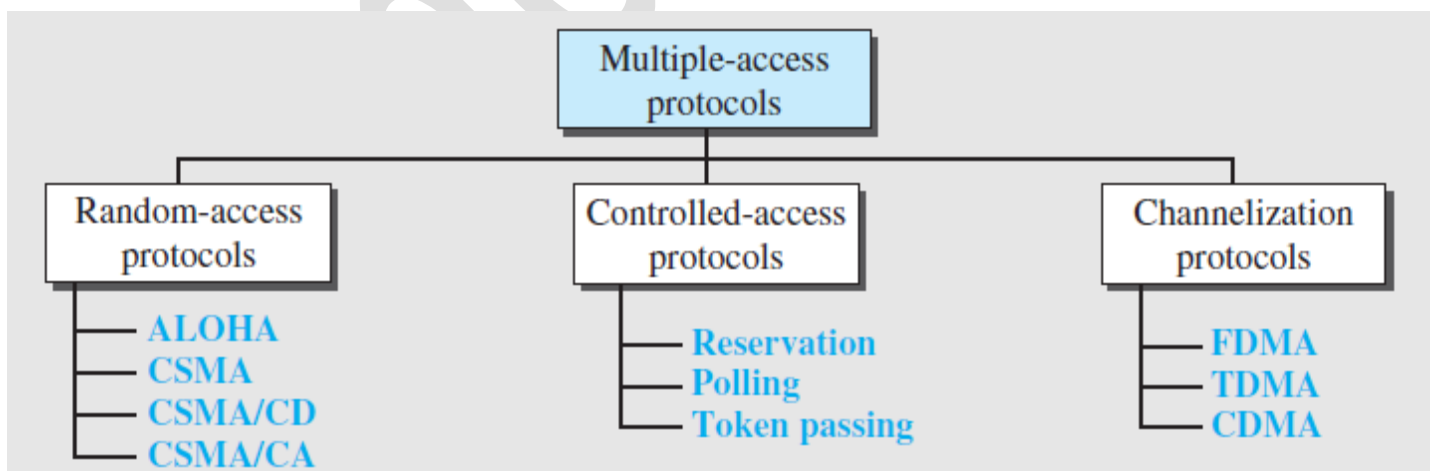
- IEEE has actually made this division for LANs. The upper sublayer that is responsible for flow and error control is called the logical link control (LLC) layer; the lower sublayer that is mostly responsible for multiple access resolution is called the media access control (MAC) layer.

*Data link layer divided into two functionality-oriented sublayers*

---



- When nodes or stations are connected and use a common link, called a *multipoint* or *broadcast link*, we need a multiple-access protocol to coordinate access to the link.
- Many protocols have been devised to handle access to a shared link. All of these protocols belong to a sublayer in the data-link layer called media access control (MAC).

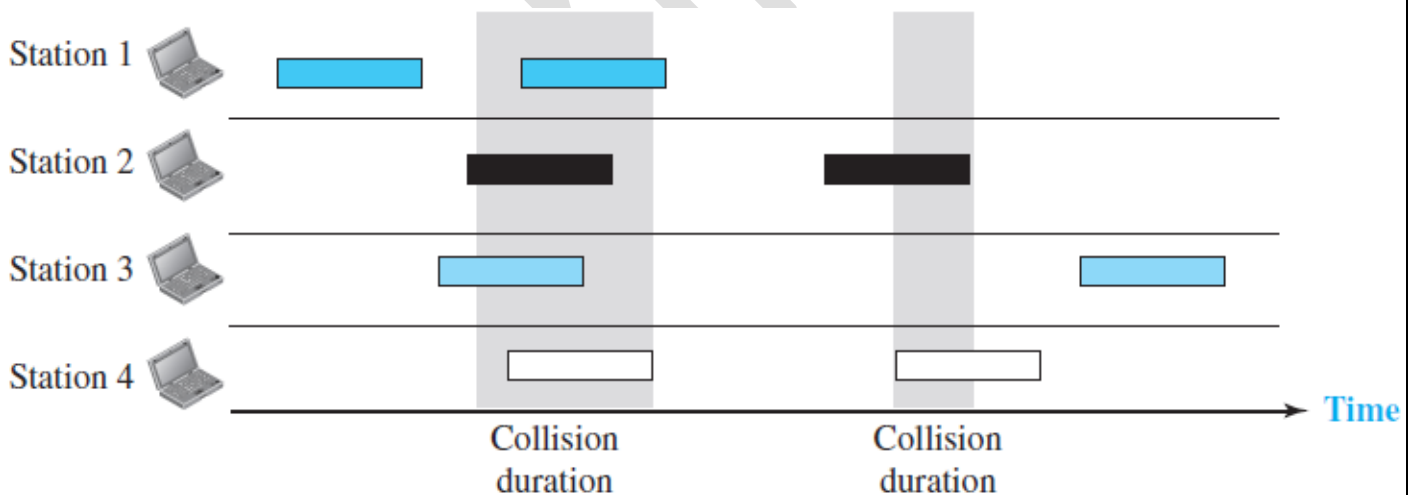


## RANDOM ACCESS

- In random access methods, no station is superior to another station and none is assigned the control over another.
- No station permits, or does not permit, another station to send.
- Two features give this method its name.
  - First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called random access.
  - Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called contention methods.
- However, if more than one station tries to send, there is an access conflict-collision-and the frames will be either destroyed or modified.
  - When can the station access the medium?
  - What can the station do if the medium is busy?
  - How can the station determine the success or failure of the transmission?
  - What can the station do if there is an access conflict?

## Aloha

- Earliest random-access method, was developed at the University of Hawaii around 1970.
- It was designed for a radio (wireless) LAN, but it can be used on any shared medium.
- The original ALOHA protocol is called pure ALOHA. This is a simple, but elegant protocol. The idea is that each station sends a frame whenever it has a frame to send. However, there is the possibility of collision between frames from different stations.
- The pure ALOHA protocol relies on acknowledgments from the receiver. When a station sends a frame, it expects the receiver to send an acknowledgment. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame.
- A collision involves two or more stations. If all these stations try to resend their frames after the time-out, the frames will collide again. Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid more collisions. We call this time the back-off time  $T_B$ .
- Pure ALOHA has a second method to prevent congesting the channel with retransmitted frames. After a maximum number of retransmissions attempts  $K_{max}$  a station must give up and try later.

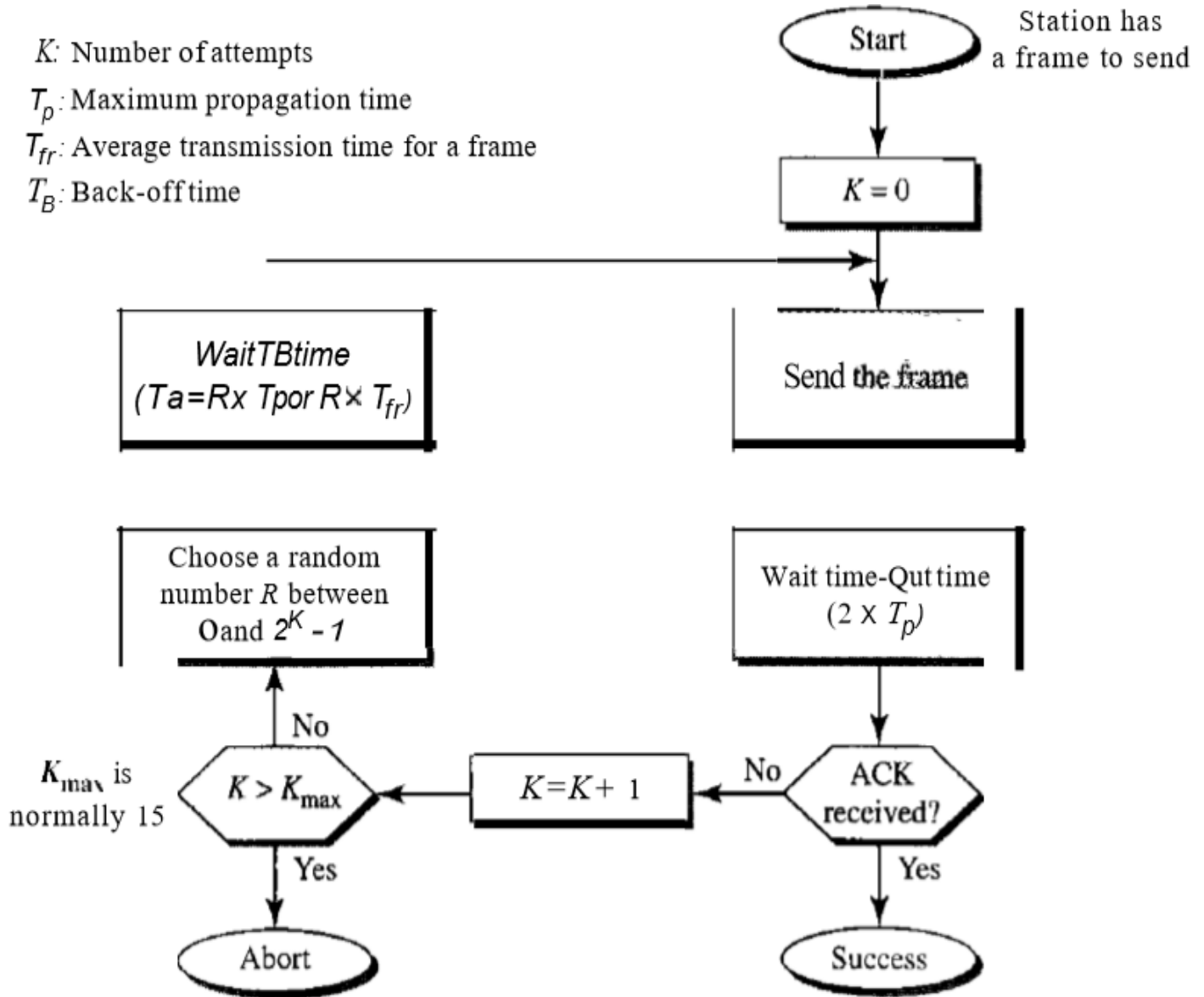


$K$ : Number of attempts

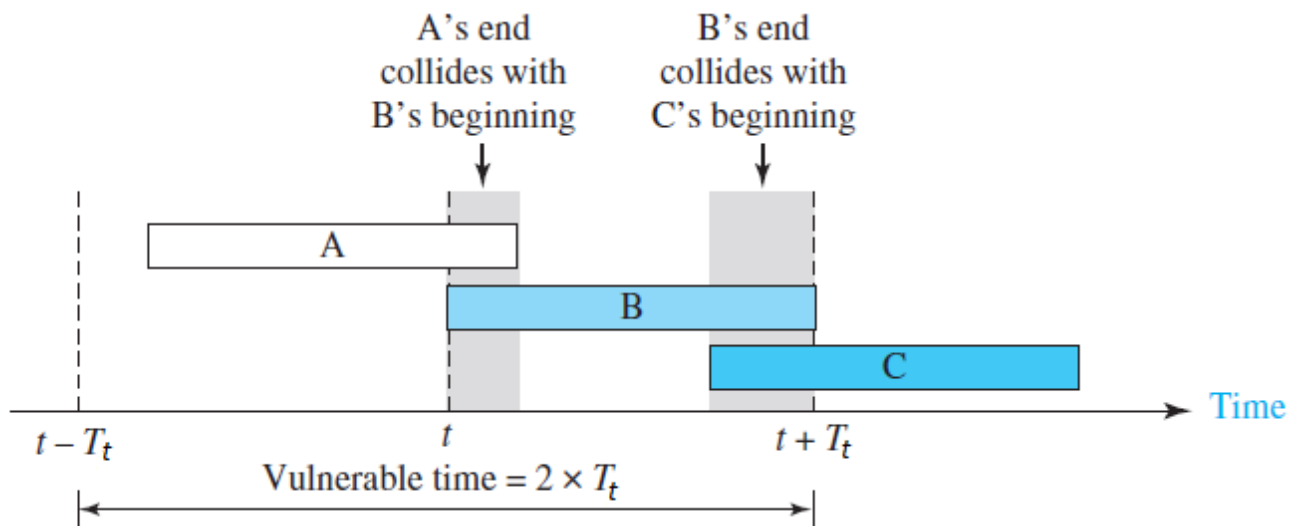
$T_p$ : Maximum propagation time

$T_{fr}$ : Average transmission time for a frame

$T_B$ : Back-off time



- Vulnerable time in which there is a possibility of collision. We assume that the stations send fixed-length frames with each frame taking  $T_{fr}$  S to send.
- Station A sends a frame at time  $t$ . Now imagine station B has already sent a frame between  $t - T_{fr}$  and  $t$ . This leads to a collision between the frames from station A and station B. The end of B's frame collides with the beginning of A's frame.
- On the other hand, suppose that station C sends a frame between  $t$  and  $t + T_{fr}$ . Here, there is a collision between frames from station A and station C. The beginning of C's frame collides with the end of A's frame. we see that the vulnerable time, during which a collision may occur in pure ALOHA, is 2 times the frame transmission time.
- Pure ALOHA vulnerable time =  $2 \times T_{fr}$



### Throughput

- Let,  $G$  be the average number of frames generated by the system during one frame transmission time. Then, average number of successfully transmitted frames for pure ALOHA is  $S$ .
- The throughput for pure ALOHA is  $S = G * e^{-2G}$ .
- The maximum throughput  $S_{max} = 1/(2e) = 0.184$  when  $G = (1/2)$ .

**Example:** The stations on a wireless ALOHA network are a maximum of 600 km apart. If we assume that signals propagate at  $3 \times 10^8$  m/s. Find back off time possibility after two consecutive collision ?

we find  $T_p = (600 \times 10^3) / (3 \times 10^8) = 2$  ms.

after two collision,  $K = 2$ , the range of  $R$  is  $\{0, 1, 2, 3\}$ .

This means that  $T_B$  can be 0, 2, 4, or 6 ms, based on the outcome of the random variable  $R$ .

**Example:** A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Ans: Transmission Time  $T_t = 200 \text{ bits} / 200 * 10^3 = 1$  ms

So, vulnerable time =  $2 * 1 = 2$ ms

**Example:** A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces 1000 frames per second?

**Ans. The frame transmission time is 200/200 kbps or 1 ms.**

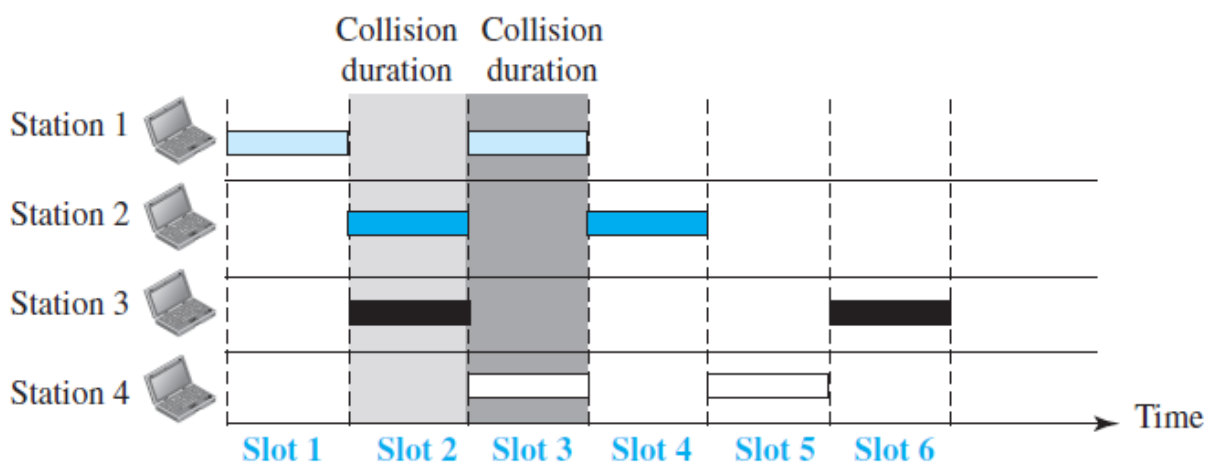
If the system creates 1000 frames per second, or 1 frame per millisecond, then  $G = 1$ .

$S = G \times e^{-2G} = 0.135$  (13.5 percent).

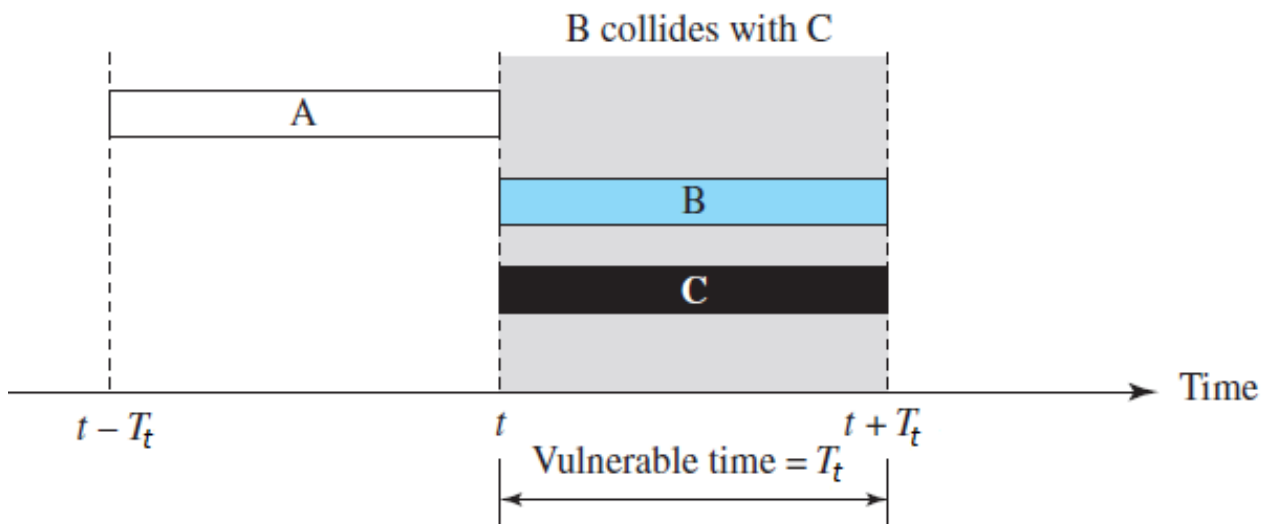
Throughput is  $1000 \times 0.135 = 135$  frames. Only 135 frames out of 1000 will probably survive.

## Slotted ALOHA

- Pure ALOHA has a vulnerable time of  $2 \times T_{fr}$ . This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished.
- Slotted ALOHA was invented to improve the efficiency of pure ALOHA. In slotted ALOHA we divide the time into slots of  $T_{fr}$  s and force the station to send only at the beginning of the time slot.
- Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame.
- Off course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to  $T_{fr}$



- **Slotted ALOHA vulnerable time** =  $T_t$
- **Throughput:** The throughput for slotted ALOHA is  $S = G * e^{-G}$ .
- The maximum throughput  $S_{max} = 0.368$  when  $G = 1$ .



**Q** Consider a LAN with four nodes  $S_1, S_2, S_3$  and  $S_4$ . Time is divided into fixed-size slots, and a node can begin its transmission only at the beginning of a slot. A collision is said to have occurred if more than one node transmits in the same slot. The probabilities of generation of a frame in a time slot by  $S_1, S_2, S_3$  and  $S_4$  are 0.1, 0.2, 0.3 and 0.4, respectively. The probability of sending a frame in the first slot without any collision by any of these four stations is \_\_\_\_\_ . (Gate-2015) (2 Marks)

- (A) 0.462                      (B) 0.711                      (C) 0.5                      (D) 0.652

**Answer: (A)**

**Q** There are  $n$  stations in a slotted LAN. Each station attempts to transmit with a probability  $p$  in each time slot. What is the probability that ONLY one station transmits in a given time slot? (Gate-2007) (2 Marks)

- a)  $np(1-p)^{n-1}$                       b)  $(1-p)^{n-1}$                       c)  $p(1-p)^{n-1}$                       d)  $1-(1-p)^{n-1}$

**ANSWER A**

**Q** A and B are the only two stations on an Ethernet. Each has a steady queue of frames to send. Both A and B attempt to transmit a frame, collide, and A wins the first backoff race. At the end of this successful transmission by A, both A and B attempt to transmit and collide. The probability that A wins the second backoff race is: (Gate-2004) (2 Marks)

- (A) 0.5                      (B) 0.625                      (C) 0.75                      (D) 1.0

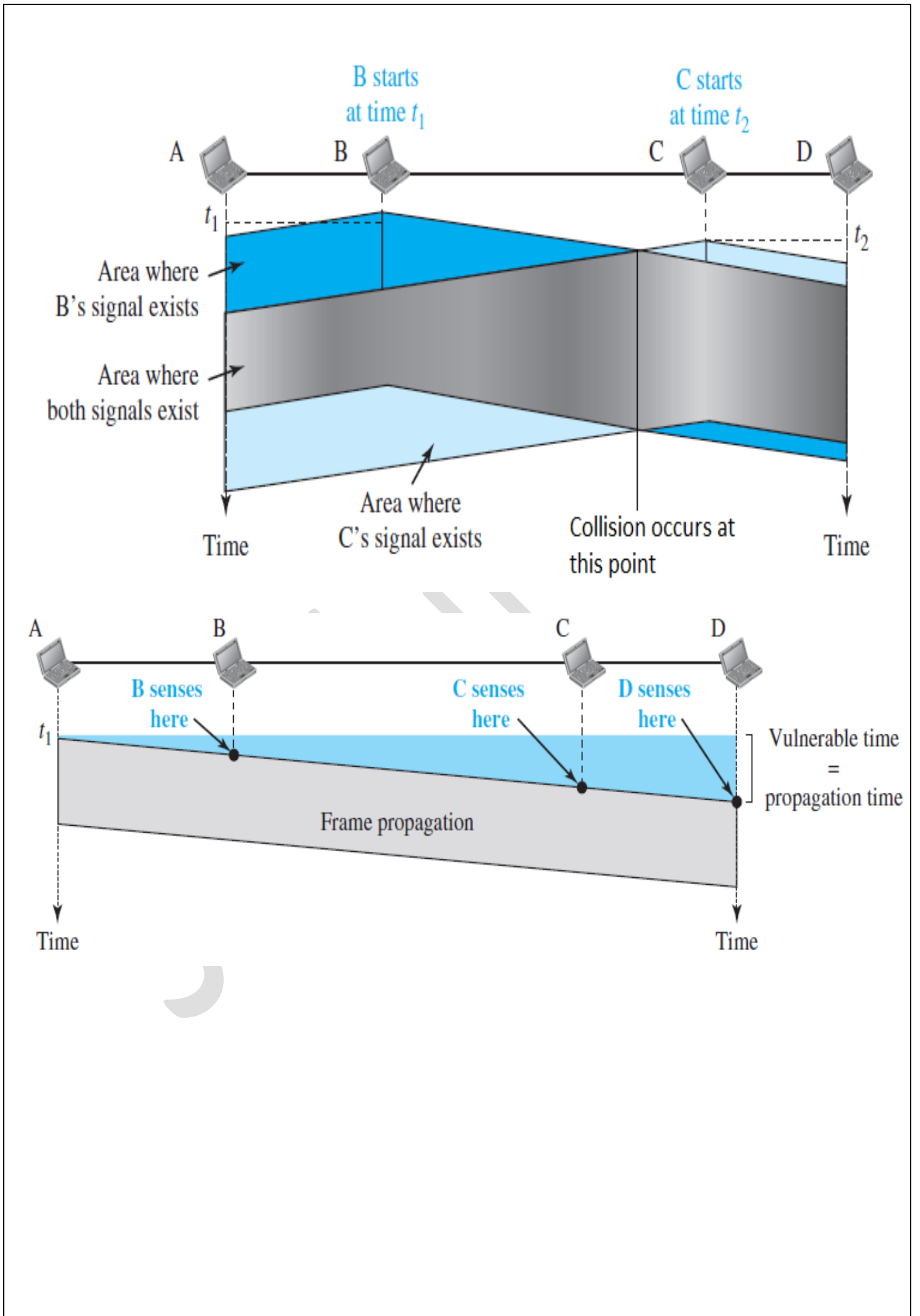
**Answer: (B)**

## Carrier Sense Multiple Access (CSMA)

- To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it.
- Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, CSMA is based on the principle "sense before transmit" or "listen before talk." CSMA can reduce the possibility of collision, but it cannot eliminate it.
- The possibility of collision still exists because of propagation delay; when a station sends a frame, it still takes time (although very short) for the first bit to reach every station and for every station to sense it.
- In other words, a station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received. At time  $t_1$  station B senses the medium and finds it idle, so it sends a frame. At time  $t_2$  ( $t_2 > t_1$ ) station C senses the medium and finds it idle because, at this time, the first bits from station B have not reached station C. Station C also sends a frame. The two signals collide and both frames are destroyed.

### Vulnerable Time

- The vulnerable time for CSMA is the **propagation time**  $T_p$ .
- When a station sends a frame and any other station tries to send a frame during this time, a collision will result.
- But if the first bit of the frame reaches the end of the medium, every station will already have heard the bit and will refrain from sending.
- Station A has sent a frame at time  $t_1$ , which reaches the rightmost station, D, at time  $t_1 + T_p$ .
- The gray area is vulnerable time.

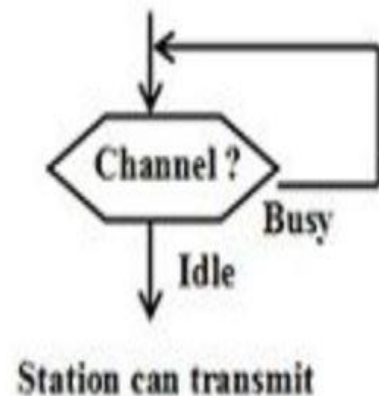
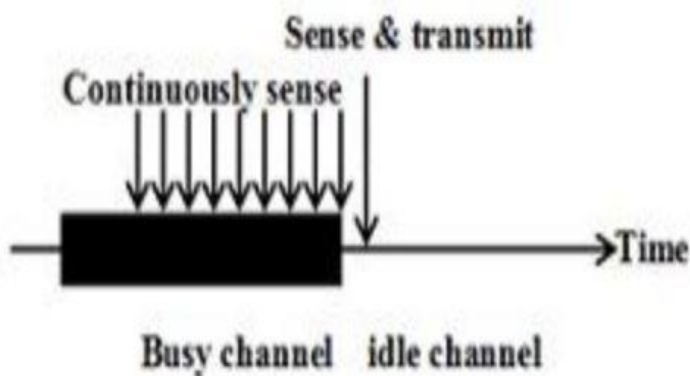


- **Persistence Methods**

- What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions: the I-persistent method, the nonpersistent method, and the p-persistent method.

- **I-Persistent**

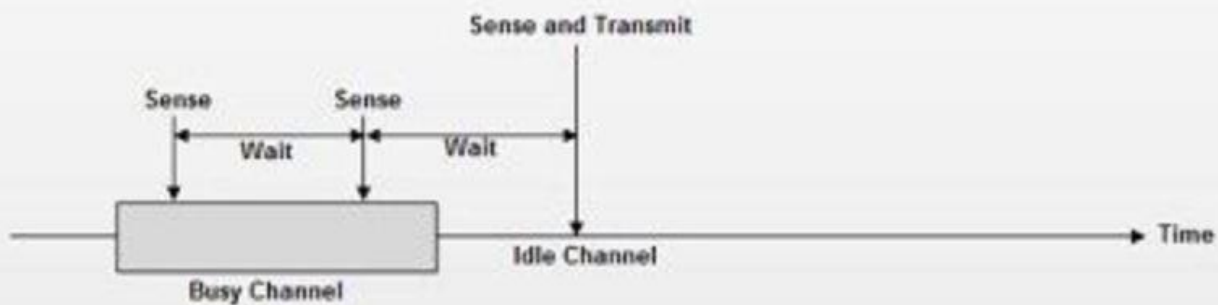
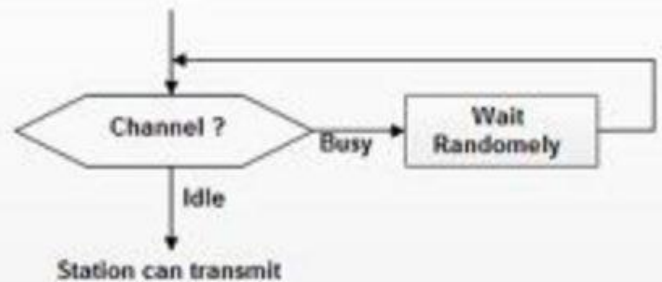
- The I-persistent method is simple and straightforward. In this method, after the station finds the line idle, it sends its frame immediately (with probability 1). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.



1-persistent CSMA

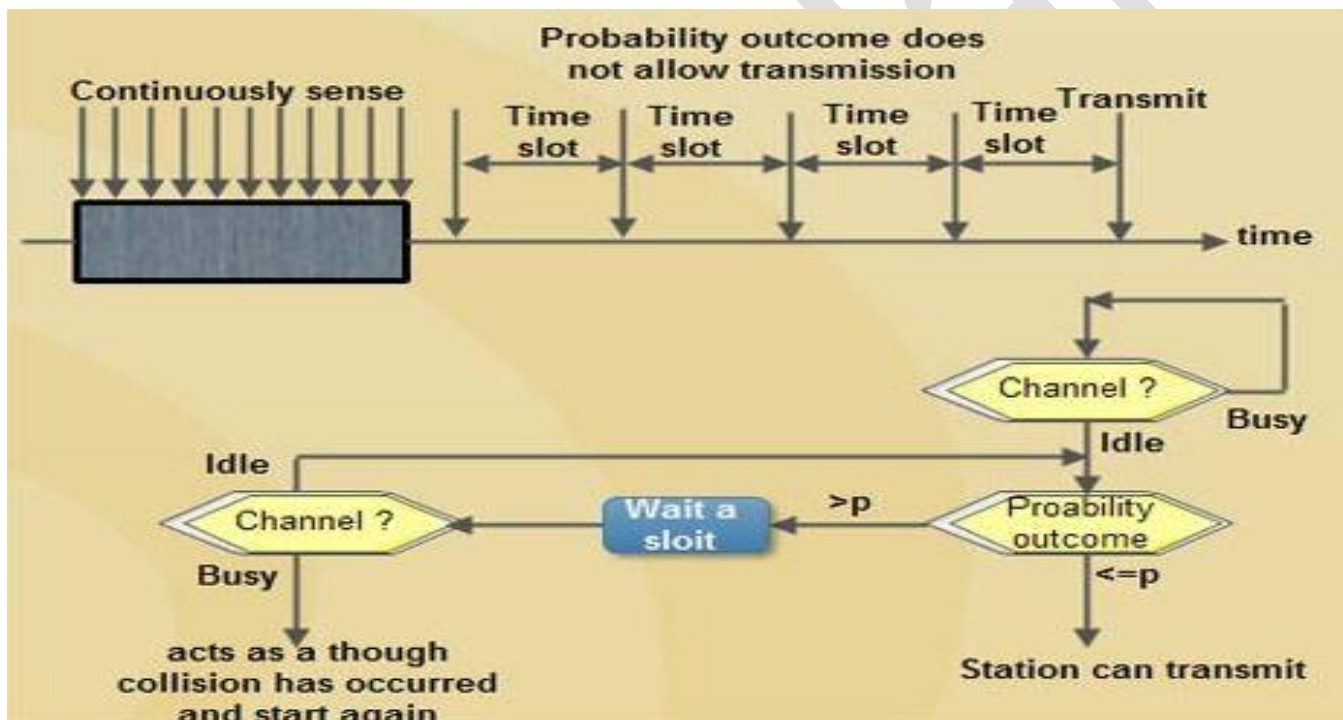
- **Nonpersistent**

- In the nonpersistent method, a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again.
- The nonpersistent approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously. However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.



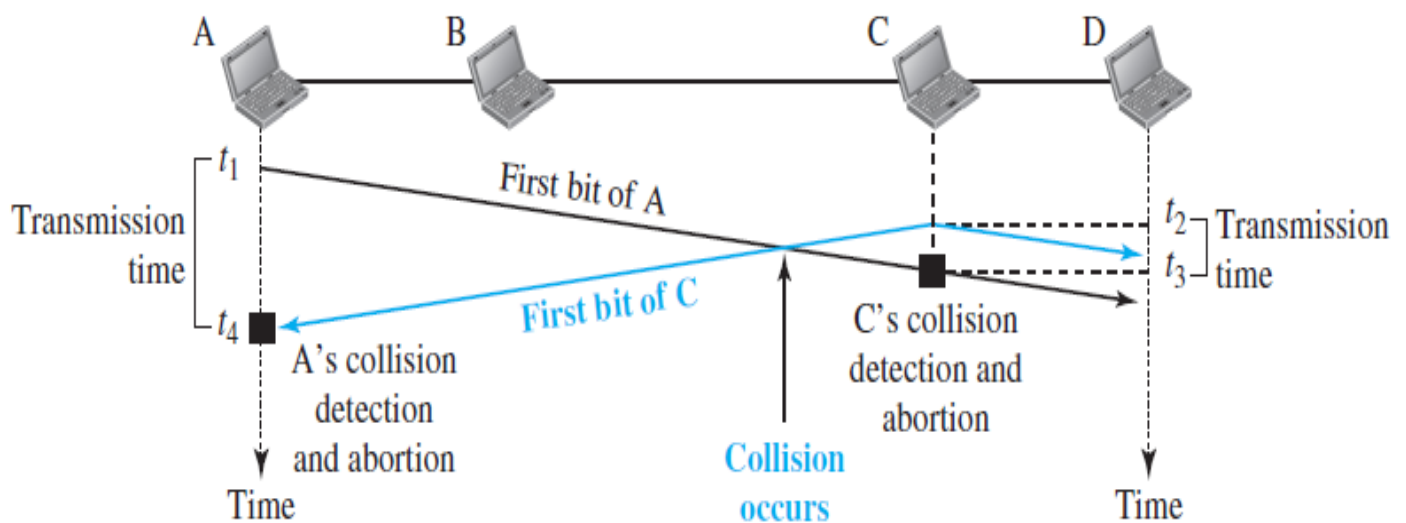
- **p-Persistent**

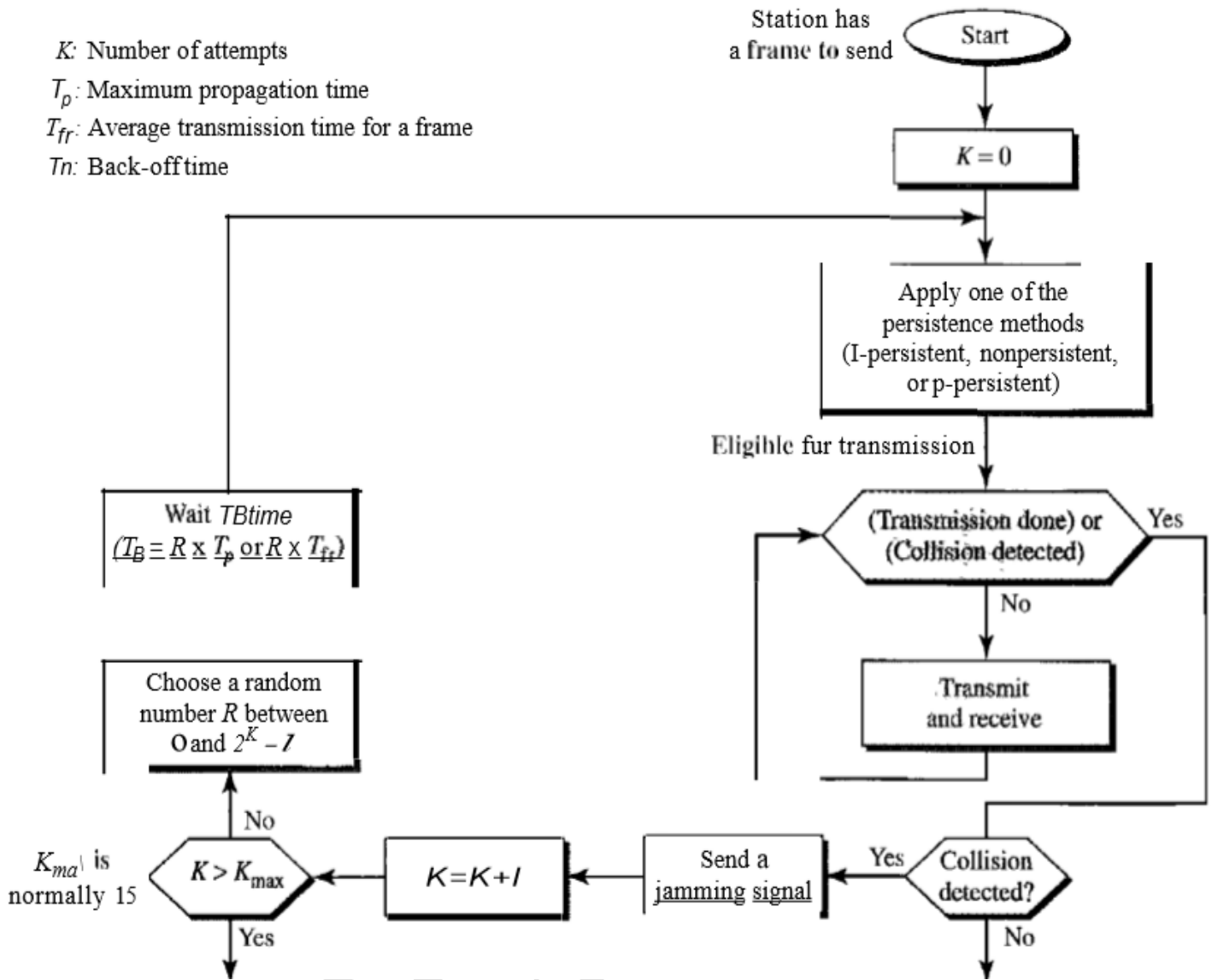
- The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time.
- The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency. In this method, after the station finds the line idle it follows these steps:
  - With probability  $p$ , the station sends its frame.
  - With probability  $q = 1 - p$ , the station waits for the beginning of the next time slot and checks the line again.
    - a. If the line is idle, it goes to step 1.
    - b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.



## Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

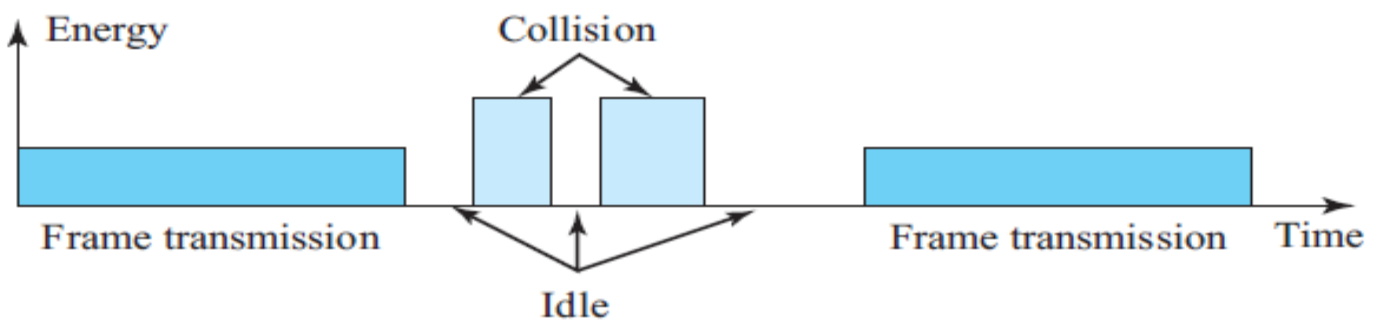
- In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again.
- Minimum Frame Size - For CSMA / CD to work, we need a restriction on the frame size. Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission.
- This is so because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection. Therefore, the frame transmission time  $T_{fr}$  must be at least two times the maximum propagation time  $T_p$ .
- To understand the reason, let us think about the worst-case scenario. If the two stations involved in a collision are the maximum distance apart, the signal from the first takes time  $T_p$  to reach the second, and the effect of the collision takes another time  $T_p$  to reach the first. So the requirement is that the first station must still be transmitting after  $2T_p$ .





• **Energy Level**

- We can say that the level of energy in a channel can have three values: zero, normal, and abnormal. At the zero level, the channel is idle. At the normal level, a station has successfully captured the channel and is sending its frame.
- At the abnormal level, there is a collision and the level of the energy is twice the normal level. A station that has a frame to send or is sending a frame needs to monitor the energy level to determine if the channel is idle, busy, or in collision mode.



**Example:** A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time is 25.6  $\mu$ s, what is the minimum size of the frame?

Ans: The minimum frame transmission time is  $T_t = 2 \times T_p = 51.2 \mu$ s.

The minimum size of the frame = 10 Mbps  $\times$  51.2  $\mu$ s =  $10 \times 10^6 \times 51.2 \times 10^{-6} = 512$  bits or 64 bytes.

$$T_{fr} = L/B$$

$T_{fr}$  is the transmission time or forwarding time

L is the length of the frame in bits

B is the bandwidth of the channel in bits

$$T_p = D/S$$

$T_{fr}$  is the time taken by frame to travel from one end point to another of access medium

L is the length of the frame in bits

B is the bandwidth of the channel in bits per second

for CSMA/CD

$$T_{fr} = 2 * T_p$$

$$L = 2 * T_p * B$$

**Q** A network has a data transmission bandwidth of  $20 \times 10^6$  bits per second. It uses CSMA/CD in the MAC layer. The maximum signal propagation time from one node to another node is 40 microseconds. The minimum size of a frame in the network is \_\_\_\_\_ bytes. **(Gate-2016) (2 Marks)**

**ANSWER 200**

**Q** Consider a CSMA/CD network that transmits data at a rate of 100 Mbps ( $10^8$  bits per second) over a 1 km (kilometre) cable with no repeaters. If the minimum frame size required for this network is 1250 bytes, what is the signal speed (km/sec) in the cable? (Gate-2015) (1 Marks)

- (A) 8000                      (B) 10000                      (C) 16000                      (D) 20000

**Answer: (D)**

**Q** A network with CSMA/CD protocol in the MAC layer is running at 1 Gbps over a 1 km cable with no repeaters. The signal speed in the cable is  $2 \times 10^8$  m/sec. The minimum frame size for this network should be (Gate-2005) (2 Marks)

- (A) 10000 bits                      (B) 10000 bytes                      (C) 5000 bits                      (D) 5000 bytes

**Answer: (A)**

**Q** A 2 km long broadcast LAN has  $10^7$  bps bandwidth and uses CSMA/CD. The signal travels along the wire at  $2 \times 10^8$  m/s. What is the minimum packet size that can be used on this network? (Gate-2003) (2 Marks)

- (A) 50 bytes                      (B) 100 bytes                      (C) 200 bytes                      (D) None of these

**Answer: (D)**

**Q** The minimum frame size required for a CSMA/CD based computer network running at 1 Gbps on a 200m cable with a link speed of  $2 \times 10^8$  m/s is (Gate-2008) (2 Marks)

- (A) 125 bytes                      (B) 250 bytes                      (C) 500 bytes                      (D) None of these

**Answer: (B)**

**Q** Consider a simple communication system where multiple nodes are connected by a shared broadcast medium (like Ethernet or wireless). The nodes in the system use the following carrier-sense based medium access protocol. A node that receives a packet to transmit will carrier-sense the medium for 5 units of time. If the node does not detect any other transmission in this duration, it starts transmitting its packet in the next time unit. If the node detects another transmission, it waits until this other transmission finishes, and then begins to carrier-sense for 5 time units again. Once they start to transmit, nodes do not perform any collision detection and continue transmission even if a collision occurs. All transmissions last for 20 units of time. Assume that the transmission signal travels at the speed of 10 meters per unit time in the medium. Assume that the system has two nodes P and Q, located at a distance  $d$  meters from each other. P starts transmitting a packet at time  $t=0$  after successfully completing its carrier-sense phase. Node Q has a packet to

transmit at time  $t=0$  and begins to carrier-sense the medium. The maximum distance  $d$  (in meters, rounded to the closest integer) that allows Q to successfully avoid a collision between its proposed transmission and P's ongoing transmission is \_\_\_\_\_. (Gate-2018) (2 Marks)

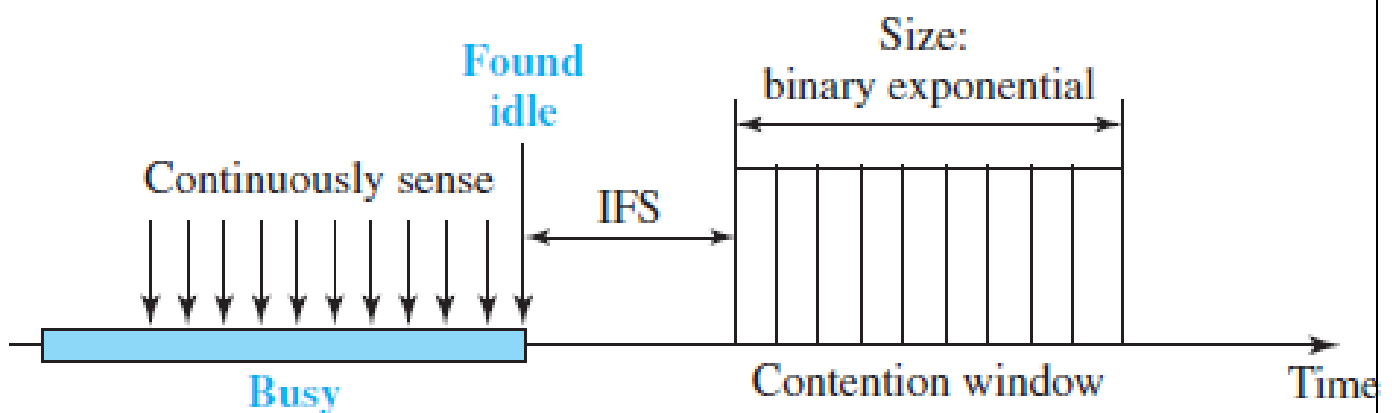
**Q** Which of the following statements is TRUE about CSMA/CD (Gate-2005) (1 Marks)

- (A) IEEE 802.11 wireless LAN runs CSMA/CD protocol
- (B) Ethernet is not based on CSMA/CD protocol
- (C) CSMA/CD is not suitable for a high propagation delay network like satellite network
- (D) There is no contention in a CSMA/CD network

**Answer: (C)**

# Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

- In a wireless network, much of the sent energy is lost in transmission. The received signal has very little energy. Therefore, a collision may add only 5 to 10 percent additional energy. This is not useful for effective collision detection.
- We need to avoid collisions on wireless networks because they cannot be detected. Carrier sense multiple access with collision avoidance (CSMA / CA) was invented for this network. Collisions are avoided through the use of CSMA / CA
- three strategies: the interframe space, the contention window, and acknowledgment



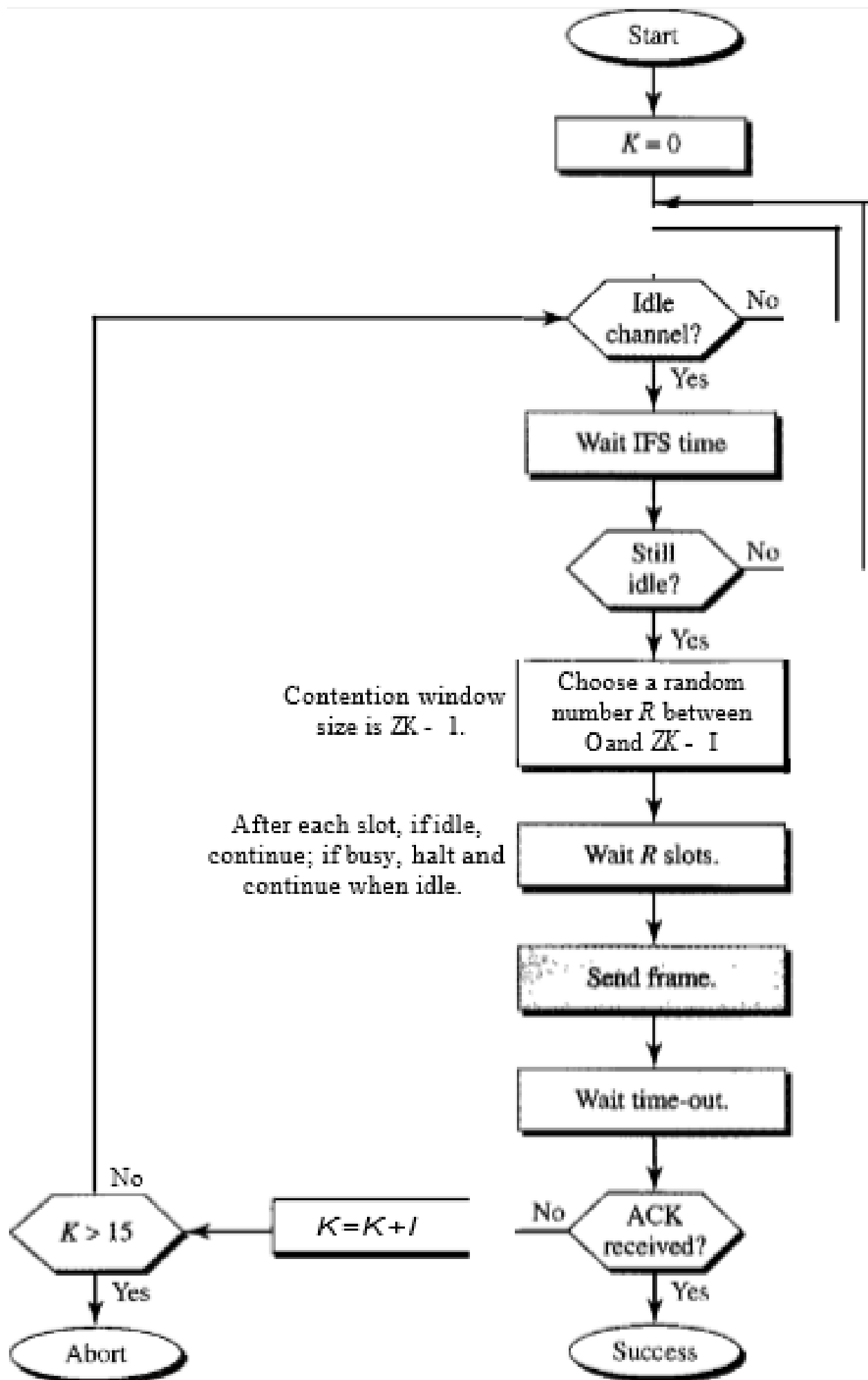
- **Interframe Space (IFS)**
  - First, collisions are avoided by deferring transmission even if the channel is found idle. When an idle channel is found, the station does not send immediately. It waits for a period of time called the interframe space or IFS.
  - Even though the channel may appear idle when it is sensed, a distant station may have already started transmitting. The distant station's signal has not yet reached this station.
  - The IFS time allows the front of the transmitted signal by the distant station to reach this station. If after the IFS time the channel is still idle, the station can send, but it still needs to wait a time equal to the contention time (described next). The IFS variable can also be used to prioritize stations or frame types.
- **Contention Window**
  - The contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time.
  - The number of slots in the window changes according to the binary exponential back-off strategy.
  - This means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time. This is very similar to the

p-persistent method except that a random outcome defines the number of slots taken by the waiting station.

- One interesting point about the contention window is that the station needs to sense the channel after each time slot.
- However, if the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle. This gives priority to the station with the longest waiting time.

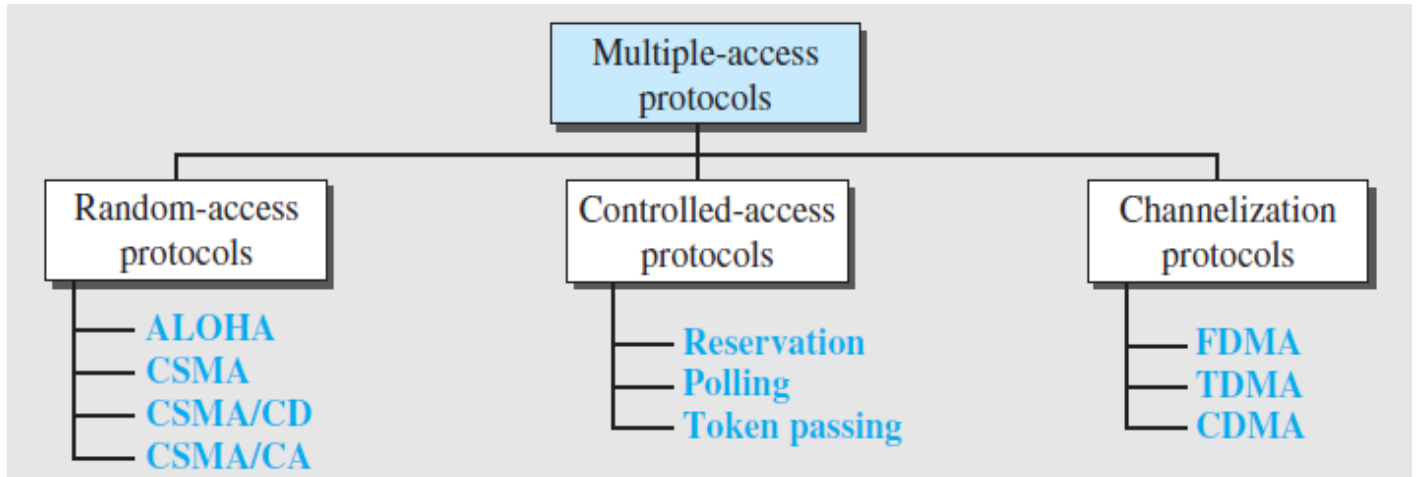
- **Acknowledgment**

- With all these precautions, there still may be a collision resulting in destroyed data. In addition, the data may be corrupted during the transmission. The positive acknowledgment and the time-out timer can help guarantee that the receiver has received the frame.



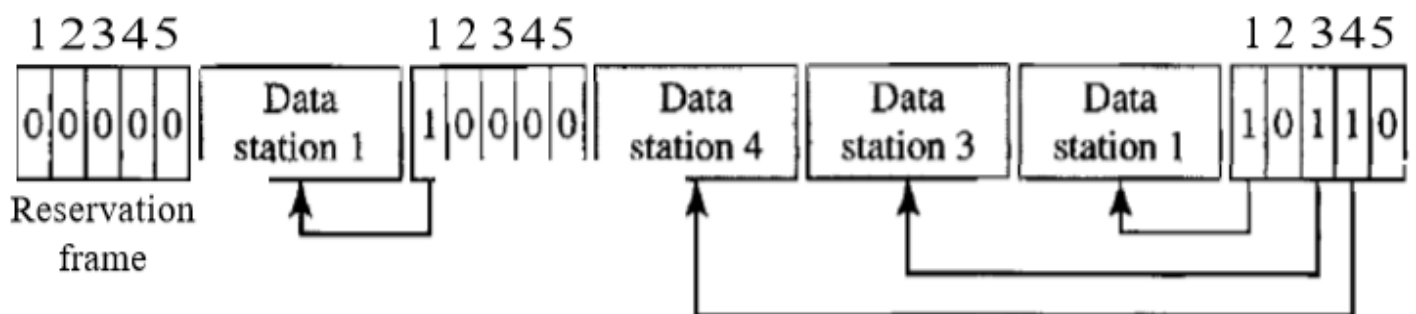
## CONTROLLED ACCESS

- In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations. We discuss three popular controlled-access methods.



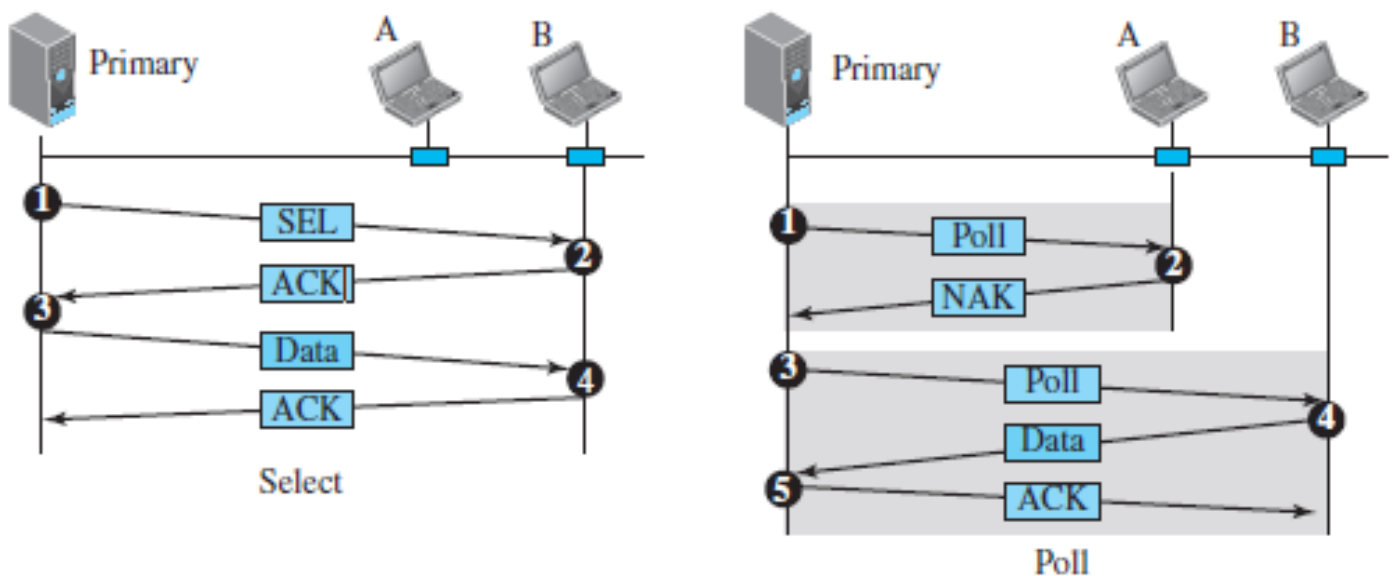
## Reservation

- In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.
- If there are N stations in the system, there are exactly N reservation mini slots in the reservation frame. Each mini slot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own mini slot. The stations that have made reservations can send their data frames after the reservation frame.



# Polling

- Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations.
- All data exchanges must be made through the primary device even when the ultimate destination is a secondary device.
- The primary device controls the link; the secondary devices follow its instructions. It is up to the primary device to determine which device is allowed to use the channel at a given time.



- If the primary wants to receive data, it asks the secondaries if they have anything to send; this is called poll function. If the primary wants to send data, it tells the secondary to get ready to receive; this is called select function.
- **Select**
  - The select function is used whenever the primary device has something to send. Remember that the primary controls the link. If the primary is neither sending nor receiving data, it knows the link is available.
  - If it has something to send, the primary device sends it. What it does not know, however, is whether the target device is prepared to receive.
  - So the primary must alert the secondary to the upcoming transmission and wait for an acknowledgment of the secondary's ready status. Before sending data, the primary creates and transmits a select (SEL) frame, one field of which includes the address of the intended secondary.
- **Poll**
  - The poll function is used by the primary device to solicit transmissions from the secondary devices.

- When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send.
- When the first secondary is approached, it responds either with a NAK frame if it has nothing to send or with data (in the form of a data frame) if it does.
- If the response is negative (a NAK frame), then the primary polls the next secondary in the same manner until it finds one with data to send.
- When the response is positive (a data frame), the primary reads the frame and returns an acknowledgment (ACK frame), verifying its receipt.

**Q** A broadcast channel has 10 nodes and total capacity of 10 Mbps. It uses polling for medium access. Once a node finishes transmission, there is a polling delay of  $80 \mu\text{s}$  to poll the next node. Whenever a node is polled, it is allowed to transmit a maximum of 1000 bytes. The maximum throughput of the broadcast channel is (Gate-2007) (2 Marks)

**(A)** 1 Mbps

**(B)** 100/11 Mbps

**(C)** 10 Mbps

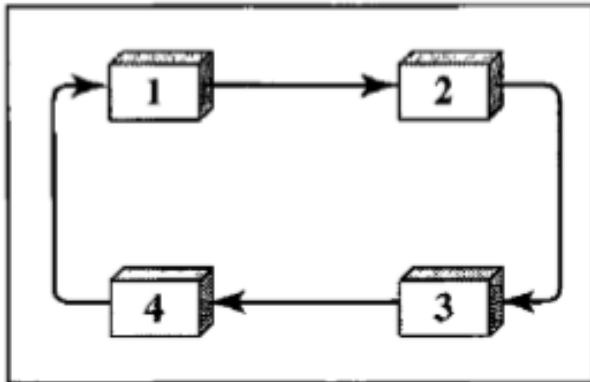
**(D)** 100 Mbps

**Answer: (B)**

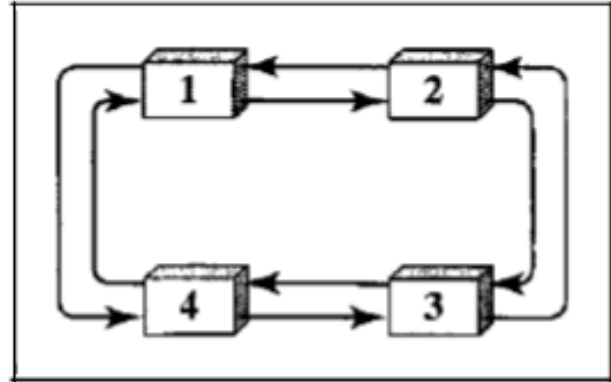
## Token Passing

- In the token-passing method, the stations in a network are organized in a logical ring.
- In other words, for each station, there is a predecessor and a successor. The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in the ring.
- The current station is the one that is accessing the channel now. The right to this access has been passed from the predecessor to the current station. The right will be passed to the successor when the current station has no more data to send. But how is the right to access the channel passed from one station to another?
- In this method, a special packet called a token circulates through the ring. The possession of the token gives the station the right to access the channel and send its data.
- When a station has some data to send, it waits until it receives the token from its predecessor. It then holds the token and sends its data. When the station has no more data to send, it releases the token, passing it to the next logical station in the ring.
- The station cannot send data until it receives the token again in the next round. In this process, when a station receives the token and has no data to send, it just passes the data to the next station.
- Token management is needed for this access method. Stations must be limited in the time they can have possession of the token. The token must be monitored to ensure it has not been lost or destroyed.
- For example, if a station that is holding the token fails, the token will disappear from the network. Another function of token management is to assign priorities to the stations and to the types of data being transmitted. And finally, token management is needed to make low-priority stations release the token to high priority stations.

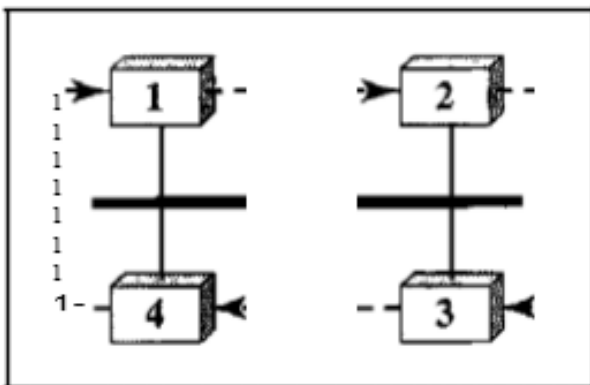
## Logical ring and physical topology in token-passing access method



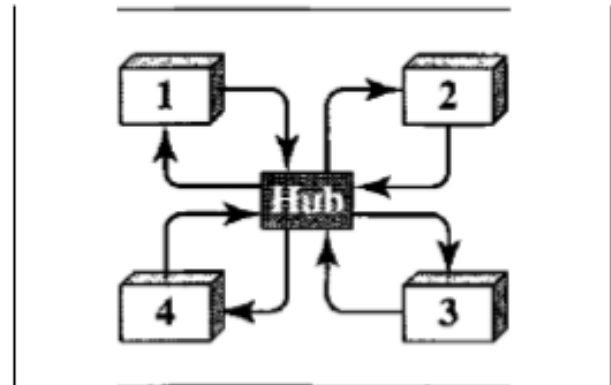
a. Physical ring



b. Dual ring



c. Bus ring



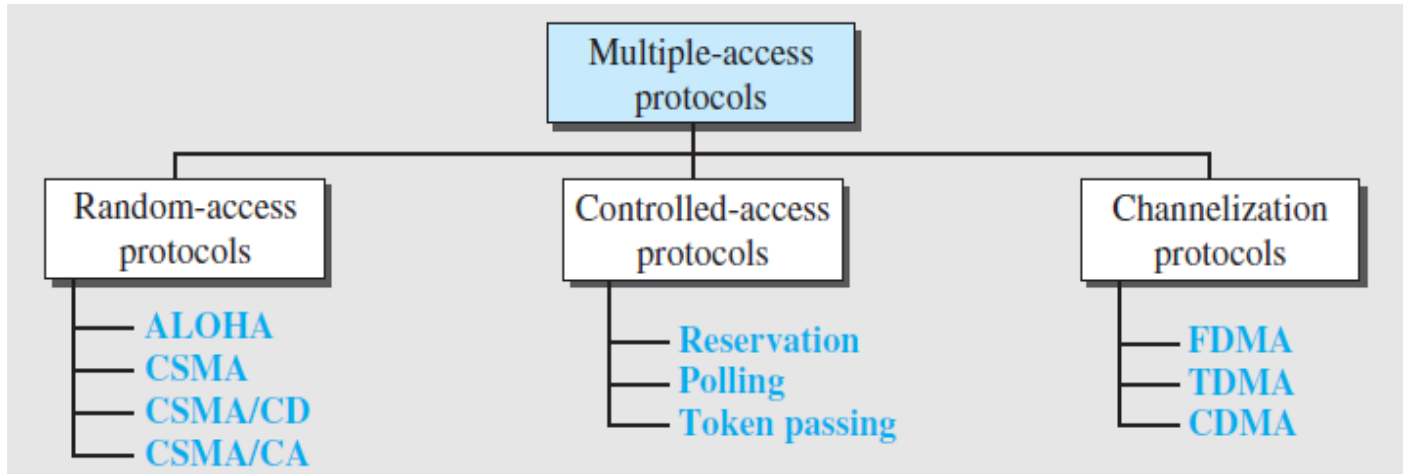
d. Star ring

- Logical Ring
- In a token-passing network, stations do not have to be physically connected in a ring; the ring can be a logical one. In the physical ring topology, when a station sends the token to its successor, the token cannot be seen by other stations; the successor is the next one in line.
- This means that the token does not have to have the address of the next successor. The problem with this topology is that if one of the links—the medium between two adjacent stations fails, the whole system fails.
- The dual ring topology uses a second (auxiliary) ring which operates in the reverse direction compared with the main ring. The second ring is for emergencies only (such as a spare tire for a car).
- If one of the links in the main ring fails, the system automatically combines the two rings to form a temporary ring. After the failed link is restored, the auxiliary ring becomes idle again.

- Note that for this topology to work, each station needs to have two transmitter ports and two receiver ports. The high-speed Token Ring networks called FDDI (Fiber Distributed Data Interface) and CDDI (Copper Distributed Data Interface) use this topology.
- In the bus ring topology, also called a token bus, the stations are connected to a single cable called a bus. They, however, make a logical ring, because each station knows the address of its successor (and also predecessor for token management purposes).
- When a station has finished sending its data, it releases the token and inserts the address of its successor in the token. Only the station with the address matching the destination address of the token gets the token to access the shared media.
- The Token Bus LAN, standardized by IEEE, uses this topology. In a star ring topology, the physical topology is a star. There is a hub, however, that acts as the connector. The wiring inside the hub makes the ring; the stations are connected to this ring through the two wire connections.
- This topology makes the network less prone to failure because if a link goes down, it will be bypassed by the hub and the rest of the stations can operate. Also adding and removing stations from the ring is easier. This topology is still used in the Token Ring LAN designed by IBM.

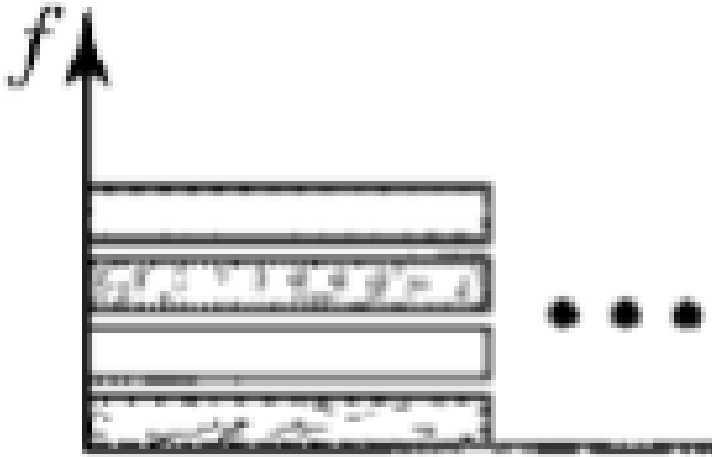
## CHANNELIZATION

- Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different stations. In this section, we discuss three channelization protocols: FDMA, TDMA, and CDMA.



## Frequency-Division Multiple Access (FDMA)

- In frequency-division multiple access (FDMA), the available bandwidth is divided into frequency bands. Each station is allocated a band to send its data.
- In other words, each band is reserved for a specific station, and it belongs to the station all the time.



## Time-Division Multiple Access (TDMA)

- In time-division multiple access (TDMA), the stations share the bandwidth of the channel in time. Each station is allocated a time slot during which it can send data. Each station transmits its data in its assigned time slot.



**Q** In a TDM medium access control bus LAN, each station is assigned one time slot per cycle for transmission. Assume that the length of each time slot is the time to transmit 100 bits plus the end-to-end propagation delay. Assume a propagation speed of  $2 \times 10^8$  m/sec. The length of the LAN is 1 km with a bandwidth of 10 Mbps. The maximum number of stations that can be allowed in the LAN so that the throughput of each station can be  $2/3$  Mbps is (Gate-2005) (2 Marks)

(A) 3

(B) 5

(C) 10

(D) 20

**Answer: (C)**

## Code-Division Multiple Access (CDMA)

- Code-division multiple access (CDMA) was conceived several decades ago. Recent advances in electronic technology have finally made its implementation possible. CDMA differs from FDMA because only one channel occupies the entire bandwidth of the link. It differs from TDMA because all stations can send data simultaneously; there is no timesharing.
- Let us first give an analogy. CDMA simply means communication with different codes. For example, in a large room with many people, two people can talk in English if nobody else understands English.
- Another two people can talk in Chinese if they are the only ones who understand Chinese, and so on. In other words, the common channel, the space of the room in this case, can easily allow communication between several couples, but in different languages (codes).

## Differences between CSMA/CD and ALOHA

- The first difference is the addition of the persistence process. We need to sense the channel before we start sending the frame by using one of the persistence processes we discussed previously (nonpersistent, 1-persistent, or p-persistent).
- The second difference is the frame transmission. In ALOHA, we first transmit the entire frame and then wait for an acknowledgment. In CSMA/CD, transmission and collision detection is a continuous process. We do not send the entire frame and then look for a collision. The station transmits and receives continuously and simultaneously (using two different ports). We use a loop to show that transmission is a continuous process.
- We constantly monitor in order to detect one of two conditions: either transmission is finished or a collision is detected. Either event stops transmission. When we come out of the loop, if a collision has not been detected, it means that transmission is complete; the entire frame is transmitted. Otherwise, a collision has occurred.
- The third difference is the sending of a short jamming signal that enforces the collision in case other stations have not yet sensed the collision.

CSMA/CD	ALOHA
It has the persistence process, sense before send	There is no persistence in ALOHA
Transmission and collision detection are continuous processes. We do not send the entire frame and then look for a collision.	We first transmit the entire frame and then wait for an acknowledgment.
Sending of a short <b>jamming signal</b> to make sure that all other stations become aware of the collision.	No jamming signals are used

### ***Throughput***

- The maximum throughput occurs at a different value of  $G$  and is based on the persistence method and the value of  $p$  in the  $p$ -persistent approach.
- For the 1-persistent method, the maximum throughput is around 50 percent when  $G = 1$ .
- For the nonpersistent method, the maximum throughput can go up to 90 percent when  $G$  is between 3 and 8.

**Q** Consider a simple communication system where multiple nodes are connected by a shared broadcast medium (like Ethernet or wireless).

The nodes in the system use the following carrier-sense based medium access protocol. A node that receives a packet to transmit will carrier-sense the medium for 5 units of time. If the node does not detect any other transmission in this duration, it starts transmitting its packet in the next time unit. If the node detects another transmission, it waits until this other transmission finishes, and then begins to carrier-sense for 5 time units again.

Once they start to transmit, nodes do not perform any collision detection and continue transmission even if a collision occurs. All transmissions last for 20 units of time. Assume that the transmission signal travels at the speed of 10 meters per unit time in the medium.

Assume that the system has two nodes  $P$  and  $Q$ , located at a distance  $d$  meters from each other.  $P$  starts transmitting a packet at time  $t=0$  after successfully completing its carrier-sense phase. Node  $Q$  has a packet to transmit at time  $t=0$  and begins to carrier-sense the medium.

The maximum distance  $d$  (in meters, rounded to the closest integer) that allows  $Q$  to successfully avoid a collision between its proposed transmission and  $P$ 's ongoing transmission is \_\_\_\_\_. **(Gate-2018) (2 Marks)**

Ans: 50

**Q** Consider a simplified time slotted MAC protocol, where each host always has data to send and transmits with probability  $p = 0.2$  in every slot. There is no backoff and one frame can be transmitted in one slot. If more than one host transmits in the same slot, then the transmissions are unsuccessful due to collision. What is the maximum number of hosts which this protocol can support, if each host has to be provided a minimum through put of 0.16 frames per time slot? (Gate-2004) (2 Marks)

(A) 1

(B) 2

(C) 3

(D) 4

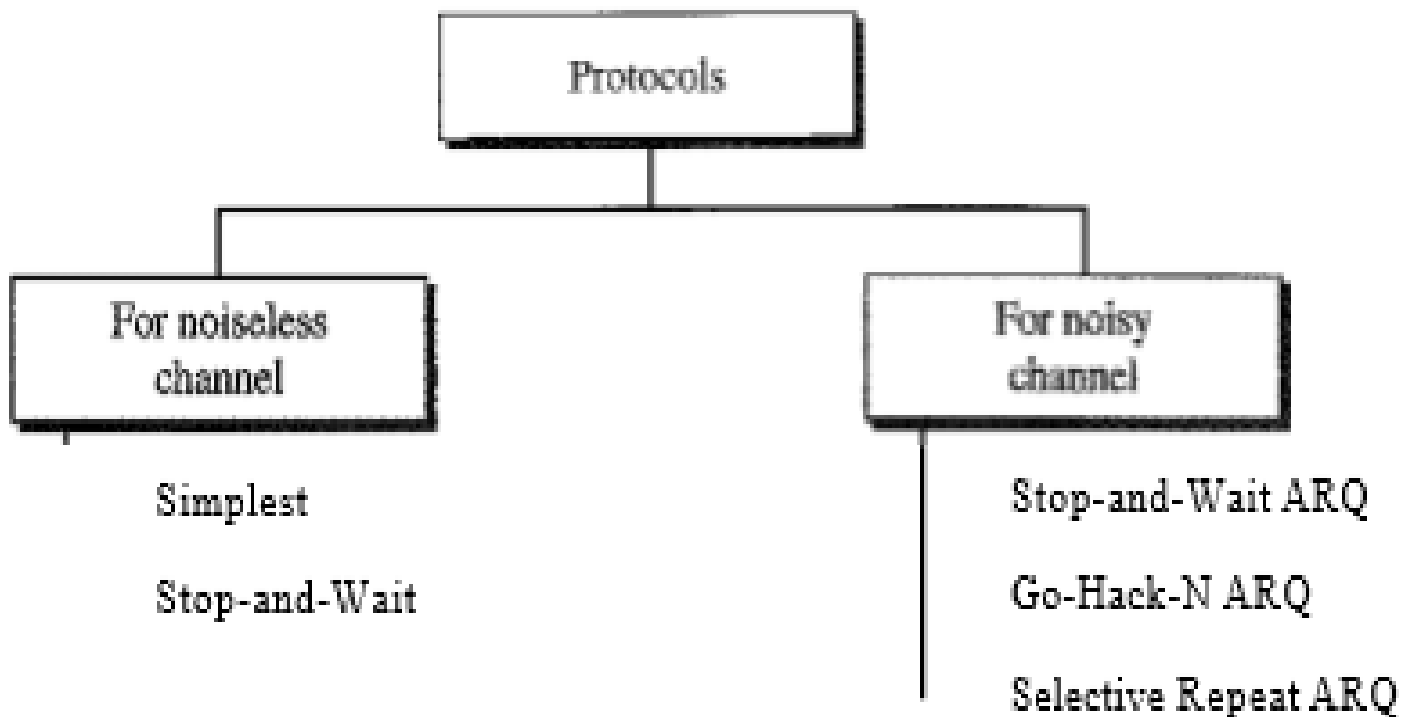
**Answer: (B)**

## FLOW AND ERROR CONTROL

- Flow control coordinates the amount of data that can be sent before receiving an acknowledgment and is one of the most important duties of the data link layer.
- The flow of data must not be allowed to overwhelm the receiver. Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data.
- The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily.
- Incoming data must be checked and processed before they can be used. The rate of such processing is often slower than the rate of transmission. For this reason, each receiving device has a block of memory, called a buffer, reserved for storing incoming data until they are processed.
- If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive.

- **Error Control (lost, out of order, corrupt) (detection and retransmission)**

- Error control is both error detection and error correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender.
- In the data link layer, the term error control refers primarily to methods of error detection and retransmission.
- Error control in the data link layer is often implemented simply: Any time an error is detected in an exchange, specified frames are retransmitted. This process is called automatic repeat request (**ARQ**)

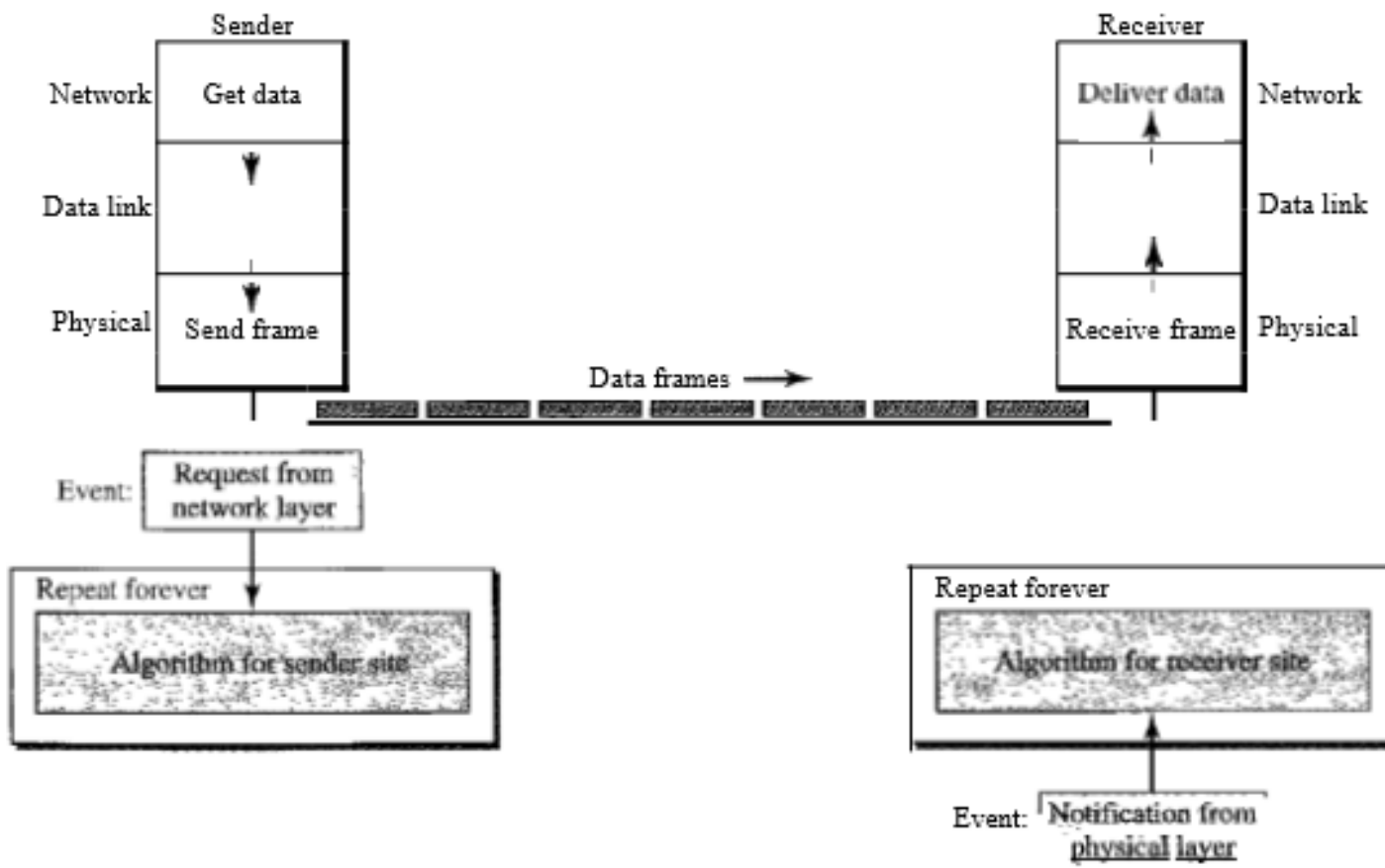


## **NOISELESS CHANNELS**

- Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted.
- We introduce two protocols for this type of channel. The first is a protocol that does not use flow control; the second is the one that does. Of course, neither has error control because we have assumed that the channel is a perfect noiseless channel.

### **Simplest Protocol**

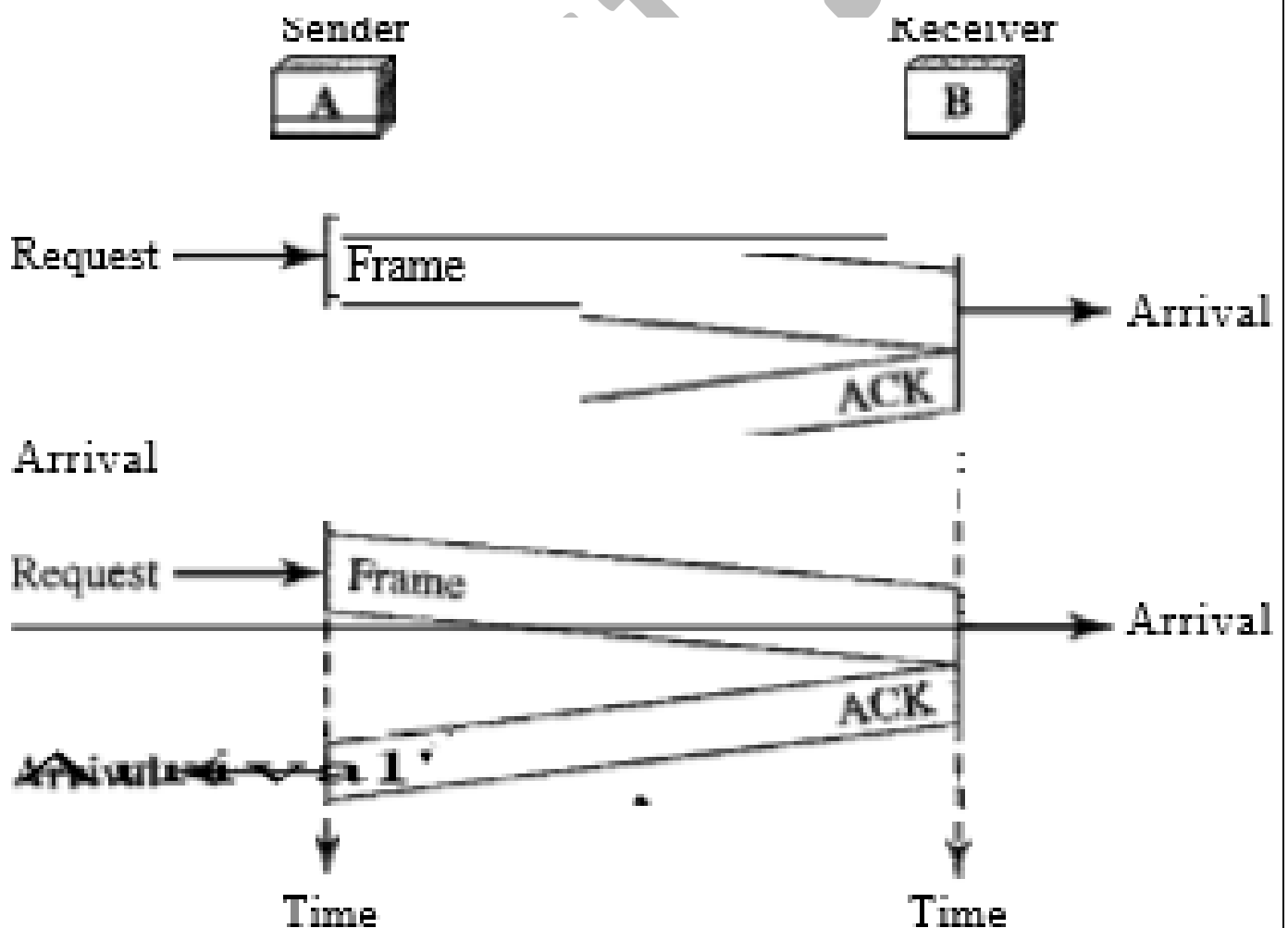
- Our first protocol, is one that has no flow or error control. it is a unidirectional protocol in which data frames are traveling in only one direction-from the sender to receiver.
- We assume that the receiver can immediately handle any frame it receives with a processing time that is small enough to be negligible.
- The data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer, which can also accept the packet immediately. In other words, the receiver can never be overwhelmed with incoming frames. There is no need for flow control in this scheme.
- We need to elaborate on the procedure used by both data link layers. The sender site cannot send a frame until its network layer has a data packet to send. The receiver site cannot deliver a data packet to its network layer until a frame arrives.



Sanchit

## Stop-and-Wait Protocol

- If data frames arrive at the receiver site faster than they can be processed, the frames must be stored until their use.
- Normally, the receiver does not have enough storage space, especially if it is receiving data from many sources. This may result in either the discarding of frames or denial of service.
- To prevent the receiver from becoming overwhelmed with frames, we somehow need to tell the sender to slow down. There must be feedback from the receiver to the sender.
- The protocol we discuss now is called the Stop-and-Wait Protocol because the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame.
- We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction. We add flow control to our previous protocol.



## NOISY CHANNELS

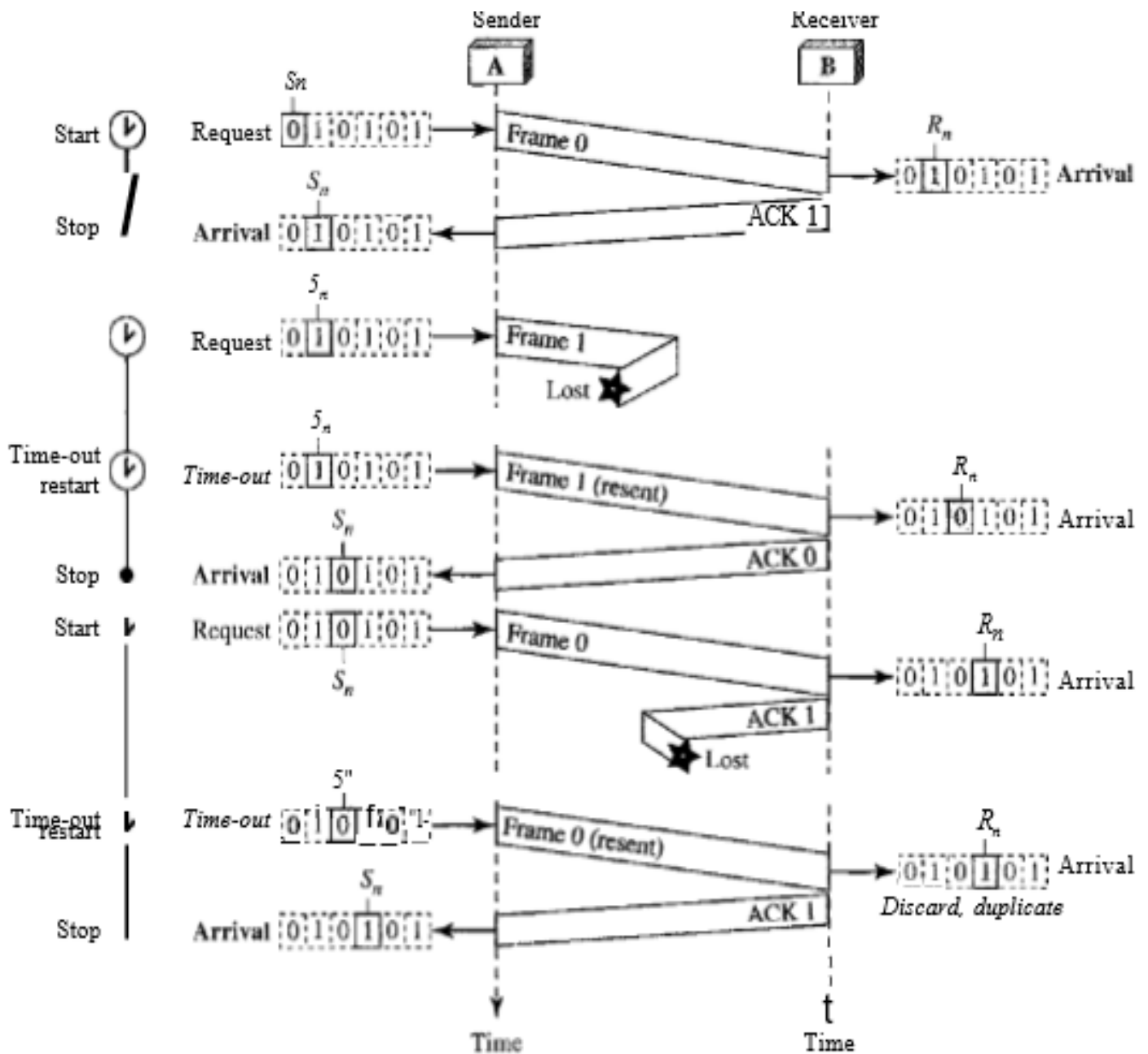
- Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are non-existent. We can ignore the error (as we sometimes do), or we need to add error control to our protocols. We discuss three protocols in this section that use error control.

### Stop-and-Wait Automatic Repeat Request

- Our first protocol, called the Stop-and-Wait Automatic Repeat Request (Stop-and-Wait ARQ), adds a simple error control mechanism to the Stop-and-Wait Protocol.
- Let us see how this protocol detects and corrects errors. To detect and correct corrupted frames, we need to add redundancy bits to our data frame.
- When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver.
- Lost frames are more difficult to handle than corrupted ones. In our previous protocols, there was no way to identify a frame. The received frame could be the correct one, or a duplicate, or a frame out of order.
- The solution is to number the frames. When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated.
- The corrupted and lost frames need to be resent in this protocol. If the receiver does not respond when there is an error, how can the sender know which frame to resend?
- To remedy this problem, the sender keeps a copy of the sent frame. At the same time, it starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted.
- Since the protocol uses the stop-and-wait mechanism, there is only one specific frame that needs an ACK even though several copies of the same frame can be in the network. Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number. The ACK frame for this protocol has a sequence number field. In this protocol, the sender simply discards a corrupted ACK frame or ignores an out-of-order one.

#### Sequence Numbers

- As we discussed, the protocol specifies that frames need to be numbered. This is done by using sequence numbers.
- A field is added to the data frame to hold the sequence number of that frame. One important consideration is the range of the sequence numbers. Since we want to minimize the frame size, we look for the smallest range that provides unambiguous communication.
- The sequence numbers of course can wrap around. For example, if we decide that the field is  $m$  bits long, the sequence numbers start from 0, go to  $2^m - 1$ , and then are repeated.
- Let us reason out the range of sequence numbers we need. Assume we have used  $x$  as a sequence number; we only need to use  $x + 1$  after that. There is no need for  $x + 2$ . To show this, assume that the sender has sent the frame numbered  $x$ . Three things can happen
- The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment. The acknowledgment arrives at the sender site, causing the sender to send the next frame numbered  $x + 1$ .
- The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment, but the acknowledgment is corrupted or lost. The sender resends the frame (numbered  $x$ ) after the time-out. Note that the frame here is a duplicate. The receiver can recognize this fact because it expects frame  $x + 1$  but frame  $x$  was received.
- The frame is corrupted or never arrives at the receiver site; the sender resends the frame (numbered  $x$ ) after the time-out.
- We can see that there is a need for sequence numbers  $x$  and  $x + 1$  because the receiver needs to distinguish between case 1 and case 2. But there is no need for a frame to be numbered  $x + 2$ .
- In case 1, the frame can be numbered  $x$  again because frames  $x$  and  $x + 1$  are acknowledged and there is no ambiguity at either site. In cases 2 and 3, the new frame is  $x + 1$ , not  $x + 2$ . If only  $x$  and  $x + 1$  are needed, we can let  $x = 0$  and  $x + 1 = 1$ . This means that the sequence is 0, 1, 0, 1, 0, and so on.



## Efficiency

- The Stop-and-Wait ARQ is very inefficient if our channel is thick and long. By thick, we mean that our channel has a large bandwidth; by long, we mean the round-trip delay is long.
- The product of these two is called the bandwidth delay product. We can think of the channel as a pipe. The bandwidth-delay product then is the volume of the pipe in bits. The pipe is always there.
- If we do not use it, we are inefficient. The bandwidth-delay product is a measure of the number of bits we can send out of our system while waiting for news from the receiver.

Transmission Delay (TT): A sender needs to put the bits in a packet on the line one by one. If the first bit of the packet is put on the line at time  $t_1$  and the last bit is put on the line at time  $t_2$ , transmission delay of the packet is  $(t_2 - t_1)$ .

$$T_t = (\text{Packet length } (L)) / (\text{Transmission rate or Bandwidth } (B)) = L / B$$

Example: Fast Ethernet LAN with the transmission rate of 100 million bits per second and a packet of 10,000 bits, the transmission delay will be:

$$(10,000) / (100,000,000) = 100 \text{ microseconds.}$$

*Propagation Delay:* Propagation delay is the time it takes for a bit to travel from point A to point B in the transmission media.

$$T_p = (\text{Distance}) / (\text{Propagation speed})$$

**Processing Delay** - It is the time required for a destination host to receive a packet from its input port, remove the header, perform an error detection procedure, and deliver the packet to the output port or deliver the packet to the upper-layer protocol (in the case of the destination host).

**Delay<sub>pr</sub>** = Time required to process a packet in a destination host

**Queuing Delay** It is measured as the time a packet waits in the input queue and output queue of a router.

**Delay<sub>qu</sub>** = The time a packet waits in input and output queues in a router

$$\text{Total delay} = (n + 1) (T_t + T_p + \text{Delay}_{pr}) + (n) (\text{Delay}_{qu})$$

**Note:**

- If we have  $n$  routers, we have  $(n + 1)$  links. Therefore, we have  $(n + 1)$  transmission delays related to  $n$  routers and the source,  $(n + 1)$  propagation delays related to  $(n + 1)$  links,  $(n + 1)$  processing delays related to  $n$  routers and the destination, and only  $n$  queuing delays related to  $n$  routers.
- In most of the numerical we will consider the processing and queuing delays as 0.

## Measuring Performance for Stop and Wait

- The total time is measured as:
- Total time =  $T_t(\text{data}) + T_p(\text{data}) + \text{Dealy}_{\text{que}} + \text{Delay}_{\text{pro}} + T_t(\text{ack}) + T_p(\text{ack})$
- Thus,  $\text{Total time} = T_t + (2 * T_p)$
- Here we have taken  $T_t(\text{ack})$  as negligible as the ack size is generally very less, the  $T_p$  for data and ack are almost going to be same and queuing delay and processing delays as already discussed are kept 0.
- In general,  $2 * T_p$  time is also called Round Trip Time (RTT)
  
- Efficiency is measured as:  $\eta = \text{Useful Time} / \text{Total Cycle time} = T_t / T_t + 2 * T_p$ , Here, Useful time in the entire cycle time is  $T_t$  and for the rest  $2 * T_p$  time we are waiting for the processing, whereas instead of waiting we could have sent more packets.
- Dividing numerator and denominator with  $T_t$ , we get:  $\eta = 1 / 1 + (2 * T_p / T_t)$
- So,  $\eta = 1 / 1 + 2a$ , (where  $a = T_p / T_t$ )
  
- Effective Bandwidth / Throughput / Bandwidth Utilization is calculated as: Throughput =  $L / T_t + 2 * T_p$
- Or, dividing and multiplying the numerator with B we get
- Throughput =  $(L/B) * B / T_t + 2 * T_p \Rightarrow T_t * B / T_t + 2T_p \Rightarrow \eta * B$  (efficiency \* bandwidth)

**Example:** A sender uses the Stop-and-Wait ARQ protocol for reliable transmission of frames. Frames are of size 1000 bytes and the transmission rate at the sender is 80 Kbps (1Kbps = 1000 bits/second). Size of an acknowledgement is 100 bytes and the transmission rate at the receiver is 8 Kbps. The one-way propagation delay is 100 milliseconds. Assuming no frame is lost, the sender throughput is \_\_\_\_\_ bytes/second. (Gate-2016) (2 Marks)

**Ans: 2500**

**Q** Suppose that the stop-and-wait protocol is used on a link with a bit rate of 64 kilobits per second and 20 milliseconds propagation delay. Assume that the transmission time for the acknowledgment and the processing time at nodes are negligible. Then the minimum frame size in bytes to achieve a link utilization of at least 50% is \_\_\_\_\_. (Gate-2015) (2 Marks)

(A) 160

(B) 320

(C) 640

(D) 220

**Answer: (B)**

**Q** A link has a transmission speed of  $10^6$  bits/sec. It uses data packets of size 1000 bytes each. Assume that the acknowledgment has negligible transmission delay, and that its propagation delay is the same as the data propagation delay. Also assume that the processing delays at nodes are negligible. The efficiency of the stop-and-wait protocol in this setup is exactly 25%. The value of the one-way propagation delay (in milliseconds) is \_\_\_\_\_ . **(Gate-2015) (1 Marks)**

**Answer: (12)**

**Q** On a wireless link, the probability of packet error is 0.2. A stop-and-wait protocol is used to transfer data across the link. The channel condition is assumed to be independent from transmission to transmission. What is the average number of transmission attempts required to transfer 100 packets? **(Gate-2006) (2 Marks)**

**(A) 100                                      (B) 125                                      (C) 150                                      (D) 200**

**Answer: (B)**

**Q** A channel has a bit rate of 4 kbps and one-way propagation delay of 20 ms. The channel uses stop and wait protocol. The transmission time of the acknowledgement frame is negligible. To get a channel efficiency of at least 50%, the minimum frame size should be **(Gate-2005) (2 Marks)**

**(A) 80 bytes                                      (B) 80 bits                                      (C) 160 bytes                                      (D) 160 bits**

**Answer: (D)**

**Q** The values of parameters for the Stop-and-Wait ARQ protocol are as given below:

- Bit rate of the transmission channel = 1 Mbps.
- Propagation delay from sender to receiver = 0.75 ms.
- Time to process a frame = 0.25 ms.
- Number of bytes in the information frame = 1980.
- Number of bytes in the acknowledge frame = 20.
- Number of overhead bytes in the information frame = 20.

Assume there are no transmission errors. Then, the transmission efficiency (expressed in percentage) of the Stop-and-Wait ARQ protocol for the above parameters is \_\_\_\_\_ (correct to 2 decimal places). **(Gate-2017) (2 Marks)**



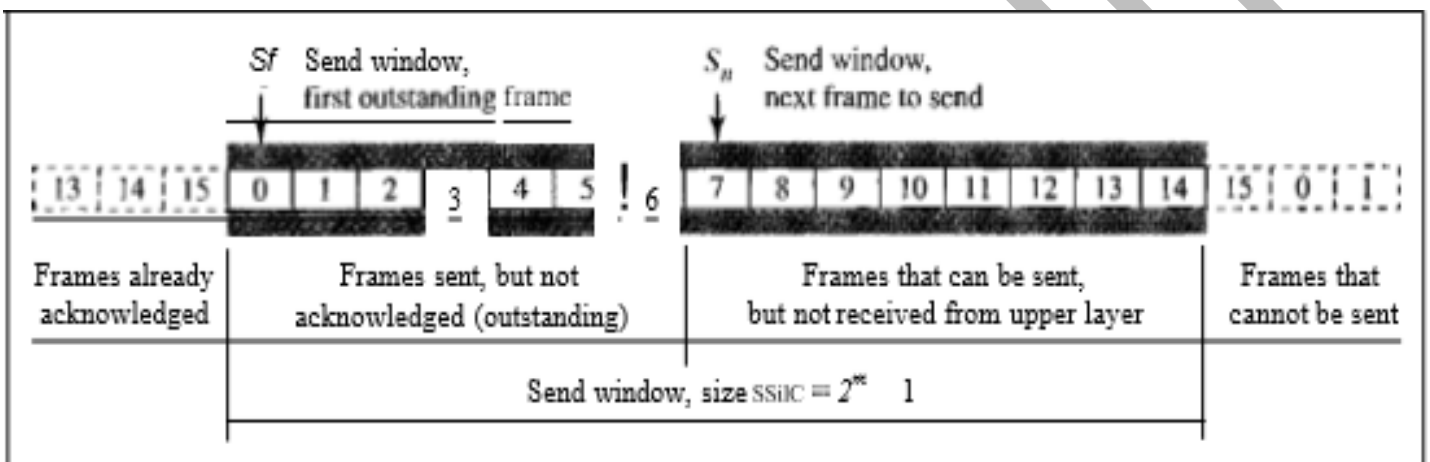
## Go-Back-N Automatic Repeat Request

- To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition while waiting for acknowledgment.
- In other words, we need to let more than one frame be outstanding to keep the channel busy while the sender is waiting for acknowledgment. The first is called Go-Back-N Automatic Repeat Request.
- In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.

### Sliding Window

- In this protocol (and the next), the sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver.
- In other words, the sender and receiver need to deal with only part of the possible sequence numbers. The range which is the concern of the sender is called the send sliding window; the range that is the concern of the receiver is called the receive sliding window.
- We discuss both here. The send window is an imaginary box covering the sequence numbers of the data frames which can be in transit. In each window position, some of these sequence numbers define the frames that have been sent; others define those that can be sent.
- The maximum size of the window is  $2^m - 1$ .
- The window at any time divides the possible sequence numbers into four regions. The first region, from the far left to the left wall of the window, defines the sequence numbers belonging to frames that are already acknowledged.
- The sender does not worry about these frames and keeps no copies of them. The second region, colored, defines the range of sequence numbers belonging to the frames that are sent and have an unknown status.
- The sender needs to wait to find out if these frames have been received or were lost. We call these outstanding frames.
- The third range, defines the range of sequence numbers for frames that can be sent; however, the corresponding data packets have not yet been received from the network layer.
- Finally, the fourth region defines sequence numbers that cannot be used until the window slides. The window itself is an abstraction; three variables define its size and location at any time.

- We call these variables  $S_f$  (send window, the first outstanding frame),  $S_n$  (send window, the next frame to be sent), and  $S_{size}$  (send window, size).
- The variable  $S_f$  defines the sequence number of the first (oldest) outstanding frame. The variable  $S_n$  holds the sequence number that will be assigned to the next frame to be sent. Finally, the variable  $S_{size}$  defines the size of the window, which is fixed in our protocol.
- The receive window makes sure that the correct data frames are received and that the correct acknowledgments are sent. The size of the receive window is always 1. The receiver is always looking for the arrival of a specific frame. Any frame arriving out of order is discarded and needs to be resent.



## Timers

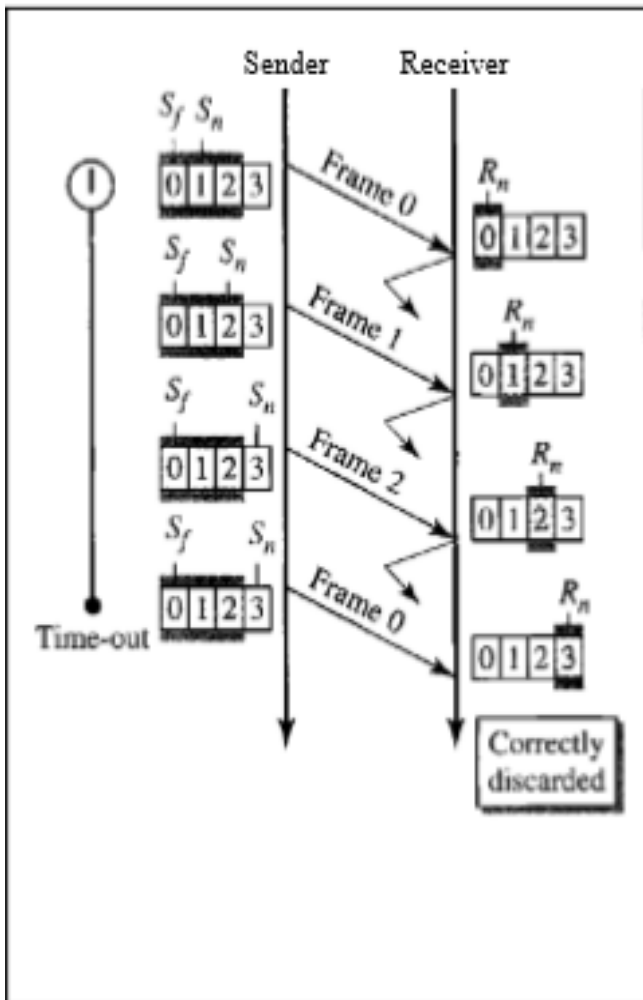
- Although there can be a timer for each frame that is sent, in our protocol we use only one. The reason is that the timer for the first outstanding frame always expires first; we send all outstanding frames when this timer expires.

## Acknowledgment

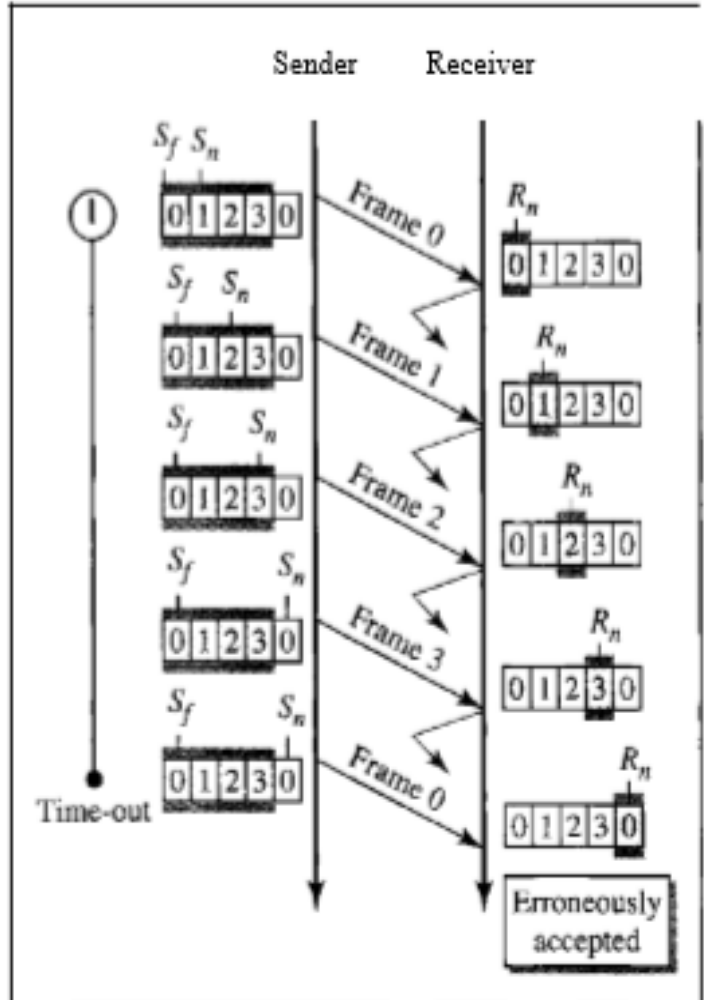
- The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order. If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting.
- The silence of the receiver causes the timer of the unacknowledged frame at the sender site to expire. This, in turn, causes the sender to go back and resend all frames, beginning with the one with the expired timer.
- The receiver does not have to acknowledge each frame received. It can send one cumulative acknowledgment for several frames.

## Resending a Frame

- When the timer expires, the sender resends all outstanding frames. For example, suppose the sender has already sent frame 6, but the timer for frame 3 expires. This means that frame 3 has not been acknowledged; the sender goes back and sends frames 3, 4, 5, and 6 again. That is why the protocol is called Go-Back-NARQ.



a. Window size  $< 2^m$



b. Window size  $= 2^m$

**Q** Consider a network connecting two systems located 8000 kilometres apart. The bandwidth of the network is  $500 \times 10^6$  bits per second. The propagation speed of the media is  $4 \times 10^6$  meters per second. It is needed to design a Go-Back-N sliding window protocol for this network. The average packet size is  $10^7$  bits. The network is to be used to its full capacity. Assume that processing delays at nodes are negligible. Then, the minimum size in bits of the sequence number field has to be \_\_\_\_\_. (Gate-2015) (2 Marks)

**Answer: (8)**

**Q** A 1Mbps satellite link connects two ground stations. The altitude of the satellite is 36,504 km and speed of the signal is  $3 \times 10^8$  m/s. What should be the packet size for a channel

utilization of 25% for a satellite link using go-back-127 sliding window protocol? Assume that the acknowledgment packets are negligible in size and that there are no errors during communication. **(Gate-2008) (2 Marks)**

**(A)** 120 bytes                      **(B)** 60 bytes                      **(C)** 240 bytes                      **(D)** 90 bytes

**Answer: (A)**

**Q** Station A needs to send a message consisting of 9 packets to Station B using a sliding window (window size 3) and go-back-n error control strategy. All packets are ready and immediately available for transmission. If every 5th packet that A transmits gets lost (but no acks from B ever get lost), then what is the number of packets that A will transmit for sending the message to B? **(Gate-2006) (2 Marks)**

**(A)** 12                      **(B)** 14                      **(C)** 16                      **(D)** 18

**Answer: (C)**

**Q** A 20 Kbps satellite link has a propagation delay of 400 ms. The transmitter employs the “go back n ARQ” scheme with n set to 10. Assuming that each frame is 100 bytes long, what is the maximum data rate possible? **(Gate-2004) (2 Marks)**

**(A)** 5Kbps                      **(B)** 10Kbps                      **(C)** 15Kbps                      **(D)** 20Kbps

**Answer: (B)**

**Example:** Consider that the transmission delay for a link is 1ms and Propagation delay is 49.5 sec. What should be the maximum window size in order to achieve maximum efficiency? And minimum number of bits required in sequence field?

The maximum window size =  $1 + 2a = (1 + 2 (49.5 / 1)) = 100$

Minimum number of bits in sequence number field =  $\log_2 (100) = 6.7 = 7$  bits

In some cases the sequence number bits are often reserved, suppose in the above example we say that Sequence number bits are fixed to be 5.

Then, with 5 bits we can generate only  $2^5 = 32$  sequence numbers, i.e. only 32 frames can be sent in a pipeline.

Thus efficiency is =  $32 / 100 = 32\%$

Thus window size can also be defined in a much better way as:  $Min (1 + 2a, 2^N)$

Sliding window protocol is implemented through: *Go-Back N ARQ and Selective Repeat ARQ*

**Example:** If  $T_t$  is given as 1msec and  $T_p$  is given as 99.5 ms and the protocol used is GB-10, calculate the throughput is bandwidth is given as 40 MBPS?

Since, we know to achieve maximum efficiency the  $W_s = 1 + 2a = 1 + 199 = 200$

But we have already fixed the  $W_s = 10$ .

So efficiency =  $10 / 200 = 1 / 20 = 5\%$ .

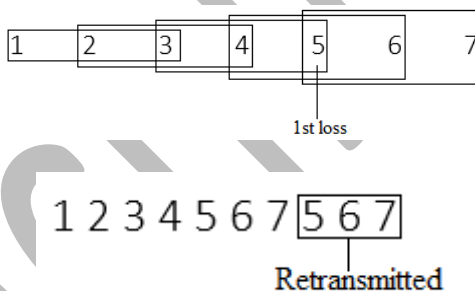
Throughput = Efficiency \* Bandwidth =  $(5 / 100) * 40 \text{ MBPS} = 2 \text{ MBPS}$

**Example:** In GB-3 if every 5<sup>th</sup> packet is lost, if we need to send 10 packets then how many transmissions are required?

Let the packets be: 1 2 3 4 5 6 7 5 6 7 8 9 10

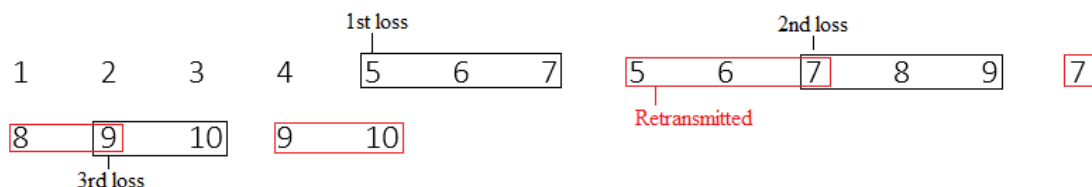
The first packet lost will be at 5, and  $W_s = 3$ .

The entire window will now be retransmitted, i.e. 5, 6, 7 will be retransmitted.



Now, we start again counting from first 6, then we get to know that next loss will be at frame 7. So again the entire frame i.e. 7 8 9 is retransmitted.

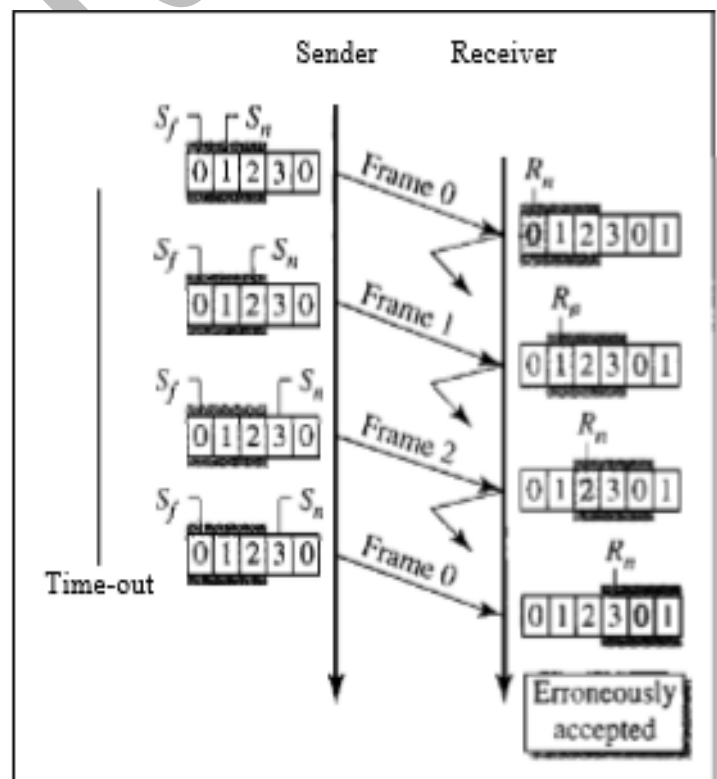
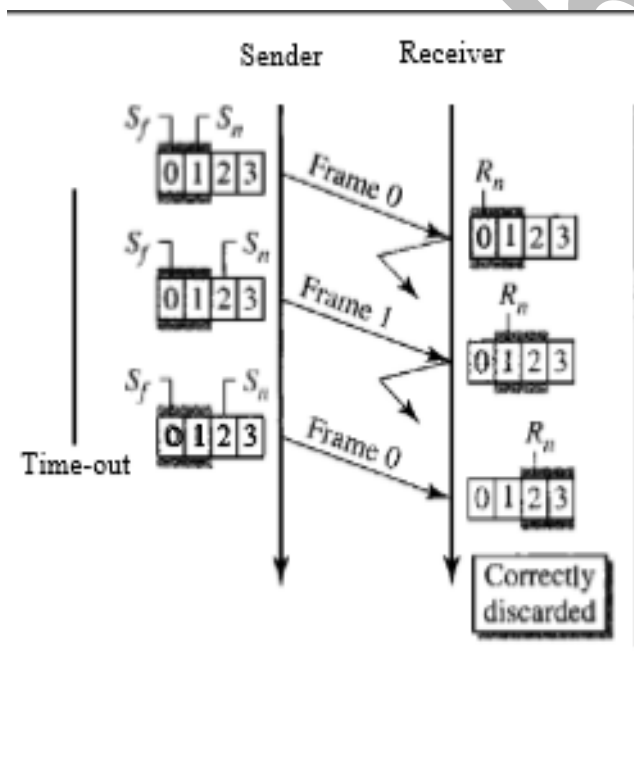
Thus, following the same procedure we get;



Total Transmissions: 18

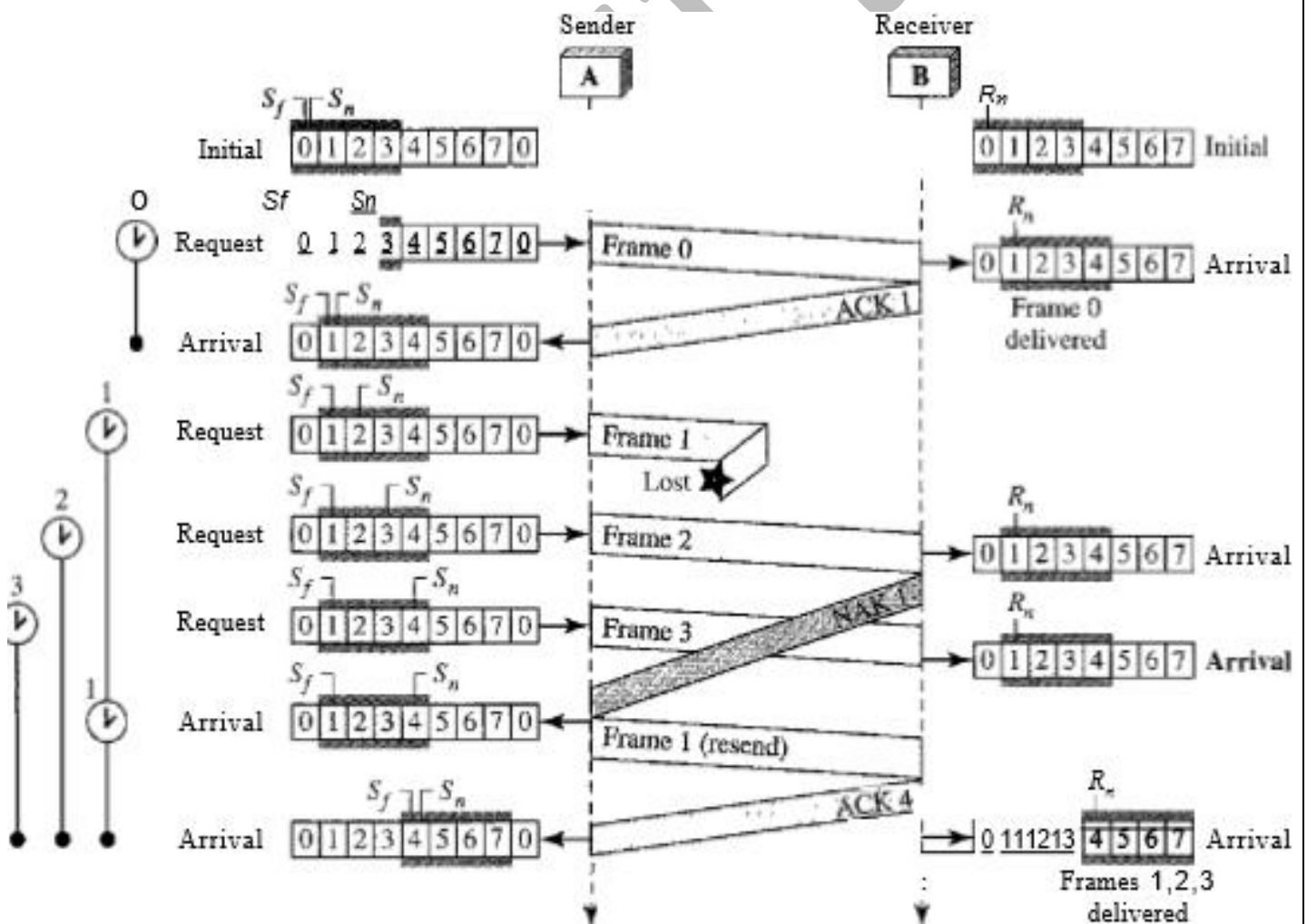
## Selective Repeat Automatic Repeat Request

- Go-Back-N ARQ simplifies the process at the receiver site. The receiver keeps track of only one variable, and there is no need to buffer out-of-order frames; they are simply discarded.
- However, this protocol is very inefficient for a noisy link. In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission.
- For noisy links, there is another mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent. This mechanism is called Selective Repeat ARQ.
- It is more efficient for noisy links, but the processing at the receiver is more complex. The Selective Repeat Protocol allows as many frames as the size of the receive window to arrive out of order and be kept until there is a set of in-order frames to be delivered to the network layer.
- Because the sizes of the send window and receive window are the same, all the frames in the send frame can arrive out of order and be stored until they can be delivered.



- In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of  $2^m$

- The handling of the request event is similar to that of the previous protocol except that one timer is started for each frame sent. The arrival event is more complicated here. An ACK or a NAK frame may arrive. If a valid NAK frame arrives, we just resend the corresponding frame. If a valid ACK arrives, we use a loop to purge the buffers, stop the corresponding timer, and move the left wall of the window. The time-out event is simpler here; only the frame which times out is resent.
- Analysis Here we need more initialization. In order not to overwhelm the other side with NAKs, we use a variable called Nak Sent. To know when we need to send an ACK, we use a variable called Ack Needed.
- Both of these are initialized to false. We also use a set of variables to mark the slots in the receive window once the corresponding frame has arrived and is stored.
- If we receive a corrupted frame and a NAK has not yet been sent, we send a NAK to tell the other site that we have not received the frame we expected.
- If the frame is not corrupted and the sequence number is in the window, we store the frame and mark the slot. If contiguous frames, starting from  $R_n$  have been marked, we deliver their data to the network layer and slide the window.



## Piggybacking

- The three protocols we discussed in this section are all unidirectional: data frames flow in only one direction although control information such as ACK and NAK frames can travel in the other direction.
- In real life, data frames are normally flowing in both directions: from node A to node B and from node B to node A. This means that the control information also needs to flow in both directions. A technique called piggybacking is used to improve the efficiency of the bidirectional protocols.
- When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.
- Note that each node now has two windows: one send window and one receive window. Both also need to use a timer. Both are involved in three types of events: request, arrival, and time-out.
- However, the arrival event here is complicated; when a frame arrives, the site needs to handle control information as well as the frame itself. Both of these concerns must be taken care of in one event, the arrival event. The request event uses only the send window at each site; the arrival event needs to use both windows. An important point about piggybacking is that both sites must use the same algorithm. This algorithm is complicated because it needs to combine two arrival events into one. We leave this task as an exercise.

## Sequence and Acknowledgement Numbers

To improve the efficiency of transmission (to fill the pipe), multiple packets must be in transition while the sender is waiting for acknowledgment

- In order to maximize the efficiency, the window size  $(W_s) = (1 + 2a)$
- The minimum number of sequence numbers required  $= (1 + 2a)$
- Number of bits required for sequence numbers  $= \text{ceil}(\log_2(1 + 2a))$

Example: Consider an SR protocol, with sender window size equals to 3 and we have to send 10 packets and every 5<sup>th</sup> packet is lost.

Soln: Let the packets be:

1    2    3    4    5    6    7    8    9    10

We know that the first packet loss will be at 5, instead of transmitting the entire window we will only transfer packet 5.

1    2    3    4    5    5    6    7    8    9    10

Next packet loss will be at 9, as it is the next fifth packet.

1 2 3 4 5 5 6 7 8 9 9 10

Total transmissions: 12

Sanchit Jain

## Comparison

	Stop and Wait ARQ	Go back N	Selective Repeat	Remarks
<b>Efficiency</b>	$1 / (1+2a)$	$N / (1+2a)$	$N / (1+2a)$	Go back N and Selective Repeat gives better efficiency than Stop and Wait ARQ.
<b>Window Size</b>	Sender Window Size = 1 Receiver Window Size = 1	Sender Window Size = N Receiver Window Size = 1	Sender Window Size = N Receiver Window Size = N	Buffer requirement in Selective Repeat is very large. If the system does not have lots of memory, then it is better to choose Go back N.
<b>Minimum number of sequence numbers required</b>	2	N+1	2 x N	Selective Repeat requires large number of bits in sequence number field.

<b>Retransmissions required if a packet is lost</b>	Only the lost packet is retransmitted	The entire window is retransmitted	Only the lost packet is retransmitted	Selective Repeat is far better than Go back N in terms of retransmissions required.
<b>Bandwidth Requirement</b>	Bandwidth requirement is Low	Bandwidth requirement is high because even if a single packet is lost, entire window has to be retransmitted. Thus, if error rate is high, it wastes a lot of bandwidth.	Bandwidth requirement is moderate	Selective Repeat is better than Go back N in terms of bandwidth requirement.
<b>CPU usage</b>	Low	Moderate	High due to searching and sorting required at sender and receiver side	Go back N is better than Selective Repeat in terms of CPU usage.
<b>Level of difficulty in Implementation</b>	Low	Moderate	Complex as it requires extra logic and sorting and searching	Go back N is better than Selective Repeat in terms of implementation difficulty.
<b>Acknowledgements</b>	Uses independent acknowledgement for each packet	Uses cumulative acknowledgements (but may use independent acknowledgements as well)	Uses independent acknowledgement for each packet	Sending cumulative acknowledgements reduces the traffic in the network but if it is lost, then the ACKs for all the corresponding packets are lost.
<b>Type of Transmission</b>	Half duplex	Full duplex	Full duplex	Go back N and Selective Repeat are better in terms of channel usage.

**Q** Consider two hosts X and Y, connected by a single direct link of rate  $10^6$  bits/sec. The distance between the two hosts is 10,000 km and the propagation speed along the link is  $2 \times 10^8$  m/s. Hosts X send a file of 50,000 bytes as one large message to hosts Y continuously. Let the transmission and propagation delays be p milliseconds and q milliseconds, respectively. Then the values of p and q are: **(Gate-2017) (2 Marks)**

**(A)** p = 50 and q = 100

**(B)** p = 50 and q = 400

**(C)** p = 100 and q = 50

**(D)** p = 400 and q = 50

**Ans: d**

**Q** Consider a  $128 \times 10^3$  bits / second satellite communication link with one way propagation delay of 150 milliseconds. Selective retransmission (repeat) protocol is used on this link to send data with a frame size of 1 kilobyte. Neglect the transmission time of acknowledgement. The minimum number of bits required for the sequence number field to achieve 100% utilization is \_\_\_\_\_. **(Gate-2016) (2 Marks)**

ANSWER 4

**Q** Consider a selective repeat sliding window protocol that uses a frame size of 1 KB to send data on a 1.5 Mbps link with a one-way latency of 50 msec. To achieve a link utilization of 60%, the minimum number of bits required to represent the sequence number field is \_\_\_\_\_. **(Gate-2014) (2 Marks)**

ANSWER 5

**Q** Consider a source computer(S) transmitting a file of size  $10^6$  bits to a destination computer(D) over a network of two routers ( $R_1$  and  $R_2$ ) and three links( $L_1$ ,  $L_2$ , and  $L_3$ ).  $L_1$  connects S to  $R_1$ ;  $L_2$  connects  $R_1$  to  $R_2$ ; and  $L_3$  connects  $R_2$  to D. Let each link be of length 100 km. Assume signals travel over each link at a speed of  $10^8$  meters per second. Assume that the link bandwidth on each link is 1Mbps. Let the file be broken down into 1000 packets each of size 1000 bits. Find the total sum of transmission and propagation delays in transmitting the file from S to D? **(Gate-2012) (2 Marks)**

**(A)** 1005 ms

**(B)** 1010 ms

**(C)** 3000 ms

**(D)** 3003 ms

**Answer: (A)**

**Q** Frames of 1000 bits are sent over a  $10^6$  bps duplex link between two hosts. The propagation time is 25ms. Frames are to be transmitted into this link to maximally pack them in transit (within the link).

Let  $I$  be the minimum number of bits that will be required to represent the sequence numbers distinctly assuming that no time gap needs to be given between transmission of two frames.

- a)  $i = 2$                       b)  $i = 3$                       c)  $i = 4$                       d)  $i = 5$

Q Suppose that the sliding window protocol is used with the sender window size of  $2^i$ , where  $i$  is the numbers of bits as mentioned earlier and acknowledgements are always piggy backed. After sending  $2^i$  frames, what is the minimum time the sender will have to wait before starting transmission of the next frame? (Identify the closest choice ignoring the frame processing time) **(Gate-2009) (2 Marks)**

- a) 16ms                      b) 18ms                      c) 20ms                      d) 22ms

**Ans: b**

Q The distance between two stations  $M$  and  $N$  is  $L$  kilometres. All frames are  $K$  bits long. The propagation delay per kilometre is  $t$  seconds. Let  $R$  bits/second be the channel capacity. Assuming that processing delay is negligible, the *minimum* number of bits for the sequence number field in a frame for maximum utilization, when the *sliding window protocol* is used, is: **(Gate-2007) (2 Marks)**

- a)  $\lceil \log_2(2LtR+2K/K) \rceil$                       b)  $\lceil \log_2(2LtR/K) \rceil$   
c)  $\lceil \log_2(2LtR+K/K) \rceil$                       d)  $\lceil \log_2(2LtR+K/2K) \rceil$

**ANSWER C**

Q Station A uses 32-byte packets to transmit messages to Station B using a sliding window protocol. The round-trip delay between A and B is 80 milliseconds and the bottleneck bandwidth on the path between A and B is 128 kbps. What is the optimal window size that A should use? **(Gate-2006) (2 Marks)**

- (A) 20                      (B) 40                      (C) 160                      (D) 320

**Answer: (B)**

Q The maximum window size for data transmission using the selective reject protocol with  $n$ -bit frame sequence numbers is: **(Gate-2005) (1 Marks)**

- (A)  $2^n$                       (B)  $2^{(n-1)}$                       (C)  $2^n - 1$                       (D)  $2^{(n-2)}$

**Answer: (B)**

Q In a sliding window ARQ scheme, the transmitter's window size is  $N$  and the receiver's window size is  $M$ . The minimum number of distinct sequence numbers required to ensure

correct operation of the ARQ scheme is (Gate-2004) (2 Marks)

(A)  $\min(M, N)$

(B)  $\max(M, N)$

(C)  $M + N$

(D)  $MN$

**Answer: (C)**

**Q** Host A is sending data to host B over a full duplex link. A and B are using the sliding window protocol for flow control. The send and receive window sizes are 5 packets each. Data packets (sent only from A to B) are all 1000 bytes long and the transmission time for such a packet is 50  $\mu$ s. Acknowledgement packets (sent only from B to A) are very small and require negligible transmission time. The propagation delay over the link is 200  $\mu$ s. What is the maximum achievable throughput in this communication? **(Gate-2003) (2 Marks)**

(A)  $7.69 \times 10^6$  bytes per second

(B)  $11.11 \times 10^6$  bytes per second

(C)  $12.33 \times 10^6$  bytes per second

(D)  $15.00 \times 10^6$  bytes per second

**Answer: (B)**

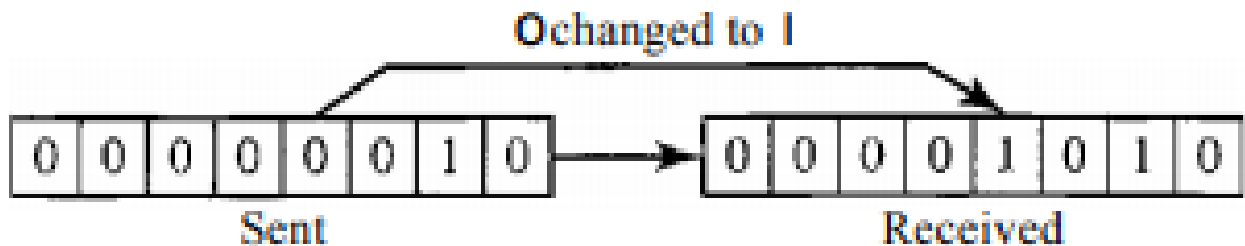
Sanchit

## Types of Errors

- Any time data are transmitted from one node to the next, they can become corrupted in passage.

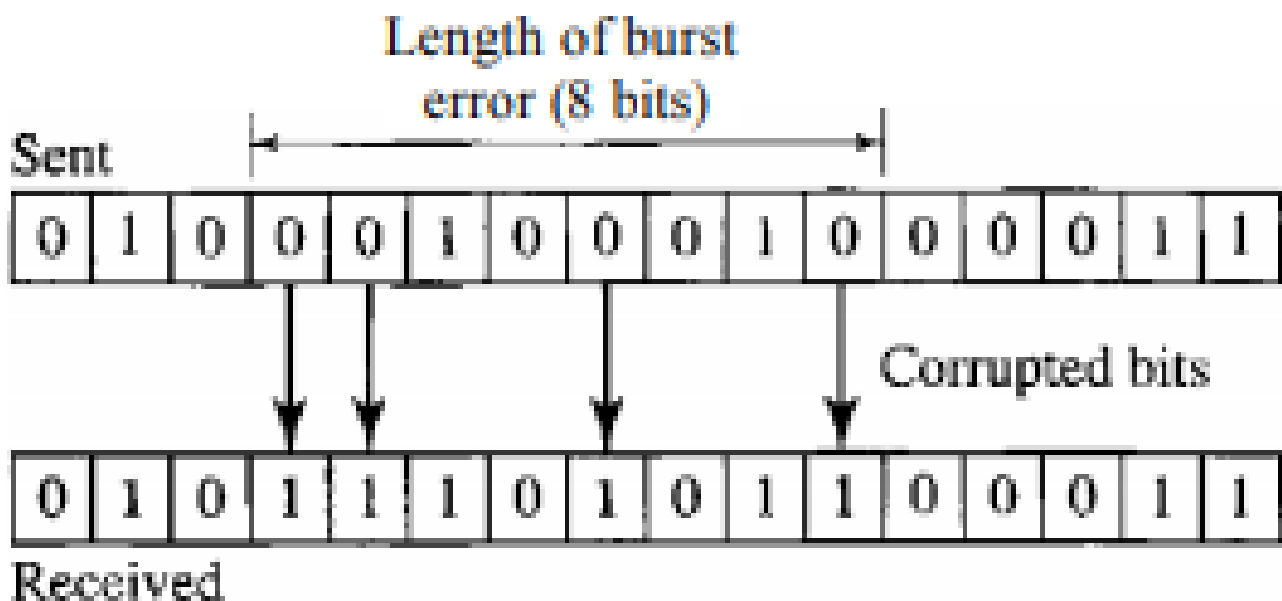
### Single-Bit Error

- The term single-bit error means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1.



### Burst Error

- The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1. Note that a burst error does not necessarily mean that the errors occur in consecutive bits. The length of the burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted.



- A burst error is more likely to occur than a single-bit error. The duration of noise is normally longer than the duration of 1 bit, which means that when noise affects data, it

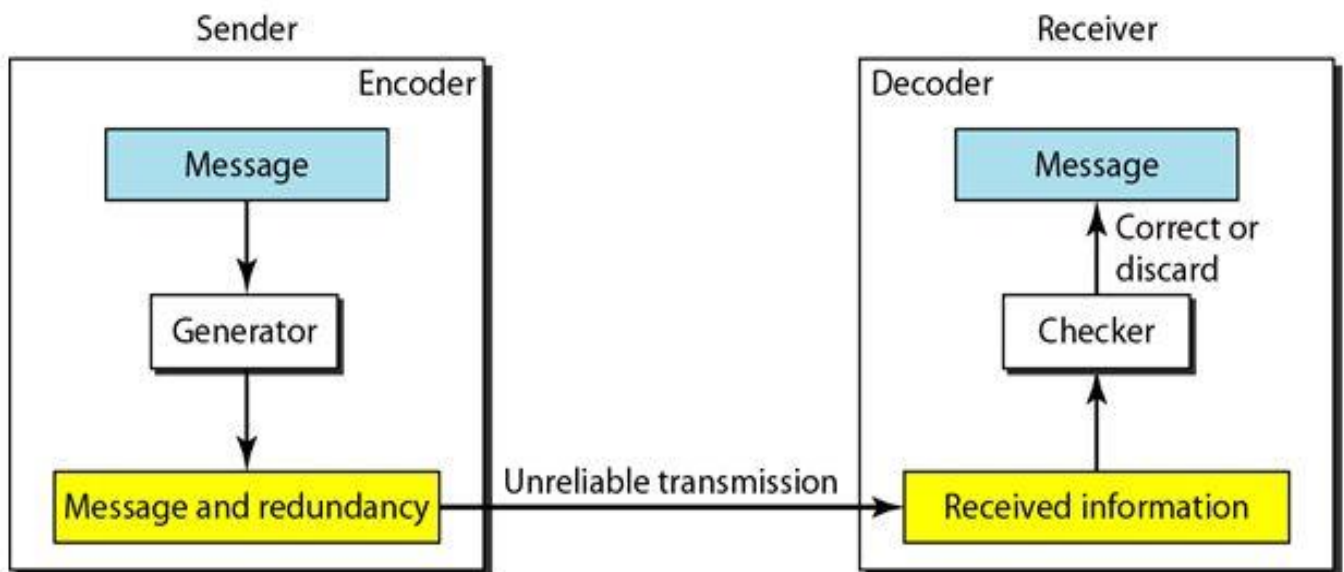
affects a set of bits. The number of bits affected depends on the data rate and duration of noise

## **Redundancy**

- The central concept in detecting or correcting errors is redundancy. To be able to detect or correct errors, we need to send some extra bits with our data.
- These redundant bits are added by the sender and removed by the receiver. Their presence allows the receiver to detect or correct corrupted bits.
- The correction of errors is more difficult than the detection. In error detection, we are looking only to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of errors. A single-bit error is the same for us as a burst error.
- In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location in the message. The number of the errors and the size of the message are important factors.

## Forward Error Correction Versus Retransmission

- There are two main methods of error correction. Forward error correction is the process in which the receiver tries to guess the message by using redundant bits. This is possible, as we see later, if the number of errors is small.
- Correction by retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. Resending is repeated until a message arrives that the receiver believes is error-free (usually, not all errors can be detected).
- Redundancy is achieved through various coding schemes. The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits. The receiver checks the relationships between the two sets of bits to detect or correct the errors.
- We can divide coding schemes into two broad categories: block coding and convolution coding.

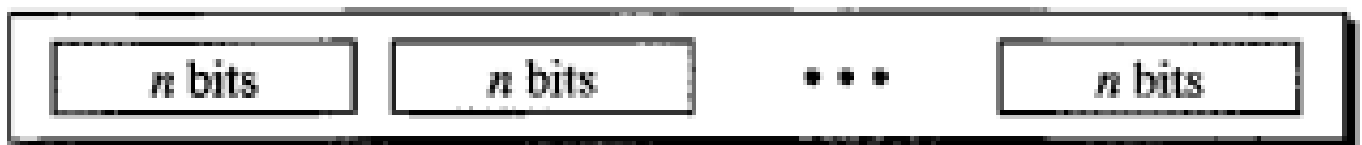


## BLOCK CODING

- In block coding, we divide our message into blocks, each of  $k$  bits, called datawords. We add  $r$  redundant bits to each block to make the length  $n = k + r$ . The resulting  $n$ -bit blocks are called codewords.
- it is important to know that we have a set of datawords, each of size  $k$ , and a set of code words, each of size of  $n$ . With  $k$  bits, we can create a combination of  $2^k$  datawords; with  $n$  bits, we can create a combination of  $2^n$  codewords.
- Since  $n > k$ , the number of possible codewords is larger than the number of possible datawords. The block coding process is one-to-one; the same dataword is always encoded as the same codeword. This means that we have  $2^n - 2^k$  codewords that are not used.



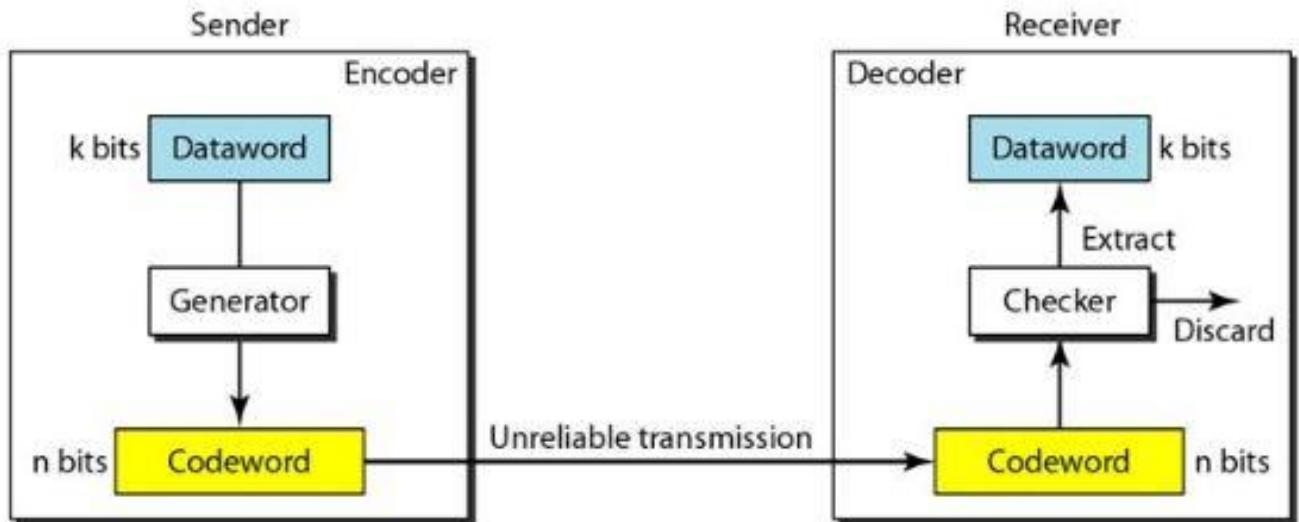
$2^k$  Datawords, each of  $k$  bits



$2^n$  Codewords, each of  $n$  bits (only  $2^k$  of them are valid)

## Error Detection

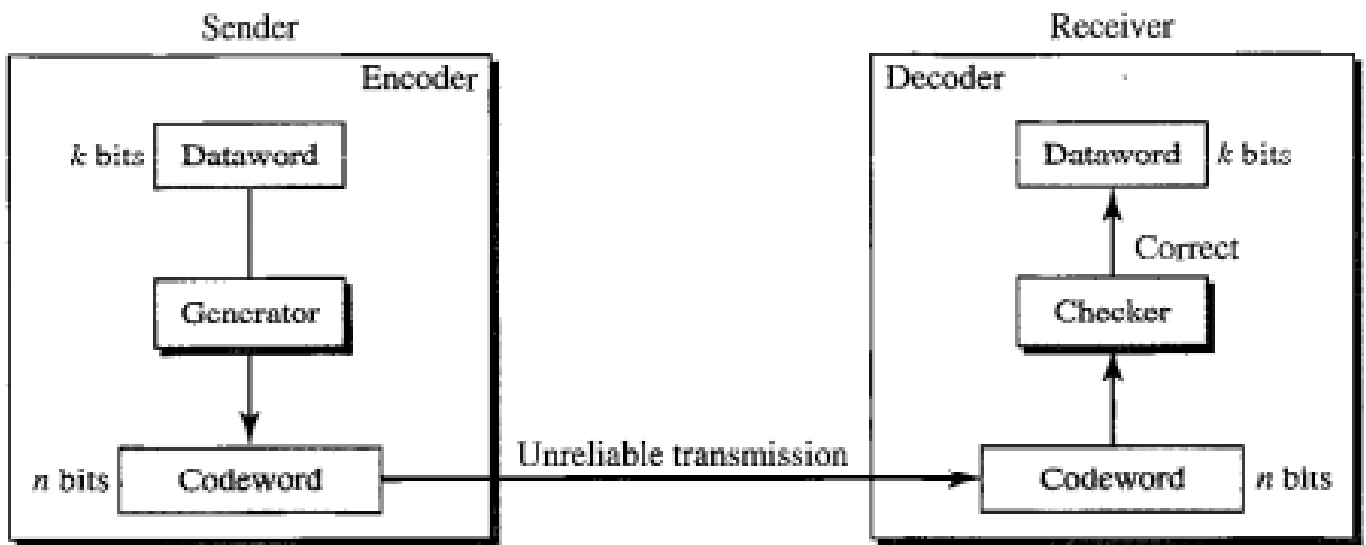
- How can errors be detected by using block coding? If the following two conditions are met, the receiver can detect a change in the original codeword.
- The receiver has (or can find) a list of valid codewords.
- The original codeword has changed to an invalid one.



- The sender creates codewords out of datawords by using a generator that applies the rules and procedures of encoding (discussed later).
- Each codeword sent to the receiver may change during transmission. If the received codeword is the same as one of the valid codewords, the word is accepted; the corresponding dataword is extracted for use.
- If the received codeword is not valid, it is discarded. However, if the codeword is corrupted during transmission but the received word still matches a valid codeword, the error remains undetected.

<i>Datawords</i>	<i>Codewords</i>
00	000
01	011
10	101
11	110

## Error Correction



- Error correction is much more difficult than error detection.
- The checker functions are much more complex.

## Hamming Distance

- The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits.
- We show the Hamming distance between two words  $x$  and  $y$  as  $d(x, y)$ .
- The Hamming distance can easily be found if we apply the XOR operation on the two words and count the number of 1s in the result.

Example:

1. The Hamming distance  $d(000, 011)$  is 2 because  $000 \text{ XOR } 011$  is  $011$  (two 1's).
2. The Hamming distance  $d(10101, 11110)$  is 3 because  $10101 \text{ XOR } 11110$  is  $01011$  (three 1s).

### Minimum Hamming Distance

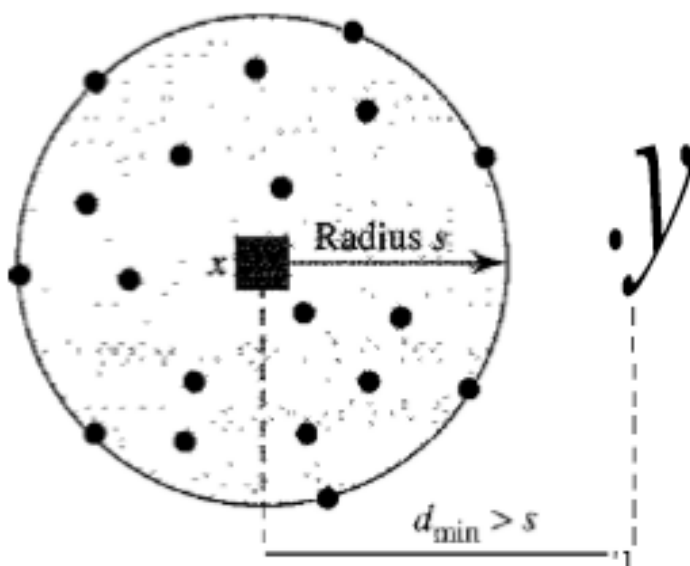
- the minimum Hamming distance is the smallest Hamming distance between all possible pairs.
- $d(00000, 01011) = 3$     $d(01011, 10101) = 4$     $d(00000, 10101) = 3$
- $d(01011, 11110) = 3$     $d(00000, 11110) = 4$     $d(10101, 11110) = 3$
- The  $d_{\min}$  in this case is 3.

## Three Parameters

- Before we continue with our discussion, we need to mention that any coding scheme needs to have at least three parameters: the codeword size  $n$ , the dataword size  $k$ , and the minimum Hamming distance  $d_{\min}$ .
- A coding scheme  $C$  is written as  $C(n, k)$  with a separate expression for  $d_{\min}$
- For example, we can call our first coding scheme  $C(3, 2)$  with  $d_{\min} = 2$  and our second coding scheme  $C(5, 2)$  with  $d_{\min} = 3$ .

### Minimum Distance for Error Detection

- Now let us find the minimum Hamming distance in a code if we want to be able to detect up to  $s$  errors.
- If  $s$  errors occur during transmission, the Hamming distance between the sent codeword and received codeword is  $s$ . If our code is to detect up to  $s$  errors, the minimum distance between the valid codes must be  $s + 1$ ,
- so that the received codeword does not match a valid codeword. In other words, if the minimum distance between all valid codewords is  $s + 1$ , the received codeword cannot be erroneously mistaken for another codeword. The distances are not enough ( $s + 1$ ) for the receiver to accept it as valid. The error will be detected.



#### Legend

- Any valid codeword
- Any corrupted codeword with 0 to  $s$  errors

## Error Correction

- In error detection, the receiver needs to know only that the received codeword is invalid; in error correction the receiver needs to find (or guess) the original codeword sent.
- We can say that we need more redundant bits for error correction than for error detection.

Q if the receiver can correct an error without knowing what was actually sent. We add 3 redundant bits to the 2-bit dataword to make 5-bit codewords.

<i>Dataword</i>	<i>Codeword</i>
00	00000
01	01011
10	10101
11	11110

$$d(00000, 01011) = 3$$

$$d(01011, 10101) = 4$$

$$d(00000, 10101) = 3$$

$$d(01011, 11110) = 3$$

$$d(00000, 11110) = 4$$

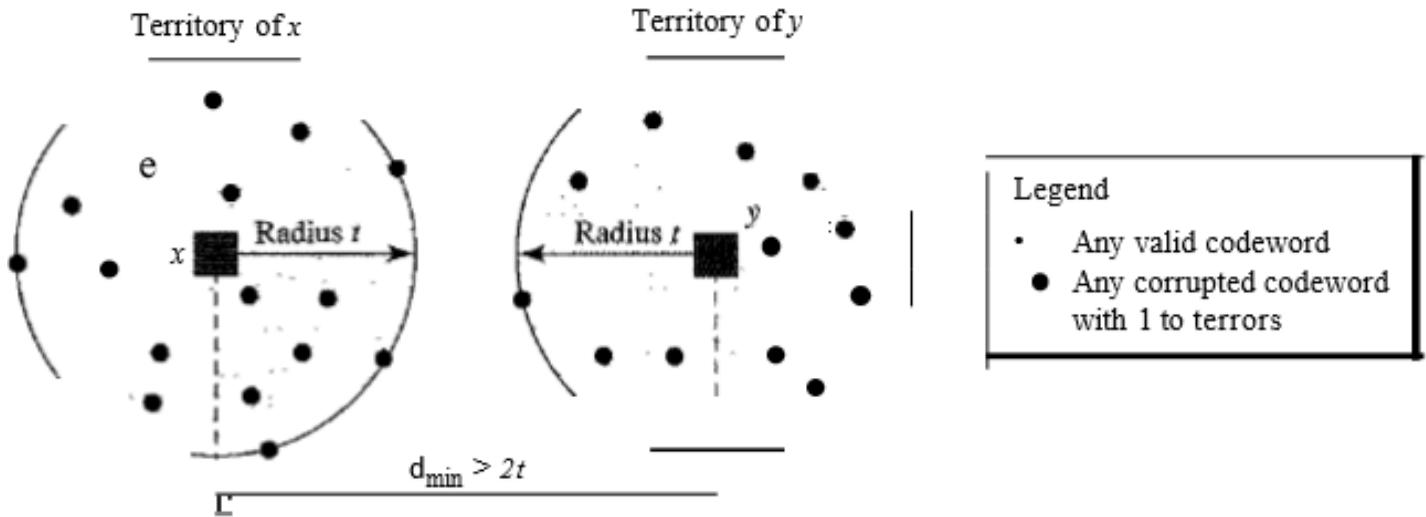
$$d(10101, 11110) = 3$$

The  $d_{min}$  in this case is 3.

Guarantee a double bit error detection

Minimum Distance for Error Correction

- When a received codeword is not a valid codeword, the receiver needs to decide which valid codeword was actually sent. The decision is based on the concept of territory, an exclusive area surrounding the codeword.
- Each valid codeword has its own territory. We use a geometric approach to define each territory. We assume that each valid codeword has a circular territory with a radius of  $t$  and that the valid codeword is at the center.
- For example, suppose a codeword  $x$  is corrupted by  $t$  bits or less. Then this corrupted codeword is located either inside or on the perimeter of this circle. If the receiver receives a codeword that belongs to this territory, it decides that the original codeword is the one at the center. Note that we assume that only up to  $t$  errors have occurred; otherwise, the decision is wrong.



- Error correction is more complex than error detection; a decision is involved.
- To guarantee correction of up to  $t$  errors in all cases, the minimum Hamming distance in a block code must be  $d_{\min} = 2t + 1$ .

**Example:** A code scheme has a Hamming distance  $d_{\min} = 4$ . What is the error detection and correction capability of this scheme?

This code guarantees the detection of up to three errors ( $s = 3$ ), but it can correct up to one error.

**Q** Consider a binary code that consists only four valid codewords as given below. a 00000, 01011, 10101, 11110

Lets minimum Hamming distance of code be  $p$  and maximum number of erroneous bits that can be corrected by the code be  $q$ . The value of  $p$  and  $q$  are: **(Gate-2017) (2 Marks)**

**(A)**  $p = 3$  and  $q = 1$

**(B)**  $p = 3$  and  $q = 2$

**(C)**  $p = 4$  and  $q = 1$

**(D)**  $p = 4$  and  $q = 2$

**Answer: (A)**

**Q** An error correcting code has the following code words:

00000000, 00001111, 01010101, 10101010, 11110000.

What is the maximum number of bit errors that can be corrected? **(Gate-2007) (2 Marks)**

**(A)** 0

**(B)** 1

**(C)** 2

**(D)** 3

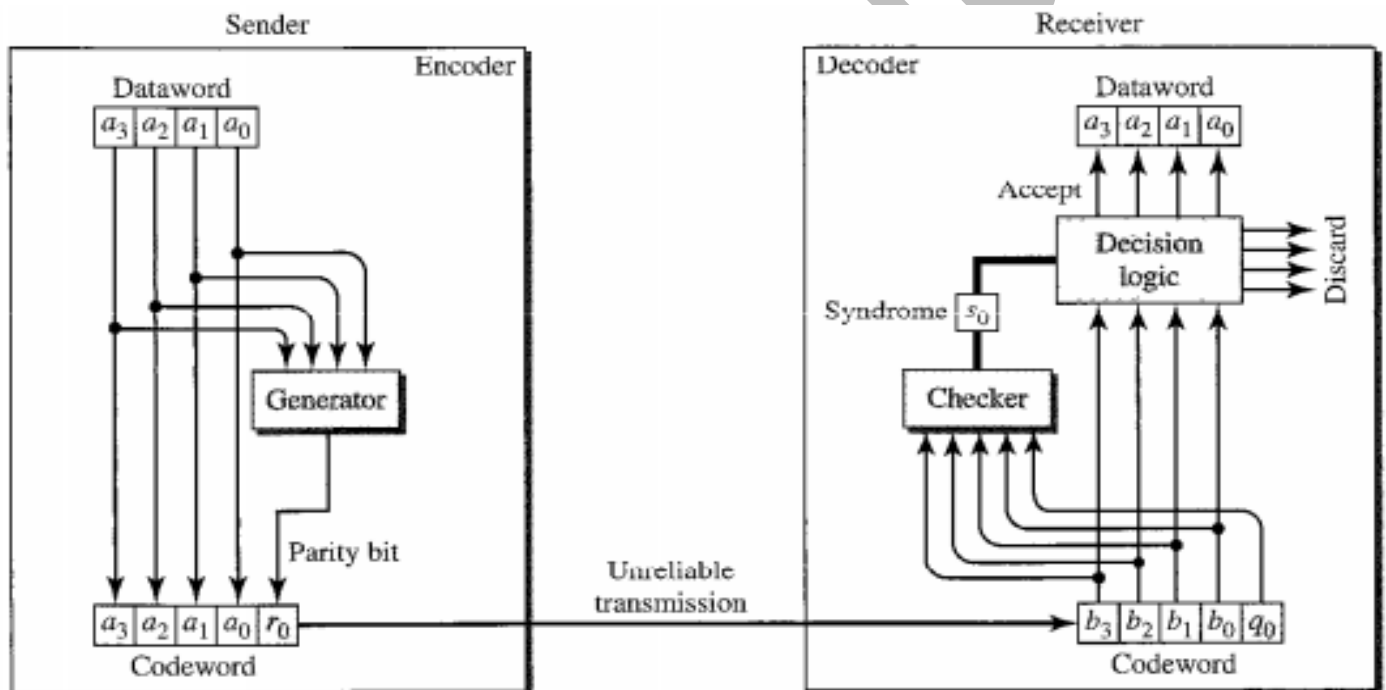
**Answer: (D)**

# LINEAR BLOCK CODES

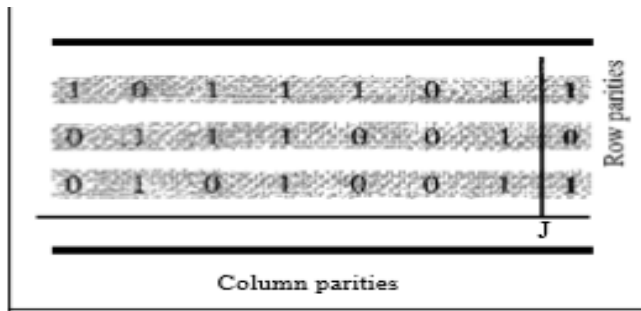
- Almost all block codes used today belong to a subset called linear block codes. The use of nonlinear block codes for error detection and correction is not as widespread because their structure makes theoretical analysis and implementation difficult. We therefore concentrate on linear block codes.

## Simple Parity-Check Code

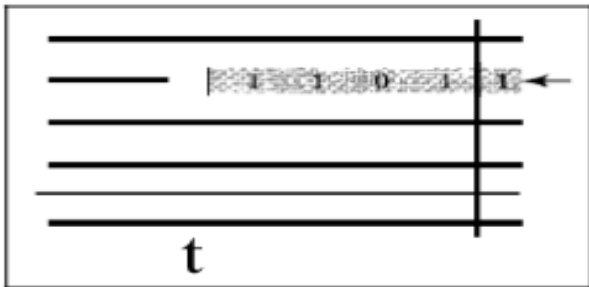
- Perhaps the most familiar error-detecting code is the simple parity-check code. In this code, a  $k$ -bit dataword is changed to an  $n$ -bit codeword where  $n = k + 1$ .
- The extra bit, called the parity bit, is selected to make the total number of 1s in the codeword even. Although some implementations specify an odd number of 1s.
- A simple parity-check code is a single-bit error-detecting code in which  $n = k + 1$  with  $d_{min} = 2$ .



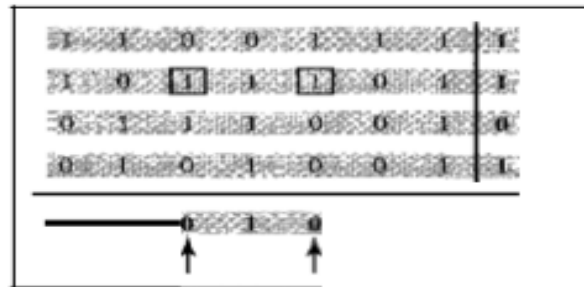
- A simple parity-check code can detect an odd number of errors.
- A better approach is the two-dimensional parity check. In this method, the dataword is organized in a table (rows and columns). The data to be sent, five 7-bit bytes, are put in separate rows. For each row and each column, 1 parity-check bit is calculated.
- The whole table is then sent to the receiver, which finds the syndrome for each row and each column., the two-dimensional parity check can detect up to three errors that occur anywhere in the table (arrows point to the locations of the created nonzero syndromes). However, errors affecting 4 bits may not be detected.



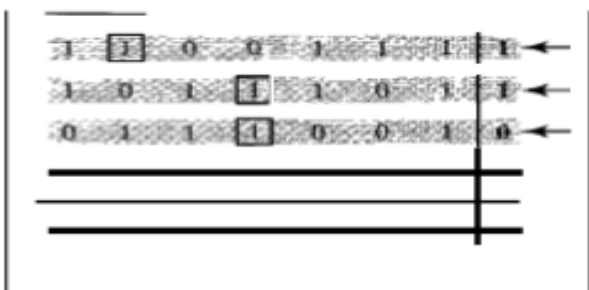
a. Design of row and column parities



b. One error affects two parities



c. Two errors affect two parities



d. Three errors affect four parities



e. Four errors cannot be detected

**Q** Data transmitted on a link uses the following 2D parity scheme for error detection: Each sequence of 28 bits is arranged in a  $4 \times 7$  matrix (rows  $r_0$  through  $r_3$ , and columns  $d_7$  through  $d_1$ ) and is padded with a column  $d_0$  and row  $r_4$  of parity bits computed using the Even parity scheme. Each bit of column  $d_0$  (respectively, row  $r_4$ ) gives the parity of the corresponding row (respectively, column). These 40 bits are transmitted over the data link. The table shows data received by a receiver and has  $n$  corrupted bits. What is the minimum possible value of  $n$ ? (Gate-2008) (2 Marks)

(A) 1

(B) 2

(C) 3

(D) 4

	$d_7$	$d_6$	$d_5$	$d_4$	$d_3$	$d_2$	$d_1$	$d_0$
$r_0$	0	1	0	1	0	0	1	1
$r_1$	1	1	0	0	1	1	1	0
$r_2$	0	0	0	1	0	1	0	0
$r_3$	0	1	1	0	1	0	1	0
$r_4$	1	1	0	0	0	1	1	0

**Answer: (C)**

## Hamming Codes

- Now let us discuss a category of error-correcting codes called Hamming codes. These codes were originally designed with  $d_{\min} = 3$ , which means that they can detect up to two errors or correct one single error.
- Although there are some Hamming codes that can correct more than one error, our discussion focuses on the single-bit error-correcting code.
- First let us find the relationship between  $n$  and  $k$  in a Hamming code. We need to choose an integer  $m \geq 3$ . The values of  $n$  and  $k$  are then calculated from  $m$  as  $n = 2^m - 1$  and  $k = n - m$ . The number of check bits  $r = m$ .
- All Hamming codes discussed in this book have  $d_{\min} = 3$ . The relationship between  $m$  and  $n$  in these codes is  $n = 2^m - 1$ .
- For example, if  $m = 3$ , then  $n = 7$  and  $k = 4$ . This is a Hamming code  $C(7, 4)$  with  $d_{\min} = 3$ . shows the datawords and codewords for this code.
- For each integer  $r \geq 2$  there is a code with block length  $n = 2^r - 1$  and message length  $k = 2^r - r - 1$ .

Bit position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Encoded data bits	p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11	p16	d12	d13	d14	d15	
Parity bit coverage	p1	X		X		X		X		X		X		X		X		X		X	
	p2		X	X			X	X			X	X			X	X			X	X	
	p4				X	X	X	X					X	X	X	X					X
	p8								X	X	X	X	X	X	X	X					
	p16																X	X	X	X	X

**Q** Consider a parity check code with three data bits and four parity check bits. Three of the code words are 0101011, 1001101 and 1110001. Which of the following are also code words? **(Gate-2004) (2 Marks)**

**I.** 0010111

**II.** 0110110

**III.** 1011010

**IV.** 0111010

**(A)** I and III

**(B)** I, II and III

**(C)** II and IV

**(D)** I, II, III and IV

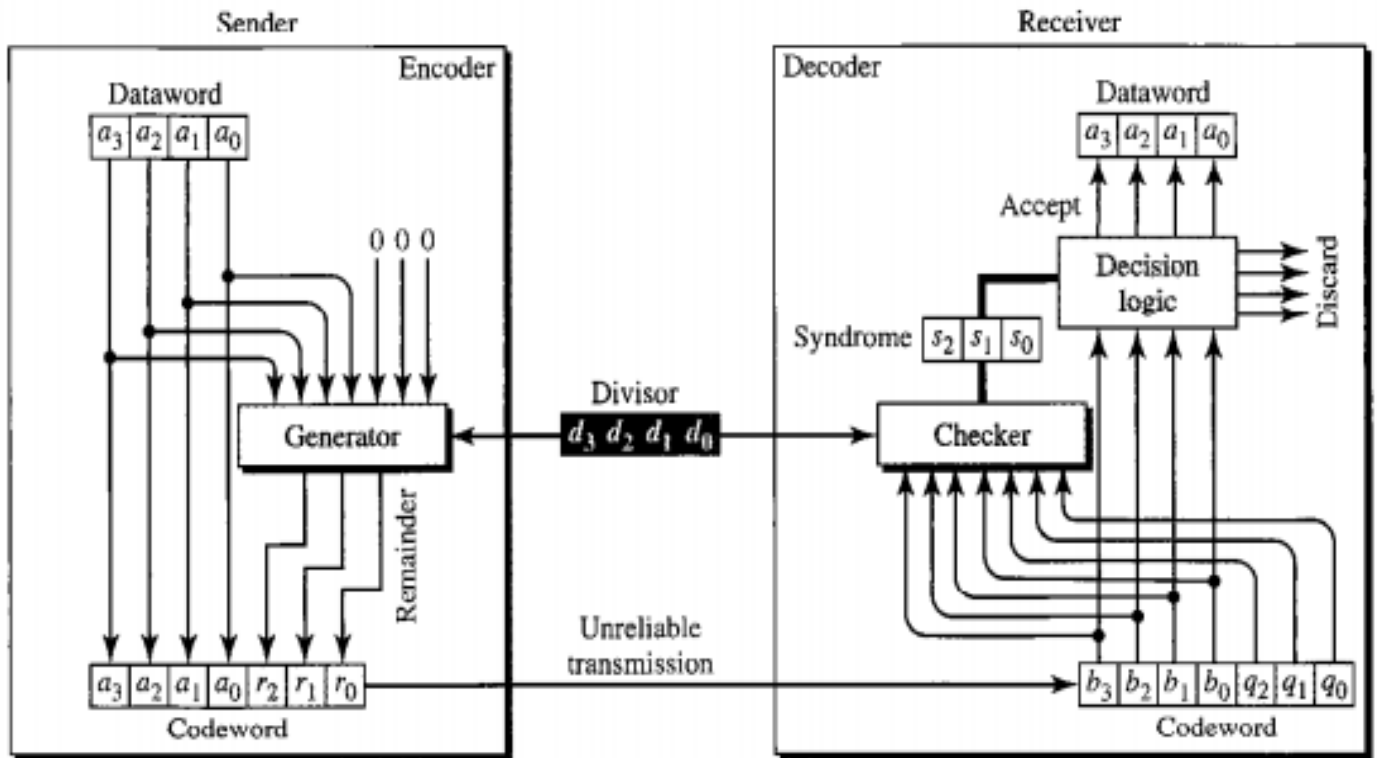
**Answer: (A)**

## CYCLIC CODES

- Cyclic codes are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.
- Example, if 1011000 is a codeword and we cyclically left-shift, then 0110001 is also a codeword.

### **Cyclic Redundancy Check**

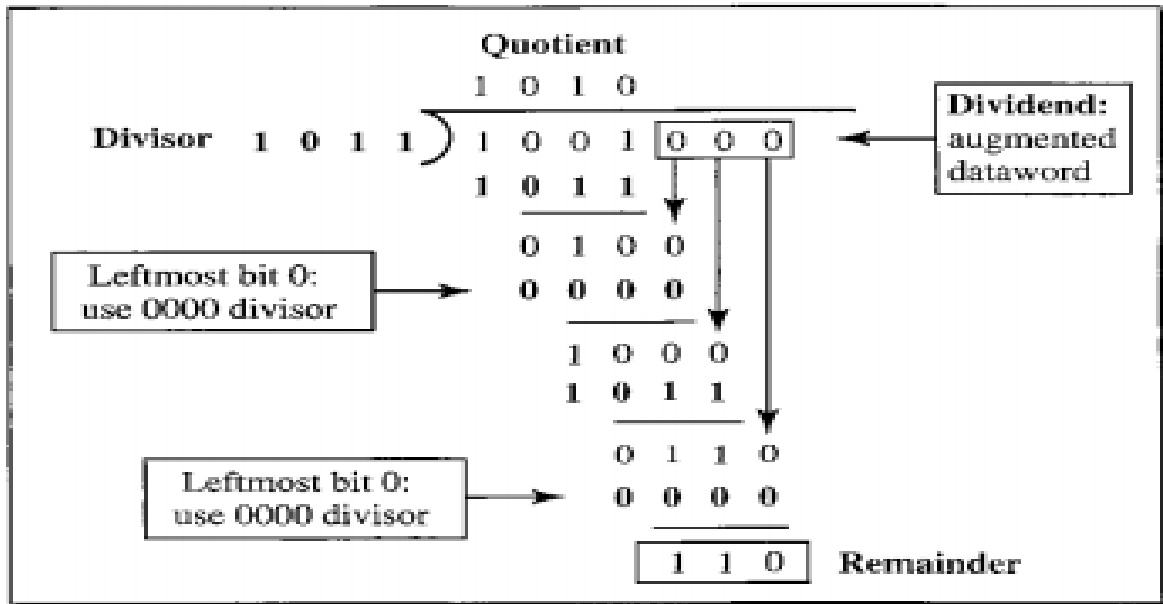
- We can create cyclic codes to correct errors. In the encoder, the dataword has  $k$  bits (4 here); the codeword has  $n$  bits (7 here).
- The size of the dataword is augmented by adding  $n - k$  (3 here) 0s to the right-hand side of the word.
- The  $n$ -bit result is fed into the generator. The generator uses a divisor of size  $n - k + 1$  (4 here), predefined and agreed upon.
- The generator divides the augmented dataword by the divisor (modulo-2 division). The quotient of the division is discarded; the remainder ( $r_2 r_1 r_0$ ) is appended to the dataword to create the codeword.
- The decoder receives the possibly corrupted codeword. A copy of all  $n$  bits is fed to the checker which is a replica of the generator.
- The remainder produced by the checker is a syndrome of  $n - k$  (3 here) bits, which is fed to the decision logic analyzer. The analyzer has a simple function. If the syndrome bits are all as, the 4 leftmost bits of the codeword are accepted as the dataword (interpreted as no error); otherwise, the 4 bits are discarded (error).



**Encoder**

Dataword **1 0 0 1**

**Division**

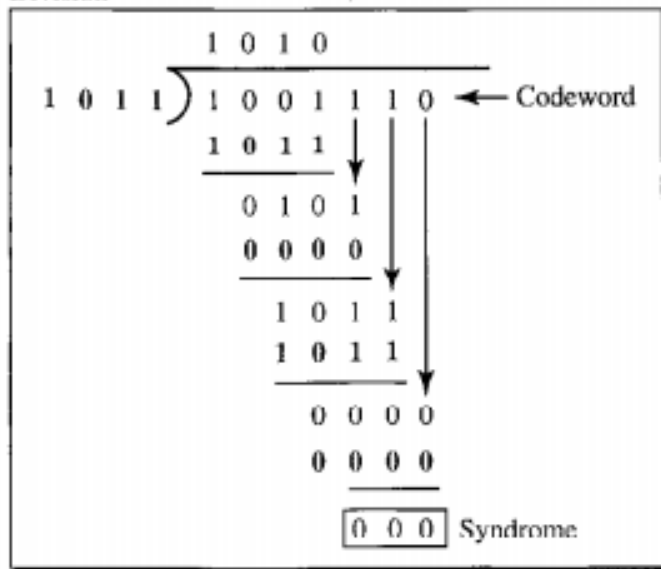


Codeword **1 0 0 1 1 1 0**  
 Dataword Remainder

**Decoder**

Codeword 1 0 0 1 | 1 1 0

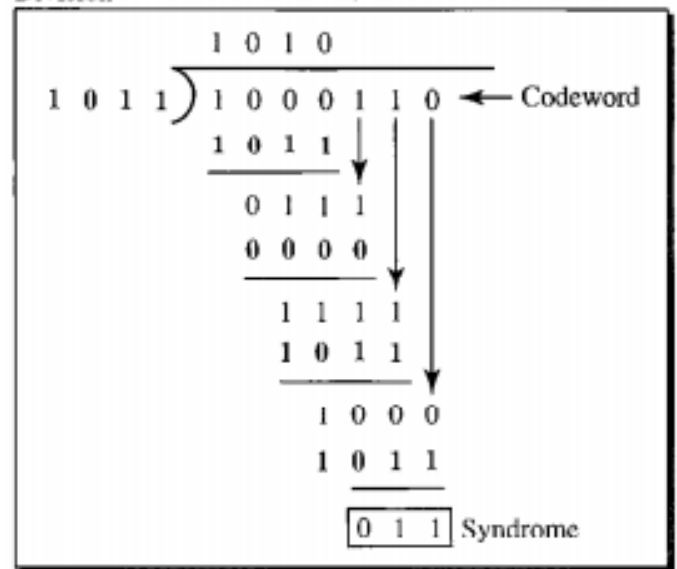
Division



Dataword accepted 1 0 0 1

Codeword 1 0 0 0 | 1 1 0

Division



Dataword discarded

Sanchit

## CHECKSUM

- The checksum is used in the Internet by several protocols although not at the data link layer.
- Suppose our data is a list of five 4-bit numbers that we want to send to a destination. In addition to sending these numbers, we send the sum of the numbers.
- For example, if the set of numbers is (7, 11, 12, 0, 6), we send (7, 11, 12, 0, 6, 36), where 36 is the sum of the original numbers. The receiver adds the five numbers and compares the result with the sum.
- If the two are the same, the receiver assumes no error, accepts the five numbers, and discards the sum. Otherwise, there is an error somewhere and the data are not accepted.
- We can make the job of the receiver easier if we send the negative (complement) of the sum, called the checksum. In this case, we send (7, 11, 12, 0, 6, -36). The receiver can add all the numbers received (including the checksum). If the result is 0, it assumes no error; otherwise, there is an error.

## One's Complement

- The previous example has one major drawback. All of our data can be written as a 4-bit word (they are less than 15) except for the checksum.
- One solution is to use one's complement arithmetic. In this arithmetic, we can represent unsigned numbers between 0 and  $2^n - 1$  using only  $n$  bits.
- If the number has more than  $n$  bits, the extra leftmost bits need to be added to the  $n$  rightmost bits (wrapping). In one's complement arithmetic, a negative number can be represented by inverting all bits (changing a 0 to a 1 and a 1 to a 0). This is the same as subtracting the number from  $2^n - 1$ .

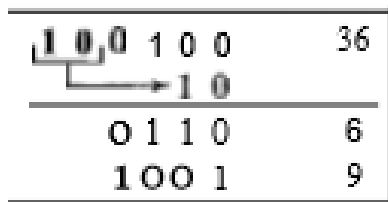
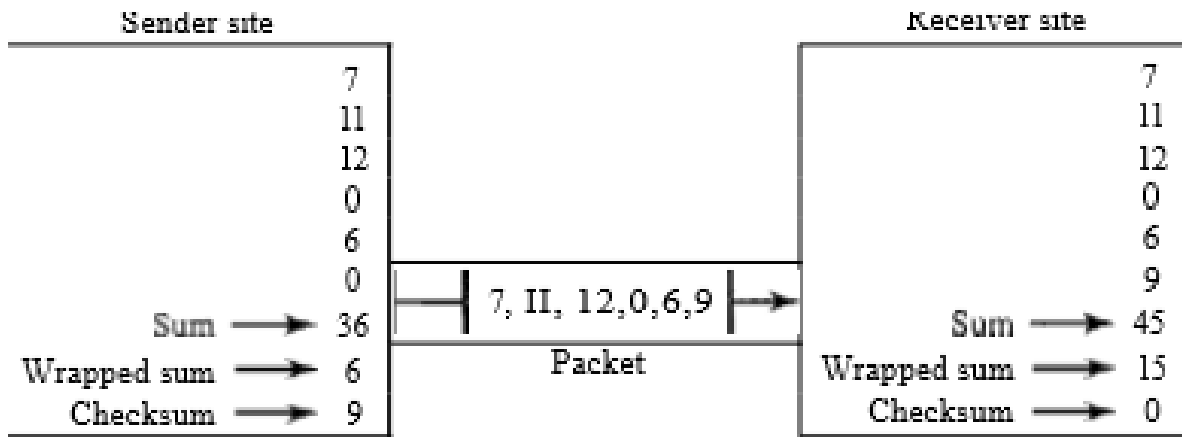
**Q** How can we represent the number 21 in one's complement arithmetic using only four bits?

**Solution** The number 21 in binary is 10101 (it needs five bits). We can wrap the leftmost bit and add it to the four rightmost bits. We have  $(0101 + 1) = 0110$  or 6.

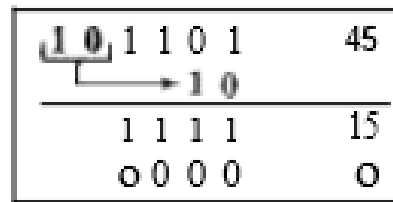
**Example 10.21**

How can we represent the number -6 in one's complement arithmetic using only four bits?

**Solution** In one's complement arithmetic, the negative or complement of a number is found by inverting all bits. Positive 6 is 0110; negative 6 is 1001. If we consider only unsigned numbers, this is 9. In other words, the complement of 6 is 9. Another way to find the complement of a number in one's complement arithmetic is to subtract the number from  $2^n - 1$  (16 - 1 in this case).



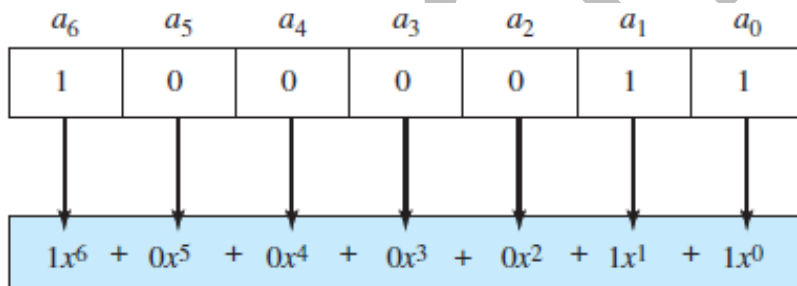
Details of wrapping and complementing



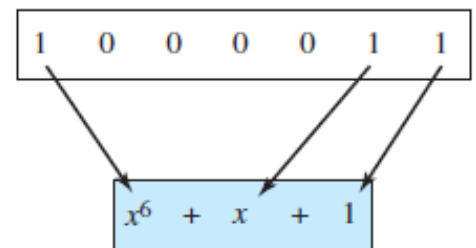
Details of wrapping and complementing

## Polynomials

- A pattern of 0s and 1s can be represented as a **polynomial** with coefficients of 0 and 1.
- The power of each term shows the position of the bit; the coefficient shows the value of the bit.



a. Binary pattern and polynomial



b. Short form

- An advantage is that a large binary pattern can be represented by short terms.

### ***Degree of a Polynomial***

- The degree of a polynomial is the highest power in the polynomial. For example, the degree of the polynomial  $x^6 + x + 1$  is 6.
- The degree of a polynomial is 1 less than the number of bits in the pattern. The bit pattern in this case has 7 bits.

### ***Adding and Subtracting Polynomials***

- Adding and subtracting polynomials in mathematics are done by adding or subtracting the coefficients of terms with the same power.
- Addition and subtraction are the same.
- Adding or subtracting is done by combining terms and deleting pairs of identical terms.

Example: Adding  $x^5 + x^4 + x^2$  and  $x^6 + x^4 + x^2$  gives us:

$$x^6 + x^5 \text{ (i.e. we delete the pairs of identical terms)}$$

#### Note:

- If we add, for example, three polynomials and we get  $x^2$  three times, we delete a pair of them and keep the third.

#### Multiplying or Dividing Terms

- In multiplying a term we just add the powers. Example,  $x^3 \times x^4$  is  $x^7$ .
- For dividing, we just subtract the power of the second term from the power of the first. Example,  $x^5/x^2$  is  $x^3$ .

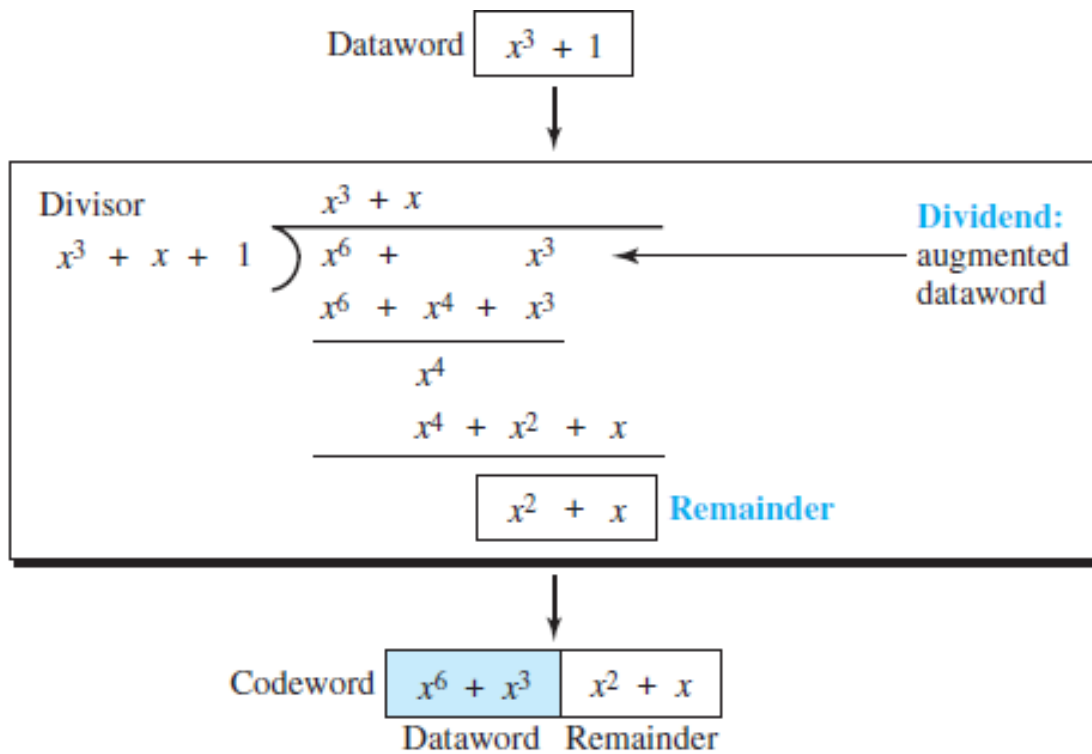
#### Shifting

**Shifting left 3 bits:** 10011 becomes 10011000       $x^4 + x + 1$  becomes  $x^7 + x^4 + x^3$

**Shifting right 3 bits:** 10011 becomes 10       $x^4 + x + 1$  becomes  $x$

#### Cyclic Code Encoder Using Polynomials

- The dataword 1001 is represented as  $x^3 + 1$ . The divisor 1011 is represented as  $x^3 + x + 1$ .
- To find the augmented dataword, we have left-shifted the dataword 3 bits (multiplying by  $x^3$ ).
- The result is  $x^6 + x^3$ . Division is straightforward. We divide the first term of the dividend,  $x^6$ , by the first term of the divisor,  $x^3$ .
- The first term of the quotient is then  $x^6/x^3$ , or  $x^3$ . Then we multiply  $x^3$  by the divisor and subtract (according to our previous definition of subtraction) the result from the dividend.
- The result is  $x^4$ , with a degree greater than the divisor's degree; we continue to divide until the degree of the remainder is less than the degree of the divisor.



- The divisor in a cyclic code is normally called the *generator polynomial* or simply the *generator*.

**Q** The message 11001001 is to be transmitted using the CRC polynomial  $x^3+1$  to protect it from errors. The message that should be transmitted is: **(Gate-2007) (2 Marks)**

- |                       |                        |
|-----------------------|------------------------|
| <b>a)</b> 11001001000 | <b>b)</b> 11001001011  |
| <b>c)</b> 11001010    | <b>d)</b> 110010010011 |

**ANSWER B**

## Cyclic Code Analysis

- Let us assume: **Dataword:  $d(x)$ , Codeword:  $c(x)$ , Generator:  $g(x)$ , Syndrome:  $s(x)$ , Error:  $e(x)$**
- If  $s(x)$  is not zero, then one or more bits is corrupted. However, if  $s(x)$  is zero, either no bit is corrupted or the decoder failed to detect any errors.
- **Received codeword =  $c(x) + e(x)$**   
Dividing both sides by  $g(x)$

$$\frac{\text{Received codeword}}{g(x)} = \frac{c(x)}{g(x)} + \frac{e(x)}{g(x)}$$

In a cyclic code, those  $e(x)$  errors that are divisible by  $g(x)$  are not caught.

### Points to Note about catching errors:

- If the generator has more than one term and the coefficient of  $x^0$  is 1, all single-bit errors can be caught.
- If a generator cannot divide  $x^t + 1$  ( $t$  between 0 and  $n - 1$ ), then all isolated double errors can be detected.
- A generator that contains a factor of  $x + 1$  can detect all odd-numbered errors.
- All burst errors with  $L \leq r$  will be detected.
- All burst errors with  $L = r + 1$  will be detected with probability  $1 - (1/2)^{r-1}$ .
- All burst errors with  $L > r + 1$  will be detected with probability  $1 - (1/2)^r$ .

### Advantages of Cyclic Codes

- Cyclic codes have a very good performance in detecting single-bit errors, double errors, an odd number of errors, and burst errors.
- They can easily be implemented in hardware and software.
- They are especially fast when implemented in hardware.

### Error Control

- Error control at the data-link layer is normally very simple and implemented using one of the two methods.
- In both methods, a CRC is added to the frame header by the sender and checked by the receiver.

1. In the first method, if the frame is corrupted, it is silently discarded; if it is not corrupted, the packet is delivered to the network layer.
2. In the second method, if the frame is corrupted, it is silently discarded; if it is not corrupted, an acknowledgment is sent (for the purpose of both flow and error control) to the sender.

**Q** A computer network uses polynomials over GF (2) for error checking with 8 bits as information bits and uses  $x^3 + x + 1$  as the generator polynomial to generate the check bits. In this network, the message 01011011 is transmitted as **(Gate-2017) (2 Marks)**

**(a)** 01011011010

**(b)** 01011011011

**(c)** 01011011101

**(d)** 01011011100

**Ans: c**

**Q** Consider the following message  $M = 1010001101$ . The cyclic redundancy check (CRC) for this message using the divisor polynomial  $x^5 + x^4 + x^2 + 1$  is: **(Gate-2005) (2 Marks)**

**(A)** 01110

**(B)** 0101

**(C)** 10101

**(D)** 10110

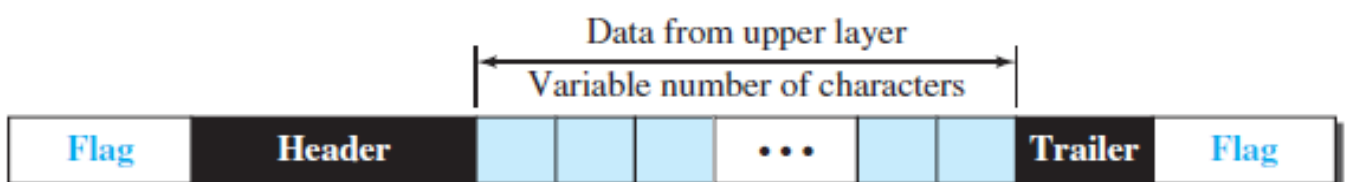
**Answer: (A)**

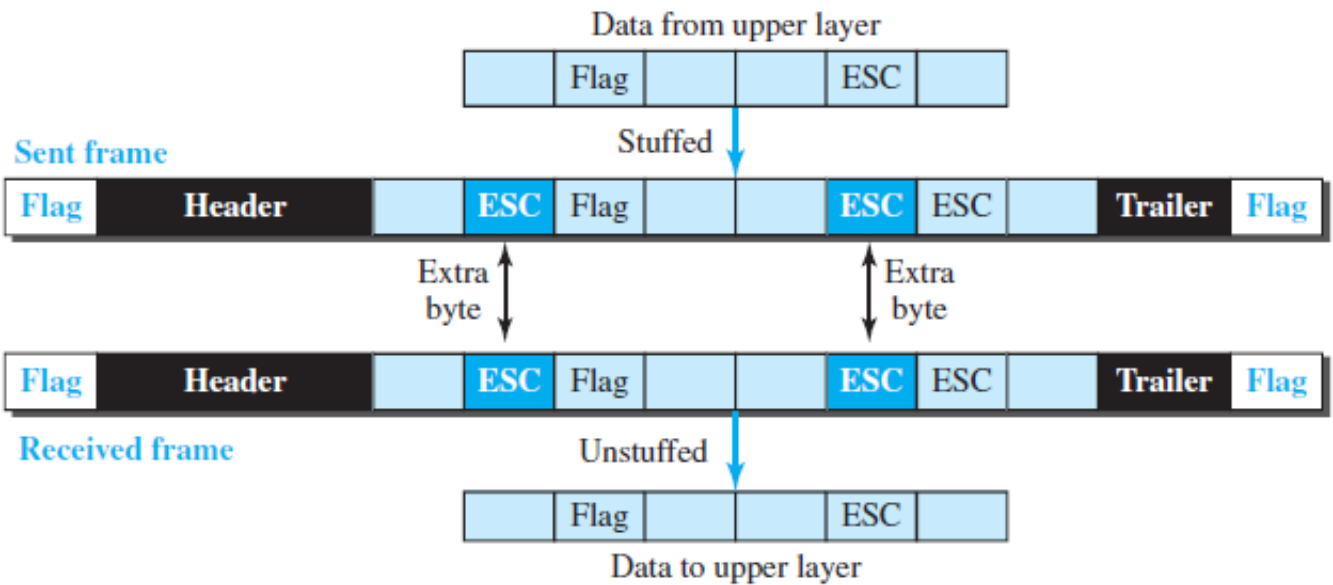
## FRAMING

- The data link layer, needs to pack bits into frames, so that each frame is distinguishable from another. Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter. In addition, each envelope defines the sender and receiver addresses since the postal system is a many-to-many carrier facility.
- Although the whole message could be packed in one frame, that is not normally done. One reason is that a frame can be very large, making flow and error control very inefficient. When a message is carried in one very large frame, even a single-bit error would require the retransmission of the whole message. When a message is divided into smaller frames, a single-bit error affects only that small frame.
- Fixed-Size Framing
  - Frames can be of fixed or variable size. In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter.
- Variable-Size Framing
  - In variable-size framing, we need a way to define the end of the frame and the beginning of the next. Historically, two approaches were used for this purpose: a character-oriented approach and a bit-oriented approach.

## Character-Oriented Protocols

- In a character-oriented protocol, data to be carried are 8-bit characters from a coding system such as ASCII. The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection or error correction redundant bits, are also multiples of 8 bits. To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame. The flag, composed of protocol-dependent special characters, signals the start or end of a frame.
- Character-oriented framing was popular when only text was exchanged by the data link layers. The flag could be selected to be any character not used for text communication. Now, however, we send other types of information such as graphs, audio, and video. Any pattern used for the flag could also be part of the information. If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame. To fix this problem, a byte-stuffing strategy was added to character-oriented framing. In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.
- Byte stuffing by the escape character allows the presence of the flag in the data section of the frame, but it creates another problem. What happens if the text contains one or more escape characters followed by a flag? The receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame. To solve this problem, the escape characters that are part of the text must also be marked by another escape character. In other words, if the escape character is part of the text, an extra one is added to show that the second one is part of the text

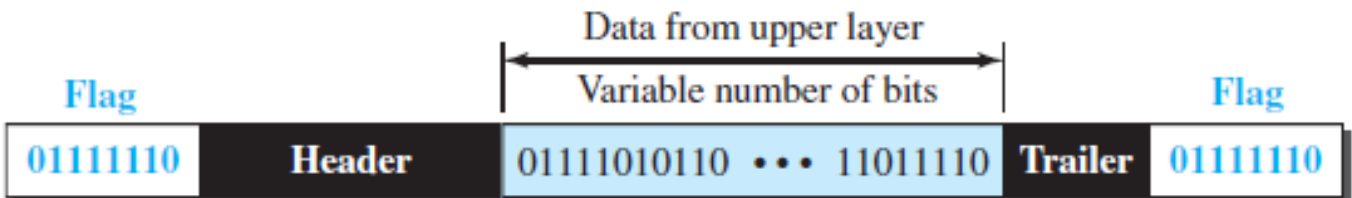




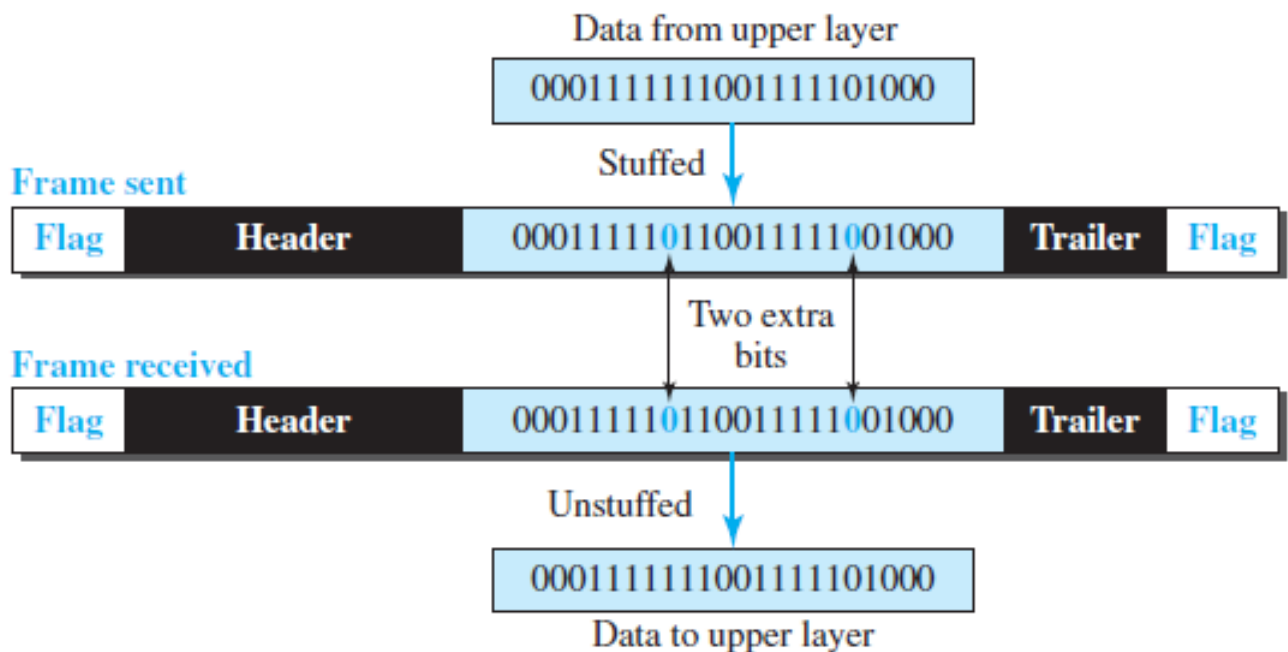
- Character-oriented protocols present another problem in data communications. The universal coding systems in use today, such as Unicode, have 16-bit and 32-bit characters that conflict with 8-bit characters. We can say that in general, the tendency is moving toward the bit-oriented protocols that we discuss next.

## Bit-Oriented Protocols

- In a bit-oriented protocol, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on. However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame,



- This flag can create the same type of problem we saw in the byte-oriented protocols. That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame. We do this by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag. The strategy is called bit stuffing. In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added. This extra stuffed bit is eventually removed from the data by the receiver. Note that the extra bit is added after one 0 followed by five 1s regardless of the value of the next bit. This guarantees that the flag field sequence does not inadvertently appear in the frame.



**Example (GATE 2014):** A bit-stuffing based framing protocol uses an 8-bit delimiter pattern of 01111110. If the output bit-string after stuffing is 01111100101, then the input bit-string is:

Ans: Since output bit is: 01111100101, that is extra bit 0 is stuffed after 0 followed by 5 consecutive one's.

01111100101  
                  └─ Stuffed 0

After removing this stuffed 0 we get:

Input bit string: 0111110101

**Q** In a data link protocol, the frame delimiter flag is given by 0111. Assuming that bit stuffing is employed, the transmitter sends the data sequence 01110110 as **(Gate-2004) (2 Marks)**

**(A)** 01101011                   **(B)** 011010110                   **(C)** 011101100                   **(D)** 0110101100

**Answer: (D)**

**Q** A bit-stuffing based framing protocol uses an 8-bit delimiter pattern of 01111110. If the output bit-string after stuffing is 01111100101, then the input bit-string is **(Gate-2014) (1 Marks)**

**A)** 0111110100

**B)** 0111110101

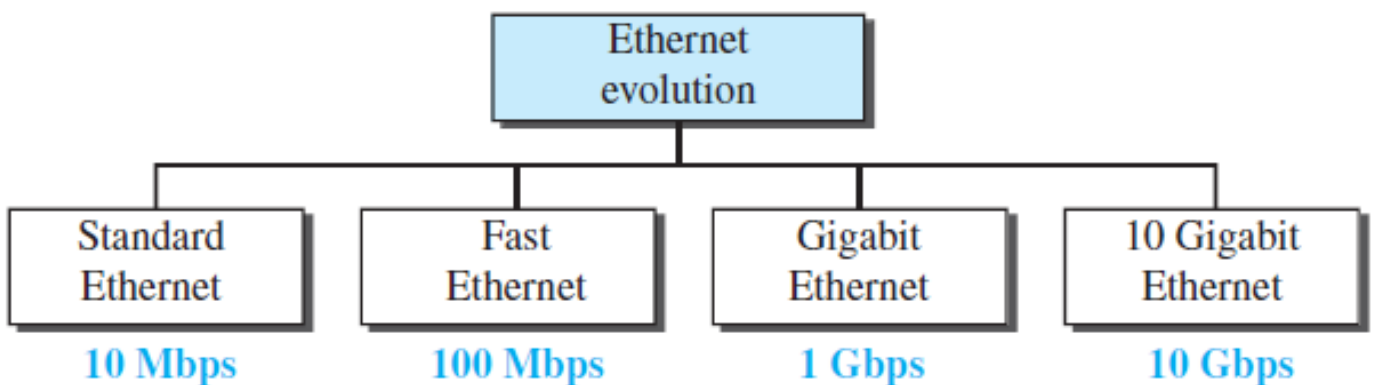
**C)** 0111111101

**D)** 0111111111

**ANSWER B**

# Ethernet

- Ethernet is a family of computer networking technologies commonly used in local area networks (LANs) and metropolitan area networks (MANs). It was commercially introduced in 1980 and first standardized in 1983 as IEEE 802.3, and has since been refined to support higher bit rates and longer link distances. Over time, Ethernet has largely replaced competing wired LAN technologies such as token ring, FDDI and ARCNET.
- The original 10BASE5 Ethernet uses coaxial cable as a shared medium, while the newer Ethernet variants use twisted pair and fibre optic links in conjunction with hubs or switches. Over the course of its history, Ethernet data transfer rates have been increased from the original 2.94 megabits per second (Mbit/s) to the latest 100 gigabits per second (Gbit/s). The Ethernet standards comprise several wirings and signalling variants of the OSI physical layer in use with Ethernet.
- Systems communicating over Ethernet divide a stream of data into shorter pieces called frames. Each frame contains source and destination addresses, and error-checking data so that damaged frames can be detected and discarded; most often, higher-layer protocols trigger retransmission of lost frames. As per the OSI model, Ethernet provides services up to and including the data link layer.
- Since its commercial release, Ethernet has retained a good degree of backward compatibility. Features such as the 48-bit MAC address and Ethernet frame format have influenced other networking protocols. The primary alternative for some uses of contemporary LANs is Wi-Fi, a wireless protocol standardized as IEEE 802.11.
- Ethernet uses Bus Topology.
- No idea of acknowledgement, if application require ack then it can send ack as a data packet.
- Encoding techniques is Manchester.
- At physical layer a packet is called Single Protocol Data Unit (SPDU)

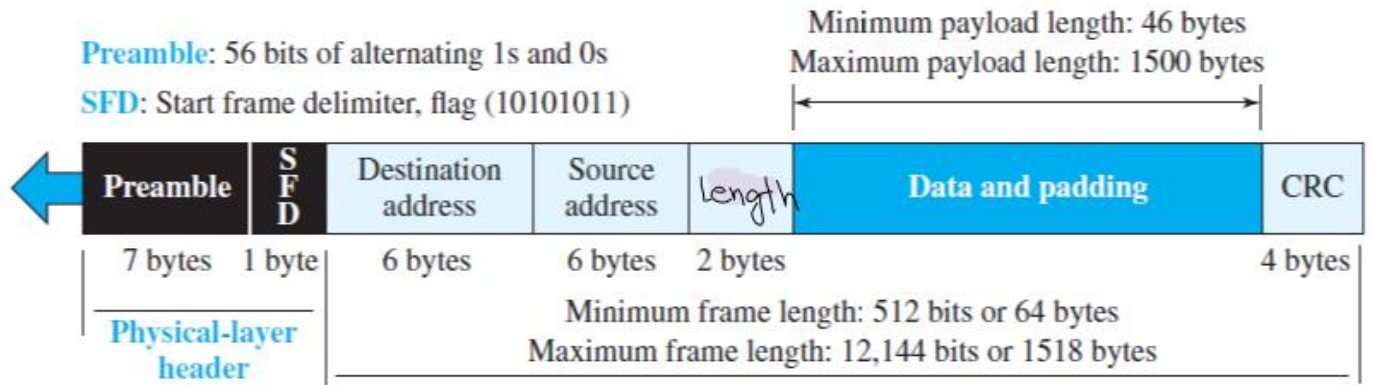


## STANDARD ETHERNET

### Connectionless and Unreliable Service:

- Each frame sent is independent of the previous or next frame. Ethernet has no connection establishment or connection termination phases.
- Ethernet is also unreliable, if a frame is corrupted during transmission and the receiver finds out about the corruption, the receiver drops the frame silently.
- In case of requirement ack can be sent separately at data packets.

## Frame Format (IEEE 802.3)



- **Preamble**

- It is a 7-byte field that contains a pattern of alternating 0's and 1's.
- It alerts the stations that a frame is going to start.
- It also enables the sender and receiver to establish bit synchronization.
- The Preamble field is added at the physical layer.

- **Start Frame Delimiter (SFD)**

- It is a 1-byte field which is always set to 10101011.
- The last two bits "11" indicate the end of Start Frame Delimiter and marks the beginning of the frame.
- The SFD field is also added at the physical layer.
- Initial only SFD was there Preamble was added later

- **Destination Address**

- It is a 6-byte field that contains the MAC address of the destination for which the data is destined. e.g. 2D : 8A : 7B : C5
- MAC address is present on NIC card.
- MAC address can be of three types
  - Unicast-LSB of the first byte is 0 (Source address will always be unicast)
  - Multicast- LSB of the first byte is 1, if we want to send, repeated messages to a group of station on the network then we can group these stations together and can assign a Multicast address to the group.
  - Broadcast-all bit are assigned 1's

- **Source Address**

- It is a 6-byte field that contains the MAC address of the source which is sending the data.

- Using some protocol, we can broadcast a request message asking MAC address of every other station in the network.
- **Length**
  - As ethernet use variable size frames therefore we need Length field
  - It is a 16-bit field.
- **Data**
  - It is a variable length field which contains the actual data, also called as a payload field.
  - The length of this field lies in the range [46 bytes, 1500 bytes], i.e. in an Ethernet frame, minimum data has to be 46 bytes and maximum data can be 1500 bytes.
  - If it is less than 46 bytes, it needs to be padded with extra 0s.
  - If more than 1500 bytes, it should be fragmented and encapsulated in more than one frame.
  - The minimum length restriction is required for the correct operation of CSMA/CD, value in general come to be 64B,  $64-6-6-2-4 = 46B$ .
  - The maximum length restriction has two historical reasons:
    - Memory was very expensive when Ethernet was designed; a maximum length restriction helped to reduce the size of the buffer.
    - The maximum length restriction prevents one station from monopolizing the shared medium, blocking other stations that have data to send.
- **CRC**
  - The last field contains error detection information.
  - At the time of transmission CRC is calculated so it is in the last.
  - It is a 4-Byte field

### **Point to Note**

- Ethernet is very simple easy to install and reconfigure.
- Should not be used with real time applications, because of collision possibility.
- if amount of data is very less then also should not be used.
- No idea of priority (Server suffer).

### **Efficiency of Standard Ethernet**

- The practical efficiency of standard Ethernet has been measured to be:

- **Efficiency =  $1 / (1 + 6.4 * a)$**
- **$a = (\text{propagation delay}) / (\text{transmission delay})$**

Sanchit Jain



(C) the University network

(D) the Internet

**Answer: (D)**

**Q** How many 8-bit characters can be transmitted per second over a 9600 baud serial communication link using asynchronous mode of transmission with one start bit, eight data bits, two stop bits, and one parity bit ? **(Gate-2004) (1 Marks)**

(A) 600

(B) 800

(C) 876

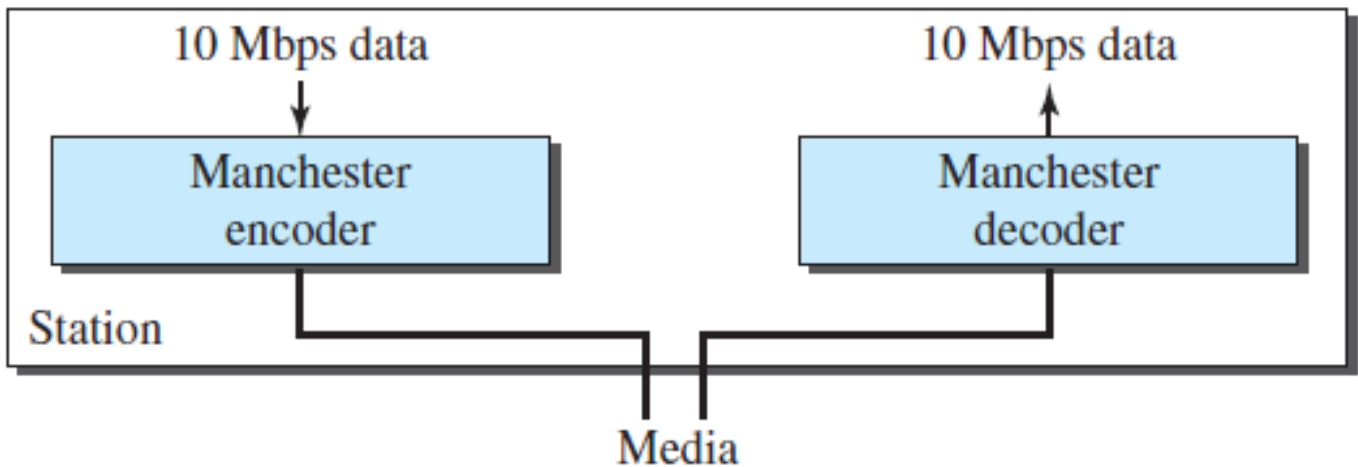
(D) 1200

**Answer: (B)**

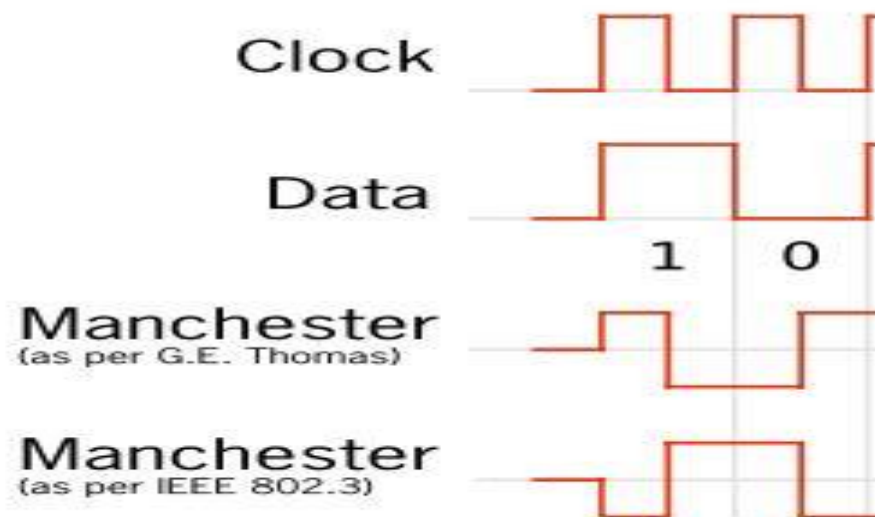
Sanchit Ja

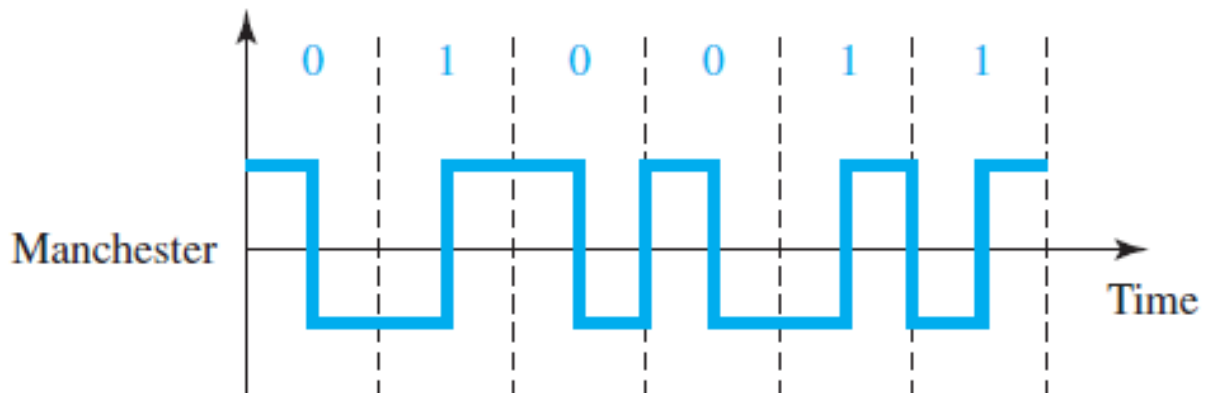
## Implementation

- At the sender, data are converted to a digital signal using the Manchester scheme; at the receiver, the received signal is interpreted as Manchester and decoded into data.

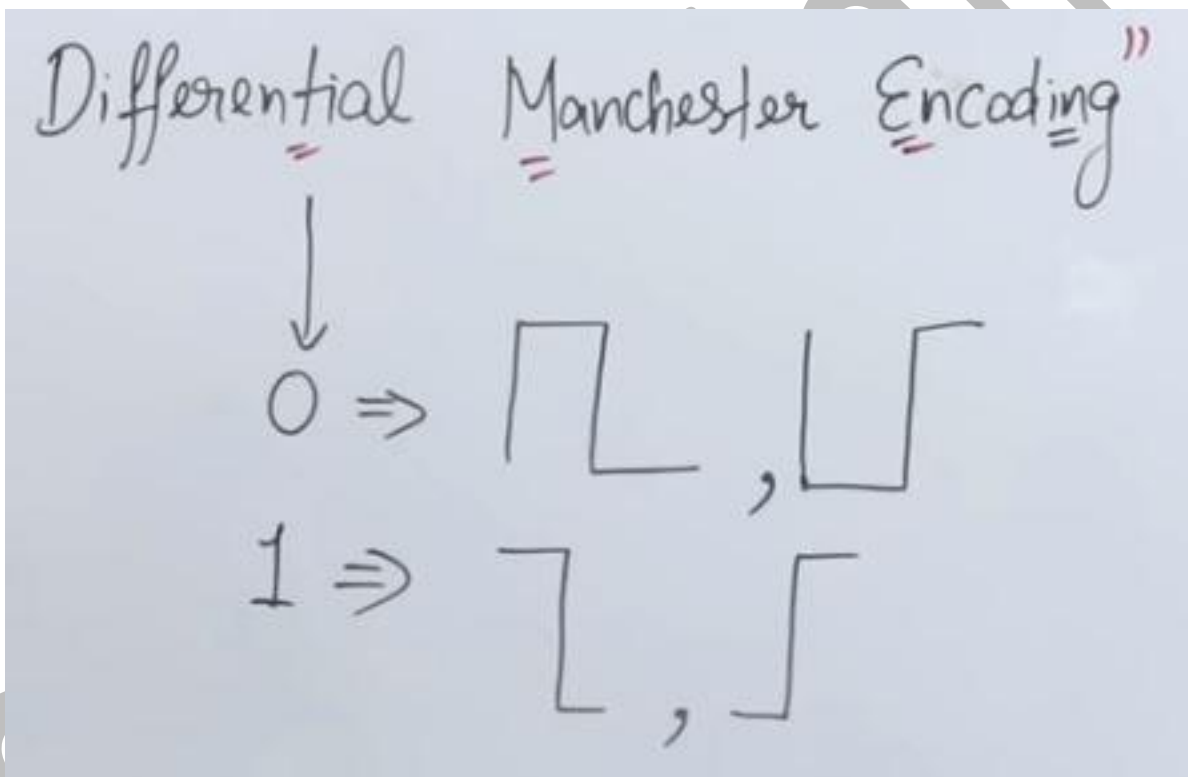


- A digital-to-digital polar encoding method in which a transition occurs at the middle of each bit interval to provide synchronization.
- In Manchester encoding, the duration of the bit is divided into two halves. The voltage remains at one level during the first half and moves to the other level in the second half. The transition at the middle of the bit provides synchronization.

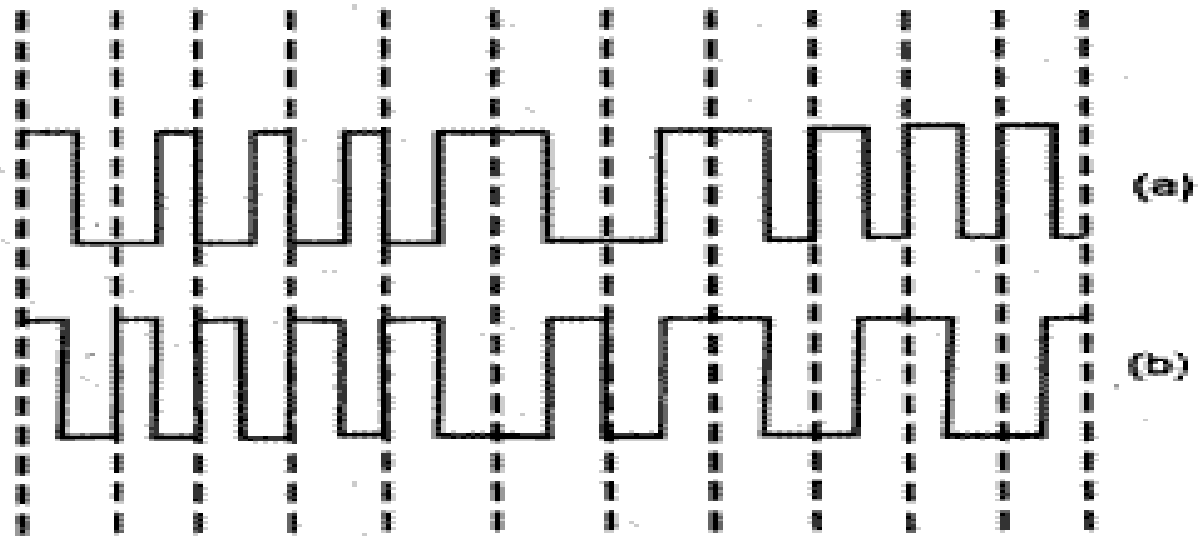




- In Manchester encoding, the bitrate is half of the baud rate.



**Q** In the waveform (a) given below, a bit stream is encoded by Manchester encoding scheme. The same bit stream is encoded in a different coding scheme in wave form (b). The bit stream and the coding scheme are **(Gate-2007) (2 Marks)**



- (A) 1000010111 and Differential Manchester respectively
- (B) 0111101000 and Differential Manchester respectively
- (C) 1000010111 and Integral Manchester respectively
- (D) 0111101000 and Integral Manchester respectively

**Answer: (A)**

**Q** In Ethernet when Manchester encoding is used, the bit rate is: **(Gate-2007) (1 Marks)**

- a) Half the baud rate.
- b) Twice the baud rate.
- c) Same as the baud rate
- d) None of the above.

**ANSWER A**

**Q** Consider that 15 machines need to be connected in a LAN using 8-port Ethernet switches. Assume that these switches do not have any separate uplink ports. The minimum number of switches needed is \_\_\_\_\_. **(Gate-2019) (1 Marks)**

**Ans: 3**

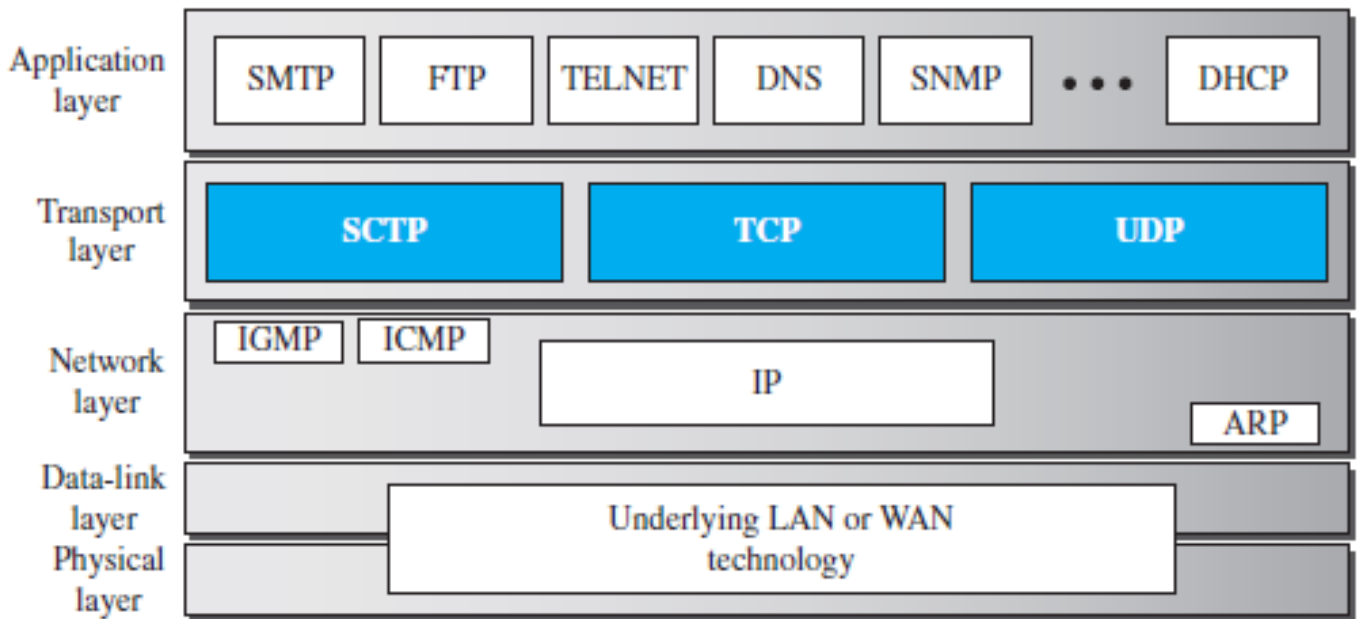
## Network layer

- The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links) i.e. through internet.

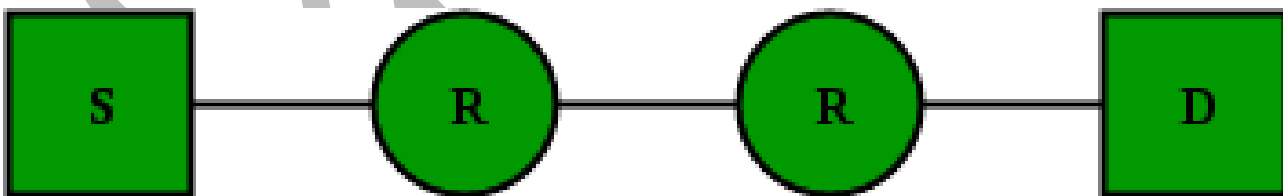
### NETWORK-LAYER SERVICES

- **Logical addressing:** If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems, the logical addresses of the sender and receiver.
- 
- **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. One of the functions of the network layer is to provide this mechanism. Network layer is responsible for routing the packet from its source to the destination.
- There is more than one route from the source to the destination.
- The network layer is responsible for finding the best one routes using routing protocols.
- **Packetizing:** Encapsulating the payload (data received from upper layer) in a network-layer packet at the source and decapsulating the payload from the network-layer packet at the destination.
- Adds a header that contains the source and destination addresses and some other information that is required by the network-layer protocol and delivers the packet to the data-link layer.
- The source is not allowed to change the content of the payload unless it is too large for delivery and needs to be fragmented.
- The routers in the path are not allowed to decapsulate the packets they received unless the packets need to be fragmented.

- **Error Control:** Error control is not directly provided in the Network layer, but checksum is added in datagram to control any corruption in header, but not in whole datagram. Although we use a protocol ICMP which provides some level of error control.
- **Flow Control:** Network Layer does not directly provide any flow control, the job of the network layer at the receiver is so simple that it may rarely be overwhelmed.
- **Congestion Control:** Congestion in the network layer is a situation in which too many datagrams are present in an area of the Internet. Congestion may occur if the number of datagrams sent by source computers is beyond the capacity of the network or routers. Leaky bucket, Token bucket can be used.



**Q** Assume that source S and destination D are connected through two intermediate routers labelled R. Determine how many times each packet has to visit the network layer and the data link layer during a transmission from S to D. **(GATE-2013) (1 Marks)**



- (A)** Network layer – 4 times and Data link layer – 4 times
- (B)** Network layer – 4 times and Data link layer – 3 times
- (C)** Network layer – 4 times and Data link layer – 6 times
- (D)** Network layer – 2 times and Data link layer – 6 times

**Answer: (C)**

**Q** Which one of the following statements is FALSE? (**GATE-2004**) (**1 Marks**)

**(A)** Packet switching leads to better utilization of bandwidth resources than circuit switching.

**(B)** Packet switching results in less variation in delay than circuit switching.

**(C)** Packet switching requires more per packet processing than circuit switching

**(D)** Packet switching can lead to reordering unlike in circuit switching

**Answer: (B)**

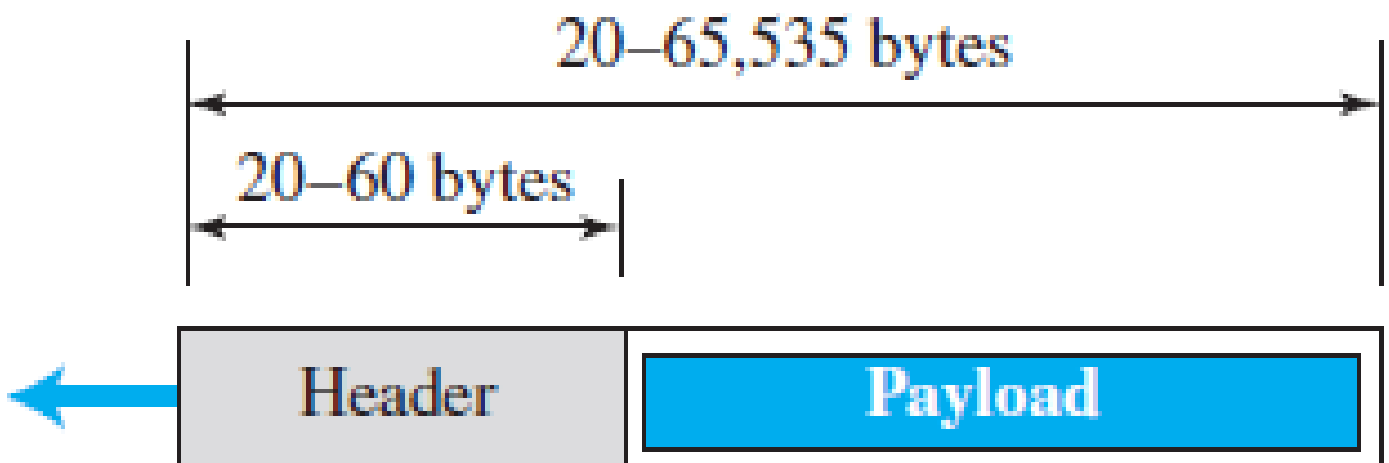
Sanchit Jain

## IPv4

- IPv4 is an **unreliable connectionless datagram protocol**—a best-effort delivery service.
- The term *best-effort* means that IPv4 packets can be corrupted, maybe lost, arrive out of order, or be delayed, and may create congestion for the network.
- **datagram** approach means Each datagram (Packet) is handled independently, and each datagram can follow a different route to the destination.
- If reliability is important, IPv4 must be paired with a reliable protocol such as TCP, so the delivery mechanism used is TCP/IP protocols.

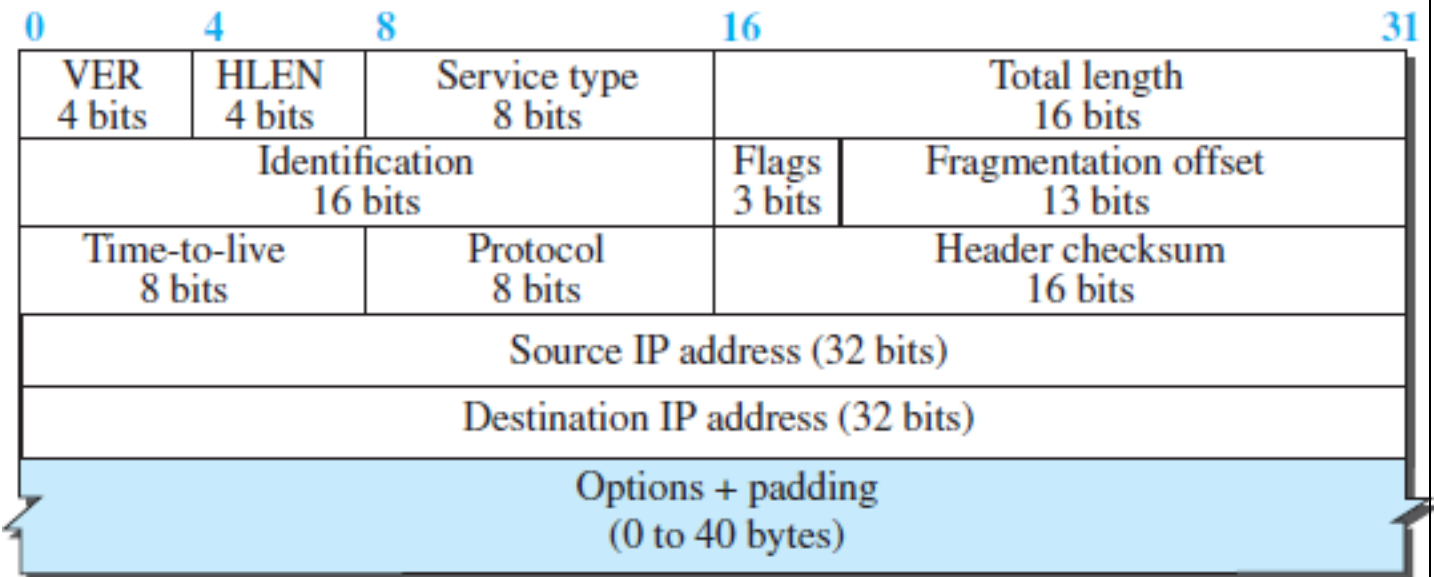
### Datagram Format

- Packets used by the IP are called **datagrams**.
- A datagram is a variable-length packet consisting of two parts: header and payload (data).
- The header is 20 to 60 bytes in length and contains information essential to routing and delivery.



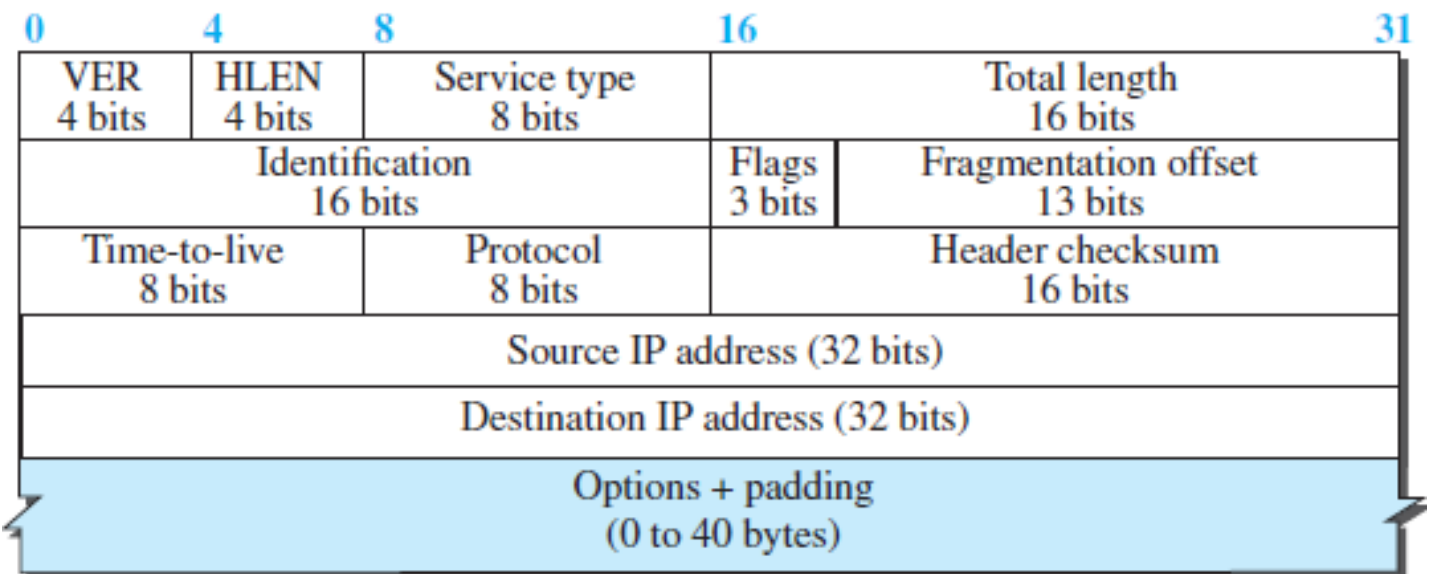
IP Datagram

## Header



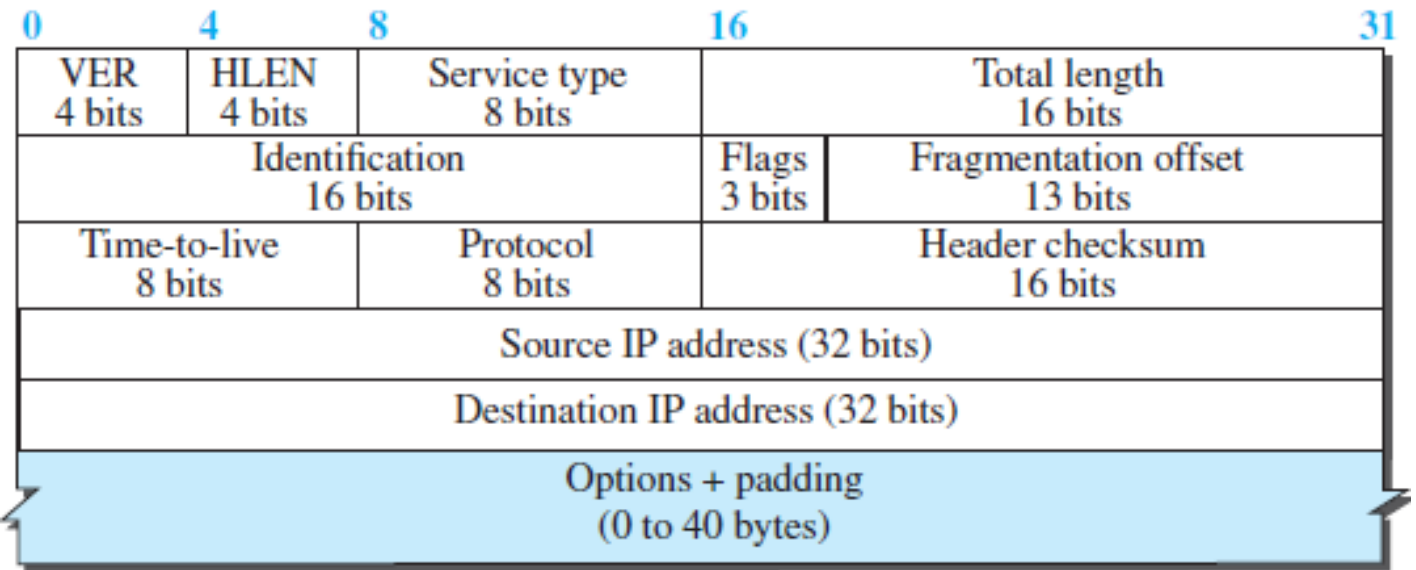
- **Version Number.** The 4-bit version number (VER) field defines the version of the IPv4 protocol, which, has the value of 4.

## Header Length



- **Header Length.** The 4-bit header length (HLEN) field defines the total length of the datagram header. The IPv4 datagram has a variable-length header.
- **Scaling Factor:**
  - To make the value of the header length (number of bytes) fit in a 4-bit header length, the total length of the header is calculated as 4-byte words.
  - The total length is divided by 4 and the value is inserted in the field.
  - The receiver needs to multiply the value of this field by 4 to find the total length.
  - Example: If header length field contains decimal value 5 (represented as 0101), then - Header length =  $5 \times 4 = 20$  bytes
- **Point to Note**
  - The length of IP header always lies in the range of [20 bytes, 60 bytes]
  - The initial 5 rows of the IP header are always used. So, **minimum length of IP header** =  $5 \times 4$  bytes = 20 bytes.
  - The size of Options field can go up to 40 bytes. So, **maximum length of IP header** = 20 bytes + 40 bytes = 60 bytes.
  - The range of header length field value is always [5, 15] as [ $20/4 = 5$ ,  $60/4 = 15$ ]
  - The range of header length is always [20, 60].

## Service Type



- **Service Type.** It defines how the datagram should be handled. Service type is an 8-bit field that is used for Quality of Service (QoS).
- IETF has changed the interpretation and name of this 8-bit field. This field, previously called service type, is now called differentiated services.
- Precedence is a 3-bit subfield ranging from 0 (000 in binary) to 7 (111 in binary). The precedence defines the priority of the datagram in issues such as congestion. If a router is congested and needs to discard some datagrams, those datagrams with lowest precedence are discarded first.
- TOS bits is a 4-bit subfield with each bit having a special meaning. Although a bit can be either 0 or 1, one and only one of the bits can have the value of 1 in each datagram

D: Minimize delay

R: Maximize reliability

T: Maximize throughput

C: Minimize cost



Precedence

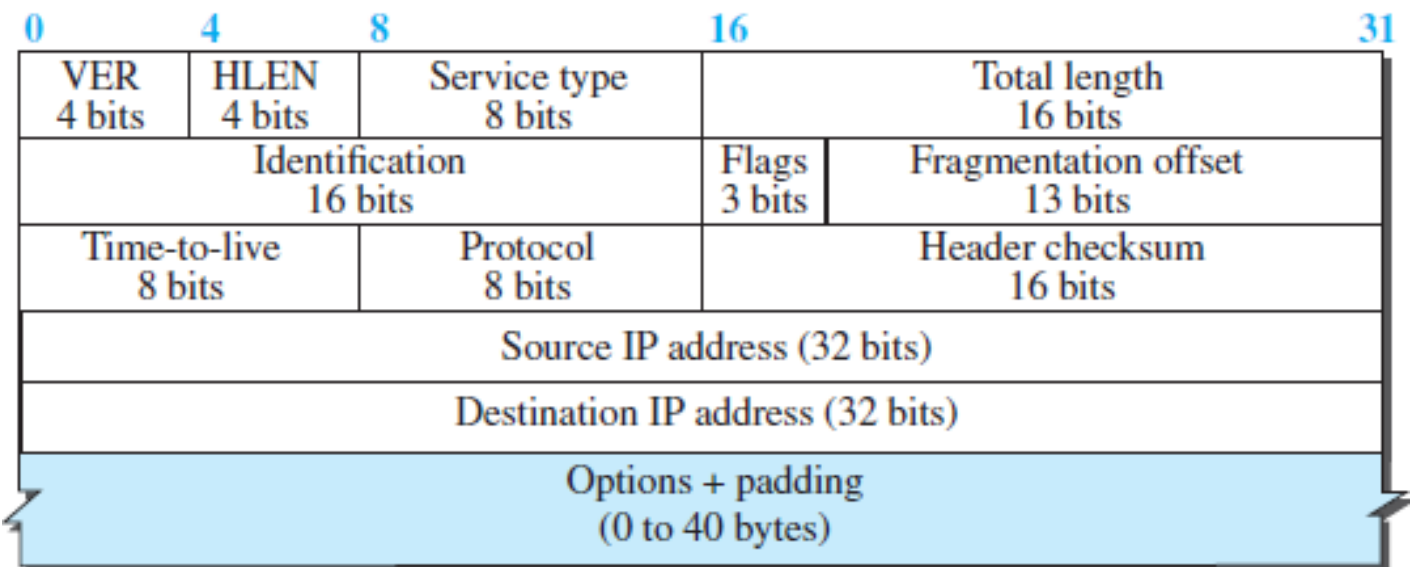
TOS bits

Service type

<i>TOS Bits</i>	<i>Description</i>
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

<i>Protocol</i>	<i>TOS Bits</i>	<i>Description</i>
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

## Total Length

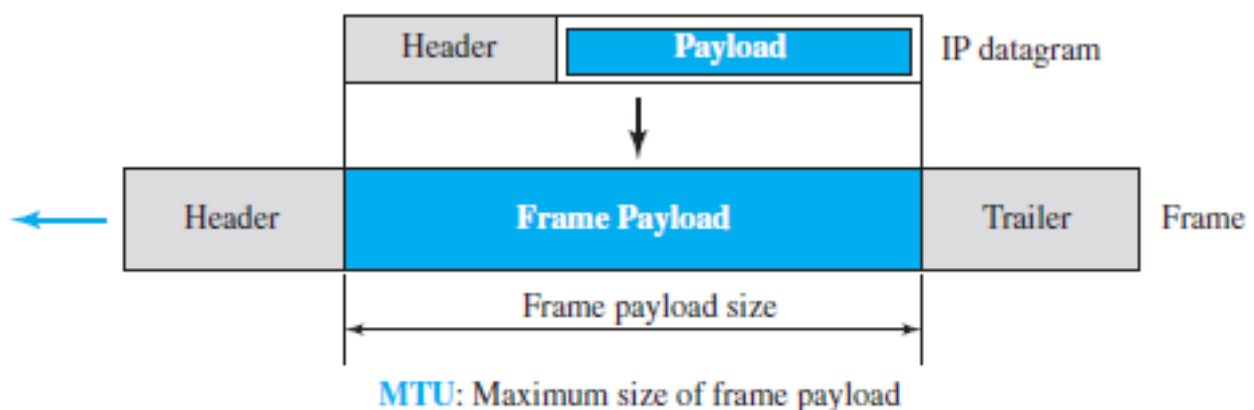


- **Total Length.** It defines the total length (header plus data) of the IP datagram in bytes. This field helps the receiving device to know when the packet has completely arrived.
- **Minimum total length of datagram** = 20 bytes (20 bytes header + 0 bytes data)
- **Maximum total length of datagram** = Maximum value of 16-bit word = 65535 bytes
- To find the length of the data coming from the upper layer, subtract the header length from the total length.
- **Length of data = total length - (HLEN) × 4**

## Maximum Transfer Unit (MTU)

- Each link-layer protocol has its own frame format.
- One of the features of each format is the maximum size of the payload that can be encapsulated.
- In other words, when a datagram is encapsulated in a frame, the total size of the datagram must be less than this maximum size.

<i>Protocol</i>	<i>MTU</i>
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



- **The value of the MTU differs from one physical network protocol to another.** For example, the value for a LAN is normally 1500 bytes, but for a WAN it can be larger or smaller.
- When a datagram is fragmented it means that the payload of the IP datagram is fragmented and each fragment has its own header with most of the fields repeated, but some have been changed such as flags, fragmentation offset, and total length and checksum is recalculated at each point.
- A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU. Thus, **datagram may be fragmented several times before it reaches the final destination.**

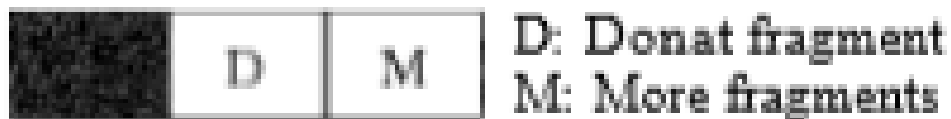
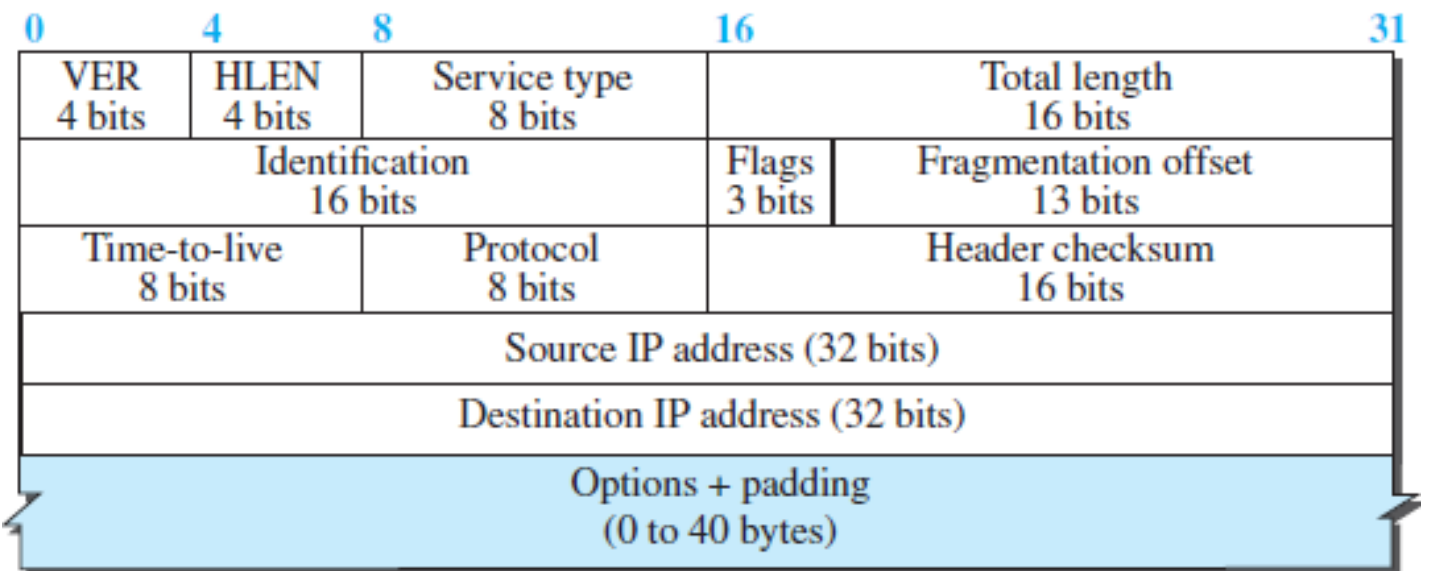
## Fragmentation

- Fragmentation is a process of dividing the datagram into fragments during its transmission.
- Datagram can be fragmented by the source host or any router in the path.
- The reassembly of the datagram, is done only by the destination host, because each fragment becomes an independent datagram.
- The fragmented datagram can travel through different routes

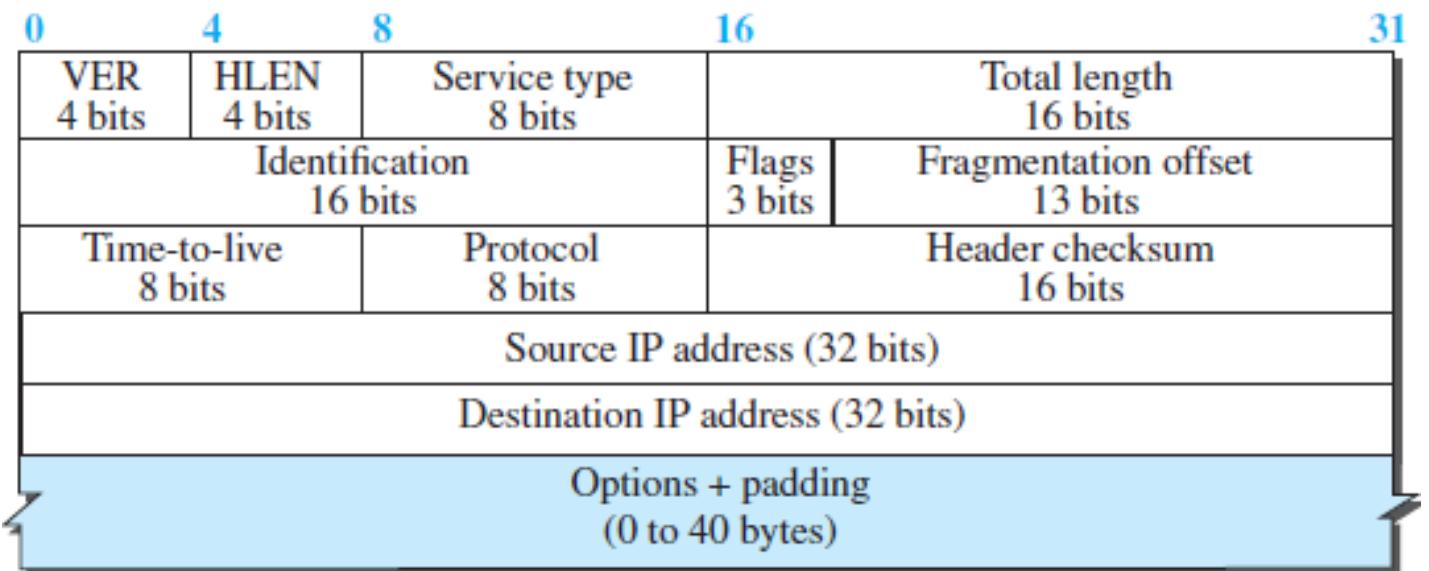
### Fields Related to Fragmentation

0	4	8	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits	Total length 16 bits	
Identification 16 bits			Flags 3 bits	Fragmentation offset 13 bits
Time-to-live 8 bits		Protocol 8 bits	Header checksum 16 bits	
Source IP address (32 bits)				
Destination IP address (32 bits)				
Options + padding (0 to 40 bytes)				

- **Identification:** 16-bit *identification field* identifies a datagram originating from the source host. To guarantee uniqueness, IP protocol uses a counter to label the datagrams.
- The counter is initialized to a positive number. When the IP protocol sends a datagram, it copies the current value of the counter to the identification field and increments the counter by one.
- When a datagram is fragmented, the value in the identification field is copied into all fragments so used for the identification of the fragments of an original IP datagram.
- The identification number helps the destination in reassembling the datagram.

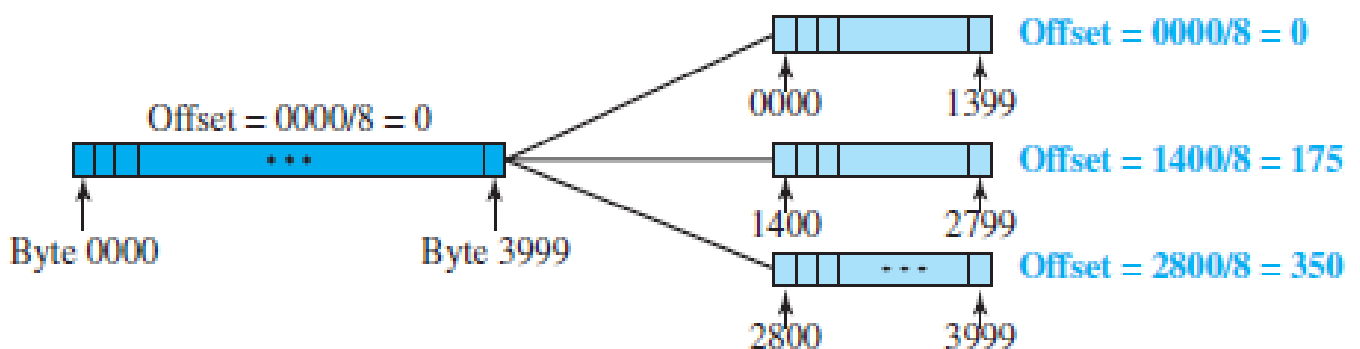


- **Flag Field:** The 3-bit *flags field* defines three flags.
  - The leftmost bit is reserved (not used).
  - The second bit (D bit) is called the *do not fragment* bit.
    - If its value is 1, the machine must not fragment the datagram.
    - If its value is 0, the datagram can be fragmented if necessary.
  - The third bit (M bit) is called the *more fragment bit*.
    - If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one.
    - If its value is 0, it means this is the last or only fragment.



- **Fragmentation Offset:** The 13-bit *fragmentation offset field* shows the relative position of this fragment with respect to the whole datagram.
- It is the offset of the data in the original datagram measured in units of 8 bytes.

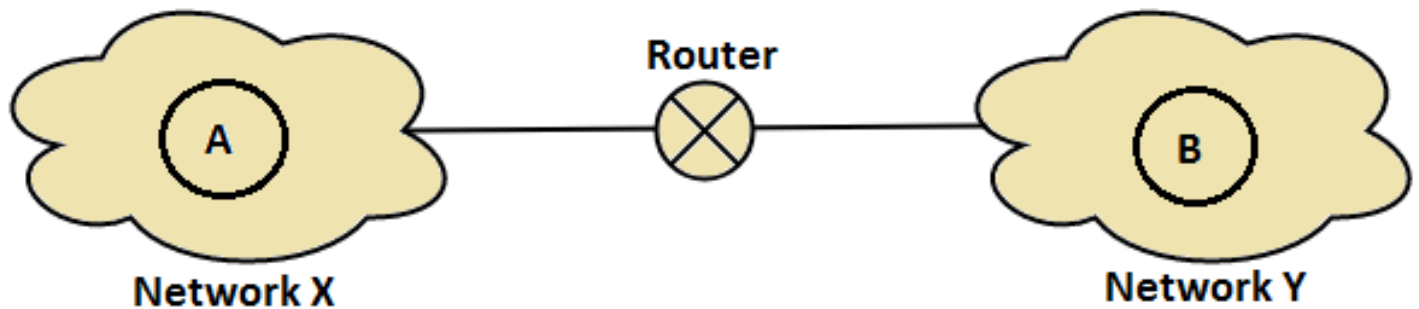
**Example:**



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 value  $\Rightarrow 0/8 = 0$ .
- The second fragment carries bytes 1400 to 2799; the offset value  $\Rightarrow 1400/8 = 175$ .
- The third fragment carries bytes 2800 to 3999. The offset value  $\Rightarrow 2800/8 = 350$ .

**Example:** Consider host A is present in network X having MTU = 520 bytes. There is another host B present in network Y having MTU = 200 bytes. Now, host A wants to send a message to host B.



- Consider that the router has received the datagram from A which have: IP Header of 20 bytes and payload length of 500 bytes and D bit is set to be 0.
- Now, router examines the MTU's of both the network and D bit and knows that it has to fragment the packet.
- The amount of data that can be sent is 200 Bytes, out of which 20 Bytes will be taken by header so the maximum data that can be sent is 180 bytes.

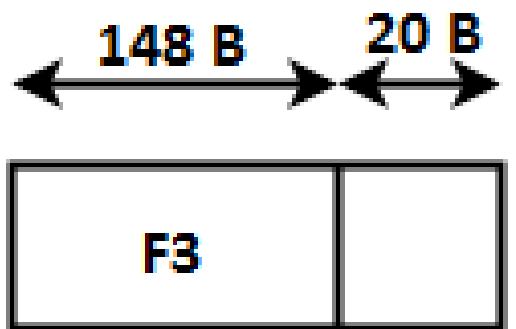
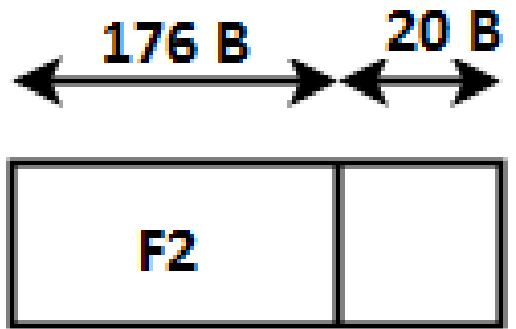
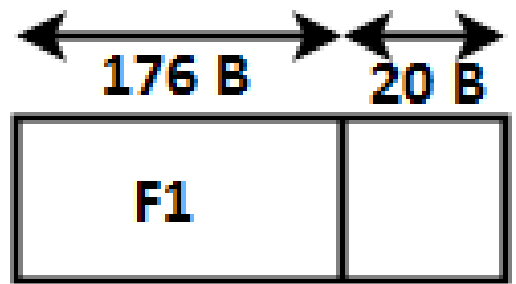
**Now certain rules that needs to be followed during fragmentation are:**

- The amount of data to be sent in one fragment is chosen in such a way that:
- It is as large as possible but should be less than or equal to MTU.
- It is a multiple of 8 so that pure decimal value can be obtained for the fragment offset field.
- It is not mandatory for the last fragment to contain the amount of data that is a multiple of 8, because it does not have to decide the fragment offset value for any other fragment.
- Using the above rules, the router is able to send maximum 176 bytes of data in one fragment as it is the greatest value that is a multiple of 8 and less than MTU.
- Now, the router will create 3 fragments of the original datagram such that:
  - **F1 = 176 Bytes**
  - **F2 = 176 Bytes**
  - **F3 = 148 Bytes**

500 B      20 B



Datagram from A



Salin

Now, the information contained in each fragment's header will be:

- **F1 Header**

- Header length field value =  $20 / 4 = 5$
- Total length field value =  $176 + 20 = 196$
- M bit = 1
- Fragment offset field value = 0
- Header checksum is recalculated.
- Identification number is same as that of original datagram.

- **F2 Header**

- Header length field value =  $20 / 4 = 5$
- Total length field value =  $176 + 20 = 196$
- M bit = 1
- Fragment offset field value =  $176 / 8 = 22$
- Header checksum is recalculated.
- Identification number is same as that of original datagram.

- **F3 Header**

- Header length field value =  $20 / 4 = 5$
- Total length field value =  $148 + 20 = 168$
- M bit = 0
- Fragment offset field value =  $(176 + 176) / 8 = 44$
- Header checksum is recalculated.
- Identification number is same as that of original datagram.

- **At destination, the receiver is receiving 3 fragmented frames, the final destination host can reassemble the original datagram from the fragments received (if none of them is lost) using the following strategy:**

- A. The first fragment has an offset field value of zero.
- B. Divide the length of the first fragment by 8. The second fragment has an offset value equal to that result.
- C. Divide the total length of the first and second fragment by 8. The third fragment has an offset value equal to that result.
- D. Continue the process. The last fragment has its M bit set to 0.
- E. Continue the process. The last fragment has a *more* bit value of 0.

## Fragmentation Overhead

- Fragmentation increases the overhead as with each fragment we have to append the header.
  - **Total Overhead = (Total number of fragmented datagrams – 1) x size of IP header**
  - **Efficiency = Data without header / data with header**
  - **Throughput = Efficiency x Bandwidth**
- **Few Important Points to Note:**
  - Source does not require fragmentation due to wise segmentation by transport layer.
  - If a datagram goes through a path where different intermediary paths are having different bandwidths. Then, while calculating the throughput, we consider the minimum bandwidth since it acts as a bottleneck.

**Example:** A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Since, M bit is set to 0 it is definitely the last fragment. We cannot know whether the packet was fragmented or not as a non-fragmented packet is considered the last fragment.

**Example:** A packet has arrived with an M bit value of 1 and a fragmentation offset value of 0. Is this the first fragment, the last fragment, or a middle fragment?

Since the M bit value is 1 it is definitely not the last fragment, and since the fragmentation offset is 0 we can conclude that it is the first fragment.

**Example:** A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?

To get the number of first byte we multiply the offset value with 8,  $100 \times 8 = 800$  but we cannot get the last value.

**Example:** A packet has arrived in which the offset value is 100, the value of HLEN is 5, and the value of the total length field is 100. What are the numbers of the first byte and the last byte?

Now, the first byte is  $100 \times 8 = 800$ .

Since the HLEN value is 5, total length of header =  $5 \times 4 = 20$  Bytes  
Total Length given is 100 Bytes out of which 20 bytes is header, which means 80 bytes are present in the datagram. If 800 is the first byte then 879 will be the last one.

**Q** Consider an IP packet with a length of 4,500 bytes that includes a 20-byte IPv4 header and a 40-byte TCP header. The packet is forwarded to an IPv4 router that supports a Maximum Transmission Unit (MTU) of 600 bytes. Assume that the length of the IP header in all the outgoing fragments of this packet is 20 bytes. Assume that the fragmentation offset value stored in the first fragment is 0. The fragmentation offset value stored in the third fragment is \_\_\_\_\_. **(Gate-2018) (2 Marks)**

**Ans: 144**

**Q** An IP datagram of size 1000 bytes arrives at a router. The router has to forward this packet on a link whose MTU (maximum transmission unit) is 100 bytes. Assume that the size of the IP header is 20 bytes. The number of fragments that the IP datagram will be divided into for transmission is \_\_\_\_\_. **(Gate-2016) (2 Marks)**

**ANSWER 13**

**Q** Host A sends a UDP datagram containing 8880 bytes of user data to host B over an Ethernet LAN. Ethernet frames may carry data up to 1500 bytes (i.e. MTU = 1500 bytes). Size of UDP header is 8 bytes and size of IP header is 20 bytes. There is no option field in IP header. How many total number of IP fragments will be transmitted and what will be the contents of offset field in the last fragment? **(Gate-2015) (2 Marks)**

**(A)** 6 and 925

**(B)** 6 and 7400

**(C)** 7 and 1110

**(D)** 7 and 8880

**Answer: (C)**

**Q** An IP router with a Maximum Transmission Unit (MTU) of 1500 bytes has received an IP packet of size 4404 bytes with an IP header of length 20 bytes. The values of the relevant fields in the header of the third IP fragment generated by the router for this packet are **(Gate-2014) (2 Marks)**

**(A)** MF bit: 0, Datagram Length: 1444; Offset: 370

**(B)** MF bit: 1, Datagram Length: 1424; Offset: 185

**(C)** MF bit: 1, Datagram Length: 1500; Offset: 37

**(D)** MF bit: 0, Datagram Length: 1424; Offset: 2960

**Answer: (A)**

**Q** In an IPv4 datagram, the M bit is 0, the value of HLEN is 10, the value of total length is 400 and the fragment offset value is 300. The position of the datagram, the sequence numbers of the first and the last bytes of the payload, respectively are **(Gate-2013) (2 Marks)**

**(A)** Last fragment, 2400 and 2789

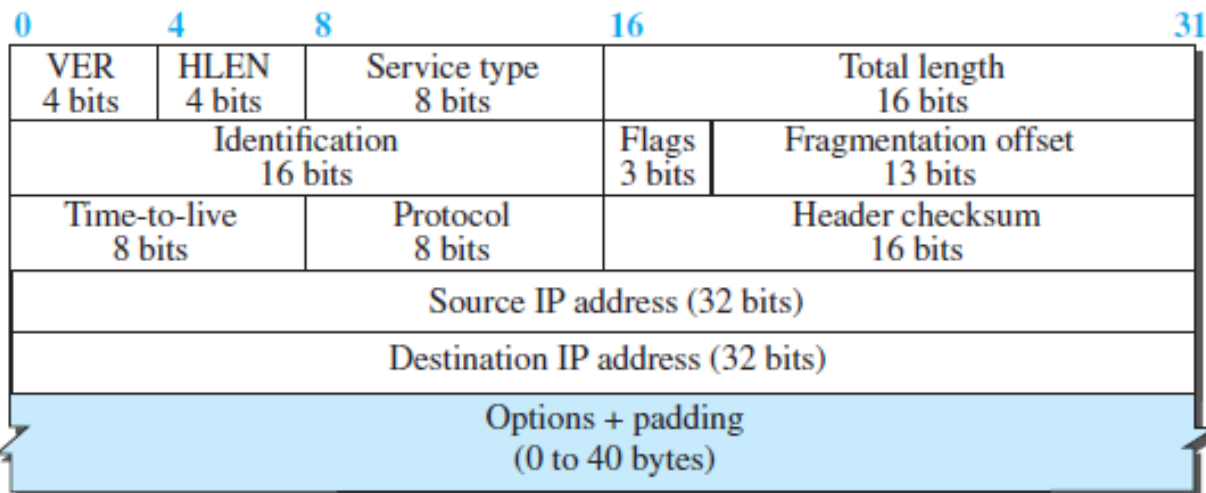
**(B)** First fragment, 2400 and 2759

**(C)** Last fragment, 2400 and 2759

**(D)** Middle fragment, 300 and 689

**Answer: (C)**

## Time-to-live



- **Time-to-live.** The time-to-live (TTL) field is used to control the maximum number of hops (routers) visited by the datagram.
- This value is approximately two times the maximum number of routers between any two hosts.
- Each router that processes the datagram decrements this number by one.
- If this value, after being decremented, is zero, the router discards the datagram.
- This field is needed because routing tables in the Internet can become corrupted. A datagram may travel between two or more routers for a long time without ever getting delivered to the destination host. This field limits the lifetime of a datagram.
- Another use of this field is to intentionally limit the journey of the packet. For example, if the source wants to confine the packet to the local network, it can store 1 in this field. When the packet arrives at the first router, this value is decremented to 0, and the datagram is discarded.

**Q** One of the header fields in an IP datagram is the Time to Live (TTL) field. Which of the following statements best explains the need for this field? (**Gate-2010**) (1 Marks)

- (A) It can be used to prioritize packets
- (B) It can be used to reduce delays
- (C) It can be used to optimize throughput
- (D) It can be used to prevent packet looping

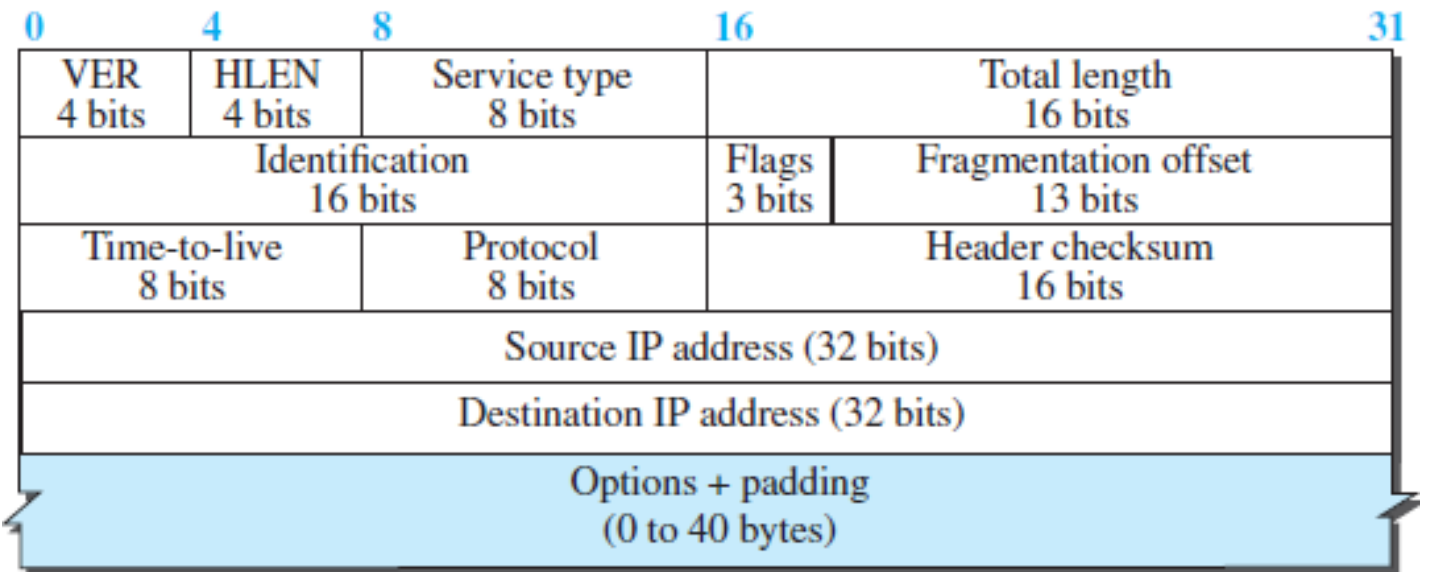
**Answer (D)**

**Q** For which one of the following reasons does Internet Protocol (IP) use the time-to-live (TTL) field in the IP datagram header (**Gate-2006**) (1 Marks)

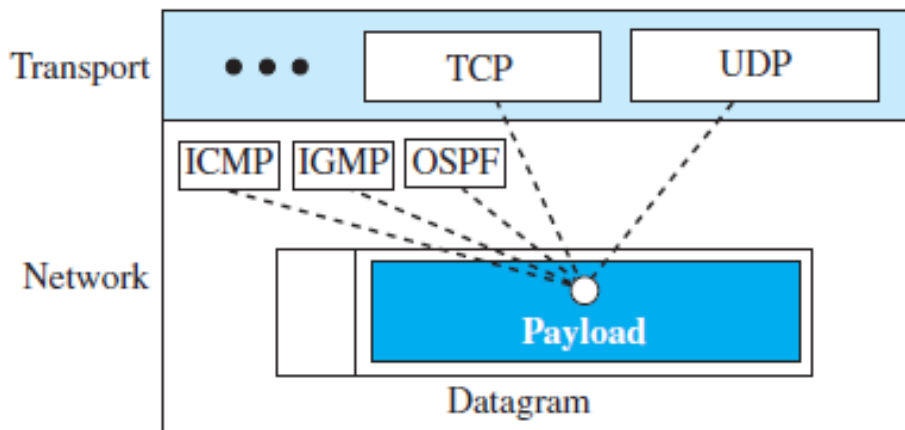
- (A) Ensure packets reach destination within that time
- (B) Discard packets that reach later than that time
- (C) Prevent packets from looping indefinitely
- (D) Limit the time for which a packet gets queued in intermediate routers.

**Answer: (C)**

## Protocol



- **Protocol.** In TCP/IP, the data section of a packet, called the payload, carries the whole packet from another protocol. A datagram, for example, can carry a packet belonging to any transport-layer protocol such as UDP or TCP.
- When the datagram arrives at the destination, the value of this field helps to define to which protocol the payload should be delivered.



### Some protocol values

ICMP	01
IGMP	02
TCP	06
UDP	17
OSPF	89

## Header checksum

0	4	8	16	31
VER 4 bits	HLEN 4 bits	Service type 8 bits	Total length 16 bits	
Identification 16 bits			Flags 3 bits	Fragmentation offset 13 bits
Time-to-live 8 bits	Protocol 8 bits		Header checksum 16 bits	
Source IP address (32 bits)				
Destination IP address (32 bits)				
Options + padding (0 to 40 bytes)				

- **Header checksum.** IP adds a header checksum field to check the header, but not the payload.
  - IP is not a reliable protocol; it does not check whether the payload carried by a datagram is corrupted during the transmission.
  - The datagram header, is added by IP, and its error-checking is the responsibility of IP.
  - Since the value of some fields, such as TTL, may change from router to router, the checksum needs to be recalculated at each router.
  - First, all higher-level protocols that encapsulate data in the IPv4 datagram have a checksum field that covers the whole packet. Therefore, the checksum for the IPv4 datagram does not have to check the encapsulated data.
  - Second, the header of the IPv4 packet changes with each visited router, but the data do not. So, the checksum includes only the part that has changed. If the data were included, each router must recalculate the checksum for the whole packet, which means an increase in processing time.

**Q** Host A (on TCP/IP v4 network A) sends an IP datagram D to host B (also on TCP/IP v4 network B). Assume that no error occurred during the transmission of D. When D reaches B, which of the following IP header field(s) may be different from that of the original datagram D? **(Gate-2014) (1 Marks)**

(i) TTL

(ii) Checksum

(iii) Fragment Offset

(A) (i) only

(B) (i) and (ii) only

(C) (ii) and (iii) only

(D) (i), (ii) and (iii)

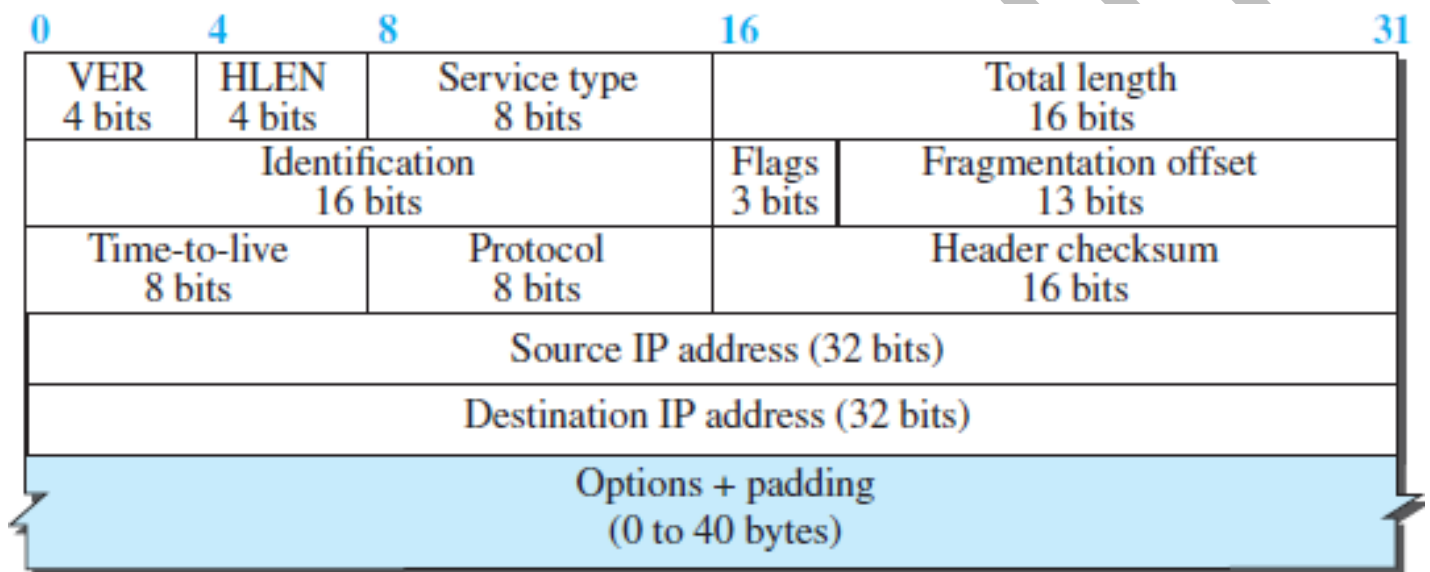
**Answer: (D)**

**Q** Which of the following statements is TRUE? (Gate-2006) (1 Marks)

- (A) Both Ethernet frame and IP packet include checksum fields
- (B) Ethernet frame includes a checksum field and IP packet includes a CRC field
- (C) Ethernet frame includes a CRC field and IP packet includes a checksum field
- (D) Both Ethernet frame and IP packet include CRC fields

**Answer: (C)**

## Source and Destination Addresses



- **Source and Destination Addresses.** These 32-bit source and destination address fields define the IP address of the source and destination respectively.
- **Options.** A datagram header can have up to 40 bytes of options. Options can be used for network testing and debugging.
  - They are not a required part of the IP header
- **Payload.** Payload, or data, is the main reason for creating a datagram. Payload is the packet coming from other protocols that use the service of IP.
  - Payload is the content of the package; the header is only the information written on the package.

**Example:** An IPv4 packet has arrived with the first 8 bits as  $(01000010)_2$ . The receiver discards the packet. Why?

- The 4 leftmost bits  $(0100)_2$  show the version, which is correct.
- The next 4 bits  $(0010)_2$  show an invalid header length ( $2 \times 4 = 8$ ). The minimum number of bytes in the header must be 20.
- The packet has been corrupted in transmission.

**Example:** In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is  $(0028)_{16}$ . How many bytes of data are being carried by this packet?

The HLEN value is 5, thus the total number of bytes in the header is  $5 \times 4 = 20$  bytes (no options).

The total length is  $(0028)_{16} = 16^1 \times 2 + 16^0 \times 8 = 40$  bytes, which means the packet is carrying 20 bytes of data ( $40 - 20$ ).

**Example:** An IPv4 packet has arrived with the first few hexadecimal digits as shown.

$(45000028000100000102...)_{16}$

How many hops can this packet travel before being dropped?

- Each hexadecimal digit i.e. Base 16 is equivalent to 4 binary digits i.e Base 2

So, each digit in hexadecimal notation defines 4 binary digits. TTL field is after 64 binary digits.

So,  $64 / 4 = 16$  hexadigits

Skipping 16 hexadigits we get, 01 as our answer (as TTL field is of 8 bits, thus will require two hexadecimal digits)

$(45000028000100000**01**02...)_{16}$

$(01)_{16} = (1)_{10}$ , so after 1 hop it will get discarded.

## Variable part

- The header of the IPv4 datagram is made of two parts: a fixed part and a variable part.
- The fixed part is 20 bytes long and was discussed in the previous section.
- The variable part comprises the options that can be a maximum of 40 bytes. Options, as the name implies, are not required for a datagram.
- They can be used for network testing and debugging. Although options are not a required part of the IPv4 header, option processing is required of the IPv4 software.
- This means that all implementations must be able to handle options if they are present in the header.
  
- **End of Option**
  - An end-of-option option is a 1-byte option used for padding at the end of the option field. It, however, can only be used as the last option.
- **Record Route**
  - A record route option is used to record the Internet routers that handle the datagram. It can list up to nine router addresses. It can be used for debugging and management purposes.
- **Strict Source Route**
  - A strict source route option is used by the source to predetermine a route for the datagram as it travels through the Internet. Dictation of a route by the source can be useful for several purposes.
  - The sender can choose a route with a specific type of service, such as minimum delay or maximum throughput.
  - Alternatively, it may choose a route that is safer or more reliable for the sender's purpose. For example, a sender can choose a route so that its datagram does not travel through a competitor's network.
  - If a datagram specifies a strict source route, all the routers defined in the option must be visited by the datagram. A router must not be visited if its IPv4 address is not listed in the datagram. If the datagram visits a router that is not on the list, the datagram is discarded and an error message is issued.
  - If the datagram arrives at the destination and some of the entries were not visited, it will also be discarded and an error message issued.
- **Loose Source Route**
  - A loose source route option is similar to the strict source route, but it is less rigid. Each router in the list must be visited, but the datagram can visit other routers as well.

○ **Timestamp**

- A timestamp option is used to record the time of datagram processing by a router. The time is expressed in milliseconds from midnight, Universal time or Greenwich mean time.
- Knowing the time, a datagram is processed can help users and managers track the behaviour of the routers in the Internet. We can estimate the time it takes for a datagram to go from one router to another. We say estimate because, although all routers may use Universal time, their local clocks may not be synchronized.

**Q** The maximum number of IPv4 router addresses that can be listed in the record route (RR) option field of an IPv4 header is \_\_\_\_\_ **(Gate-2017) (1 Marks)**

**Ans: 9**

**Q** Which one of the following fields of an IP header is NOT modified by a typical IP router? **(Gate-2015) (1 Marks)**

- (A) Checksum (B) Source address  
(C) Time to Live (TTL) (D) Length

**Answer: (B)**

**Q** Which of the following assertions is FALSE about the Internet Protocol (IP)? **(Gate-2003) (1 Marks)**

- (A) It is possible for a computer to have multiple IP addresses  
(B) IP packets from the same source to the same destination can take different routes in the network  
(C) IP ensures that a packet is discarded if it is unable to reach its destination within a given number of hops  
(D) The packet source cannot set the route of an outgoing packets; the route is determined only by the routing tables in the routers on the way

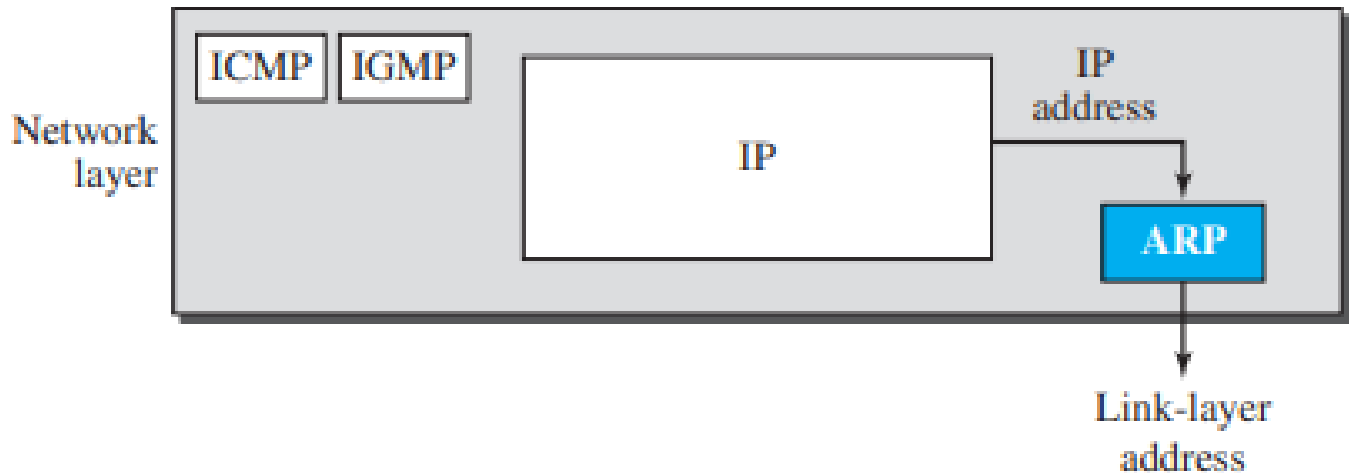
**Answer: (D)**

## Additional protocols

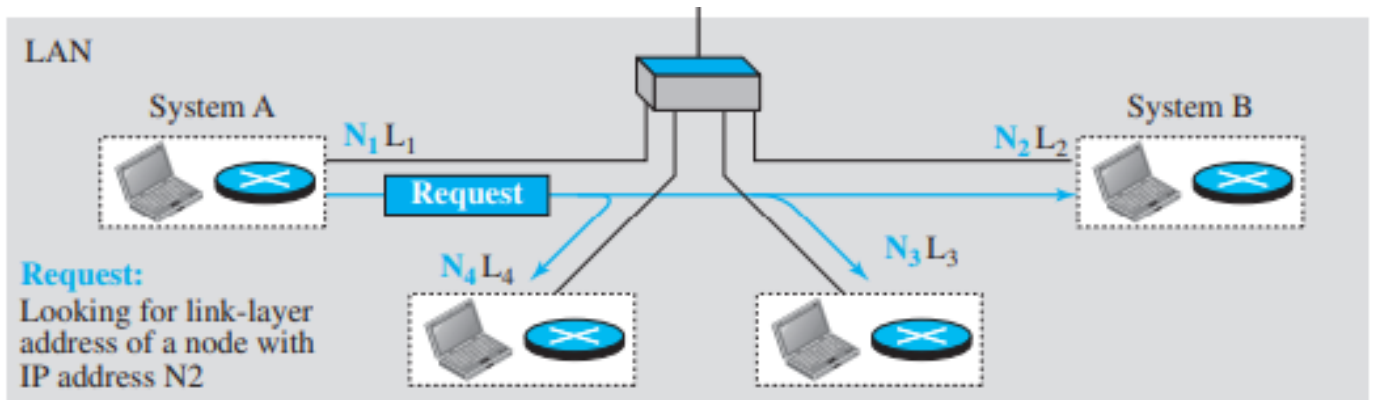
- IP packets, however, need to be encapsulated in a frame, which needs physical addresses (node-to-node). We will see that a protocol called ARP, the Address Resolution Protocol.
- We sometimes need reverse mapping-mapping a physical address to a logical address. For example, when booting a diskless network or leasing an IP address to a host, RARP is used.
- Lack of flow and error control in the Internet Protocol has resulted in another protocol, ICMP, that provides alerts. It reports congestion and some types of errors in the network or destination host
- IP was originally designed for unicast delivery, one source to one destination. As the Internet has evolved, the need for multicast delivery, one source to many destinations, has increased tremendously. IGMP gives IP a multicast capability.

## Address Resolution Protocol (ARP)

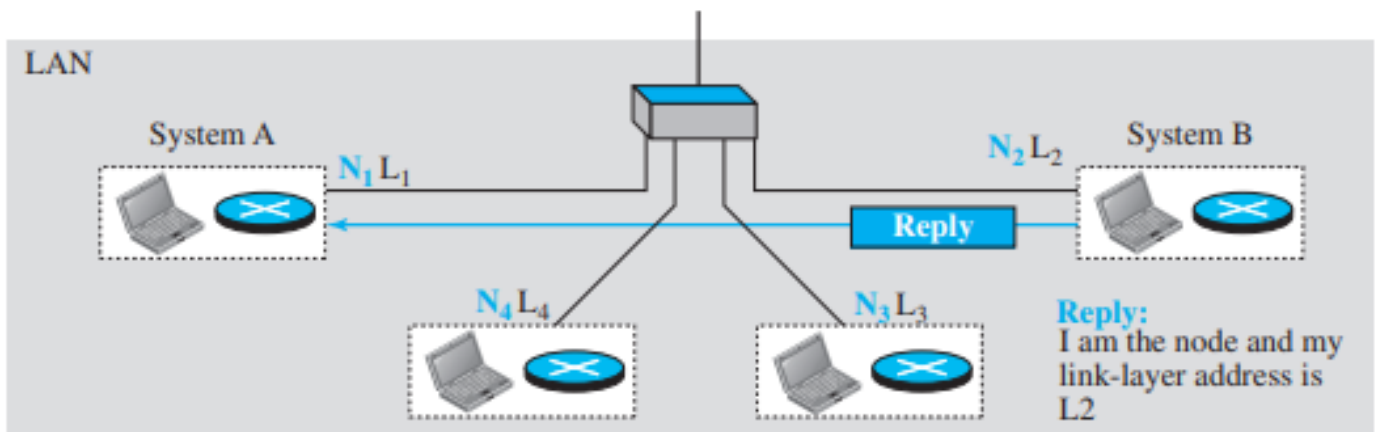
- The IP address of the next node alone is not helpful in moving a frame through a link; we need the link-layer address of the next node so that data link layer can work.
- ARP maps an IP address to a logical-link address.
- ARP accepts an IP address from the IP protocol, maps the address to the corresponding link-layer address, and passes it to the data-link layer.
- The ARP protocol is one of the auxiliary protocols defined in the **network layer**.



- Anytime a host or a router needs to find the link-layer address of another host or router in its network, it sends an ARP request packet.
- The packet includes the **link-layer and IP addresses of the sender and the IP address of the receiver**.
- Because the sender does not know the link-layer address of the receiver, the query is broadcast over the link.
- Every host or router on the network receives and processes the ARP request 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 link-layer addresses.
- **The packet is unicast directly to the node that sent the request packet.**



a. ARP request is broadcast



b. ARP reply is unicast

Sanchini

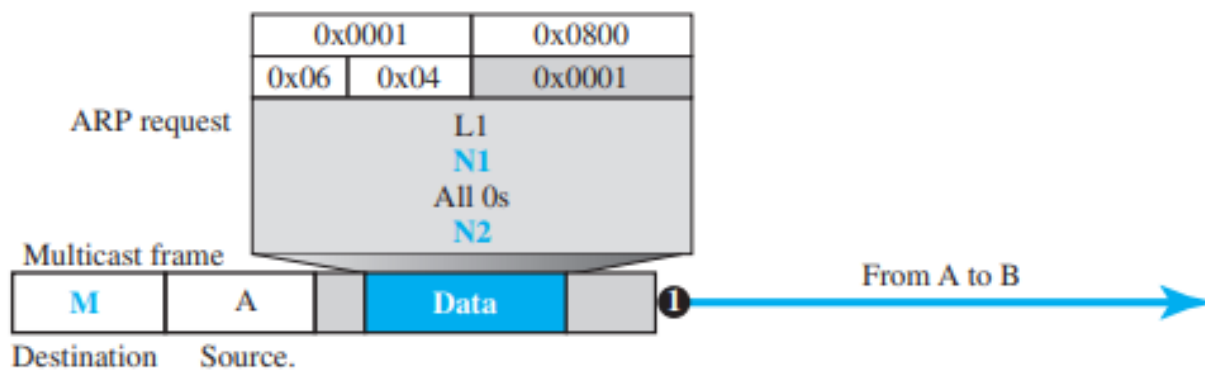
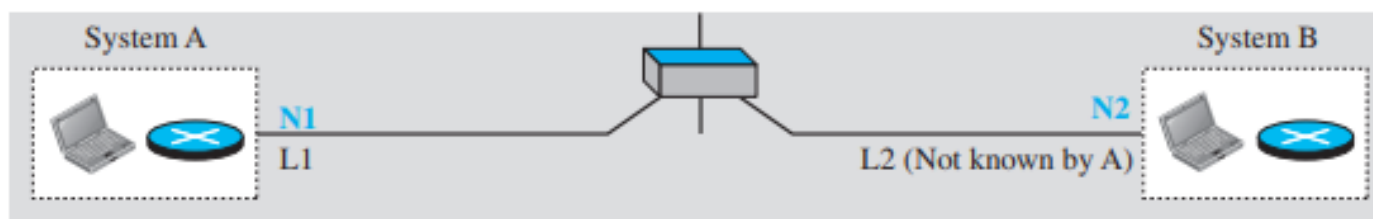
## ARP Packet Format

- **Hardware type:** This is a 16-bit field defining the type of the network on which ARP is running. Each LAN has been assigned an integer based on its type. 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, ARP can be used with any higher-level protocol.
- **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 (for example, IP) 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 (for example, IP) address of the target. For the IPv4 protocol, this field is 4 bytes long

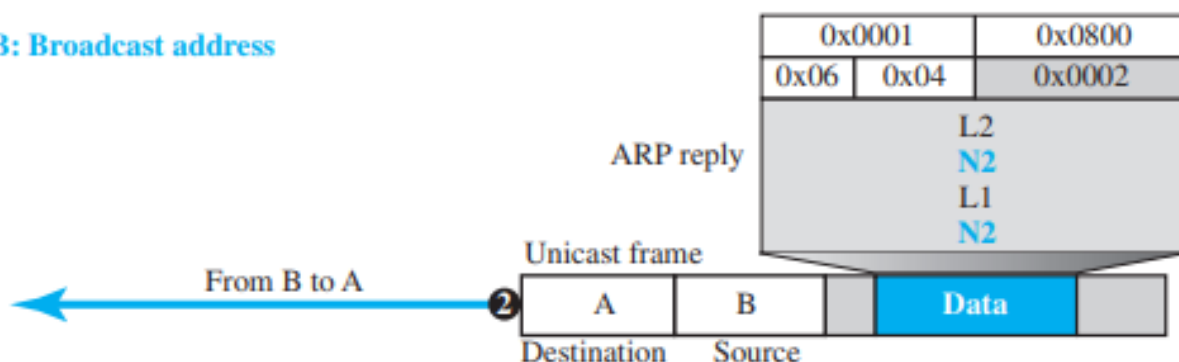
0	8	16	31
Hardware Type		Protocol Type	
Hardware length	Protocol length	Operation <b>Request:1, Reply:2</b>	
Source hardware address			
Source protocol address			
Destination hardware address (Empty in request)			
Destination protocol address			

Sanchit

Example: A host with IP address N1 and MAC address L1 has a packet to send to another host with IP address N2 and physical address L2 (which is unknown to the first host). The two hosts are on the same network. Figure below shows the ARP request and response messages.



**B: Broadcast address**



Sai

## RARP

- Reverse Address Resolution Protocol (RARP) finds the logical address for a machine that knows only its physical address.
- Each host or router is assigned one or more logical (IP) addresses, which are unique and independent of the physical (hardware) address of the machine. To create an IP datagram, a host or a router needs to know its own IP address or addresses.
- 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.

Hardware type		Protocol type
Hardware length	Protocol length	Operation <i>Request 3, Reply 4</i>
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)		

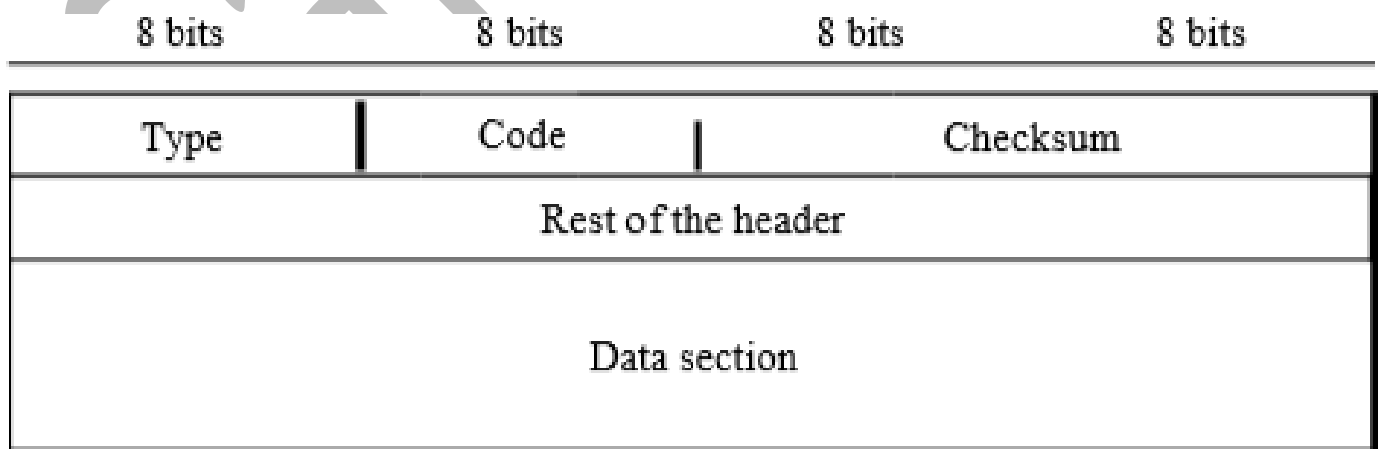
- The machine can get its physical address (by reading its NIC, for example), which is unique locally. It can then use the physical address to get the logical address by using the RARP protocol.
- A RARP request is created and broadcast on the local network. Another machine on the local network that knows all the IP addresses 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. There is a serious problem with RARP: Broadcasting is done at the data link layer.
- The physical broadcast address, allis in the case of Ethernet, does not pass the boundaries of a network. This means that if an administrator has several networks or several subnets, it needs to assign a RARP server for each network or subnet. This is the reason that RARP is almost obsolete.
- BOOTP and DHCP, are replacing RARP.

## ICMP

- IP has two deficiencies: lack of error control and lack of assistance mechanisms. The IP protocol has no error-reporting or error-correcting mechanism. What happens if something goes wrong? 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. Although the general format of the header is different for each message type, the first 4 bytes are common to all message.
- The code field specifies the reason for the particular message type. The last common field is the checksum field. The rest of the header is specific for each message type.
- The data section in error messages carries information for finding the original packet that had the error. In query messages, the data section carries extra information based on the type of the query.

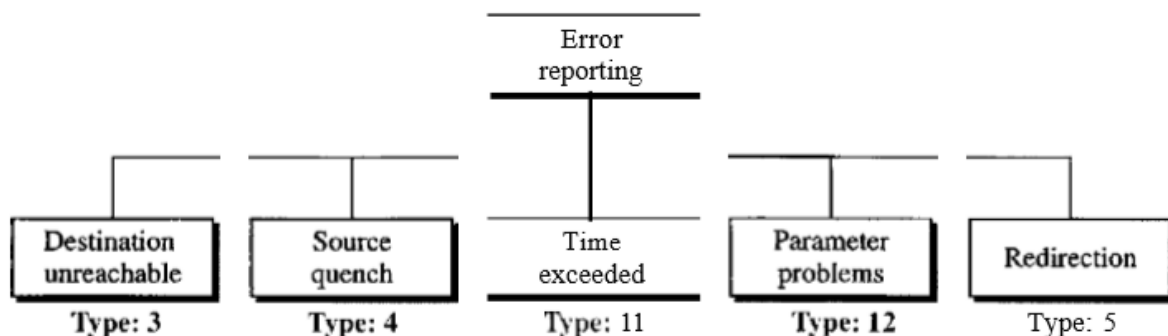


## Types of Messages

- ICMP messages are divided into two broad categories: error-reporting messages and query messages.
- The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.
- The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host. For example, nodes can discover their neighbours. Also, hosts can discover and learn about routers on their network, and routers can help a node redirect its message.

## Error Reporting

- One of the main responsibilities of ICMP is to report errors. Although technology has produced increasingly reliable transmission media, errors still exist and must be handled. IP, is an unreliable protocol.
- This means that error checking and error control are not a concern of IP.
- ICMP was designed, in part, to compensate for this shortcoming. However, ICMP does not correct errors-it simply reports them.
- Error correction is left to the higher-level protocols. Error messages are always sent to the original source because the only information available in the datagram about the route is the source and destination IP addresses.
- ICMP uses the source IP address to send the error message to the source (originator) of the datagram.

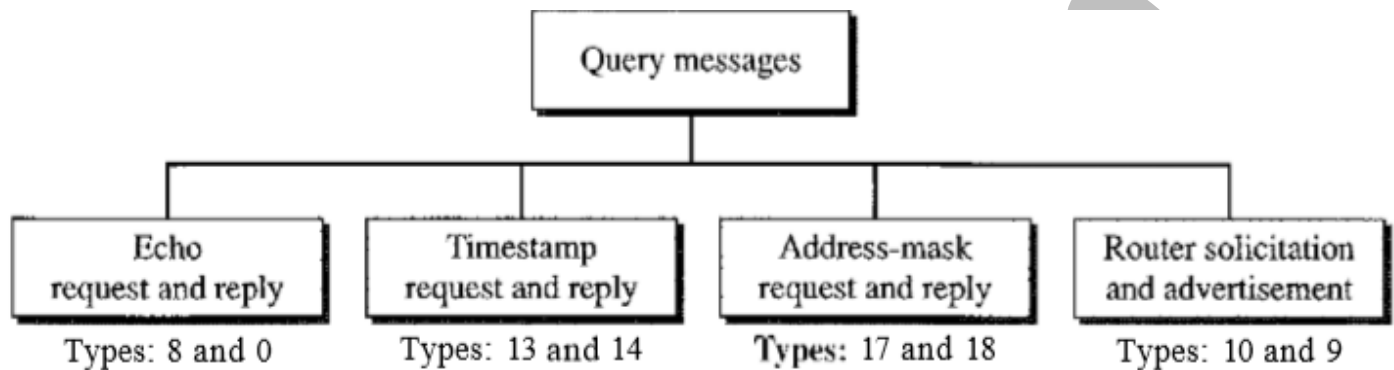


The following are important points about ICMP error messages:

- O** No ICMP error message will be generated in response to a datagram carrying an ICMP error message.
  - D** No ICMP error message will be generated for a fragmented datagram that is not the first fragment.
  - D** No ICMP error message will be generated for a datagram having a multicast address.
  - D** No ICMP error message will be generated for a datagram having a special address such as 127.0.0.0 or 0.0.0.0.
- Note that all error messages contain a data section that includes the IP header of the original datagram plus the first 8 bytes of data in that datagram.
  - The original datagram header is added to give the original source, which receives the error message, information about the datagram itself.

## 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.
- In this type of ICMP message, a node sends a message that is answered in a specific format by the destination node. A query message is encapsulated in an IP packet, which in turn is encapsulated in a data link layer frame. However, in this case, no bytes of the original IP are included in the message



## Echo Request and Reply

- The echo-request and echo-reply messages are designed for diagnostic purposes.
- Network managers and users utilize this pair of messages to identify network problems.
- 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. Today, most systems provide a version of the ping command that can create a series (instead of just one) of echo-request and echo-reply messages, providing statistical information.

## **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. If the host knows the address of the router, it sends the request directly to the router. If it does not know, it broadcasts the message.
- 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 full IP address to get its subnet address.

## **Router Solicitation and Advertisement**

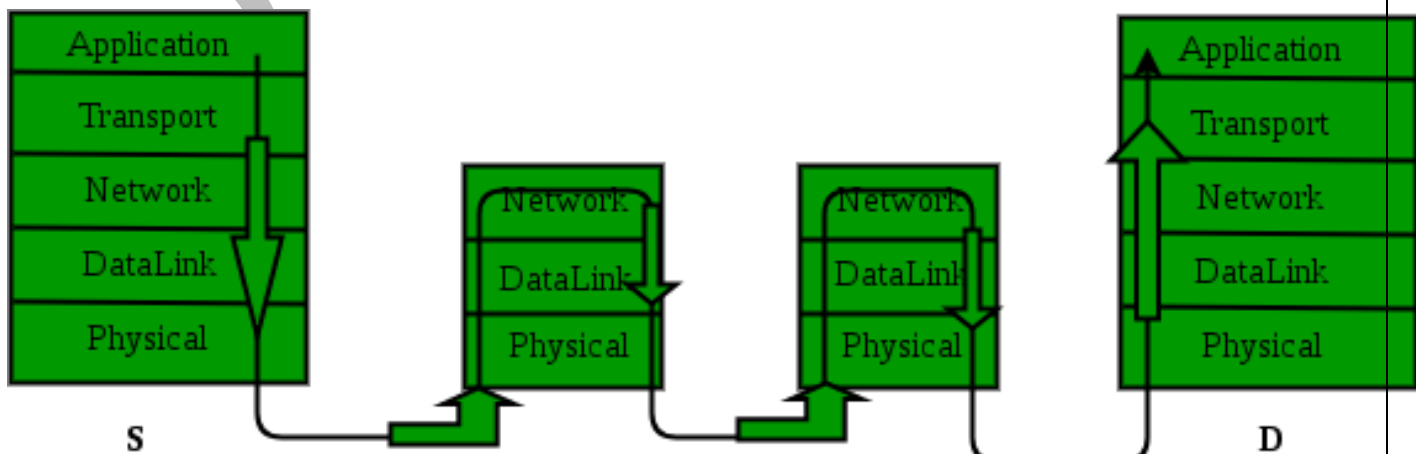
- As we discussed in the redirection message section, a host that wants to send data to a host on another network needs to know the address of routers connected to its own network.
- Also, the host must know if the routers are alive and functioning. The router-solicitation and router-advertisement messages can help in this situation.
- 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.
- A router can also periodically send router-advertisement messages even if no host has solicited. Note that 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.

## IGMP

- The IP protocol can be involved in two types of communication: unicasting and multicasting. Unicasting is the communication between one sender and one receiver. It is a one-to-one communication.
- However, some processes sometimes need to send the same message to a large number of receivers simultaneously. This is called multicasting, which is a one-to-many communication.
- Multicasting has many applications. For example, multiple stockbrokers can simultaneously be informed of changes in a stock price, or travel agents can be informed of a plane cancellation. Some other applications include distance learning and video-on-demand. The Internet Group Management Protocol (IGMP) is one of the necessary, but not sufficient (as we will see Internet Protocol version 4 (IPv4), is responsible for packetizing, forwarding, and delivery of a packet at the network layer.
- The Internet Control Message Protocol version 4 (ICMPv4) helps IPv4 to handle some errors that may occur in the network-layer delivery.
- The Internet Group Management Protocol (IGMP) is used to help IPv4 in multicasting.
- The Address Resolution Protocol (ARP) is used to glue the network and data-link layers in mapping network-layer addresses to link-layer addresses.
- **Example:** Assume that source S and destination D are connected through two intermediate routers labelled R. Determine how many times each packet has to visit the network layer and the data link layer during a transmission from S to D.



- Ans. 4 times Network layer and 6 times data link layer.



## NETWORK-LAYER PERFORMANCE

The performance of a network can be measured in terms of *delay*, *throughput*, and *packet loss*.

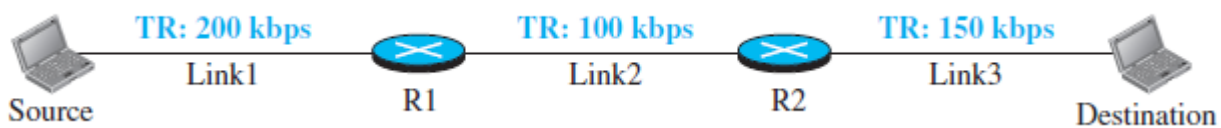
**Delay:** Total delay is similar to the delays in data link layers.

$$\text{Total Delay} = T_t + T_p + T_{que} + T_{proc}$$

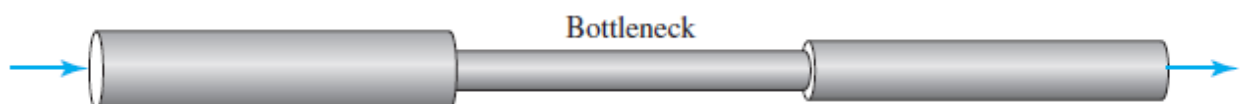
### Throughput

- Throughput at any point in a network is defined as the number of bits passing through the point in a second.
- In a path from source to destination, a packet may pass through several links (networks), each with a different transmission rate.

**Example:** To identify the throughput consider the example, where we have three links, each with a different transmission rate:



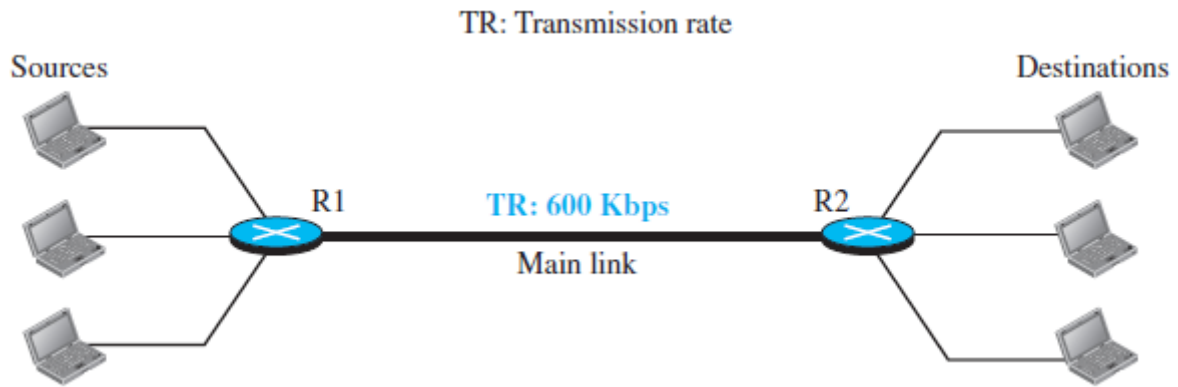
- The data can flow at the rate of 200 kbps in Link1.
- When the data arrives at router R1, it cannot pass at this rate. Data needs to be queued at the router and sent at 100 kbps.
- When data arrives at router R2, it could be sent at the rate of 150 kbps, but there is not enough data to be sent, the average rate of the data flow in Link3 is also going to be 100 kbps, due to **bottlenecking**.



- So, the Throughput is:  $\min \{TR_1, TR_2, TR_3 \dots TR_n\}$ , where TR is transmission rate of different links.

### Effective Throughput in shared links

Consider the figure below:



- In the figure above the transmission rate of the main link in the calculation of the throughput is only 200 kbps because the link is shared between three paths.

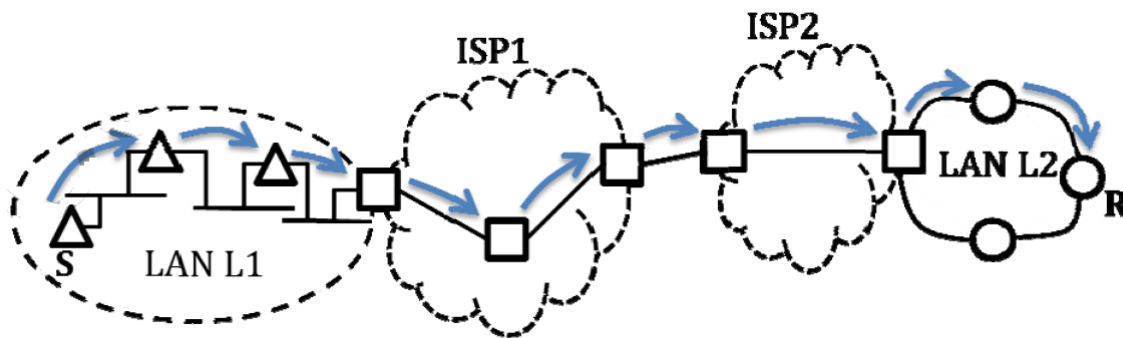
**Q** Consider the store and forward packet switched network given below. Assume that the bandwidth of each link is  $10^6$  bytes / sec. A user on host A sends a file of size  $10^3$  bytes to host B through routers  $R_1$  and  $R_2$  in three different ways. In the first case a single packet containing the complete file is transmitted from A to B. In the second case, the file is split into 10 equal parts, and these packets are transmitted from A to B. In the third case, the file is split into 20 equal parts and these packets are sent from A to B. Each packet contains 100 bytes of header information along with the user data. Consider only transmission time and ignore processing, queuing and propagation delays. Also assume that there are no errors during transmission. Let  $T_1$ ,  $T_2$  and  $T_3$  be the times taken to transmit the file in the first, second and third case respectively. Which one of the following is CORRECT? **(Gate-2014) (2 Marks)**



- (A)  $T_1 < T_2 < T_3$  (B)  $T_1 > T_2 > T_3$   
 (C)  $T_2 = T_3, T_3 < T_1$  (D)  $T_1 = T_3, T_3 > T_2$

**Answer: (D)**

**Q** In the diagram shown below  $L_1$  is an Ethernet LAN and  $L_2$  is a Token-Ring LAN. An IP packet originates from sender S and traverses to R, as shown. The link within each ISP, and across two ISPs, are all point to point optical links. The initial value of TTL is 32. The maximum possible value of TTL field when R receives the datagram is **(Gate-2014) (1 Marks)**



(A) 25

(B) 24

(C) 26

(D) 28

Answer: (C)

**Q** A link of capacity 100 Mbps is carrying traffic from a number of sources. Each source generates an on-off traffic stream; when the source is on, the rate of traffic is 10 Mbps, and when the source is off, the rate of traffic is zero. The duty cycle, which is the ratio of on-time to off-time, is 1 : 2. When there is no buffer at the link, the minimum number of sources that can be multiplexed on the link so that link capacity is not wasted and no data loss occurs is  $S_1$ . Assuming that all sources are synchronized and that the link is provided with a large buffer, the maximum number of sources that can be multiplexed so that no data loss occurs is  $S_2$ . The values of  $S_1$  and  $S_2$  are, respectively, **(Gate-2006) (2 Marks)**

(A) 10 and 30

(B) 12 and 25

(C) 5 and 33

(D) 15 and 22

Answer: (A)

**Q** In a communication network, a packet of length  $L$  bits takes link  $L_1$  with a probability of  $p_1$  or link  $L_2$  with a probability of  $p_2$ . Link  $L_1$  and  $L_2$  have bit error probability of  $b_1$  and  $b_2$  respectively. The probability that the packet will be received without error via either  $L_1$  or  $L_2$  is **(Gate-2005) (2 Marks)**

(A)  $(1 - b_1)^L p_1 + (1 - b_2)^L p_2$

(B)  $[1 - (b_1 + b_2)^L] p_1 p_2$

(C)  $(1 - b_1)^L (1 - b_2)^L p_1 p_2$

(D)  $1 - (b_1^L p_1 + b_2^L p_2)$

Answer: (A)

**Q** In a packet switching network, packets are routed from source to destination along a single path having two intermediate nodes. If the message size is 24 bytes and each packet contain a header of 3 bytes, then the optimum packet size is: **(Gate-2005) (2 Marks)**

(a) 4

(b) 6

(c) 7

(d) 9

Answer (d)

**Q** The address resolution protocol (ARP) is used for **(Gate-2005) (1 Marks)**

(A) Finding the IP address from the DNS

(B) Finding the IP address of the default gateway

- (C) Finding the IP address that corresponds to a MAC address
- (D) Finding the MAC address that corresponds to an IP address

**Answer: (D)**

**Q** Consider three IP networks A, B and C. Host HA in network A sends messages each containing 180 bytes of application data to a host HC in network C. The TCP layer prefixes a 20-byte header to the message. This passes through an intermediate network B. The maximum packet size, including 20 byte IP header, in each network is

A : 1000 bytes

B : 100 bytes

C : 1000 bytes

The network A and B are connected through a 1 Mbps link, while B and C are connected by a 512 Kbps link (bps = bits per second).



Assuming that the packets are correctly delivered, how many bytes, including headers, are delivered to the IP layer at the destination for one application message, in the best case ?

Consider only data packets. **(Gate-2004) (2 Marks)**

(A) 200

(B) 220

(C) 240

(D) 260

**Answer: (D)**

**Q** What is the rate at which application data is transferred to host HC? Ignore errors, acknowledgements, and other overheads. **(Gate-2004) (2 Marks)**

(A) 325.5 Kbps

(B) 354.5 Kbps

(C) 409.6 Kbps

(D) 512.0 Kbps

**Answer: (B)**

**Q** Traceroute reports a possible route that is taken by packets moving from some host A to some other host B. Which of the following options represents the technique used by traceroute to identify these hosts **(GATE-2005) (1 Marks)**

(A) By progressively querying routers about the next router on the path to B using ICMP packets, starting with the first router

(B) By requiring each router to append the address to the ICMP packet as it is forwarded to B. The list of all routers en-route to B is returned by B in an ICMP reply packet

(C) By ensuring that an ICMP reply packet is returned to A by each router en-route to B, in the ascending order of their hop distance from A

(D) By locally computing the shortest path from A to B

**Answer: (A)**

**Q** Which of the following is NOT true with respect to a transparent bridge and a router?  
**(Gate-2004) (1 Marks)**

(A) Both bridge and router selectively forward data packets

(B) A bridge uses IP addresses while a router uses MAC addresses

(C) A bridge builds up its routing table by inspecting incoming packets

(D) A router can connect between a LAN and a WAN

**Answer: (B)**

**Q** Every host in an IPv4 network has a 1-second resolution real-time clock with battery backup. Each host needs to generate up to 1000 unique identifiers per second. Assume that each host has a globally unique IPv4 address. Design a 50-bit globally unique ID for this purpose. After what period (in seconds) will the identifiers generated by a host wrap around? **(Gate-2014) (1 Marks)**

**Q** For a host machine that uses the token bucket algorithm for congestion control, the token bucket has a capacity of 1 megabyte and the maximum output rate is 20 megabytes per second. Tokens arrive at a rate to sustain output at a rate of 10 megabytes per second. The token bucket is currently full and the machine needs to send 12 megabytes of data. The minimum time required to transmit the data is \_\_\_\_\_ seconds. **(Gate-2017) (2 Marks)**

**Q** A computer on a 10Mbps network is regulated by a token bucket. The token bucket is filled at a rate of 2Mbps. It is initially filled to capacity with 16Megabits. What is the maximum duration for which the computer can transmit at the full 10Mbps? **(GATE-2008) (2 Marks)**

(A) 1.6 seconds

(B) 2 seconds

(C) 5 seconds

(D) 8 seconds

**Answer: (B)**

## IPV4 ADDRESSES

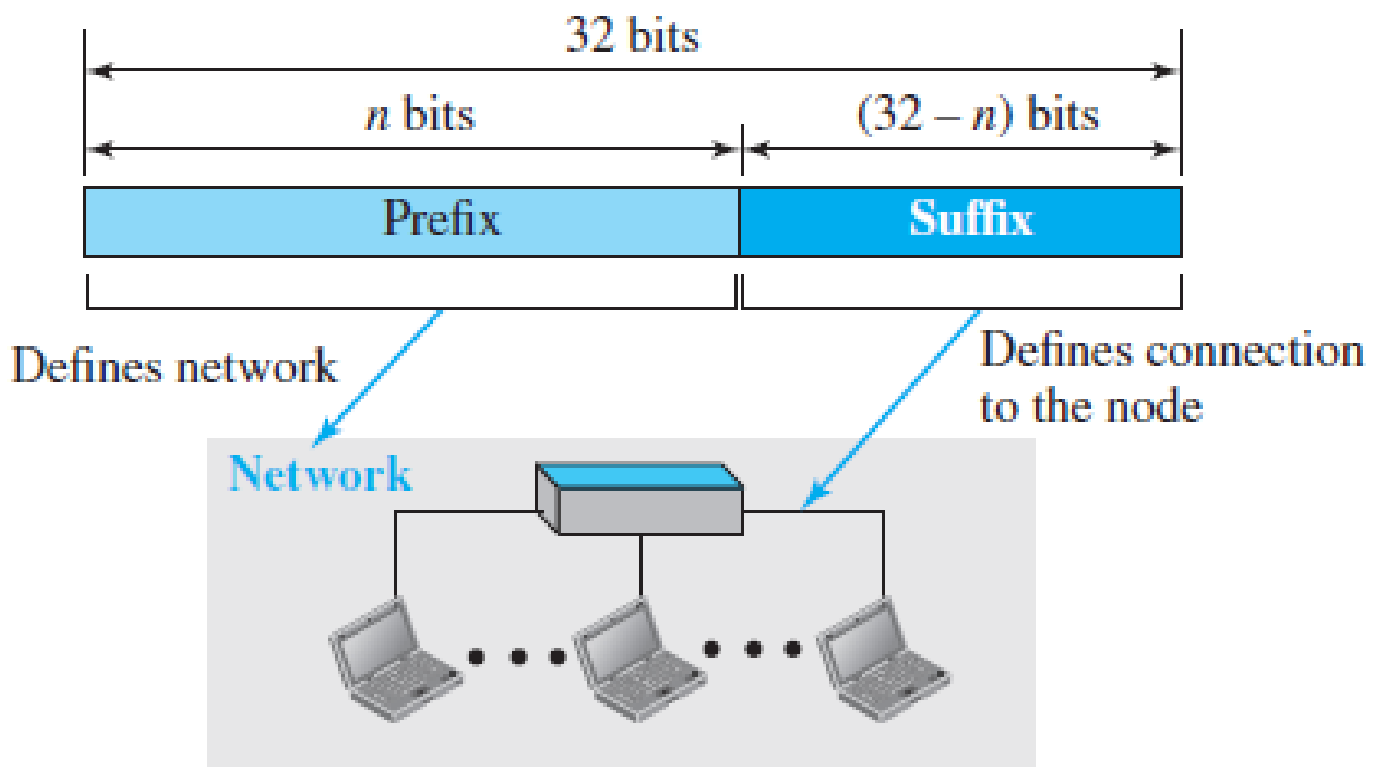
- The Internet Protocol addresses are 32 bits in length; this gives us a maximum of  $2^{32}$  addresses. These addresses are referred to as IPv4 (IP version 4) addresses or simply IP addresses if there is no confusion.
- This means that, theoretically, if there were no restrictions, more than 4 billion (4,294,967,296) devices could be connected to the Internet. The actual number is much less because of the restrictions imposed on the addresses.
- The need for more addresses, in addition to other concerns about the IP layer, motivated a new design of the IP layer called the new generation of IP or IPv6 (IP version 6). In this version, the Internet uses 128-bit addresses that give much greater flexibility in address allocation ( $3.4 \times 10^{38}$ ). These addresses are referred to as IPv6 (IP version 6) addresses.
- An IP address is uniquely and universally defining the connection of a host or a router to the Internet.
- They are unique in the sense that each 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.
- The IPv4 addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet.
- The IP address is the address of the connection, not the host or the router, because if the device is moved to another network, the IP address may be changed.

## Notations

- There are two prevalent 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. So, it is common to hear an IPv4 address referred to as a 32-bit address or a 4-byte address. 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

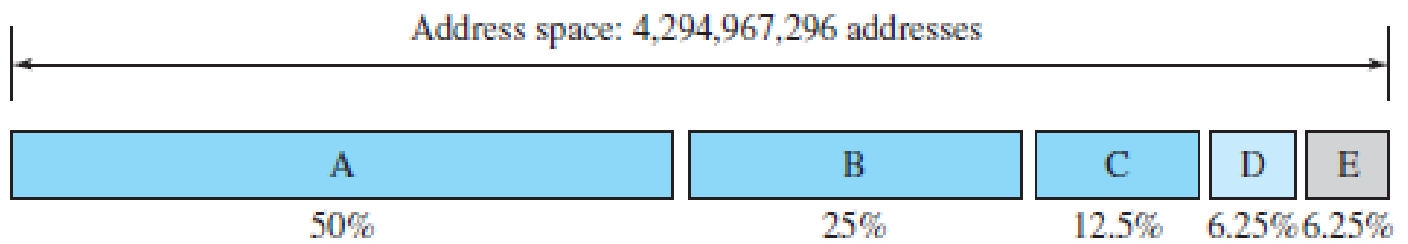
## Classful Addressing

- IPv4 addressing, at its inception, used the concept of classes. This architecture is called classful addressing.
- A 32-bit IPv4 address is hierarchical and divided only into two parts:
- The first part of the address, called the *prefix*, defines the network (NetworkID).
- The second part of the address, called the *suffix*, defines the node (connection of a device to the Internet (HostID)).
- The prefix length is  $n$  bits and the suffix length is  $(32 - n)$  bits.



- A prefix can be fixed length or variable length.
- IPv4 was first designed as a fixed-length prefix and is referred to as classful addressing,
- In classful addressing, the address space is divided into five classes: A, B, C, D, and E. Each class occupies some part of the address space.
- Now-a-days it has become obsolete because of many problems, now the new scheme referred to as classless addressing, uses a variable-length network prefix.

- To accommodate both small and large networks, three fixed-length prefixes were designed ( $n = 8, n = 16,$  and  $n = 24$ ).
- The whole address space was divided into five classes (class A, B, C, D, and E).
- This scheme is referred to as classful addressing.



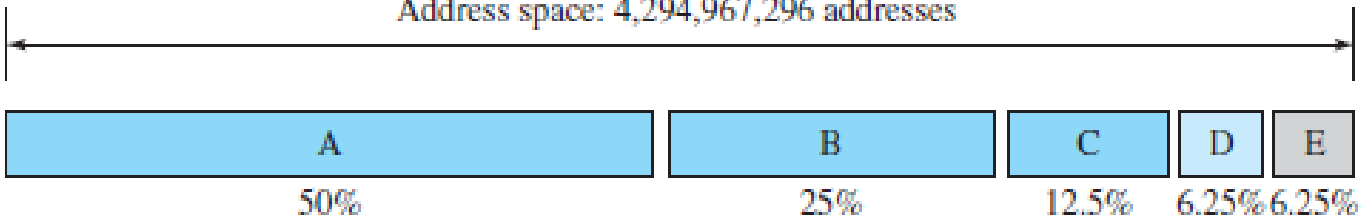
Class A	0	Prefix	Suffix
Class B	10	Prefix	Suffix
Class C	110	Prefix	Suffix
Class D	1110	Multicast addresses	
Class E	1111	Reserved for future use	

Class	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

# Class A

- In Class A NetID = 8 bits and HostID = 24.
- How to identify class A address
  - First bit is reserved to 0 in binary notation
  - Range of 1<sup>st</sup> octet is [0, 127] in dotted decimal notation
- Total number of connections in class A is  $2^{31}$  (2,14,74,83,648)
- There are  $2^7 - 2 = 126$  networks in Class A network.
  - In Class A, total network available are 2 less, because:
  - IP Address 0.0.0.0 is reserved for broadcasting requirements
  - IP Address 127.0.0.1 is reserved for loopback address used for software testing.
  - The range of 1<sup>st</sup> octet is [0, 127] but since two addresses are reserved it is: [1, 126].
- There are  $2^{24} - 2$  (1,67,77,214) HostID in Class A.
  - In all the classes, total number of hosts that can be configured are 2 less because:
  - This is to account for the two reserved IP addresses in which all the bits for host ID are either zero or one.
  - When all Host ID bits are 0, it represents the Network ID for the network.
  - When all Host ID bits are 1, it represents the Broadcast Address.
- Class A is used by organizations requiring very large size networks like Indian Railways.

Address space: 4,294,967,296 addresses



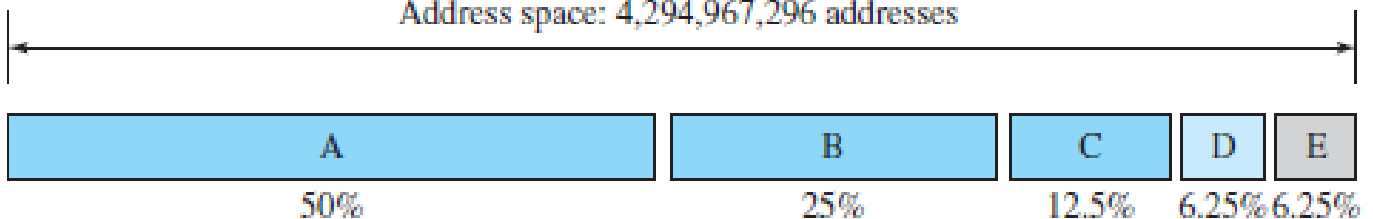
Class A	0	Prefix	Suffix
Class B	10	Prefix	Suffix
Class C	110	Prefix	Suffix
Class D	1110 Multicast addresses		
Class E	1111 Reserved for future use		

Class	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

## Class B

- In Class B NetID = 16 bits and HostID = 16.
- How to identify class B address
  - First two bits are reserved to 10 in binary notation
  - Range of 1<sup>st</sup> octet is [128, 191] in dotted decimal notation
- Total number of connections in class B is  $2^{30}$  (1,07,37,41,824)
- Total number of networks available in class B is  $2^{14}$  (16,384)
- Total number of hosts that can be configured in every network in class B is  $2^{16} - 2$  (65,534)
- Class B is used by organizations requiring medium size networks

Address space: 4,294,967,296 addresses

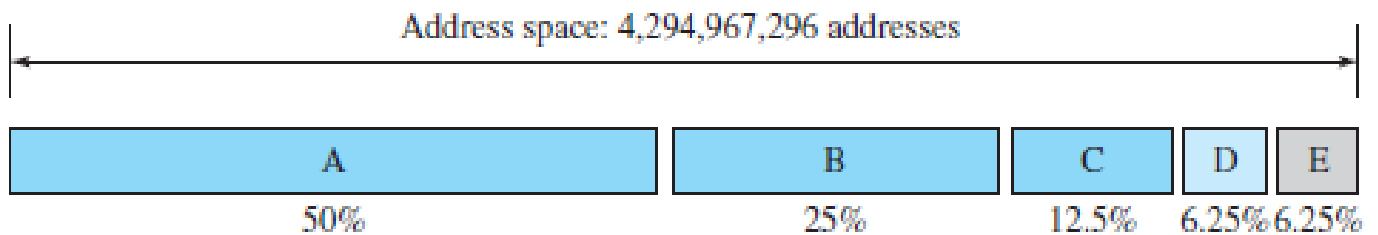


Class A	0	Prefix	Suffix
Class B	10	Prefix	Suffix
Class C	110	Prefix	Suffix
Class D	1110	Multicast addresses	
Class E	1111	Reserved for future use	

Class	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

## Class C

- In Class C NetID = 24 bits and HostID = 8.
- How to identify class C address
  - First three bits are reserved to 110 in binary notation
  - Range of 1<sup>st</sup> octet is [192, 223] in dotted decimal notation
- Total number of connections in class C is  $2^{29}$  (53,68,70,912)
- Total number of networks available in class C is  $2^{21}$  (20,97,152)
- Total number of hosts that can be configured in every network in class C is  $2^8 - 2$  (254)
- Class C is used by organizations requiring small to medium size networks.



	8 bits	8 bits	8 bits	8 bits	
	0 Prefix		Suffix		
Class A	10 Prefix		Suffix		
Class B	110 Prefix			Suffix	
Class C	1110 Multicast addresses				
Class D	1111 Reserved for future use				
Class E	A	Prefixes	First byte		
	A	$n = 8$ bits	0 to 127		
	B	$n = 16$ bits	128 to 191		
	C	$n = 24$ bits	192 to 223		
	D	Not applicable	224 to 239		
	E	Not applicable	240 to 255		

**Q** In the IPv4 addressing format, the number of networks allowed under Class C addresses is (Gate-2012) (1 Marks)

(A)  $2^{14}$

(B)  $2^7$

(C)  $2^{21}$

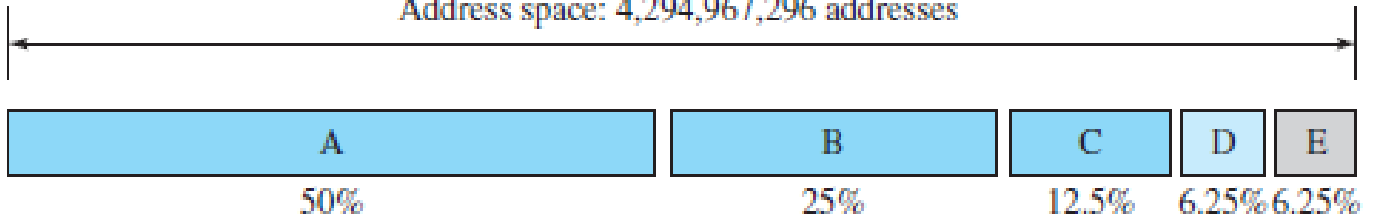
(D)  $2^{24}$

**Answer: (C)**

## Class D

- Class D is not divided into Network ID and Host ID.
- How to identify class D address
  - First four bits are reserved to 1110 in binary notation
  - Range of 1<sup>st</sup> octet is [224, 239] in dotted decimal notation
- Total number of IP Addresses available in class D =  $2^{28}$  (26,84,35,456)
- Class D is reserved for multicasting, in multicasting, there is no need to extract host address from the IP Address, this is because data is not destined for a particular host.

Address space: 4,294,967,296 addresses

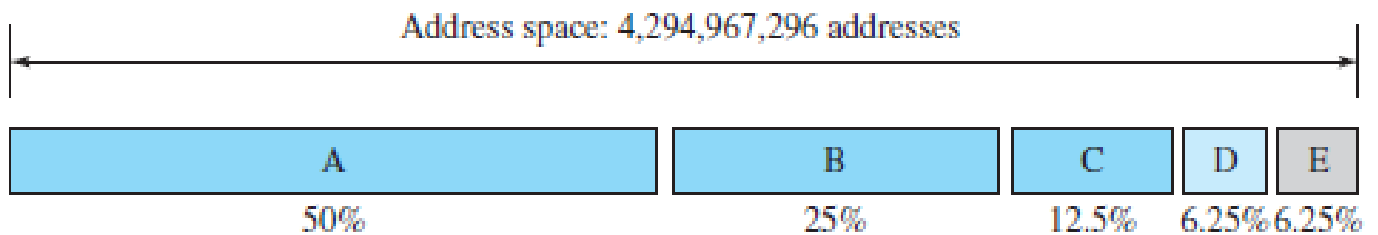


Class A	0	Prefix	Suffix
Class B	10	Prefix	Suffix
Class C	110	Prefix	Suffix
Class D	1110	Multicast addresses	
Class E	1111	Reserved for future use	

Class	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

# Class E

- Class E is not divided into Network ID and Host ID.
  - How to identify class E address
    - First four bits are reserved to 1111 in binary notation
    - Range of 1<sup>st</sup> octet is [240, 255] in dotted decimal notation
- If the 32-bit binary address starts with bits 1111, then IP Address belongs to class E.
- Range of 1st octet = [240, 255]
- Total number of IP Addresses available in class E =  $2^{28}$  (26,84,35,456)
- Class E is reserved for future or experimental purposes.



Class A	0	Prefix	Suffix
Class B	10	Prefix	Suffix
Class C	110	Prefix	Suffix
Class D	1110	Multicast addresses	
Class E	1111	Reserved for future use	

Class	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

## Points to note

- All the hosts in a single network always have the same network ID but different Host ID.
- Two hosts in two different networks can have the same host ID.
- Only those devices which have the network layer will have IP Address, switches, hubs and repeaters does not have any IP Address.

**Q** If the IP address of the system 63.12.11.13 to which class it belongs to?

**Q** if the IP address of a system is 36.11.119.14 calculate the net id, directed broadcast add, first and last add?

**Q** if the IP address of a system is 141.119.89.63 calculate the net id, directed broadcast add, first and last add?

**Q** if the IP address of a system is 23.19.18.17 calculate the net id, directed broadcast add, first and last add?

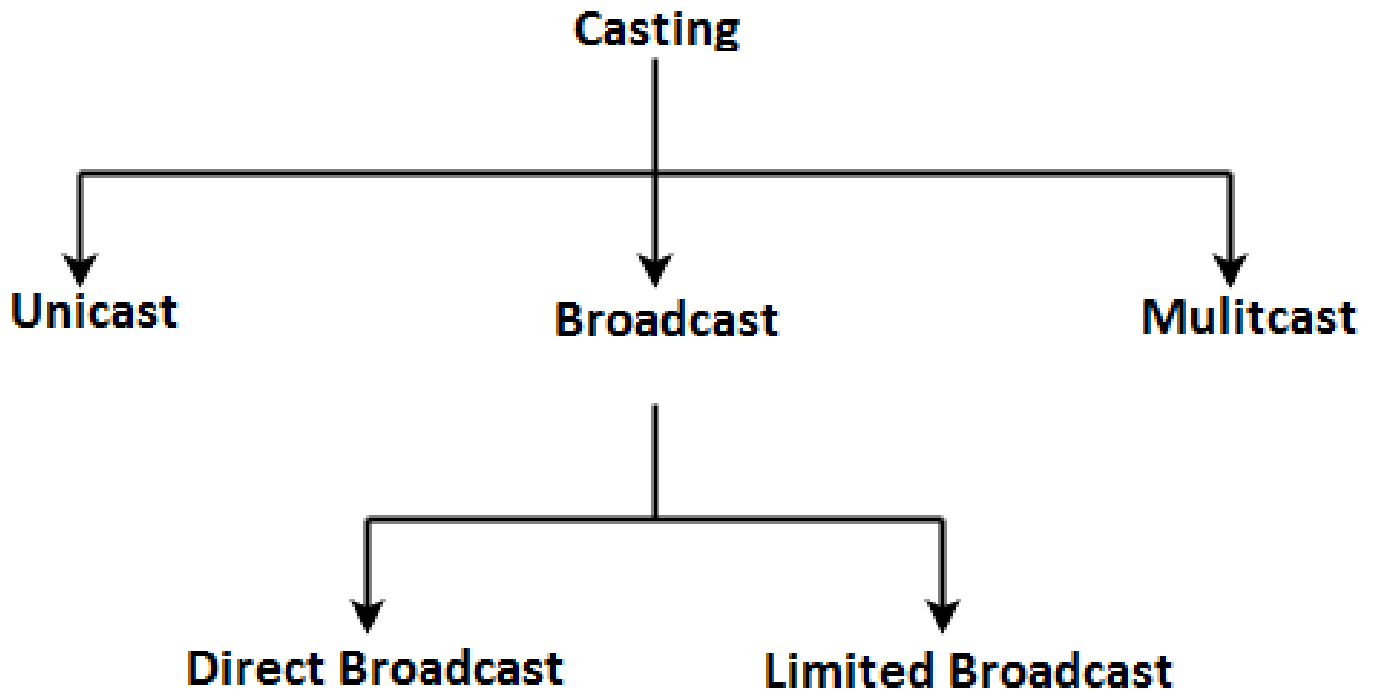
**Q** if the IP address of a system is 131.86.17.18 calculate the net id, directed broadcast add, first and last add?

**Q** if the IP address of a system is 97.15.21.16 calculate the net id, directed broadcast add, first and last add?

# Casting in Networks

## Types of Casting

- Casting in a network is basically of three type: Unicast, Multicast and Broadcast.

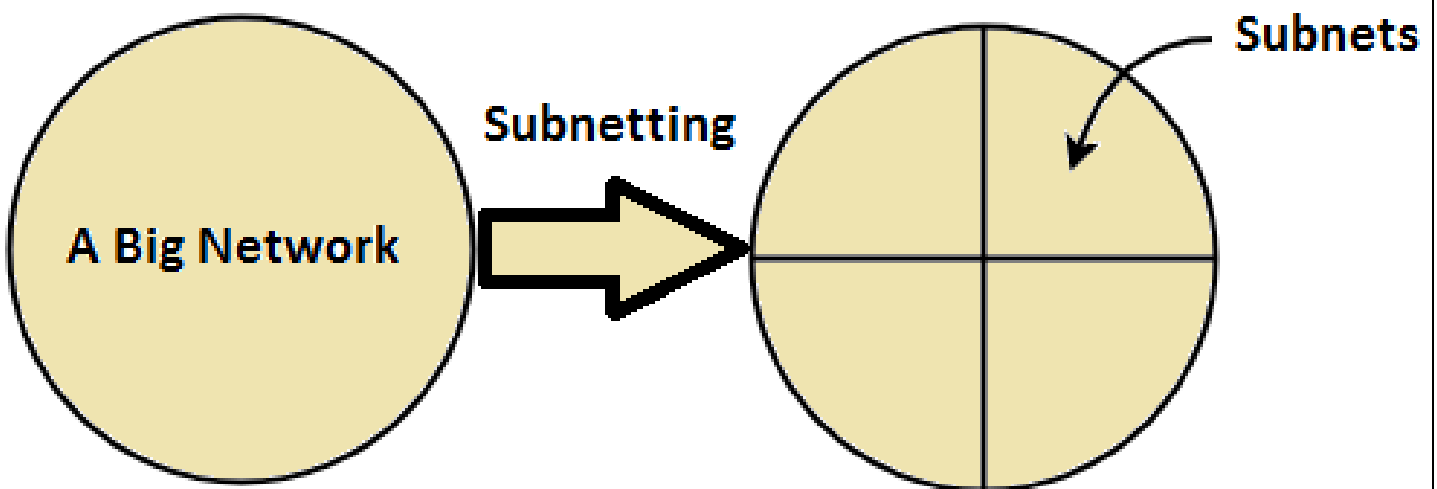


- **Unicast:** Transmitting data from one source host to one destination host is called as **unicast**. It is a one to one transmission.

- **Broadcast:** Transmitting data from one source host to all other hosts residing in a network either same or other network is called as **broadcast**. It is a one to all transmission.
  - **Limited Broadcast:** Transmitting data from one source host to all other hosts residing in the same network is called as limited broadcast. Limited Broadcast Address for any network is
    - All 32 bits set to 1 = 11111111.11111111.11111111.11111111 = 255.255.255.255
  - **Direct Broadcast:** Transmitting data from one source host to all other hosts residing in some other network is called as direct broadcast.
    - Direct Broadcast Address for any network is the IP Address where, Network ID is the IP Address of the network where all the destination hosts are present and Host ID bits are all set to 1.
- **Multicast:** Transmitting data from one source host to a particular group of hosts having interest in receiving the data is called as multicast. It is a one to many transmissions.

## Subnetting

- Maintenance of a very big network like class A and class B is very difficult for network administrator.
- Having all the computer from different departments in a company on the same networks is less secure from company prospective.
- So, if an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks (called subnets) or, in rare cases, share part of the addresses with neighbours.
- **Conclusion:** An organization (or an ISP) that is granted a range of addresses may divide the range into several subranges and assign each subrange to a subnetwork (or subnet). A subnetwork can be divided into several sub-subnetworks. A sub-subnetwork can be divided into several sub-sub-subnetworks, and so on.



### Advantages

- It improves the security.
- The maintenance and administration of subnets is easy.

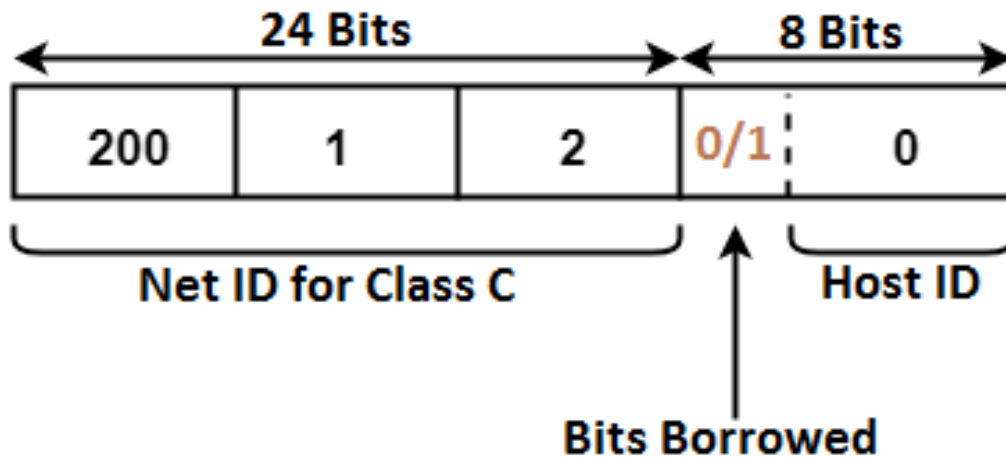
### Disadvantages

- Identification of a station is difficult
- Not possible to directed broadcast from outside network.
- 2 IP addresses are waisted in every subnet

**Example 1.** Consider the network having IP Address 200.1.2.0. Divide this network into two subnets.

Now, since the given class of IP address belong to class C. To design two subnets and to represent their subnet IDs, we require 1 bit.

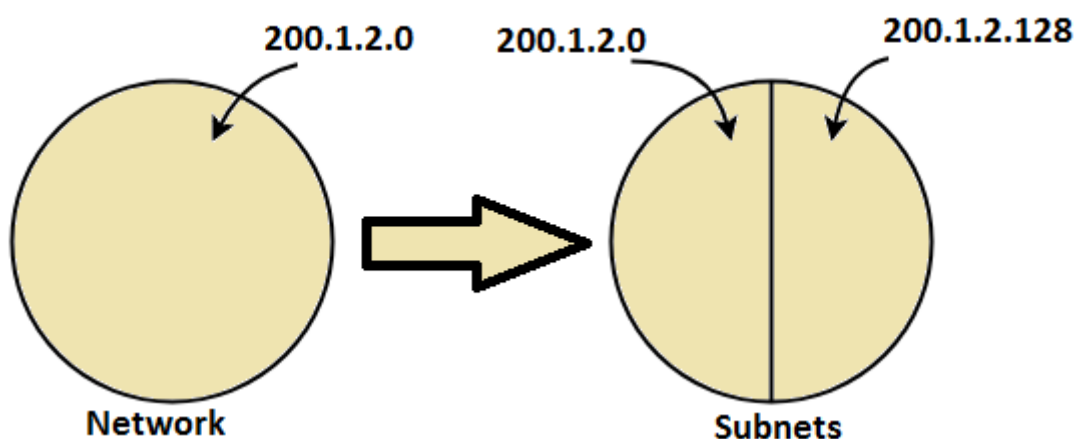
- We borrow one bit from the Host ID part.
- Now, Host ID part remains with only 7 bits.



- If borrowed bit is 0, it represents the first subnet. If borrowed bit is 1, it represents the second subnet.

Network Address of the two subnets will be represented as:

- 200.1.2.00000000 = 200.1.2.0 (When borrowed bit is 0)
- 200.1.2.10000000 = 200.1.2.128 (When borrowed bit is 1)



### **1st Subnet**

- IP Address of the subnet / Subnet id = 200.1.2.0
- Total number of IP Addresses =  $2^7 = 128$
- Total number of hosts that can be configured =  $128 - 2 = 126$
- Range of IP Addresses = [200.1.2.0, 200.1.2.127]
- Direct Broadcast Address = 200.1.2.01111111 = 200.1.2.127
- Limited Broadcast Address = 255.255.255.255

### **2nd Subnet**

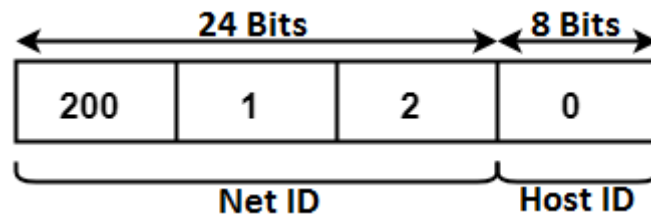
- IP Address of the subnet / Subnet id = 200.1.2.128
- Total number of IP Addresses =  $2^7 = 128$
- Total number of hosts that can be configured =  $128 - 2 = 126$
- Range of IP Addresses = [200.1.2.128, 200.1.2.255]
- Direct Broadcast Address = 200.1.2.11111111 = 200.1.2.255
- Limited Broadcast Address = 255.255.255.255

### **Subnet ID**

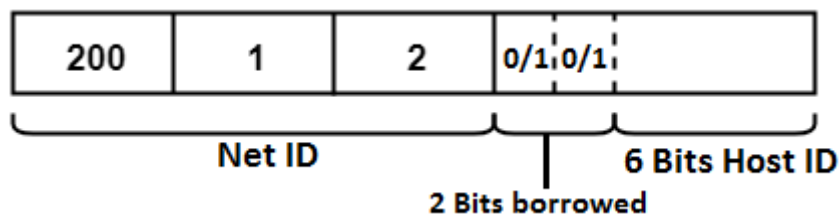
- Each subnet has its unique network address known as its Subnet ID.
- The subnet ID is created by borrowing some bits from the Host ID part of the IP Address.
- The number of bits borrowed depends on the number of subnets created.
- Subnetting increases the number of 1's in the mask

**Example 2:** Consider we have a big single network having IP Address 200.1.2.0. We want to do subnetting and divide this network into 4 subnets.

The IP address is clearly in Class C Network.



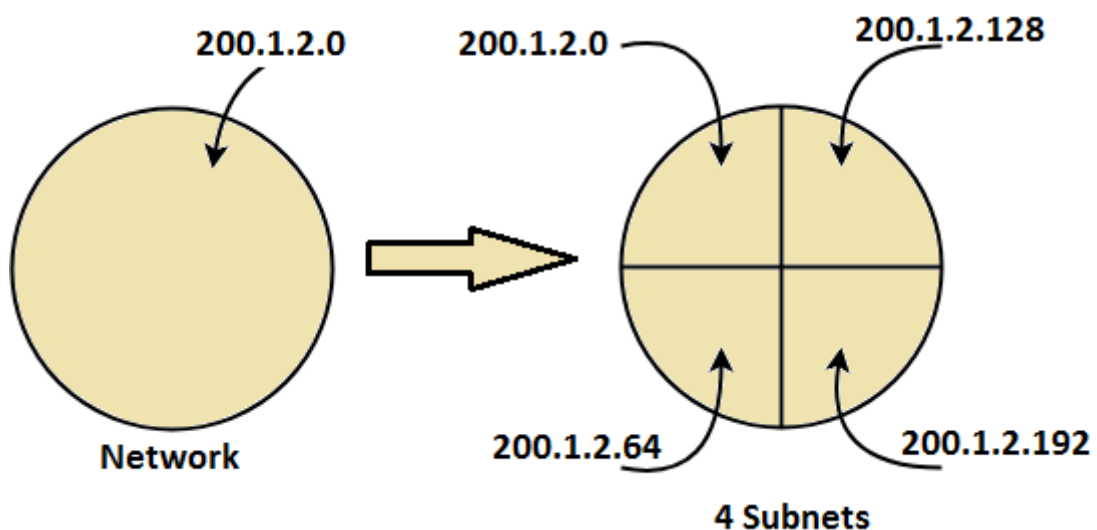
Now, to divide a network into 4 subnets we will need to borrow 2 bits from the host. The host will now remain with 6 bits.



- If borrowed bits is 00, it will represent the 1st subnet.
- If borrowed bits is 01, it will represent the 2nd subnet.
- If borrowed bits is 10, it will represent the 3rd subnet.
- If borrowed bits is 11, it will represent the 4th subnet.

**IP Address of the four subnets respectively are:**

- 200.1.2.**00**000000 = 200.1.2.0
- 200.1.2.**01**000000 = 200.1.2.64
- 200.1.2.**10**000000 = 200.1.2.128
- 200.1.2.**11**000000 = 200.1.2.192



### **1st Subnet**

- IP Address of the subnet / Subnet id = 200.1.2.0
- Total number of IP Addresses =  $2^6 = 64$
- Total number of hosts that can be configured =  $64 - 2 = 62$
- Range of IP Addresses = [200.1.2.0, 200.1.2.63]
- Direct Broadcast Address = 200.1.2.**00**111111 = 200.1.2.63
- Limited Broadcast Address = 255.255.255.255

### **2nd Subnet**

- IP Address of the subnet / Subnet id = 200.1.2.64
- Total number of IP Addresses =  $2^6 = 64$
- Total number of hosts that can be configured =  $64 - 2 = 62$
- Range of IP Addresses = [200.1.2.64, 200.1.2.127]
- Direct Broadcast Address = 200.1.2.**01**111111 = 200.1.2.127
- Limited Broadcast Address = 255.255.255.255

### **3rd Subnet**

- IP Address of the subnet / Subnet id = 200.1.2.128
- Total number of IP Addresses =  $2^6 = 64$
- Total number of hosts that can be configured =  $64 - 2 = 62$
- Range of IP Addresses = [200.1.2.128, 200.1.2.191]
- Direct Broadcast Address = 200.1.2.**10**111111 = 200.1.2.191
- Limited Broadcast Address = 255.255.255.255

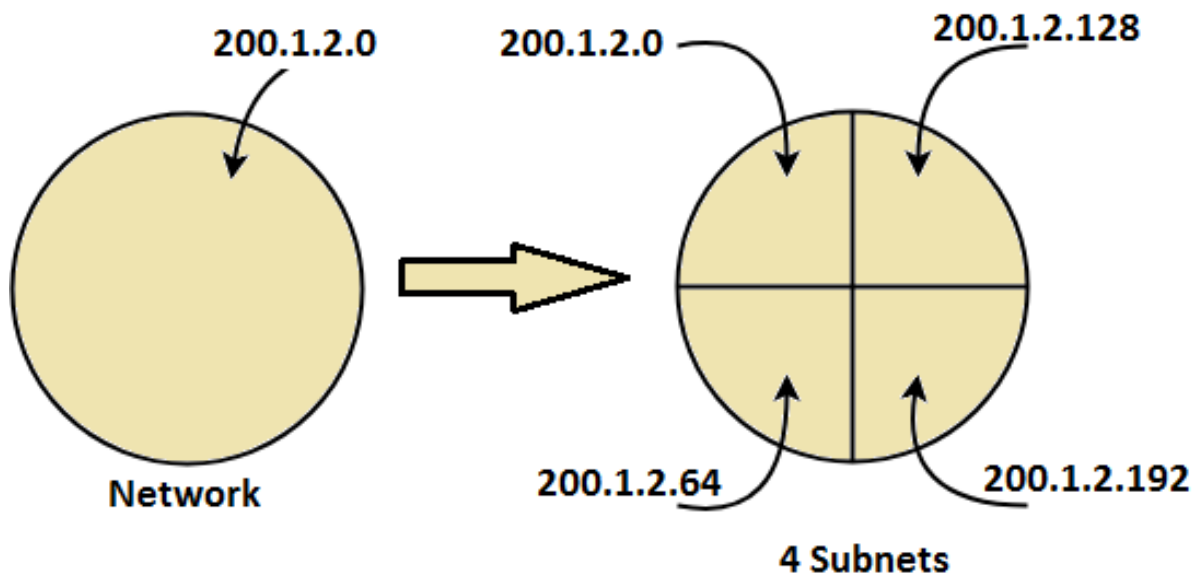
### **4th Subnet**

- IP Address of the subnet / Subnet id = 200.1.2.192
- Total number of IP Addresses =  $2^6 = 64$
- Total number of hosts that can be configured =  $64 - 2 = 62$
- Range of IP Addresses = [200.1.2.192, 200.1.2.255]
- Direct Broadcast Address = 200.1.2.**11**111111 = 200.1.2.255
- Limited Broadcast Address = 255.255.255.255

## Subnet Masks

- In case of subnetting the problem is how to identify to which subnet the incoming packet from outside the network must be delivered. To solve this problem, we use the idea of subnet mask.
- Subnet mask is a 32-bit number which is a sequence of 1's followed by a sequence of 0's where:
  - 1's represents the Network ID part along with the subnet ID.
  - 0's represents the host ID part.
- Default mask for different classes of IP Address are:
  - Default subnet mask of Class A = 255.0.0.0
  - Default subnet mask for Class B = 255.255.0.0
  - Default subnet mask for Class C = 255.255.255.0
- Networks of same size always have the same subnet mask.

For, example: The subnet masks of the following 4 subnets will be:



As it is fixed length subnets thus for each subnet the subnet mask will be:

- 11111111.11111111.11111111.11000000 (i.e. we set the first 26 bits to 1's (Network ID + SubNetID) and remaining bits to 0.)
- Subnet mask = 255.255.255.192

**Q** if the IP address of a system is 141.121.119.168, with subnet mask 255.255.252.0, calculate the net id, directed broadcast add, first and last add?

**Q** if the IP address of a system is 203.112.111.117, with subnet mask 255.255.255.224, calculate the net id, directed broadcast add, first and last add?

**Q** if the IP address of a system is 61.119.189.176, with subnet mask 255.255.192.0, calculate the net id, directed broadcast add, first and last add?

**Q** how to identify to which subnet the incoming packet from outside the network must be delivered.

- Take Subnet mask in binary notation
- Take ip address of the incoming packet in binary notation
- Do bit wise and operation
- we will get Subnet id

Net id	Subnet Mask	Interface
200.1.2.0	255.255.255.192	w
200.1.2.64	255.255.255.192	x
200.1.2.128	255.255.255.192	y
200.1.2.192	255.255.255.192	z
0.0.0.0	0.0.0.0	default

## GATE | GATE-CS-2015 (Set 2) | Question 65

Consider the following routing table at an IP router

Network No.	Net Mask	Next Hop
128.96.170.0	255.255.254.0	Interface 0
128.96.168.0	255.255.254.0	Interface 1
128.96.166.0	255.255.254.0	R2
128.96.164.0	255.255.254.0	R3
0.0.0.0	Default	R4

For each IP address in Group-I identify the correct choice of the next hop from Group-II using the entries from the routing table above.

List-I

List-II

- |                   |                |
|-------------------|----------------|
| A. 128.96.171.92  | 1. Interface 0 |
| B. 128.96.167.151 | 2. Interface 1 |
| C. 128.96.163.121 | 3. R2          |
| D. 128.96.165.121 | 4. R3          |
|                   | 5. R4          |

Codes :

A B C D

- (a) 1 3 5 4  
(b) 1 4 2 5  
(c) 2 3 4 5  
(d) 2 3 5 4

**Ans: a**

# Types of Subnetting

Subnets can be of two types:

1. Fixed Length Subnetting
2. Variable Length Subnetting

## Fixed Length Subnetting

- Fixed length subnetting (classful subnetting) divides the network into subnets such that:
  - All the subnets are of same size.
  - All the subnets have equal number of hosts.
  - All the subnets have same subnet mask.

## Variable Length Subnetting

- Variable length subnetting (classless subnetting) divides the network into subnets such that:
  - All the subnets are not of same size.
  - All the subnets do not have equal number of hosts.
  - All the subnets do not have same subnet mask.

## Point to Note:

- For, dividing a subnet into three subnets we will first divide the subnet into two parts and will then further divide it into one of them into two parts.

**Q** Consider we have a big single network having IP Address 200.1.2.0. We want to do subnetting and divide this network into 3 subnets, such that first contains 128 hosts, and other two contains 64 hosts each?

### **1st Subnet**

- Total number of IP Addresses =  $2^7 = 128$
- Total number of hosts that can be configured =  $128 - 2 = 64$
- Range of IP Addresses = [200.1.2.0, 200.1.2.127]
- IP Address of the subnet / Subnet id = 200.1.2.0
- Direct Broadcast Address = 200.1.2.**01111111** = 200.1.2.127
- Limited Broadcast Address = 255.255.255.255
- Subnet Mask: 255.255.255.128

### **2nd Subnet**

- Total number of IP Addresses =  $2^6 = 64$
- Total number of hosts that can be configured =  $64 - 2 = 62$
- Range of IP Addresses = [200.1.2.128, 200.1.2.191]
- IP Address of the subnet / Subnet id = 200.1.2.128
- Direct Broadcast Address = 200.1.2.**10111111** = 200.1.2.191
- Limited Broadcast Address = 255.255.255.255
- Subnet Mask: 255.255.255.192

### **3rd Subnet**

- Total number of IP Addresses =  $2^6 = 64$
- Total number of hosts that can be configured =  $64 - 2 = 62$
- Range of IP Addresses = [200.1.2.192, 200.1.2.255]
- IP Address of the subnet / Subnet id = 200.1.2.192
- Direct Broadcast Address = 200.1.2.**11111111** = 200.1.2.255
- Limited Broadcast Address = 255.255.255.255
- Subnet Mask: 255.255.255.192

**Q** Consider we have a big single network having IP Address 200.1.2.0. We want to do subnetting and divide this network into 3 subnets, such that first and second contains 64 hosts, and third contains 128 hosts each?

### **1st Subnet**

- Total number of IP Addresses =  $2^6 = 64$
- Total number of hosts that can be configured =  $64 - 2 = 62$
- Range of IP Addresses = [200.1.2.0, 200.1.2.63]
- IP Address of the subnet / Subnet id = 200.1.2.0
- Direct Broadcast Address = 200.1.2.**00**111111 = 200.1.2.127
- Limited Broadcast Address = 255.255.255.255
- Subnet Mask: 255.255.255.192

### **2st Subnet**

- Total number of IP Addresses =  $2^6 = 64$
- Total number of hosts that can be configured =  $64 - 2 = 62$
- Range of IP Addresses = [200.1.2.0, 200.1.2.63]
- IP Address of the subnet / Subnet id = 200.1.2.0
- Direct Broadcast Address = 200.1.2.**00**111111 = 200.1.2.127
- Limited Broadcast Address = 255.255.255.255
- Subnet Mask: 255.255.255.192

### **3rd Subnet**

- Total number of IP Addresses =  $2^7 = 128$
- Total number of hosts that can be configured =  $128 - 2 = 126$
- Range of IP Addresses = [200.1.2.128, 200.1.2.255]
- IP Address of the subnet / Subnet id = 200.1.2.128
- Direct Broadcast Address = 200.1.2.**11**111111 = 200.1.2.255
- Limited Broadcast Address = 255.255.255.255
- Subnet Mask: 255.255.255.128

Note:

- Practically we always take most significant bits of the host id to create sub-net.
- Theoretically we can take any bits from the host id for e.g. 255.255.255.1

**Examples:** If the subnet mask 255.255.255.128 belongs to class C, find the total number of subnets and Number of hosts in each subnet.

Since it is given that the subnet mask is of class C. That is, the first three octets represent the network ID and the remaining 1 octet represents Host ID.

255.255.255.128 = 11111111. 11111111. 11111111. 10000000

Now, it is clearly visible that 1 bit is borrowed from the host ID so, total number of subnets are:

$$2^1 = 2$$

And the number of hosts in each subnet will be:  $2^7 - 2 = 126$

**Q** Suppose computers A and B have IP addresses 10.105.1.113 and 10.105.1.91 respectively and they both use the same netmask N. Which of the values of N given below should not be used if A and B should belong to the same network? **(Gate-2010) (2 Marks)**

**(A)** 255.255.255.0

**(B)** 255.255.255.128

**(C)** 255.255.255.192

**(D)** 255.255.255.224

**Answer: (D)**

**Q** If a class B network on the Internet has a subnet mask of 255.255.248.0, what is the maximum number of hosts per subnet? **(Gate-2008) (2 Marks)**

**(A)** 1022

**(B)** 1023

**(C)** 2046

**(D)** 2047

**Answer (C)**

**Q** Host X has IP address 192.168.1.97 and is connected through two routers R1 and R2 to another host Y with IP address 192.168.1.80. Router R1 has IP addresses 192.168.1.135 and 192.168.1.110. R2 has IP addresses 192.168.1.67 and 192.168.1.155. The netmask used in the network is 255.255.255.224. Which IP address should X configure its gateway as? **(Gate-2008) (2 Marks)**

**(A)** 192.168.1.67

**(B)** 192.168.1.110

**(C)** 192.168.1.135

**(D)** 192.168.1.155

**Answer: (B)**

**Q** which ip address should x configure its gateway as? **(Gate-2008) (2 Marks)**

- a) 192.168.1.67
- c) 192.168.1.135

- b) 192.168.1.110
- d) 192.168.1.155

**Q** The address of a class B host is to be split into subnets with a 6-bit subnet number. What is the maximum number of subnets and the maximum number of hosts in each subnet?

**(Gate-2007) (2 Marks)**

- (A)** 62 subnets and 262142 hosts.
- (C)** 62 subnets and 1022 hosts.

- (B)** 64 subnets and 262142 hosts.
- (D)** 64 subnets and 1024 hosts.

**Answer: (C)**

**Q** A sub netted Class B network has the following broadcast address: 144.16.95.255. Its subnet mask **(Gate-2006) (2 Marks)**

- (A)** is necessarily 255.255.224.0
- (B)** is necessarily 255.255.240.0
- (C)** is necessarily 255.255.248.0
- (D)** could be any one of 255.255.224.0, 255.255.240.0, 255.255.248.0

**Answer: (D)**

**Q** An organization has a class B network and wishes to form subnets for 64 departments. The subnet mask would be: **(Gate-2005) (1 Marks)**

- (a)** 255.255.0.0
- (c)** 255.255.128.0
- (b)** 255.255.64.0
- (d)** 255.255.252.0

**Answer (d)**

**Q** A company has a class C network address of 204.204.204.0. It wishes to have three subnets, one with 100 hosts and two with 50 hosts each. Which one of the following options represents a feasible set of subnet address/subnet mask pairs? **(Gate-2005) (2 Marks)**

<b>(A)</b> 204.204.204.128/255.255.255.192 204.204.204.0/255.255.255.128 204.204.204.64/255.255.255.128	<b>(B)</b> 204.204.204.0/255.255.255.192 204.204.204.192/255.255.255.128 204.204.204.64/255.255.255.128
<b>(C)</b> 204.204.204.128/255.255.255.128 204.204.204.192/255.255.255.192 204.204.204.224/255.255.255.192	<b>(D)</b> 204.204.204.128/255.255.255.128 204.204.204.64/255.255.255.192 204.204.204.0/255.255.255.192

**Answer: (D)**

**Q** A subnet has been assigned a subnet mask of 255.255.255.192. What is the maximum number of hosts that can belong to this subnet? **(Gate-2004) (1 Marks)**

(A) 14

(B) 30

(C) 62

(D) 126

Answer: (C)

**Q** The subnet mask for a particular network is 255.255.31.0. Which of the following pairs of IP addresses could belong to this network? **(Gate-2003) (2 Marks)**

(A) 172.57.88.62 and 172.56.87.233

(B) 10.35.28.2 and 10.35.29.4

(C) 191.203.31.87 and 191.234.31.88

(D) 128.8.129.43 and 128.8.161.55

Answer: (D)

## Address Depletion

- The addresses were not distributed properly as class A and B are usually very large for any organization and class C is usually very small
- flexibility is not there in classful addressing, we cannot have the exact allocation as we want for e.g. if some company wants 50 IP address then must go for 256, resulting into address depletion.
- Wastage of addresses, for example: Class E addresses were almost never used, wasting the whole class.
- **Conclusion:** The Internet was faced with the problem of the addresses being rapidly used up, resulting in no more addresses available for organizations and individuals that needed to be connected to the Internet.

## Classless Addressing (Blocks/Network)

- Classless Addressing is an improved IP Addressing system.
- The class privilege is removed from the distribution to compensate for the address depletion, so no class.
- Here we can ask exact set of IP address which are required and a Variable-length blocks are assigned which satisfy the request.

## CIDR Notation

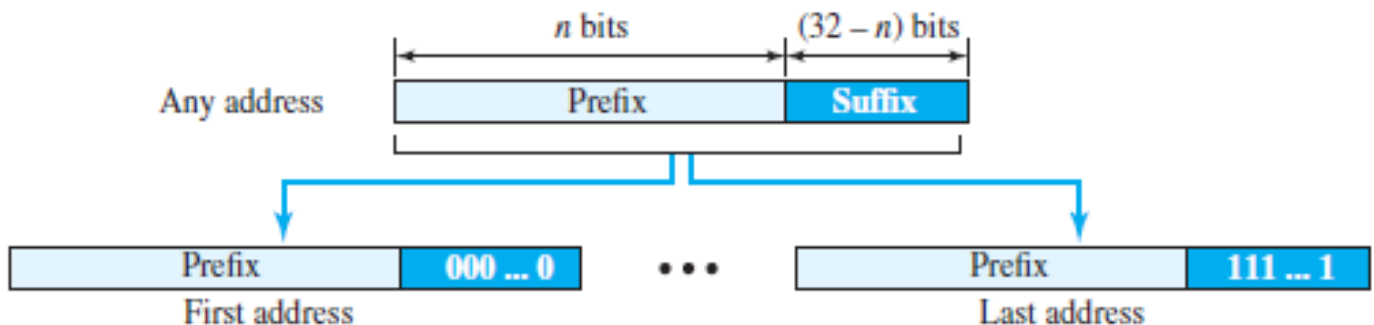
- The question is as there are no classes, how to identify block id and host id, as address in classless addressing does not define the block or network to which the address belongs.
- To solve this problem now we have a new CIDR notation, this notation is informally referred to as *slash notation* and formally as **classless interdomain routing** or **CIDR**.
- To find the prefix(net\_id),  $n$  is added to the address, separated by a slash.
- $n$  represent number of bits in net\_id



**Examples:**  
12.24.76.8/8  
23.14.67.92/12  
220.8.24.255/25

## Extracting Information from an Address

- The number of addresses in the block is found as  $N = 2^{32-n}$ .
- To find the first address, we keep the  $n$  leftmost bits and set the  $(32 - n)$  rightmost bits all to 0s.
- To find the last address, we keep the  $n$  leftmost bits and set the  $(32 - n)$  rightmost bits all to 1s.



Example: Address: 167.199.170.82/27 (10100111 11000111 10101010 01010010)

First address can be found by setting 32-n rightmost bits as 0 as shown below:

**First address:** 167.199.170.64/27 (10100111 11000111 10101010 01000000)

Last address can be found by setting 32-n rightmost bits as 1.

**Last Address:** 167.199.170.95/27 10100111 11000111 10101010 01011111

Q one of the address of the block is 17.63.110.24/27, find the no of address, bid, broadcast address?

Q one of the address of the block is 110.23.120.14/20+, find the no of address, bid, broadcast address?

## Address Mask

- The address mask is a 32-bit number in which the  $n$  leftmost bits are set to 1s and the rest of the bits ( $32 - n$ ) are set to 0s.
- It is another way to find the first and last addresses in the block.
- Using the three bit-wise operations NOT, AND, and OR a computer can find:
  1. The number of addresses in the block  $N = \text{NOT}(\text{mask}) + 1$ .
  2. The first address in the block = (Any address in the block) **AND** (mask).
  3. The last address in the block = (Any address in the block) **OR** [(**NOT** (mask))].

Example: A classless address is given as 167.199.170.82/27.

The mask by setting 27 leftmost bits to 1 and 5 rightmost bits to 0 in dotted-decimal notation is: 255.255.255.224 (11111111 11111111 11111111 11100000)

- Number of addresses in the block:  $N = \text{NOT}(\text{mask}) + 1 =$ 
  - 00000000 00000000 00000000 00011111 + 1 = 0.0.0.31 + 1 = 32 addresses
- First address: First = (address) **AND** (mask) =
  - 10100111 11000111 10101010 01010010
  - 11111111 11111111 11111111 11100000
  - 167.199.170.64

- Last address: Last = (address) **OR (NOT mask)** =
  - 10100111 11000111 10101010 01010010
  - 00000000 00000000 00000000 00011111
  - 167.199.170.255

**Q** In the network 200.10.11.144/27, the fourth octet (in decimal) of the last IP address of the network which can be assigned to a host is \_\_\_\_\_ **(Gate-2015) (2 Marks)**

**Q** An Internet Service Provider (ISP) has the following chunk of CIDR-based IP addresses available with it: 245.248.128.0/20. The ISP wants to give half of this chunk of addresses to Organization A, and a quarter to Organization B, while retaining the remaining with itself. Which of the following is a valid allocation of addresses to A and B? **(Gate-2012) (2 Marks)**

- (A)** 245.248.136.0/21 and 245.248.128.0/22
- (B)** 245.248.128.0/21 and 245.248.128.0/22
- (C)** 245.248.132.0/22 and 245.248.132.0/21
- (D)** 245.248.136.0/24 and 245.248.132.0/21

**Answer: (A)**

## Rules for Creating CIDR Block (Network)

- All the IP Addresses in the CIDR block must be contiguous.
- The size of the block (total number of IP Addresses contained in the block) must be presentable as power of 2, size of any CIDR block will always be in the form  $2^1, 2^2, 2^3, 2^4, 2^5$  and so on. (calculation can be easy)
- First IP Address of the block must be divisible by the size of the block. (so that we get the host id from all 0 to all 1)

**Q** Consider a block of IP Addresses ranging from 100.1.2.32 to 100.1.2.47.

1. Is it a CIDR block?
2. If yes, give the CIDR representation?

- We need to conform 3 rules here:
  - Are the addresses contiguous? 32 to 47 is contiguous.
  - Are the total addresses in power of 2? Total addresses =  $47 - 32 + 1 = 16 = 2^4$
  - Is the first address divisible by size of the block?

- First IP address = 100.1.2.32 = 100.1.2.00100000, it is divisible by  $2^4$  as last four LSB's are 0. So, the given block is a CIDR.
- **CIDR Notation**
  - Since the block has  $2^4$  total addresses, thus 4 bits are required in HostID therefore the number of bits present in the network ID part is:  $32-4 = 28$ .
  - So, the CIDR notation is: 100.1.2.32 / 28

**Q** Consider a block of IP Addresses ranging from 20.10.30.32 to 20.10.30.63

1. Is it a CIDR block?
2. If yes, give the CIDR representation?

**Q** Consider a block of IP Addresses ranging from 150.10.20.64 to 150.10.20.127

1. Is it a CIDR block?
2. If yes, give the CIDR representation?

**Point to note:**

- Any binary pattern is divisible by  $2^n$ , if and only if its least significant n bits are 0.
- Example: 01100100.00000001.00000010.01000000 (i.e. 100.1.2.64).
  - It is divisible by  $2^5$  since its least significant 5 bits are zero.
  - It is divisible by  $2^6$  since its least significant 6 bits are zero.
  - It is not divisible by  $2^7$  since its least significant 7 bits are not zero.

## Subnetting in CIDR

**Q** Consider the network having IP Address 20.30.40.10/25 Divide this network into two subnets.

20.30.40.0/26

20.30.40.64/26

**Q** Consider the network having IP Address 20.30.40.10/25 Divide this network into Four subnets.

20.30.40.0/27

20.30.40.32/27

20.30.40.64/27

20.30.40.96/27

**Q** Consider the network having IP Address 20.30.40.10/25 Divide this network into Three subnets.

20.30.40.0/26

20.30.40.64/27

20.30.40.96/27

**Q** Consider the network having IP Address 20.30.40.10/25 Divide this network into Three subnets.

20.30.40.0/26

20.30.40.32/27

20.30.40.64/27

**Q** Consider the network having IP Address 40.30.10.20/20 Divide this network into Three subnets.

40.30.0.0/21 - 40.30.7.255/21

40.30.8.0/22 - 40.30.11.255/22

40.30.12.0/22 - 40.30.15.255/22

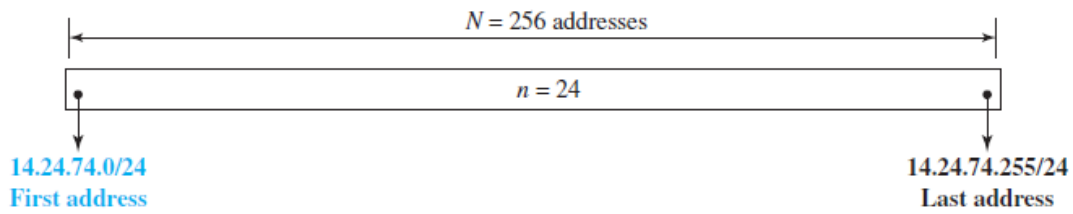
**Example:** An organization is granted a block of addresses with the beginning address 14.24.74.0/24. The organization needs to have 3 sub blocks of addresses to use in its three subnets: one sub-block of 10 addresses, one sub block of 60 addresses, and one sub block of 120 addresses. Design the sub blocks.

Clearly, There are  $2^{32-24} = 256$  addresses in this block.

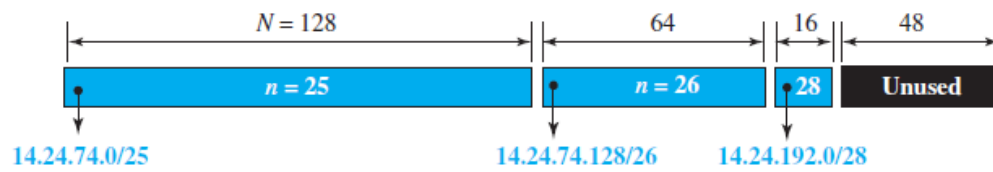
- The first address is 14.24.74.0/24.
- The last address is 14.24.74.255/24.

We assign addresses to subblocks, starting with the largest and ending with the smallest one.

- The largest subblock requires 120 addresses, it is not a power of 2. So, we allocate 128 addresses.
  - The subnet mask for this subnet can be found as  $n_1 = 32 - \log_2 128 = 25$ .
  - The first address in this block is 14.24.74.0/25.
  - The last address is 14.24.74.127/25.
- The second largest sub block, it requires 60 addresses, since it is not a power of 2 either. We allocate 64 addresses.
  - The subnet mask for this subnet can be found as  $n_2 = 32 - \log_2 64 = 26$ .
  - The first address in this block is 14.24.74.128/26.
  - The last address is 14.24.74.191/26.
- The number of addresses in the smallest sub block, which requires 10 addresses, is not a power of 2 either. We allocate 16 addresses.
  - The subnet mask for this subnet can be found as  $n_3 = 32 - \log_2 16 = 28$ .
  - The first address in this block is 14.24.74.192/28.
  - The last address is 14.24.74.207/28.



a. Original block



- If we add all addresses in the previous sub-blocks ( $128 + 64 + 16$ ), the result is 208 addresses, which means 48 addresses are left in reserve.

## Designing subnets for CIDR Notations

- Assume:
  - The total number of addresses granted to the organization is **N**
  - The prefix length is **n**
  - The assigned number of addresses to each sub-network is **N<sub>sub</sub>**
  - The prefix length for each sub-network is **n<sub>sub</sub>**.

Then,

- The number of addresses in each sub-network should be a power of 2.
- The prefix length for each sub-network should be found using the following formula: **n<sub>sub</sub> = 32 - log<sub>2</sub>N<sub>sub</sub>**
- The starting address in each sub-network should be divisible by the number of addresses in that sub-network. This can be achieved if we first assign addresses to larger sub-networks.

## Disadvantages of Subnetting

- Subnetting leads to loss of IP Addresses, in each subnet we lose two IP addresses one for network address and one for DBA.
- Communication process gets complicated.

## Super netting in Classful addressing

- Subnetting increase size of routing table, super netting is a perception so a counter idea is also possible which is super netting
- The size of a class C block with a maximum number of 256 addresses did not satisfy the needs of most organizations. Even a midsize organization needed more addresses.
- One solution was super netting. In super netting, an organization can combine several blocks to create a larger range of addresses. In other words, several networks are combined to create a super network or a supernet.
- An organization can apply for a set of class C blocks instead of just one. For example, an organization that needs 1000 addresses can be granted four contiguous class C blocks. The organization can then use these addresses to create one super network.
- Super netting decreases the number of 1s in the mask.

## Super netting / Aggregation with CIDR

- Rules for Super netting in CIDR
  - All network should be contiguous
  - Size of all the network should be same
  - first net id should be divisible by size of the block

Let us take an example:

Perform CIDR aggregation on the following IP Addresses-

**128.56.24.0/24**

**128.56.25.0/24**

**128.56.26.0/24**

**128.56.27.0/24**

- Are the blocks contiguous? Clearly the blocks are contiguous in nature.
- Are the total number of IP addresses in power of 2?  
Clearly we are having 24 bits as network ID in each 4 blocks, host ID is:  $32 - 24 = 8$  bits.  
Total addresses for 4 blocks:  $2^8 + 2^8 + 2^8 + 2^8 = 2^{10}$   
So, the total addresses are also power of 2.
- Is first address divisible by total addresses?  
The first address: 128.56.24.0/24 = 128.56.00011000.00000000 is divisible by  $2^{10}$  since its 10 least significant bits are zero.

Now, all the rules for CIDR block are **satisfied**.

To aggregate them, we have total  $2^{10}$  addresses that means we need to have 10 bits for host ID and 22 bits for network ID

The CIDR representation of aggregation would be: **128.56.24.0/22**

Q Consider the following networks and merger them to have a supernet

200.1.0.0/24

200.1.1.0/24

200.1.2.0/24

200.1.2.0/24

200.1.0.0/22

Q Consider the following networks and merger them to have a supernet

100.1.2.0/25

100.1.2.128/26

100.1.2.192/26

100.1.2.0/24

- ISP provides four things
  - IP address
  - DGW (router which is connecting us to internet)
  - SM (using subnet mask we understand weather the destination is in our network or some other network)
  - DNS

**Q** Two computers  $C_1$  and  $C_2$  are configured as follows.  $C_1$  has IP address 203.197.2.53 and netmask 255.255.128.0.  $C_2$  has IP address 203.197.75.201 and netmask 255.255.192.0.

which one of the following statements is true? **(Gate-2006) (2 Marks)**

- (A)**  $C_1$  and  $C_2$  both assume they are on the same network
- (B)**  $C_2$  assumes  $C_1$  is on same network, but  $C_1$  assumes  $C_2$  is on a different network
- (C)**  $C_1$  assumes  $C_2$  is on same network, but  $C_2$  assumes  $C_1$  is on a different network
- (D)**  $C_1$  and  $C_2$  both assume they are on different networks.

**Answer (C)**

## Problem

- Here problem is when a router receives an IP packet with destination address then how can it decide where to send this packet.
- This decision at the router is taken with the help of a routing table, actually the process of designing a routing table is called routing. Taking a packet and sending it to some path is actually switching.
- One question, is it possible that a packet reaches its destination without routing table, actually yes, the process is called flooding.
- That is if we do not know to which way a packet should be sent then, we can send it to all possible way and then we can it all possible way then then we can be sure that at least one packet will reach the destination.
- Flooding Advantage
  - No Routing is required
  - Shortest Path is guaranteed
  - Highly Reliable
- Flooding Disadvantage
  - Duplicate packets will arrive
  - Traffic is high
- Note: In Defence network mostly, flooding is used
- The source host needs no forwarding table because it delivers its packet to the default router in its local network.

## ROUTING

- A routing table contains information about the network, and it helps deciding to which interface the incoming packet should be sent in order to reach destination.
- Actually, the process of designing routing table is called Routing algorithm.
- Routing table can be either static or dynamic.
- A static table is one with manual entries, i.e. if someone has information about all the routers in the world and can compute the shortest distance from one router to another and can upload the routing table for each router then it is called Static Routing.
  - Now-a days as we know the internet is so complex that no one can have complete information about the entire internet
  - internet keeps on changing some new routers come and some old routers goes, means topology and traffic keeps on changing.
  - Conclusion Static routing is not possible.

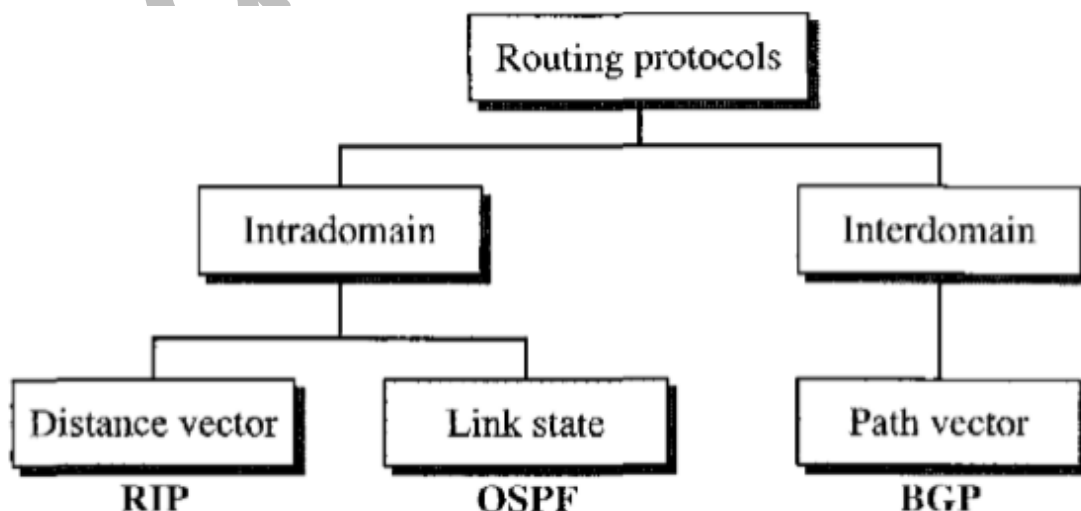
- A dynamic table, on the other hand, is one that is updated automatically when there is a change somewhere in the internet either in topology or traffic.

## UNICAST ROUTING PROTOCOLS

- Routing protocols have been created in response to the demand for dynamic routing tables. A routing protocol is a combination of rules and procedures that lets routers in the internet inform each other of changes.
- It allows routers to share whatever they know about the internet or their neighbourhood. The sharing of information allows a router in Delhi to know about the failure of a network in Agra.
- The routing protocols also include procedures for combining information received from other routers.
- Router to have several routing tables based on the required type of service.

## Intra-domain and Interdomain Routing

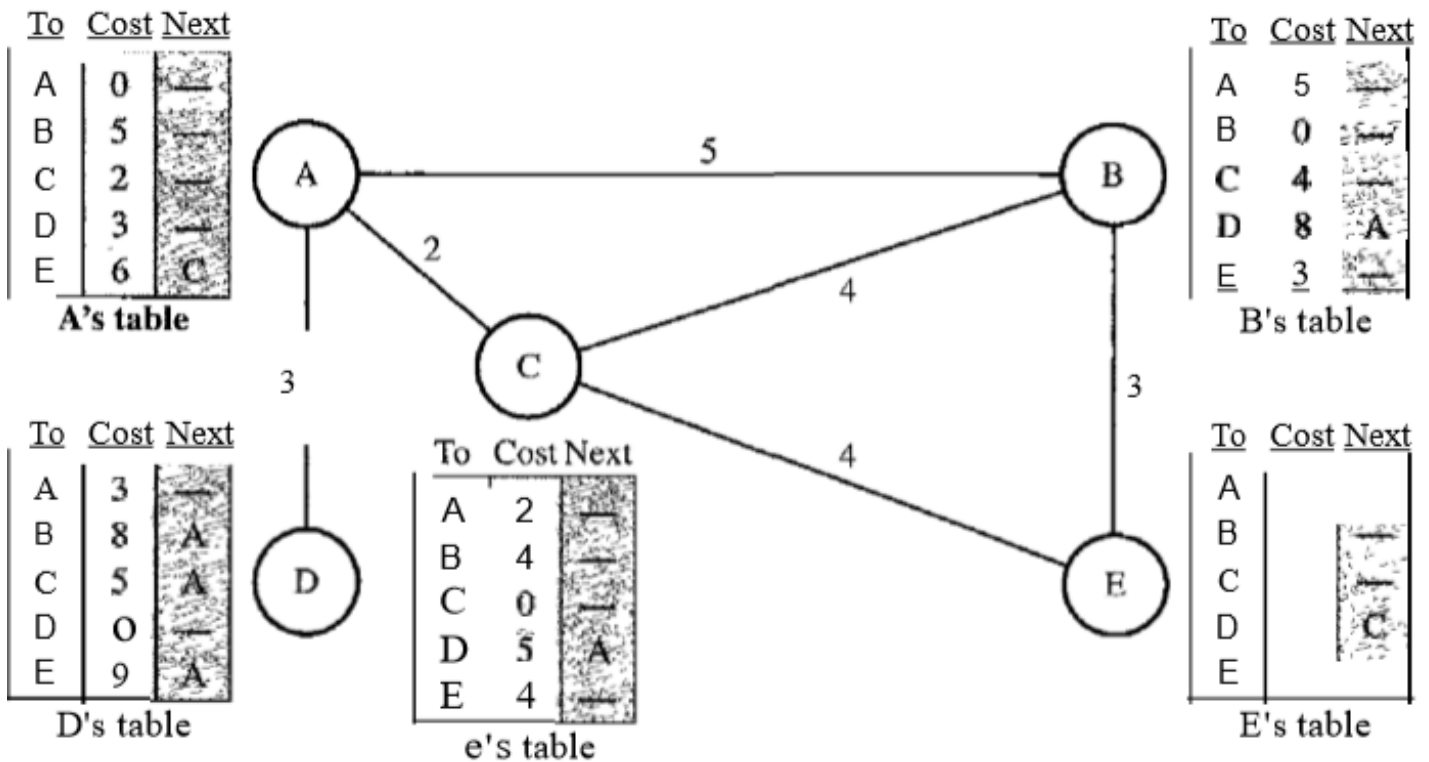
- Today, an internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers. For this reason, an internet is divided into autonomous systems.
- An autonomous system (AS) is a group of networks and routers under the authority of a single administration. Routing inside an autonomous system is referred to as intradomain routing. Routing between autonomous systems is referred to as interdomain routing.
- Each autonomous system can choose one or more intradomain routing protocols to handle routing inside the autonomous system. However, only one interdomain routing protocol handles routing between autonomous systems



- Which of the available pathways is the optimum pathway? What is the definition of the term optimum? One approach is to assign a cost for passing through a network. We call this cost a metric.
- However, the metric assigned to each network depends on the type of protocol. Some simple protocols, such as the Routing Information Protocol (RIP), treat all networks as equals. The cost of passing through a network is the same; it is one hop count. So if a packet passes through 10 networks to reach the destination, the total cost is 10 hop counts.
- Other protocols, such as Open Shortest Path First (OSPF), allow the administrator to assign a cost for passing through a network based on the type of service required. A route through a network can have different costs (metrics). For example, if maximum throughput is the desired type of service, a satellite link has a lower metric than a fibre-optic line. On the other hand, if minimum delay is the desired type of service, a fibre-optic line has a lower metric than a satellite link.

## Distance Vector Routing

- In distance vector routing, the least-cost route between any two nodes is the route with minimum distance.
- In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node. The table at each node also guides the packets to the desired node by showing the next stop in the route (next-hop routing).



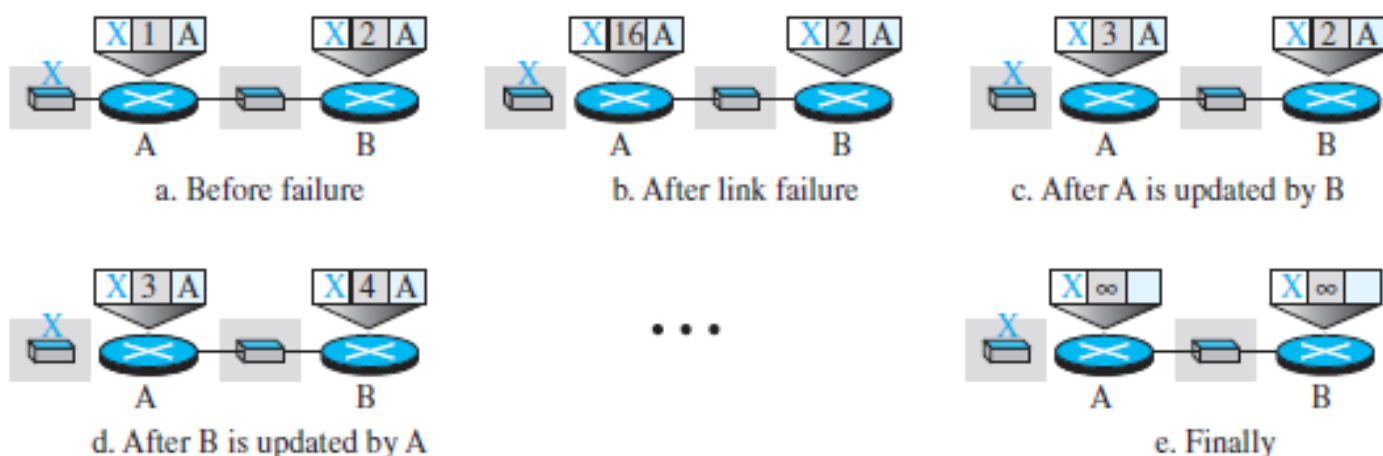
- The whole idea of distance vector routing is the sharing of information between neighbours. Although node A does not know about node E, node C does. So, if node C shares its routing table with A, node A can also know how to reach node E.
- On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbours, can improve their routing tables if they help each other.

- There is only one problem. How much of the table must be shared with each neighbour? A node is not aware of a neighbour's table. The best solution for each node is to send its entire table to the neighbour and let the neighbour decide what part to use and what part to discard.
- However, the third column of a table (next hop) is not useful for the neighbour. When the neighbour receives a table, this column needs to be replaced with the sender's name. If any of the rows can be used, the next node is the sender of the table.
- A node therefore can send only the first two columns of its table to any neighbour. In other words, sharing here means sharing only the first two columns.
  
- The question now is, when does a node send its partial routing table (only two columns) to all its immediate neighbours? The table is sent both periodically and when there is a change in the table.
- Periodic Update A node sends its routing table, normally every 30 s, in a periodic update. The period depends on the protocol that is using distance vector routing.
- Triggered Update A node sends its two-column routing table to its neighbours anytime there is a change in its routing table. This is called a triggered update. The change can result from the following.
  - A node receives a table from a neighbour, resulting in changes in its own table after updating.
  - A node detects some failure in the neighbouring links which results in a distance change to infinity.

## Two-Node Loop Instability

- If a link is broken (cost becomes infinity), every other router should be aware of it immediately, but in distance-vector routing, this takes some time as the algorithms is designed in such a way that it reports the minimum first.
- The problem is referred to as *count to infinity*. It sometimes takes several updates before the cost for a broken link is recorded as infinity by all routers.

- Consider three nodes A, B and X



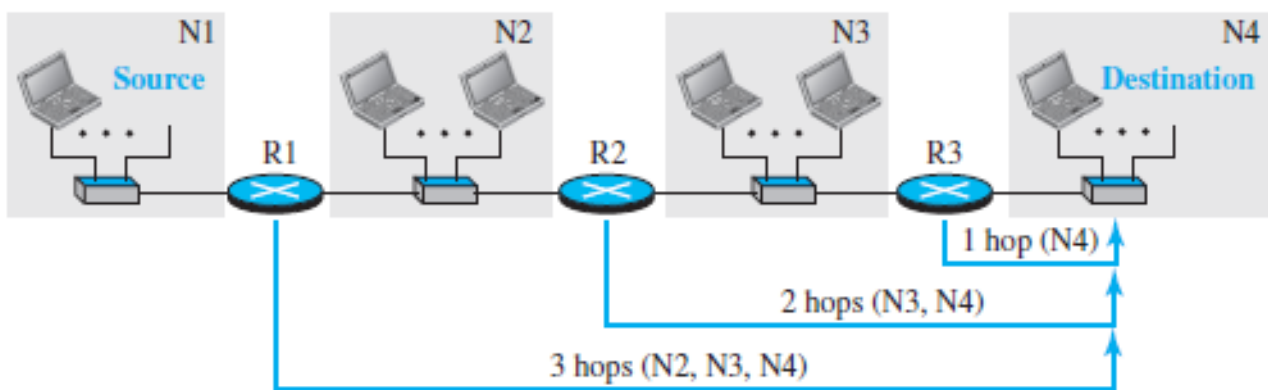
- Initially in fig.(a) nodes A and B know how to reach node X.
  - But suddenly, the link between A and X fails. Node A changes its table.
  - If A can send its table to B immediately, everything is fine.
  - The system becomes unstable if B sends its forwarding table to A before receiving A's forwarding table.
  - Node A receives the update assumes that B has found a way to reach X, it updates its forwarding table.
  - Now A sends its new update to B. Now B thinks that something has been changed around A and updates its forwarding table.
  - The cost of reaching X increases gradually until it reaches infinity.
  - At this moment, both A and B know that X cannot be reached.
- During this time the system is not stable. Node A thinks that the route to X is via B; node B thinks that the route to X is via A.
  - If A receives a packet destined for X, the packet goes to B and then comes back to A and vice versa. Packets bounce between A and B, creating a two-node loop problem.

## Split Horizon

- In this strategy, instead of flooding the table through each interface, each node sends only part of its table through each interface.
- If, according to its table, node B thinks that the optimum route to reach X is via A, it does not need to advertise this piece of information to A; the information has come from A (A already knows).
- Taking information from node A, modifying it, and sending it back to node A creates the confusion. In our scenario, node B eliminates the last line of its routing table before it sends it to A.
- In this case, node A keeps the value of infinity as the distance to X. Later when node A sends its routing table to B, node B also corrects its routing table. The system becomes stable after the first update: both node A and B know that X is not reachable

## Routing Information Protocol (RIP)

- The **Routing Information Protocol (RIP)** is one of the most widely used intradomain routing protocols based on the distance-vector routing algorithm
- **Metric**
  - **Hop Count:** To make the implementation of the cost simpler, the cost is defined as the number of hops, which means the number of networks (subnets) a packet needs to travel through from the source router to the final destination host.
  - The network in which the source host is connected is not counted in this calculation because the source host does not use a forwarding table; the packet is delivered to the default router.
  - *In RIP, the maximum cost of a path can be 15, which means 16 is considered as infinity.*



### Forwarding Tables

- A forwarding table in RIP is a three-column table in which the first column is the address of the destination network.
- The second column is the address of the next router to which the packet should be forwarded.
- The third column is the cost (the number of hops) to reach the destination network.

Forwarding table for R1

Destination network	Next router	Cost in hops
N1	—	1
N2	—	1
N3	R2	2
N4	R2	3

Forwarding table for R2

Destination network	Next router	Cost in hops
N1	R1	2
N2	—	1
N3	—	1
N4	R3	2

Forwarding table for R3

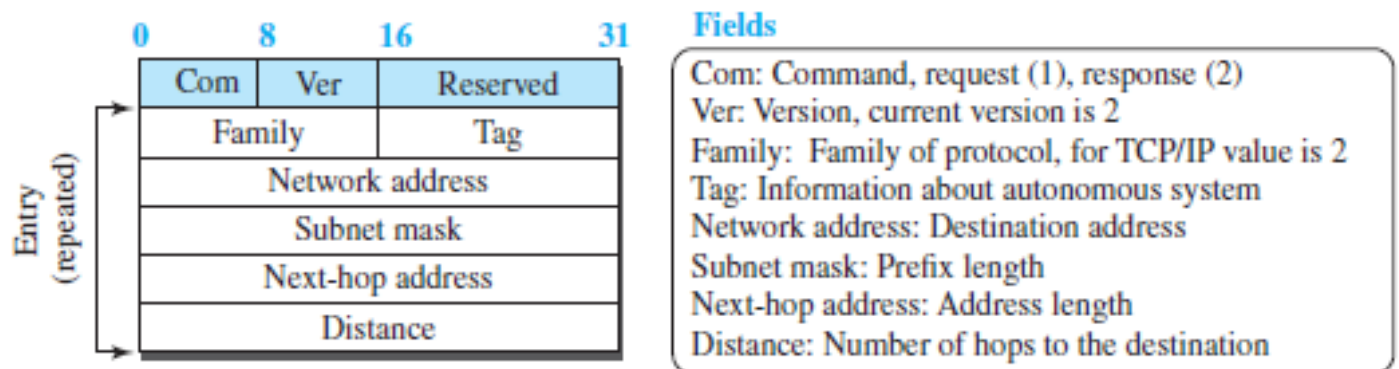
Destination network	Next router	Cost in hops
N1	R2	3
N2	R2	2
N3	—	1
N4	—	1

## RIP Implementation

- *RIP is implemented as a process that uses the service of UDP on the well-known port number 520.*
- RIP runs at the application layer, but creates forwarding tables for IP at the network layer.
- RIP has gone through two versions: RIP-1 and RIP-2.

## RIP Messages

- RIP-2 defines the format of the message as:



- RIP has two types of messages: request and response.
- A request message is sent by a router that has just come up or by a router that has some time-out entries. A request message can ask about specific entries or all entries.
- A response (or update) message can be either solicited or unsolicited.
- A solicited response message is sent only in answer to a request message. It contains information about the destination specified in the corresponding request message.
- An unsolicited response message, is sent periodically, every 30 seconds or when there is a change in the forwarding table.

## RIP Algorithm

- RIP implements the same algorithm as the distance-vector routing algorithm, but some changes need to be made to the algorithm to enable a router to update its forwarding table:
  - Instead of sending only distance vectors, a router needs to send the whole contents of its forwarding table in a response message.
  - The receiver adds one hop to each cost and changes the next router field to the address of the sending router.
  - The received router selects the old routes as the new ones except in the following three cases:
    - If the received route does not exist in the old forwarding table, it should be added to the route.
    - If the cost of the received route is lower than the cost of the old one, the received route should be selected as the new one.
    - If the cost of the received route is higher than the cost of the old one, but the value of the next router is the same in both routes, the received route should be selected as the new one.
  - The new forwarding table needs to be sorted according to the destination route (mostly using the longest prefix first).

## Timers in RIP

RIP uses three timers to support its operation:

- The **periodic timer** controls the advertising of regular update messages. Generally used to prevent all routers sending their messages at the same time and creating excess traffic.
- The **expiration timer** governs the validity of a route.
  - When a router receives update information for a route, the expiration timer is set to 180 seconds for that particular route.
  - Every time a new update for the route is received, the timer is reset. If there is a problem on an internet and no update is received within the allotted 180 seconds, the route is considered expired and the hop count of the route is set to 16.
- The **garbage collection timer** is used to purge a route from the forwarding table.
  - When the information about a route becomes invalid, the router does not immediately purge that route from its table.
  - Instead, it continues to advertise the route with a metric value of 16.
  - At the same time, a garbage collection timer is set to 120 seconds for that route. When the count reaches zero, the route is purged from the table.

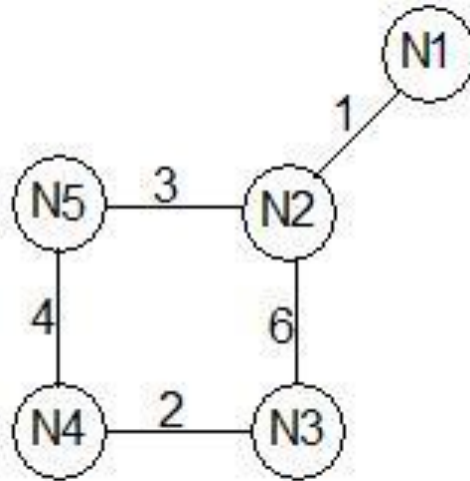
## ***Performance of RIP***

- ***Less Traffic:*** The update messages in RIP have a very simple format and are sent only to neighbours. They do not normally create traffic because the routers try to avoid sending them at the same time.
- ***Convergence of Forwarding Tables:*** since RIP allows only 15 hops in a domain (16 is considered as infinity), there is normally no problem in convergence. The only problems that may slow down convergence are count-to-infinity and loops created in the domain.
- ***Robustness:*** If there is a failure or corruption in one router, the problem will be propagated to all routers and the forwarding in each router will be affected, as it works on distance vector routing.

## **RIP**

- The Routing Information Protocol (RIP) is an intradomain routing protocol used inside an autonomous system. It is a very simple protocol based on distance vector routing. RIP implements distance vector routing directly with some considerations:
  - In an autonomous system, we are dealing with routers and networks (links). The routers have routing tables; networks do not.
  - The destination in a routing table is a network, which means the first column defines a network address.
  - The metric used by RIP is very simple; the distance is defined as the number of links (networks) to reach the destination. For this reason, the metric in RIP is called a hop count.
  - Infinity is defined as 16, which means that any route in an autonomous system using RIP cannot have more than 15 hops.
  - The next-node column defines the address of the router to which the packet is to be sent to reach its destination.

Q Consider a network with five nodes,  $N_1$  to  $N_5$ , as shown below.



The network uses a Distance Vector Routing Distance Vector Routing protocol. Once the routes have stabilized, the distance vectors at different nodes are as following.

$N_1$ :(0, 1, 7, 8, 4)

$N_2$ :(1, 0, 6, 7, 3)

$N_3$ :(7, 6, 0, 2, 6)

$N_4$ :(8, 7, 2, 0, 4)

$N_5$ :(4, 3, 6, 4, 0)

Each distance vector is the distance of the best known path at that instance to nodes,  $N_1$  to  $N_5$ , where the distance to itself is 0. Also, all links are symmetric and the cost is identical in both directions. In each round, all nodes exchange their distance vectors with their respective neighbours. Then all nodes update their distance vectors. In between two rounds, any change in cost of a link will cause the two incident nodes to change only that entry in their distance vectors.

Q The cost of link  $N_2$ - $N_3$  reduces to 2 (in both directions). After the next round of update what will be the new distance vector at node,  $N_3$ ? (GATE-2011) (2 Marks)

(A) (3, 2, 0, 2, 5)

(B) (3, 2, 0, 2, 6)

(C) (7, 2, 0, 2, 5)

(D) (7, 2, 0, 2, 6)

Answer (A)

Q After the update in the previous question, the link  $N_1$ - $N_2$  goes down.  $N_2$  will reflect this change immediately in its distance vector as cost,  $\infty$ . After the NEXT ROUND of update, what will be cost to  $N_1$  in the distance vector of  $N_3$ ? (GATE-2011) (2 Marks)

(A) 3

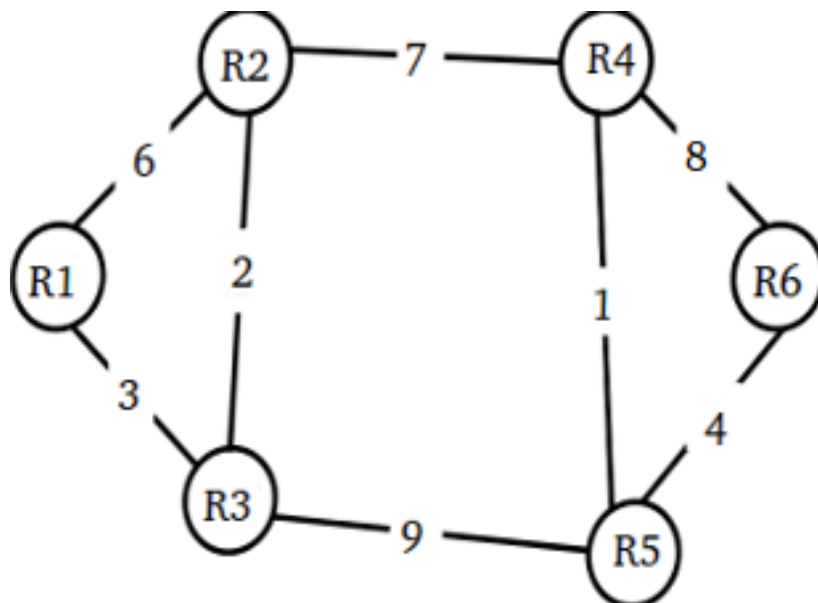
(B) 9

(C) 10

(D)  $\infty$

Answer (C)

**Q** Consider a network with 6 routers  $R_1$  to  $R_6$  connected with links having weights as shown in the following diagram



All the routers use the distance vector-based routing algorithm to update their routing tables. Each router starts with its routing table initialized to contain an entry for each neighbour with the weight of the respective connecting link. After all the routing tables stabilize, how many links in the network will never be used for carrying any data? **(GATE-2010) (2 Marks)**

**(A) 4**

**(B) 3**

**(C) 2**

**(D) 1**

**Answer (C)**

**Q** Suppose the weights of all unused links in the previous question are changed to 2 and the distance vector algorithm is used again until all routing tables stabilize. How many links will now remain unused? **(GATE-2010) (2 Marks)**

**(A) 0**

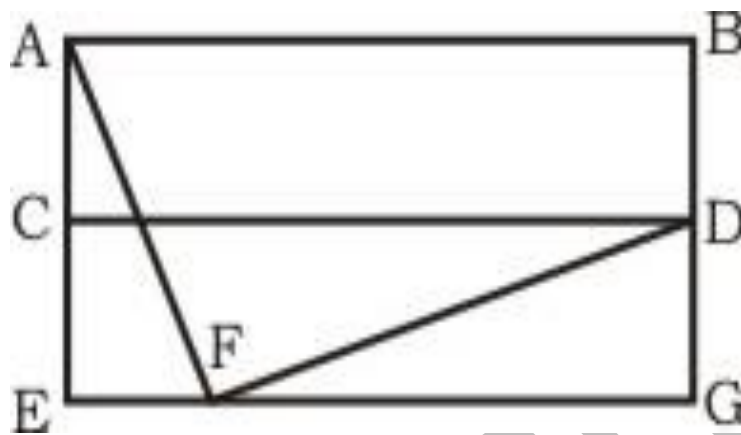
**(B) 1**

**(C) 2**

**(D) 3**

**Answer (B)**

**Q** For the network given in the figure below, the routing tables of the four nodes A, E, D and G are shown. Suppose that F has estimated its delay to its neighbours, A, E, D and G as 8, 10, 12 and 6 msec respectively and updates its routing table using distance vector routing technique. **(GATE-2007) (2 Marks)**



Routing Table of A	
A	0
B	40
C	14
D	17
E	21
F	9
G	24

Routing Table of D	
A	20
B	8
C	30
D	0
E	14
F	7
G	22

Routing Table of E	
A	24
B	27
C	7
D	20
E	0
F	11
G	22

Routing Table of G	
A	21
B	24
C	22
D	19
E	22
F	10
G	0

A	8
B	20
C	17
D	12
E	10
F	0
G	6

A	21
B	8
C	7
D	19
E	14
F	0
G	22

**A**

**B**

A	8
B	20
C	17
D	12
E	10
F	16
G	6

A	8
B	8
C	7
D	12
E	10
F	0
G	6

C

D

Answer: (A)

Sanchit Jain

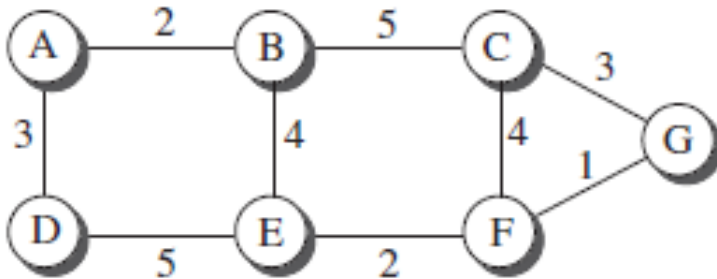
## Link State Routing

- Link state routing has a different philosophy from that of distance vector routing. In link state routing, if each node in the domain has the entire topology of the domain the list of nodes and links, how they are connected including the type, cost (metric), and condition of the links (up or down)-the node can use Dijkstra's algorithm to build a routing table.
- Each node uses the same topology to create a routing table, but the routing table for each node is unique because the calculations are based on different interpretations of the topology.
- The topology must be dynamic, representing the latest state of each node and each link. If there are changes in any point in the network (a link is down, for example), the topology must be updated for each node.
- How can a common topology be dynamic and stored in each node? No node can know the topology at the beginning or after a change somewhere in the network.
- Link state routing is based on the assumption that, although the global knowledge about the topology is not clear, each node has partial knowledge
- it knows the state (type, condition, and cost) of its links. In other words, the whole topology can be compiled from the partial knowledge of each node.
- In this algorithm the cost associated with an edge defines the state of the link.

## Building Routing Tables

- In link state routing, four sets of actions are required to ensure that each node has the routing table showing the least-cost node to every other node.
  - Creation of the states of the links by each node, called the link state packet (LSP).
  - Dissemination of LSPs to every other router, called flooding, in an efficient and reliable way.
  - Formation of a shortest path tree for each node.
  - Calculation of a routing table based on the shortest path tree
- Formation of Shortest Path Tree: Dijkstra Algorithm After receiving all LSPs, each node will have a copy of the whole topology. However, the topology is not sufficient to find the shortest path to every other node; a shortest path tree is needed.
- A tree is a graph of nodes and links; one node is called the root. All other nodes can be reached from the root through only one single route.
- A shortest path tree is a tree in which the path between the root and every other node is the shortest. What we need for each node is a shortest path tree with that node as the root.
- The Dijkstra algorithm creates a shortest path tree from a graph. The algorithm divides the nodes into two sets: tentative and permanent. It finds the neighbours of a current node, makes them tentative, examines them, and if they pass the criteria, makes them permanent.
- To find the shortest path in each step, we need the cumulative cost from the root to each node, which is shown next to the node. Nodes and lists with the cumulative costs.

- The collection of state for all links is called the **link-state database (LSDB)**.
- There is only one LSDB for the whole internet; each node needs to have a duplicate of it to be able to create the least-cost tree.

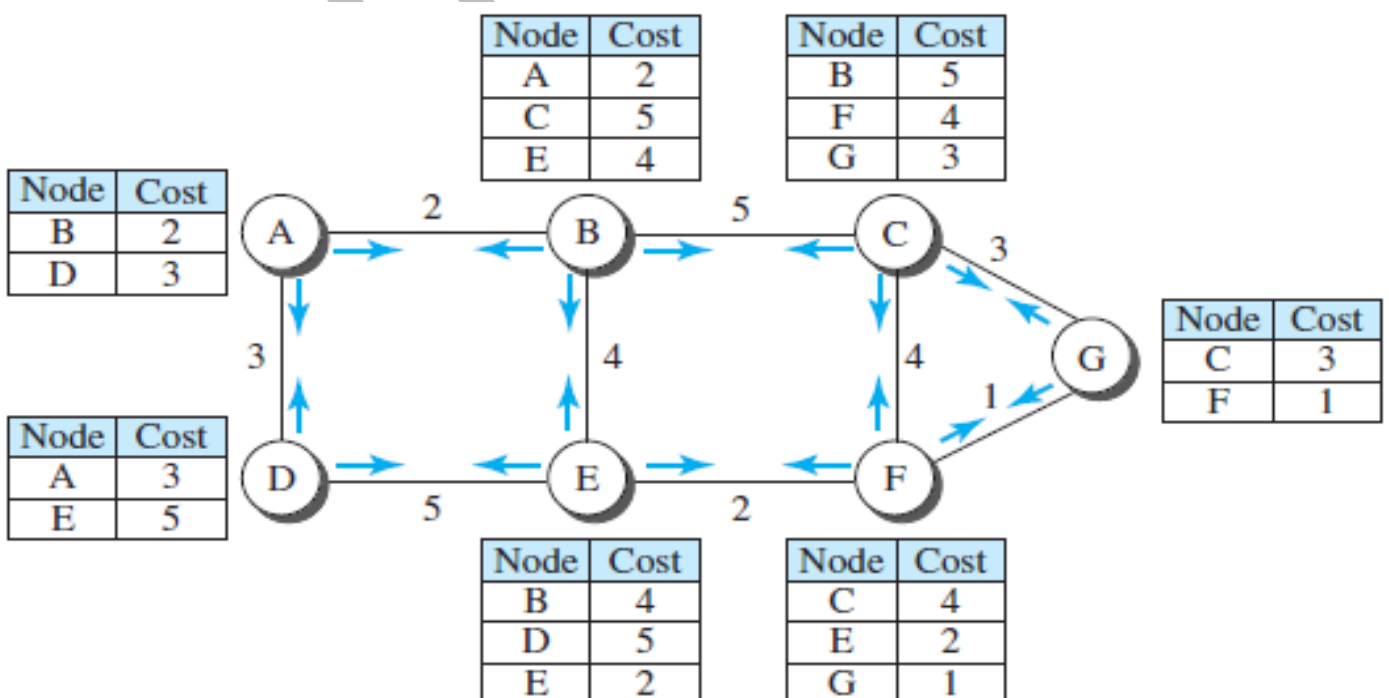


a. The weighted graph

	A	B	C	D	E	F	G
A	0	2	$\infty$	3	$\infty$	$\infty$	$\infty$
B	2	0	5	$\infty$	4	$\infty$	$\infty$
C	$\infty$	5	0	$\infty$	$\infty$	4	3
D	3	$\infty$	$\infty$	0	5	$\infty$	$\infty$
E	$\infty$	4	$\infty$	5	0	2	$\infty$
F	$\infty$	$\infty$	4	$\infty$	2	0	1
G	$\infty$	$\infty$	3	$\infty$	$\infty$	1	0

b. Link state database

- The LSDB can be represented as a two-dimensional array (matrix) in which the value of each cell defines the cost of the corresponding link.
- Each node can create this LSDB that contains information about the whole internet by a process called **flooding**.
- Each node sends some greeting messages to all its immediate neighbours (those nodes to which it is connected directly) to collect two pieces of information: **the identity of the node** and **the cost of the link**.
- The combination of these two pieces of information is called the **LS packet (LSP)**; the LSP is sent out of each interface.



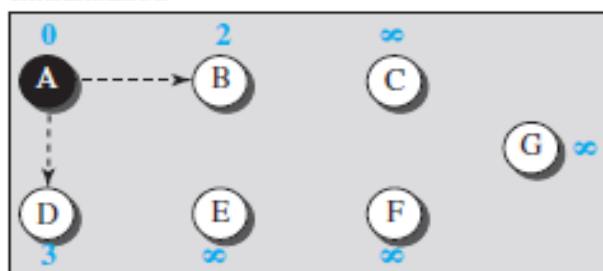
- When a node receives an LSP from one of its interfaces, it compares the LSP with the copy it may already have.
  - If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP.
  - If it is newer or the first one received, the node discards the old LSP (if there is one) and keeps the received one.
  - It then sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the network.

Sanchit Jain

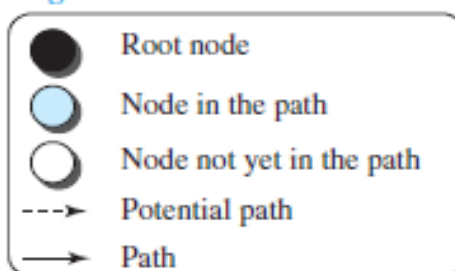
## Formation of Least-Cost Trees

- To create a least-cost tree for itself, using the shared LSDB, each node needs to run the **Dijkstra Algorithm**. This iterative algorithm uses the following steps:
  - The node chooses itself as the root of the tree, creating a tree with a single node, and sets the total cost of each node based on the information in the LSDB.
  - The node selects one node, among all nodes not in the tree, which is closest to the root, and adds this to the tree. After this node is added to the tree, the cost of all other nodes not in the tree needs to be updated because the paths may have been changed.
  - The node repeats step 2 until all nodes are added to the tree.

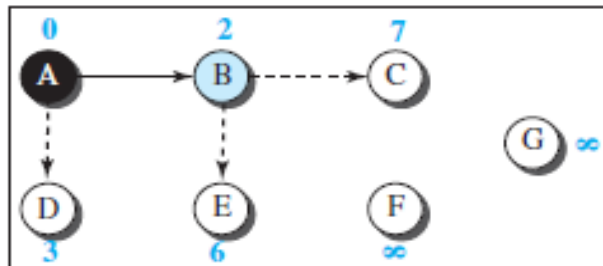
Initialization



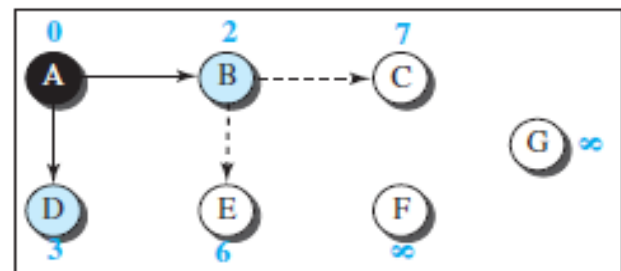
Legend



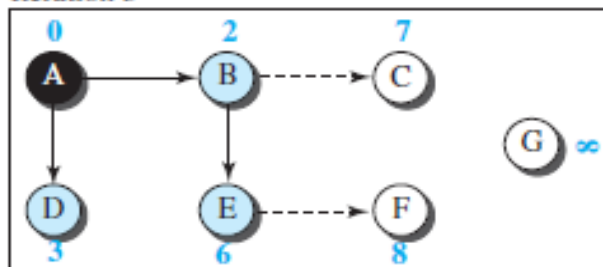
Iteration 1



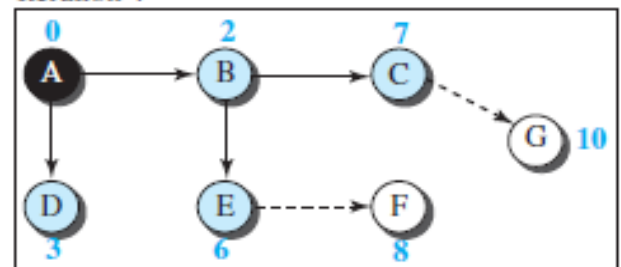
Iteration 2



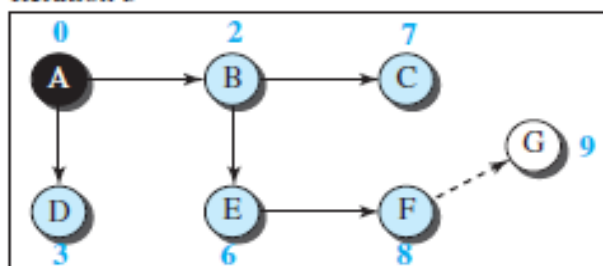
Iteration 3



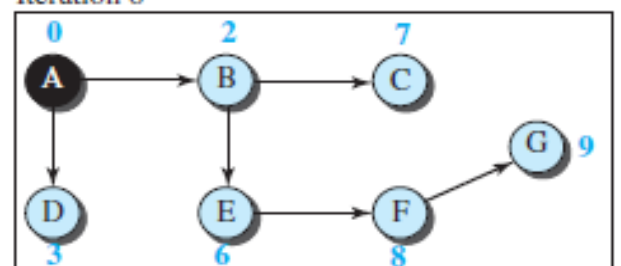
Iteration 4



Iteration 5



Iteration 6



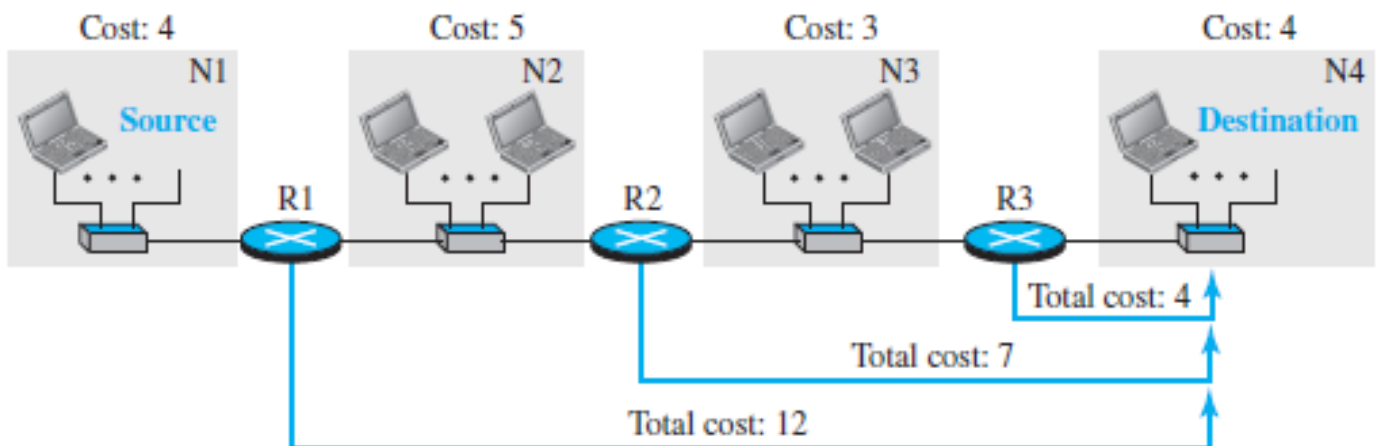
- Both link-state and distance-vector routing are based on the least-cost goal.

## Open Shortest Path First (OSPF)

- **Open Shortest Path First (OSPF)** is also an intradomain routing protocol like RIP, but it is based on the link-state routing protocol

### **Metric**

- The cost of reaching a destination from the host is calculated from the source router to the destination network.



- An interesting point about the cost in OSPF is that different service types (TOSs) can have different weights as the cost.
- Each link (network) can be assigned a weight based on the throughput, round-trip time, reliability, and so on.

### **Forwarding Table**

- Each OSPF router can create a forwarding table after finding the shortest-path tree between itself and the destination using Dijkstra's algorithm.

Forwarding table for R1

Destination network	Next router	Cost
N1	—	4
N2	—	5
N3	R2	8
N4	R2	12

Forwarding table for R2

Destination network	Next router	Cost
N1	R1	9
N2	—	5
N3	—	3
N4	R3	7

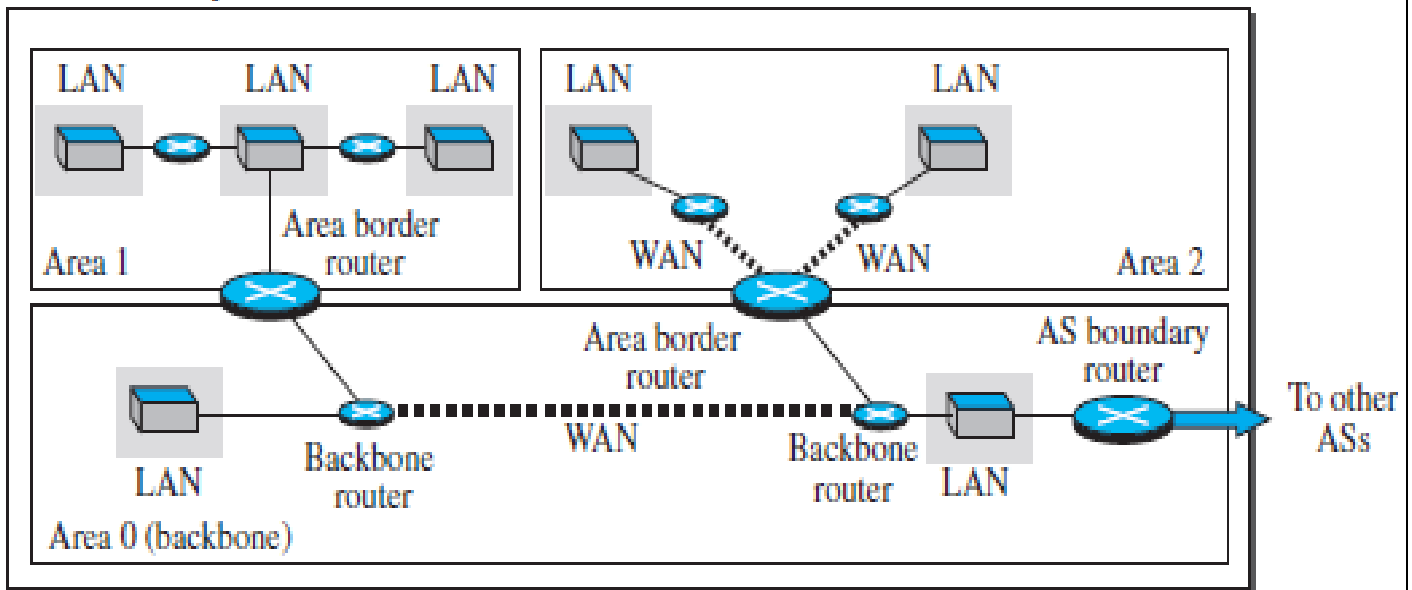
Forwarding table for R3

Destination network	Next router	Cost
N1	R2	12
N2	R2	8
N3	—	3
N4	—	4

## Areas

- RIP is normally used in small ASs, OSPF was designed to handle routing in a small or large autonomous system.
- In Large AS's, as OSPF creates a LSDB by flooding this can lead to creation of huge traffic in network, to deal with this a large AS is divided into small **areas**.
- Each area acts as a small independent domain for flooding LSPs.

### Autonomous System (AS)



- Each router in an area needs to know the information about the link states not only in its area but also in other areas.
- For this reason, one of the areas in the AS is designated as the *backbone area*, responsible for gluing the areas together.
- The routers in the backbone area are responsible for passing the information collected by each area to all other areas.

## **OSPF Implementation**

- OSPF is implemented as a program *in the network layer, using the service of the IP for propagation.*

## **Performance**

- **Update Messages.** The link-state messages in OSPF have a somewhat complex format. They also are flooded to the whole area. If the area is large, these messages may create heavy traffic and use a lot of bandwidth.
- **Convergence of Forwarding Tables.** When the flooding of LSPs is completed, each router can create its own shortest-path tree and forwarding table; convergence is fairly quick
- **Robustness.** The OSPF protocol is more robust than RIP because Corruption or failure in one router does not affect other router as seriously as in RIP.

**Q** Consider the following statements about the routing protocols, Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) in an IPv4 network.

- I.** RIP uses distance vector routing
- II.** RIP packets are sent using UDP
- III.** OSPF packets are sent using TCP
- IV.** OSPF operation is based on link-state routing

Which of the following above are CORRECT? **(Gate-2017) (1 Marks)**

- (A)** I and IV only
- (B)** I, II and III only
- (C)** I, II and IV only
- (D)** II, III and IV only

**Answer: (C)**

**Q** Which one of the following is TRUE about interior Gateway routing protocols – Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) **(GATE-2014) (1 Marks)**

- (A)** RIP uses distance vector routing and OSPF uses link state routing
- (B)** OSPF uses distance vector routing and RIP uses link state routing
- (C)** Both RIP and OSPF use link state routing
- (D)** Both RIP and OSPF use distance vector routing

**Answer: (A)**

**Q** Consider the following three statements about link state and distance vector routing protocols, for a large network with 500 network nodes and 4000 links.

[S1] The computational overhead in link state protocols is higher than in distance vector protocols.

[S2] A distance vector protocol (with split horizon) avoids persistent routing loops, but not a link state protocol.

[S3] After a topology change, a link state protocol will converge faster than a distance vector protocol.

Which one of the following is correct about S1, S2, and S3? **(GATE-2014) (1 Marks)**

**(A)** S1, S2, and S3 are all true.

**(B)** S1, S2, and S3 are all false.

**(C)** S1 and S2 are true, but S3 is false

**(D)** S1 and S3 are true, but S2 is false

**Answer: (D)**

**Q** Two popular routing algorithms are Distance Vector (DV) and Link State (LS) routing.

Which of the following are true? **(GATE-2008) (2 Marks)**

**(S1)** Count to infinity is a problem only with DV and not LS routing

**(S2)** In LS, the shortest path algorithm is run only at one node

**(S3)** In DV, the shortest path algorithm is run only at one node

**(S4)** DV requires lesser number of network messages than LS

**(A)** S1, S2 and S4 only

**(B)** S1, S3 and S4 only

**(C)** S2 and S3 only

**(D)** S1 and S4 only

**Answer: (D)**

## Routing Table

- A table is maintained by the internal router called as **Routing table**.
- It helps the internal router to decide on which interface the data packet should be forwarded.
- Routing table consists of:
  - IP Address of the destination subnet
  - Subnet mask of the subnet
  - Interface

### How routing is done in subnets

- When a data packet arrives at an internal router, the following steps are followed:
  1. Router performs the ***bitwise ANDing of Destination IP Address mentioned on the data packet and all the subnet masks one by one.***
  2. Router compares each result with their corresponding IP Address of the destination subnet in the routing table.
  3. Then, following three cases may occur:
    1. If there occurs only one match, Router forwards the data packet on the corresponding interface.
    2. If there occurs more than one match, Router forwards the data packet on the interface corresponding to the longest subnet mask.
    3. If there occurs no match, Router forwards the data packet on the interface corresponding to the default entry.

## Points to Note

- In fixed length subnetting, since all the subnets have the same subnet mask. Bitwise ANDing is performed only once.
- If the result matches to any of the destination subnet IP Address, Router forwards the data packet on its corresponding interface. Otherwise, it is forwarded on the default interface.
- In variable length subnetting, all the subnets have different subnet mask. So, bitwise ANDing is performed once with each subnet mask.
- Then, the above three cases are followed.

Example: A router uses the following routing table-

Destination	Mask	Interface
144.16.0.0	255.255.0.0	eth0
144.16.64.0	255.255.224.0	eth1
144.16.68.0	255.255.255.0	eth2
144.16.68.64	255.255.255.224	eth3

A packet bearing a destination address 144.16.68.117 arrives at the router. On which interface will it be forwarded?

- Router performs the bitwise ANDing of the Destination address mentioned on the data packet and each subnet mask one by one.

**For 1<sup>st</sup> Row:** 144.16.68.117 AND 255.255.0.0 = 144.16.0.0

Since result is same as the given destination address, so a match occurs.

**For 2<sup>nd</sup> Row:** 144.16.68.117 AND 255.255.224.0 = 144.16.64.0

Since result is same as the given destination address, so a match occurs.

**For 3<sup>rd</sup> Row:** 144.16.68.117 AND 255.255.255.0 = 144.16.68.0

Since result is same as the given destination address, so a match occurs.

**For 4<sup>th</sup> Row:** 144.16.68.117 AND 255.255.255.224 = 144.16.68.96

Since result is not same as the given destination address, so a match does not occur.

- Clearly, there occurs more than one match. So, router forwards the packet on the interface corresponding to the longest subnet mask.
- Out of all, 255.255.255.0 is the longest subnet mask since it has maximum number of 1s.
- So, the router forwards the packet to the corresponding interface eth2.

**Q Classless Inter-Domain Routing (CIDR) receives a packet with address 131.23.151.76. The router's routing table has the following entries: (Gate-2014) (2 Marks)**

Prefix	Output Interface Identifier
131.16.0.0/12	3
131.28.0.0/14	5
131.19.0.0/16	2
131.22.0.0/15	1

The identifier of the output interface on which this packet will be forwarded is \_\_\_\_\_.

**Answer: (1)**

**Q A router uses the following routing table:**

Destination	Mask	Interface
144.16.0.0	255.255.0.0	eth0
144.16.64.0	255.255.224.0	eth1
144.16.68.0	255.255.255.0	eth2
144.16.68.64	255.255.255.224	eth3

A packet bearing a destination address 144.16.68.117 arrives at the router. On which interface will it be forwarded? **(Gate-2006) (2 Marks)**

**(A) eth0**                      **(B) eth1**                      **(C) eth2**                      **(D) eth3**

**Answer: (C)**

**Q** Count to infinity is a problem associated with **(GATE-2005) (1 Marks)**

**(A)** link state routing protocol.

**(B)** distance vector routing protocol

**(C)** DNS while resolving host name.

**(D)** TCP for congestion control.

**Answer: (B)**

**Q** The routing table of a router is shown below:

Destination	Sub net mask	Interface
128.75.43.0	255.255.255.0	Eth0
128.75.43.0	255.255.255.128	Eth1
192.12.17.5	255.255.255.255	Eth3
default		Eth2

On which interfaces will the router forward packets have addressed to destinations 128.75.43.16 and 192.12.17.10 respectively? **(Gate-2004) (2 Marks)**

**(A)** Eth1 and Eth2

**(B)** Eth0 and Eth2

**(C)** Eth0 and Eth3

**(D)** Eth1 and Eth3

**Answer: (A)**

**Q** A group of 15 routers is interconnected in a centralized complete binary tree with a router at each tree node. Router  $i$  communicates with router  $j$  by sending a message to the root of the tree. The root then sends the message back down to router  $j$ . The mean number of hops per message, assuming all possible router pairs are equally likely is **(GATE-2007) (2 Marks)**

**A)** 3

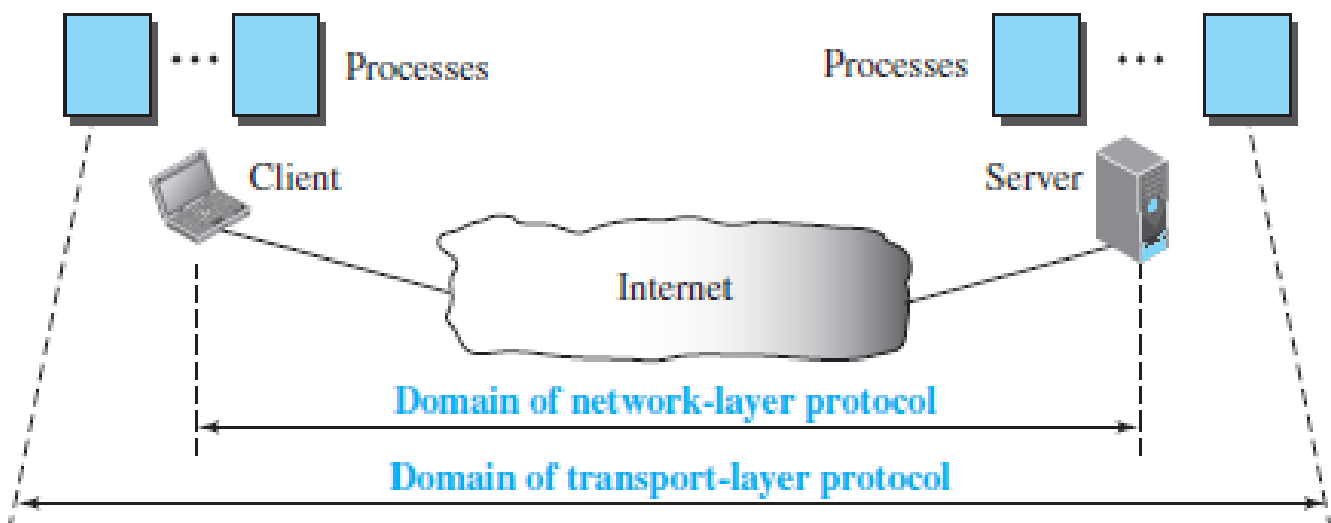
**B)** 4.26

**C)** 4.53

**D)** 5.26

## Transport-Layer Services

- **Process-to-Process Communication:** A process is an application-layer entity (running program) that uses the services of the transport layer.
- The network layer is responsible for communication at the computer level (host-to-host communication). A network-layer protocol can deliver the message only to the destination computer.
- However, this is an incomplete delivery, as the message still needs to be handed to the correct process.
- A transport-layer protocol is responsible for delivery of the message to the appropriate process. TL provides end to end or process to process communication

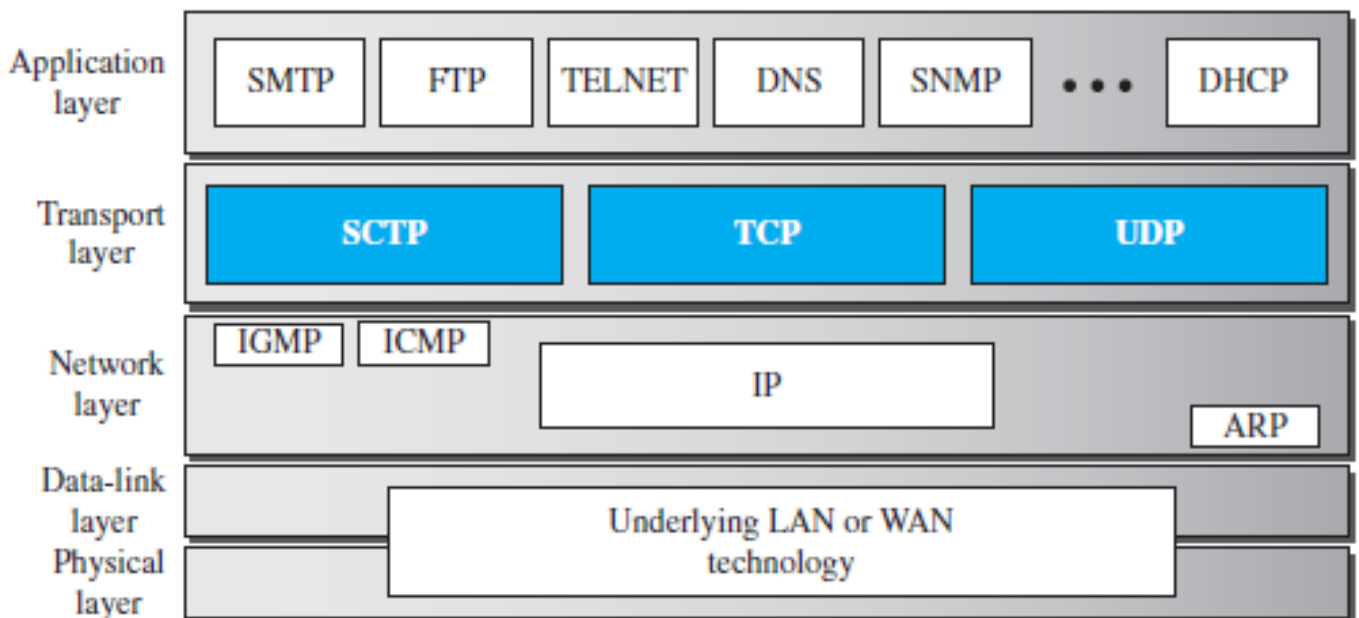


**Q** Which of the following functionalities must be implemented by a transport protocol over and above the network protocol? (Gate-2003) (1 Marks)

- (A) Recovery from packet losses
- (B) Detection of duplicate packets
- (C) Packet delivery in the correct order
- (D) End to end connectivity

**Answer: (D)**

- A transport layer protocol can be either connectionless or connection-oriented.
  - A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine.
  - A connection-oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data is transferred, the connection is terminated.
- Like the data link layer, the transport layer may be responsible for flow and error control. However, flow and error control at this layer is performed end to end rather than across a single link.
- Reliable Versus Unreliable
  - If the application layer program needs reliability, we use a reliable transport layer protocol by implementing flow and error control at the transport layer. This means a slower and more complex service.
  - On the other hand, if the application program does not need reliability because it uses its own flow and error control mechanism or it needs fast service or the nature of the service does not demand flow and error control (real-time applications), then an unreliable protocol can be used.
- In the Internet, there are three common different transport layer protocols.
  - UDP is connectionless and unreliable;
  - TCP and SCTP are connection oriented and reliable. These three can respond to the demands of the application layer programs.
- TCP offers *full-duplex service*, where data can flow in both directions at the same time.



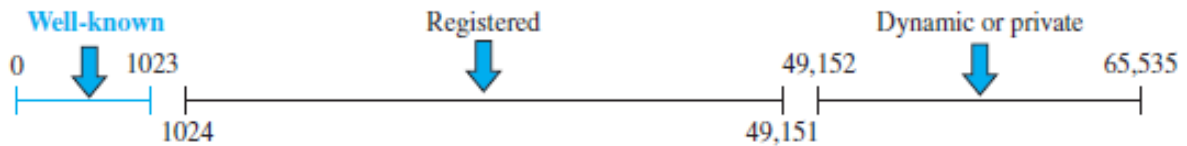
## Addressing: Port Numbers

- For communication, we must define the local host, local process, remote host, and remote process.
- Local and Remote host are defined by IP Addresses. To define the processes inside a host, we need second identifiers, called port numbers, they are 16-bits integers ranging from (0 to  $2^{16} - 1$ ) or (0 to 65535).

Sanchit Jain

## ICANN Ranges

- ICANN (Internet Corporation for Assigned Names and Numbers) has divided the port numbers into three ranges: well-known, registered, and dynamic (or private).



The IANA (Internet Assigned Number Authority) has divided the port numbers into three ranges: well known, registered, and dynamic (or private).

- **Well-known ports:**

- The ports ranging from 0 to 1023 are assigned and controlled by IANA. These are the well-known ports.
- The server process must also define itself with a port number. This port number, however, cannot be chosen randomly as the client has to request the data from server.
- TCP/IP has decided to use universal port numbers for servers; these are called **well-known port numbers.**

Port Number	Protocol	Application
20	TCP	FTP data
21	TCP	FTP control
22	TCP	SSH
23	TCP	Telnet
25	TCP	SMTP
53	UDP,TCP	DNS
67, 68	UDP	DHCP
69	UDP	TFTP
80	TCP	HTTP (WWW)
110	TCP	POP3
161	UDP	SNMP
443	TCP	HTTPS (SSL)
16384–32767	UDP	RTP-based voice (VoIP) and video

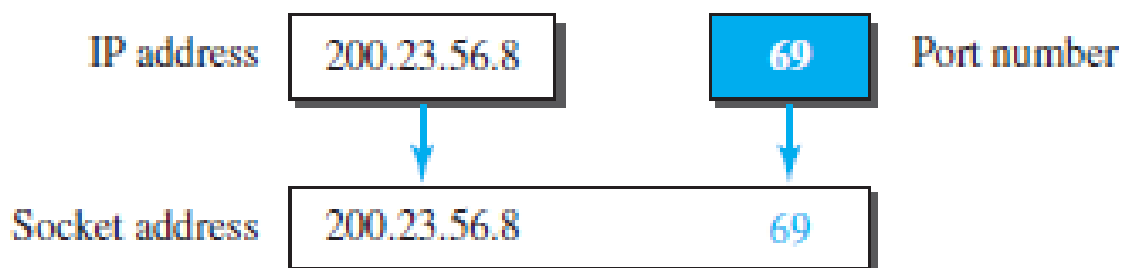
- **Registered ports:**
  - The ports ranging from 1024 to 49,151 are not assigned or controlled by IANA. They can only be registered with IANA to prevent duplication.
- **Dynamic/Ephemeral ports:**
  - The ports ranging from 49,152 to 65,535 are neither controlled nor registered. They can be used by any process.
  - The client program defines itself with a port number, chosen randomly by the transport layer software running on the client host.
  - Ephemeral means “short-lived” and is used because the life of a client is normally short.

## Socket Addresses

- A transport-layer protocol in the TCP suite needs both the IP address and the port number, at each end, to make a connection. To use the services of the transport layer in the Internet, we need a pair of socket addresses: the client socket address and the server socket address.
- The combination of an IP address and a port number is called a **socket address**.

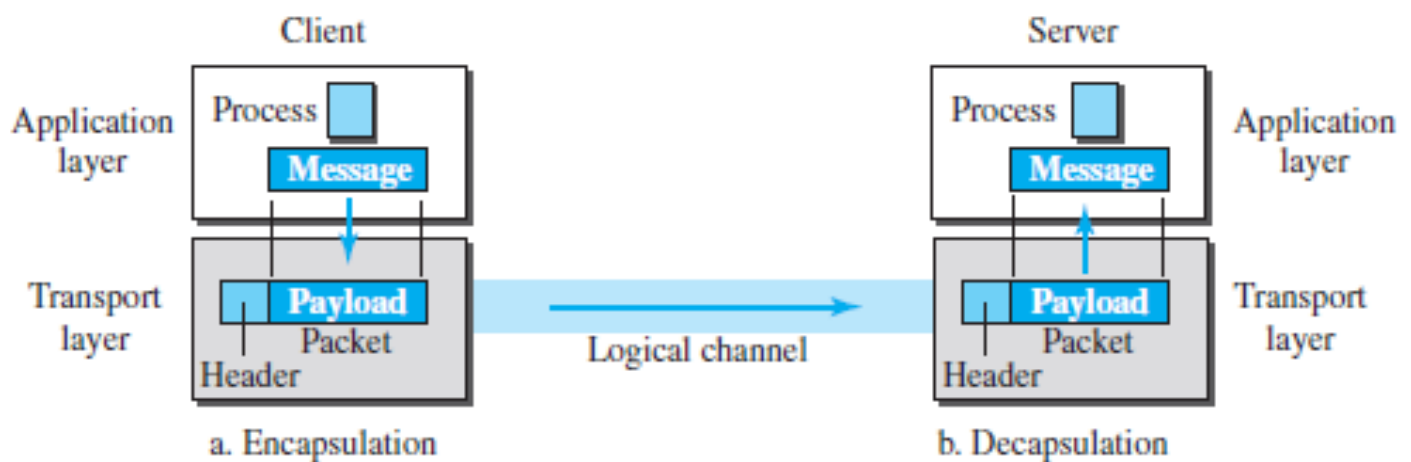
## Socket Addresses

- A transport-layer protocol in the TCP suite needs both the IP address and the port number, at each end, to make a connection. To use the services of the transport layer in the Internet, we need a pair of socket addresses: the client socket address and the server socket address.
- The combination of an IP address and a port number is called a **socket address**.



## Encapsulation and Decapsulation

- To send a message from one process to another, the transport-layer protocol encapsulates and decapsulates messages.
- Encapsulation happens at the sender site. When a process has a message to send, it passes the message to the transport layer along with a pair of socket addresses.
- The transport layer receives the data and adds the transport-layer header. The packets at the transport layer in the Internet are called *user datagrams, segments, or packets*.



- Decapsulation happens at the receiver site.
- When the message arrives at the destination transport layer, the header is dropped and the transport layer delivers the message to the process running at the application layer.

**Q** What is the maximum size of data that the application layer can pass on to the TCP layer below? (Gate-2008) (1 Marks)

- (A) Any size  
(C)  $2^{16}$  bytes

- (B)  $2^{16}$  bytes – size of TCP header  
(D) 1500 bytes

**Answer: (A)**

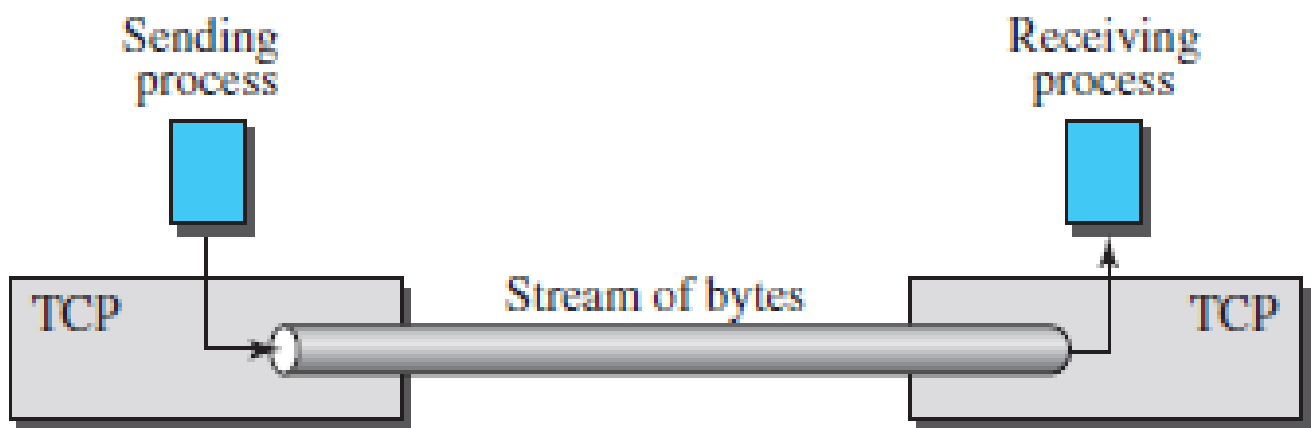
**Q** A TCP message consisting of 2100 bytes is passed to IP for delivery across two networks. The first network can carry a maximum payload of 1200 bytes per frame and the second network can carry a maximum payload of 400 bytes per frame, excluding network overhead. Assume that IP overhead per packet is 20 bytes. What is the total IP overhead in the second network for this transmission? (Gate-2004) (2 Marks)

- (A) 40 bytes                      (B) 80 bytes                      (C) 120 bytes                      (D) 160 bytes

**Answer: (C)**

## TCP (Transmission Control Protocol)

- TCP is a **reliable connection-oriented protocol**, it must be used in any application where **reliability is important**.
- It creates a virtual connection between two TCPs to send data. In addition, TCP uses flow and error control mechanisms at the transport level.
- TCP allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.
- TCP creates an environment in which the two processes seem to be connected by an imaginary "tube" that carries their data across the Internet.
- It adds connection-oriented and reliability features to the services of IP.



**Q** In TCP, a unique sequence number is assigned to each **(Gate-2004) (1 Marks)**

**(A)** byte

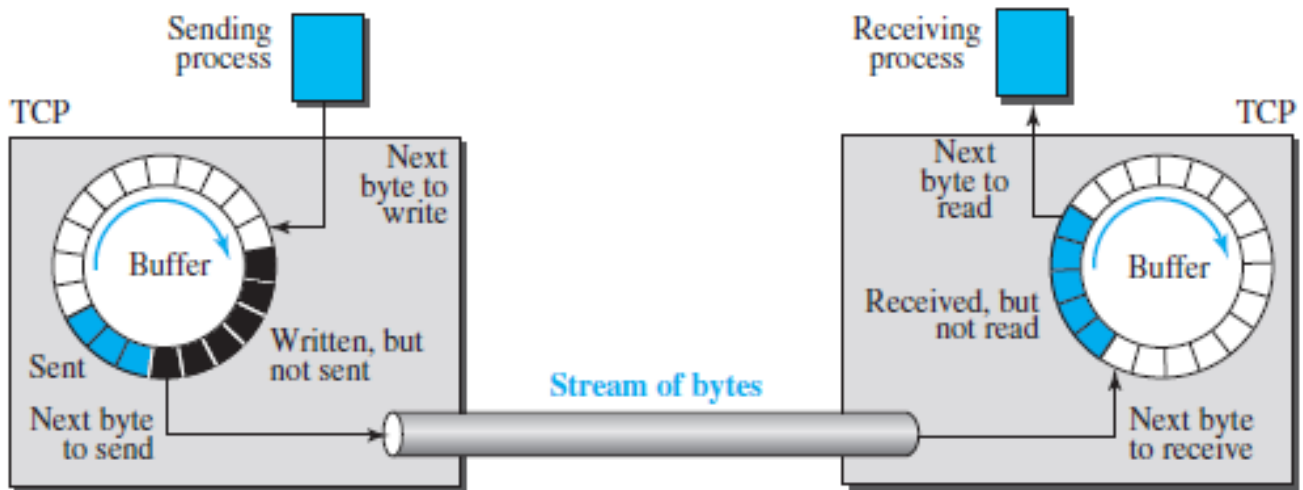
**(B)** word

**(C)** segment

**(D)** message

**Answer: (A)**

- Connection oriented means some resources will be reserved at the receiver end, like Bandwidth, CPU time, Buffer etc.
- Sending and Receiving Buffers Because the sending and the receiving processes may not write or read data at the same speed, TCP needs buffers for storage.



- **Flow Control**

- The receiver of the data controls the amount of data that are to be sent by the sender. This is done to prevent the receiver from being overwhelmed with data. The numbering system allows TCP to use a byte-oriented flow control.

- **Error Control**

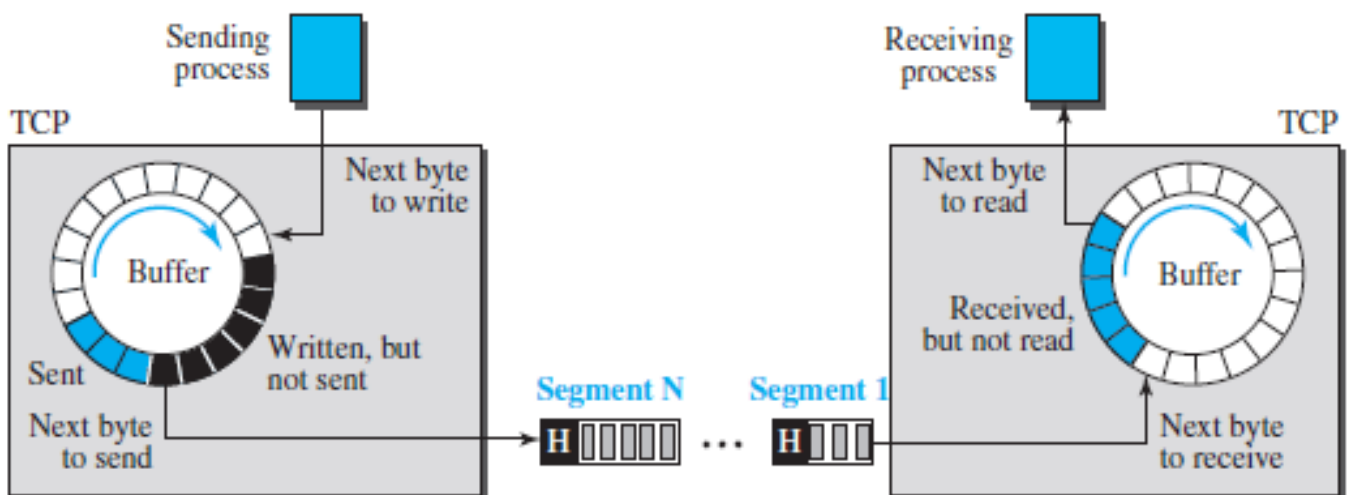
- To provide reliable service, TCP implements an error control mechanism. Although error control considers a segment as the unit of data for error detection (loss or corrupted segments), error control is byte-oriented.

- **Congestion Control**

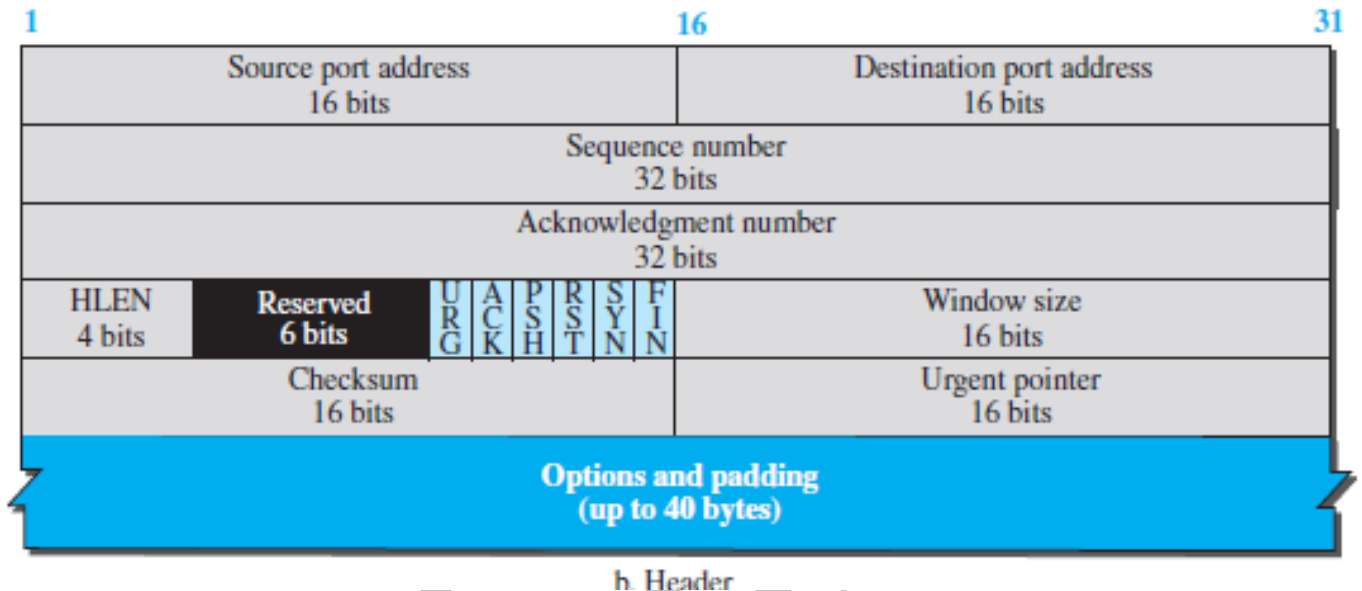
- TCP, unlike UDP, takes into account congestion in the network. The amount of data sent by a sender is not only controlled by the receiver (flow control), but is also determined by the level of congestion in the network.

## Segments

- The network layer, as a service provider for TCP, needs to send data in packets, not as a stream of bytes. At the transport layer, TCP groups a number of bytes together into a packet called a *segment*.
- TCP adds a header to each segment (for control purposes) and delivers the segment to the network layer for transmission. The segments are encapsulated in an IP datagram and transmitted.



## TCP Header



- The segment consists of a 20- to 60-byte header, followed by data from the application program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options.

**Q** In the TCP/IP protocol suite, which one of the following is NOT part of the IP header?  
**(Gate-2004) (2 Marks)**

**(A)** Fragment Offset

**(B)** Source IP address

**(C)** Destination IP address

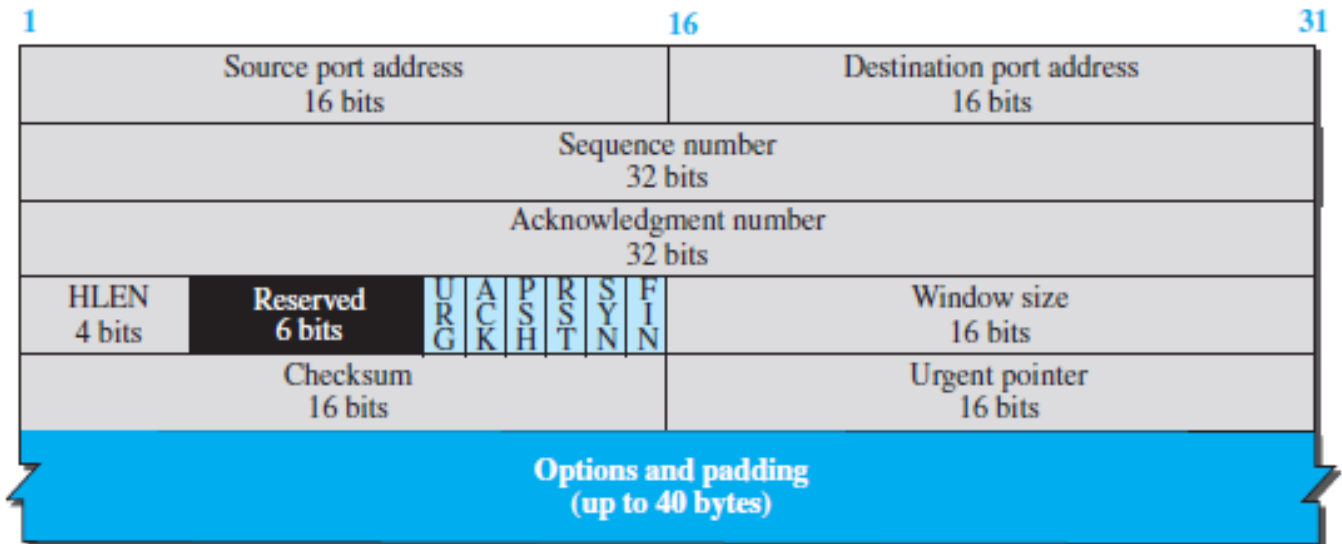
**(D)** Destination port number

**Answer: (D)**

## Source & Destination port address



a. Segment



b. Header

- **Source port address.** This is a 16-bit field that defines the port number of the application program in the host that is sending the segment.
- **Destination port address.** This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment.

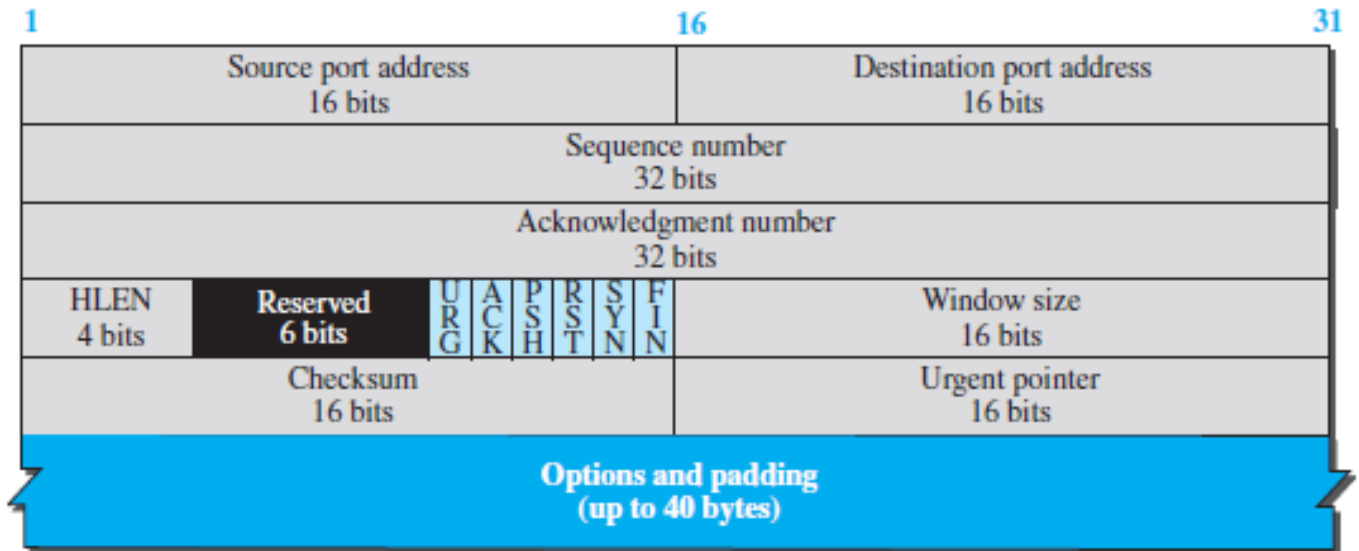
## Byte Number

- TCP numbers all data bytes (octets) that are transmitted in a connection.
- Numbering is independent in each direction.
- When TCP receives bytes of data from a process, TCP stores them in the sending buffer and numbers them.
- The numbering does not necessarily start from 0.
- TCP chooses an arbitrary number between 0 and  $2^{32} - 1$  for the number of the first byte.

## Sequence number



a. Segment



h. Header

- TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. Sequence number is 32-bit field defines the number assigned to the first byte of data contained in this segment.
- So, maximum number of possible sequence numbers =  $2^{32}$ . These sequence numbers lie in the range  $[0, 2^{32} - 1]$ .
- In IP every packet is counted not Byte, in DLL every bit is counted with HDLC protocol.
- During connection establishment, each party uses a random number generator to create an initial sequence number (ISN), which is usually different in each direction. Sequence number should be started at random, to remove duplication problem.
- The sequence number of any other segment is the sequence number of the previous segment plus the number of bytes (real or imaginary) carried by the previous segment.

**Example:** Suppose a TCP connection is transferring a file of 5000 bytes. The first byte is numbered 10001. What are the sequence numbers for each segment if data are sent in five segments, each carrying 1000 bytes?

Segment 1	→	Sequence Number:	10001	Range:	10001	to	11000
Segment 2	→	Sequence Number:	11001	Range:	11001	to	12000
Segment 3	→	Sequence Number:	12001	Range:	12001	to	13000
Segment 4	→	Sequence Number:	13001	Range:	13001	to	14000
Segment 5	→	Sequence Number:	14001	Range:	14001	to	15000

- This does not imply that only  $2^{32}$  bytes = 4 GB data can be sent using TCP. The concept of wrap around allows to send unlimited data using TCP.
- After all the  $2^{32}$  sequence numbers are used up and more data is to be sent, the sequence numbers can be wrapped around and used again from the starting.

### Wrap Around Time

- Time taken to use up all the  $2^{32}$  sequence numbers is called as **wrap around time**.
- It depends on the bandwidth of the network i.e. the rate at which the bytes go out.

$$\text{Wrap Around Time} \propto 1 / \text{Bandwidth}$$

If bandwidth of the network = x bytes/sec, then

$$\text{Wrap Around Time} = 2^{32} / x \text{ sec.}$$

### Life Time of TCP Segment

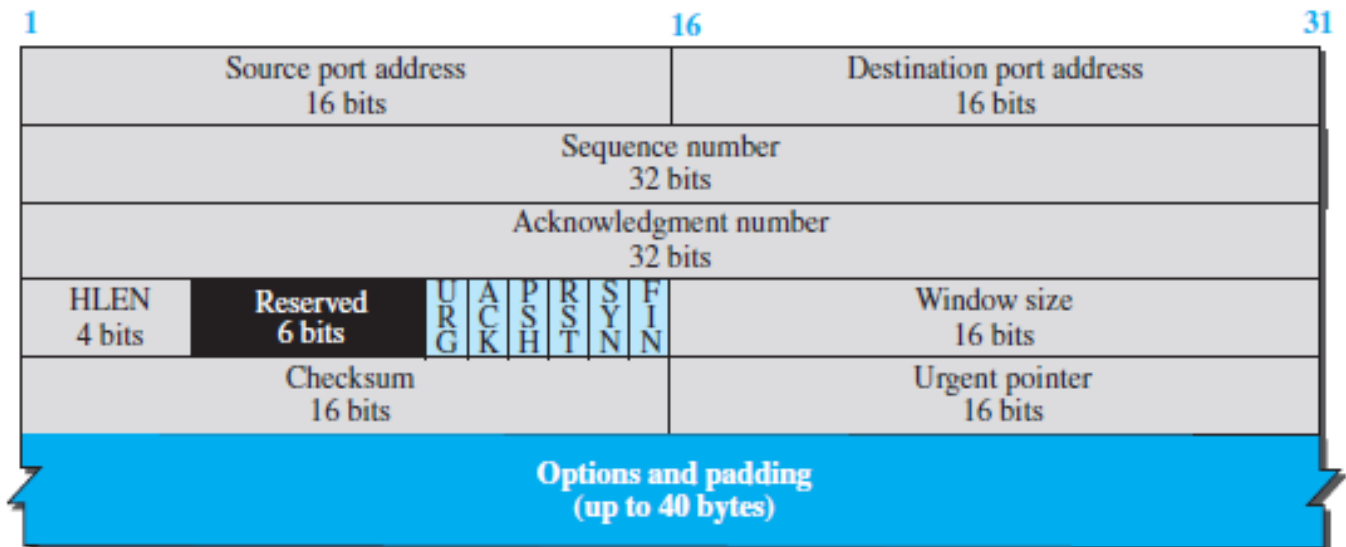
- Life time of a TCP segment is 180 seconds or 3 minutes.
- It means after sending a TCP segment, it might reach the receiver taking 3 minutes in the worst case.
- In the last we will do rap around, wrap around time is the time taken to wrap around
  - if  $\text{WAT} > \text{LT}$  then there is no problem
  - if  $\text{wat} < \text{LT}$  then destination will get same sequence no again and again, to solve this problem additional bits can be put in options, called time-stamp from the time of the clock least significant 32 bits are taken

**Q** Consider a long-lived TCP session with an end-to-end bandwidth of 1 Gbps (=  $10^9$  bits/second). The session starts with a sequence number of 1234. The minimum time (in seconds, rounded to the closest integer) before this sequence number can be used again is \_\_\_\_\_. **(GATE-2018) (1 Marks)**

# Acknowledgment Number



a. Segment



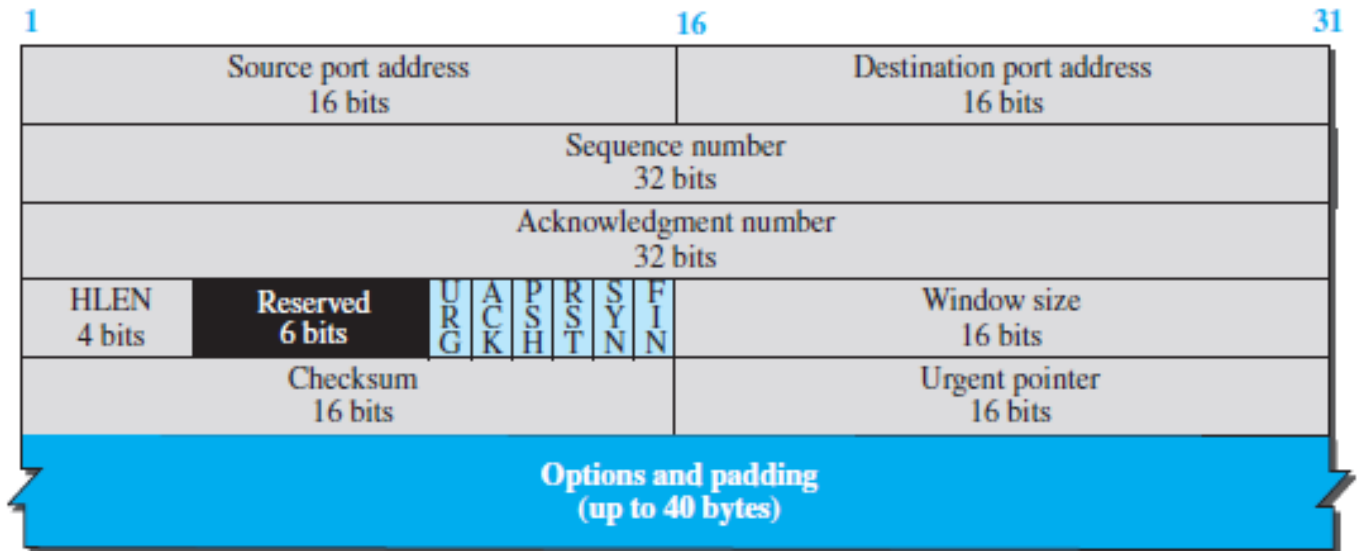
h. Header

- This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number  $x$  from the other party, it defines  $x + 1$  as the acknowledgment number. Acknowledgment and data can be piggybacked together.
- The acknowledgment number is cumulative, which means that the party takes the number of the last byte that it has received, safe and sound, adds 1 to it, and announces this sum as the acknowledgment number.
- Acknowledgment number no can be calculated by subtracting header length of IP and TCP to get the total byte count of the TCP segment and then can find the ack no

## Header length



a. Segment



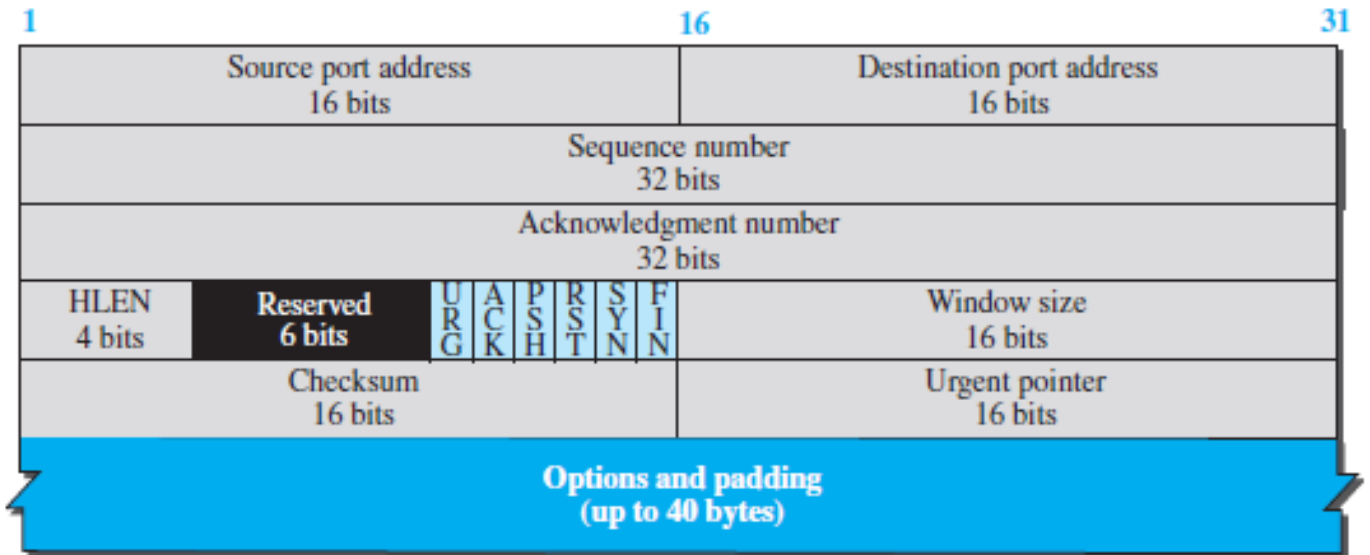
h. Header

- Header length: This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field can be between 5 ( $5 \times 4 = 20$ ) and 15 ( $15 \times 4 = 60$ ).
- Concept of Scaling Factor
  - $Header\ length = Header\ length\ field\ value \times 4\ bytes$
- Reserved. This is a 6-bit field reserved for future use.

# Checksum

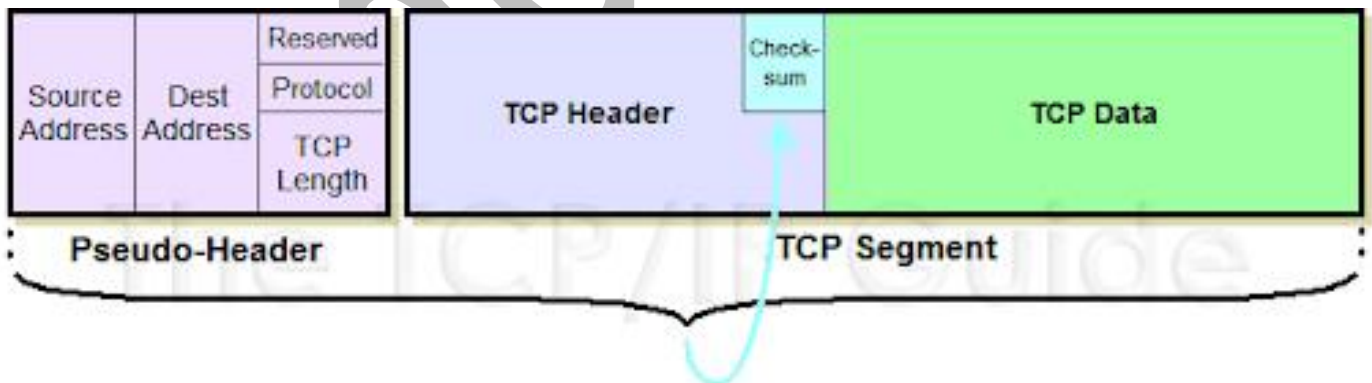


a. Segment



b. Header

- **Checksum:** This 16-bit field contains the checksum.
- While calculation of the checksum for TCP, Entire TCP segment and pseudo header (IP) is considered.
- For the TCP pseudo header, the value for the protocol field is 6.

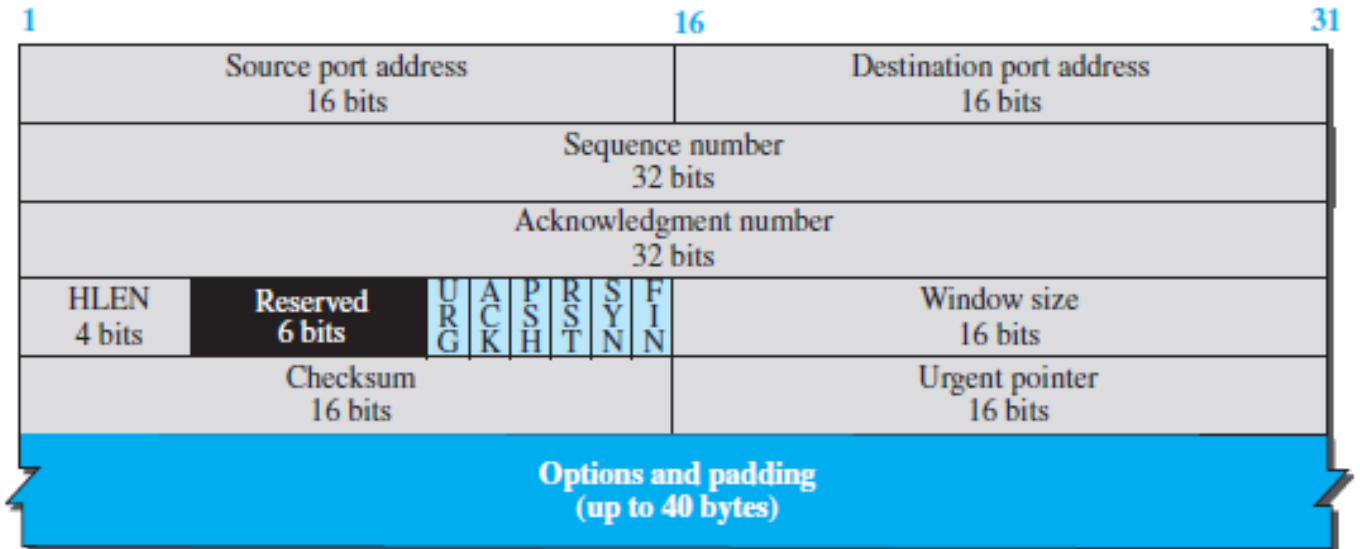


**Checksum Calculated Over Pseudo Header and TCP Segment**

## Window Size



a. Segment



b. Header

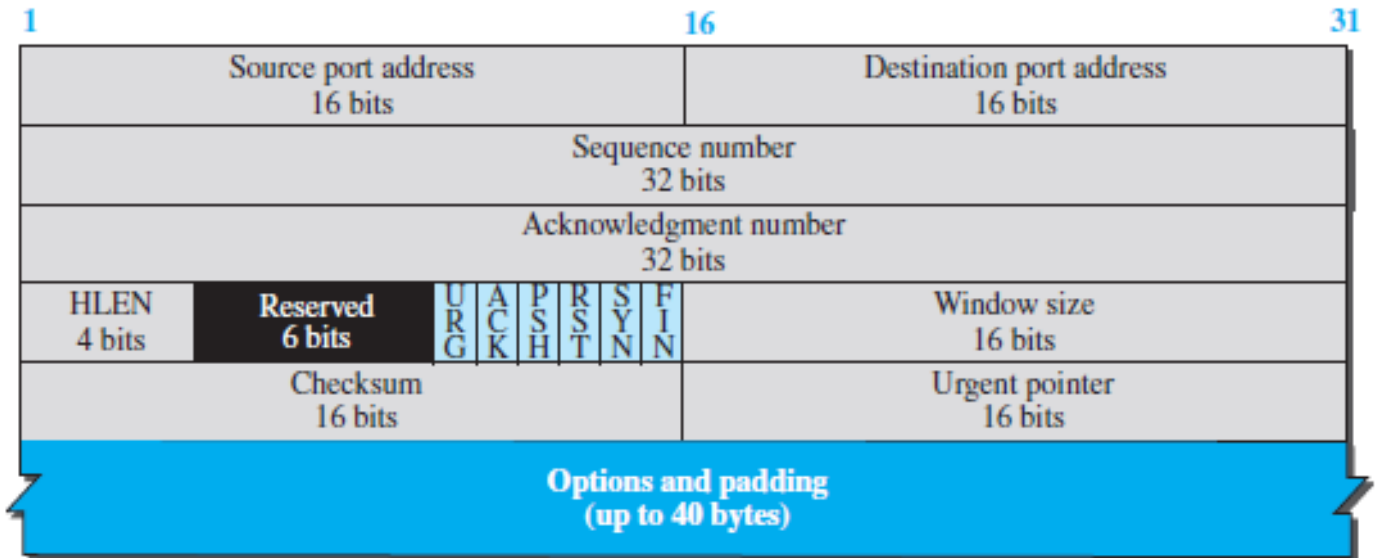
- **Window size:** Window size. This field defines the size of the window, in bytes, that the other party must maintain.
- Note that the length of this field is 16 bits, which means that the maximum size of the window is 65,535 bytes. This value is normally referred to as the receiving window and is determined by the receiver. The sender must obey the dictation of the receiver in this case.

Sample

## Urgent pointer



a. Segment



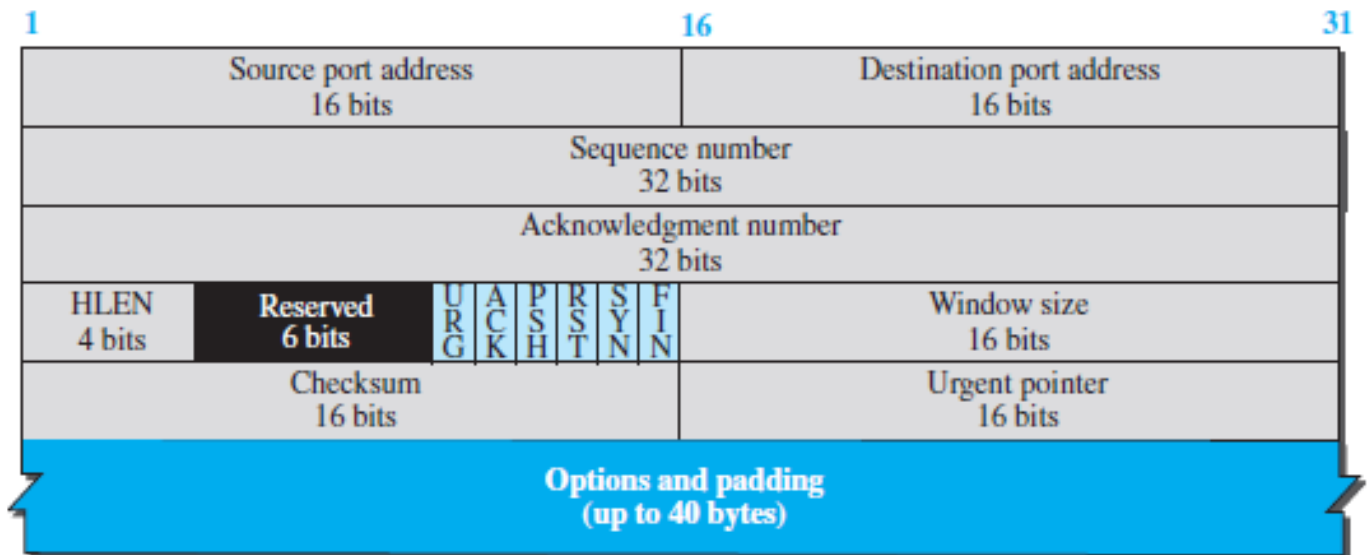
h. Header

- Urgent pointer:** This 16-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data. It defines the number that must be added to the sequence number to obtain the number of the last urgent byte in the data section of the segment.

## Control Flag



a. Segment



b. Header

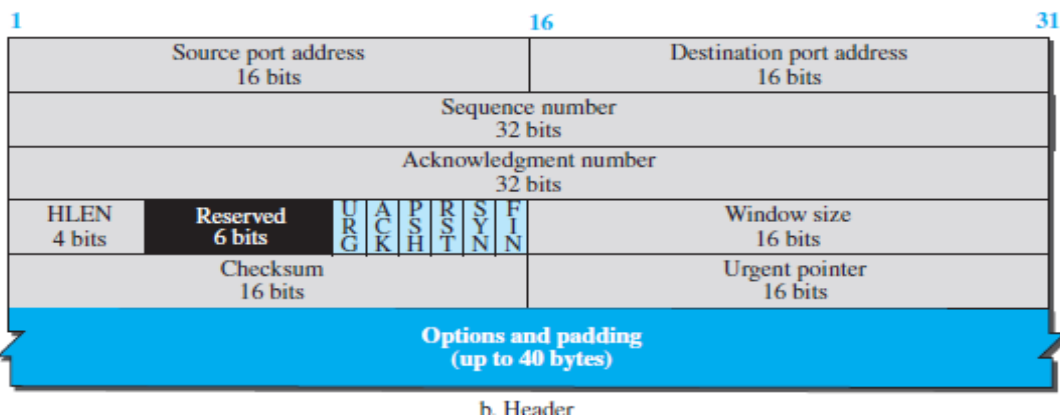
- **Control Flag:** This field defines 6 different control bits or flags. One or more of these bits can be set at a time.
- These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP.

<i>Flag</i>	<i>Description</i>
URG	The value of the urgent pointer field is valid.
ACK	The value of the acknowledgment field is valid.
PSH	Push the data.
RST	Reset the connection.
SYN	Synchronize sequence numbers during connection.
FIN	Terminate the connection.

## PUSH Flag

- **Push (PSH)** – Transport layer by default waits for some time for application layer to send enough data equal to maximum segment size so that the number of packets transmitted on network minimizes which is not desirable by some application like interactive applications(chatting).
- Similarly transport layer at receiver end buffers packets and transmit to application layer if it meets certain criteria. This problem is solved by using PSH. Transport layer sets PSH = 1 and immediately sends the segment to network layer as soon as it receives signal from application layer.
- Receiver transport layer, on seeing PSH = 1 immediately forwards the data to application layer. In general, it tells the receiver to process these packets as they are received instead of buffering them.

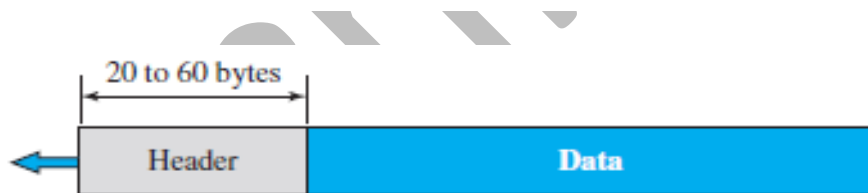
<i>Flag</i>	<i>Description</i>
URG	The value of the urgent pointer field is valid.
ACK	The value of the acknowledgment field is valid.
PSH	Push the data.
RST	Reset the connection.
SYN	Synchronize sequence numbers during connection.
FIN	Terminate the connection.



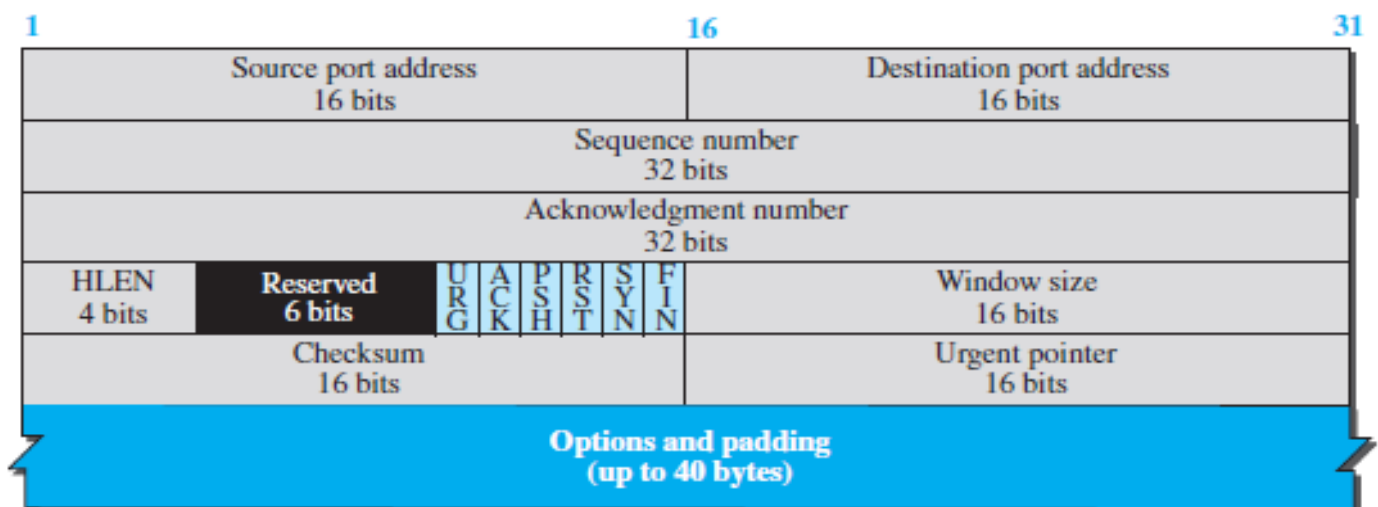
## RST Flag

- **Reset (RST)** – It is used to terminate the connection if the sender or receiver feels something is wrong with the TCP connection or that the conversation should not exist.
- It can get send from receiver side when packet is sent to particular host that was not expecting it

<i>Flag</i>	<i>Description</i>
URG	The value of the urgent pointer field is valid.
ACK	The value of the acknowledgment field is valid.
PSH	Push the data.
RST	Reset the connection.
SYN	Synchronize sequence numbers during connection.
FIN	Terminate the connection.



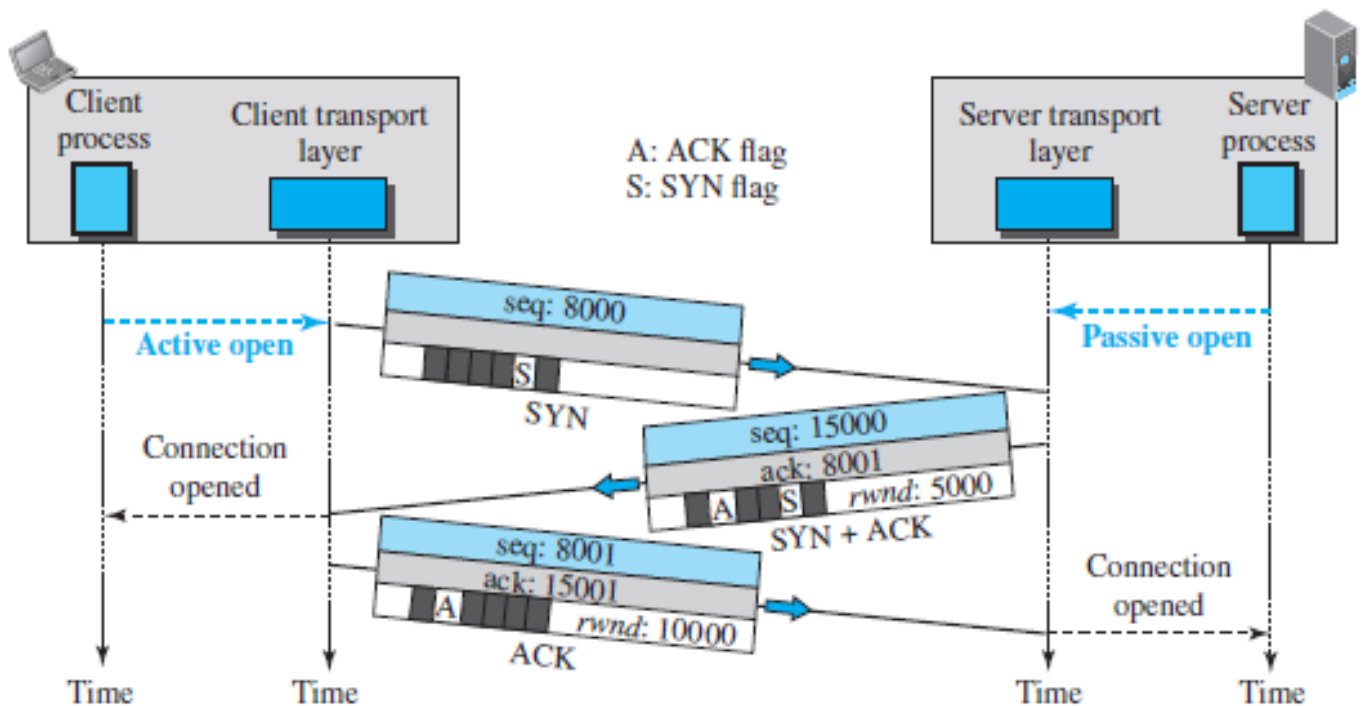
a. Segment



b. Header

## A TCP Connection

- The connection establishment in TCP is called *three-way handshaking*.



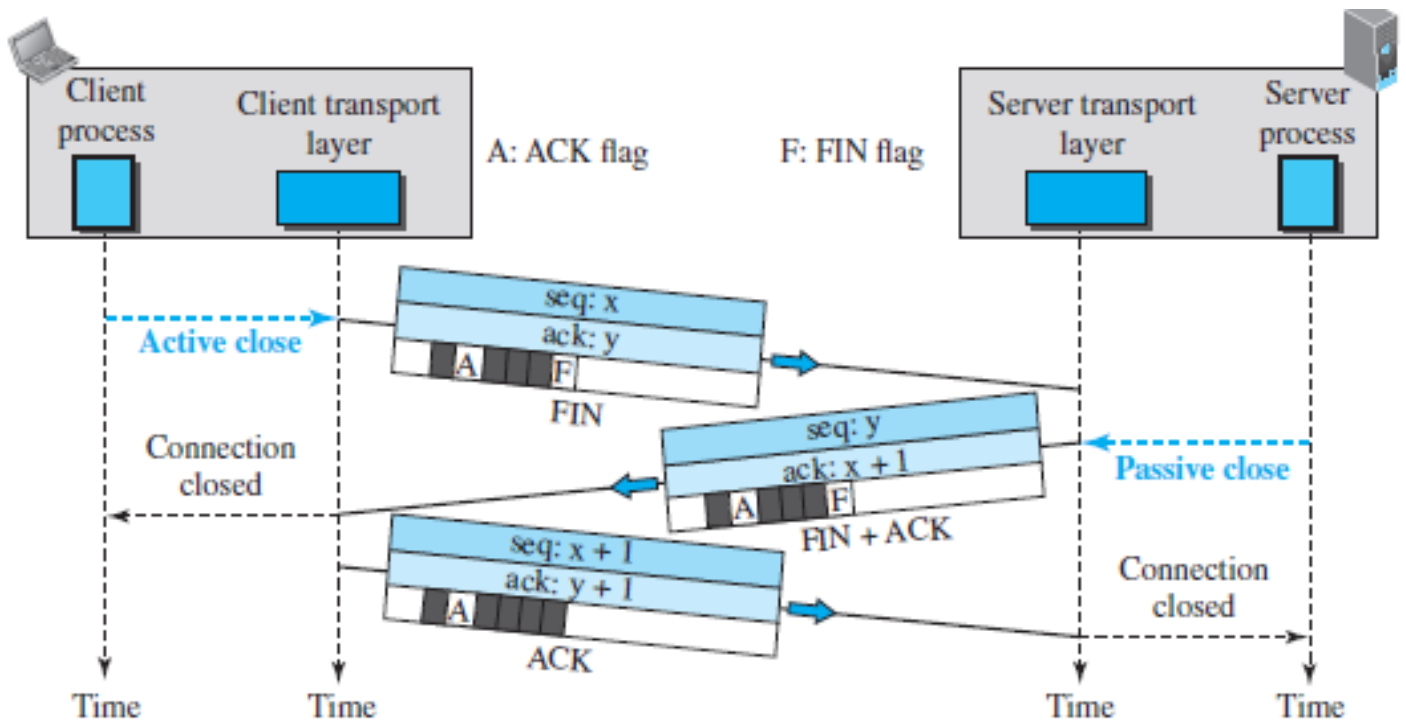
- In the above figure, an application program, called the *client*, wants to make a connection with another application program, called the *server*, using TCP as the transport-layer protocol.
- The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This request is called a *passive open*.
- Although the server TCP is ready to accept a connection from any machine in the world, it cannot make the connection itself.
- The client program issues a request for an *active open*. A client that wishes to connect to an open server tells its TCP to connect to a particular server. TCP can now start the three-way handshaking process.

## Connection Establishment (Three-way handshaking)

- The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization of sequence numbers.
  - Client sends the **initial sequence number (ISN)**.
  - This segment does not contain an acknowledgment number
  - SYN segment is a control segment and carries no data. However, it consumes one sequence number because it needs to be acknowledged.
- The server sends the second segment, a SYN + ACK segment with two flag bits set as: SYN and ACK.
  - It is a SYN segment for communication in the other direction.
  - The server uses this segment to initialize a sequence number for numbering the bytes sent from the server to the client.
  - The server also acknowledges the receipt of the SYN segment from the client by setting the ACK flag and displaying the next sequence number it expects to receive from the client.
  - A SYN + ACK segment cannot carry data, but it does consume one sequence number.
- The client sends the third segment.
  - This is just an ACK segment.
  - It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field.
- ACK segment does not consume any sequence numbers if it does not carry data.
- After connection is established, bidirectional data transfer can take place.

## Connection Termination (Three-way handshaking)

- Either of the two parties involved in exchanging data (client or server) can close the connection



- In this situation, the client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set.
  - The FIN segment consumes one sequence number if it does not carry data.
- The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN + ACK segment, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction.
  - If it does not carry data, it consumes only one sequence number because it needs to be acknowledged.
- The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server.
  - This segment cannot carry data and consumes no sequence numbers.

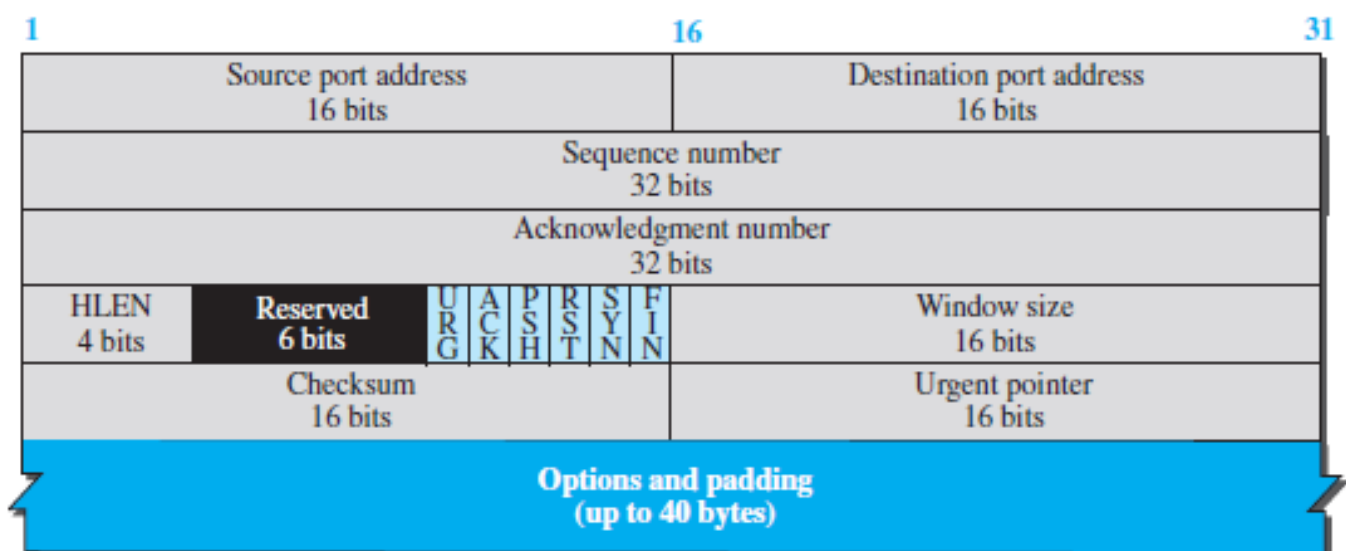
## States for TCP

<i>State</i>	<i>Description</i>
<b>CLOSED</b>	No connection exists
<b>LISTEN</b>	Passive open received; waiting for SYN
<b>SYN-SENT</b>	SYN sent; waiting for ACK
<b>SYN-RCVD</b>	SYN + ACK sent; waiting for ACK
<b>ESTABLISHED</b>	Connection established; data transfer in progress
<b>FIN-WAIT-1</b>	First FIN sent; waiting for ACK
<b>FIN-WAIT-2</b>	ACK to first FIN received; waiting for second FIN
<b>CLOSE-WAIT</b>	First FIN received, ACK sent; waiting for application to close
<b>TIME-WAIT</b>	Second FIN received, ACK sent; waiting for 2MSL time-out
<b>LAST-ACK</b>	Second FIN sent; waiting for ACK
<b>CLOSING</b>	Both sides decided to close simultaneously

## Options



a. Segment



h. Header

- There can be up to 40 bytes of optional information in the TCP header.

**Q** Suppose two hosts use a TCP connection to transfer a large file. Which of the following statements is/are **False** with respect to the TCP connection? **(Gate-2015) (1 Marks)**

1. If the sequence number of a segment is  $m$ , then the sequence number of the subsequent segment is always  $m+1$ .
2. If the estimated round-trip time at any given point of time is  $t$  sec, the value of the retransmission timeout is always set to greater than or equal to  $t$  sec.
3. The size of the advertised window never changes during the course of the TCP connection.
4. The number of unacknowledged bytes at the sender is always less than or equal to the advertised window

**(A)** 3 only      **(B)** 1 and 3 only      **(C)** 1 and 4 only      **(D)** 2 and 4 only

**Answer: (B)**

**Q** While opening a TCP connection, the initial sequence number is to be derived using a time-of-day(ToD) clock that keeps running even when the host is down. The low order 32 bits of the counter of the ToD clock is to be used for the initial sequence numbers. The clock counter increments once per millisecond. The maximum packet lifetime is given to be 64s. Which one of the choices given below is closest to the minimum permissible rate at which sequence numbers used for packets of a connection can increase? **(Gate-2009) (2 Marks)**

**(A)** 0.015/s      **(B)** 0.064/s      **(C)** 0.135/s      **(D)** 0.327/s

**Answer: (A)**

## SYN Flooding Attack

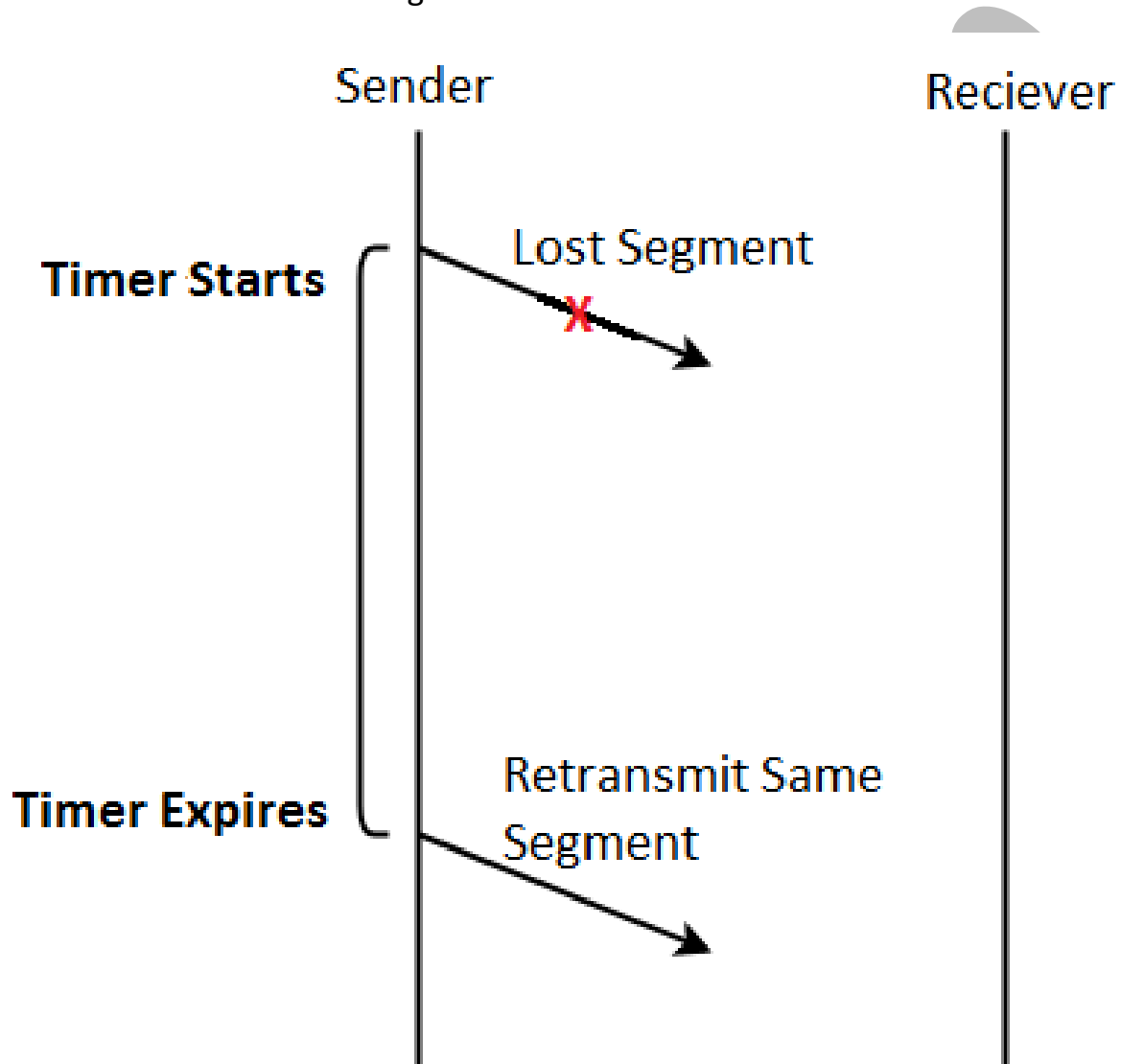
- When one or more malicious attackers send a large number of SYN segments to a server pretending that each of them is coming from a different client by faking the source IP addresses in the datagrams.
- The server allocates the necessary resources, such as creating transfer control block (TCB) tables and setting timers.
- TCP server then sends the SYN + ACK segments to the fake clients, which are lost.
- The server waits for the third leg of the handshaking process and resources are allocated without being used.
- During this short period of time, if the number of SYN segments is large, the server eventually runs out of resources and may be unable to accept connection requests from valid clients.
- SYN flooding attack belongs to denial of service attack.

## TCP Retransmission

- After establishing the connection, Sender starts transmitting TCP segments to the receiver. A TCP segment sent by the sender may get lost on the way before reaching the receiver.
- This causes the receiver to send the acknowledgement with same ACK number to the sender. As a result, sender retransmits the same segment to the receiver. This is called as **TCP retransmission**.
- Sender discovers that the TCP segment is lost when
  - Either Time Out Timer expire or it receives three duplicate acknowledgements

## Retransmission after Time out Timer Expiry

- Each time sender transmits a TCP segment to the receiver, it starts a Time Out Timer. Following two cases are possible
  - Sender receives an acknowledgement for the sent segment before the timer goes off. In this case, sender stops the timer.
  - Sender does not receive any acknowledgement for the sent segment and the timer goes off. In this case, sender assumes that the sent segment is lost. Sender retransmits the same segment to the receiver and resets the timer.



## Retransmission After Receiving 3 Duplicate Acknowledgements/ Early Retransmission

- Consider sender receives three duplicate acknowledgements for a TCP segment sent by it. Then, sender assumes that the corresponding segment is lost.
- So, sender retransmits the same segment without waiting for its time out timer to expire. This is known as **early retransmission** or Fast retransmission.

### Example:

Consider Sender sends 5 TCP segments to the receiver. The second TCP segment gets lost before reaching the receiver. The sequence of steps that will take place are

- On receiving segment-1, receiver sends acknowledgement asking for segment-2 next. (Original ACK)
- On receiving segment-3, receiver sends acknowledgement asking for segment-2 next. (1st duplicate ACK)
- On receiving segment-4, receiver sends acknowledgement asking for segment-2 next. (2nd duplicate ACK)
- On receiving segment-5, receiver sends acknowledgement asking for segment-2 next. (3rd duplicate ACK)
- Now, Sender receives 3 duplicate acknowledgements for segment-2 in total. So, sender assumes that the segment-2 is lost. So, it retransmits segment-2 without waiting for its timer to go off.

### Points to Note

- In case time out timer expires before receiving the acknowledgement for a TCP segment, then there is a strong possibility of congestion in the network.
- Retransmission on receiving 3 duplicate acknowledgements is a way to improve the performance over retransmission on time out.
- TCP uses SR (80%) and GBN (20%) both, as  $W_s = W_r$  (SR) out of order packets will be accepted and in GBN use cumulative acknowledgement
- Question is why only 3 duplicate ack, experimentally it is found out that this works best.

**Q** Consider a TCP connection in a state where there are no outstanding ACKs. The sender sends two segments back to back. The sequence numbers of the first and second segments are 230 and 290 respectively. The first segment was lost but the second segment was received correctly by the receiver.

Let X be the amount of data carried in the first segment (in bytes) and Y be the ACK number sent by the receiver. The values of X and Y are

Ans. Sequence number of 1st segment = 230 and Sequence number of 2nd segment = 290

Range of sequence numbers contained in the 1st segment = [230,289].

Total number of sequence numbers contained in the 1st segment =  $289 - 230 + 1 = 60$ .

TCP assigns 1 sequence number to 1 byte of data, so total data is **60 Bytes**.

On receiving the 2nd segment, Receiver sends the acknowledgement asking for the first segment only. This is because it expects the 1st segment first. Thus, **Acknowledgement number = Sequence number of the 1st segment = 230**.

**Q** Consider a TCP client and a TCP server running on two different machines. After completing data transfer, the TCP client calls *close* to terminate the connection and a FIN segment is sent to the TCP server. Server-side TCP responds by sending an ACK, which is received by the client-side TCP. As per the TCP connection state diagram (RFC 793), in which state does the client-side TCP connection wait for the FIN from the server-side TCP?

**(GATE-2017) (1 Marks)**

(a) LAST-ACK

(b) TIME-WAIT

(c) FIN-WAIT-1

(d) FIN-WAIT-2

Ans: d

**Q** Assume that the bandwidth for a TCP connection is 10,48,560 bits/sec. Let  $\alpha$  be the value of RTT in milliseconds (rounded off to the nearest integer) after which the TCP window scale option is needed. Let  $\beta$  be the maximum possible window size with window scale option. Then the values of  $\alpha$  and  $\beta$  are. **(Gate-2015) (2 Marks)**

(A) 63 milliseconds  $65535 \times 2^{14}$

(B) 63 milliseconds  $65535 \times 2^{16}$

(C) 500 milliseconds  $65535 \times 2^{14}$

(D) 500 milliseconds  $65535 \times 2^{16}$

Answer: (C)

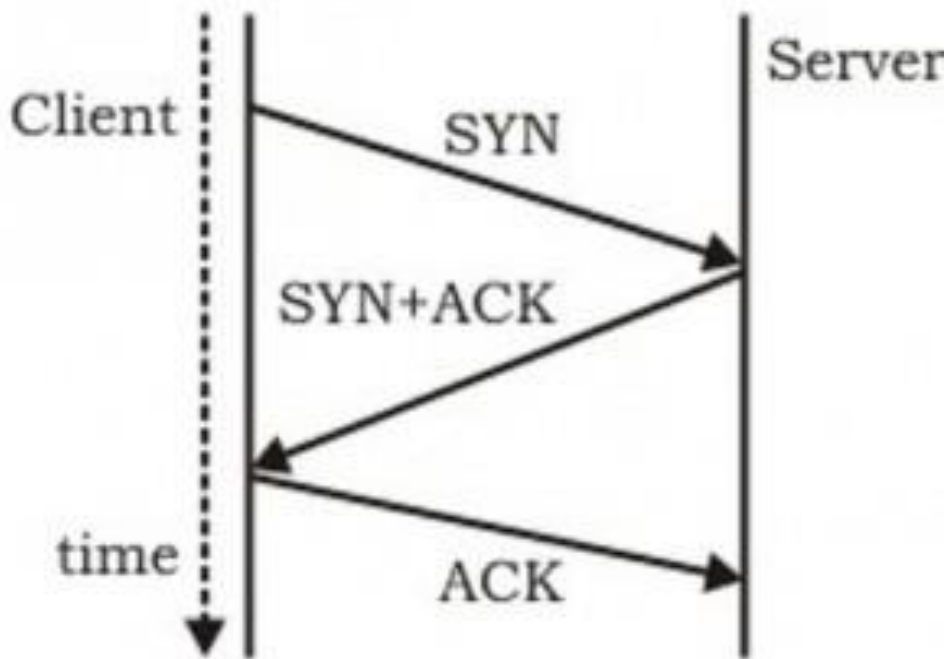
**Q** Consider the following statements. **(Gate-2015) (1 Marks)**

- I. TCP connections are full duplex.
- II. TCP has no option for selective acknowledgment
- III. TCP connections are message streams.

- (A) Only I is correct
- (B) Only I and II are correct
- (C) Only II and III are correct
- (D) All of I, II and III are correct

Answer: (A)

Q The three-way handshake for TCP connection establishment is shown below.



Which of the following statements are TRUE?

- (S1) Loss of SYN + ACK from the server will not establish a connection
  - (S2) Loss of ACK from the client cannot establish the connection
  - (S3) The server moves LISTEN → SYN\_RCVD → SYN\_SENT → ESTABLISHED in the state machine on no packet loss
  - (S4) The server moves LISTEN → SYN\_RCVD → ESTABLISHED in the state machine on no packet loss. (Gate-2008) (2 Marks)
- (A) S2 and S3 only      (B) S1 and S4      (C) S1 and S3      (D) S2 and S4

Answer: (B)

Q Consider a TCP connection in a state where there are no outstanding ACKs. The sender sends two segments back to back. The sequence numbers of the first and second segments are 230 and 290 respectively. The first segment was lost, but the second segment was received correctly by the receiver. Let X be the amount of data carried in the first segment

(in bytes), and Y be the ACK number sent by the receiver. The values of X and Y (in that order) are **(Gate-2007) (1 Marks)**

**(A)** 60 and 290

**(B)** 230 and 291

**(C)** 60 and 231

**(D)** 60 and 230

**Answer: (D)**

Sanchit Jain

## Congestion Control

- **Congestion:** Congestion refers to a network state where, the message traffic becomes so heavy that it slows down the network response time.
- Congestion control refers to techniques and mechanisms that can: Either prevent congestion before it happens or remove congestion after it has happened
  - TCP reacts to Congestion by reducing the sender window size.
  - TCP uses a combination of GBN and SR protocols to provide reliability.

## Windows in TCP

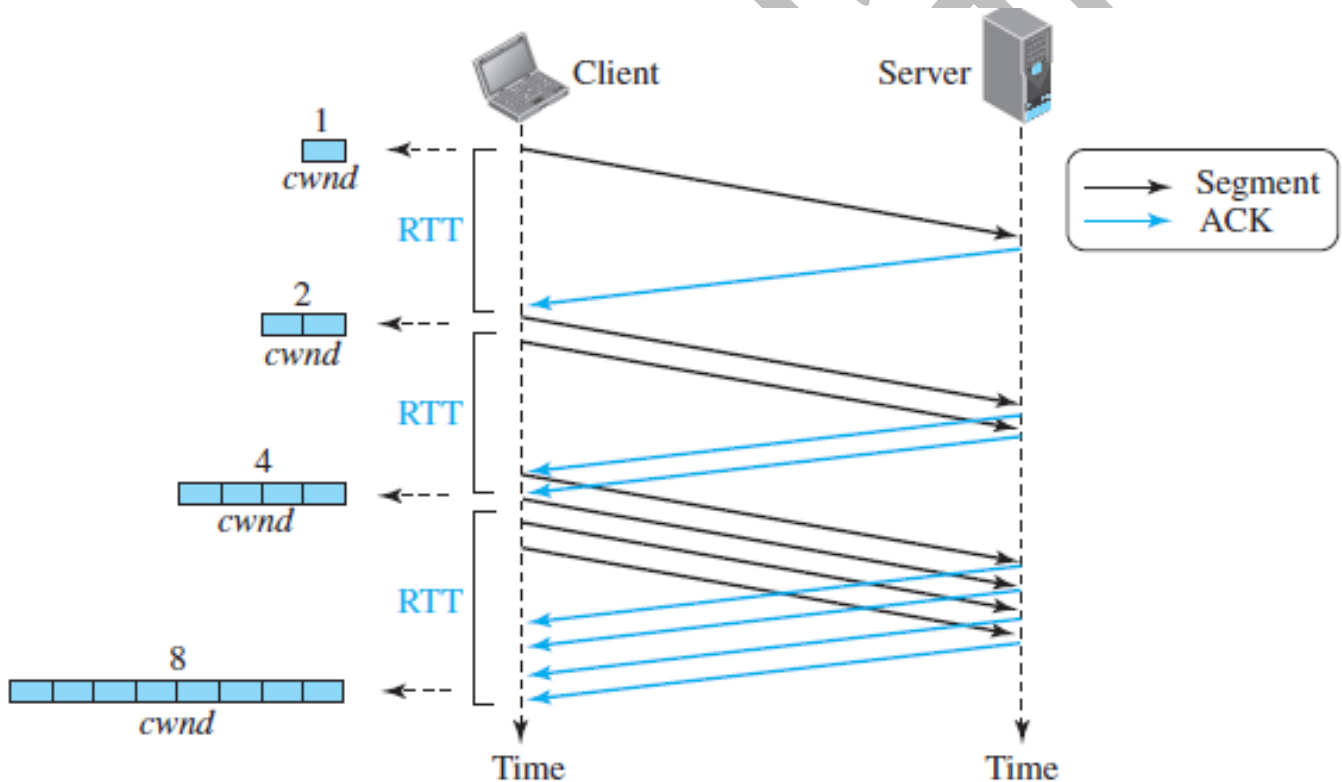
- TCP uses two windows (send window and receive window) for each direction of data transfer, i.e. four windows for a bidirectional communication.
- **Send Window**
  - The size of the sender window is determined by the following two factors
  - **Receiver window size and Congestion window size.**
- **Receive Window**
  - Sender should not send data greater than receiver window size. Otherwise, it leads to dropping the TCP segments which causes TCP Retransmission.
  - So, sender should always send data less than or equal to receiver window size. Receiver dictates its window size to the sender through TCP Header.
- **Congestion Window**
  - Sender should not send data greater than congestion window size. Otherwise, it leads to dropping the TCP segments which causes TCP Retransmission.
  - So, sender should always send data less than or equal to congestion window size.
  - Different variants of TCP use different approaches to calculate the size of congestion window. Congestion window is known only to the sender and is not sent over the links.
- **In general, Sender window size = Minimum (Receiver window size, Congestion window size)**

## TCP Congestion Policy

- TCP's general policy for handling congestion consists of following three phases
  - Slow Start (Exponential Increase)
  - Congestion Avoidance (Additive Increase)
  - Congestion Detection

### Slow Start Phase (Exponential Increase)

- Initially, sender sets congestion window size = Maximum Segment Size (1 MSS).
- After receiving each acknowledgment, the size of congestion window increases exponentially.



- After 1 round trip time, congestion window size =  $(2)^1 = 2$  MSS
- After 2 round trip time, congestion window size =  $(2)^2 = 4$  MSS
- After 3 round trip time, congestion window size =  $(2)^3 = 8$  MSS and so on.
- This phase continues until the congestion window size reaches the slow start threshold.
- **Threshold = Maximum number of TCP segments that receiver window can accommodate / 2 = (Receiver window size / Maximum Segment Size) / 2**

**Q** Consider the following statements regarding the slow start phase of the TCP congestion control algorithm. Note that *cwnd* stands for the TCP congestion window and MSS denotes the Maximum Segment Size.

- (i) The *cwnd* increase by 2 MSS on every successful acknowledgement.
- (ii) The *cwnd* approximately doubles on every successful acknowledgement.
- (iii) The *cwnd* increase by 1 MSS every round-trip time.
- (iv) The *cwnd* approximately doubles every round-trip time.

Which one of the following is correct? **(GATE-2018) (1 Marks)**

- (a) Only (ii) and (iii) are true
- (b) Only (i) and (iii) are true
- (c) Only (iv) is true
- (d) Only (i) and (iv) is true

**Q** In the slow start phase of the TCP congestion algorithm, the size of the congestion window: **(Gate-2008) (2 Marks)**

- a) does not increase
- b) increase linearly
- c) increases quadratically
- d) increases exponentially

## Congestion Avoidance Phase

- After reaching the threshold, Sender increases the congestion window size linearly to avoid the congestion.
- On receiving each acknowledgement, sender increments the congestion window size by 1.
- **Congestion window size = Congestion window size + 1**, This phase continues until the congestion window size becomes equal to the receiver window size.

## Congestion Detection Phase

- When sender detects the loss of segments, it reacts in different ways depending on how the loss is detected
- **Detection On Time Out**
  - Time Out Timer expires before receiving the acknowledgement for a segment. It suggests the strong possibility of congestion in the network. There are chances that a segment has been dropped in the network.
  - **Reaction:** In this case, sender reacts by
    - Setting the slow start threshold to half of the current congestion window size.
    - Decreasing the congestion window size to 1 MSS.
    - Resuming the slow start phase.
- **Detection On Receiving 3 Duplicate Acknowledgements**
  - Sender receives 3 duplicate acknowledgements for a segment. This case suggests the weaker possibility of congestion in the network. There are chances that a segment has been dropped but few segments sent later may have reached.
  - **Reaction**
    - In this case, sender reacts by setting the slow start threshold to half of the current congestion window size.
    - Decreasing the congestion window size to slow start threshold.
    - Resuming the congestion avoidance phase.

**Q** Let the size of congestion window of a TCP connection be 32 KB when a timeout occurs. The round-trip time of the connection is 100 msec and the maximum segment size used is 2 KB. The time taken (in msec) by the TCP connection to get back to 32 KB congestion window is \_\_\_\_\_. **(Gate-2014) (2 Marks)**

**ANSWER 1100 to 1300**

**Q** Consider an instance of TCP's Additive Increase Multiplicative Decrease (AIMD) algorithm where the window size at the start of the slow start phase is 2 MSS and the threshold at the start of the first transmission is 8 MSS. Assume that a timeout occurs during the fifth transmission. Find the congestion window size at the end of the tenth transmission. **(Gate-2012) (2 Marks)**

a) 8 MSS

b) 14 MSS

c) 7 MSS

d) 12 MSS

**ANSWER 3**

**Q** On a TCP connection, current congestion window size is Congestion Window = 4 KB. The window size advertised by the receiver is Advertise Window = 6 KB. The last byte sent by the sender is LastByteSent = 10240 and the last byte acknowledged by the receiver is LastByteAked = 8192. The current window size at the sender is **(Gate-2005) (2 Marks)**

(A) 2048 bytes

(B) 4096 bytes

(C) 6144 bytes

(D) 8192 bytes

**Answer: (B)**

**Q** Suppose that the maximum transmit window size for a TCP connection is 12000 bytes. Each packet consists of 2000 bytes. At some point of time, the connection is in slow-start phase with a current transmit window of 4000 bytes. Subsequently, the transmitter receives two acknowledgements. Assume that no packets are lost and there are no time-outs. What is the maximum possible value of the current transmit window? **(Gate-2004) (2 Marks)**

(A) 4000 bytes

(B) 8000 bytes

(C) 10000 bytes

(D) 12000 bytes

**Answer: (B)**

**Q** Which one of the following statements is FALSE? **(Gate-2004) (1 Marks)**

(A) TCP guarantees a minimum communication rate

(B) TCP ensures in-order delivery

(C) TCP reacts to congestion by reducing sender window size

(D) TCP employs retransmission to compensate for packet loss

**Answer: (A)**

## Timer

- Time-wait timer (Take care of late packets)
  - never close connection immediately, otherwise the port no will be available for some other process, generally we wait for  $2*LT$ . If some packet arrives late then there will be a problem.
- Keep-alive timer
  - Server periodically check connection and close them.
  - after keep-alive timer sends 10 probe messages with a gap of 75 seconds and in case of no reply, will close the connection.
- Persistent timer
  - Window size zero advertise
- Acknowledgement time
  - ack timer is used to generate cumulative ack and piggybacking ack
- Time-out timer
  - will be discussed in detail in next section.

## Network Traffic And Time Out Timer

- TCP uses a time out timer for retransmission of lost segments.
- The value of time out timer is dynamic and changes with the amount of traffic in the network.
- Consider Receiver has sent the ACK to the sender and the ACK is on its way through the network. Now, following two cases are possible
  - **High traffic:** If there is high traffic in the network, the time taken by the ACK to reach the sender will be more. So, as per the high traffic, the value of time out timer should be kept large.
    1. **If the value is kept small**, then timer will time out soon. It causes the sender to assume that the segment is lost before reaching the receiver. However, in actual the ACK is delayed due to high traffic. Sender keeps retransmitting the same segment. This overburdens the network and might lead to congestion.
  - **Low traffic:** If there is low traffic in the network, the time taken by the ACK to reach the sender will be less. So, as per the low traffic, the value of time out timer should be kept small.
    1. **If the value is kept large**, Timer will not time out soon. Sender keeps waiting for the ACK even when it is actually lost. This causes excessive delay.

- **Conclusion:** The setting of the time-out timer is very important if we want to use the network efficiently, and the value of the timer must change based on the change in the network scenario.

## Algorithms for Computing Time Out Timer Value

- The algorithms used for computing the value of time out timer dynamically are-
  1. Basic Algorithm
  2. Jacobson's Algorithm
  3. Karn's modification

## General Rules for Algorithms (Basic algorithm)

- **Rule-01**
  - The value of time out timer for the next segment is increased when Actual round-trip time for the previous segment is found to be increased indicating there is high traffic in the network.
- **Rule-02**
  - The value of time out timer for the next segment is decreased when Actual round-trip time for the previous segment is found to be decreased indicating there is low traffic in the network.
- **Basic Algorithm** The steps followed under Basic Algorithm are-
- **Step-01: Sending 1st Segment**
  - Sender assumes any random value of initial RTT say  $IRTT_1$ .
  - So, after sending the 1st segment, sender expects its ACK to arrive in time  $IRTT_1$ .
  - Sender sets time out timer value (TOT) for the 1st segment to be-  $TOT_1 = 2 \times IRTT_1$
  - Suppose ACK for the 1st segment arrives in time  $ARTT_1$ . Here,  $ARTT_1 =$  Actual Round-Trip Time for the 1st segment.
- **Step-02: Sending 2nd Segment**
  - Sender computes the value of initial RTT for the 2nd segment using the relation  $IRTT_{n+1} = \alpha IRTT_n + (1 - \alpha) ARTT_n$
  - Here,  $\alpha$  is called smoothing factor where  $0 \leq \alpha \leq 1$  (Its value will be given in questions)
  - Substituting  $n=1$ , sender gets  $IRTT_2 = \alpha IRTT_1 + (1 - \alpha) ARTT_1$ .

- So, after sending the 2nd segment, sender expects its ACK to arrive in time  $IRTT_2$ .
- Sender sets time out timer value (TOT) for the 2nd segment to be-  **$TOT_2 = 2 \times IRTT_2$**
- Suppose ACK for the 2nd segment arrives in time  $ARTT_2$ .
- Here,  $ARTT_2 =$  Actual Round-Trip Time for the 2nd segment.
- In the similar manner, algorithm computes the time out timer value for all the further segments.

### Advantages

- Time out timer value is flexible to dynamic round trip time.
- It takes into consideration all the previously sent segments to derive the initial RTT for the current segment.

### Disadvantage

- It always considers Time out timer value = 2 x Initial round trip time.
- There is no logic behind using the number 2.

## Jacobson's Algorithm

- Jacobson's Algorithm is a modified version of the basic algorithm.
- It gives better performance than Basic Algorithm.
- The steps involved in Jacobson's Algorithm are
- **Step-01: Sending 1st Segment-**
  - Sender assumes any random value of initial RTT say  $IRTT_1$ .
  - So, after sending the 1st segment, sender expects its ACK to arrive in time  $IRTT_1$ .
  - Sender assumes any random value of initial deviation say  $ID_1$ .
  - So, after sending the 1st segment, sender expects there will be a deviation of  $ID_1$  time from  $IRTT_1$ .
  - Sender sets time out timer value (TOT) for the 1st segment to be-
    1.  **$TOT_1 = 4 \times ID_1 + IRTT_1$**
  - Suppose ACK for the 1st segment arrives in time  $ARTT_1$ . Here,  $ARTT_1 =$  Actual Round-Trip Time for the 1st segment.
  - Then, Actual deviation from  $IRTT_1$  is given by-
  - $AD_1 = | IRTT_1 - ARTT_1 |$

## • Step-02: Sending 2nd Segment-

- Sender computes the value of initial RTT for the 2nd segment using the relation-
  1.  $IRTT_{n+1} = \alpha IRTT_n + (1 - \alpha) ARTT_n$
  2. Here,  $\alpha$  is called smoothing factor where  $0 \leq \alpha \leq 1$  (Its value will be given in questions)
- Sender computes the value of initial deviation for the 2nd segment using the relation
  1.  $ID_{n+1} = \alpha ID_n + (1 - \alpha) AD_n$
  2. Here,  $\alpha$  is called smoothing factor where  $0 \leq \alpha \leq 1$  (Its value will be given in questions)
- Substituting  $n=1$ , sender gets
  1.  $IRTT_2 = \alpha IRTT_1 + (1 - \alpha) ARTT_1$
  2.  $ID_2 = \alpha ID_1 + (1 - \alpha) AD_1$
- So after sending the 2nd segment, sender expects its ACK to arrive in time  $IRTT_2$  with deviation of  $ID_2$  time.
- Sender sets time out timer value (TOT) for the 2nd segment to be-
  1.  $TOT_2 = 4 \times ID_2 + IRTT_2$
- Suppose ACK for the 2nd segment arrives in time  $ARTT_2$ . Here,  $ARTT_2 =$  Actual Round Trip Time for the 2nd segment.
- Then, Actual deviation from  $IRTT_2$  is given by-
  1.  $AD_2 = | IRTT_2 - ARTT_2 |$
- In the similar manner, algorithm computes the time out timer value for all the further segments.

## Problems with Basic Algorithm and Jacobson's Algorithm

- To calculate initial round trip time, both the algorithms depend on the actual round-trip time of the previous segment through the relation
  1.  $IRTT_{n+1} = \alpha IRTT_n + (1 - \alpha) ARTT_n$
- Consider ACK of some segment arrives to the sender after its initial time out timer goes off. Then, sender will have to re transmit the segment.
- Now for the segment being re transmitted, what should be the initial time out timer value is the concern.
- This is because the ACK is delayed and will arrive after time out. So, ARTT is not available.
- This problem is resolved by Karn's modification.

## Karn's Modification

- Karn's modification states:
  - Whenever a segment has to be re transmitted, do not apply either of Basic or Jacobson's algorithm since actual RTT is not available.
  - Instead, double the time out timer (TOT) whenever the timer times out and make a retransmission.

Sanchit Jain

## Silly Window Syndrome

- Silly Window Syndrome is a problem that arises due to the poor implementation of TCP.
- It degrades the TCP performance and makes the data transmission extremely inefficient.
- The problem is called so because
  1. It causes the sender window size to shrink to a silly value.
  2. The window size shrinks to such an extent where the data being transmitted is smaller than TCP Header.
- **The problem arises due to following causes**
  1. Sender transmitting data in small segments repeatedly
  2. Receiver accepting only few bytes at a time repeatedly
- This problem is solved using Nagle's Algorithm.

## Nagle's Algorithm

- Nagle's Algorithm tries to solve the problem caused by the sender delivering 1 data byte at a time. Nagle's algorithm suggests
  - Sender should send only the first byte on receiving one-byte data from the application.
  - Sender should buffer all the rest bytes until the outstanding byte gets acknowledged. In other words, sender should wait for 1 RTT.
  - After receiving the acknowledgement, sender should send the buffered data in one TCP segment.
  - Then, sender should buffer the data again until the previously sent data gets acknowledged.

## Clark's Solution

- **Receiver Accepting Only Few Bytes Repeatedly**
  - Consider the receiver continues to be unable to process all the incoming data.
  - In such a case, its window size becomes smaller and smaller.
  - A stage arrives when it repeatedly sends the window size of 1 byte to the sender.
  - **This problem is solved using Clark's Solution.**
- Clark's Solution tries to solve the problem caused by the receiver sucking up one data byte at a time. **Clark's solution suggests-**
  - Receiver should not send a window update for 1 byte.
  - Receiver should wait until it has a decent amount of space available.
  - Receiver should then advertise that window size to the sender.
  - **Specifically, the receiver should not send a window update**
    - Until it can handle the MSS it advertised during Three Way Handshake
    - Or until its buffer is half empty, whichever is smaller.
- **Important Notes**
  - **Nagle's algorithm is turned off for the applications that require data to be sent immediately.** This is because Nagle's algorithm sends only one segment per round trip time. This impacts the latency by introducing a delay.
  - **Nagle's algorithm and Clark's solution are complementary.** Both Nagle's solution and Clark's solution can work together. The ultimate goal is sender should not send the small segments and receiver should not ask for them.

**Q** Consider the following statements about the timeout value used in TCP.

**i.** The timeout value is set to the RTT (Round Trip Time) measured during TCP connection establishment for the entire duration of the connection.

**ii.** Appropriate RTT estimation algorithm is used to set the timeout value of a TCP connection.

**iii.** Timeout value is set to twice the propagation delay from the sender to the receiver.

Which of the following choices hold? **(Gate-2007) (1 Marks)**

**(A)** (i) is false, but (ii) and (iii) are true

**(B)** (i) and (iii) are false, but (ii) is true

**(C)** (i) and (ii) are false, but (iii) is true

**(D)** (i), (ii) and (iii) are false

**Answer: (B)**

**Q** Identify the correct order in which a server process must invoke the function calls accept, bind, listen, and recv according to UNIX socket API. **(Gate-2015) (1 Marks)**

**(A)** listen, accept, bind, recv

**(B)** bind, listen, accept, recv

**(C)** bind, accept, listen, recv

**(D)** accept, listen, bind, recv

**Answer: (B)**

**Q** Which one of the following socket API functions converts an unconnected active TCP socket into a passive socket? **(Gate-2014) (1 Marks)**

**(a)** CONNECT

**(b)** BIND

**(c)** LISTEN

**(d)** ACCEPT

**ANSWER C**

**Q** A client process P needs to make a TCP connection to a server process S. Consider the following situation: the server process S executes a socket (), a bind () and a listen () system call in that order, following which it is pre-empted. Subsequently, the client process P executes a socket () system call followed by connect () system call to connect to the server process S. The server process has not executed any accept () system call. Which one of the following events could take place? **(Gate-2008) (2 Marks)**

**(A)** connect () system call returns successfully

**(B)** connect () system call blocks

**(C)** connect () system call returns an error

**(D)** connect () system call results in a core dump

**Answer: (C)**

**Q** Which of the following system calls results in the sending of SYN packets? **(Gate-2008) (1 Marks)**

**(A)** socket

**(B)** bind

**(C)** listen

**(D)** connect

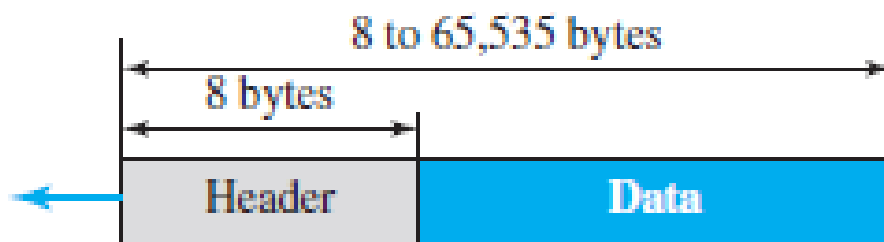
**Answer (D)**

## USER DATAGRAM PROTOCOL (UDP)

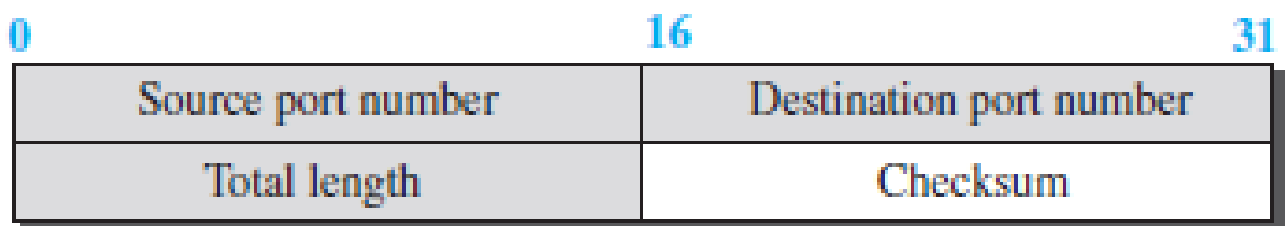
- The **User Datagram Protocol (UDP)** is a connectionless, unreliable transport protocol.
- It does not add anything to the services of IP except for providing process-to-process communication instead of host-to-host communication.
- **Why to use UDP**
  - UDP is a very simple protocol using a minimum of overhead.
  - If a process wants to send a small message and does not care much about reliability, it can use UDP.
  - Sending a small message using UDP takes much less interaction between the sender and receiver than using TCP.

### User Datagram

- UDP packets, called **user datagrams**, have a fixed-size header of 8 bytes made of four fields, each of 2 bytes (16 bits).



a. UDP user datagram



b. Header format

- The first two fields define the source and destination port numbers.
- The third field defines the total length of the user datagram, header plus data.
- The 16 bits can define a total length of 0 to 65,535 bytes. However, the total length needs to be less because a UDP user datagram is stored in an IP datagram with the total length of 65,535 bytes.
- The last field can carry the optional checksum

**Example:** The following is the content of a UDP header in hexadecimal format.

**CB8400D001C001C**

Identify: Source port number, destination port number, total length of the user datagram, length of the data.

- The source port number is the first four hexadecimal digits (CB84)<sub>16</sub>, that is the source port number is 52100. (as 1 digit of hexa defines 4 binary digits)
- The destination port number is the second four hexadecimal digits (000D)<sub>16</sub>, that is the destination port number is 13.
- The third four hexadecimal digits (001C)<sub>16</sub>, define the length of the whole UDP packet as 28 bytes.
- The length of the data is the length of the whole packet minus the length of the header,  $28 - 8 = 20$  bytes.

### UDP Services

- **Process-to-Process Communication**
  - UDP provides process-to-process communication using **socket addresses**, a combination of IP addresses and port numbers.
- **Connectionless Services**
  - Each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagrams.
  - The user datagrams are not numbered.
  - There is no connection establishment and no connection termination unlike TCP.
- Only those processes sending short messages, messages less than 65,507 bytes (65,535 minus 8 bytes for the UDP header and minus 20 bytes for the IP header), can use UDP.

### ***Flow Control***

- There is no *flow control*, and hence no window mechanism.

### ***Error Control***

- There is no *error control* mechanism in UDP except for the checksum.

### ***Congestion Control***

- It does not provide congestion control.

### **UDP Applications**

- UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control. It is not usually used for a process such as ***FTP*** that needs to send bulk data.
- UDP is suitable for a process with internal flow- and error-control mechanisms. For example, the ***Trivial File Transfer Protocol (TFTP)*** process includes flow and error control. It can easily use UDP.
- UDP is a suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software but not in the TCP software.
- UDP is used for management processes such as SNMP
- UDP is used for some route updating protocols such as Routing Information Protocol (RIP)
- UDP is normally used for interactive real-time applications that cannot tolerate uneven delay between sections of a received message.

**Q Match the following: (GATE-2018) (1 Marks)**

<u>Field</u>	<u>Length in bits</u>
P. UDP Header's Port Number	I. 48
Q. Ethernet MAC Address	II. 8
R. IPv6 Next Header	III. 32
S. TCP Header's Sequence Number	IV. 16

(a) P-III, Q-IV, R-II, S-I

(b) P-II, Q-I, R-IV, S-III

(c) P-IV, Q-I, R-II, S-III

(d) P-IV, Q-I, R-III, S-II

Ans: c

**Q Consider socket API on a Linux machine that supports connected UDP sockets. A connected UDP socket is a UDP socket on which **connect** function has already been called. Which of the following statement is/are CORRECT? (Gate-2017) (1 Marks)**

**I.** A connected UDP socket can be used to communicate with multiple peers simultaneously.

**II.** A process can successfully call **connect** function again for an already connected UDP socket

(a) I only

(b) II only

(c) Both I and II

(d) Neither I nor II

Ans: b

**Q Which of the following statements are TRUE? (Gate-2008) (2 Marks)**

**(S1)** TCP handles both congestion and flow control

**(S2)** UDP handles congestion but not flow control

**(S3)** Fast retransmit deals with congestion but not flow control

**(S4)** Slow start mechanism deals with both congestion and flow control

(A) S1, S2 and S3 only

(B) S1 and S3 only

(C) S3 and S4 only

(D) S1, S3 and S4 only

**Answer: (B)**

**Q A program on machine X attempts to open a UDP connection to port 5376 on a machine Y, and a TCP connection to port 8632 on machine Z. However, there are no applications listening at the corresponding ports on Y and Z. An ICMP Port Unreachable error will be generated by (Gate-2006) (2 Marks)**

(A) Y but not Z

(B) Z but not Y

(C) Neither Y nor Z

(D) Both Y and Z

**Answer: (D)**

**Q Packets of the same session may be routed through different paths in: (Gate-2005) (1 Marks)**

(a) TCP, but not UDP

(b) TCP and UDP

(c) UDP, but not TCP

(d) Neither TCP nor UDP

**Answer (b)**

**Q** A firewall is to be configured to allow hosts in a private network to freely open TCP connections and send packets on open connections. However, it will only allow external hosts to send packets on existing open TCP connections or connections that are being opened (by internal hosts) but not allow them to open TCP connections to hosts in the private network. To achieve this the minimum capability of the firewall should be that of

**(GATE-2007) (1 Marks)**

- (A)** A combinational circuit
- (B)** A finite automaton
- (C)** A pushdown automaton with one stack
- (D)** A pushdown automaton with two stacks

**Answer: (D)**

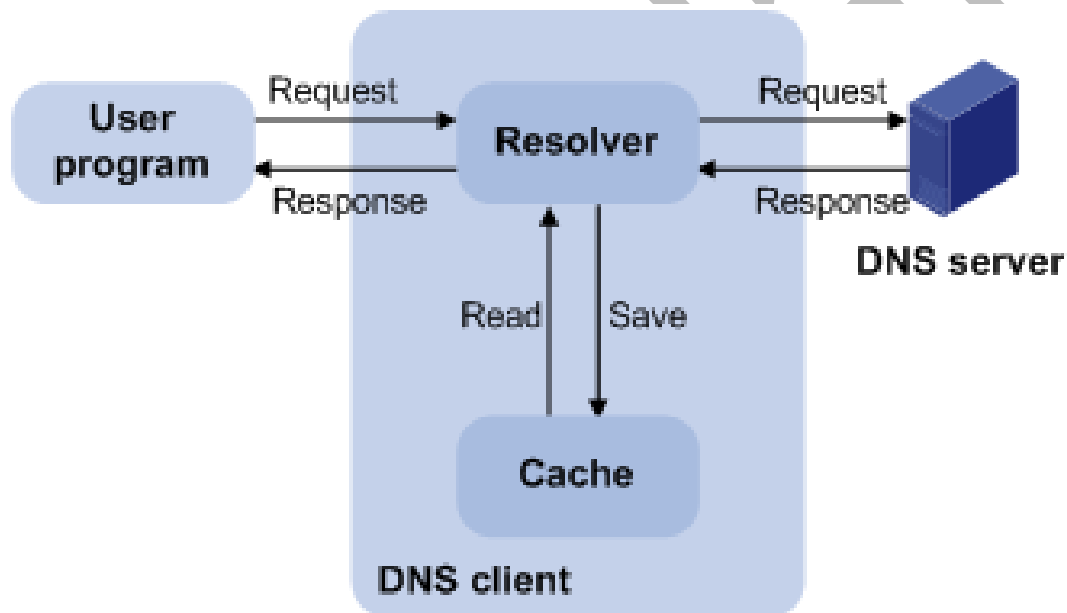
**Q** Which one of the following statements is FALSE? **(Gate-2004) (1 Marks)**

- (A)** TCP guarantees a minimum communication rate
- (B)** TCP ensures in-order delivery
- (C)** TCP reacts to congestion by reducing sender window size
- (D)** TCP employs retransmission to compensate for packet loss

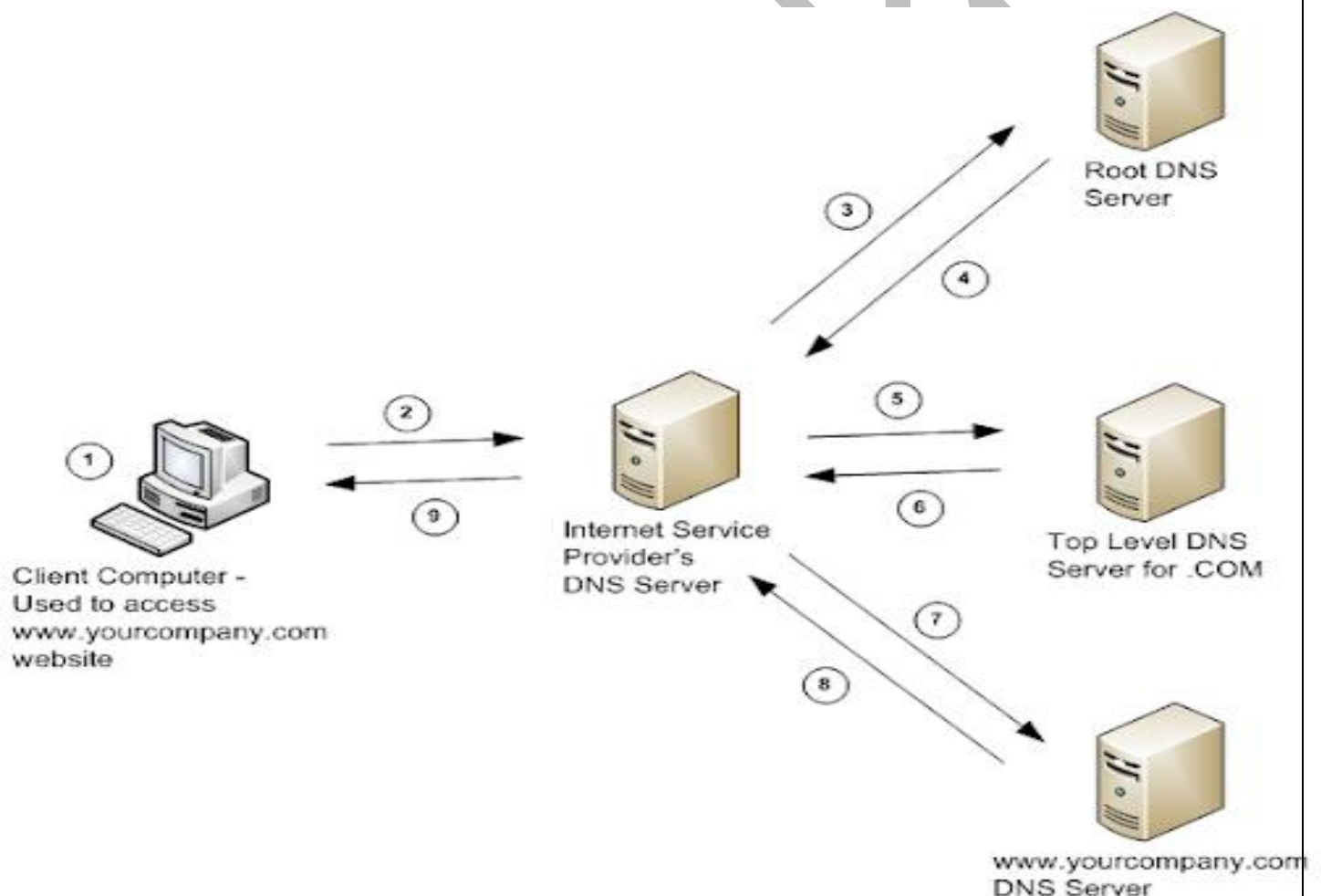
**Answer: (A)**

# DNS

- As we know human beings are not comfortable in remembering numbers so to remember IP address of a website or mail account in internet is difficult.
- Secondly IP addresses of mail or websites keeps on changing, so we have to come up with one more level of addressing which is easy to remember and do not change with time.
- Solution is Name addressing, i.e. we give some names to websites and mail account like we do to humans in real world.
- But then if someone write a name of the website in the browser we need some mechanism to convert it back into IP address.
- Domain Name System solve this problem .



- This diagram perfectly represent how DNS works, A user of a website may know the name of the website; however, the IP protocol needs the IP address.
- The DNS client program sends a request to a DNS server to map the Web-site address to the corresponding IP address.
- today we divide this huge amount of information into smaller parts and store each part on a different computer. In this method, the host that needs mapping can contact the closest computer holding the needed information.
- It is Very difficult to find out the ip address associated to a website because there are millions of websites and with all those websites, we should be able to generate the ip address immediately, there should not be a lot of delay for that to happen organization of database is very important.



## Hierarchy of Name Servers

- **Root name servers** – It is contacted by name servers that cannot resolve the name. It contacts authoritative name server if name mapping is not known. It then gets the mapping and return the IP address to the host.
- **Top level server** – It is responsible for com, org, edu etc and all top-level country domains like uk, fr, ca, in etc. They have info about authoritative domain servers and know names and IP addresses of each authoritative name server for the second level domains.
- **Authoritative name servers** This is organization's DNS server, providing authoritative hostName to IP mapping for organization servers. It can be maintained by organization or service provider. In order to reach cse.dtu.in we have to ask the root DNS server, then it will point out to the top-level domain server and then to authoritative domain name server which actually contains the IP address. So, the authoritative domain server will return the associative ip address.

## NAME SPACE

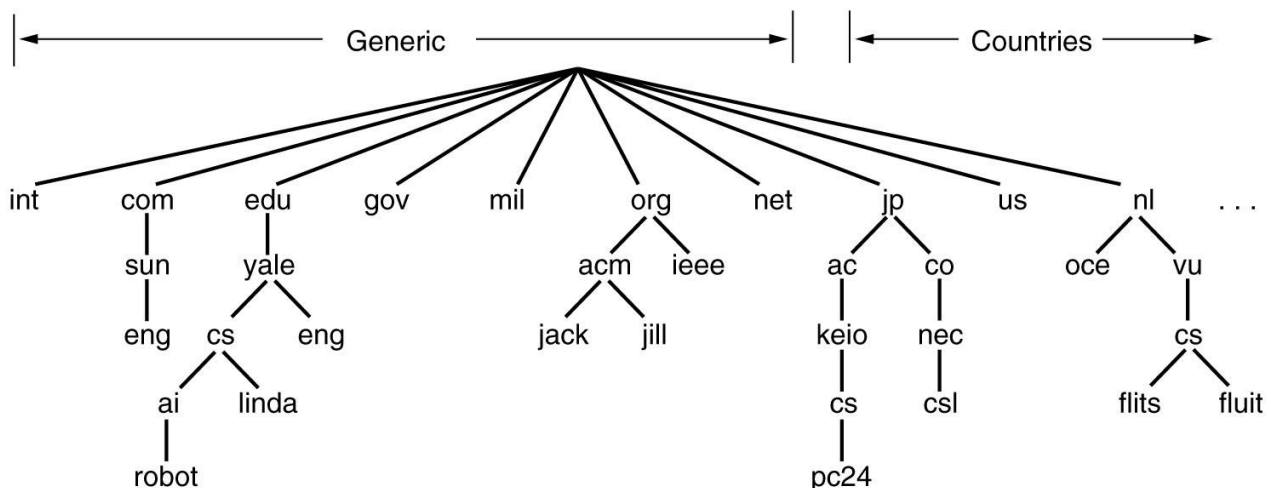
- To be unambiguous, the names must be unique because the addresses are unique. A name space that maps each address to a unique name can be organized in two ways: flat or hierarchical.

### Flat Name Space

- In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure.
- The main disadvantage of a flat name space is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.
- So, Solution is Hierarchical Name Space

## Hierarchical Name Space

- In a hierarchical name space, each name is made of several parts. The first part can define the nature of the organization
- the second part can define the name of an organization, the third part can define departments in the organization, and so on.
- In this case, the authority to assign and control the name spaces can be decentralized. A central authority can assign the part of the name that defines the nature of the organization and the name of the organization.
- The responsibility of the rest of the name can be given to the organization itself.
- The management of the organization need not worry that the prefix chosen for a host is taken by another organization because, even if part of an address is the same, the whole address is different



- Generic domain: .com(commercial) .edu(educational) .mil(military) .org (non-profit organization) .net (similar to commercial) all these are generic domain.
- Country domain .in (india) .us .uk
- Inverse domain if we want to know what is the domain name of the website. Ip to domain name mapping. So, DNS can provide both the mapping for example to find the ip addresses of www.knowledgegate.in then we have to type nslookup www.knowledgegate.in

## DOMAIN NAME SPACE

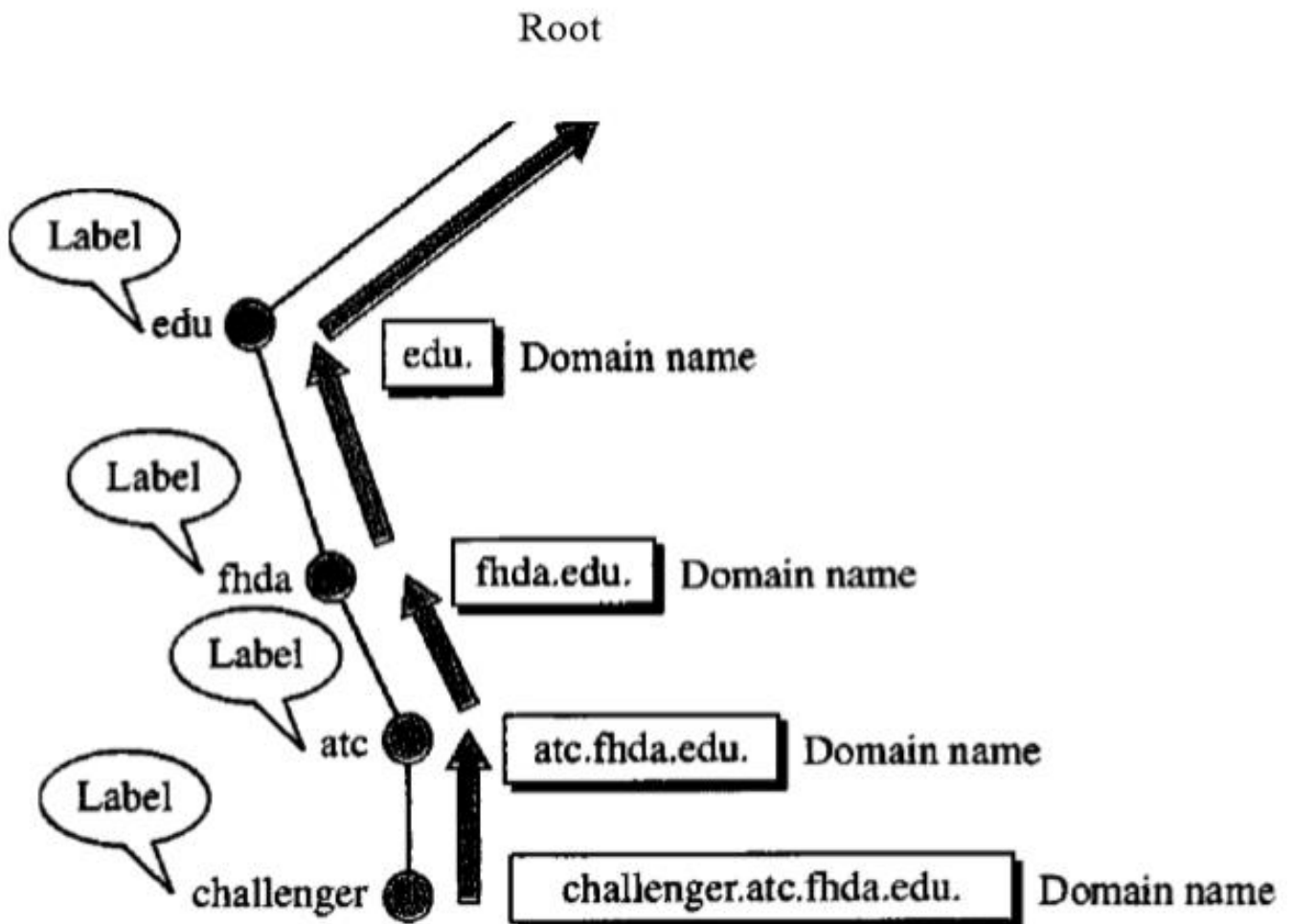
- To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127

<i>Label</i>	<i>Description</i>
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to "com")
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International organizations
mil	Military groups
museum	Museums and other nonprofit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

- **Label**
  - Each node in the tree has a label, which is a string with a maximum of 63 characters.
  - The root label is a null string (empty string).
  - DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

- **Domain Name**

- Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.).
- The domain names are always read from the node up to the root. The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.

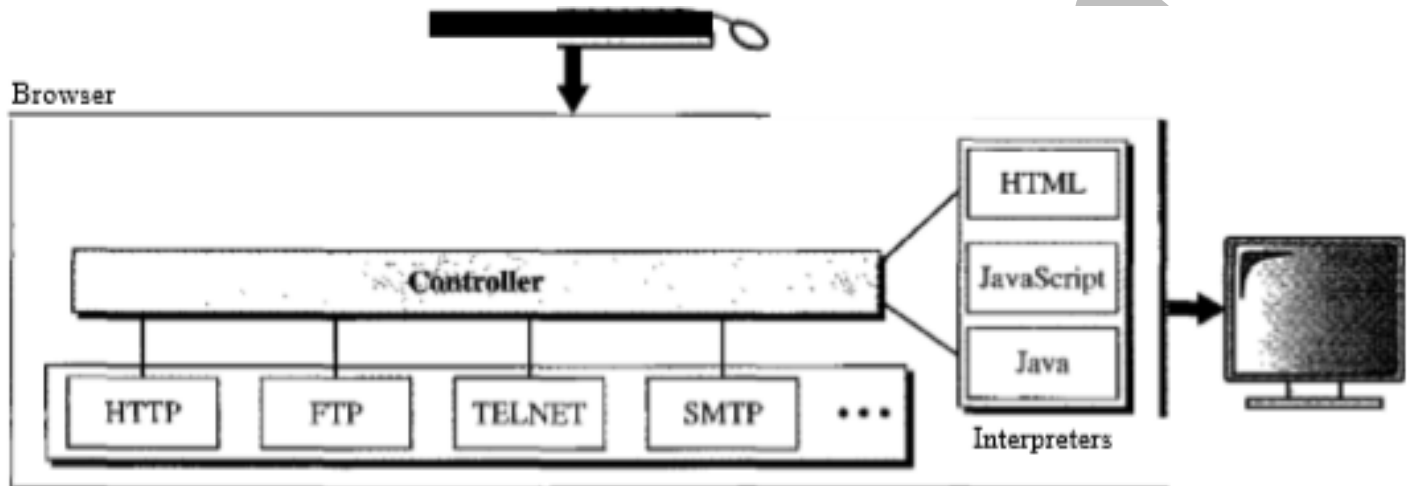


## WWW

- The idea of the Web was first proposed by Tim Berners-Lee in 1989 at *CERN*
  - The World Wide Web (WWW) is a repository of information linked together from points all over the world.
  - The WWW has a unique combination of flexibility, portability, and user-friendly features that distinguish it from other services provided by the Internet.
  - The WWW project was initiated by CERN (European Laboratory for Particle Physics) to create a system to handle distributed resources necessary for scientific research.
- **ARCHITECTURE**
    1. The WWW today is a distributed client server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called sites.
    2. Each site holds one or more documents, referred to as Web pages. Each Web page can contain a link to other pages in the same site or at other sites. The pages can be retrieved and viewed by using browsers.

- **Client (Browser)**

1. A variety of vendors offer commercial browsers that interpret and display a Web document, and all use nearly the same architecture.
2. Each browser usually consists of three parts: a controller, client protocol, and interpreters.
3. The controller receives input from the keyboard or the mouse and uses the client programs to access the document. After the document has been accessed, the controller uses one of the interpreters to display the document on the screen.



- **Server**

1. The Web page is stored at the server. Each time a client request arrives, the corresponding document is sent to the client.
2. To improve efficiency, servers normally store requested files in a cache in memory; memory is faster to access than disk.
3. A server can also become more efficient through multithreading or multiprocessing. In this case, a server can answer more than one request at a time.
4. A client that wants to access a Web page needs the address. To facilitate the access of documents distributed throughout the world, HTTP uses locators.

5. **Uniform Resource Locator (URL)**

1. A web page, as a file, needs to have a unique identifier to distinguish it from other web pages.
2. To define a web page, we need four identifiers in general: **Protocol, host, port, and path.**
3. **Protocol.** Which client-server application we are using is called protocol. Although most of the time the protocol is HTTP (Hyper Text Transfer

Protocol), we can also use other protocols such as FTP (File Transfer Protocol).

4. **Host.** The host identifier can be the IP address of the server or the unique name to the server.
5. **Port.** The port, a 16-bit integer, is normally predefined for the client-server application.
6. To combine these four pieces together, the **uniform resource locator (URL)** has been designed; it uses three different separators between the four pieces as shown below:

Sanchit Jain

- **Cookies**

1. The World Wide Web was originally designed as a stateless entity. A client sends a request; a server responds. Their relationship is over. The original design of WWW, retrieving publicly available documents, exactly fits this purpose. Today the Web has other functions; some are listed here.
  1. Some websites need to allow access to registered clients only.
  2. Websites are being used as electronic stores that allow users to browse through the store, select wanted items, put them in an electronic cart, and pay at the end with a credit card.
  3. Some websites are used as portals: the user selects the Web pages he wants to see.
  4. Some websites are just advertising.
2. For these purposes, the cookie mechanism was devised.
  1. Creation and Storage of Cookies: - The creation and storage of cookies depend on the implementation; however, the principle is the same.
    1. When a server receives a request from a client, it stores information about the client in a file or a string. The information may include the domain name of the client, the contents of the cookie (information the server has gathered about the client such as name, registration number, and so on), a timestamp, and other information depending on the implementation.
    2. The server includes the cookie in the response that it sends to the client.
    3. When the client receives the response, the browser stores the cookie in the cookie directory, which is sorted by the domain server name.
  2. Using Cookies: - When a client sends a request to a server, the browser looks in the cookie directory to see if it can find a cookie sent by that server. If found, the cookie is included in the request.
    1. When the server receives the request, it knows that this is an old client, not a new one. Note that the contents of the cookie are never read by the browser or disclosed to the user. It is a cookie made by the server and eaten by the server. Now let us see how a cookie is used for the four previously mentioned purposes:
      2. The site that restricts access to registered clients only sends a cookie to the client when the client registers for the first time. For any repeated access, only those clients that send the appropriate cookie are allowed.

3. An electronic store (e-commerce) can use a cookie for its client shoppers. When a client selects an item and inserts it into a cart, a cookie that contains information about the item, such as its number and unit price, is sent to the browser. If the client selects a second item, the cookie is updated with the new selection information. And so on. When the client finishes shopping and wants to check out, the last cookie is retrieved and the total charge is calculated.
4. A Web portal uses the cookie in a similar way. When a user selects her favourite pages, a cookie is made and sent. If the site is accessed again, the cookie is sent to the server to show what the client is looking for.
5. A cookie is also used by advertising agencies. An advertising agency can place banner ads on some main website that is often visited by users. The advertising agency supplies only a URL that gives the banner address instead of the banner itself. When a user visits the main website and clicks on the icon of an advertised corporation, a request is sent to the advertising agency. The advertising agency sends the banner, a GIF file, for example, but it also includes a cookie with the ID of the user. Any future use of the banners adds to the database that profiles the Web behaviour of the user. The advertising agency has compiled the interests of the user and can sell this information to other parties. This use of cookies has made them very controversial. Hopefully, some new regulations will be devised to preserve the privacy of users.

# HTTP

- The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web. The **Hyper Text Transfer Protocol (HTTP)** is used to define how the client-server programs can be written to retrieve web pages from the Web.
- HTTP uses the services of TCP on well-known port 80, the client uses a temporary port number.
- It is a connection-oriented and reliable protocol.
- HTTP functions as a combination of FTP and SMTP.
- It is similar to FTP because it transfers files and uses the services of TCP. However, it is much simpler than FTP because it uses only one TCP connection. There is no separate control connection; only data are transferred between the client and the server.
- HTTP is like SMTP because the data transferred between the client and the server look like SMTP messages. In addition, the format of the messages is controlled by MIME-like headers.
- Unlike SMTP, the HTTP messages are not destined to be read by humans; they are read and interpreted by the HTTP server and HTTP client (browser).
- SMTP messages are stored and forwarded, but HTTP messages are delivered immediately.
- The commands from the client to the server are embedded in a request message. The contents of the requested file or other information are embedded in a response message.
- **Proxy Server**
  1. HTTP supports proxy servers. A proxy server is a computer that keeps copies of responses to recent requests.
  2. The HTTP client sends a request to the proxy server. The proxy server checks its cache. If the response is not stored in the cache, the proxy server sends the request to the corresponding server.
  3. Incoming responses are sent to the proxy server and stored for future requests from other clients.
  4. The proxy server reduces the load on the original server, decreases traffic, and improves latency. However, to use the proxy server, the client must be configured to access the proxy instead of the target server.

## ***Nonpersistent versus Persistent Connections***

- **Nonpersistent Connections**: - In a **nonpersistent connection**, one TCP connection is made for each request/response. The following lists the steps in this strategy:
  - The client opens a TCP connection and sends a request.
  - The server sends the response and closes the connection.
  - The client reads the data until it encounters an end-of-file marker; it then closes the connection.
  - **For example**: If a file contains links to N different pictures in different files (all located on the same server), the connection must be opened and closed N + 1 times.
  - **Disadvantage** The nonpersistent strategy imposes high overhead on the server because the server needs N + 1 different buffer each time a connection is opened.
- **Persistent Connections**
  - HTTP version 1.1 specifies a **persistent connection** by default.
  - In a persistent connection, the server leaves the connection open for more requests after sending a response.
  - The server can close the connection at the request of a client or if a time-out has been reached.
  - **Advantages**
    - Time and resources are saved using persistent connections.
    - Only one set of buffers and variables needs to be set for the connection at each site.
    - The round-trip time for connection establishment and connection termination is saved.

### ***It is important to know that HTTP is a stateless protocol as:***

- HTTP server does not maintain any state. It forgets about the client after sending the response.
- It treats every new request independently.

## ***HTTP Security***

- HTTP per se does not provide security.
- HTTP can be run over the Secure Socket Layer (SSL). In this case, HTTP is referred to as HTTPS.
- HTTPS provides confidentiality, client and server authentication, and data integrity.

**Q** Identify the correct sequence in which the following packets are transmitted on the network by a host when a browser requests a webpage from a remote server, assuming that the host has just been restarted. **(GATE-2016) (2 Marks)**

**(A)** HTTP GET request, DNS query, TCP SYN

**(B)** DNS query, HTTP GET request, TCP SYN

**(C)** DNS query, TCP SYN, HTTP GET request

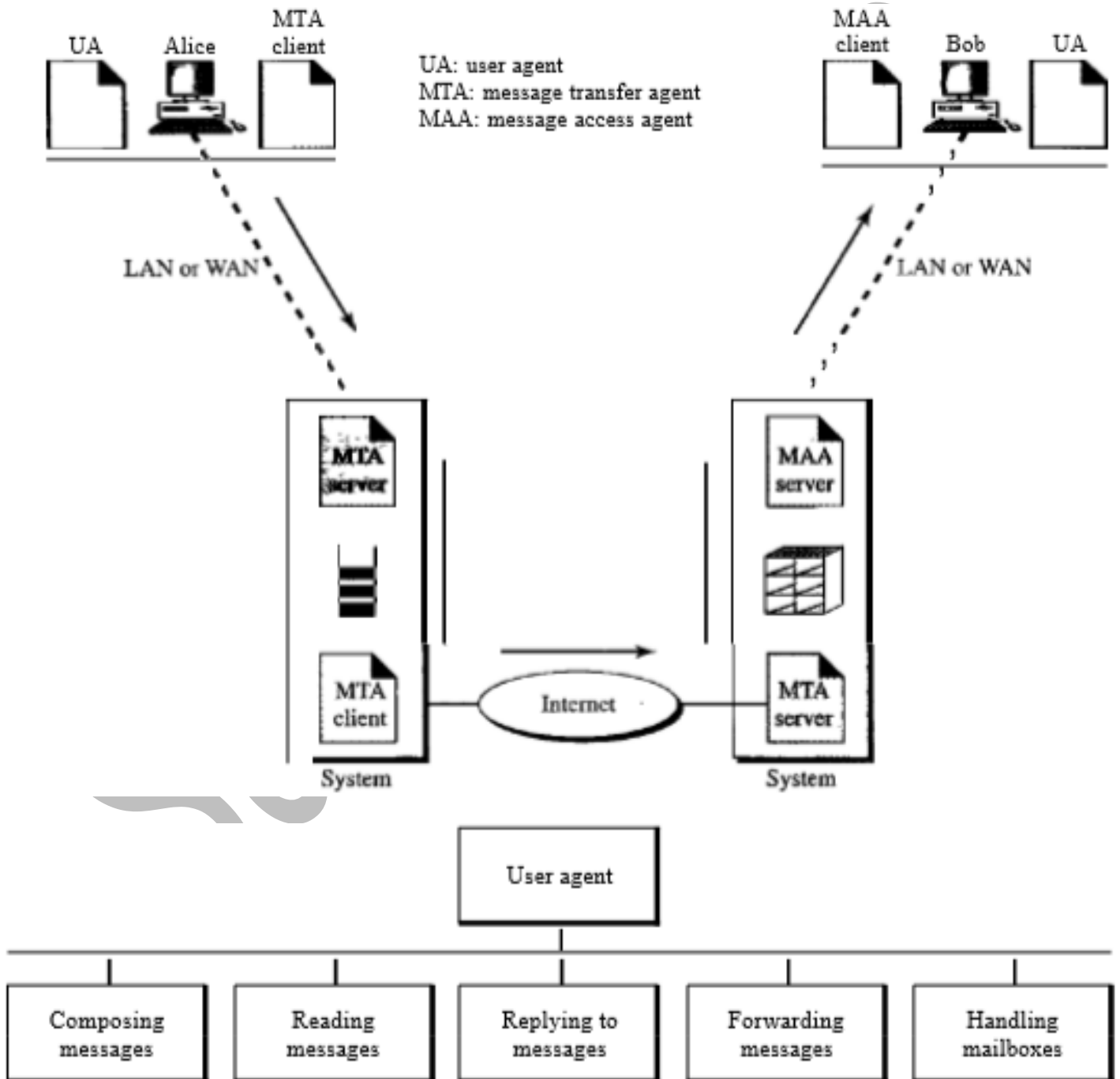
**(D)** TCP SYN, DNS query, HTTP GET request

**Answer: (C)**

Sanchit Jain

# ELECTRONIC MAIL

- One of the most popular Internet services is electronic mail (e-mail). The designers of the Internet probably never imagined the popularity of this application program.
- At the beginning of the Internet era, the messages sent by electronic mail were short and consisted of text only; they let people exchange quick memos.
- Today, electronic mail is much more complex. It allows a message to include text, audio, and video. It also allows one message to be sent to one or more recipients.

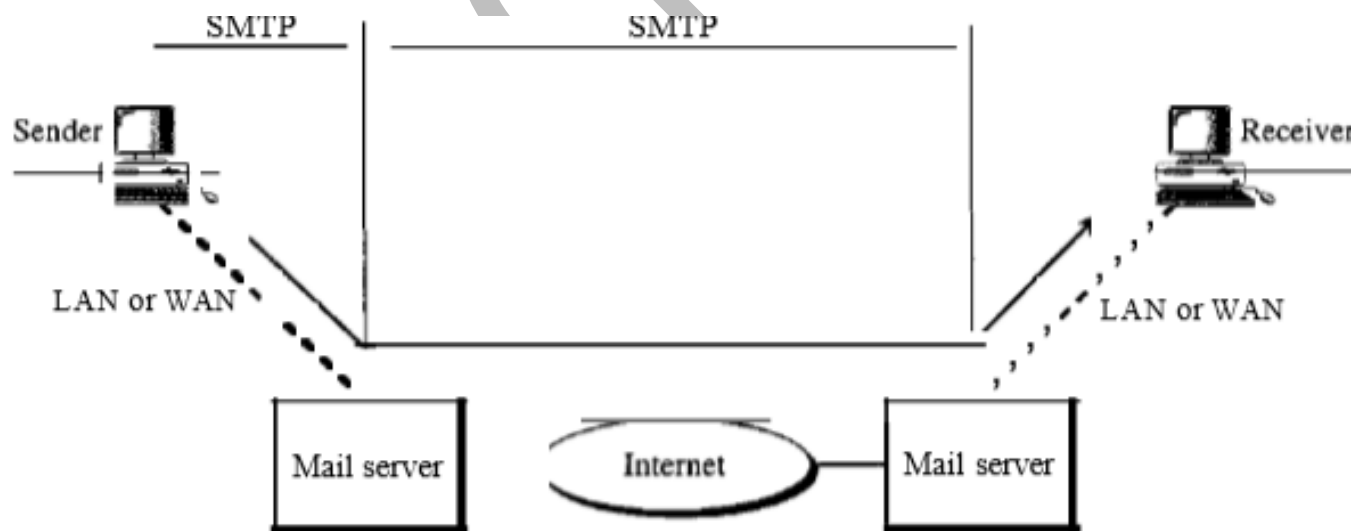


## MIME

- Electronic mail has a simple structure. Its simplicity, however, comes at a price. It can send messages only in NVT 7-bit ASCII format. In other words, it has some limitations. For example, it cannot be used for languages that are not supported by 7-bit ASCII characters (such as Hindi, French, German, Hebrew, Russian, Chinese, and Japanese).
- Also, it cannot be used to send binary files or video or audio data.
- Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that allows non-ASCII data to be sent through e-mail.
- MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers them to the client MTA to be sent through the Internet. The message at the receiving side is transformed back to the original data.

## Message Transfer Agent: SMTP

- The actual mail transfer is done through message transfer agents. To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA.
- The formal protocol that defines the MTA client and server in the Internet is called the Simple Mail Transfer Protocol (SMTP).



- SMTP is used two times, between the sender and the sender's mail server and between the two mail servers
- SMTP simply defines how commands and responses must be sent back and forth.
- Mail Transfer Phases: - The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination.

## **Message Access Agent: POP and IMAP**

- The first and the second stages of mail delivery use SMTP. However, SMTP is not involved in the third stage because SMTP is a push protocol; it pushes the message from the client to the server.
- On the other hand, the third stage needs a pull protocol; the client must pull messages from the server.
- The third stage uses a message access agent. Currently two message access protocols are available: Post Office Protocol, version 3 (POP3) and Internet Mail Access Protocol, version 4 (IMAP4).

### **POP3**

- Post Office Protocol, version 3 (POP3) is simple and limited in functionality. The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.
- Mail access starts with the client when the user needs to download e-mail from the mailbox on the mail server. The client opens a connection to the server on TCP port 110. It then sends its user name and password to access the mailbox.
- The user can then list and retrieve the mail messages, one by one. POP3 has two modes: the delete mode and the keep mode. In the delete mode, the mail is deleted from the mailbox after each retrieval. In the keep mode, the mail remains in the mailbox after retrieval.
- The delete mode is normally used when the user is working at her permanent computer and can save and organize the received mail after reading or replying.
- The keep mode is normally used when the user accesses her mail away from her primary computer (e.g., a laptop). The mail is read but kept in the system for later retrieval and organizing.

## **IMAP4**

- Another mail access protocol is Internet Mail Access Protocol, version 4 (IMAP4). IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex.
- POP3 is deficient in several ways. It does not allow the user to organize her mail on the server; the user cannot have different folders on the server. (Of course, the user can create folders on her own computer.)
- In addition, POP3 does not allow the user to partially check the contents of the mail before downloading. IMAP4 provides the following extra functions:
  1. A user can check the e-mail header prior to downloading.
  2. A user can search the contents of the e-mail for a specific string of characters prior to downloading.
  3. A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
  4. A user can create, delete, or rename mailboxes on the mail server.
  5. A user can create a hierarchy of mail boxes in a folder for e-mail storage.

## **Web-Based Mail**

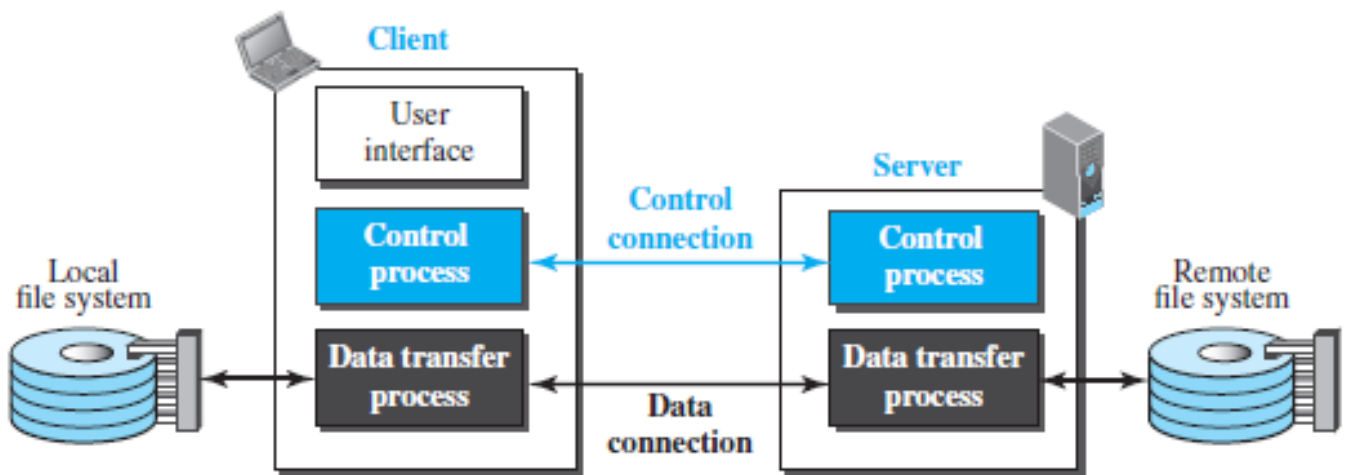
- E-mail is such a common application that some websites today provide this service to anyone who accesses the site. Two common sites are Hotmail and Yahoo.
- The idea is very simple. Mail transfer from Alice's browser to her mail server is done through HTTP.
- The transfer of the message from the sending mail server to the receiving mail server is still through SMTP.
- Finally, the message from the receiving server (the Web server) to Bob's browser is done through HTTP.
- The last phase is very interesting. Instead of POP3 or IMAP4, HTTP is normally used. When Bob needs to retrieve his e-mails, he sends a message to the website (Hotmail, for example).
- The website sends a form to be filled in by Bob, which includes the log-in name and the password. If the log-in name and password match, the e-mail is transferred from the Web server to Bob's browser in HTML format.

## FILE TRANSFER

- Transferring files from one computer to another is one of the most common tasks expected from a networking or internetworking environment. As a matter of fact, the greatest volume of data exchange in the Internet today is due to file transfer.

### File Transfer Protocol (FTP)

- It is used for exchanging files over the internet and enables the users to upload and download the files from the internet.
- File Transfer Protocol (FTP) is the standard mechanism provided by TCP/IP for copying a file from one host to another. Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first. For example, two systems may use different file name conventions.
- Two systems may have different ways to represent text and data. Two systems may have different directory structures.
- All these problems have been solved by FTP in a very simple and elegant approach. FTP differs from other client/server applications in that it establishes two connections between the hosts.
- One connection is used for data transfer, the other for control information (commands and responses). Separation of commands and data transfer makes FTP more efficient.
- The control connection uses very simple rules of communication. We need to transfer only a line of command or a line of response at a time.
- The data connection, on the other hand, needs more complex rules due to the variety of data types transferred. However, the difference in complexity is at the FTP level, not TCP. For TCP, both connections are treated the same.
- FTP uses two well-known TCP ports: Port 21 is used for the control connection, and port 20 is used for the data connection.
- The control connection remains connected during the entire interactive FTP session.
- The data connection is opened and then closed for each file transfer activity.



### Security for FTP

- The FTP protocol was designed when security was not a big issue.
- Although FTP requires a password, the password is sent in plaintext (unencrypted), which means it can be intercepted and used by an attacker.
- The data transfer connection also transfers data in plaintext, which is insecure.
- To be secure, one can add a Secure Socket Layer between the FTP application layer and the TCP layer. In this case FTP is called SSL-FTP.

**Q** Which of the following protocol pairs can be used to send and retrieve e-mails (in that order)? **(Gate-2019) (1 Marks)**

**(a)** SMTP, MIME

**(b)** SMTP, POP3

**(c)** IMAP, POP3

**(d)** IMAP, SMTP

Ans: b

**Q** Which of the following transport layer protocols is used to support electronic mail? **(Gate-2012) (1 Marks)**

**(A)** SMTP

**(B)** IP

**(C)** TCP

**(D)** UDP

**Answer: (C)**

**Q** In one of the pairs of protocols given below, both the protocols can use multiple TCP connections between the same client and the server. Which one is that? **(GATE-2015) (1 Marks)**

**(A)** HTTP, FTP

**(B)** HTTP, TELNET

**(C)** FTP, SMTP

**(D)** HTTP, SMTP

**Answer: (A)**

**Q** The transport layer protocols used for real time multimedia, file transfer, DNS and email, respectively are: **(GATE-2013) (1 Marks)**

**(A)** TCP, UDP, UDP and TCP

**(B)** UDP, TCP, TCP and UDP

**(C)** UDP, TCP, UDP and TCP

**(D)** TCP, UDP, TCP and UDP

**Answer: (C)**

**Q** Consider different activities related to email:

**m1:** Send an email from a mail client to a mail server

**m2:** Download an email from mailbox server to a mail client

**m3:** Checking email in a web browser

Which is the application level protocol used in each activity? **(GATE-2011) (1 Marks)**

**(A)** m1: HTTP m2: SMTP m3: POP

**(B)** m1: SMTP m2: FTP m3: HTTP

**(C)** m1: SMTP m2: POP m3: HTTP

**(D)** m1: POP m2: SMTP m3: IMAP

**Answer: (C)**

**Q** Consider the following clauses:

**i.** Not inherently suitable for client authentication.

**ii.** Not a state sensitive protocol.

**iii.** Must be operated with more than one server.

**iv.** Suitable for structured message organization.

**v.** May need two ports on the serve side for proper operation.

The option that has the maximum number of correct matches is **(GATE-2007) (2 Marks)**

**(A)** IMAP-(i), FTP-(ii), HTTP-(iii), DNS-(iv), POP3-(v)

**(B)** FTP-(i), POP3-(ii), SMTP-(iii), HTTP-(iv), IMAP-(v)

**(C)** POP3-(i), SMTP-(ii), DNS-(iii), IMAP-(iv), HTTP-(v)

**(D) SMTP-(i), HTTP-(ii), IMAP-(iii), DNS-(iv), FTP-(v)**

**Answer: (D)**

**Q Which one of the following uses UDP as the transport protocol? (GATE-2007) (1 Marks)**

**(A) HTTP**

**(B) Telnet**

**(C) DNS**

**(D) SMTP**

**Answer: (C)**

**Q Match the following: (GATE-2007) (2 Marks)**

(P) SMTP	(1) Application layer
(Q) BGP	(2) Transport layer
(R) TCP	(3) Data link layer
(S) PPP	(4) Network layer
	(5) Physical layer

**(A) P – 2 Q – 1 R – 3 S – 5**

**(B) P – 1 Q – 4 R – 2 S – 3**

**(C) P – 1 Q – 4 R – 2 S – 5**

**(D) P – 2 Q – 4 R – 1 S – 3**

**Answer: (B)**

**Q which of the following is not a client -server application? (GATE-2010) (1 Marks)**

**a) Internet chat**

**b) Web browsing**

**c) E-mail**

**d) Ping**

**Q The protocol data unit (PDU) for the application layer in the Internet stack is (GATE-2012) (1 Marks)**

**(A) Segment**

**(B) Datagram**

**(C) Message**

**(D) Frame**

**Answer: (C)**