# SYLLABUS

**Unit 1:** **Hrs: 06**

Layered Tasks, OSI model, Layers in OSI model, TCP/IP Suite, Addressing.
Telephone and cable networks for data transmission, Telephone Networks, Dial up modem, DSL, Cable TV for data transmission.

**Unit 2: Data Link control** **Hrs: 07**

Framing, Flow & Error control, Protocols, Noiseless channels & Noisy channels, HDLC.

**Unit 3: Multiple Accesses** **Hrs: 06**

Random access, Controlled access, channelization

**Unit 4:** **Hrs: 07**

IEEE standards, standard Ethernet, changes in the standards, Fast Ethernet, Gigabit Ethernet, Wireless LAN IEEE 802.11

**Unit 5:** **Hrs: 06**

Connecting LANs, Backbone and virtual LANs, Connecting devices, Backbone networks, Virtual LANs.

**Unit 6:** **Hrs: 07**

Network layer, Logical addressing, IPv4 addresses, IPv6 addresses, IPv4 and IPv6 transition from IPv4 to IPv6.

**Unit 7:** **Hrs: 06**

Delivery, Forwarding, Unicast Routing protocols, Multicast Routing protocols

**Unit 8:** **Hrs: 06**

Transport layer process to process delivery, UDP, TCP, Domain name system, Resolution.

**Prescribed & Reference Books:**

| Sl. No. | Particulars | Book Title | Book Author | Book Publications |
|---|---|---|---|---|
| 1 | Prescribed Books | Data Communication & Networking | B Forouzan | 4th Ed, 2006, TMH |
| 2 | Reference Books | Computer Networks | James F. Kursoe, Keith W. Ross | 2nd Ed, 2003, Pearson |
| 3 | | Introduction to Data communication & Networking | Wayne Tomasi | 2007, Pearson |

# DATA COMMUNICATIONS INTRODUCTION

Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable. For data communications to occur, the communicating devices must be part of a communication system made up of a combination of hardware (physical equipment) and software (programs). The effectiveness of a data communications system depends on four fundamental characteristics: delivery, accuracy, timeliness, and jitter.

Delivery. The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user.

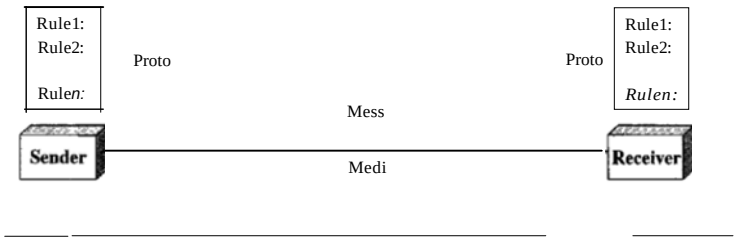
Accuracy. The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable.

Timeliness. The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called real-time transmission.

Jitter. Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets.

COMPONENTS

A data communications system has five components




Message:The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.

Sender. The sender is the device that sends the data message. It can be a computer,

workstation, telephone handset, video camera, and so on.

Receiver. The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.

Transmission medium. The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves.

Protocol. A protocol is a set of rules that govern data communications.
sents an agreement between the communicating devices.

## DATA REPRESENTATION:

Information today comes in different forms such as text, numbers, images, audio, and video.

Text
In data communications, text is represented as a bit pattern, a sequence of bits (Os or Is). Different sets of bit patterns have been designed to represent text symbols. Each set is called a code, and the process of representing symbols is called coding.
Numbers
Numbers are also represented by bit patterns. However, a code such as ASCII is not used to represent numbers; the number is directly converted to a binary number to simplify mathematical operations.

Images
Images are also represented by bit patterns. In its simplest form, an image is composed of a matrix of pixels (picture elements), where each pixel is a small dot. The size of the pixel depends on the resolution
Audio
Audio refers to the recording or broadcasting of sound or music. Audio is by nature different from text, numbers, or images. It is continuous, not discrete.

Video

Video refers to the recording or broadcasting of a picture or movie. Video can either be produced as a continuous entity (e.g., by a TV camera), or it can be a combination of images, each a discrete entity, arranged to convey the idea of motion.

## 1.1 Layered Tasks:

Consider two friends who communicate through mail

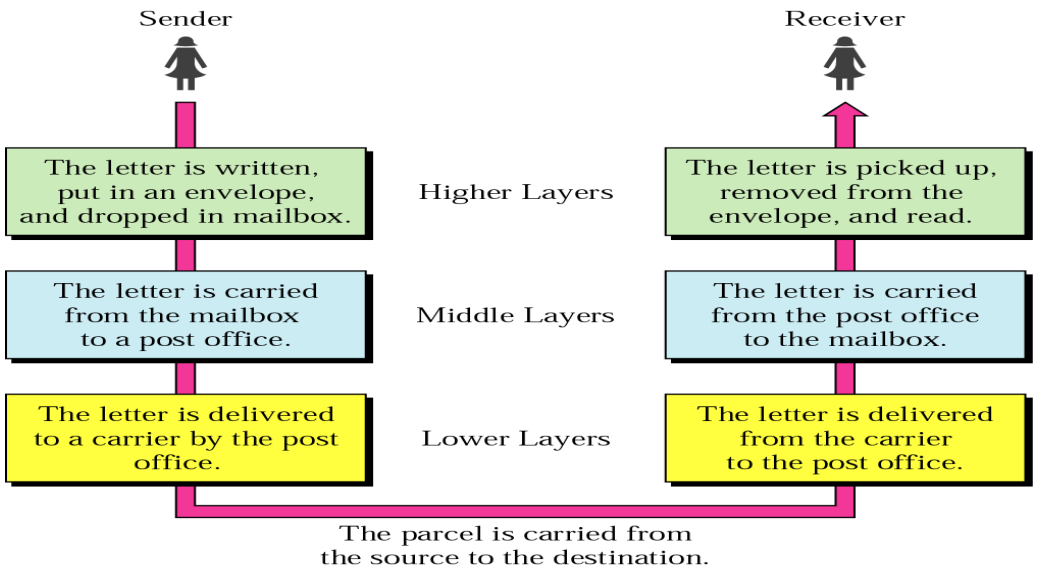



Fig (i): Tasks involved in sending a letter

In the above fig (i), we have a sender, a receiver & a carrier that transports the letter. There is a hierarchy of tasks.

At the sender site:

- The sender writes the letter, inserts the letter in an envelope, writes the sender & receiver addresses, and drops the letter in a mail box.
- Middle layer: The letter is picked up by a letter carrier and delivered to the post office.
- Lower Layer: The letter is sorted at the post office, a carrier transports the letter.

On the way:

The letter is then on its way to the recipient. On the way to the recipients local post office, the letter may actually go through a central office. In addition, it may be transported by truck, train, airplane, boat, or a combination of these.

At the receiver site:

- Lower layer: The carrier transports the letter to the post office.
- Middle Layer: The letter is sorted & delivered to the recipients mailbox.
- Higher Layer: The receiver picks up the letter, opens the envelope, and reads it.

Hierarchy:
The task of transporting the letter between the sender and the receiver is done by the carrier. The tasks must be done in the order given in the hierarchy. At the sender site, the letter must be written and dropped in the mailbox before being picked up by the letter carrier and delivered to the post office. At the receiver site, the letter must be dropped in the recipient mailbox before being picked up & read by the recipient.
Services:
Each layer at the sending site uses the services of the layer immediately below it. The sender at the higher layer uses the services of the middle layer. The middle layer uses the services of the lower layer. The lower layer uses the services of the carrier.

## 1.2 The OSI model:

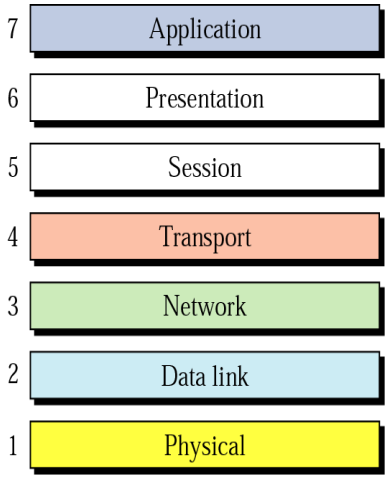



Fig (ii): Seven layers of the OSI model

The OSI model shown in fig (ii) is based on the proposal developed by the International Standards Organization (ISO) as a first step towards international standardization of the protocols used in the various layers. The model is called the OSI (Open System Interconnection) reference model because it deals with connecting open systems, i.e., systems that are open for communication with other systems. 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.

The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network. The principles that were applied to arrive at the seven layers are as follows:

* A layer should be created where a different level of abstraction is needed.
* Each layer should perform a well-defined function.
* The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
* The layer boundaries should be chosen to minimize the information flow across the interfaces.
* The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy.

## Layered Architecture:

The OSI model is composed of seven layers: Physical, Data link, Network, Transport, Session, Presentation, Application layers. Fig (iii) shows the layers involved when a message travels from A to B, it may pass through many intermediate nodes. These intermediate nodes involve only the first 3 layers of the OSI model.

Within a single machine, each layer calls upon the services of the layer just below it, layer 3 for ex. Uses the services provided by layer 2 & 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 & Conventions called protocols. The processes on each machine that communicate at a given layer are called peer – to – peer processes. Communication between machines is therefore a peer – to –peer process using the protocols appropriate to a given layer.

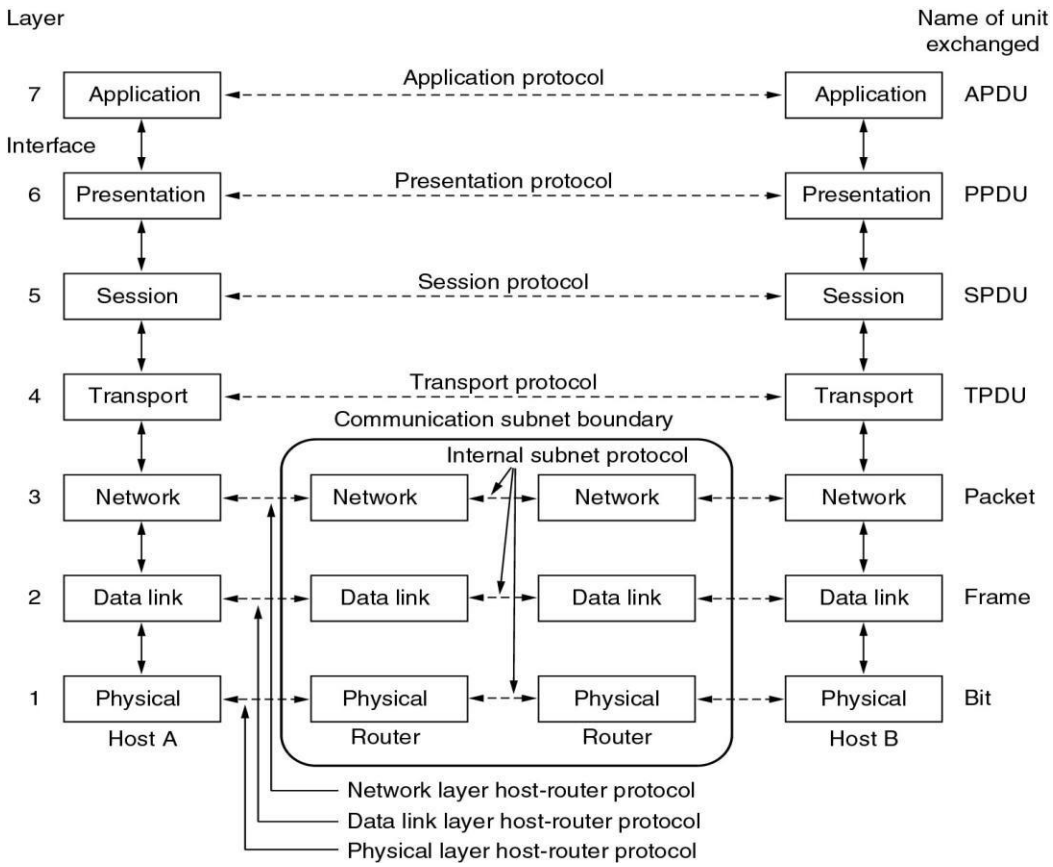


Fig (iii): Interaction between layers in the OSI model

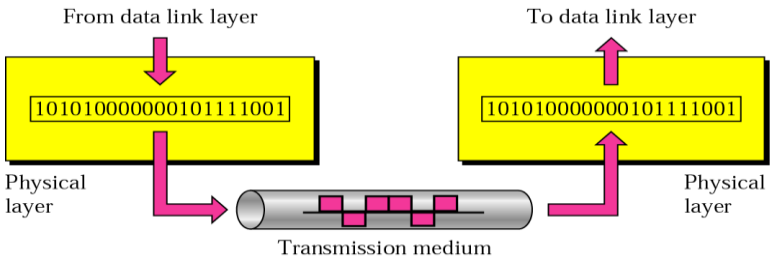
## 1.33: Layers in the OSI model:

i) Physical Layer:

### Physical Layer

The physical layer coordinates the functions required to carry a bit stream over a physi- cal medium. It deals with the mechanical and electrical specifications of the interface and transmission medium. It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to Occur. Figure shows the position of the physical layer with respect to the transmission medium and the data link layer.

*Physical Layer is responsible for movements of individual bits from one node to the next*




The physical layer is also concerned with the following:

**Physical characteristics of interfaces and medium**. The physical layer defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.

**Representation of bits**. The physical layer data consists of a stream of bits (sequence of Os or 1s) with no interpretation. To be transmitted, bits must be
encoded into signals--electrical or optical. The physical layer defines the type of encoding (how Os and Is are changed to signals).

**Data rate**. The transmission rate-the number of bits sent each second-is also
defined by the physical layer. In other words, the physical layer defines the duration of a bit, which is how long it lasts.

**Synchronization of bits.** The sender and receiver not only must use the same bit rate but also must be synchronized at the bit level. In other words, the sender and the receiver clocks must be synchronized.

**Line configuration**. The physical layer is concerned with the connection of

devices to the media. In a point-to-point configuration, two devices are connected through a dedicated link. In a multipoint configuration, a link is shared among several devices.

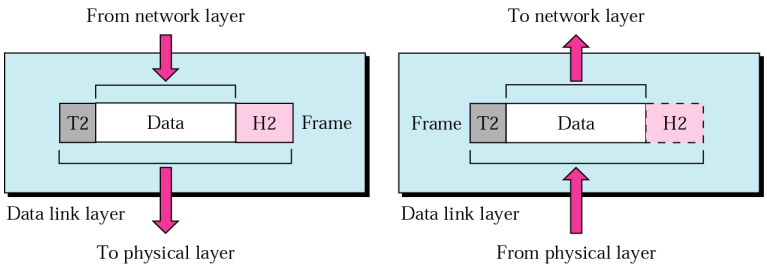
**Physical topology.** The physical topology defines how devices are connected to make a network. Devices can be connected by using a mesh topology (every device is connected to every other device), a star topology (devices are connected through a central device), a ring topology (each device is connected to the next, forming a ring), a bus topology (every device is on a common link), or a hybrid topology (this is a combination of two or more topologies).

**Transmission mode**. The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex. In simplex mode, only one device can send; the other can only receive. The simplex mode is a one-way communication. In the half-duplex two devices can send and receive, but not at the same time. In a full-duplex (or simply duplex) mode, two devices can send and receive at the same time.

ii) 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). Figure shows the relationship of the data link layer to the network and physical layers.

*Data link layer is responsible for moving frames from one node to the next.*




Other responsibilities of the data link layer include the following:
**Framing.** The data link layer divides the stream of bits received from the networklayer 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. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects the network to the next one.

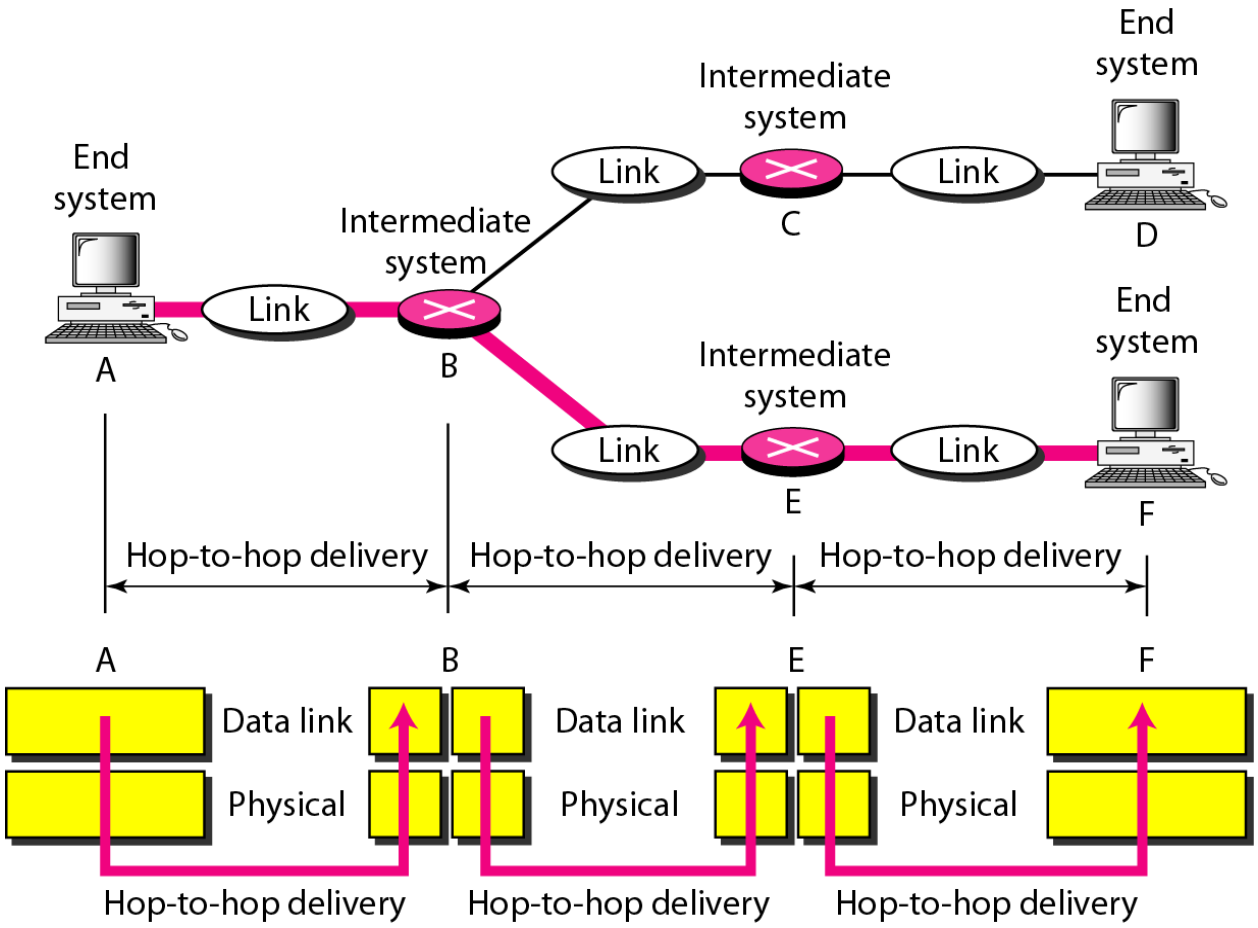
Flow control. If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid

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.

Figure illustrates hop-to-hop (node-to-node) delivery by the data link layer.

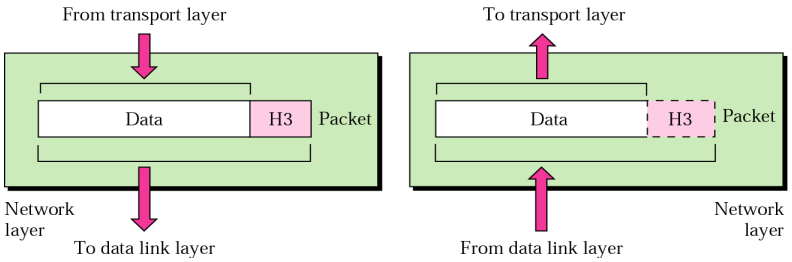



iii) Network Layer:

The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links). Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links), the network layer ensures that each packet gets from its point of origin to its final destination.

If two systems are connected to the same link, there is usually no need for a network layer. However, if the two systems are attached to different networks (links) with connecting devices between the networks (links), there is often a need for the network layer to accomplish source-to-destination delivery. Figure  shows the relationship of the network layer to the data link and transport layers.

*Network layer is responsible for the delivery of individual packets from the source host to the destination host.*



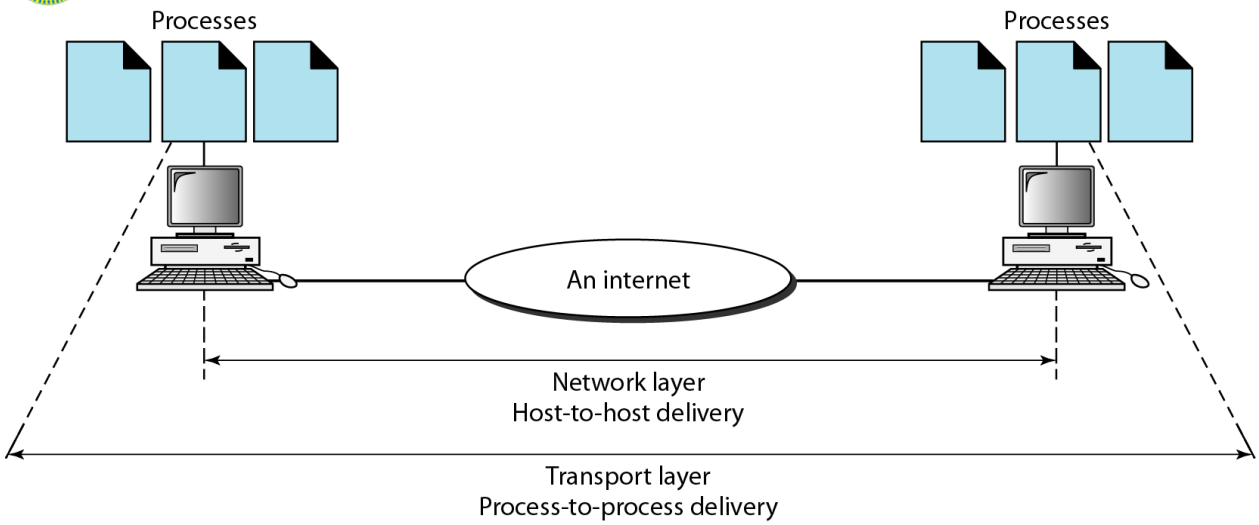

Other responsibilities of the network layer include the following:

**Logical addressing.** The physical addressing implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems. The network layer adds a header to the packet coming from the upper layer that, among other things, includes the logical addresses of the sender and receiver.

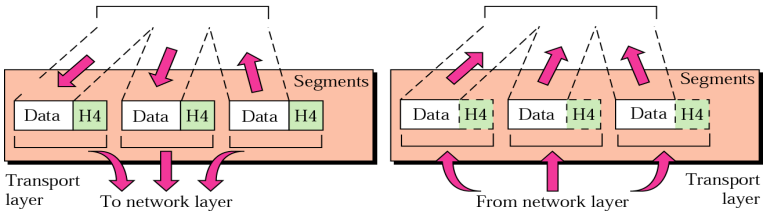
**Routing.** When independent networks or links are connected to create internetetworks(network of networks) or a large network, the connecting devices (called routers)

iv) 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. It treats each one independently, as though each piece belonged to a separate message, whether or not it does. The transport layer, on the other hand, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level. Figure shows the relationship of the transport layer to the network and session layers




*Transport layer is responsible for the delivery of a message from one process to another.*




**Service-point addressing**: Computers often run several programs at the same time. For this reason, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other. The transport layer header must therefore 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.

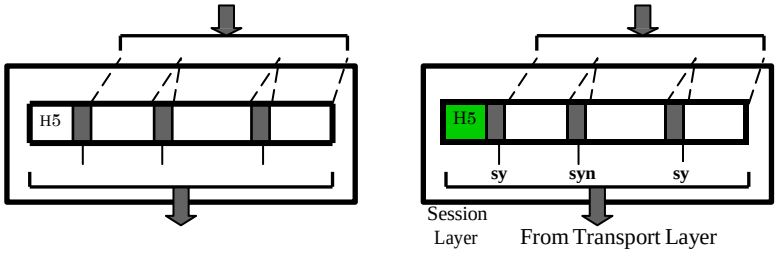
**Flow control:** Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather thanacross a single link.

**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.

v) Session Layer:

*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. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent. Figure  illustrates the relationship of the session layer to the transport and presentation layers

vi) Presentation Layer:

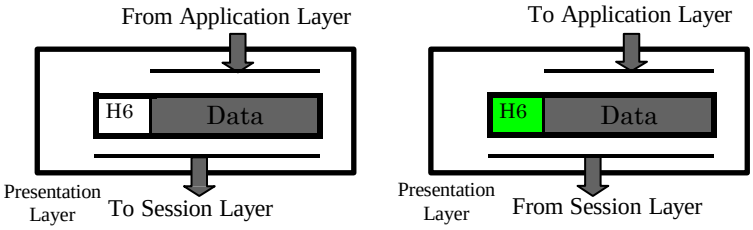
**Translation:** The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on. The information must be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation 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.

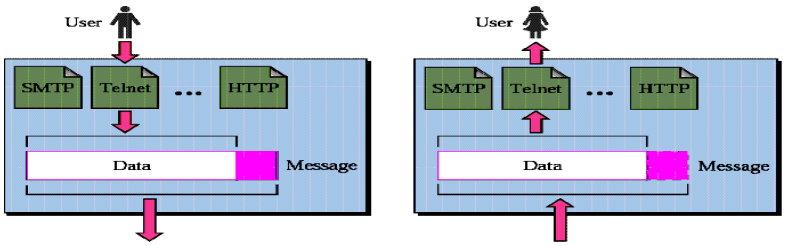
*The Presentation layer is responsible for translation, compression, and encryption.*



- Concerned with syntax and semantics of the information.
- Translation.
- Encryption.
- Compression.

vii) Application Layer:

*The Application layer is responsible for providing services to the user.*




- Network virtual terminal.
- File transfer, access, and management.
- Mail services.
- Directory 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.

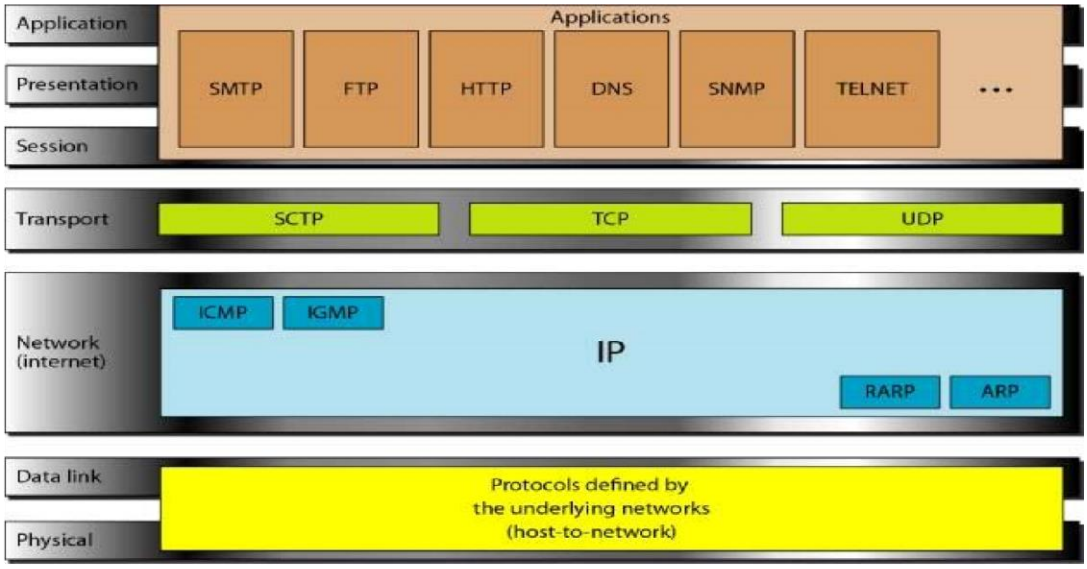



Fig 1.7 TCP/IP Model Vs OSI Model

**A) APPLICATION LAYER**

Application layer protocols define the rules when implementing specific network applications. It relies on the underlying layers to provide accurate and efficient data delivery. Typical protocols are:

- FTP – File Transfer Protocol: For file transfer
- Telnet – Remote terminal protocol: For remote login on any other computer on the network
- SMTP – Simple Mail Transfer Protocol: For mail transfer
- HTTP – Hypertext Transfer Protocol: For Web browsing

**B) TRANSPORT LAYER**

Transport Layer protocols define the rules of dividing a chunk of data into segments and then reassemble segments into the original chunk. Typical protocols are:TCP – Transmission Control Protocol: Provide functions such as reordering and data resend.

- UDP – User Datagram Service: Use when the message to be sent fit exactly into a datagram and Use also when a more simplified data format is required.
- SCTP - Stream Control Transmission Protocol: The Stream Control Transmission Protocol (SCTP) provides support for newer applications such as voice over the Internet.

## 1.4 TCP/IP Protocol suite:

**C) NETWORK LAYER**

Network layer protocols define the rules of how to find the routes for a packet to the destination It only gives best effort delivery. Packets can be delayed, corrupted, lost, duplicated, out-of-order.

- IP – Internet Protocol: Provide packet delivery
- ARP – Address Resolution Protocol: Define the procedures of network address / MAC address translation i.e The Address Resolution Protocol (ARP) is used to associate a logical address with a physical address. ARP is used to find the physical address of the node when its Internet address is known.
- RARP – Reverse Address Resolution Protocol: The Reverse Address Resolution Protocol (RARP) allows a host to discover its Internet address when it knows only its physical address.
- ICMP – Internet Control Message Protocol: The Internet Control Message Protocol (ICMP) is a mechanism used by hosts and gateways to send notification of datagram problems back to the sender.
- IGMP – Internet Control Message Protocol: The Internet Group Message Protocol (IGMP) is used to facilitate the simultaneous transmission of a message to a group of recipients.

**D) PHYSICAL AND DATA LINK LAYER**

At the physical and data link layers, TCP-IP does not define any specific protocol. Rather, it supports all the standard protocols.

**1.6 ADDRESSING**

There are four levels of addresses are used in an internet employing the TCP/IP protocols: physical, logical, port, and specific. Figure 1.8 below shows relationship of layers and addresses in TCP/IP:

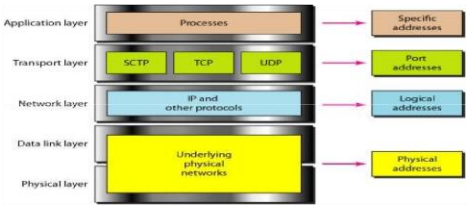



Fig 1.8 TCP/IP Addressing

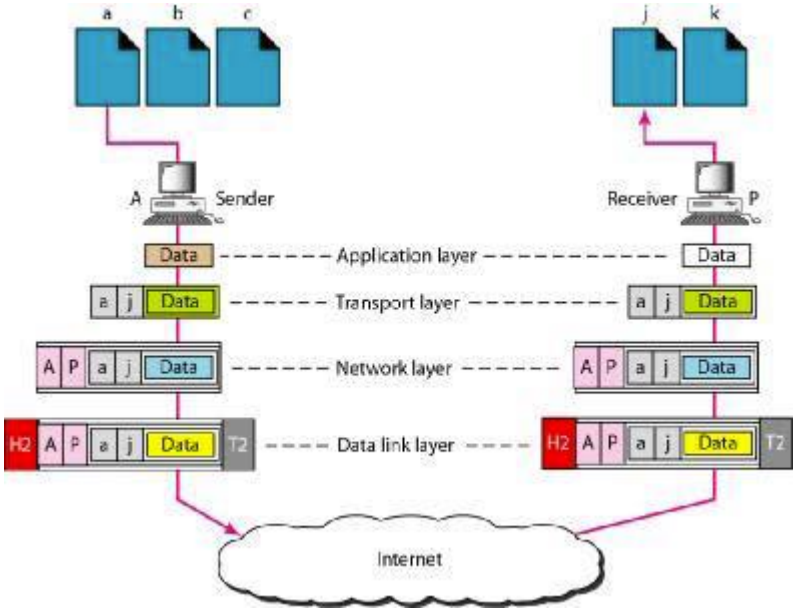
The TCP/IP protocol suite has four layers: Host – to – Network, Internet, Transport and Application. Comparing TCP/IP to OSI model: the Host – to – Network layer is equivalent to the combination of physical and data link layers, the Internet layer is equivalent to the network layer, the Transport layer in TCP/IP taking care of part of the duties of the session layer, and the application layer is roughly doing the job of the session, presentation, & application layers.

Four levels of addresses are used in an internet employing the TCP/IP Protocols:

i) Physical addresses
ii) Logical addresses
iii) Port addresses
iv) Specific addresses

The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet. A system that sends nothing but data from one computer to another

is not complete. Today, computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating with another process. For example, computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP). For these processes to receive data simultaneously, we need a method to label the different processes .In other words, they need addresses. In the TCPIIP architecture, the label assigned to a process is called a port address. A port address in TCPIIP is 16 bits in length.




Each address is related to a specific layer in the TCP/IP architecture, as shown in the above fig.

# UNIT - I

**1.** a) Explain briefly with relevant examples the four levels of addresses that are used in an internet employing TCP/IP protocols **(July 2013/2011 10 marks)**

b) Briefly describe the function of physical layer and data link layer. **(July 2013 06 marks)**

c) Explain the operation of ADSL using 'Discrete Multitone Technique' indicating the different channels with a diagram **(July 2013/2011 04 marks)**

**2.** a) Describe ISO OSI reference model of a computer Network. Discuss the function of each layer **(Jan 2013 10marks)**

b) Describe SS7 service and its relation to telephone network **(Jan 2013 05marks)**

c)Distinguish between DSL modem and a DSLAM **(Jan 2013 05marks)**

**3.** a) With a neat diagram, explain TCP/IP reference model **(July 2012 10marks)**

b) Explain in detail, the cable TV network used for data transfer **(July 2012 06marks)**

c) Differentiate between CM and CMTS **(July 2011 04 marks)**

**4.** a) Calculate the minimum time to download the one million bytes of information using each of the following technologies:

i)V.32 modem

ii) V.32 bits modem

iii) V.90 modem **(July 2012 10marks)**

**b)** Explain the different services provided by telephone networks **(Dec 2014 4 marks )**

**c)** What are the different types of services provided by telephone network **(Dec 2010 06 marks)**

**5.** a)Explain the differences between OSI reference model and TCP/IP reference model **(Dec 2011 05 marks)**

b)Match the following to one or more layers in OSI model:

i) Route determination

ii) Flow control

iii) Interface to transmission media

iv) Provides access for the end user

v) Format and code conversion services **(Dec 2011 05 marks)**

**c)**What is DSL technology? What are the services provided by the telephone companies using DSL? Distinguish between DSL and DSLAM. **(Dec 2011 10 marks)**

**6.** a)What are the levels of addresses that are used in an internet, employing the TCP/IP protocols? **(Dec 2010 10 marks)**

b)What are the different types of services provided by telephone network **(Dec 2010 06 marks)**

c)Name the major components of a telephone network **(Dec 2010 04 marks)**

**7.** a) Discuss the TCP/IP model with functionalities of each layer .Consider source destination and intermediate nodes for discussion. **(Dec 2014 10 marks )**

b)How addresses employed in internet employing TCP/IP protocol can be classified **(June 2010 02 marks)**

c)What is DSL technology? List different DSLs available. Discuss salient features of ADSL **(June 2010 08 marks)**

**8.** **a)** a)Explain OSI model, with a neat block diagram. Consider a source, destination machine and some intermediate nodes for discussion **(Dec 2010 10 marks)**

**b)** Differentiate between CM and CMTS **(July 2011 10 marks)**

**9.** **a)** Describe the layer presentation in the TCP/IP model and explain the protocols of each layer **(June 2014 10marks)**

b) What is ADSL? Explain the operation of ADSL using "Discrete multitone technique" indicating the different channels with a diagram **(June 2014 10marks )**

**10.** a)List diagram types of addressing explain any one type of addressing with suitable examples **(June 2014 10marks )**

**b)** Describe four levels of addressing used in internet TCP/IP with examples **(Dec 2014 10 marks)**

**UNIT II**

## 2.1 Framing:

Framing in the data link layer separates a message from one source to a destination by adding a sender address & a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt. 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 & 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.

Frames can be of fixed or variable size.

i) Fixed-size framing:

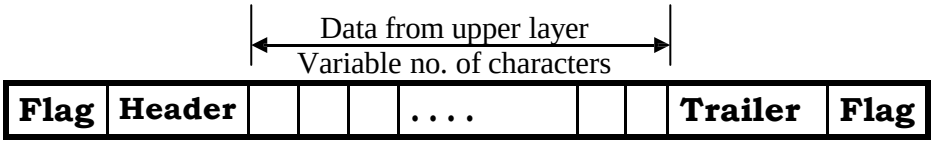
In this there is no need for defining the boundaries of the frames, the size itself can be used as a delimiter.
Ex: ATM wide area network which uses frames of fixed size called cells.

ii) Variable-size framing:

In this, we need a way to define the end of the frame and the beginning of the next.
Ex: LAN

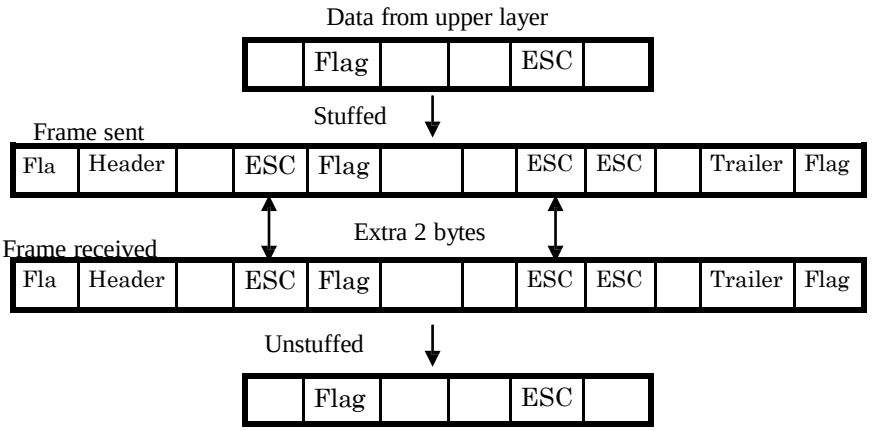
Historically 2 approaches were used for variable size framing: Character-oriented & bit-oriented.

a) Character-oriented approach:

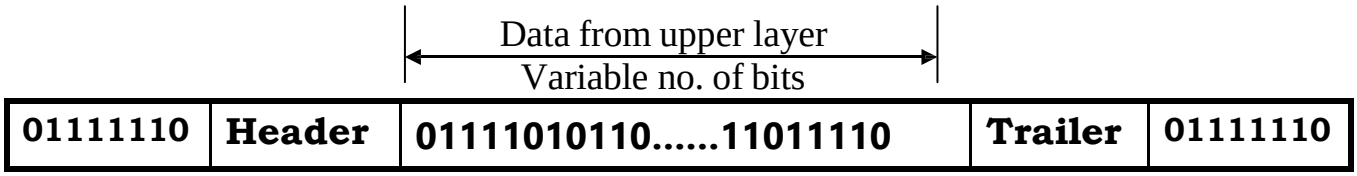



*SOF and EOF-01111110 and 01111110*

*Character oriented protocol – Byte Stuffing & De-stuffing*

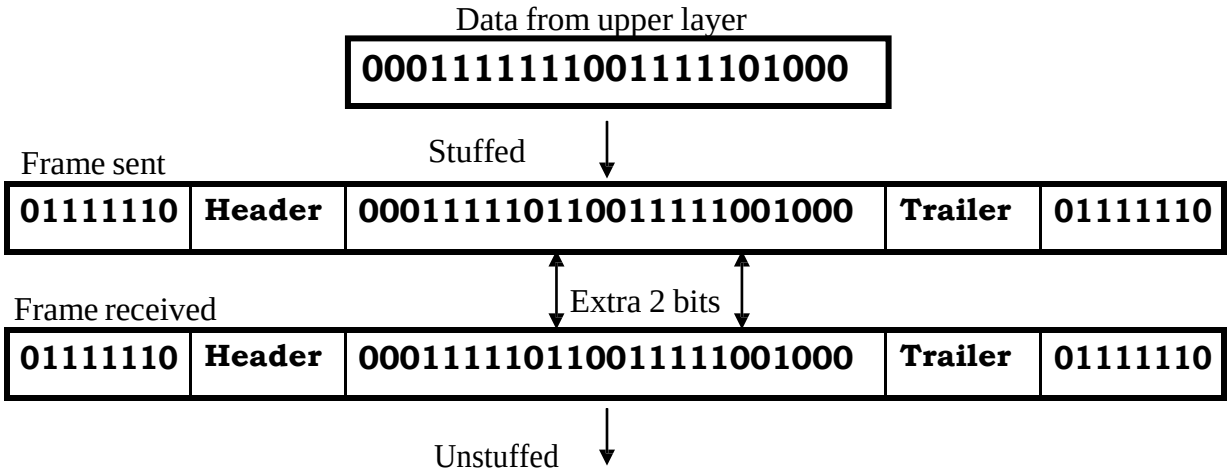



Data to upper layer




***Bit oriented protocol – Bit Stuffing & De-stuffing***

*Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver doesn't mistake the pattern 01111110 for a flag*



**0001111111001111101000**

Data to upper layer

## 2.2 Flow & Error Control:

### Flow Control:

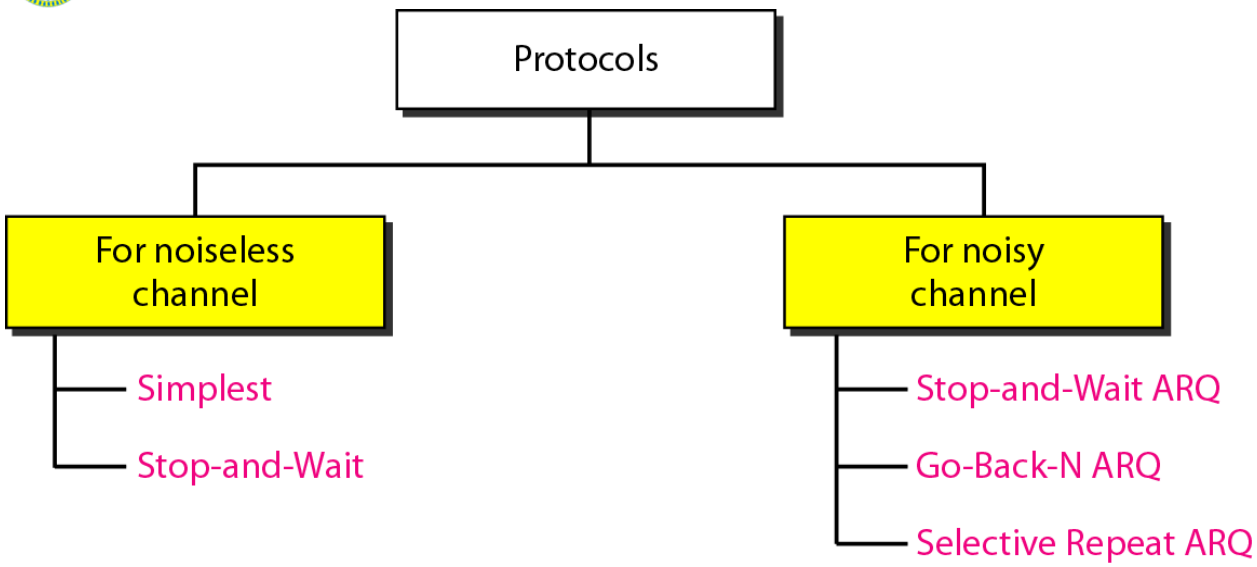
- Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data.
- The flow of data must not be allowed to overwhelm the receiver.
- The receiver must be able to tell the sender to halt transmission until it is once again able to receive.
- Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgement.

### Error Control:

- Errors occur due to noises in the channel.
- Error control is both error detection & error correction.
- In the data link layer error control refers to methods of error detection and retransmission.
- To sender should add certain amount of redundant bits to the data, based on which the receiver will be able to detect errors.

## 2.3 PROTOCOLS:

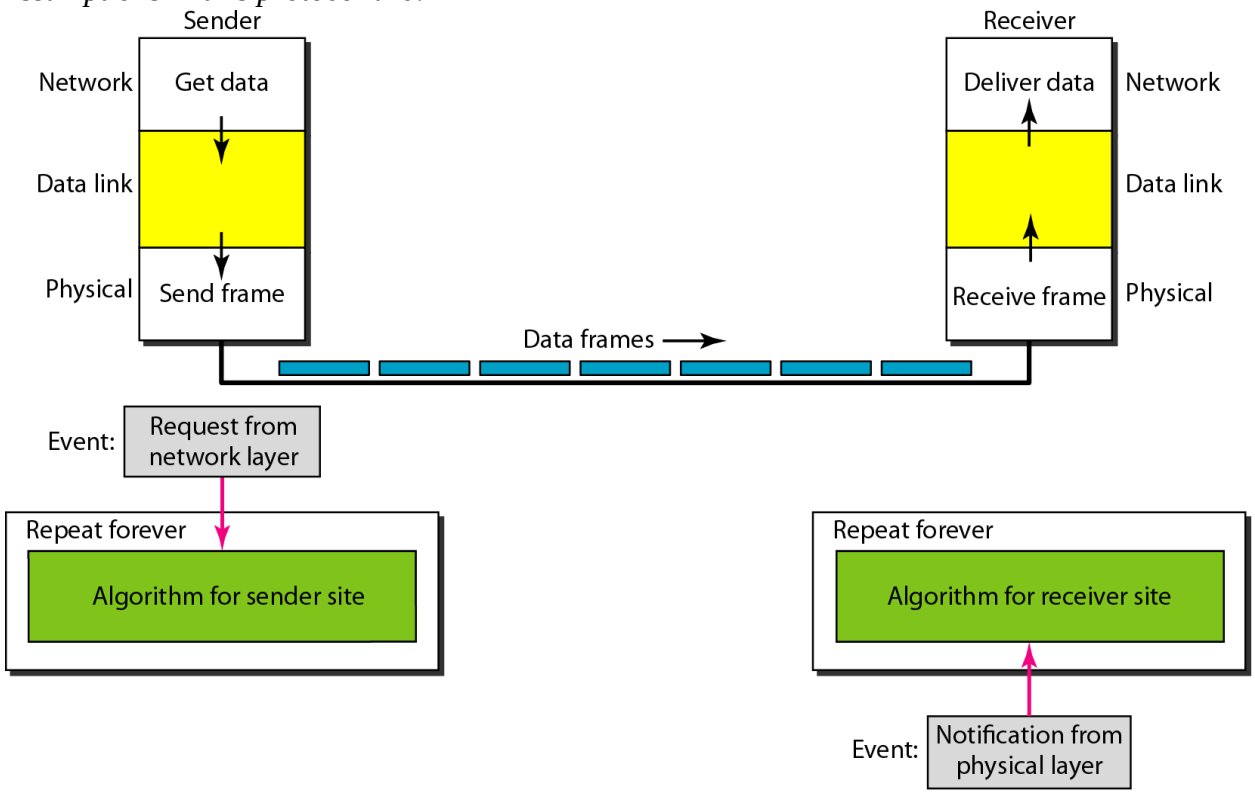
- A protocol is a set of rules that govern data communication.
- A protocol defines what, how it is communicated, and when it is communicated.
- The key elements of a protocol are syntax, semantics & timing.
- Syntax refers to the structure or format of the data, i.e., the order in which they are presented.
- Semantics refers to the meaning of each section of bits.
- Timing refers to when data should be sent & how fast they can be sent.
- Protocols are implemented in software by using any of the common programming languages.




## 2.4 Noiseless channels:

### i) Simplest protocol:

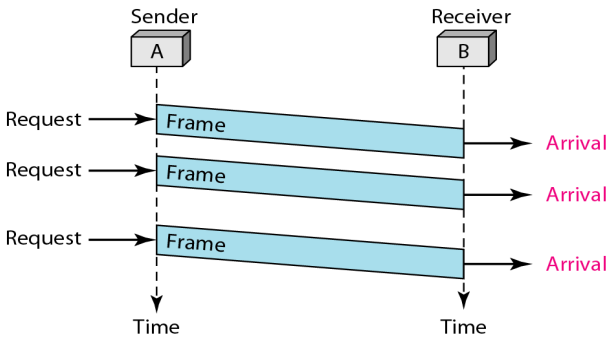
*Assumptions in this protocol are:*



```
1  while(true)                          // Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(RequestToSend))           //There is a packet to send
5    {
6       GetData();
7       MakeFrame();
8       SendFrame();                    //Send the frame
9    }
10 }
```

```
1  while(true)                          // Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(ArrivalNotification))     //Data frame arrived
5    {
6       ReceiveFrame();
7       ExtractData();
8       DeliverData();                  //Deliver data to network layer
9    }
10 }
```

Figure shows an example of communication using this protocol. It is very simple. The sender sends a sequence of frames without even thinking about the receiver. To send three frames, three events occur at the sender site and three events at the receiver site. Note that the data frames are shown by tilted boxes; the height of the box defines the transmission time difference between the first bit and the last bit in the frame.



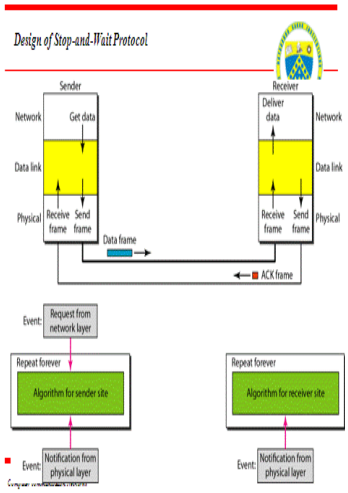

**For noisy channel**

**Stop-and-wait**

**Go-Back-N ARQ Selective repeat**

- Data transfer is unidirectional.
- Both sender & receiver network layers are always ready.
- Processing time can be ignored.
- Infinite buffer space is available.
- Frames are never damaged or lost.

***Design:***



ii) Stop - & - Wait Protocol:

*Assumptions in this protocol are:*

- Data transfer is unidirectional.
- Both sender & receiver network layers are always ready.
- Receiver doesn't have enough storage space.
- Receiver is slower than sender in processing.
- Frames are never damaged or lost.

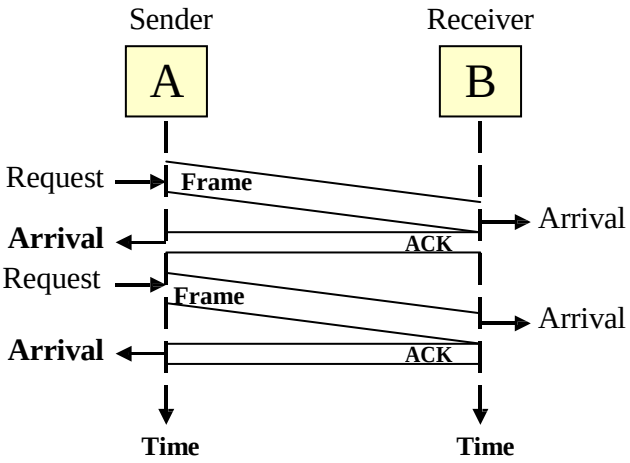
***Sender-site algorithm for the Stop-&-wait Protocol:***

```
While(true)                          //Repeat forever
Cansend = true                       //Allow the first frame to go
{
  waitForEvent();                          //Sleep until an event occurs
  if(event(requestTosend) AND cansend) //there is a packet to send
    {
      GetData();                           //get data from n/w layer
      MakeFrame();                         //make a frame
      SendFrame();                         //send the frame
      cansend = false;             //can't send until ACK arrives
    }
  waitForEvent();                          //Sleep until an event occurs
  if(Event(ArrivalNotification))//an ACK has arrived
   {
     ReceiveFrame();               //Receive the ACK frame
      cansend = true;
    }
}
```

***Receiver-site algorithm for the Stop-&-Wait Protocol:***

```
While(true)                          //Repeat forever
{
  waitForEvent();                          //Sleep until an event occurs
  if(event(ArrivalNotification)) //data frame arrived
    {
      ReceiveFrame();              //receive frame from the physical layer
      ExtractData();                       //extract data from a frame
      deliverData();                       //deliver the data to the n/w layer
      SendFrame();                         //Send an ACK frame
     }
}
```

***Flow diagram to illustrate Stop-&-Wait protocol:***




## 2.5 Noisy channels:

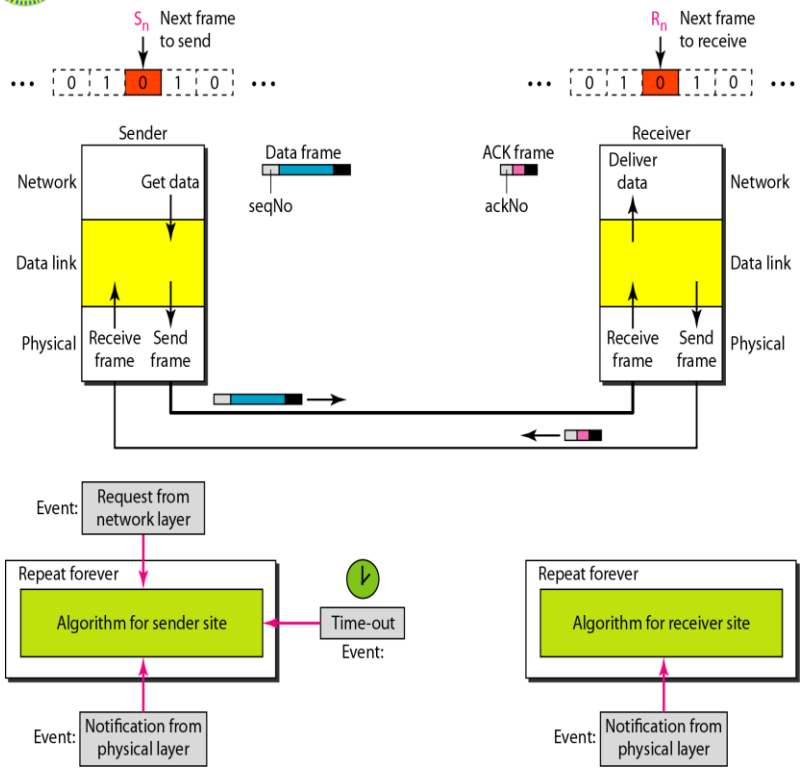
*Assumptions are:*

- Data transfer is unidirectional, but half-duplex link.
- Both sender & receiver network layers are always ready.
- Receiver doesn't have enough storage space.
- Receiver is slower than sender in processing.
- Frames are damaged or lost because of the noises in the channel.

i) Stop - & - Wait Automatic Repeat Request (ARQ):

- Adds a simple error control mechanism to the Stop-&-Wait protocol.
- Error detection & retransmission is used for error control.
- Sending device keeps a copy of the last frame transmitted until acknowledgement for that frame is received.
- Data frames uses Sequence numbers, to avoid duplication of frames.
- Range of sequence numbers = 2m – 1.
- ACK frame uses Acknowledgement number, & always announces the sequence number of the next frame expected.
- Timers are used in case of loss of ACK frames.

***Design:***

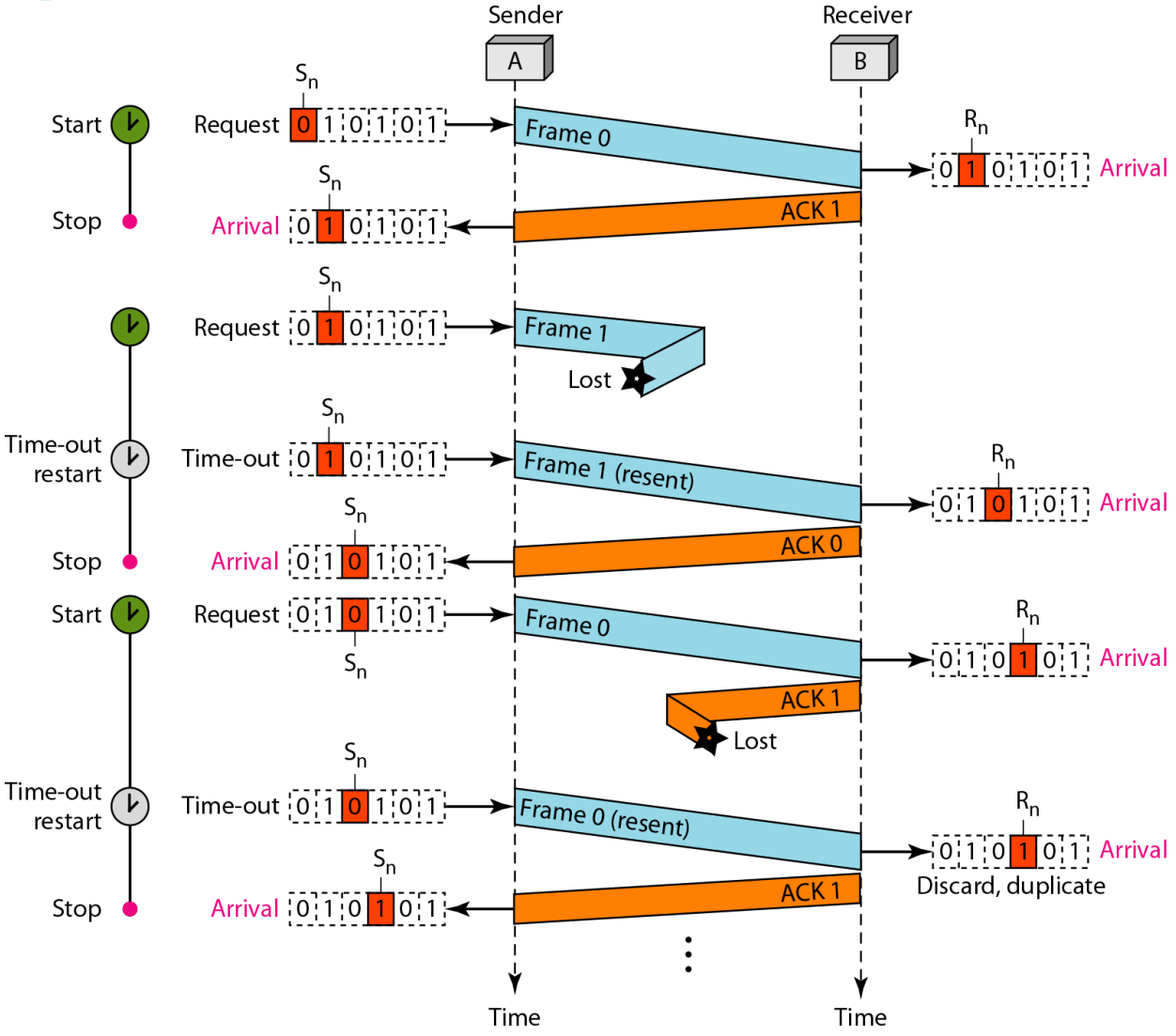



***Sender-site algorithm for Stop-and-Wait ARQ***

```
1  Sn = 0;                          // Frame 0 should be sent first
2  canSend = true;                  // Allow the first request to go
3  while(true)                      // Repeat forever
4  {
5    WaitForEvent();                // Sleep until an event occurs
6    if(Event(RequestToSend) AND canSend)
7    {
8       GetData();
9       MakeFrame(Sn);                    //The seqNo is Sn
10      StoreFrame(Sn);                   //Keep copy
11      SendFrame(Sn);
12      StartTimer();
13      Sn = Sn + 1;
14      canSend = false;
15   }
16   WaitForEvent();                      // Sleep
```

```
17      if(Event(ArrivalNotification)      // An ACK has arrived
18      {
19        ReceiveFrame(ackNo);             //Receive the ACK frame
20        if(not corrupted AND ackNo == Sn) //Valid ACK
21          {
22            Stoptimer();
23            PurgeFrame(Sn-1);            //Copy is not needed
24            canSend = true;
25          }
26       }
27
28      if(Event(TimeOut)                  // The timer expired
29      {
30       StartTimer();
31       ResendFrame(Sn-1);                //Resend a copy check
32      }
33  }
```

shows an example of Stop-and-Wait ARQ. Frame 0 is sent and acknowledged. Frame 1 is lost and resent after the time-out. The resent frame 1 is acknowledged and the timer stops. Frame 0 is sent and acknowledged, but the acknowledgment is lost. The sender has no idea if the frame or the acknowledgment is lost, so after the time-out, it resends frame 0, which is acknowledged.




Assume that, in a Stop-and-Wait ARQ system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20 ms to make a round trip. What is the bandwidth-delay product? If the system data frames are 1000 bits in length, what is the utilization percentage of the link?

**Solution**

**The bandwidth-delay product is**

$$(1 \times 10^6) \times (20 \times 10^{-3}) = 20{,}000 \text{ bits}$$

The system can send 20,000 bits during the time it takes for the data to go from the sender to the receiver and then back again. However, the system sends only 1000 bits. We can say that the link

utilization is only 1000/20,000, or 5 percent. For this reason, for a link with a high bandwidth or long delay, the use of Stop-and-Wait ARQ wastes the capacity of the link.

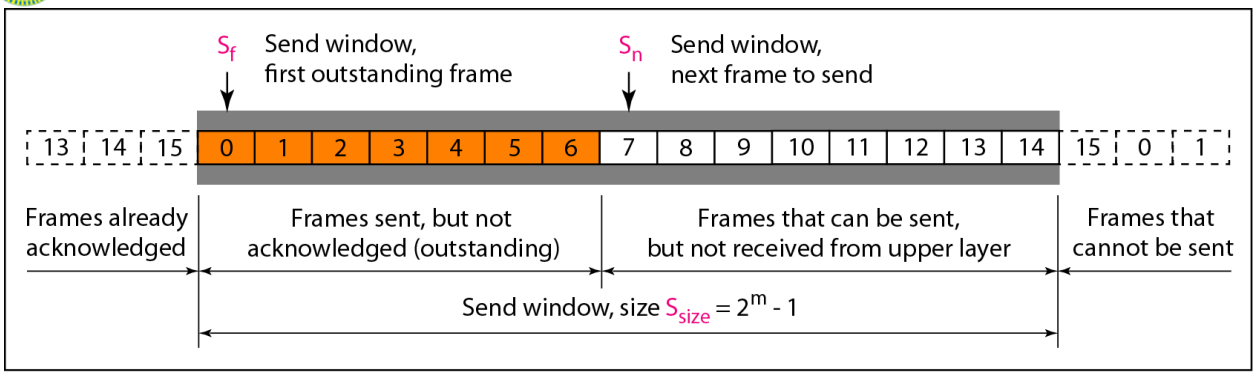
What is the utilization percentage of the link in Example 11.4 if we have a protocol that can send up to 15 frames before stopping and worrying about the acknowledgments?

The bandwidth-delay product is still 20,000 bits. The system can send up to 15 frames or 15,000 bits during a round trip. This means the utilization is 15,000/20,000, or 75 percent. Of course, if there are damaged frames, the utilization percentage is much less because frames have to be resent.

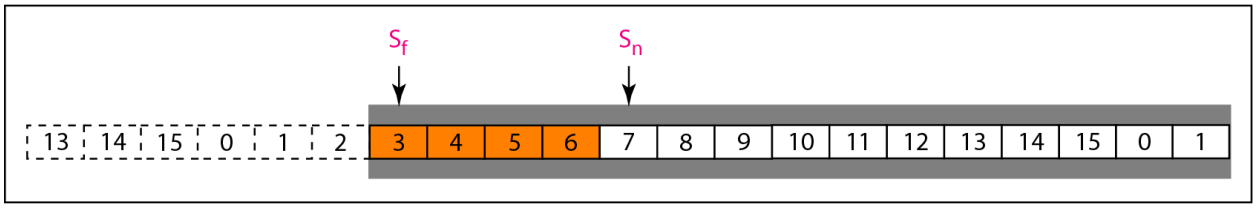
Go-Back-N ARQ Protocol:

- Bandwidth-Delay product.
- Pipelining.
- Sequence numbers range from 0 to 2m – 1.
- Sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the Sender & Receiver.
- Timers – timer for the first outstanding frame always expires first, we resend all outstanding frames when this timer expires.
- Acknowledgement – sends ACK for safe and in order frames, receiver will be silent for corrupted & out of order frames.
- Resending a frame – when the timer expires, the sender resends all the outstanding frames.

***Send window for Go-Back-N ARQ***




a. Send window before sliding

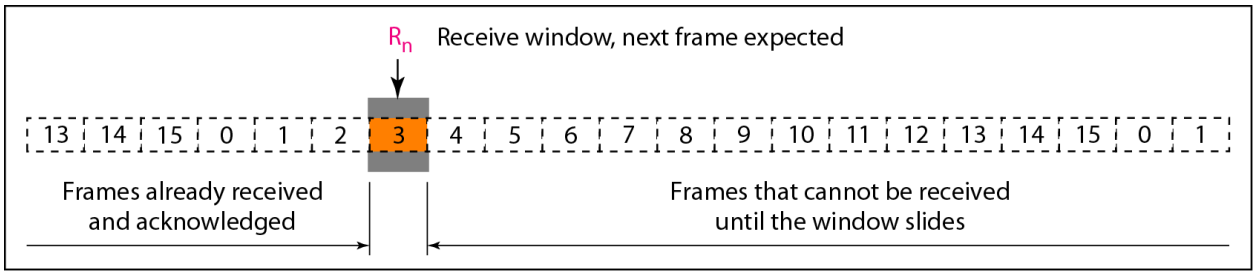



b. Send window after sliding

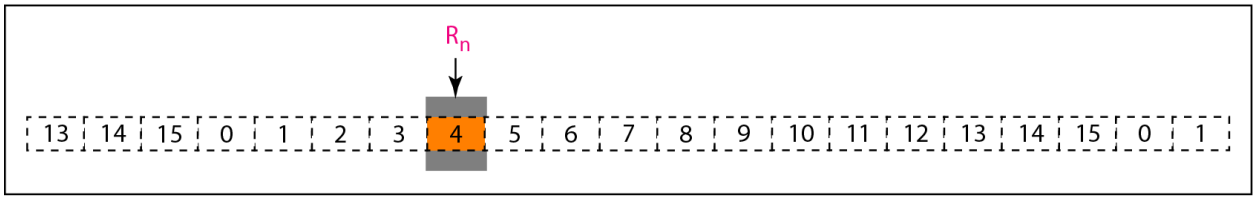
**The send window is an abstract concept defining an imaginary box of size $2^m - 1$ with three variables: $S_f$, $S_n$, and $S_{size}$.**

**The send window can slide one or more slots when a valid acknowledgment arrives.**

**Figure *Receive window for Go-Back-N ARQ***




a. Receive window

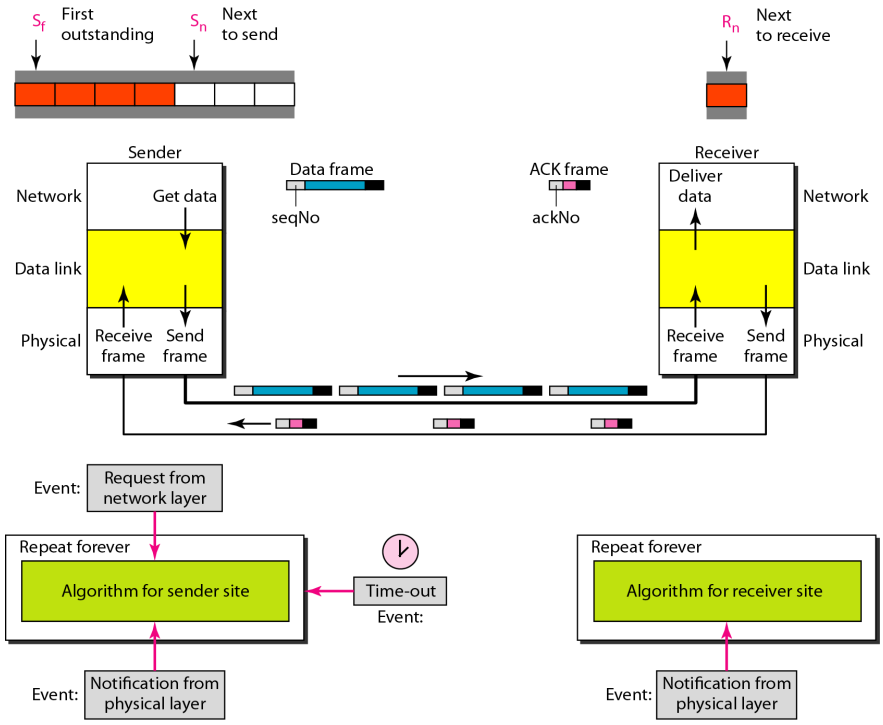



b. Window after sliding

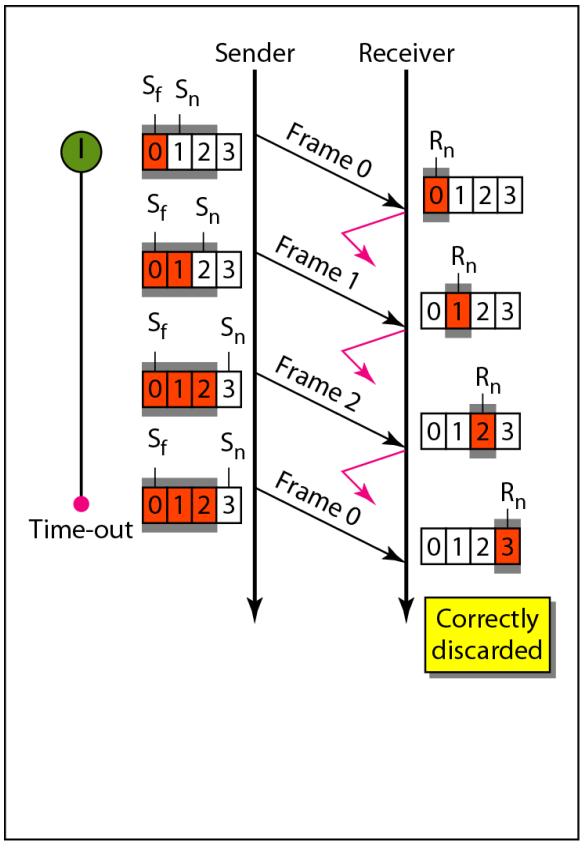
The receive window is an abstract concept defining an imaginary box of size 1 with one single variable $R_n$.

The window slides when a correct frame has arrived; sliding occurs one slot at a time

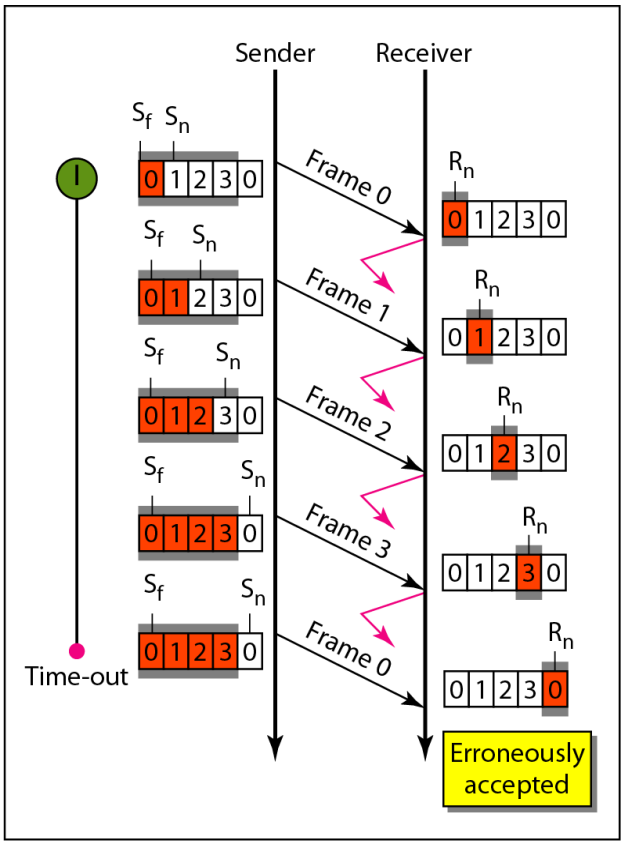
**DESIGN**



***Window size for Go-Back-N ARQ***




a. Window size < $2^m$



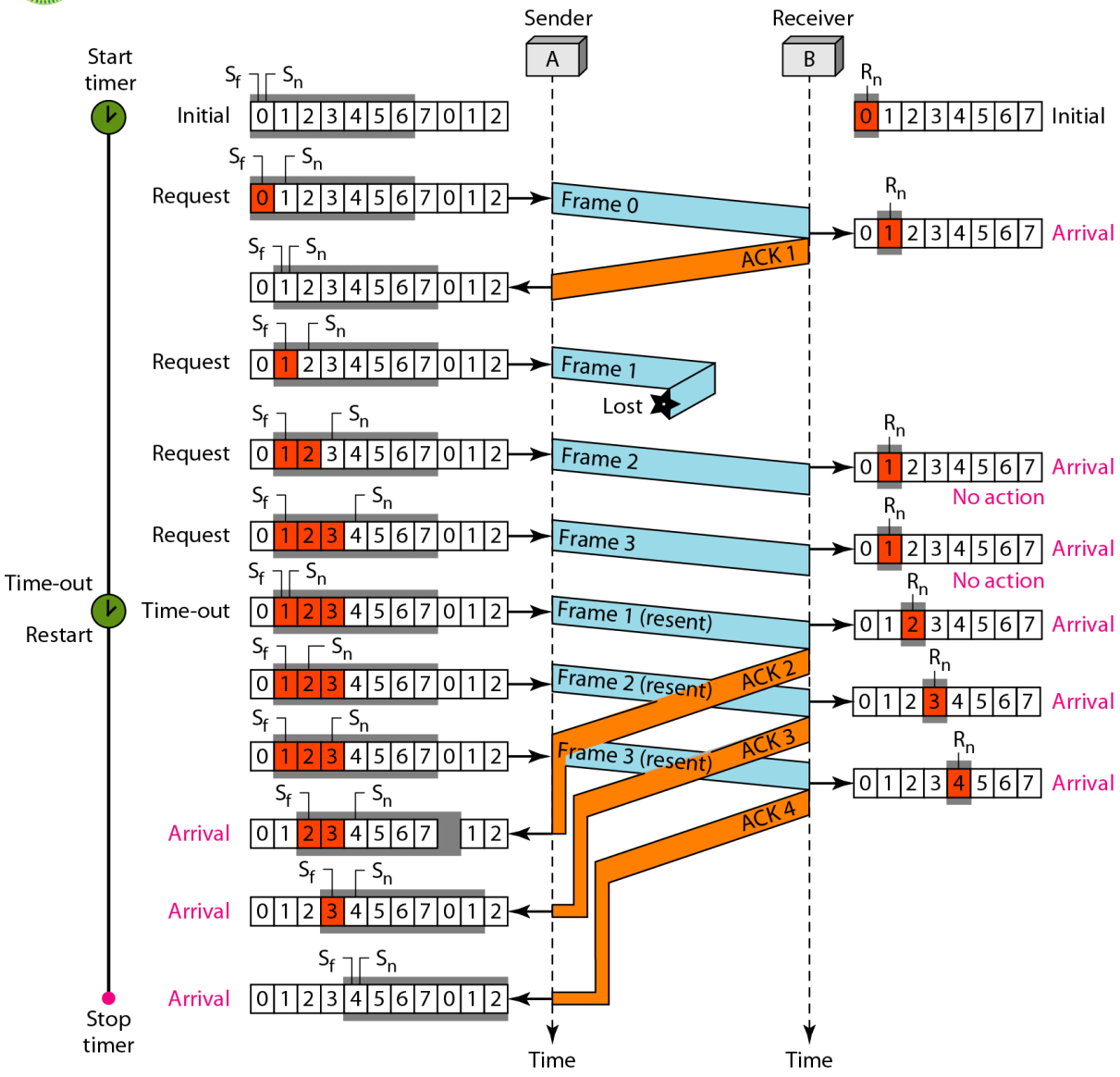

b. Window size = $2^m$

***Sender-site algorithm:***

```
Sw = 2^m – 1;
Sf = 0;
Sn = 0;                                  //Frame 0 should be sent first
While(true)                              //Repeat forever
{
  waitForEvent();                                //Sleep until an event occurs
  if(event(requestTosend)) //there is a packet to send
    {
      if(Sn – Sf >= Sw)                  //If window is full
         Sleep();
      GetData();                                 //get data from n/w layer
      MakeFrame(Sn);                     //make a frame
      StoreFrame(Sn);                             //copy of frame
      SendFrame(Sn);                     //send the frame
      StartTimer();
      Sn = Sn + 1;                               //Mod-2 addition
      if(timer not running);                     //can't send until ACK arrives
         StartTimer();
    }
if(Event(ArrivalNotification))                   //an ACK has arrived
    {
      Receive (ACK);                     //Receive the ACK frame
       if(corrupted (ACK))
         Sleep();
       if((ackNo > Sf && (ackNo <= Sn)))
       While(Sf  <= ackNo)
          {
           Purgeframe(Sn-1);
           Sf = Sf + 1;
           }
        StopTimer();
     }
If(Event(Timeout))
  {
    StartTimer();
    Temp = Sf;
    While(Temp < Sn)
       {
         SendFrame(Sf);                  //sendframe(temp);
          Sf = Sf + 1;                   //temp = temp+1;
        }
  }
}
```

***Receiver-site algorithm :***

```
Rn = 0;                                  //Frame 0 expected to arrive first
While(true)                      //Repeat forever
{


  waitForEvent();                        //Sleep until an event occurs
  if(event(ArrivalNotification))         //data frame arrived
    {
      ReceiveFrame();          //receive frame from the physical layer
      if(Corrupted(Frame))
         sleep();
      if(seqNo == Rn)
        {
         DeliverData();                  //deliver the data to the n/w layer
         Rn = Rn +1;             //Slide window
         SendACK(Rn);
        }
    }
}
```

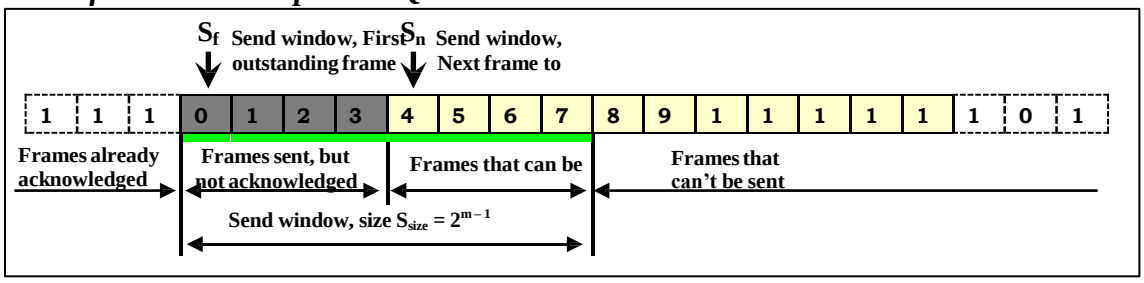

Sender
Receiver
A
B
Start timer
Initial
Request
Frame 0
ACK 1
Arrival
Request
Frame 1
Lost
Request
Frame 2
Arrival
No action
Request
Frame 3
Arrival
No action
Time-out
Time-out
Restart
Frame 1 (resent)
Arrival
Frame 2 (resent)
ACK 2
Arrival
Frame 3 (resent)
ACK 3
Arrival
ACK 4
Arrival
Stop timer
Time
Time

**Stop-and-Wait ARQ is actually a Go-Back-N ARQ in which there are only two sequence numbers and the send window size is 1.**
**i.e., m = 1; $2^m - 1 = 1$**

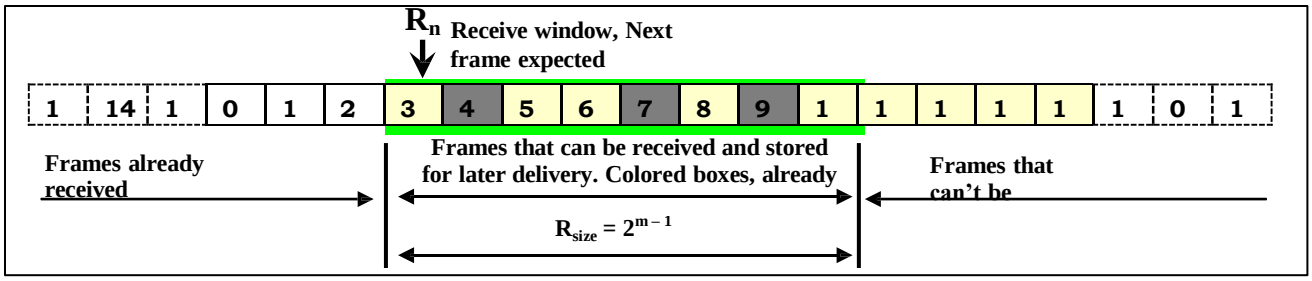
ii) Selective Repeat ARQ Protocol:

- In a noisy link frame has a higher probability of damage, means resending of multiple frames.
- This uses more bandwidth & slows down the transmission.
- Instead of sending N frames when just one frame is damaged, a mechanism is required to resend only the damaged frame.
- Sequence numbers range from 0 to 2m – 1.
- Both send & receive window size = 2(m – 1).

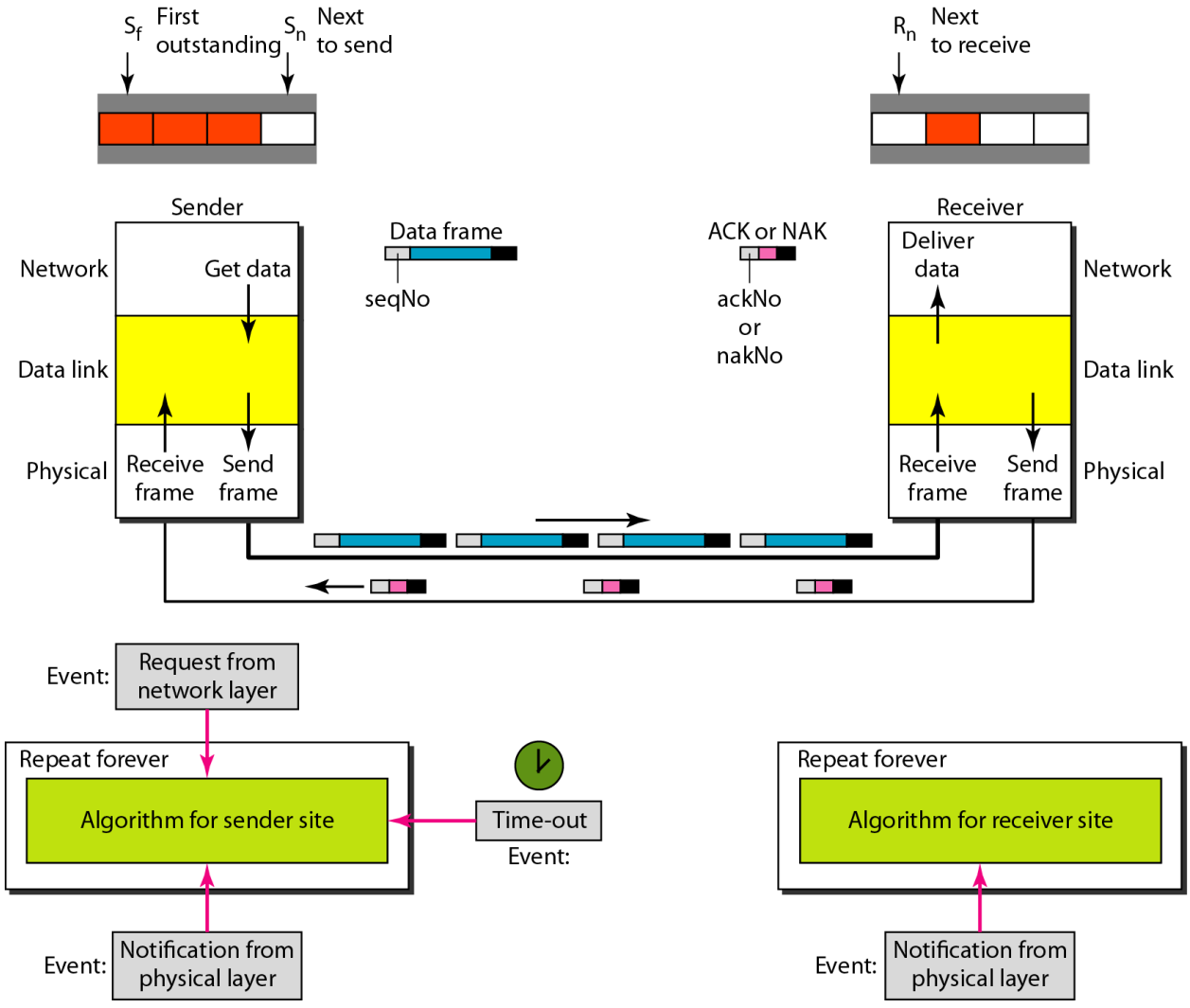
***Send window for Selective Repeat ARQ:***



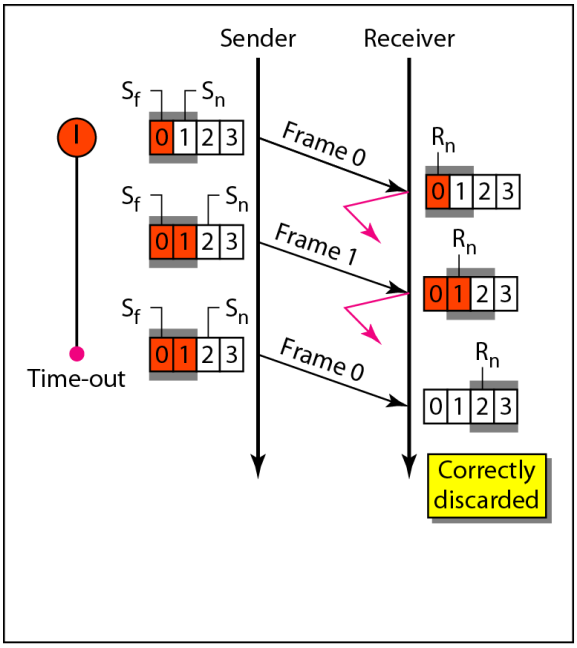

***Receive window for Selective Repeat ARQ:***


$R_n$ Receive window, Next frame expected
1 14 1 0 1 2 3 4 5 6 7 8 9 1 1 1 1 1 1 0 1
Frames already received
Frames that can be received and stored for later delivery. Colored boxes, already
Frames that can't be
$R_{size} = 2^{m-1}$

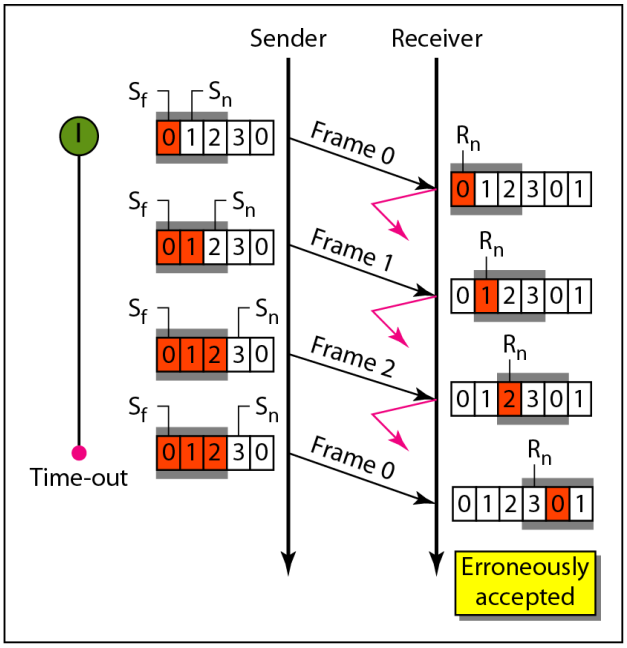
***Design:***


$S_f$ First outstanding
$S_n$ Next to send
$R_n$ Next to receive
Sender
Receiver
Network
Get data
Data link
Physical
Receive frame
Send frame
Data frame
seqNo
ACK or NAK
ackNo or nakNo
Deliver data
Network
Data link
Physical
Receive frame
Send frame
Event:
Request from network layer
Repeat forever
Algorithm for sender site
Time-out
Event:
Event:
Notification from physical layer
Repeat forever
Algorithm for receiver site
Event:
Notification from physical layer

***Selective Repeat ARQ, window size***




a. Window size = $2^{m-1}$




b. Window size > $2^{m-1}$

***Sender-site algorithm:***

```
Sw = 2m-1 ;
Sf = 0;
Sn = 0;                          //Frame 0 should be sent first
While(true)                      //Repeat forever
{
  waitForEvent();                          //Sleep until an event occurs
  if(event(requestTosend)) //there is a packet to send
    {
      if(Sn – Sf >= Sw)                    //If window is full
         Sleep();
      GetData();                           //get data from n/w layer
      MakeFrame(Sn);                       //make a frame
      StoreFrame(Sn);                           //copy of frame
      SendFrame(Sn);           //send the frame
      Sn = Sn + 1;
      StartTimer(Sn);
}
if(Event(ArrivalNotification))           //an ACK has arrived
   {   Receive (frame);                    //Receive the ACK frame
      if (corrupted (frame))
        Sleep();
       if (FrameType == NAK)
          if (nakNo between Sf & Sn)
             {   resend(nakNo);
                 StartTimer(nakNo);
             }
          if (FrameType == ACK)
             if (ackNo between Sf & Sn)
```

```
                {    While (Sf  <= ackNo)
                        {
                         Purge (Sf);
                         StopTimer (Sf);
                         Sf = Sf + 1;
                        }
                }
        }
If (Event (Timeout (t)))
  {
    StartTimer(t);
    SendFrame(t);
   }
}
```

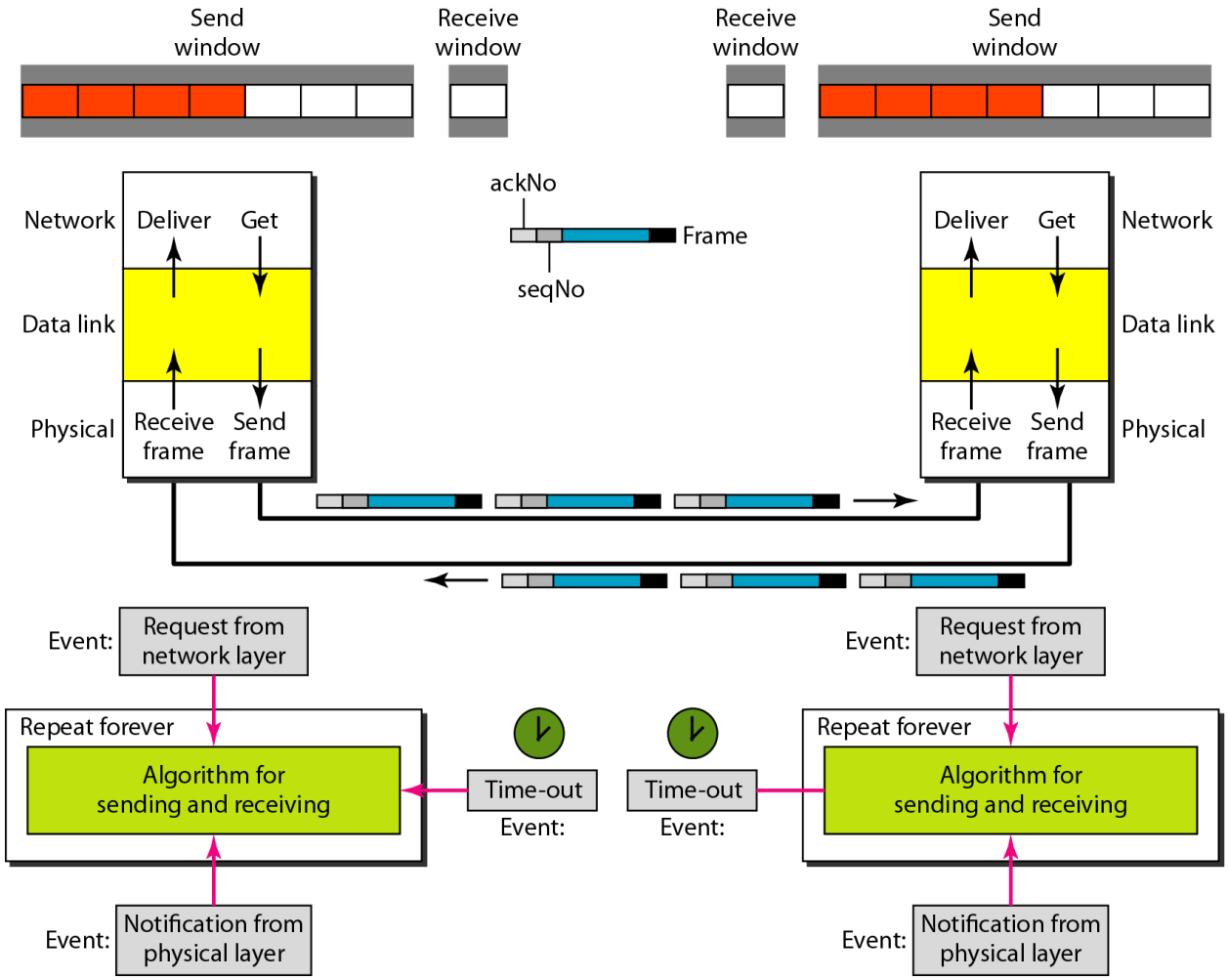
***Receiver-site algorithm :***

```
Rn = 0;                    //Frame 0 expected to arrive first
Naksent = false;
AckNeeded = false;
Repeat( for all slots)
     Marked (slot) = false;

While(true)                        //Repeat forever
{
  waitForEvent();                          //Sleep until an event occurs
  if(event(ArrivalNotification))           //data frame arrived
    {
      ReceiveFrame();
      if(Corrupted(Frame) && NOT NakSent)
        {
         SendNAK(Rn);
         Naksent  =  true;
         sleep();
         }
if(seqNo < > Rn && NOT Naksent)
   {    SendNAK(Rn);
        Naksent = true;
        if((seqno in window) && (!Marked(seqno)))
          {   StoreFrame(seqNo);
              Marked(seqNo) = true;
              while(Marked(Rn))
                 {  DeliverData(Rn);              //deliver the data to the n/w layer
                    purge(Rn);
                    Rn = Rn +1;            //Slide window
                    AckNeeded = true;  }
```

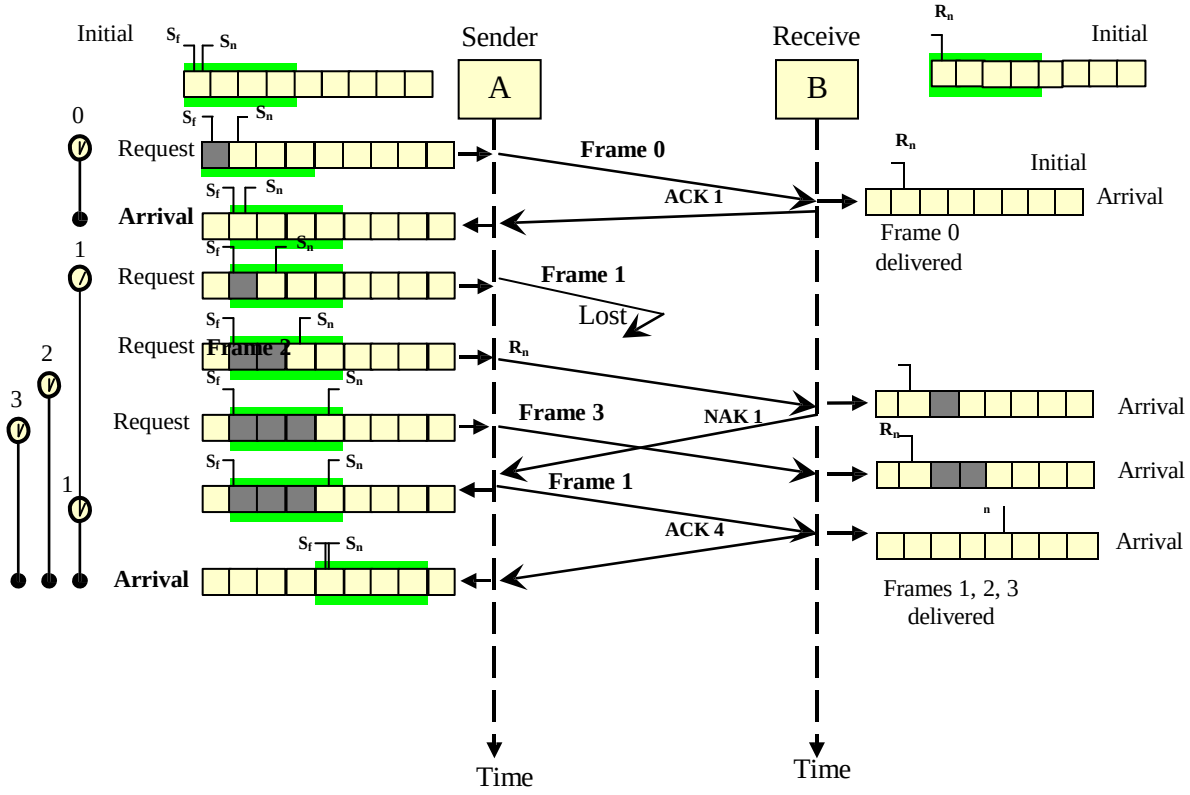
```
            if(AckNeeded)

              {  SendACK(Rn);
                 AckNeeded = false;
                 NakSent = false;       }
          }
    }
    }
}
```

***Piggybacking***

- When the data transfer is bidirectional, i.e., from node A to node B and from node B to node A, the control information also needs to flow in both the directions.
- Efficiency can be improved if the control information can be passed to the other end along with the data itself which is flowing to that end. This concept is called as piggybacking.

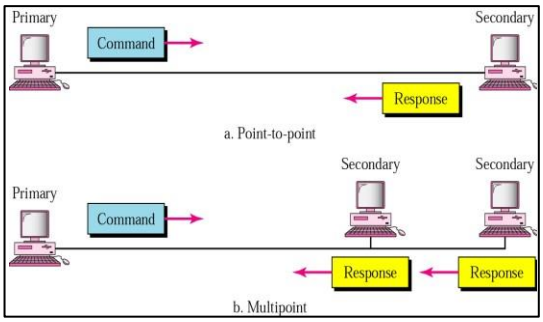


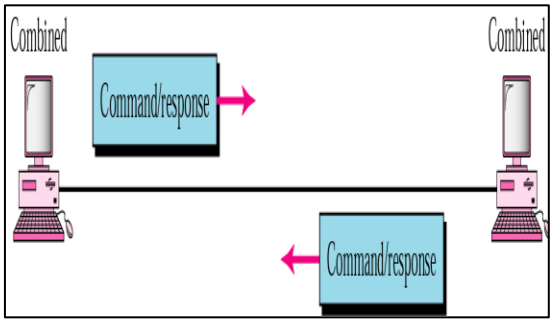
***Flow diagram:***



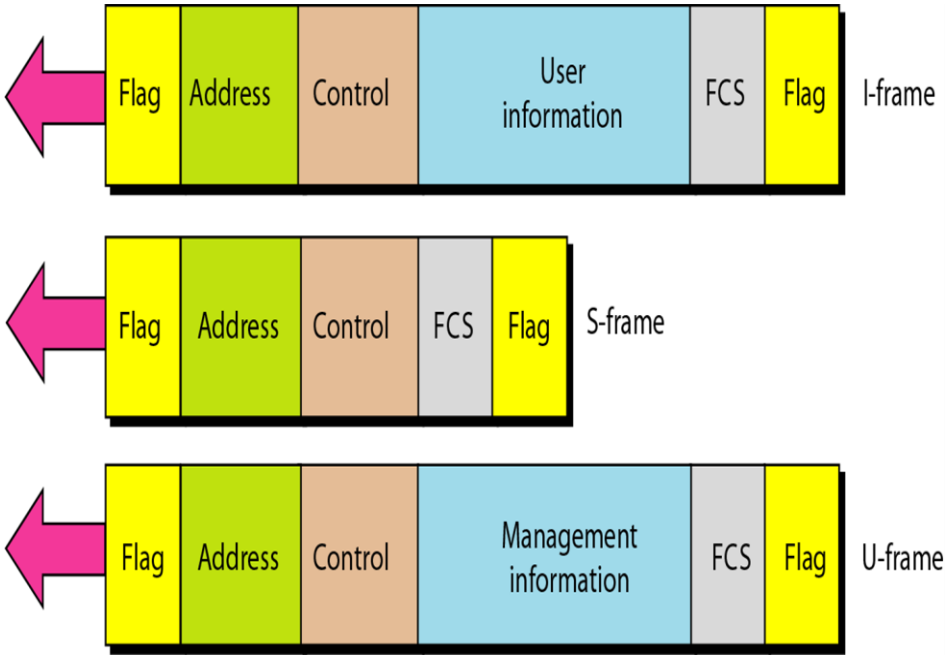
## 2.6 High-level Data Link Control (HDLC):

- HDLC is a bit-oriented protocol for communication over point-to-point and multipoint links.
- It implements the ARQ mechanisms.
- HDLC provides two common transfer modes.

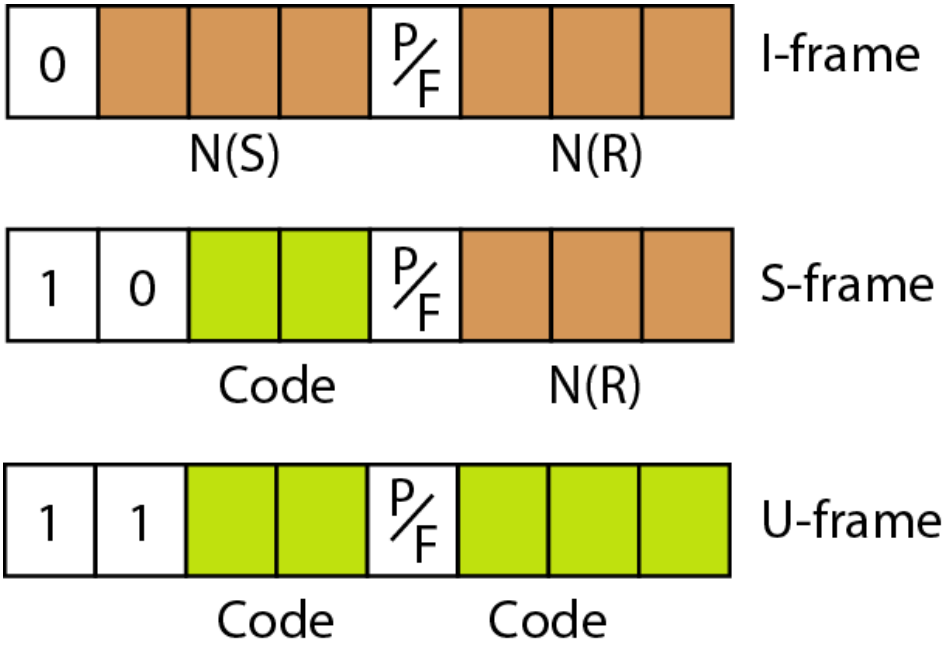



***Frame format :***



***Control field format for the different frame types***




***U-frame control command and response***

| *Code* | *Command* | *Response* | *Meaning* |
|---|---|---|---|
| **00 001** | SNRM | | Set normal response mode |
| **11 011** | SNRME | | Set normal response mode, extended |
| **11 100** | SABM | **DM** | Set asynchronous balanced mode or **disconnect mode** |
| **11 110** | SABME | | Set asynchronous balanced mode, extended |
| **00 000** | UI | **UI** | Unnumbered information |
| **00 110** | | **UA** | **Unnumbered acknowledgment** |
| **00 010** | DISC | **RD** | Disconnect or **request disconnect** |
| **10 000** | SIM | **RIM** | Set initialization mode or **request information mode** |
| **00 100** | UP | | Unnumbered poll |
| **11 001** | RSET | | Reset |
| **11 101** | XID | **XID** | Exchange ID |
| **10 001** | FRMR | **FRMR** | Frame reject |

# UNIT – II

**1.** a) Explain the mechanism of selective ARQ with diagram showing send window and receive window. **(July 2013 10marks)**

b) With suitable block diagram explain the stop and wait protocol for noiseless channels also write the sender site algorithm **(July 2013 06 marks)**

c) Perform bit stuffing and unstuffing on the given bit stream: 0001111111001111101000 Assume flag as 01111110 **(July 2013 04 marks)**

**2.** a) Differentiate between character stuffing and bit stuffing with example **(Jan 2013 05marks)**

**b)** Explain different HDLC frames **(Jan 2013 05marks)**

C) What is sliding window protocol? Explain Go-Back-N protocol for noisy channel **(Jan 2013 10marks)**

**3.** a) With a neat diagram of piggy backing in GO-Back-N protocol, explain the following:

i) Frame structure of piggy backing

ii) Types of event occurred in piggybacking

iii) Advantages of piggybacking **(July 2012 10marks)**

b) With the neat diagram explain the different types of high level data link control (HDLC) frames **(July 2012 06marks)**

c) The following character encoding is used in data link protocol:

A:01000111; B:11100011; FLAG:01111110; ESC=11100000. Show the bit sequence transmitted in binary for the four character frame:A B ESC FLAG when each of the following framing methods are used:

i) Character count

ii) Flag bytes with byte stuffing

iii)Starting and ending flag bytes with bit stuffing **(July 2012 04marks)**

**4.** a) Explain byte stuffing and unstuffing and bit stuffing and unstuffing with necessary diagrams **(July 2011 10 marks)**

b)With a neat diagram explain three different types of HDLC frames **(July 2011 10 marks)**

**5.** a)In stop and wait ARQ system, the bandwidth of the line is 1Mbps and it takes 20ms to make round trip. What is the bandwidth delay product? If the system data frames are of 1000 bit length, what is the utilization percentage of link? What is the channel utilization percentage of link if the protocol that can send up to 15 K mes before stopping and worrying about the acknowledgement? Write the comment**.** **(Dec 2011 05 marks)**

**b)** Explain briefly the bit and character stuffing. **(Dec 2011 05 marks)**

c)With a neat diagram, explain the HDLC frame format. **(Dec 2011 10 marks)**

**6.** a) Explain Stop-and-Wait protocol, for noisy channel **(Dec 2010 10 marks)**

b) What are the three types of frames in HDLC protocol? Explain each of them briefly

**(Dec 2010 10 marks)**

**7.** a )What is framing? How frames can be classified? Explain bit stuffing with the help of an example. **(June 2010 06 marks)**

b) What is the meaning of datalink control? Explain stop-and-wait ARQ, using a suitable block diagram. **(June 2010 10 marks)**

c) In a stop-and-wait ARQ system the bandwidth of the line is 1Mbps and 1 bit takes 20ms to make a round trip. What is the bandwidth delay product? If the system data frames are 1000 bits in length s, what is the percentage utilization of the link? **(June 2010 04 marks)**

**8.** a) What is framing? Explain bitstuffing with diagram **(Dec 2014 10 marks**)

**b)**Expalin stop and wait automatic reapeat request protocol for noisy channel (**Dec 2014 10 marks)**

**9.** a)With frame format,given an elaborate account on HDLC (**Jun 2014 10 marks)**

b) What is ARQ ? Describe in detail about GO-BACK N ARQ (**Jun 2014 10 marks)**

**10. a)**Explain bitstuffing with example? **(Jun 2014 10marks)**

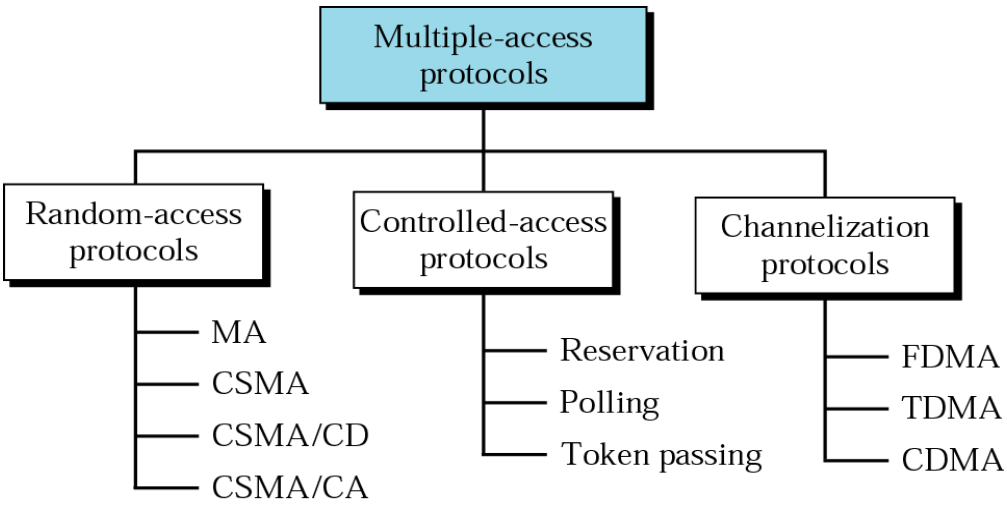
b)With neat diagram explain HDLC frame format (**Dec 2014 10marks)**

## Unit 3: Multiple Accesses

## 3.1 Introduction:

- Data link layer has two sub layers: Upper sub layer – Data link Control & Lower sub layer – Multiple access Control.
- The upper sub layer is responsible for flow and error control.
- The lower sub layer is responsible for multiple access resolution.
- When the nodes are connected using a dedicated link, lower sub layer is not required, but when the nodes are connected using a multipoint link (broadcast), multiple access resolution is required.

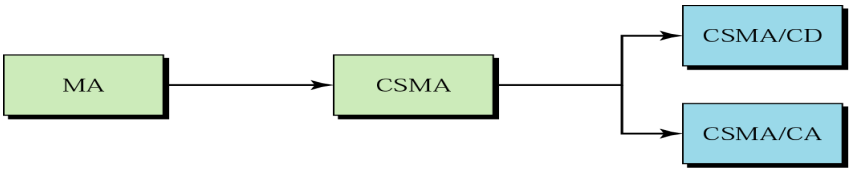
***Taxonomy of Multiple access protocols***



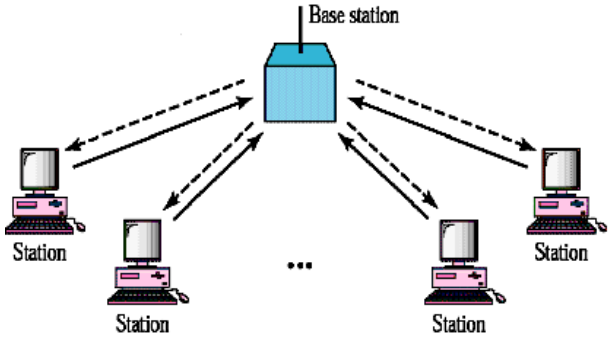

## *3.2 RANDOM ACCESS*

- No station is superior to another station and none is assigned control over another.
- No station permits, or does not permit, another station to send.
- At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.
- Transmission is random among the stations.
- Each station has the right to the medium without being controlled by any other station. All stations compete with one another to access the medium. Random access methods are also called as *contention methods*.
- If more than one station tries to send, there is an access conflict – collision, frames will be either destroyed or modified.

- In random access or contention 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. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.



## *i) ALOHA*

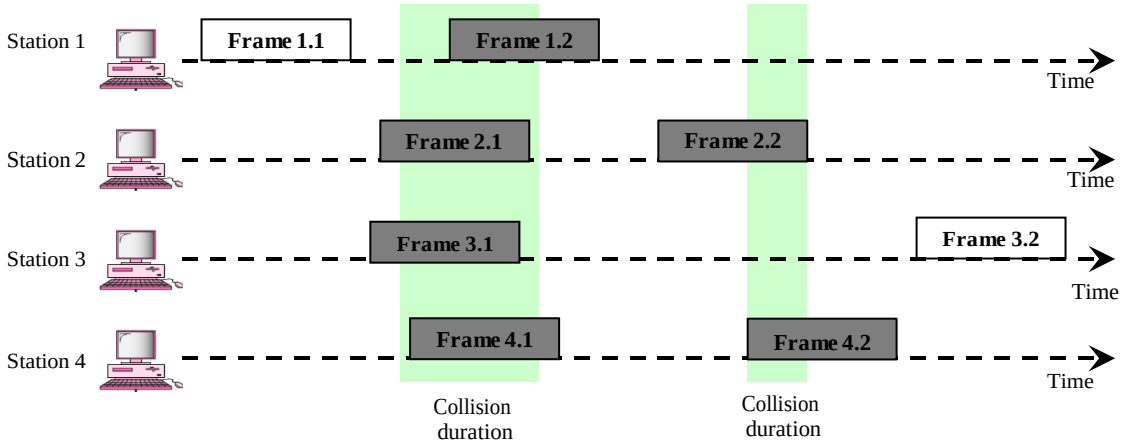



- The earliest random access method, was developed at the university of Hawaii in early 1970.
- It was designed for a radio (wireless) LAN, but it can be used on any shared medium.
- The medium is shared between the stations. When a station sends data, another station may attempt to do so at the same time. The data from the two stations collide and become garbled.

## *a) PURE ALOHA*

- The original ALOHA protocol is called pure ALOHA.
- Each station sends a frame whenever it has a frame to send.
- Since there is only one channel to share, there is possibility of collision between frames from different stations.

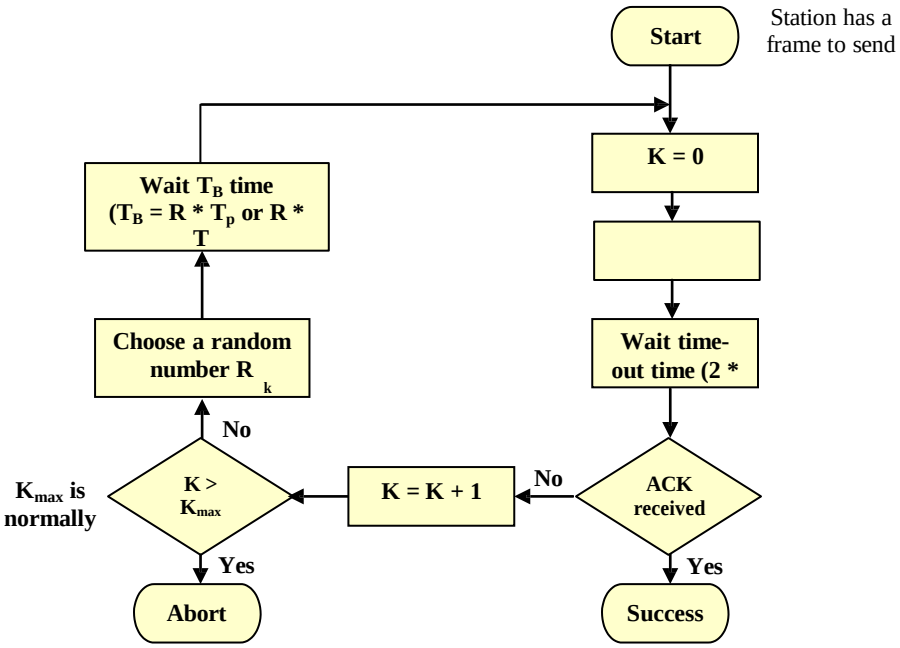
## *Frames in a pure ALOHA network*



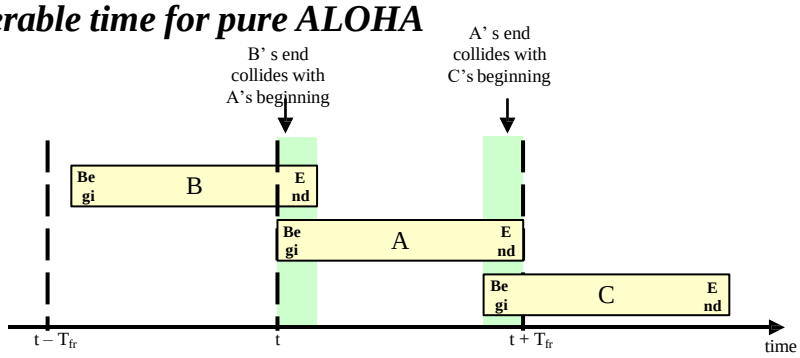

- Even if one bit of a frame coexists on the channel with one bit from another frame, there is a collision and both will be destroyed.

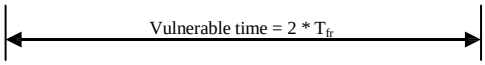
- Retransmission of frames are required for the destroyed frames.
- The *pure ALOHA* protocol relies on acknowledgements from the receiver. When a station sends a frame, it expects the receiver to send an acknowledgement.
- If the acknowledgement does not arrive after a time-out period, the station assumes that the ACK has been destroyed and resends a 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. This time is called back-off time TB. This randomness will help avoid more collisions.
- To prevent congesting the channel with retransmitted frames, *pure ALOHA* dictates that after a maximum number of retransmission attempts Kmax, a station must give up and try later.

## *Procedure for pure ALOHA protocol*




## *Vulnerable time for pure ALOHA*






***Throughput***

G=load = average number of frames generated by the system during one frame transmission time.

Successful transmissions for pure ALOHA is

$$S = G * e^{-2G}$$

$$D(S)/D(G)=0;$$

$$e^{-2G} + G * e^{-2G} * -2 = 0;$$

$$G=1/2;$$

$$S=1/2 * e^{-2*1/2}$$

$$=18.39;$$

$$=18.4$$

The maximum throughput, Smax is for G = ½. i.e., Smax = 0.184 = 18.4%.

The stations on a wireless ALOHA network are a maximum of 600 km apart. If we assume that signals propagate at 3 × 108 m/s, we find

Tp = (600 × 105 ) / (3 × 108 ) = 2 ms.

Now we can find the value of TB for different values of K .

For K = 1, the range is {0, 1}. The station needs to| generate a random number with a value of 0 or 1. This means that TB is either 0 ms (0 × 2) or 2 ms (1 × 2), based on the outcome of the random variable.

For K = 2, the range is {0, 1, 2, 3}. This means that TB can be 0, 2, 4, or 6 ms, based on the outcome of the random variable.

For K = 3, the range is {0, 1, 2, 3, 4, 5, 6, 7}. This means that TB can be 0, 2, 4, . . . , 14 ms, based on the outcome of the random variable.

We need to mention that if K > 10, it is normally set to 10

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?

Average frame transmission time Tfr is 200 bits/200 kbps or 1 ms. The vulnerable time is 2 × 1 ms

= 2 ms. This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the one 1-ms period that this station is sending.

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
a. 1000 frames per second b. 500 frames per second
c. 250 frames per second.

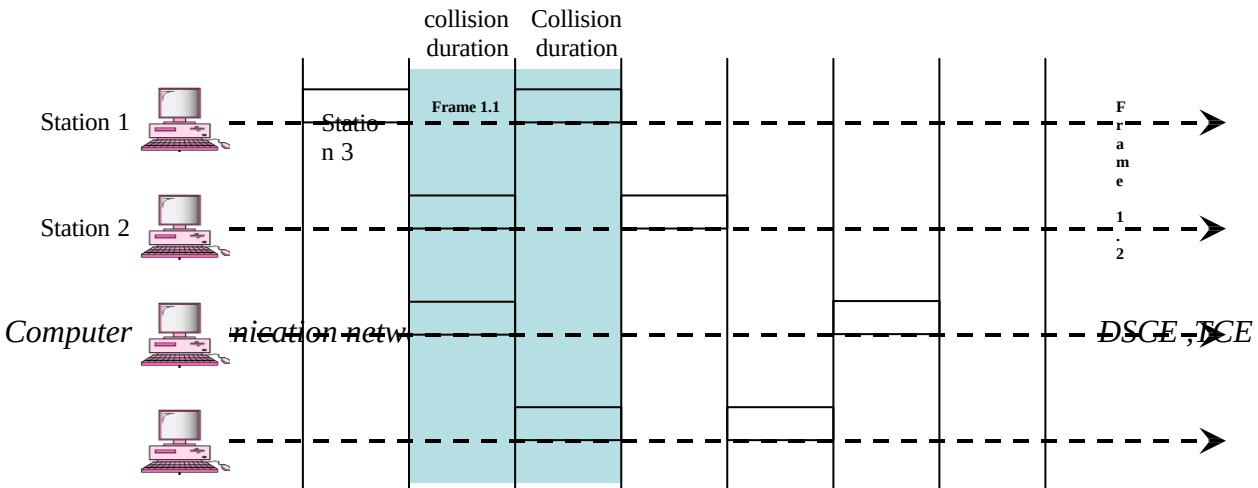
The frame transmission time is 200/200 kbps or 1 ms.
a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case
S = G× e−2 G or S = 0.135 (13.5 percent). This means that the throughput is 1000 × 0.135 = 135 frames. Only 135 frames out of 1000 will probably survive.

b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case S = G × e $^{-2G}$ or S = 0.184 (18.4 percent). This means that the throughput is 500 × 0.184 = 92 and that only 92 frames out of 500 will probably survive. Note that this is the maximum throughput case, percentagewise.

c.If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case S = G × e −2G or S = 0.152 (15.2 percent). This means that the throughput is 250 × 0.152 = 38. Only 38 frames out of 250 will probably survive.

## *b) SLOTTED ALOHA*

### *Frames in a slotted ALOHA network*



Frame 2.1 Frame 2.2 Time

Frame 3.1 Time Frame 3.2

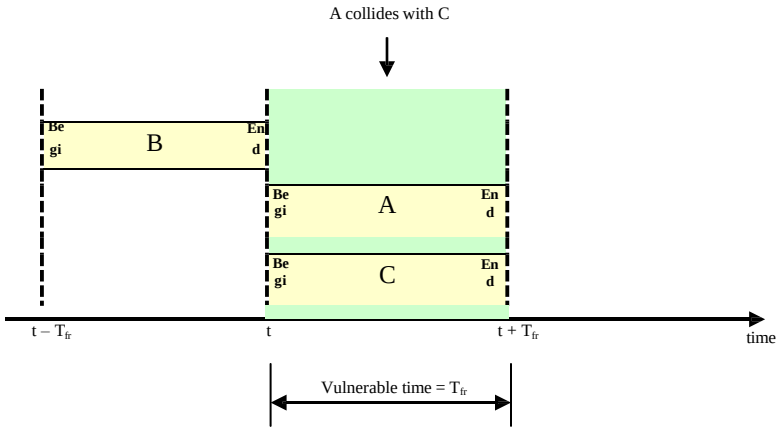
Time

Station 4 Frame 4.1 Frame 4.2 Time

**Slot 1 Slot 2 Slot 3 Slot 4 Slot 5 Slot 6 Slot 7**

Slotted ALOHA Vulnerable time = $T_{fr}$

***Vulnerable time for slotted ALOHA***



### *Throughput*

G = average number of frames generated by the system during one frame transmission time.

Successful transmissions for slotted ALOHA is

$$S = G * e^{-G}$$

The maximum throughput, Smax is for G =1 i.e., Smax = 0.368 = 36.8%.

A slotted ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces
a. 1000 frames per second b. 500 frames per second
c. 250 frames per second.

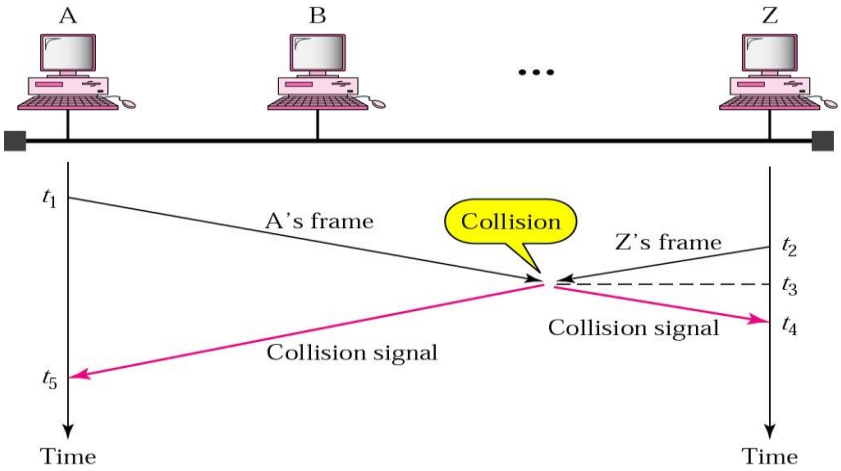
The frame transmission time is 200/200 kbps or 1 ms.
a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case S = G× e−G or S = 0.368 (36.8 percent). This means that the throughput is 1000 × 0.368 = 368 frames. Only 368 frames out of 1000 will probably survive

b.If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case S = G × e−G or S = 0.303 (30.3 percent). This means that the throughput is 500 × 0.0303 = 151. Only 151 frames out of 500 will probably survive.

C If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case S = G × e −G or S = 0.195 (19.5 percent). This means that the throughput is 250 × 0.195 = 49. Only 49 frames out of 250 will probably survive.

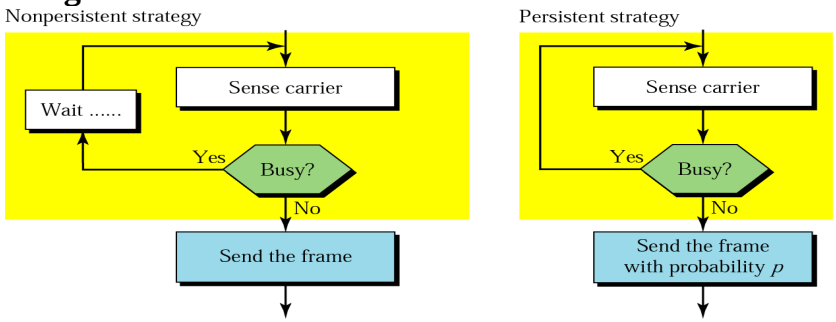
### *C) Carrier Sense Multiple Access (CSMA)*

The chance of collision can be reduced if a station senses the medium before trying to use it. CSMA requires that each station first listen to the medium before sending. CSMA is based on the principle " sense before transmit" or "listen before talk".

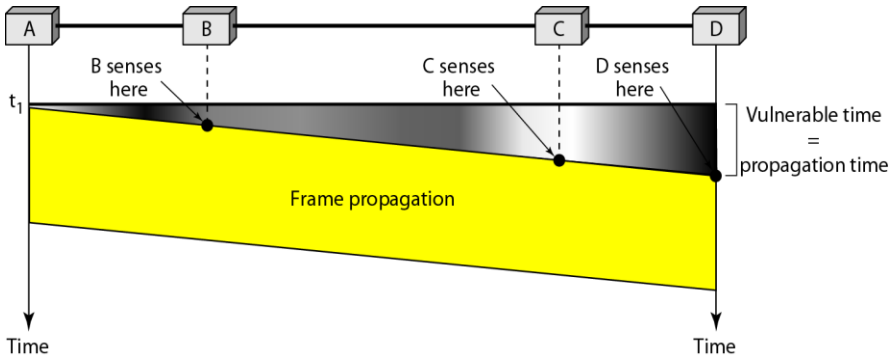


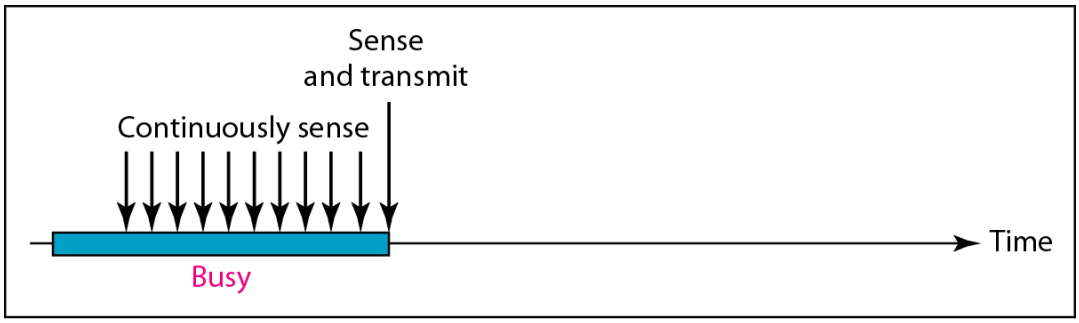

CSMA can reduce the possibility of collision, but it can't eliminate it. The reason for this is shown in the above figure, a space and time model of a CSMA network. Stations are connected to a shared channel. The possibility of collision still exists because of the propagation delay.

## Persistence strategies

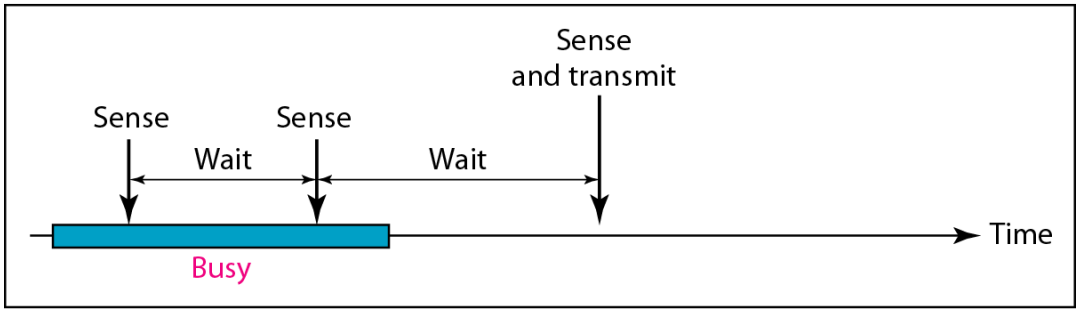



***Vulnerable time in CSMA***

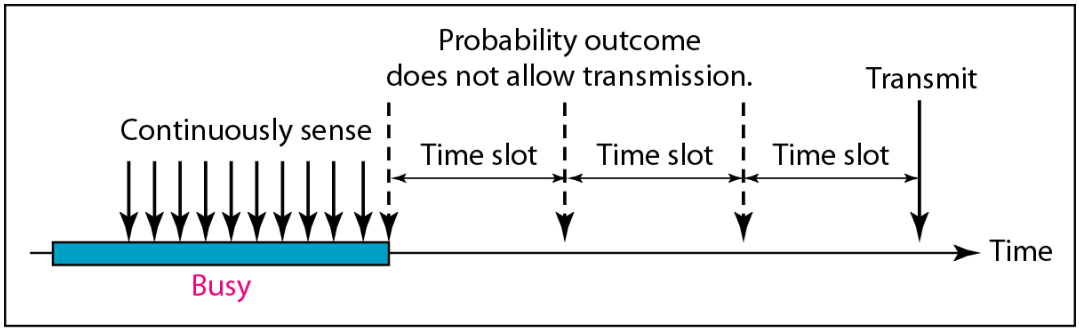





a. 1-persistent




b. Nonpersistent



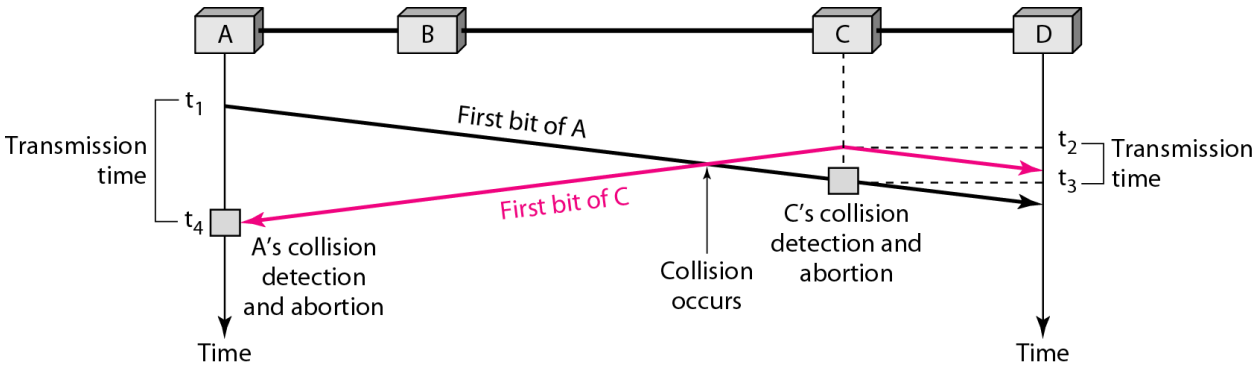

c. p-persistent

### *C Carrier Sense Multiple Access with collision detection (CSMA/CD)*

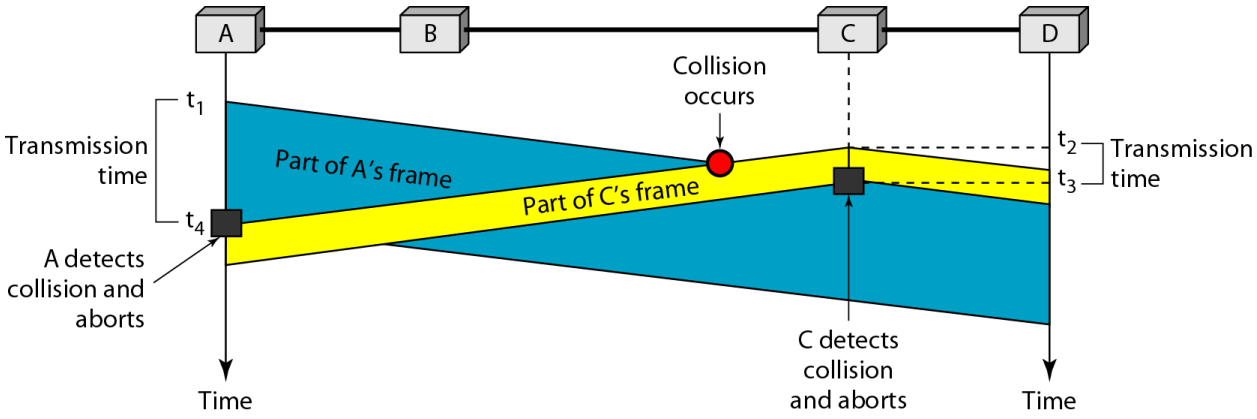
The CSMA method does not specify the procedure following the collision. CSMA with collision detection augments the algorithm to handle the collision.

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.

***Collision of the first bit in CSMA/CD***

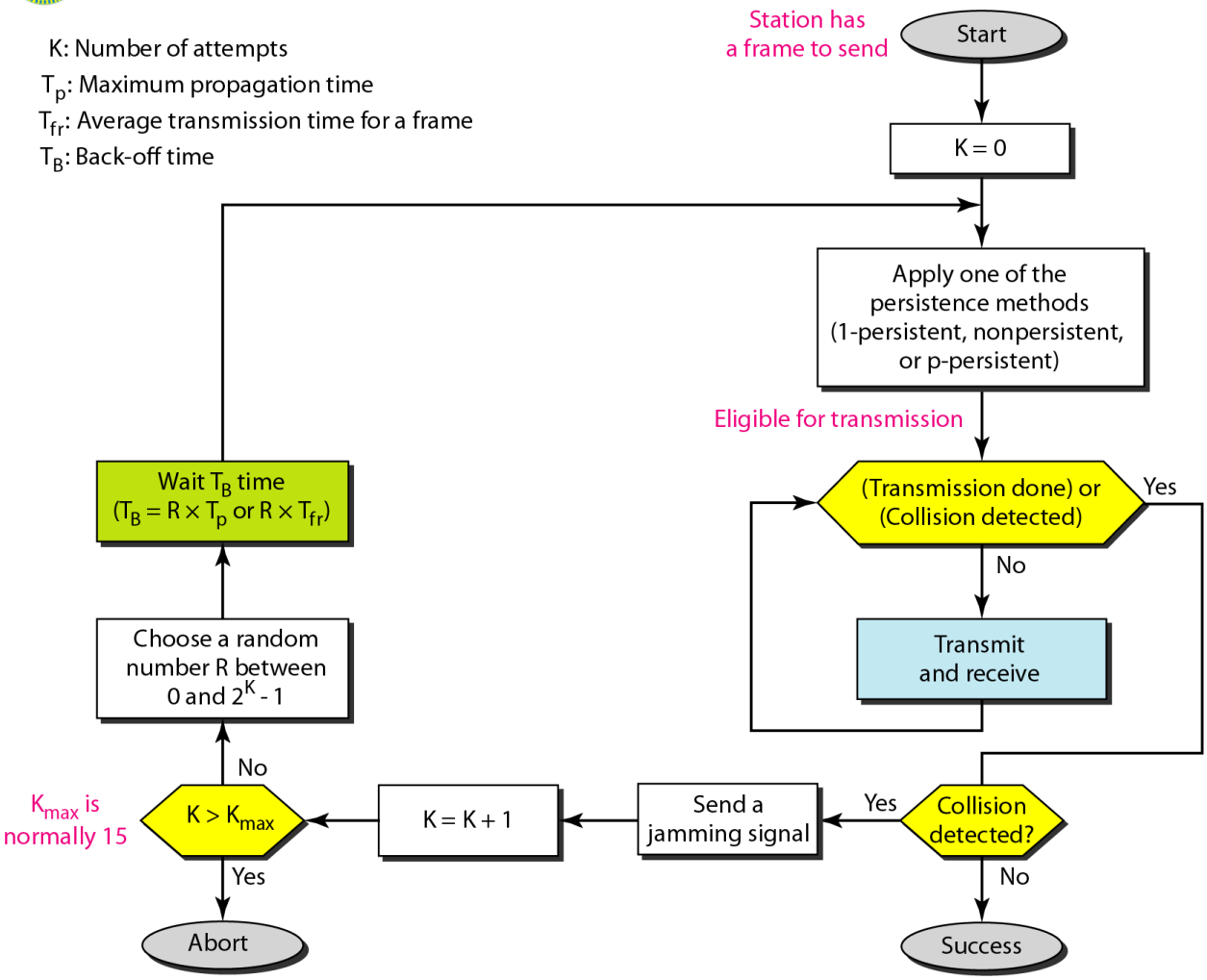



***Collision and abortion in CSMA/CD***




A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal, as we see later) is 25.6 μs, what is the minimum size of the frame?

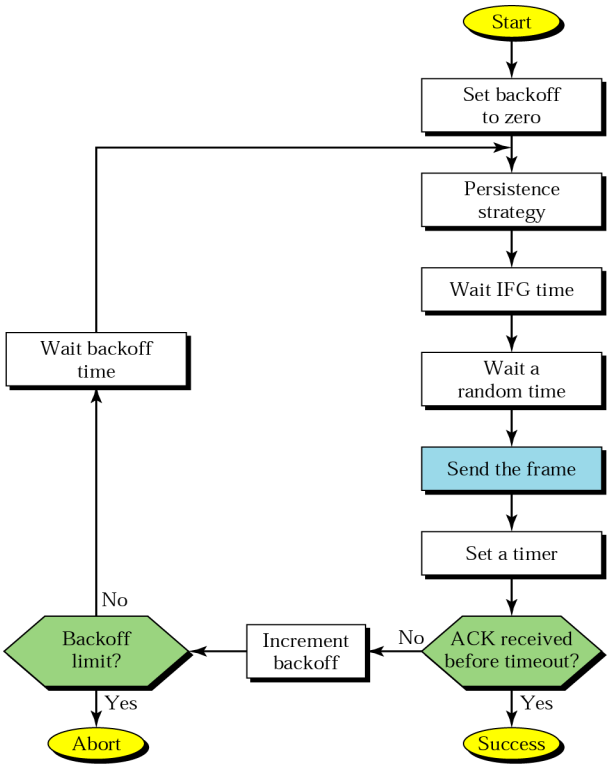
The frame transmission time is $T_{fr} = 2 \times T_p = 51.2$ μs. This means, in the worst case, a station needs to transmit for a period of 51.2 μs to detect the collision. The minimum size of the frame is 10 Mbps × 51.2 μs = 512 bits or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet.




## *C) Carrier Sense Multiple Access with collision avoidance (CSMA/CA)*

The basic idea behind CSMA/CD is that a station needs to be able to receive while transmitting to detect a collision. When there is no collision, the station receives one signal; its own signal. When there is a collision, the station receives two signals; its own signal and the signal transmitted by the second station.

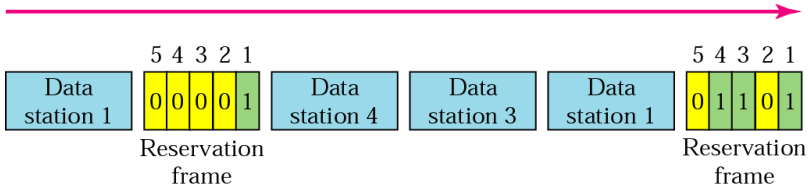
Collisions are avoided by deferring transmissions even if the channel is found idle. When an idle channel is found, the station does not send immediately. It waits for a period of time called the inter frame space or IFS. Even though the channel may appear idle when it is sensed, a distant station's 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.




## *3.3 CONTROLLED ACCESS*

In controlled access, the stations consult one another to find which station has the right to send. A station can't send unless it has been authorized by other stations.

### a) Reservation:



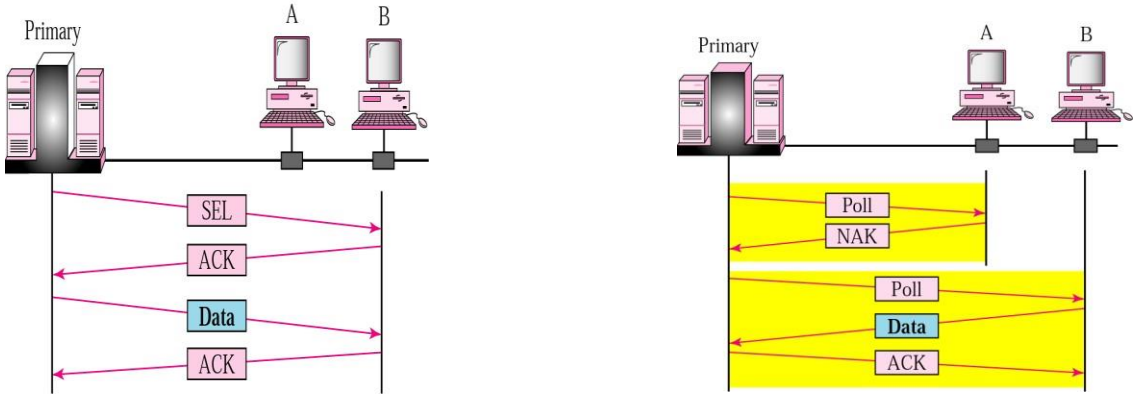

In the reservation method, a station needs to make a reservation before sending the data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in the interval.

### b) Polling:

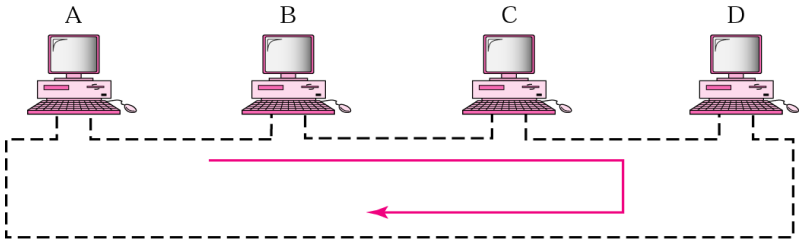
Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations. All data exchanges must be made through the

primary device even when the ultimate destination is a secondary device.

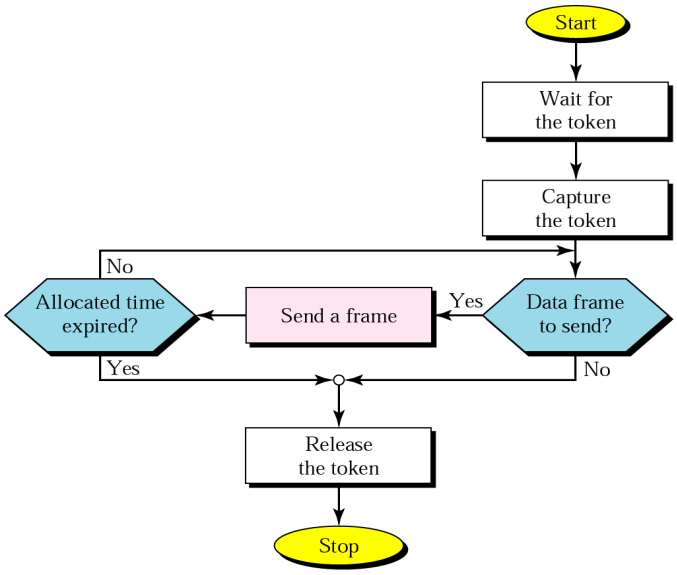
The select function is used whenever the primary device has something to send. The poll function is used by the primary device to solicit transmissions from the secondary device.



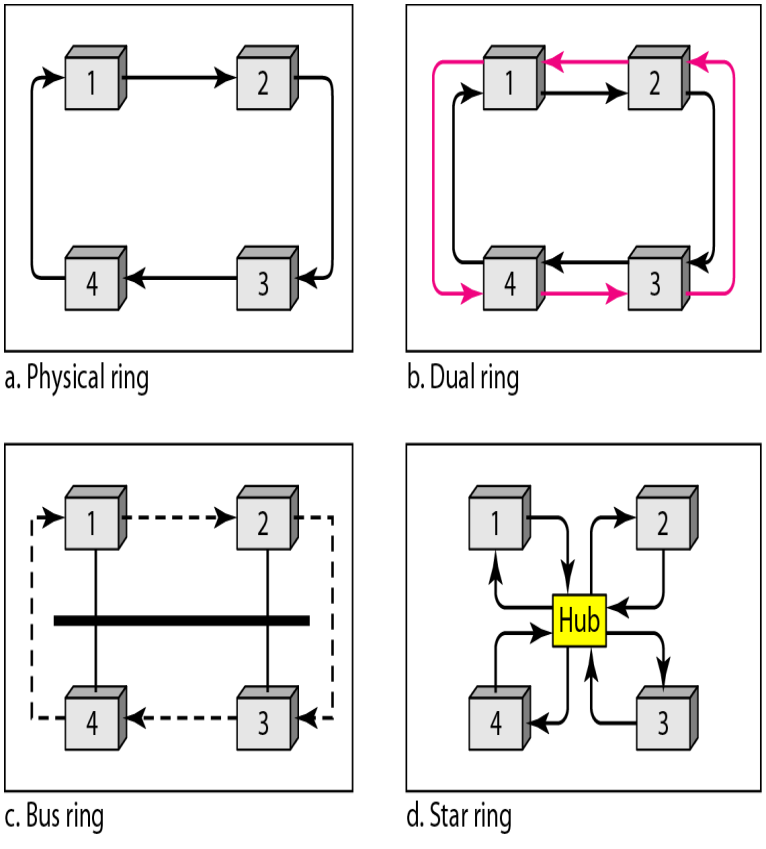

## c) Token Passing:




In the token-passing method, the stations in a network are organized in a logical ring. 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.



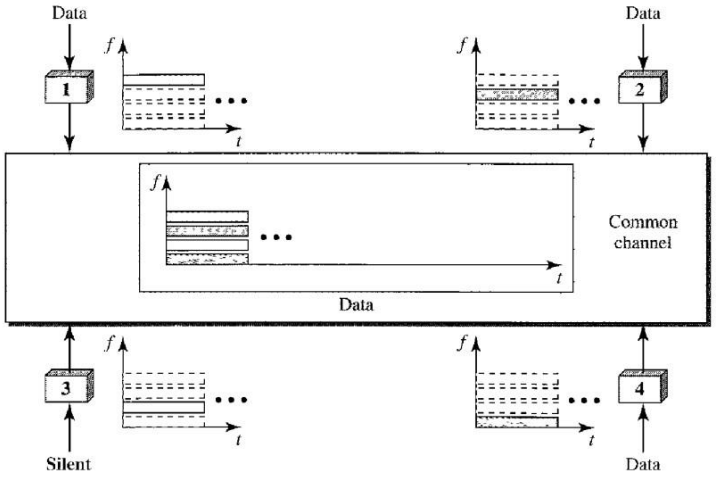

***Logical ring and physical topology in token-passing access***



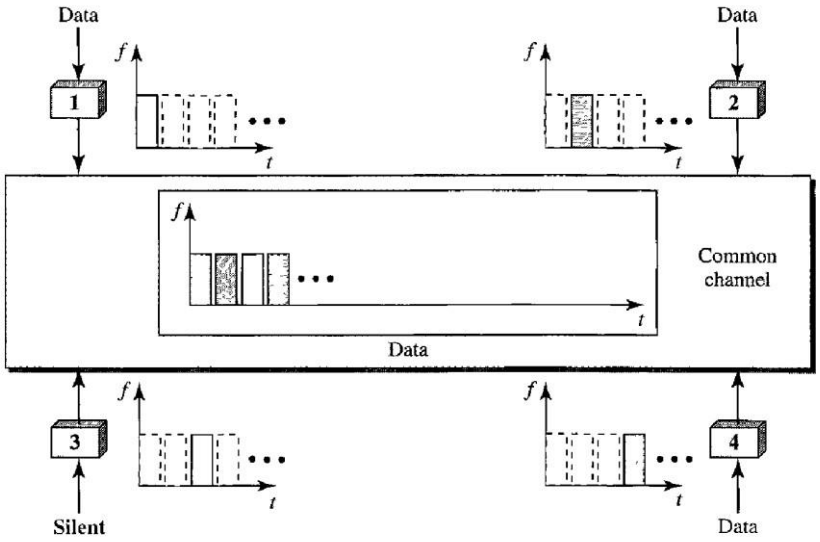
## *3.4 CHANNELIZATION*

### *a) Frequency Division Multiple Access (FDMA)*

In FDMA, the bandwidth is divided into channels. Each station is allocated a band to send its data. Each station also uses a band pass filter to confine the transmitter frequencies. To prevent station interferences, the allocated bands are separated from one another by small guard bands.




### *b) Time Division Multiple Access (TDMA)*




In 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.

### *c) Code Division Multiple Access (CDMA)*

CDMA differs from FDMA because only one channel occupies the entire bandwidth of the link and it differs from TDMA because all stations can send data simultaneously.

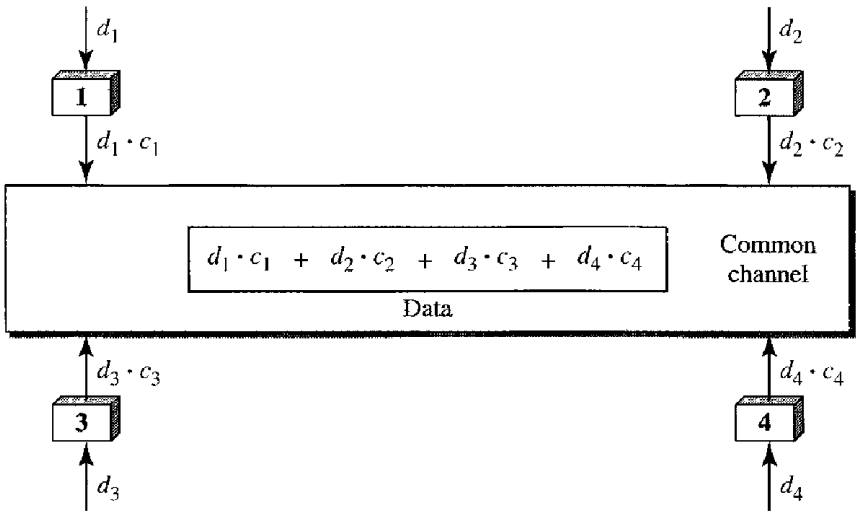
#### *Analogy*

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).

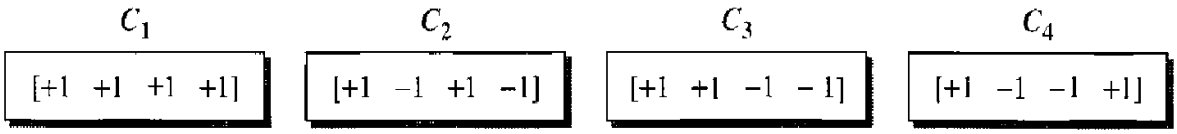
#### *Idea*

Let us assume we have four stations 1, 2, 3, and 4 connected to the same channel. The data from station 1 are $d_1$, from station 2 are $d_2$, and so on. The code assigned to the first station is $c_1$, to the second is $c_2$, and so on. We assume that the assigned codes have two properties.

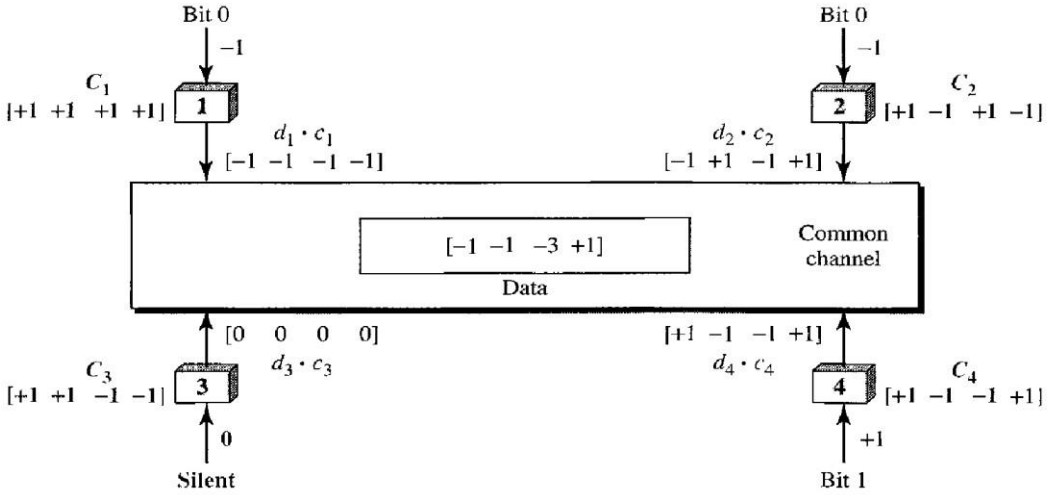
1. If we multiply each code by another, we get 0.
2. If we multiply each code by itself, we get 4 (the number of stations).




Chip sequences




**Sharing channel in CDMA**




1)

Find the chips for a network with

a. Two stations

b. Four stations

**Solution**

We can use the rows of $W_2$ and $W_4$ in Figure 12.29:

a. For a two-station network, we have [+1 +1] and [+1 −1].

b. For a four-station network we have [+1 +1 +1 +1], [+1 −1 +1 −1], [+1 +1 −1 −1], and [+1 −1 −1 +1].

2)

What is the number of sequences if we have 90 stations in our network?

**Solution**

The number of sequences needs to be $2^m$. We need to choose $m = 7$ and $N = 2^7$ or 128. We can then use 90 of the sequences as the chips.

3)

Prove that a receiving station can get the data sent by a specific sender if it multiplies the entire data on the channel by the sender's chip code and then divides it by the number of stations.

**Solution**

Let us prove this for the first station, using our previous four-station example. We can say that the data on the channel $D = (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4)$. The receiver which wants to get the data sent by station 1 multiplies these data by $c_1$.

$$\begin{aligned} D \cdot c_1 &= (d_1 \cdot c_1 + d_2 \cdot c_2 + d_3 \cdot c_3 + d_4 \cdot c_4) \cdot c_1 \\ &= d_1 \cdot c_1 \cdot c_1 + d_2 \cdot c_2 \cdot c_1 + d_3 \cdot c_3 \cdot c_1 + d_4 \cdot c_4 \cdot c_1 \\ &= d_1 \times N + d_2 \times 0 + d_3 \times 0 + d_4 \times 0 \\ &= d_1 \times N \end{aligned}$$

When we divide the result by $N$, we get $d_1$.

## UNIT - III

**1.** a) Explain how collision are avoided through use of 'IFS contention window and acknowledgment' in CSMA/CA. With the help of flow chart show the procedure for CSMA/CA **(July 2013 10marks)**

b) Explain 'Token Passing' method of controlled access of the channel **(July 2013 06 marks)**

c) A slotted ALOHA network transmit 200 bit frames using a shared channel with 200kbps bandwidth. Find the throughput if the system produces i) 1000 frames per second ii) 500 frames per second iii) 250 frames per second **(July 2013/Dec 2010 04 marks)**

**2.** a )Compare pure ALOHA with slotted ALOHA. What are the reasons for poor channel utilization in ALOHA system? How the same is improved in CSMA **(Jan 2013 10marks)**

**b)** Explain the working of CSMA/CD. Suppose a point to point link is set up between earth and a rover on MARS. The distance from earth to MARS is approximately 55Gm and data

travels over the link at a speed of light 3*10^8 m/s. Calculate the minimum round trip propagation time **(Jan 2013 10marks)**

**3.** a) With a suitable flow diagram explain CSMA/CD protocol and discuss the frame transmission time **(July 2012 08marks)**

b) Explain the following controlled access methods:

i) Reservation ii) polling iii) Token passing **(July 2012/2011 08marks)**

c) Show that the throughput for pure ALOHA is $S=Ge^{-2G}$ and maximum throughput $S_{max}=0.184$ **(July 2012 04marks)**

**4.** a) Define random access method explain three different protocol in this category **(July 2011 10 marks)**

b) Write the different physical topologies used in the logical ring method and explain briefly. **(Dec 2011 10 marks)**

**5.** a) In CSMA/CD, the data rate is 20Mbps, the distance between the situations 'A' and 'C' is 3000m and propagation is $2 * 10^8$ mts. Station A starts sending a long frame at time $t_1=0$; station C starts sending a long frame at $t_2$ =4 micro sec. The size of the frame is long enough to guarantee the detection of collision by the stations.

Find:

i) The time when station 'C' hears the collision ($t_3$)

i i) The time when station 'A' hears the collision ($t_4$)

iii) The number of bits station A has sent before detecting the collision

iv) The number of bits station C has sent before detecting the collision **(Dec 2011 10 marks)**

**b)** Explain pure ALOHA protocol **(Dec 2010 06 marks)**

**c)** A slotted ALOHA network transmits 200 bit frames using a shared channel with 200k bandwidth. Find the throughput if the system produces 500 frames/sec. **(June 2010 04 marks)**

**6.** a) A network using CSMA/CD has a bandwidth of 10Mbps. If the maximum propagation time is 25.6µsec, what is the minimum size of a frame? **(June 2010 04marks)**

b) What is channelization in the context of multiple access? What are the various available channelization techniques? List the properties of orthogonal sequences used in CDMA.

**(June 2010 10 marks)**

**7.** a) With a flow diagram explain CSMA/CD **(Dec 2014 8marks)**

b) What is channelization? Give brief account on CDMA (**Dec 2014 8marks)**

**8.** a) A pure ALOHA transmits 200 bit frames on shared channel of 200kbps what is the throughput if system produces i)1000 frames per second ii) 500 frames per second**(June 2014 10marks )**

b) Expalin 1 persistent,non perisistent and p-persistent with flow diagram,(**June 2014 10marks**

**9.** a) What are the reaons for poor channel utilization in ALOHA systems?How the same is improved in CSMA **(Dec 2014 10marks)**

b) Expalin polling as controlled access technique ,(**June 2014 10marks )**

**10.** a) Explain token passing method of controlled access of the channel. **(June 2010 10 marks)**

b) Discuss the concepts of i) 1-persistent CSMA ii) Non-persistent CSMA **(Jan 2013 10marks)**

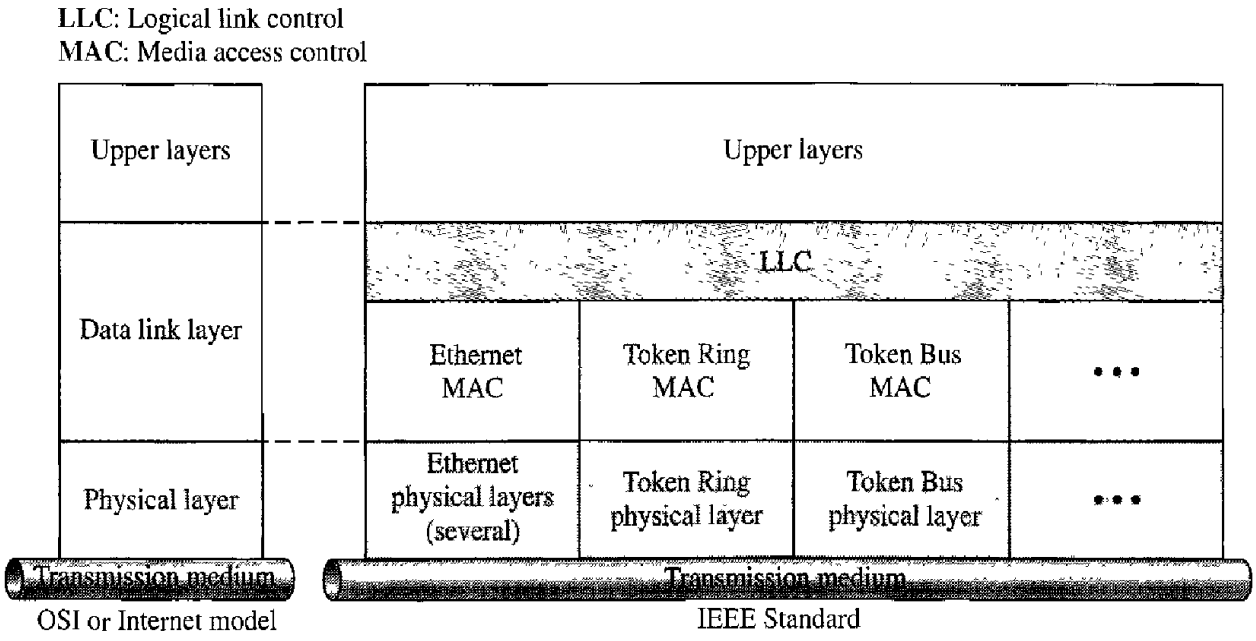
**Unit 4:**

## 4.1 IEEE Standards

In 1985, the Computer Society of the IEEE started a project, called **Project 802,** to set standards to enable intercommunication among equipment from a variety of manufacturers. Project 802 does not seek to replace any part of the OSI or the Internet model. Instead, it is a way of specifying functions of the physical layer and the data link layer of major LAN protocols.

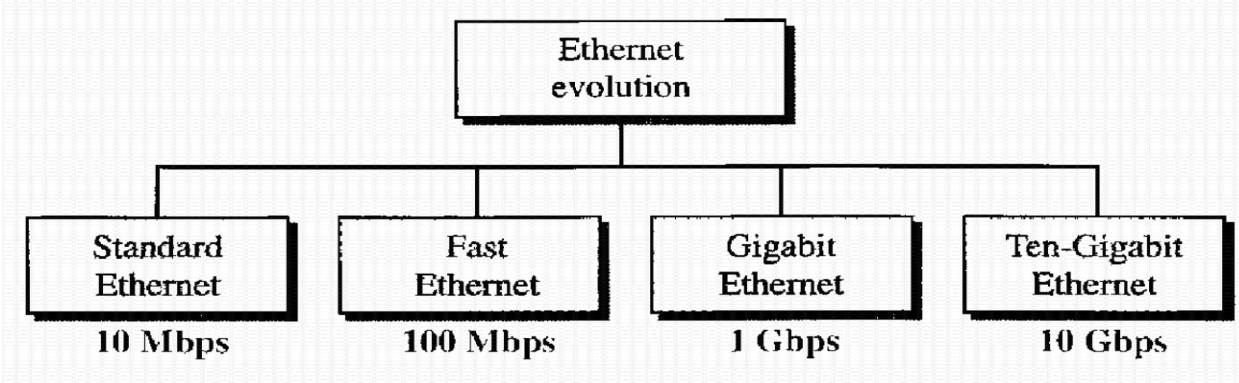
The standard was adopted by the American National Standards Institute (ANSI). In 1987, the International Organization for Standardization (ISO) also approved it as an international standard under the designation ISO 8802.

The relationship of the 802 Standard to the traditional OSI model is shown in Figure 13.1. The IEEE has subdivided the data link layer into two sublayers: **logical link control (LLC)** and **media access control (MAC).** IEEE has also created several physical layer standards for different LAN protocols.




## 4.2 Standard Ethernet

The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations: **Standard Ethernet** ($10^{\dagger}$ Mbps), **Fast Ethernet** (100 Mbps), **Gigabit Ethernet** (1 Gbps), and **Ten-Gigabit Ethernet** (10 Gbps), as shown in Figure 13.3. We briefly discuss all these generations starting with the first, Standard (or traditional) Ethernet.




## MAC Sublayer

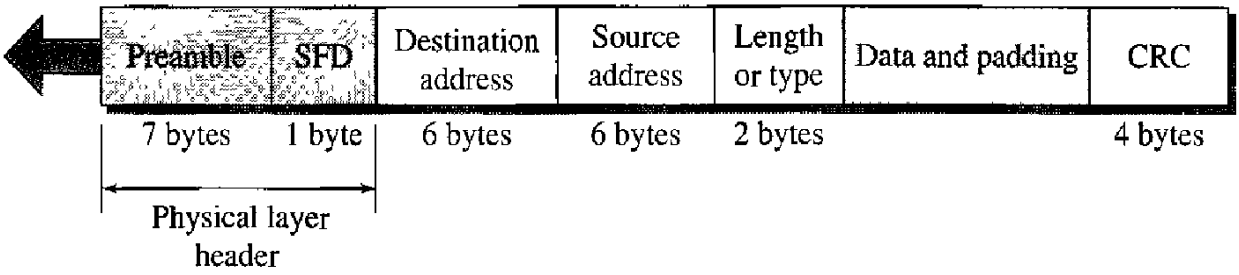
In Standard Ethernet, the MAC sublayer governs the operation of the access method. It also frames data received from the upper layer and passes them to the physical layer.

### *Frame Format*

The Ethernet frame contains seven fields: preamble, SFD, DA, SA, length or type of protocol data unit (PDU), upper-layer data, and the CRC. Ethernet does not provide any mechanism for acknowledging received frames, making it what is known as an unreliable medium. Acknowledgments must be implemented at the higher layers. The format

**Preamble**: 56 bits of alternating 1s and 0s.

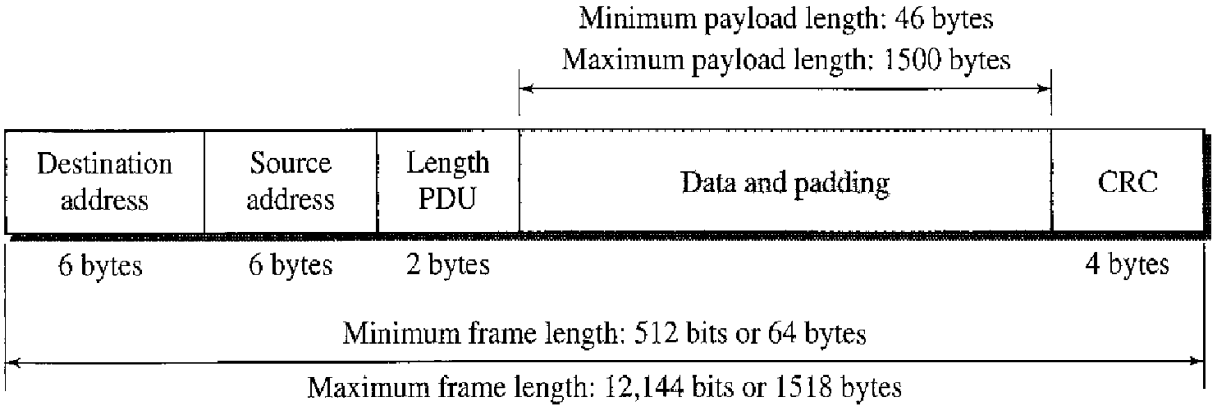
**SFD**: Start frame delimiter, flag (10101011)



- **Preamble.** The first field of the 802.3 frame contains 7 bytes (56 bits) of alternating 0s and 1s that alerts the receiving system to the coming frame and enables it to synchronize its input timing. The pattern provides only an alert and a timing pulse. The 56-bit pattern allows the stations to miss some bits at the beginning of the frame. The **preamble** is actually added at the physical layer and is not (formally) part of the frame.
- **Start frame delimiter (SFD).** The second field (1 byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is the last chance for synchronization. The last 2 bits is 11 and alerts the receiver that the next field is the destination address.
- **Destination address (DA).** The DA field is 6 bytes and contains the physical address of the destination station or stations to receive the packet. We will discuss addressing shortly.
- **Source address (SA).** The SA field is also 6 bytes and contains the physical address of the sender of the packet. We will discuss addressing shortly.
- **Length or type.** This field is defined as a type field or length field. The original Ethernet used this field as the type field to define the upper-layer protocol using the MAC frame. The IEEE standard used it as the length field to define the number of bytes in the data field. Both uses are common today.
- **Data.** This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes, as we will see later.
- **CRC.** The last field contains error detection information, in this case a CRC-32

### *Frame Length*

Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame,



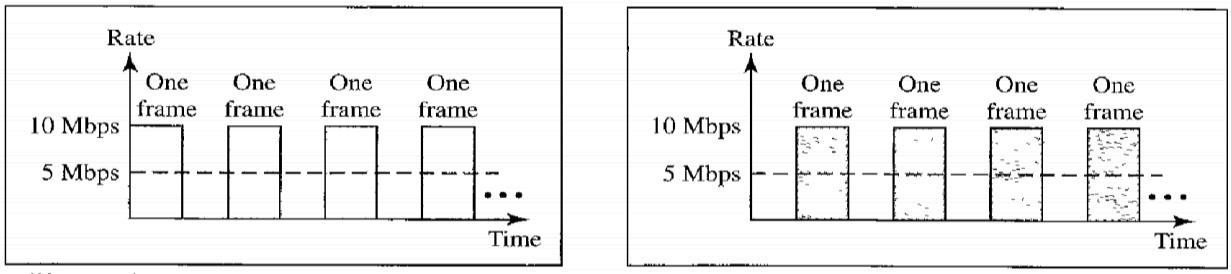
The 10-Mbps Standard Ethernet has gone through several changes before moving to the higher data rates. These changes actually opened the road to the evolution of the Ethernet to become compatible with other high-data-rate LANs. We discuss some of these changes in this section.

## Bridged Ethernet

The first step in the Ethernet evolution was the division of a LAN by **bridges.** Bridges have two effects on an Ethernet LAN: They raise the bandwidth and they separate colli-

### *Raising the Bandwidth*

In an unbridged Ethernet network, the total capacity (10 Mbps) is shared among all stations with a frame to send; the stations share the bandwidth of the network. If only one station has frames to send, it benefits from the total capacity (10 Mbps). But if more than one station needs to use the network, the capacity is shared. For example, if two stations have a lot of frames to send, they probably alternate in usage. When one station is sending, the other one refrains from sending. We can say that, in this case, each station on average, sends at a rate of 5 Mbps. Figure 13.14 shows the situation.

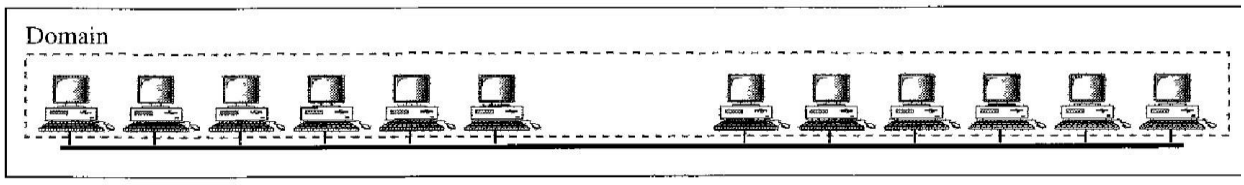



**a. First station** **b. Second station**

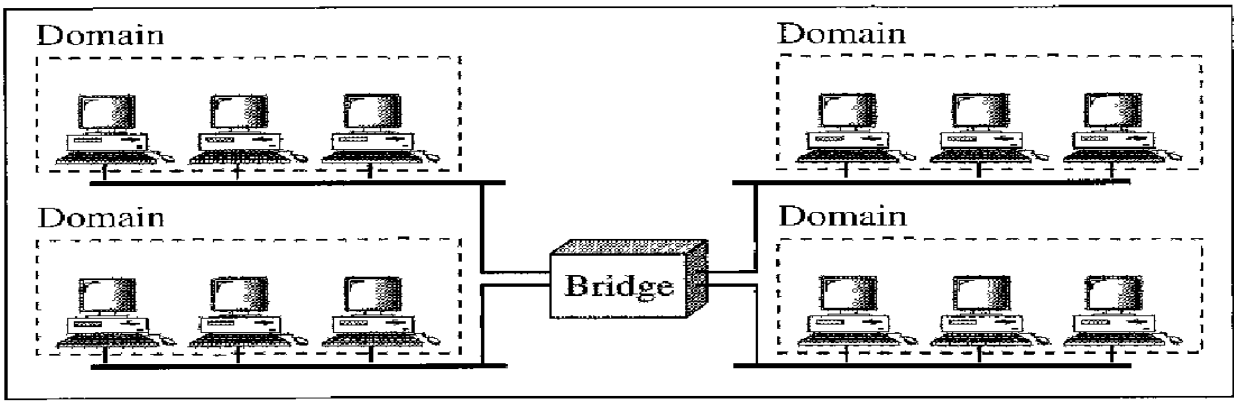
It is obvious that if we further divide the network, we can gain more bandwidth fo each segment. For example, if we use a four-port bridge, each station is now offere 10/3 Mbps, which is 4 times more than an unbridged network.

### *Separating Collision Domains*

Another advantage of a bridge is the separation of the **collision domain.** Figure 13.16 shows the collision domains for an unbridged and a bridged network. You can see that the collision domain becomes much smaller and the probability of collision is reduced tremendously. Without bridging, 12 stations contend for access to the medium; with bridging only 3 stations contend for access to the medium.



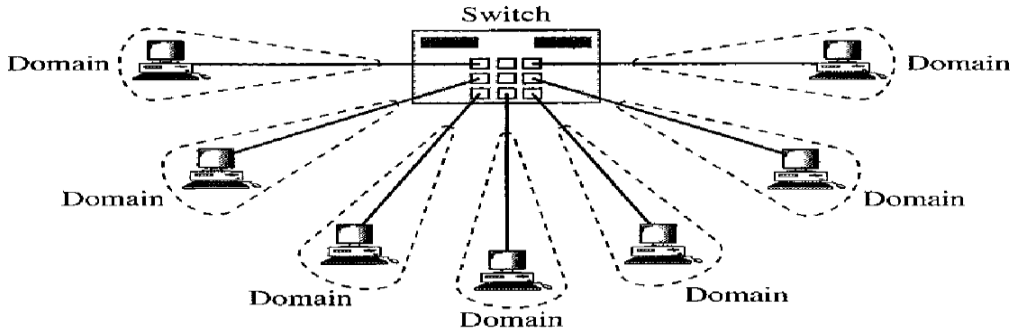

**a. Without bridging**




**b. With bridging**

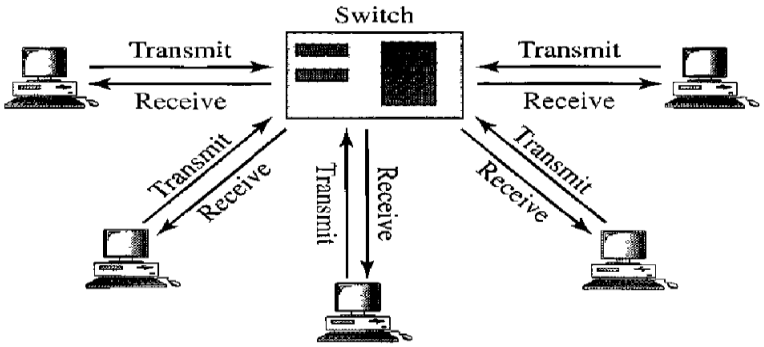
## Switched Ethernet

The idea of a bridged LAN can be extended to a switched LAN. Instead of having two to four networks, why not have *N* networks, where *N* is the number of stations on the LAN? In other words, if we can have a multiple-port bridge, why not have an *N*-port switch? In this way, the bandwidth is shared only between the station and the switch (5 Mbps each). In addition, the collision domain is divided into *N* domains.




## Full-Duplex Ethernet

One of the limitations of 10Base5 and 10Base2 is that communication is half-duplex (10Base-T is always full-duplex); a station can either send or receive, but may not do both at the same time. The next step in the evolution was to move from switched Ethernet to **full-duplex switched Ethernet.** The full-duplex mode increases the capacity of each domain from 10 to 20 Mbps. Figure 13.18 shows a switched Ethernet in full-duplex mode. Note that instead of using one link between the station and the switch, the configuration uses two links: one to transmit and one to receive.




## 4.4 Fast Ethernet:

Fast Ethernet was designed to compete with LAN protocols such as FDDI or Fiber Channel (or Fibre Channel, as it is sometimes spelled). IEEE created Fast Ethernet under the name 802.3u. Fast Ethernet is backward-compatible with Standard Ethernet, but it can transmit data 10 times faster at a rate of 100 Mbps. The goals of Fast Ethernet can be summarized as follows:

1. Upgrade the data rate to 100 Mbps.
2. Make it compatible with Standard Ethernet.
3. Keep the same 48-bit address.
4. Keep the same frame format.
5. Keep the same minimum and maximum frame lengths.

## MAC Sublayer

A main consideration in the evolution of Ethernet from 10 to 100 Mbps was to keep the MAC sublayer untouched. However, a decision was made to drop the bus topologies and keep only the star topology. For the star topology, there are two choices, as we saw before: half duplex and full duplex. In the half-duplex approach, the stations are connected via a hub; in the full-duplex approach, the connection is made via a switch with buffers at each port.

The access method is the same (CSMA/CD) for the half-duplex approach; for full-duplex Fast Ethernet, there is no need for CSMA/CD. However, the implementations keep CSMA/CD for backward compatibility with Standard Ethernet.

### *Autonegotiation*

A new feature added to Fast Ethernet is called **autonegotiation.** It allows a station or a hub a range of capabilities. Autonegotiation allows two devices to negotiate the mode

or data rate of operation. It was designed particularly for the following purposes:

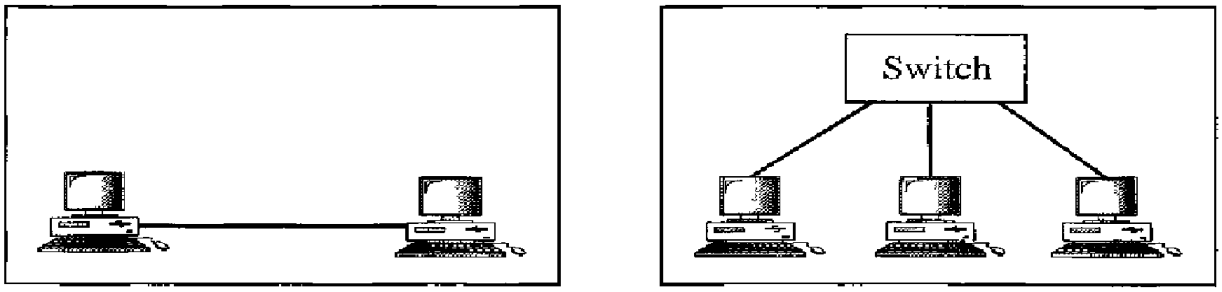
- To allow incompatible devices to connect to one another. For example, a device with a maximum capacity of 10 Mbps can communicate with a device with a 100 Mbps capacity (but can work at a lower rate).
- To allow one device to have multiple capabilities.
- To allow a station to check a hub's capabilities.

## Physical Layer

The physical layer in Fast Ethernet is more complicated than the one in Standard Ethernet. We briefly discuss some features of this layer.

### *Topology*

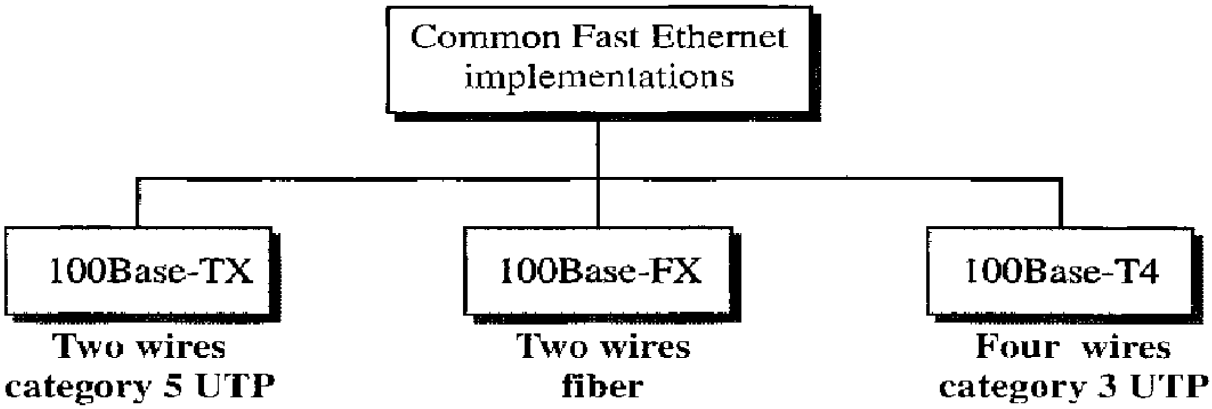
Fast Ethernet is designed to connect two or more stations together. If there are only two stations, they can be connected point-to-point. Three or more stations need to be connected in a star topology with a hub or a switch at the center, as shown in Figure 13.19.




**a. Point-to-point** **b. Star**

### *Implementation*

Fast Ethernet implementation at the physical layer can be categorized as either two-wire or four-wire. The two-wire implementation can be either category 5 UTP (100Base-TX) or fiber-optic cable (100Base-FX). The four-wire implementation is designed only for category 3 UTP (100Base-T4). See Figure 13.20.




## 4.5 Gigabit Ethernet:

The need for an even higher data rate resulted in the design of the Gigabit Ethernet protocol (1000 Mbps). The IEEE committee calls the Standard 802.3z. The goals of the Gigabit Ethernet design can be summarized as follows:

1. Upgrade the data rate to 1 Gbps.
2. Make it compatible with Standard or Fast Ethernet.
3. Use the same 48-bit address.
4. Use the same frame format.
5. Keep the same minimum and maximum frame lengths.
6. To support autonegotiation as defined in Fast Ethernet.

### MAC Sublayer

A main consideration in the evolution of Ethernet was to keep the MAC sublayer untouched. However, to achieve a data rate 1 Gbps, this was no longer possible. Gigabit Ethernet has two distinctive approaches for medium access: half-duplex and full-duplex. Almost all implementations of Gigabit Ethernet follow the full-duplex approach. However, we briefly discuss the half-duplex approach to show that Gigabit Ethernet can be compatible with the previous generations.

### *Full-Duplex Mode*

In full-duplex mode, there is a central switch connected to all computers or other switches. In this mode, each switch has buffers for each input port in which data are stored until they are transmitted. There is no collision in this mode, as we discussed before. This means that CSMA/CD is not used. Lack of collision implies that the maximum length of the cable is determined by the signal attenuation in the cable, not by the collision detection process.

### *Half-Duplex Mode*

Gigabit Ethernet can also be used in half-duplex mode, although it is rare. In this case, a switch can be replaced by a hub, which acts as the common cable in which a collision might occur. The half-duplex approach uses CSMA/CD. However, as we saw before, the maximum length of the network in this approach is totally dependent on the minimum frame size. Three methods have been defined: traditional, carrier extension, and frame bursting.

**Traditional** In the traditional approach, we keep the minimum length of the frame as in traditional Ethernet (512 bits). However, because the length of a bit is 1/100 shorter in Gigabit Ethernet than in 10-Mbps Ethernet, the slot time for Gigabit Ethernet is 512 bits × 1/1000 μs, which is equal to 0.512 μs. The reduced slot time means that collision is detected 100 times earlier. This means that the maximum length of the network is 25 m. This length may be suitable if all the stations are in one room, but it may not even be long enough to connect the computers in one single office.

**Carrier Extension** To allow for a longer network, we increase the minimum frame length. The **carrier extension** approach defines the minimum length of a frame as 512 bytes (4096 bits). This means that the minimum length is 8 times longer. This method forces a station to add extension bits (padding) to any frame that is less than 4096 bits. In this way, the maximum length of the network can be increased 8 times to a length of 200 m. This allows a length of 100 m from the hub to the station.

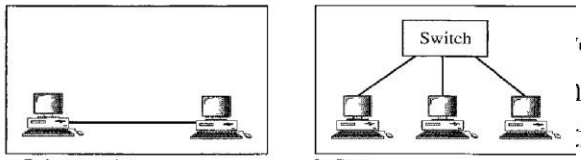
**Frame Bursting** Carrier extension is very inefficient if we have a series of short frames to send; each frame carries redundant data. To improve efficiency, **frame bursting** was proposed. Instead of adding an extension to each frame, multiple frames are sent. However, to make these multiple frames look like one frame, padding is added between the frames (the same as that used for the carrier extension method) so that the channel is not idle. In other words, the method deceives other stations into thinking that a very large frame has been transmitted.

## Physical Layer

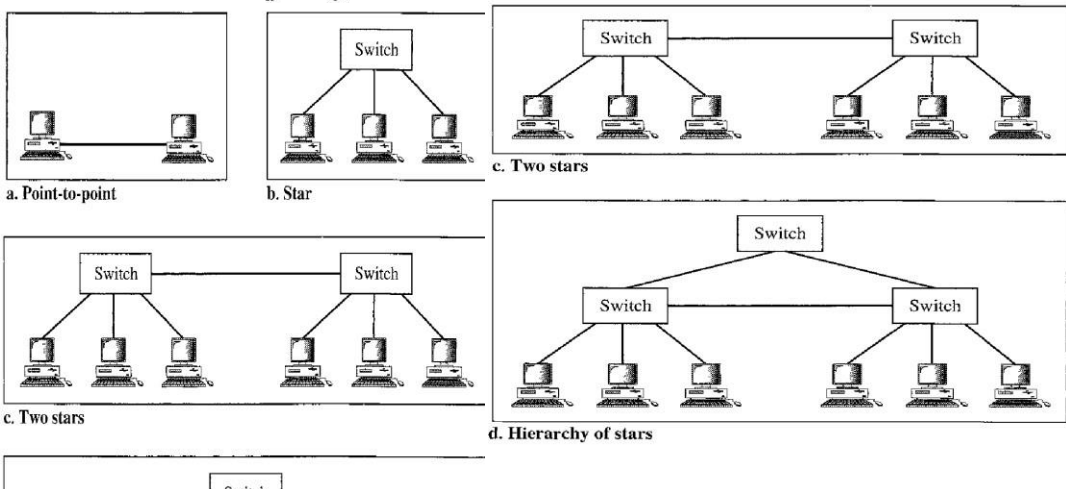
The physical layer in Gigabit Ethernet is more complicated than that in Standard or Fast Ethernet. We briefly discuss some features of this layer.

### *Topology*

Gigabit Ethernet is designed to connect t o sta-
tions, they can be connected point-to-poi ected
in a star topology with a hub or a switch ion is
to connect several star topologies or let wn in



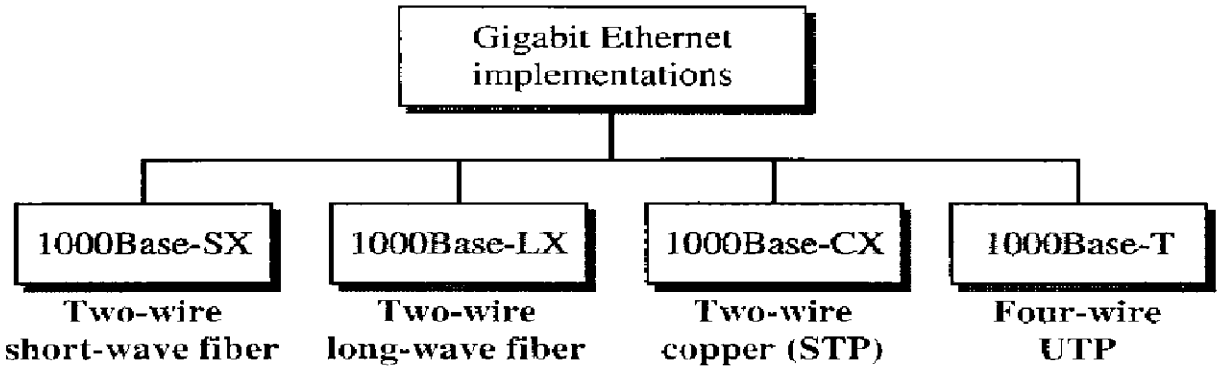

a. Point-to-point b. Star




a. Point-to-point b. Star c. Two stars c. Two stars d. Hierarchy of stars

### *Implementation*

Gigabit Ethernet can be categorized as either a two-wire or a four-wire implementation. The two-wire implementations use fiber-optic cable (**1000Base-SX, short-wave,** or **1000Base-LX, long-wave**), or STP (**1000Base-CX**). The four-wire version uses category 5 twisted-pair cable (**1000Base-T**). In other words, we have four implementations, as shown in Figure 13.23. 1000Base-T was designed in response to those users who had already installed this wiring for other purposes such as Fast Ethernet or telephone services.




| *Characteristics* | *1000Base-SX* | *1000Base-LX* | *1000Base-CX* | *1000Base-T* |
|---|---|---|---|---|
| Media | Fiber short-wave | Fiber long-wave | STP | Cat 5 UTP |
| Number of wires | 2 | 2 | 2 | 4 |
| Maximum length | 550 m | 5000 m | 25 m | 100 m |
| Block encoding | 8B/10B | 8B/10B | 8B/10B | |
| Line encoding | NRZ | NRZ | NRZ | 4D-PAM5 |

## UNIT - IV

**1.** a) List the goals of fast Ethernet . Explain the features of physical layer in fast ethernet **(July 2013 10marks)**

b) Explain two different kinds of services as defined in IEEE 802.11 **(July 2013 06 marks)**

c) Write a note on Piconet and Scatternet in Bluetooth **(July 2013 04 marks)**

**2.** a) Compare the data rates for standard ethernet, fast ethernet, gigabit Ethernet and ten gigabit ethernet **(Jan 2013/july 2011 04 marks)**

b) What is the difference between a unicast , multicast and broad cast address? Define the type of the following destination addresses:

i) 4A:30:10:21:10:1A

ii) 47:20:1B:2E:08:EE

iii) FF:FF:FF:FF:FF:FF **(Jan 2013 08marks)**

c) Explain the following with respect to Fast Ethernet:

i) Implementation ii) Encoding iii) 100 Base-TX iv) 100 Base-FX **(Jan 2013 08 marks)**

**3.** a) With a neat diagram, explain 802.3 MAC frame format **(July 2012 10marks)**

b) Explain the following standard Ethernet physical layer implementation

i) 10 base 5: thick Ethernet

ii) 10 base 2: thick Ethernet

iii) 10 base T: twisted pair Ethernet

iv) 10 base F: Fiber Ethernet **(July 2012 10marks)**

**4.** a)Explain DCP and PCF modes of 802.11 MAC protocol **(July 2011 10 marks)**

**b)** What are the advantages of dividing an Ethernet LAN with a bridge? Explain with neat diagram **(July 2011 04 marks)**

**c)** Mention the four different types of Ethernet format. Explain the same briefly.

**(Dec 2011 6 marks)**

**5.** a) List the different goals of giga bit Ethernet and explain the different implementation of same.

**(Dec 2011 10 marks)**

**b)** Explain the goals, MAC sub layer and physical layer of fast Ethernet **(Dec 2010 10 marks)**

**6.** a) Explain briefly the baseband layer in Bluetooth layers. **(Dec 2010 10 marks)**

**b)** Explain 802.3 MAC frame format. **(June 2010 10 marks)**

**7.** a) An Ethernet MAC sublayer receives 38 bytes of data from upper layer. How many bytes of padding must be added to the data? **(June 2010 10 marks)**

**b)** What is fast Ethernet? Explain auto negotiation. What are the purposes of using this feature in design of fast Ethernet? **(June 2010 10 marks)**

**8.** a) Explain the frame format of 802.3 MAC frame (**June 2014 6marks)**

**b)Define the type of the following destination address and justify the answer**

**i)4A:30:10:21:10:1A ii)47:20:1B:2E:8:EE (June 2014 4 marks)**

c) Explain bridge Ethernet ,switched Ethernet and full duplex Ethernet **(June 2014 10marks)**

**9.** a) What are the reasons for not implementing CSMA/CD in wireless LANs?With a flow chart and frame exchange time line diagram explain CSMA/CA **(Dec 2014 10marks)**

**b)** What are the advantages of having bridged Ethernet **(Dec 2014 10marks)**

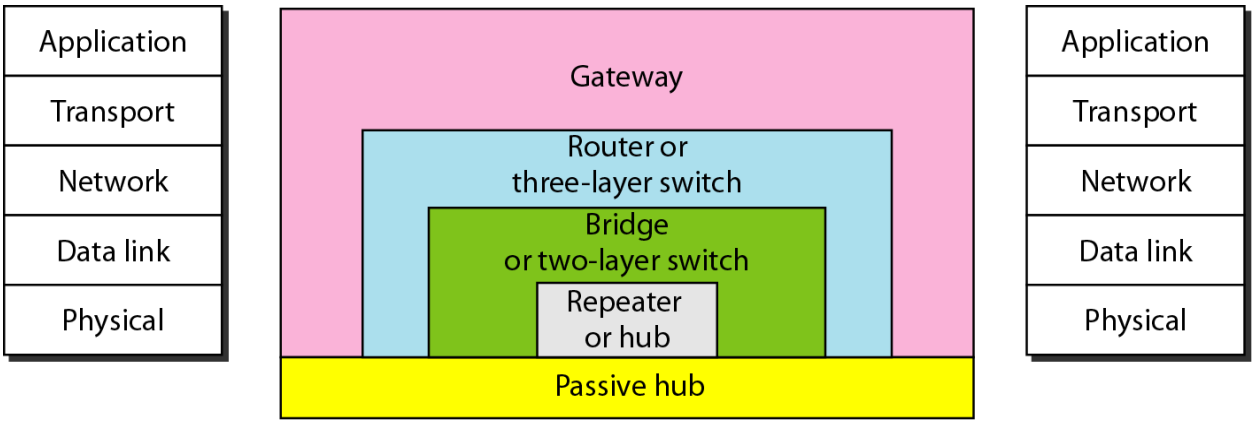
**10.** a) List the goals of the fast Ethernet ?Enumerate fast Ethernet implementations **(Dec 2014 10marks)**

**b)** Explain two different kinds of services as defined in IEEE 802.11. **(June 2010 10 marks)**

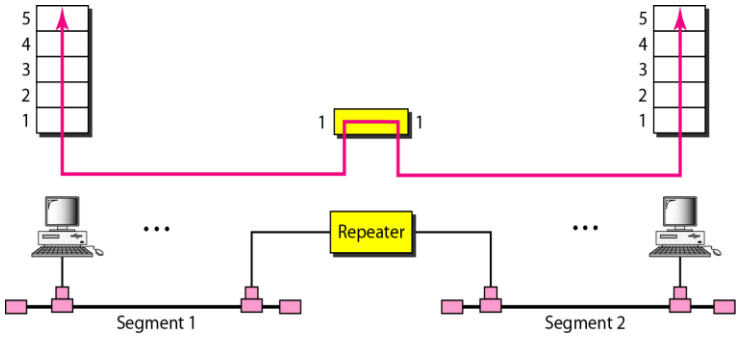
# PART B

## Unit 5:

### 5.1 Connecting devices:



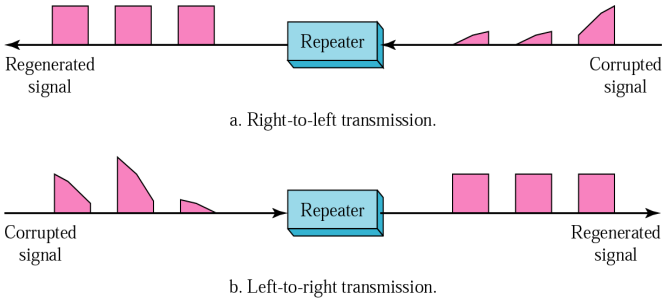

A repeater connecting two segments of a LAN



## *PASSIVE HUB*

* A passive hub is just a connector.
* It connects the wires coming from different branches.
* Its location in the Internet model is below the physical layer

## *REPEATERS*

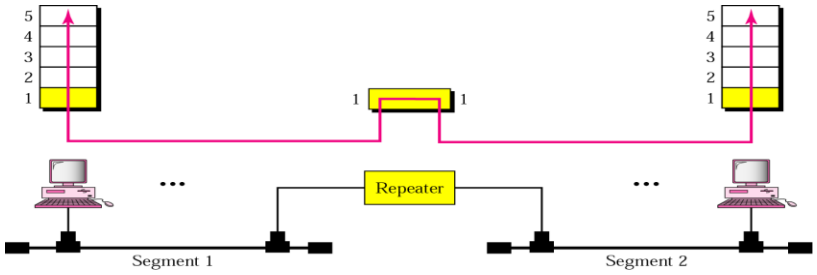



* Signals that carry information can travel a fixed distance & faces attenuation.
* A repeater receives a weakened signal, regenerates the original bit pattern and sends the refreshed signal.

## *Repeater Vs Amplifier*

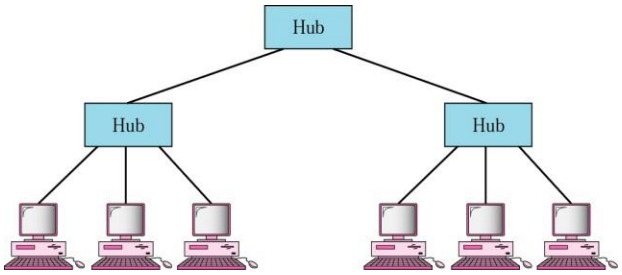
* An amplifier can't discriminate between the intended signal and noise, it amplifies equally everything fed into it.
* A repeater doesn't amplify the signal, it regenerates the signal.

## *REPEATERS*




* A repeater is a device that operates only in the physical layer.
* A repeater can extend the physical length of a LAN.
* Repeater doesn't connect two LANs, & of different protocols, it connects two segments of same LAN.
* The location of a repeater on a link is vital.
* Repeaters overcome the restriction of 10Base5 Ethernet.

## *ACTIVE HUBS*

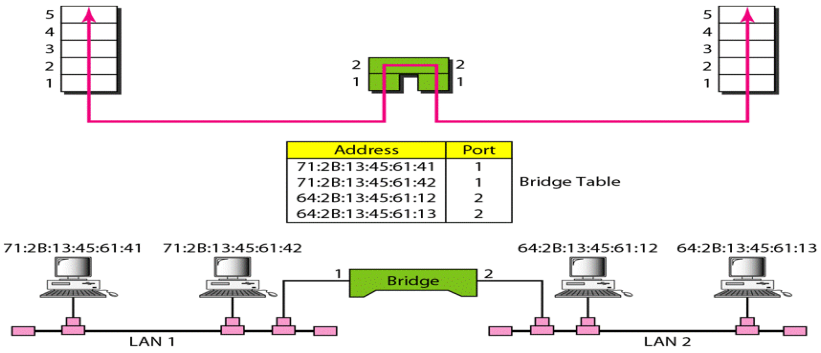



* An active hub is actually a multi-port repeater.
* It is normally used tocreate connections between stations in a physical star topology.
* Hubs can also be used to create multiple levels of hierarchy.
* The hierarchical use of hubs removes the length limitation of 10Base-T.

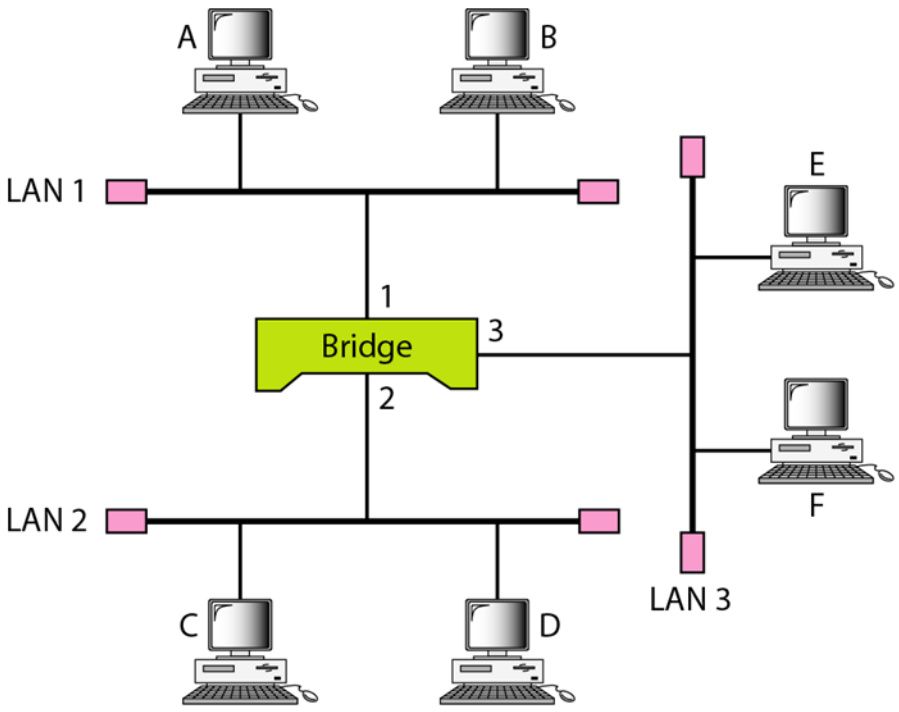
## *BRIDGES*

* A Bridge operates in both the Physical and the Data link layer.
* As a Physical layer device, it regenerates the signal it receives.
* As a data link layer device, the bridge can check the physical addresses contained in the frame.
* Compared to the repeaters, a BRIDGE has a filtering capability – forwarding & dropping of frames.
* If the frame is to be forwarded, the port must be specified.
* A bridge has a table used in filtering decisions.
* A bridge doesn't change the physical addresses contained in the frame.

***A bridge connecting two LANs***



***A learning bridge and the process of learning***




| Address | Port |
|---|---|
| | |

a. Original

| Address | Port |
|---|---|
| A | 1 |

b. After A sends a frame to D

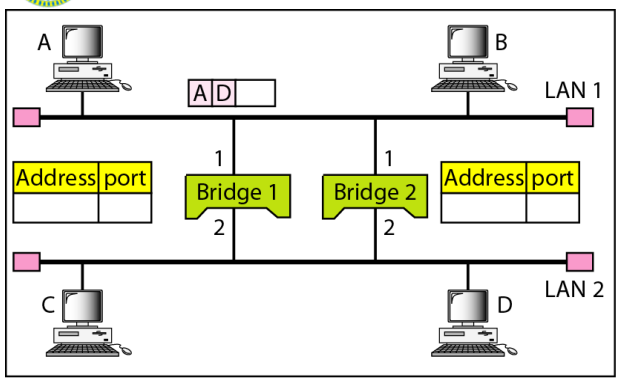
| Address | Port |
|---|---|
| A | 1 |
| E | 3 |

c. After E sends a frame to A

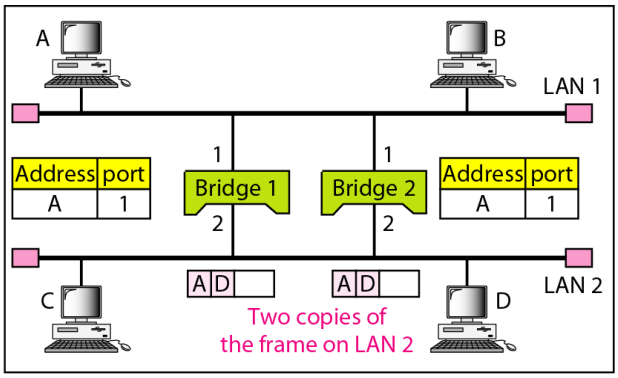
| Address | Port |
|---|---|
| A | 1 |
| E | 3 |
| B | 1 |

d. After B sends a frame to C

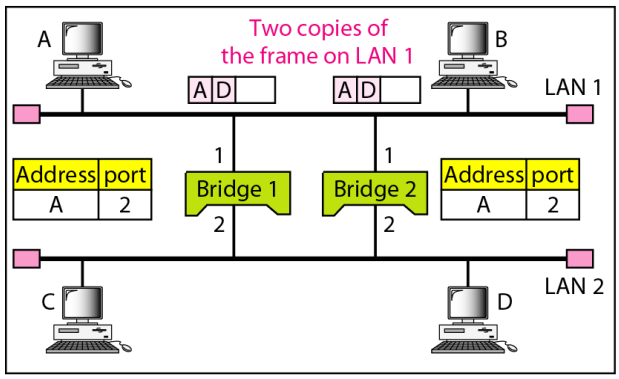
***Loop problem in a learning bridge***



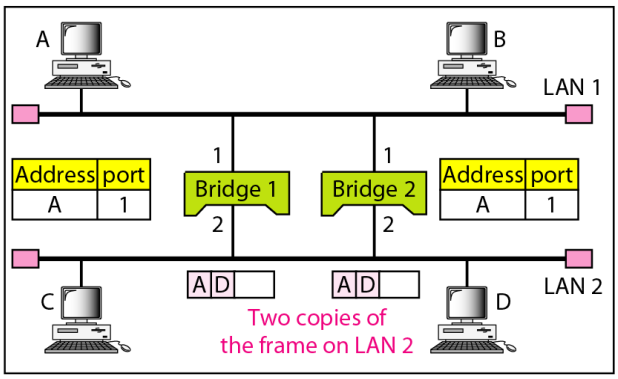

a. Station A sends a frame to station D




b. Both bridges forward the frame

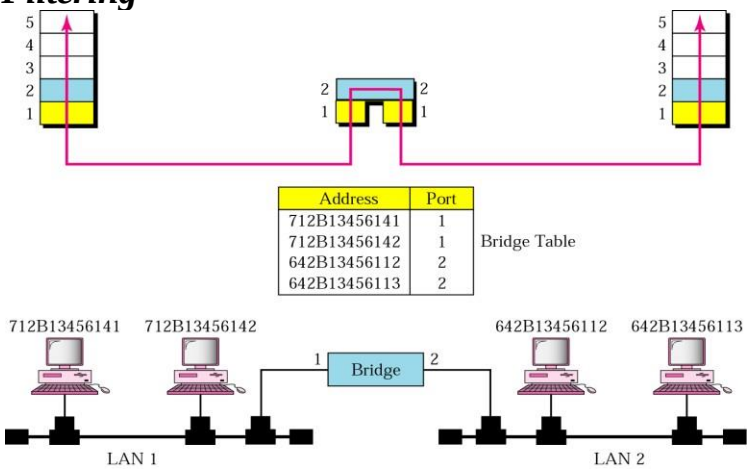



c. Both bridges forward the frame




d. Both bridges forward the frame

## *BRIDGES - Filtering*




## *Transparent bridges*

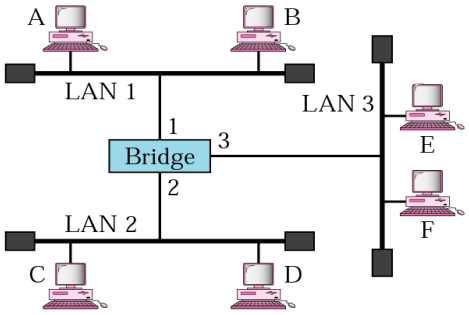
* A Transparent bridge is a bridge in which the stations are completely unaware of bridge's existence.
* If a bridge is added or deleted from the system, reconfiguration of the stations is

unnecessary.

* According to IEEE 802.1d, a system equipped with transparent bridges must meet three criteria.

1. Forwarding.
2. Learning.
3. Loops should not be present.

## *Learning Process*




| Address | Port |
|---|---|

a. Original

| Address | Port |
|---|---|
| A | 1 |

b. After A sends a frame to D

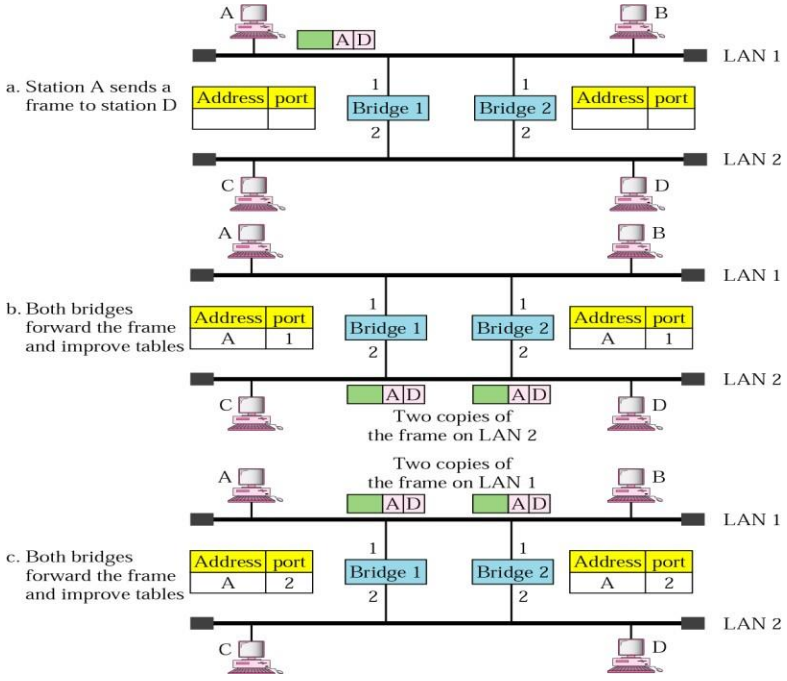
| Address | Port |
|---|---|
| A | 1 |
| E | 3 |

c. After E sends a frame to A

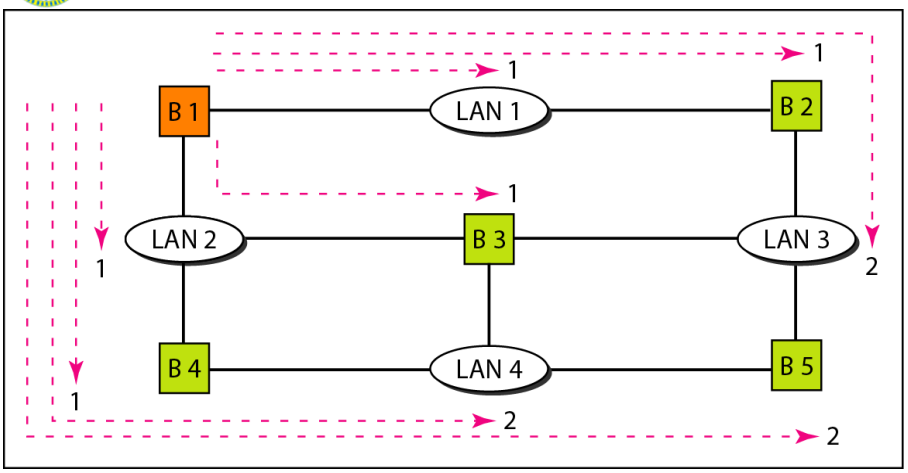
| Address | Port |
|---|---|
| A | 1 |
| E | 3 |
| B | 1 |

d. After B sends a frame to C

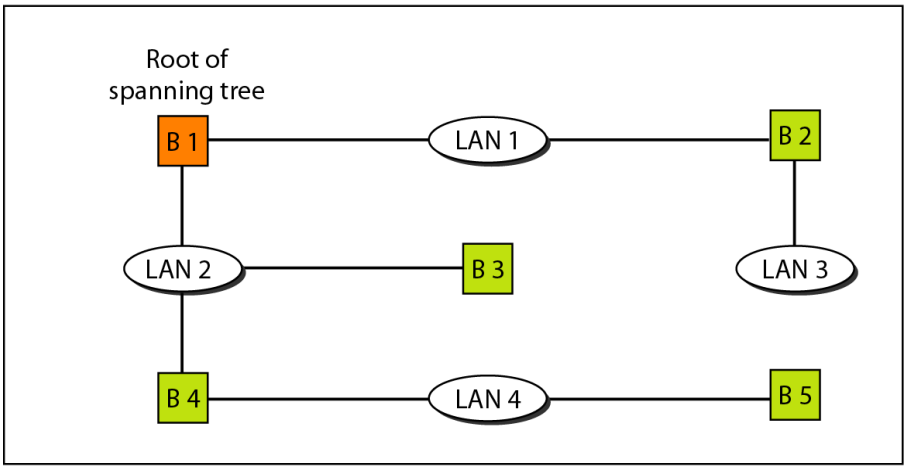
## *Loop Problem*




## *Finding the shortest paths and the spanning tree in a system of bridges*



a. Shortest paths



b. Spanning tree

## *SOURCE ROUTING BRIDGES*

* Alternate method to prevent Loops in the system with redundant bridges.
* Filtering frames is done by the source station, to some extent by the destination station also.
* A sending station defines the bridges that the frame must visit.
* The frames contain not only the source and destination addresses, but also the addresses of the bridges to be visited.
* The source gets the addresses through exchange of special frames with the destination prior to sending data frames.
* Ex.- Token ring LANs

### *Bridges connecting different LANs*

*Ex.- Ethernet LAN to a Wireless LAN*

* Frame Format.
* Maximum data size.
* Data rate.
* Bit order.
* Security.
* Multimedia support

## *TWO-LAYER SWITCHES*

* A Two-layer switch is a bridge with multiple ports and a design that allows better performance.
* As a bridge, two-layer switch makes a filtering decision based on physical address of the frame it received.
* It is more sophisticated – it can have a buffer to hold the frames for processing.
* Ex – Cut-through switches.

## *ROUTERS or THREE-LAYER SWITCH*

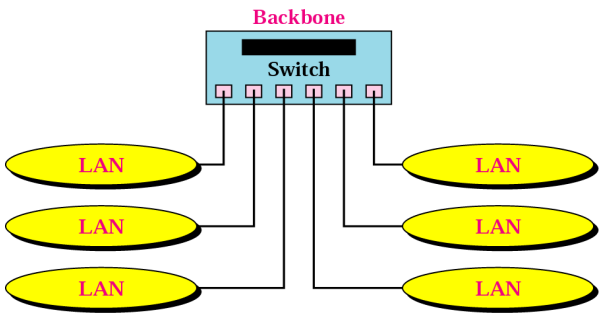
* A ROUTER is a three-layer device that routes packets based on their logical addresses.
* A router normally connects LANs & WANs in the internet and has a routing table that is used for making decisions about the route.
* the routing tables are normally dynamic and are updated using routing protocols.
** A Three-Layer switch is a router, but a faster and more sophisticated.

## *GATEWAY*

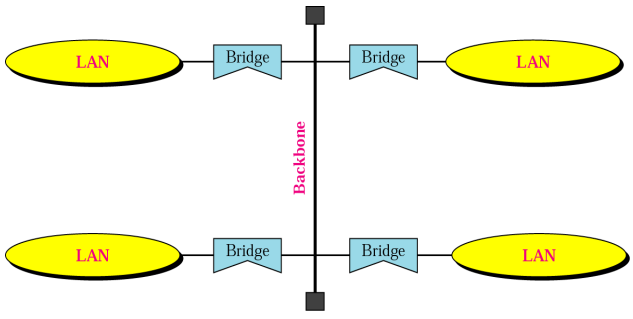
* A GATEWAY is normally a computer that operates in all five layers of the internet model and seven layers of the OSI model.
* A Gateway takes an application message, reads it and interprets it – means it can be used as a connecting device between two internetworks that use different models.
* A network designed to use the OSI model can be connected to another network using the internet model.
* Gateways can provide security.

## *5. 2 BACKBONE NETWORK*

* A Backbone network allows several LANs to be connected.
* The backbone is itself a LAN that uses a LAN protocol such as Ethernet.
* Different configurations of Backbone networks:
  1. Star backbone.
  2. Bus backbone.

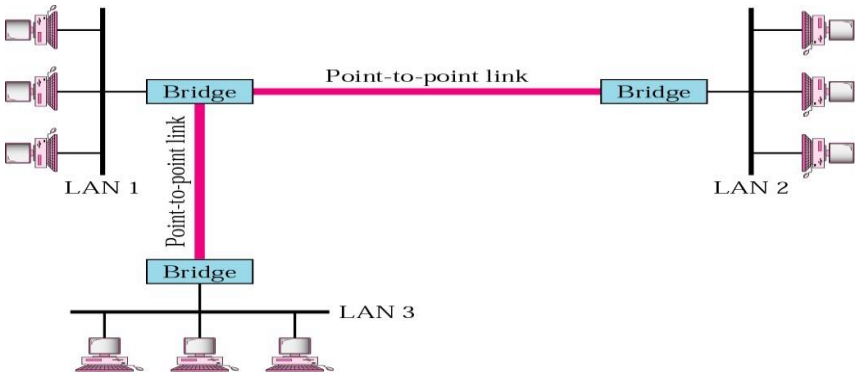



***1. Star Backbone***



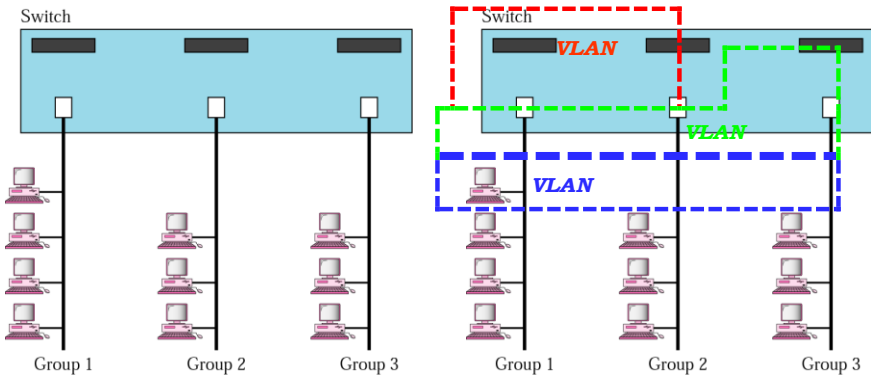

***2. Bus Backbone***

## *Connecting Remote LANs with bridges*

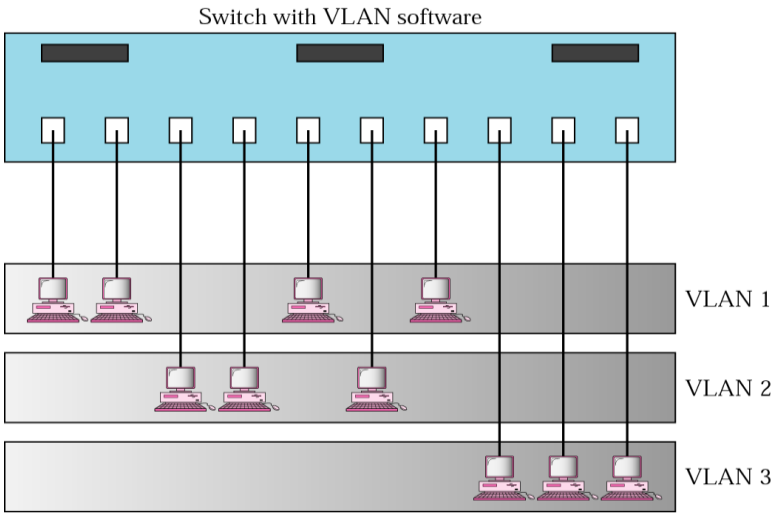



A point-to-point link acts as a LAN without stations in a remote backbone connected by remote bridges.

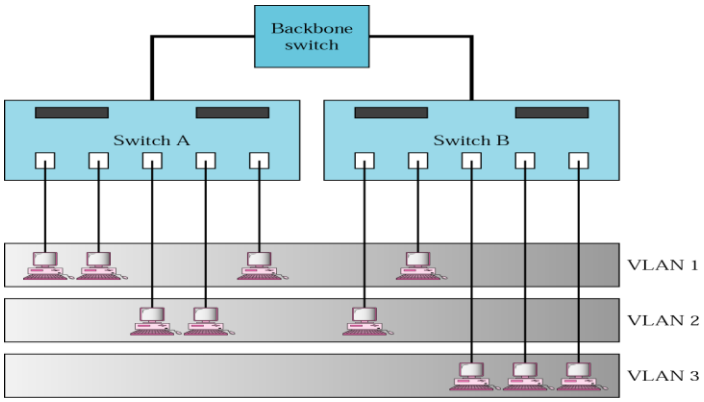
## *5.3 VIRTUAL LANs*




## *A switch using VLAN software*

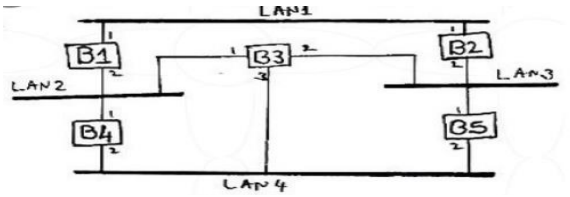


***Two switches in a backbone using VLAN software***



# UNIT - V

**1.** a) A system with four LANs and five bridges is shown in fig below. Choose B1 as the root bridge. Show the forwarding and blocking ports after applying the spanning tree procedure.




**(July 2013 10 marks)**

b) Define repeater, bridge and router with necessary diagrams **(July 2013/2011 06 marks)**

c) Differentiate between bus backbone network and star backbone network

**(July 2013 04marks)**

**2.** a) Explain the following connecting devices: i) Repeater ii) Bridge iii) Router iv) Gateway

**(Jan 2013 08 marks)**

**b)** What is spanning tree? Explain with suitable example **(Jan 2013 08 marks)**

**c)** What is VLAN? Explain **(Jan 2013 04 marks)**

**3.** a) Explain the following in brief**:** i) Passive hubs ii) Active hubs iii) Bridges iv) Router v) Gateway **(July 2012 10marks)**

**b)** Explain virtual LAN system and how the membership is allocated in V-LAN syste

**(July 2012 /Dec 2010 10marks)**

**4.** a) Create a system of three LANs with four bridges. The bridges(B1 to B4) connect the LANs as follows:

i) B1 connects LAN1 and LAN2

ii) B2 connects LAN1 and LAN3

iii) B3 connects LAN2 and LAN3

iv) B4 connects LAN1 ,LAN2 and LAN3

Chose B1 as the root bridge. Show the forwarding and blocking parts after applying the spanning tree procedure **(July 2011 10 marks)**

**b)** Why spanning tree algorithm is used? Explain the same, with a graphical representation. **(Dec 2011 10 marks)**

**5.** a) Mention the different characteristics of VLAN and explain briefly. **(Dec 2011 10 marks)**

**b)** Explain three criteria of transparent bridge **(Dec 2010 10 marks)**

**6.** a)What are the five different categories of connecting devices, based on the layer at which they operate in a network? Explain each of them. **(June 2010 10 marks)**

**c)** Differentiate between a bus backbone network and star backbone network. **(June 2010 06 marks)**

**d)** Explain the concept of VLAN, in brief. **(June 2010 04 marks)**

**7.** a)Explain what is loop problem and solution for a loop problem in a bridge with suitable diagram **(June 2014 10marks)**

b) What is VLAN ? Explain briefly **(June 2014 10marks)**

**8.** a)Explain backbone networks **(June 2014 10marks)**

b) Explain star backbone networks**(June 2014 10marks)**

**9.** a)Briefly explain three criteria of transparent bridge with example? (**Dec 2014 10marks)**

b)Explain the following connecting devices i)passive hub ii)reapeater (**Dec 2014 10marks)**

**10.** Explain the following connecting devices i)Bridge ii)router iii)gateway iv)passive hub **Dec 2014 20marks)**

**Unit 6:**

***An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet***

**Address Space**
**Notations**
**Classful Addressing**
**Classless Addressing**
**Network Address Translation (NAT)**

**An IPv4 address is 32 bits long.**
**The IPv4 addresses are unique and universal.**
**The address space of IPv4 is $2^{32}$ or 4,294,967,296**

## 6.1 IPv4 Addresses

An **IPv4 address** is a 32-bit address that *uniquely* and *universally* defines the connection of a device (for example, a computer or a router) to the Internet.

IPv4 addresses are unique. 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. We will see later that, by using some strategies, an address may be assigned to a device for a time period and then taken away and assigned to another device.

## Address Space

A protocol such as IPv4 that defines addresses has an **address space.** An address space is the total number of addresses used by the protocol. If a protocol uses $N$ bits to define an address, the address space is $2^N$ because each bit can have two different values (0 or 1) and $N$ bits can have $2^N$ values.

IPv4 uses 32-bit addresses, which means that the address space is $2^{32}$ or 4,294,967,296 (more than 4 billion). This means that, theoretically, if there were no restrictions, more than 4 billion devices could be connected to the Internet. We will see shortly that the actual number is much less because of the restrictions imposed on the addresses.

## 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. The following is an example of an IPv4 address in binary notation:

01110101 10010101 00011101 00000010

### *Dotted-Decimal Notation*

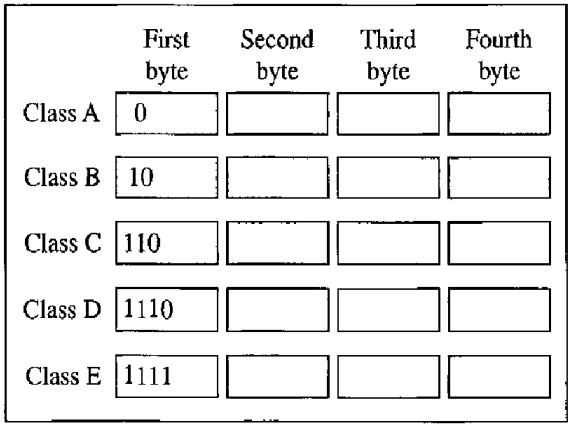
To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes. The following is the **dotted-decimal notation** of the above address:

117.149.29.2

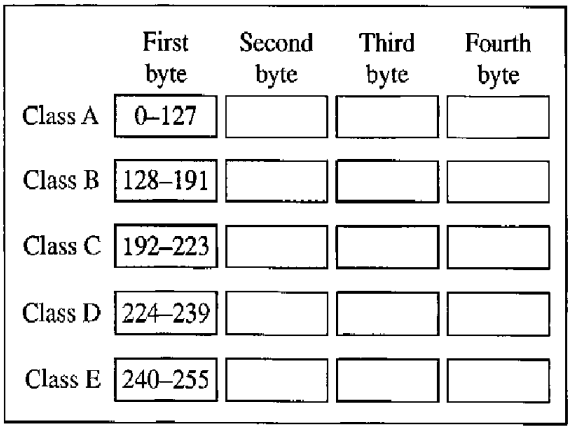
## Classful Addressing

IPv4 addressing, at its inception, used the concept of classes. This architecture is called **classful addressing.** Although this scheme is becoming obsolete, we briefly discuss it here to show the rationale behind classless 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.




**a. Binary notation**




**b. Dotted-decimal notation**

a)00000001 00001011 00001011 11101111 -A
b. 11000001 10000011 00011011 11111111-C
c. 14.23.120.8 -A
d. 252.5.15.111-E

## Classless Addressing

To overcome address depletion and give more organizations access to the Internet, **classless addressing** was designed and implemented. In this scheme, there are no classes, but the addresses are still granted in blocks.

### *Address Blocks*

In classless addressing, when an entity, small or large, needs to be connected to the Internet, it is granted a block (range) of addresses. The size of the block (the number of addresses) varies based on the nature and size of the entity. For example, a household may be given only two addresses; a large organization may be given thousands of addresses. An ISP, as the Internet service provider, may be given thousands or hundreds of thousands based on the number of customers it may serve.

**Restriction** To simplify the handling of addresses, the Internet authorities impose three restrictions on classless address blocks:

1. The addresses in a block must be contiguous, one after another.
2. The number of addresses in a block must be a power of 2 (1, 2, 4, 8, . . . ).
3. The first address must be evenly divisible by the number of addresses.

*Mask*

A better way to define a block of addresses is to select any address in the block and the mask. As we discussed before, a mask is a 32-bit number in which the *n* leftmost bits are 1s and the 32 – *n* rightmost bits are 0s. However, in classless addressing the mask for a block can take any value from 0 to 32. It is very convenient to give just the value of *n* preceded by a slash (CIDR notation).

---

**In IPv4 addressing, a block of addresses can be defined as**
**x.y.z.t /*n***
**in which x.y.z.t defines one of the addresses and the /*n* defines the mask.**

---

The address and the /*n* notation completely define the whole block (the first address, the last address, and the number of addresses).

**First Address** The first address in the block can be found by setting the 32 – *n* rightmost bits in the binary notation of the address to 0s.

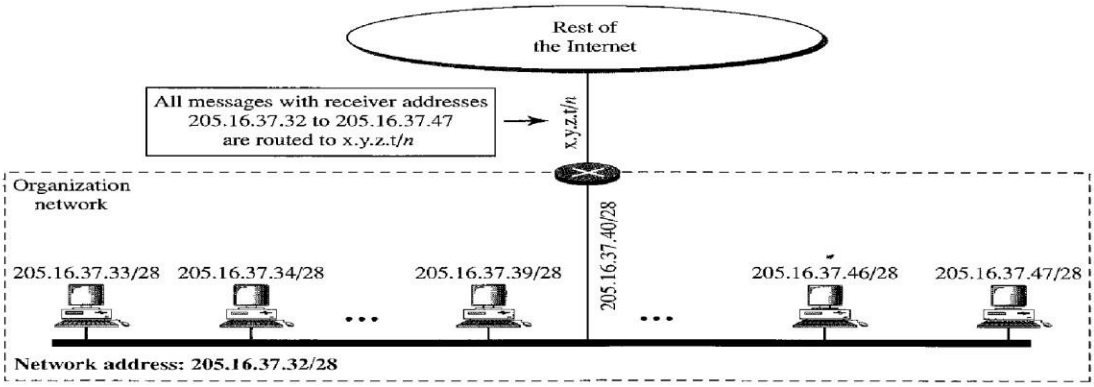
---

**The first address in the block can be found by setting the rightmost 32 – *n* bits to 0s.**

---

*Network Addresses*

A very important concept in IP addressing is the **network address.** When an organization is given a block of addresses, the organization is free to allocate the addresses to the devices that need to be connected to the Internet. The first address in the class, however, is normally (not always) treated as a special address. The first address is called the network address and defines the organization network. It defines the organization itself to the rest of the world. In a later chapter we will see that the first address is the one that is used by routers to direct the message sent to the organization from the outside.


Rest of
the Internet
All messages with receiver addresses
205.16.37.32 to 205.16.37.47
are routed to x.y.z.t/n
x.y.z.t/n
Organization
network
205.16.37.40/28
205.16.37.33/28
205.16.37.34/28
205.16.37.39/28
205.16.37.46/28
205.16.37.47/28
Network address: 205.16.37.32/28

## Network Address Translation (NAT)

The number of home users and small businesses that want to use the Internet is ever increasing. In the beginning, a user was connected to the Internet with a dial-up line, which means that she was connected for a specific period of time. An ISP with a block of addresses could dynamically assign an address to this user. An address was given to a user when it was needed. But the situation is different today. Home users and small businesses can be connected by an ADSL line or cable modem. In addition, many are not happy with one address; many have created small networks with several hosts and need an IP address for each host. With the shortage of addresses, this is a serious problem.

A quick solution to this problem is called **network address translation (NAT).** NAT enables a user to have a large set of addresses internally and one address, or a small set of addresses, externally. The traffic inside can use the large set; the traffic outside, the small set.

| *Range* | | | *Total* |
|---|---|---|---|
| 10.0.0.0 | to | 10.255.255.255 | $2^{24}$ |
| 172.16.0.0 | to | 172.31.255.255 | $2^{20}$ |
| 192.168.0.0 | to | 192.168.255.255 | $2^{16}$ |

## 6.2 IPv6 Addresses

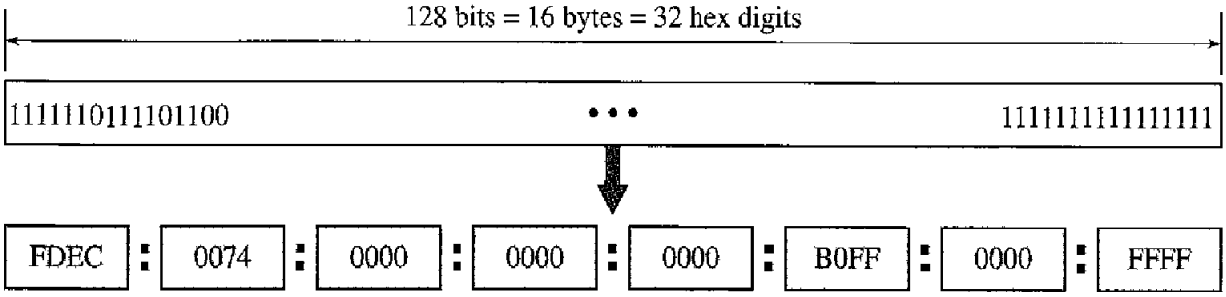
### Structure

An **IPv6 address** consists of 16 bytes (octets); it is 128 bits long.

**An IPv6 address is 128 bits long.**

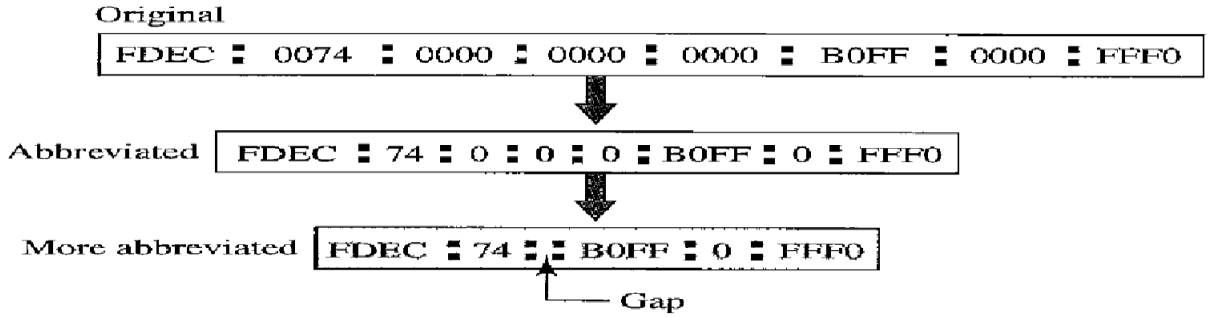
#### *Hexadecimal Colon Notation*

To make addresses more readable, IPv6 specifies **hexadecimal colon notation.** In this notation, 128 bits is divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal notation requires four hexadecimal digits. Therefore, the address consists of 32 hexadecimal




#### *Abbreviation*

Although the IP address, even in hexadecimal format, is very long, many of the digits are zeros. In this case, we can abbreviate the address. The leading zeros of a section (four digits between two colons) can be omitted. Only the leading zeros can be dropped, not the trailing zeros (see Figure 19.15).



Using this form of abbreviation, 0074 can be written as 74, OOOF as F, and 0000 as 0. Note that 3210 cannot be abbreviated. Further abbreviations are possible if there are consecutive sections consisting of zeros only. We can remove the zeros altogether and replace them with a double semicolon. Not that this type of abbreviation is allowed only once per address. If there are two runs of zero sections, only one of them can be abbreviated. Re expansion of the abbreviated address is very simple: Align the unabbre- viated portions and insert zeros to get the original expanded address.

## Address Space

1Pv6 has a much larger address space; 2128 addresses arc available. The designers of IPv6 divided the address into several categories. A few leftmost bits, called the type prefix, in each address define its category. The type prefix is variable in length, but it is designed such that no code is identical to the first part of any other code. In this way, there is no ambiguity ; when an address is given, the type prefix can easily be deter- mined. Table 19.5 shows the prefix for each type of address. The third column shows the fraction of each type of address relative to the whole address space.

| *Type Prefix* | *Type* | *Fraction* |
|---|---|---|
| 0000 0000 | Reserved | 1/256 |
| 0000 0001 | Unassigned | 1/256 |
| 0000 001 | ISO network addresses | 1/128 |
| 0000 010 | IPX (Novell) network addresses | 1/128 |
| 0000 011 | Unassigned | 1/128 |
| 0000 1 | Unassigned | 1/32 |
| 0001 | Reserved | 1/16 |
| 001 | Reserved | 1/8 |
| 010 | Provider-based unicast addresses | 1/8 |
| *Unicast Addresses* | | |

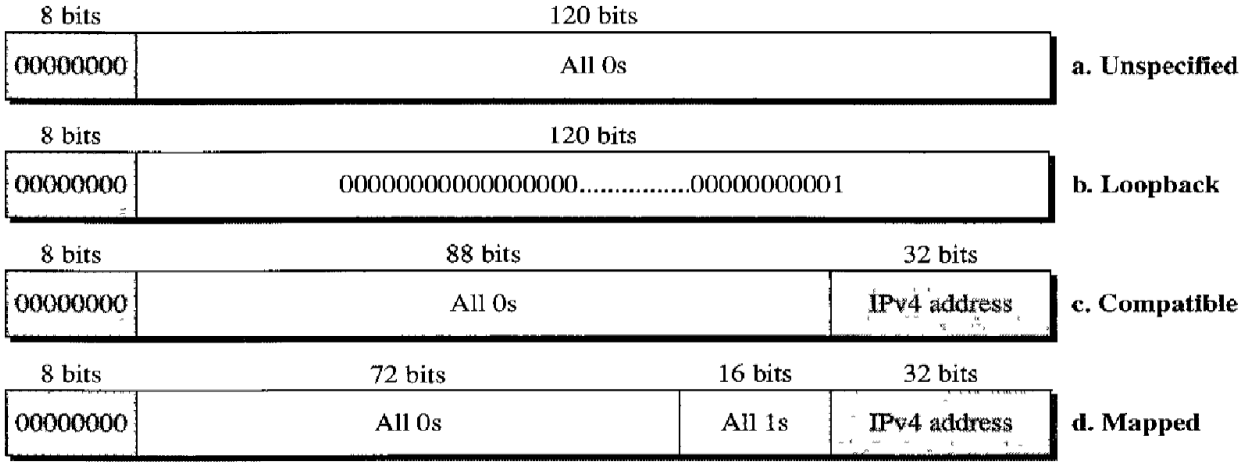
A unicast address defines a single computer. The packet sent to a unicast address must be delivered to that specific computer. IPv6 defines two types of unicast addresses: geo- graphically based and provider-based. We discuss the second type here; the first type is left for future definition. The provider-based address is generally used by a normal host as a unicast address. The address format is shown in Figure

### *Anycast Addresses*

IPv6 also defines anycast addresses. An **anycast address,** like a multicast address, also defines a group of nodes. However, a packet destined for an anycast address is delivered to only one of the members of the anycast group, the nearest one (the one with the shortest route). Although the definition of an anycast address is still debatable, one possible use is to assign an anycast address to all routers of an ISP that covers a large logical area in the Internet. The routers outside the ISP deliver a packet destined for the ISP to the nearest ISP router. No block is assigned for anycast addresses.

### *Reserved Addresses*

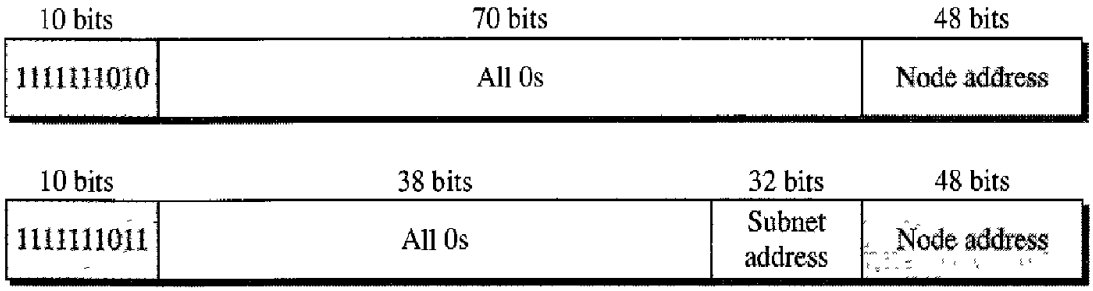
Another category in the address space is the **reserved address.** These addresses start with eight 0s (type prefix is 0000 0000). A few subcategories are defined in this category,




An **unspecified address** is used when a host does not know its own address and sends an inquiry to find its address. A **loopback address** is used by a host to test itself without going into the network. A **compatible address** is used during the transition from IPv4 to IPv6 (see Chapter 20). It is used when a computer using IPv6 wants to send a message to another computer using IPv6, but the message needs to pass through a part of the network that still operates in IPv4. A **mapped address** is also used during transition. However, it is used when a computer that has migrated to IPv6 wants to send a packet to a computer still using IPv4.

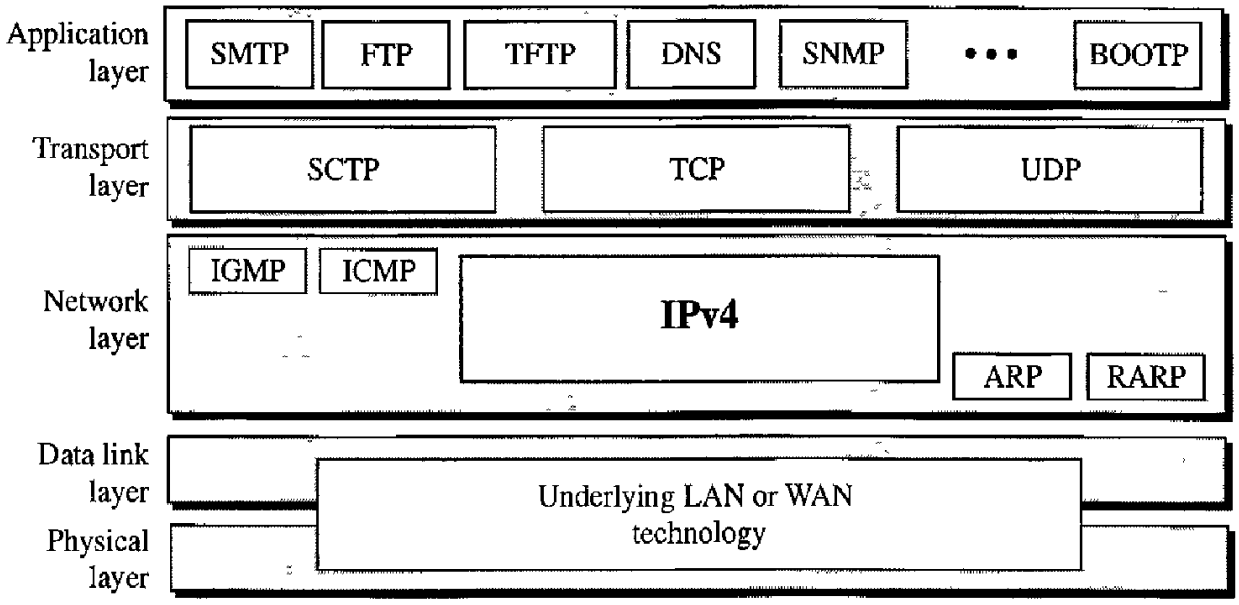
### *Local Addresses*

These addresses are used when an organization wants to use IPv6 protocol without being connected to the global Internet. In other words, they provide addressing for private networks. Nobody outside the organization can send a message to the nodes using these

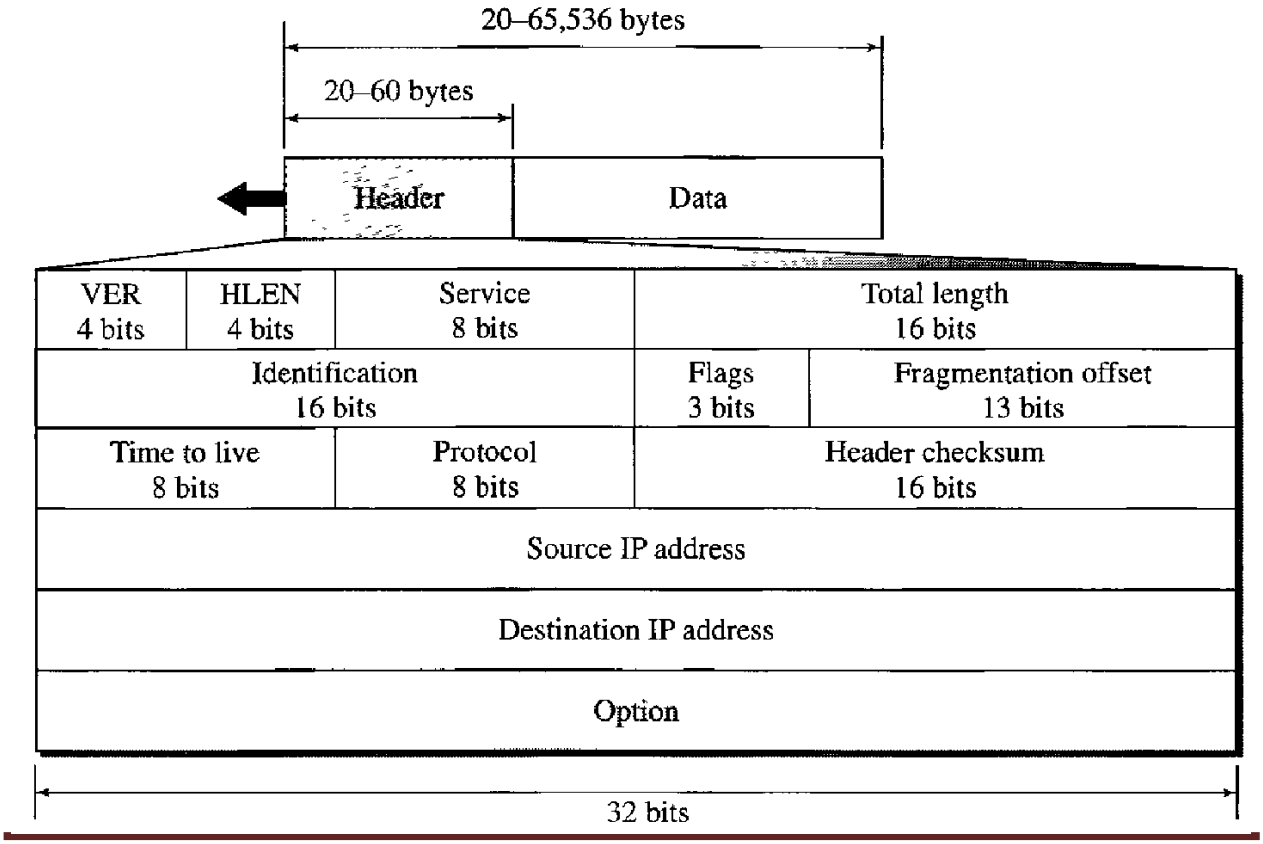



A **link local address** is used in an isolated subnet; a **site local address** is used in an isolated site with several subnets.

### *6.3 IPv4 protocol:*



## ***IPv4 Datagram format:***




A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. What is the first address in the block?
Solution:
n=28.
32-n=32-28=4.
11001101 00010000 00100101 00100111

11001101 00010000 00100101 0010000
205.16.37.32/28
or

The binary representation of the given address is 11001101 00010000 00100101 00100111
If we set 32−28 rightmost bits to 0, we get 11001101 00010000 00100101 0010000
or

205.16.37.32.

The binary representation of the given address is 11001101 00010000 00100101 00100111
If we set 32 − 28 rightmost bits to 1, we get 11001101 00010000 00100101 00101111
or
205.16.37.47

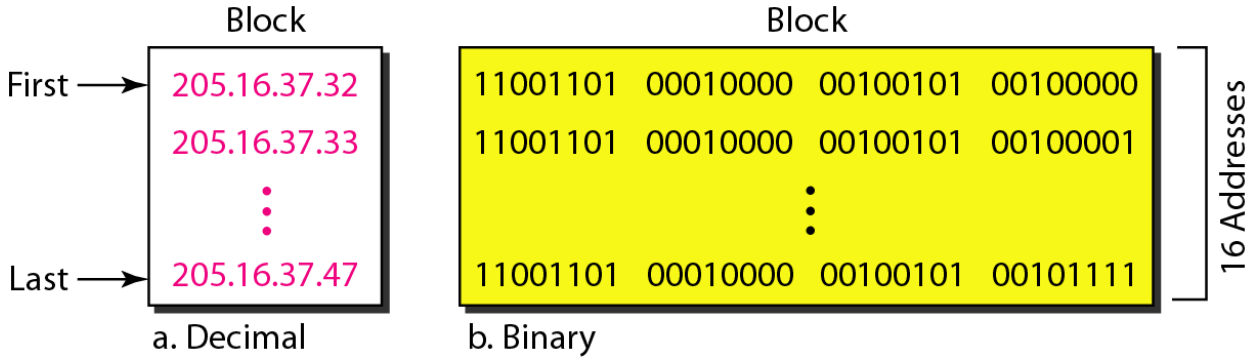
Another way to find the first address, the last address, and the number of addresses is to represent the mask as a 32-bit binary (or 8-digit hexadecimal) number. This is particularly useful when we are writing a program to find these pieces of information. In Example 19.5 the /28 can be represented as
11111111 11111111 11111111 11110000
(twenty-eight 1s and four 0s).
Find
a. The first address
b. The last address
c. The number of addresses.

The first address can be found by ANDing the given addresses with the mask. ANDing here is done bit by bit. The result of ANDing 2 bits is 1 if both bits are 1s; the result is 0 otherwise.

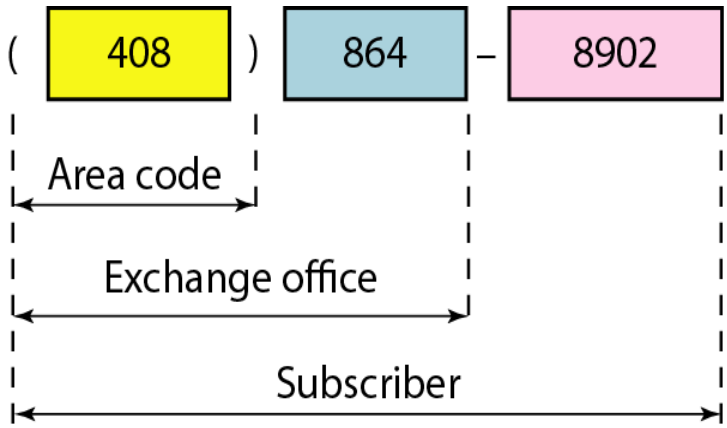
| | | | | |
|---|---|---|---|---|
| Address: | 11001101 | 00010000 | 00100101 | 00100111 |
| Mask: | **11111111** | **11111111** | **11111111** | **11110000** |
| First address: | 11001101 | 00010000 | 00100101 | 00100000 |

The last address can be found by ORing the given addresses with the complement of the mask. ORing here is done bit by bit. The result of ORing 2 bits is 0 if
both bits are 0s; the result is 1 otherwise. The complement of a number is found by changing each 1 to 0 and each 0 to 1.

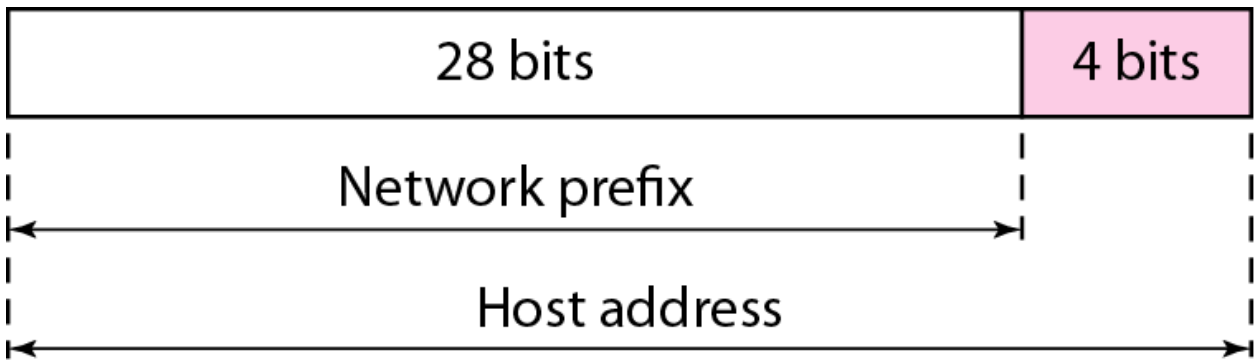
A network configuration for the block 205.16.37.32/28



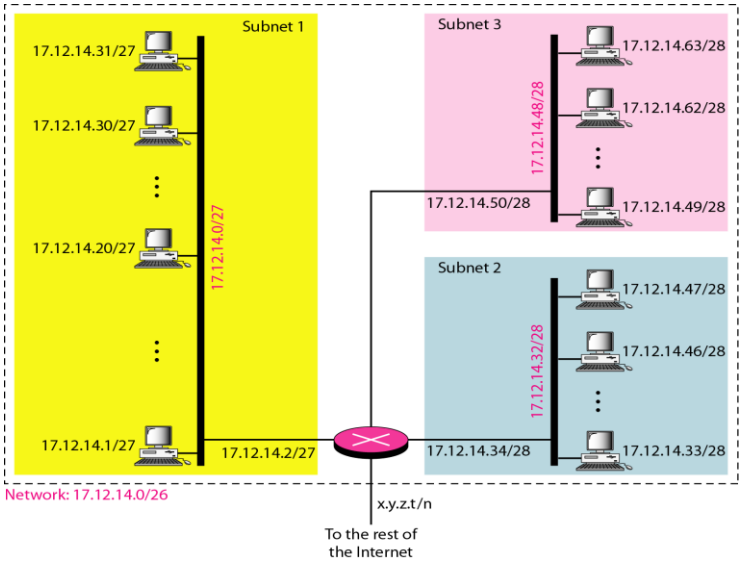

**The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world.**



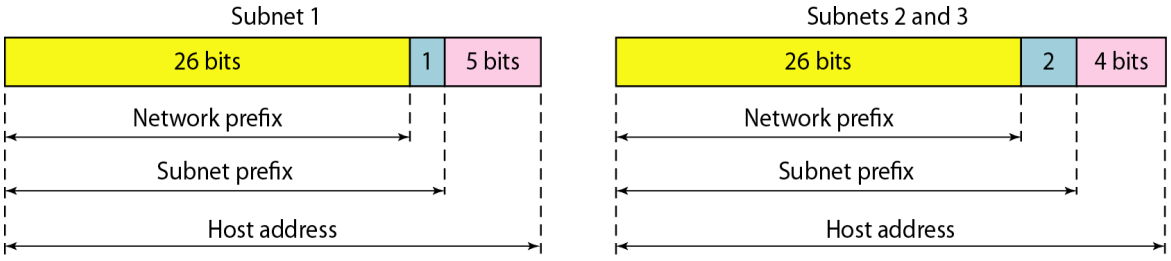

***A frame in a character-oriented protocol***



## *Configuration and addresses in a subnetted network*




## *Three-level hierarchy in an IPv4 address*




An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

a. The first group has 64 customers; each needs 256 addresses.

b. The second group has 128 customers; each needs 128 addresses.

c. The third group has 128 customers; each needs 64 addresses.

Design the subblocks and find out how many addresses are still available after these allocations.

Group 1

For this group, each customer needs 256 addresses. This means that 8 (log2 256) bits are needed to define each host. The prefix length is then 32 − 8 = 24. The addresses are

| | | |
|---|---|---|
| 1st Customer: | 190.100.0.0/24 | 190.100.0.255/24 |
| 2nd Customer: | 190.100.1.0/24 | 190.100.1.255/24 |
| . . . | | |
| 64th Customer: | 190.100.63.0/24 | 190.100.63.255/24 |
| Total = 64 × 256 = 16,384 | | |

***For this group, each customer needs 128 addresses. This means that 7 (log2 128) bits are needed to define each host. The prefix length is then 32 − 7 = 25. The addresses are***

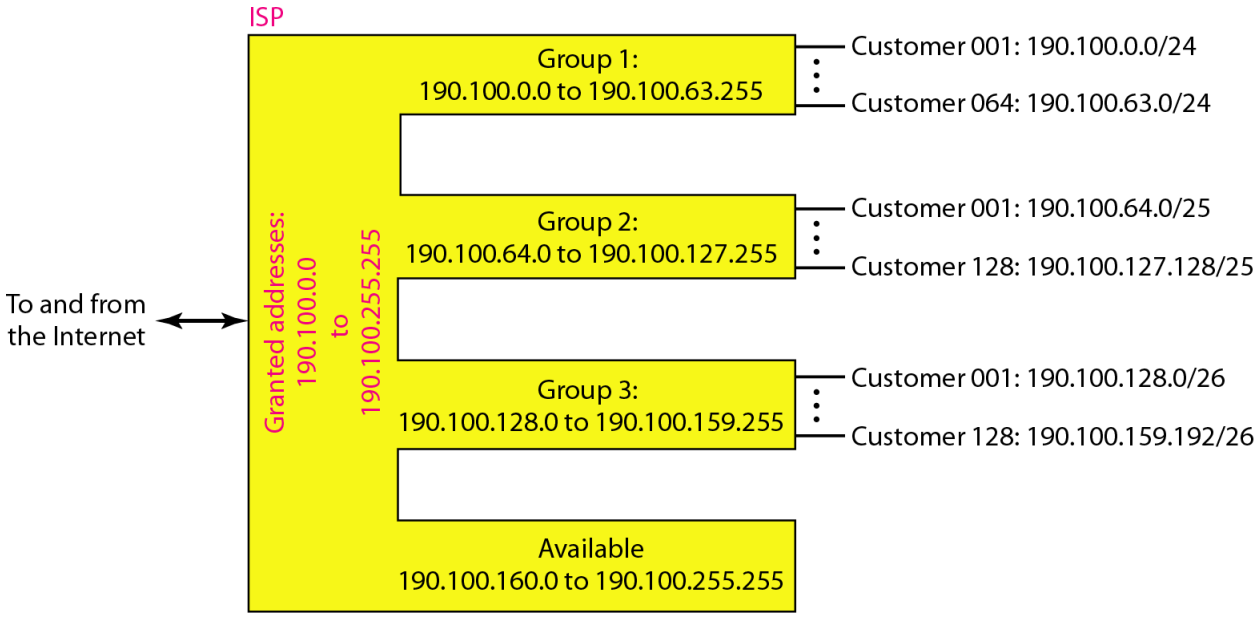
| | | |
|---|---|---|
| 1st Customer: | 190.100.64.0/25 | 190.100.64.127/25 |
| 2nd Customer: | 190.100.64.128/25 | 190.100.64.255/25 |
| . . . | | |
| 128th Customer: | 190.100.127.128/25 | 190.100.127.255/25 |
| Total = 128 × 128 = 16,384 | | |

***For this group, each customer needs 64 addresses. This means that 6 ($log_2 64$) bits are needed to each host. The prefix length is then 32 − 6 = 26. The addresses are***

| | | |
|---|---|---|
| 1st Customer: | 190.100.128.0/26 | 190.100.128.63/26 |
| 2nd Customer: | 190.100.128.64/26 | 190.100.128.127/26 |
| . . . | | |
| 128th Customer: | 190.100.159.192/26 | 190.100.159.255/26 |
| Total = 128 × 64 = 8192 | | |

***Number of granted addresses to the ISP: 65,536***
***Number of allocated addresses by the ISP: 40,960***
***Number of available addresses: 24,576***

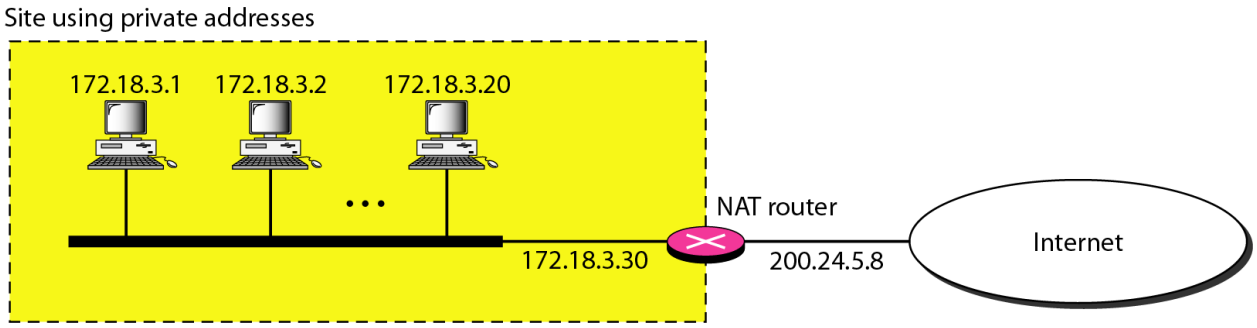
***An example of address allocation and distribution by an ISP***



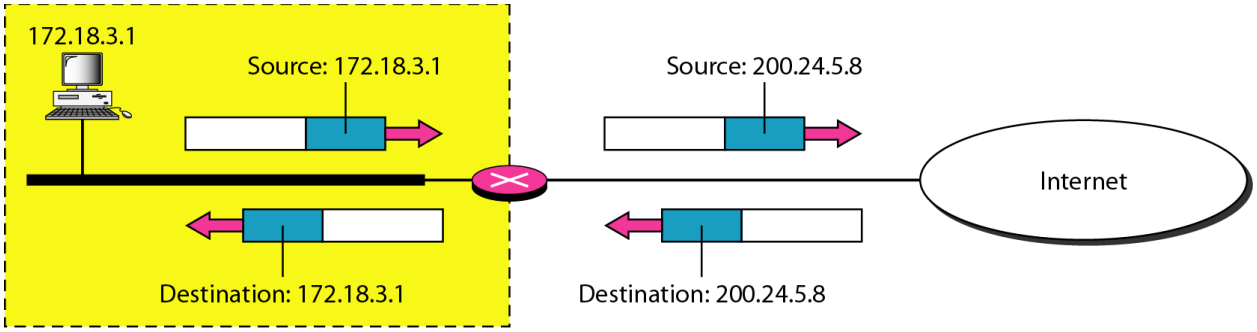

***Addresses for private networks***

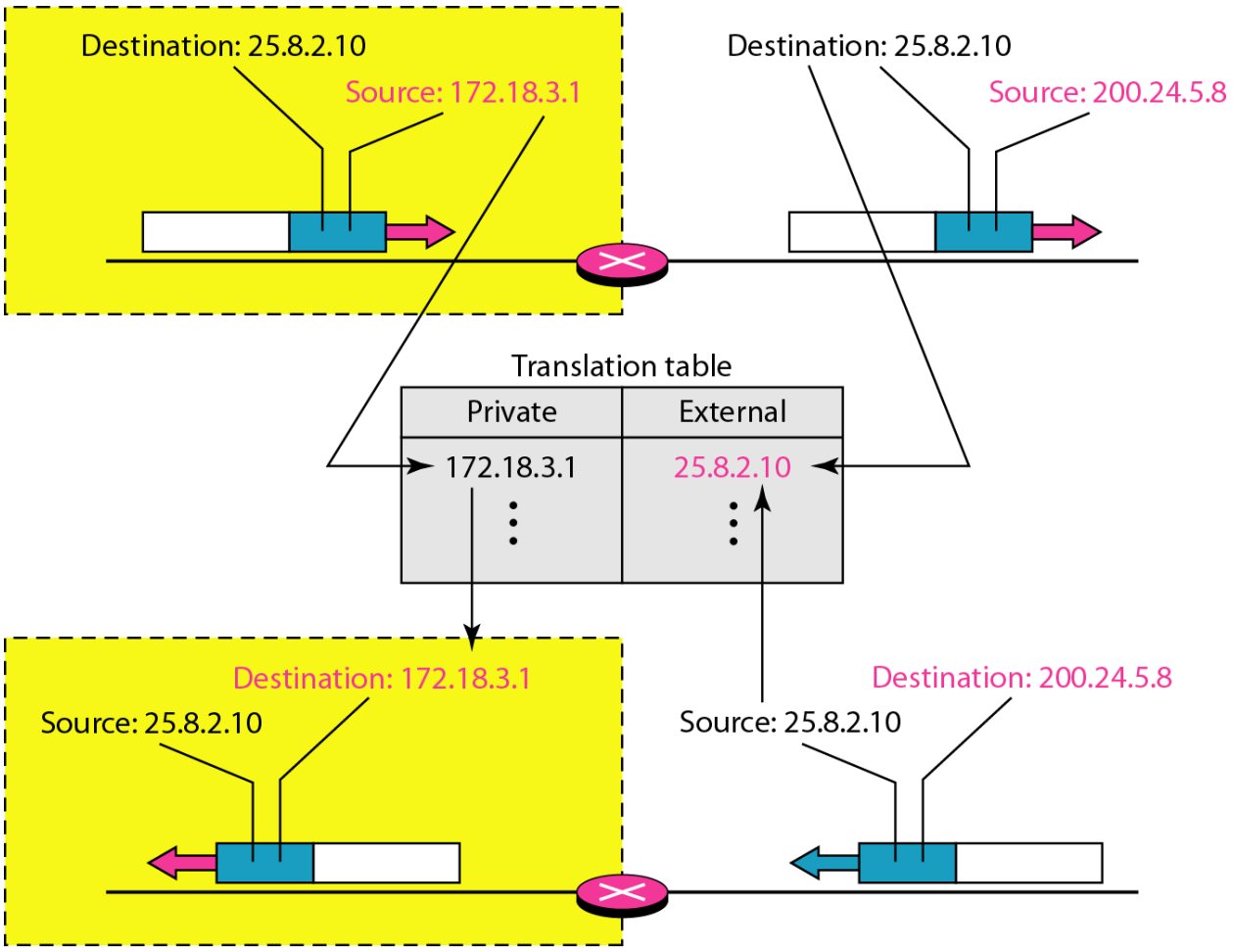
| *Range* | | | *Total* |
|---|---|---|---|
| 10.0.0.0 | to | 10.255.255.255 | $2^{24}$ |
| 172.16.0.0 | to | 172.31.255.255 | $2^{20}$ |
| 192.168.0.0 | to | 192.168.255.255 | $2^{16}$ |

## *A NAT implementation*



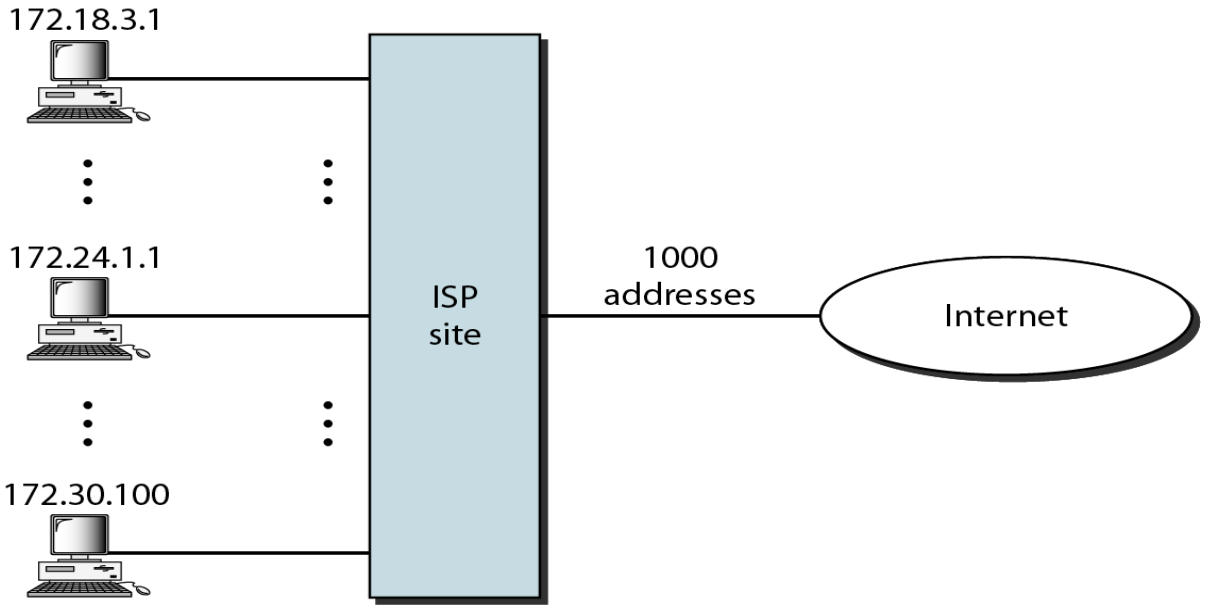

## *Addresses in a NAT*



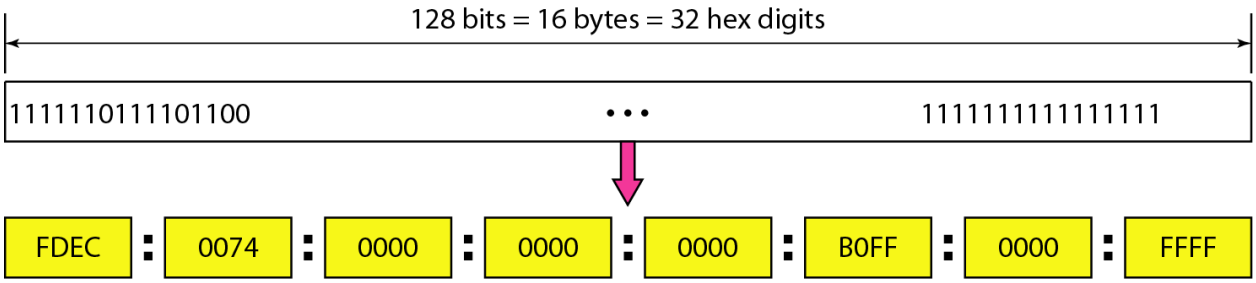


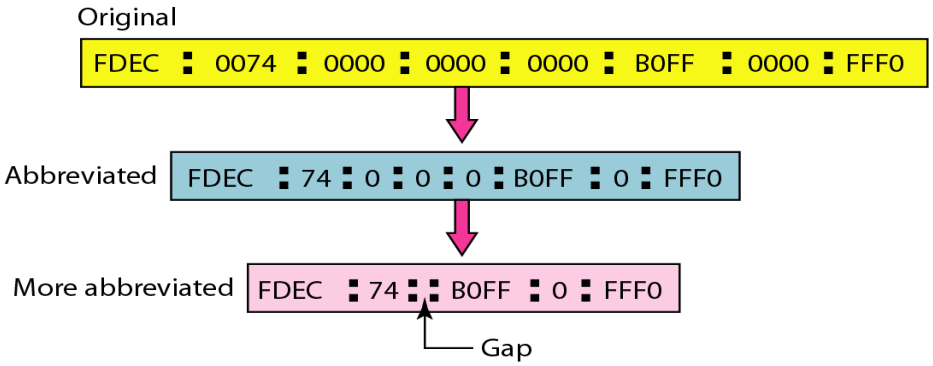

| *Private Address* | *Private Port* | *External Address* | *External Port* | *Transport Protocol* |
|---|---|---|---|---|
| 172.18.3.1 | 1400 | 25.8.3.2 | 80 | TCP |
| 172.18.3.2 | 1401 | 25.8.3.2 | 80 | TCP |
| . . . | . . . | . . . | . . . | . . . |

***An ISP and NAT***




**An IPv6 address is 128 bits long.**



Expand the address 0:15::1:12:1213 to its original.

We first need to align the left side of the double colon to the left of the original pattern and the right side of the double colon to the right of the original pattern to find how many 0s we need to replace the double colon

```
XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX
  0:   15:                  :    1:   12:1213
```

***This means that the original address is.***

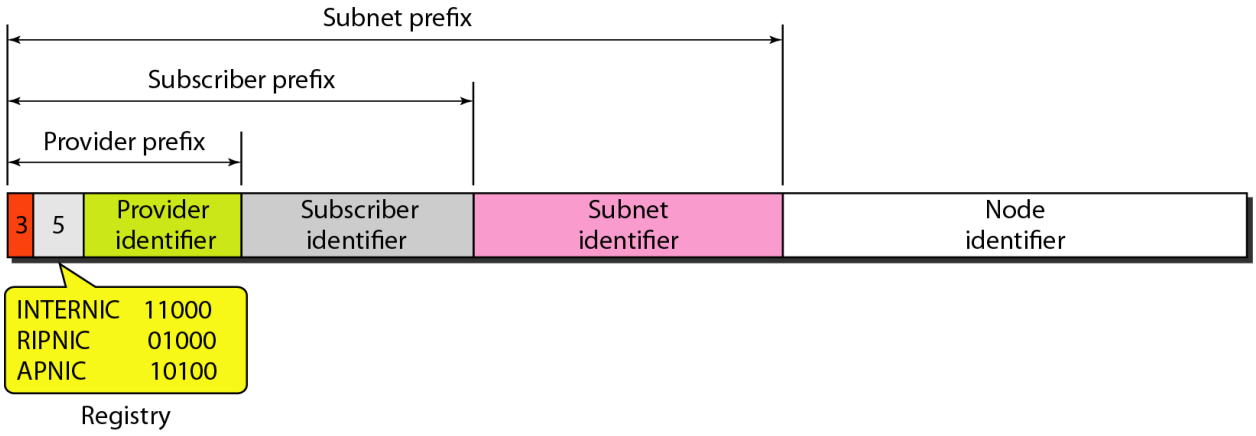
```
0000:0015:0000:0000:0000:0001:0012:1213
```

***Type prefixes for IPv6 addresses***

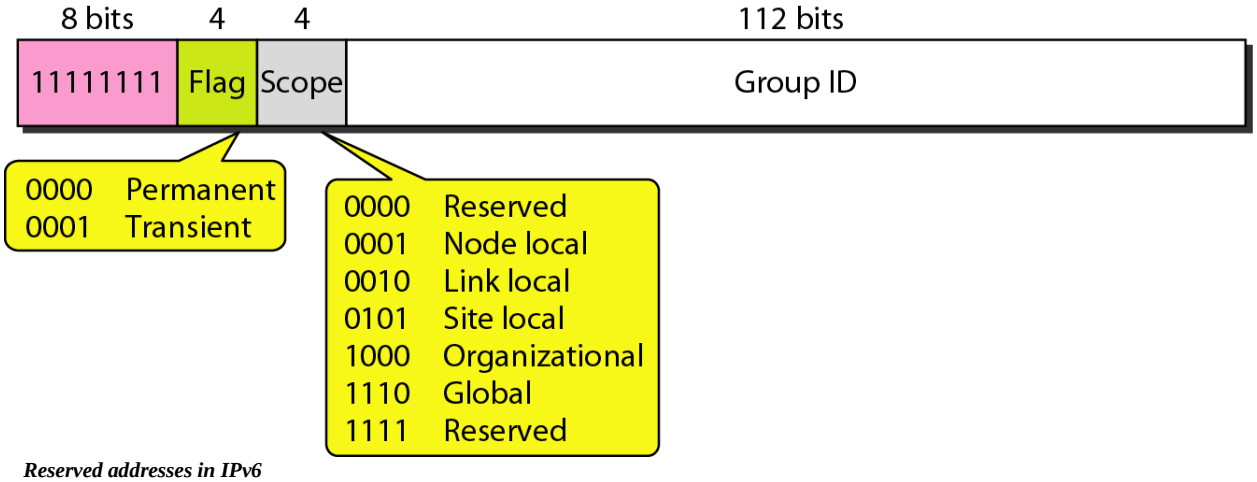
| *Type Prefix* | *Type* | *Fraction* |
|---|---|---|
| 0000 0000 | Reserved | 1/256 |
| 0000 0001 | Unassigned | 1/256 |
| 0000 001 | ISO network addresses | 1/128 |
| 0000 010 | IPX (Novell) network addresses | 1/128 |
| 0000 011 | Unassigned | 1/128 |
| 0000 1 | Unassigned | 1/32 |
| 0001 | Reserved | 1/16 |
| 001 | Reserved | 1/8 |
| **010** | **Provider-based unicast addresses** | **1/8** |

| *Type Prefix* | *Type* | *Fraction* |
|---|---|---|
| 011 | Unassigned | 1/8 |
| 100 | Geographic-based unicast addresses | 1/8 |
| 101 | Unassigned | 1/8 |
| 110 | Unassigned | 1/8 |
| 1110 | Unassigned | 1/16 |
| 1111 0 | Unassigned | 1/32 |
| 1111 10 | Unassigned | 1/64 |
| 1111 110 | Unassigned | 1/128 |
| 1111 1110 0 | Unassigned | 1/512 |
| 1111 1110 10 | Link local addresses | 1/1024 |
| 1111 1110 11 | Site local addresses | 1/1024 |
| 1111 1111 | Multicast addresses | 1/256 |

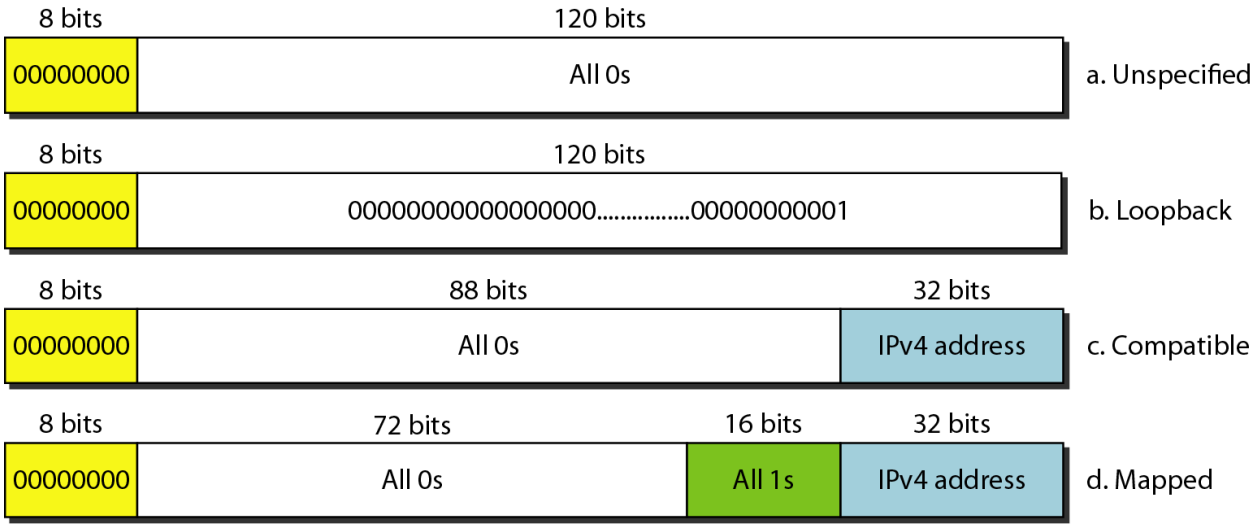
Prefixes for provider-based unicast address



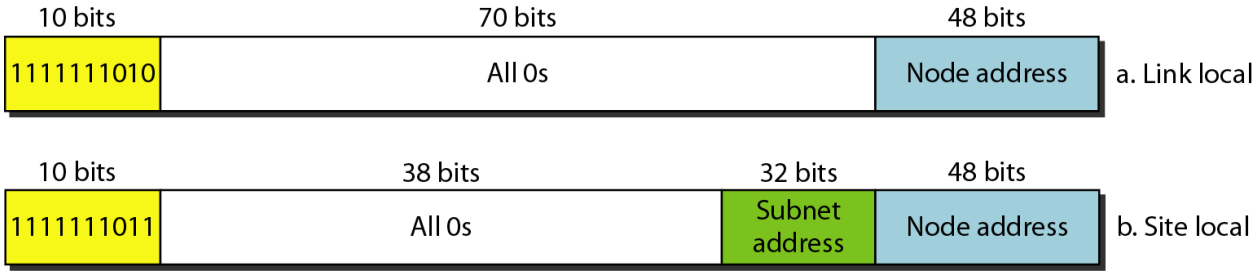

Multicast address in IPv6




***Reserved addresses in IPv6***




***Local addresses in IPv6***

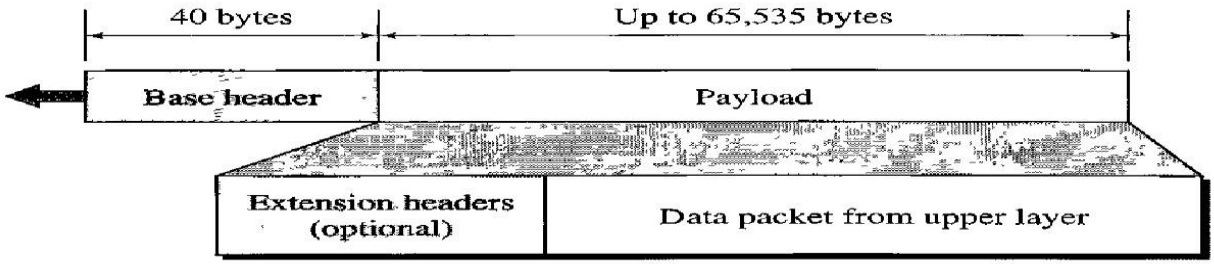


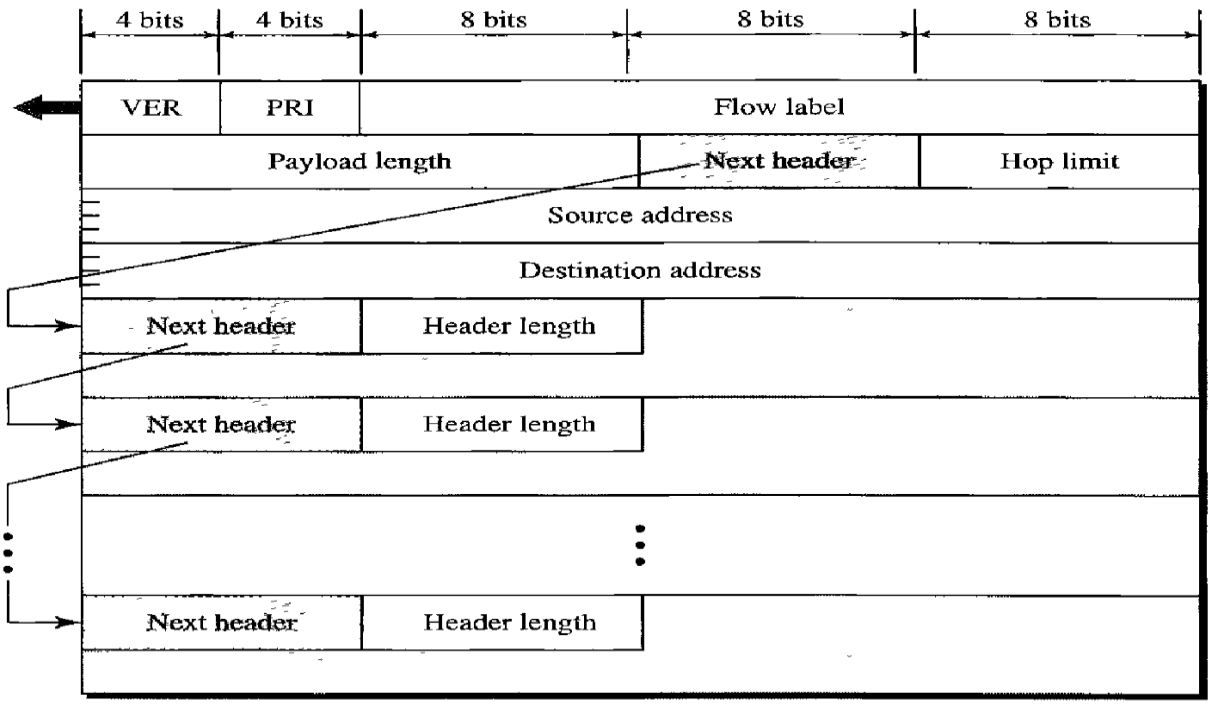
## Advantages

The next-generation IP, or IPv6, has some advantages over IPv4 that can be summarized as follows:

- **Larger address space.** An IPv6 address is 128 bits long, as we discussed in Chapter 19. Compared with the 32-bit address of IPv4, this is a huge ($2^{96}$) increase in the address space.
- **Better header format.** IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the upper-layer data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
- **New options.** IPv6 has new options to allow for additional functionalities.
- **Allowance for extension.** IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- **Support for resource allocation.** In IPv6, the type-of-service field has been removed, but a mechanism (called *flow label*) has been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.
- **Support for more security.** The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

### *IPv6 packet format:*






- **Flow label.** The **flow label** is a 3-byte (24-bit) field that is designed to provide special handling for a particular flow of data. We will discuss this field later.
- **Payload length.** The 2-byte payload length field defines the length of the IP datagram excluding the base header.
- **Next header.** The **next header** is an 8-bit field defining the header that follows the base header in the datagram. The next header is either one of the optional extension headers used by IP or the header of an encapsulated packet such as UDP or TCP. Each extension header also contains this field. Table 20.6 shows the values of next headers. Note that this field in version 4 is called the *protocol*.
- **Hop limit.** This 8-bit **hop limit** field serves the same purpose as the TTL field in IPv4.
- **Source address.** The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram.
- **Destination address.** The destination address field is a 16-byte (128-bit) Internet address that usually identifies the final destination of the datagram. However, if source routing is used, this field contains the address of the next router.

### *Priority*

The priority field of the IPv6 packet defines the priority of each packet with respect to other packets from the same source. For example, if one of two consecutive datagrams must be discarded due to congestion, the datagram with the lower **packet priority** will be discarded. IPv6 divides traffic into two broad categories: congestion-controlled and noncongestion-controlled.

**Congestion-Controlled Traffic** If a source adapts itself to traffic slowdown when there is congestion, the traffic is referred to as **congestion-controlled traffic.** For example, TCP, which uses the sliding window protocol, can easily respond to traffic. In congestion-controlled traffic, it is understood that packets may arrive delayed, lost, or out of order. Congestion-controlled data are assigned priorities from 0 to 7, as listed in

| *Priority* | *Meaning* |
|---|---|
| 0 | No specific traffic |
| 1 | Background data |
| 2 | Unattended data traffic |
| 3 | Reserved |
| 4 | Attended bulk data traffic |
| 5 | Reserved |
| 6 | Interactive traffic |
| 7 | Control traffic |

The priority descriptions are as follows:

- **No specific traffic.** A priority of 0 is assigned to a packet when the process does not define a priority.
- **Background data.** This group (priority 1) defines data that are usually delivered in the background. Delivery of the news is a good example.
- **Unattended data traffic.** If the user is not waiting (attending) for the data to be received, the packet will be given a priority of 2. E-mail belongs to this group. The recipient of an e-mail does not know when a message has arrived. In addition, an e-mail is usually stored before it is forwarded. A little bit of delay is of little consequence.
- **Attended bulk data traffic.** A protocol that transfers data while the user is waiting (attending) to receive the data (possibly with delay) is given a priority of 4. FTP and HTTP belong to this group.
- **Interactive traffic.** Protocols such as TELNET that need user interaction are assigned the second-highest priority (6) in this group.
- **Control traffic.** Control traffic is given the highest priority (7). Routing protocols such as OSPF and RIP and management protocols such as SNMP have this priority.

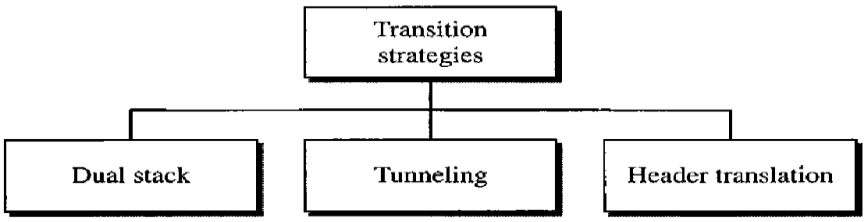
**Noncongestion-Controlled Traffic** This refers to a type of traffic that expects minimum delay. Discarding of packets is not desirable. Retransmission in most cases is impossible. In other words, the source does not adapt itself to congestion. Real-time audio and video are examples of this type of traffic.

## Comparison between IPv4 and IPv6:

| *Comparison* |
|---|
| 1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version. |
| 2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field. |
| 3. The total length field is eliminated in IPv6 and replaced by the payload length field. |
| 4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header. |
| 5. The TTL field is called hop limit in IPv6. |
| 6. The protocol field is replaced by the next header field. |
| 7. The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level. |
| 8. The option fields in IPv4 are implemented as extension headers in IPv6. |

## Transition from IPv4 to IPv6:

Because of the huge number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly. It takes a considerable amount of time before every system in the Internet can move from IPv4 to IPv6. The transition must be smooth to prevent any problems between IPv4 and IPv6 systems. Three strategies have been




## Dual Stack

It is recommended that all hosts, before migrating completely to version 6, have a **dual stack** of protocols. In other words, a station must run IPv4 and IPv6 simultaneously until all the Internet uses IPv6. See Figure 20.19 for the layout of a dual-stack configuration.

## Tunneling

**Tunneling** is a strategy used when two computers using IPv6 want to communicate with each other and the packet must pass through a region that uses IPv4. To pass through this region, the packet must have an IPv4 address. So the IPv6 packet is encapsulated in an IPv4 packet when it enters the region, and it leaves its capsule when it exits the region. It seems as if the IPv6 packet goes through a tunnel at one end and emerges at the other end. To make it clear that the IPv4 packet is carrying an IPv6

## Header Translation

**Header translation** is necessary when the majority of the Internet has moved to IPv6 but some systems still use IPv4. The sender wants to use IPv6, but the receiver does not understand IPv6. Tunneling does not work in this situation because the packet must be in the IPv4 format to be understood by the receiver. In this case, the header format must be totally changed through header translation. The header of the IPv6 packet is con-

| *Header Translation Procedure* |
|---|
| 1. The IPv6 mapped address is changed to an IPv4 address by extracting the rightmost 32 bits. |
| 2. The value of the IPv6 priority field is discarded. |
| 3. The type of service field in IPv4 is set to zero. |
| 4. The checksum for IPv4 is calculated and inserted in the corresponding field. |
| 5. The IPv6 flow label is ignored. |
| 6. Compatible extension headers are converted to options and inserted in the IPv4 header. Some may have to be dropped. |
| 7. The length of IPv4 header is calculated and inserted into the corresponding field. |
| 8. The total length of the IPv4 packet is calculated and inserted in the corresponding field. |

# UNIT - VI

**1.** a) An ISP is granted a block of addresses starting with 150.80.0.0/16. The isp wants to distribute these blocks of 2600 customers as follows:

i) The first group has 200 medium size business: each need 16 addresses.

ii) The second group has 400 small business: each need 8 addresses

iii) The third group has 2000 households: each need 4 addresses

Design the subblocks and give the slash notation for each subblock. Find out how many addresses are still available after these allocations **(July 2013 10 marks)**

b) Explain briefly strategies used to handle the transistion from IPv4 to IPv6 **(July 2013 06 marks)**

c) A block of addresses is granted to a small organization. One of the addresses is 205.16.37.39/28. What is the first address, last address and number of address in the block **(July 2013 04 marks)**

**2. a)** Explain the address format for IPv4 and IPv6 address **(Jan 2013 10 marks)**

b) List the classes in classful addressing and define the application of each class **(Jan 2013 10marks)**

**3.** a) What are the differences between classful and classless addressing **(July 2012 10marks)**

**b)** Draw the IPV4 datagram format and explain its field **(July 2012 10 marks)**

**4.** a)What is subnetting? Why it is required ? What is the maximum number of subnets in class C network with following subnet mask?

i) 255.255.255.0

ii) 255.255.255.224

iii) 255.255.255.248 **(July 2011 04 marks)**

**b)** Explain IPV4 header format **(July 2011 08 marks)**

c) Find the range of address in the following blocks

i) 123.56.77.32/29

ii) 200.17.21.128/27

iii) 17.34.16.0/23

iv) 180.34.64.64/30 **(Dec 2011 8 marks)**

**5.** a)Explain the IPV4 datagram format. **(Dec 2011 10 marks)**

**b)** Explain IPV6 addresses **(Dec 2010 10 marks)**

**6.** a)Find the class of following IP addresses:

i) 237.14.2.1

ii) 208.35.54.12

iii) 129.14.6.8

iv) 114.34.2.8 **(Dec 2010 04 marks)**

**b)** What is the need of transition from IPV4 to IPV6? What are the strategies deviced by IETF to help the transition? **(June 2010 12 marks)**

c) Find the error, if any, in the following IPV4 addresses:

**i) 75.45.301.14 ii) 221.34.7.8.20 (June 2010 02 marks)**

**d)** What are the classeless addressing in IPV4? What is a mask? Explain. **(June 2010 04 marks)**

**7.** a) What is NAT? Explain how address translation done in NAT**(Dec 2014 10 marks)**

b)Why IPv4 to IPv6 transition is required ?What are the various techniques used in transition ? **(Dec 20114 10marks)**

**8. a)**What is NAT? Explain how NAT help in address depletion**(Jun 2014 10marks)**

b)Explain IPv4 Datagram (**Jun 2014 10marks)**

**9. a)** Explain class full addressing for IP address (**Jun 2014 10marks)**

**b)** An ISP granted a block of addressing with 190.100.0.0/16 .The ISP needs to distribute these address to three groups of customers as following i)First group has 64 customers each with 256 addresses ii)second group has 128 customers each with 128 addresses (**Jun 2014 10marks)**

**10.** a)Distinguish between class A, class B and class C addressing **(July 2011 10 marks)**

**b)** What is NAT and explain in brief? How can NAT help in address depletion

**(July 2012 /Jan 2013/Dec 2010 10marks)**

# 7.1 Delivery:

The network layer supervises the handling of the packets by the underlying physical networks.We define this handling as the delivery of a packet.

## Direct versus Indirect delivery

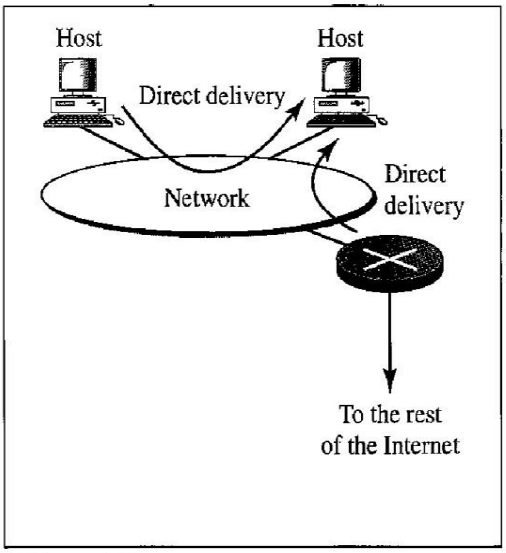
### *Direct Delivery*

In a **direct delivery,** the final destination of the packet is a host connected to the same physical network as the deliverer. Direct delivery occurs when the source and destination of the packet are located on the same physical network or when the delivery is between the last router and the destination host.

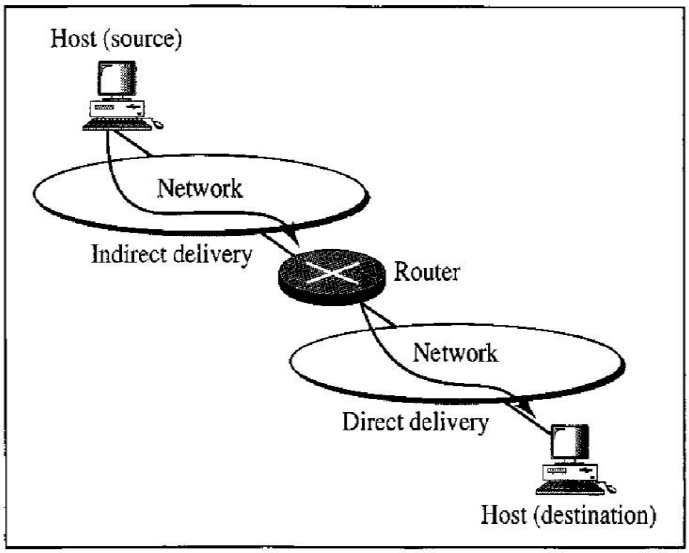
The sender can easily determine if the delivery is direct. It can extract the network address of the destination (using the mask) and compare this address with the addresses of the networks to which it is connected. If a match is found, the delivery is direct.

### *Indirect Delivery*

If the destination host is not on the same network as the deliverer, the packet is delivered indirectly. In an **indirect delivery,** the packet goes from router to router until it reaches the one connected to the same physical network as its final destination. Note that a delivery always involves one direct delivery but zero or more indirect deliveries. Note also that the last delivery is always a direct delivery.




**a. Direct delivery**




**b. Indirect and direct delivery**

# 7.2 Forwarding:

Forwarding means to place the packet in its route to its destination. Forwarding requires a host or a router to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination.

## Forwarding Techniques

Several techniques can make the size of the routing table manageable and also handle issues such as security. We briefly discuss these methods here.

### *Next-Hop Method Versus Route Method*

One technique to reduce the contents of a routing table is called the **next-hop method.** In this technique, the routing table holds only the address of the next hop instead of information about the complete route **(route method).** The entries of a routing table must be consistent with one another. Figure 22.2 shows how routing tables can be simplified by using this technique.

### *Network-Specific Method Versus Host-Specific Method*

A second technique to reduce the routing table and simplify the searching process is called the **network-specific method.** Here, instead of having an entry for every destination host connected to the same physical network **(host-specific method),** we have only one entry that defines the address of the destination network itself. In other words, we treat all hosts connected to the same network as one single entity. For example, if 1000 hosts are attached to the same network, only one entry exists in the routing table instead of 1000. Figure 22.3 shows the concept.

**a. Routing tables based on route**

| Destination | Route |
|---|---|
| Host B | R1, R2, host B |

Routing table for host A

| Destination | Route |
|---|---|
| Host B | R2, host B |

Routing table for R1

| Destination | Route |
|---|---|
| Host B | Host B |

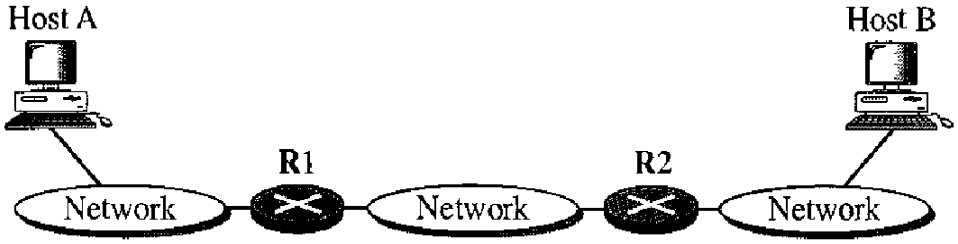
Routing table for R2

**b. Routing tables based on next hop**

| Destination | Next hop |
|---|---|
| Host B | R1 |

| Destination | Next hop |
|---|---|
| Host B | R2 |

| Destination | Next hop |
|---|---|
| Host B | --- |



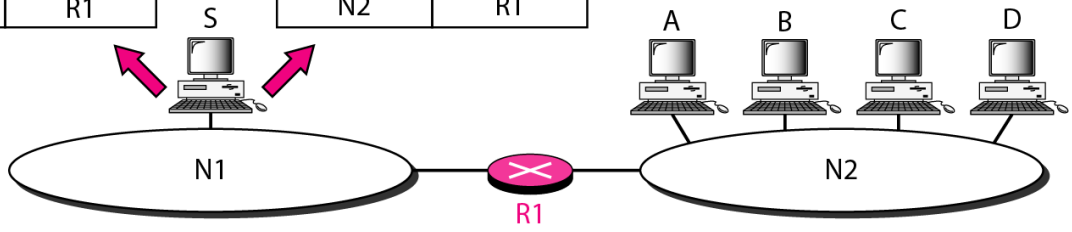

***Host-specific versus network-specific method***

Routing table for host S based on host-specific method

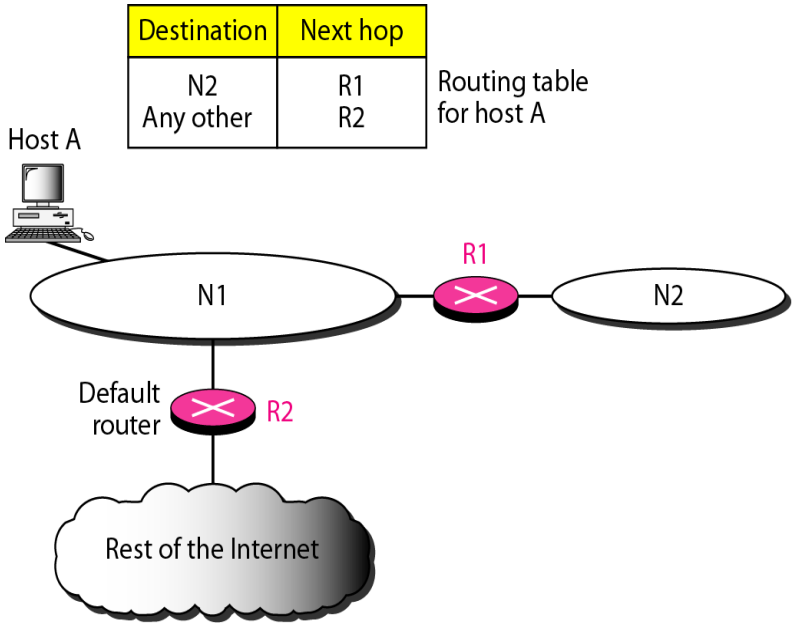
| Destination | Next hop |
|---|---|
| A | R1 |
| B | R1 |
| C | R1 |
| D | R1 |

Routing table for host S based on network-specific method

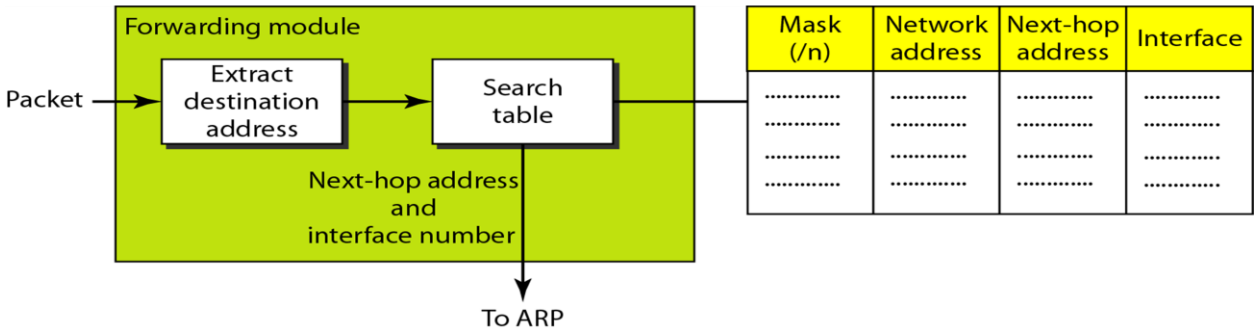
| Destination | Next hop |
|---|---|
| N2 | R1 |



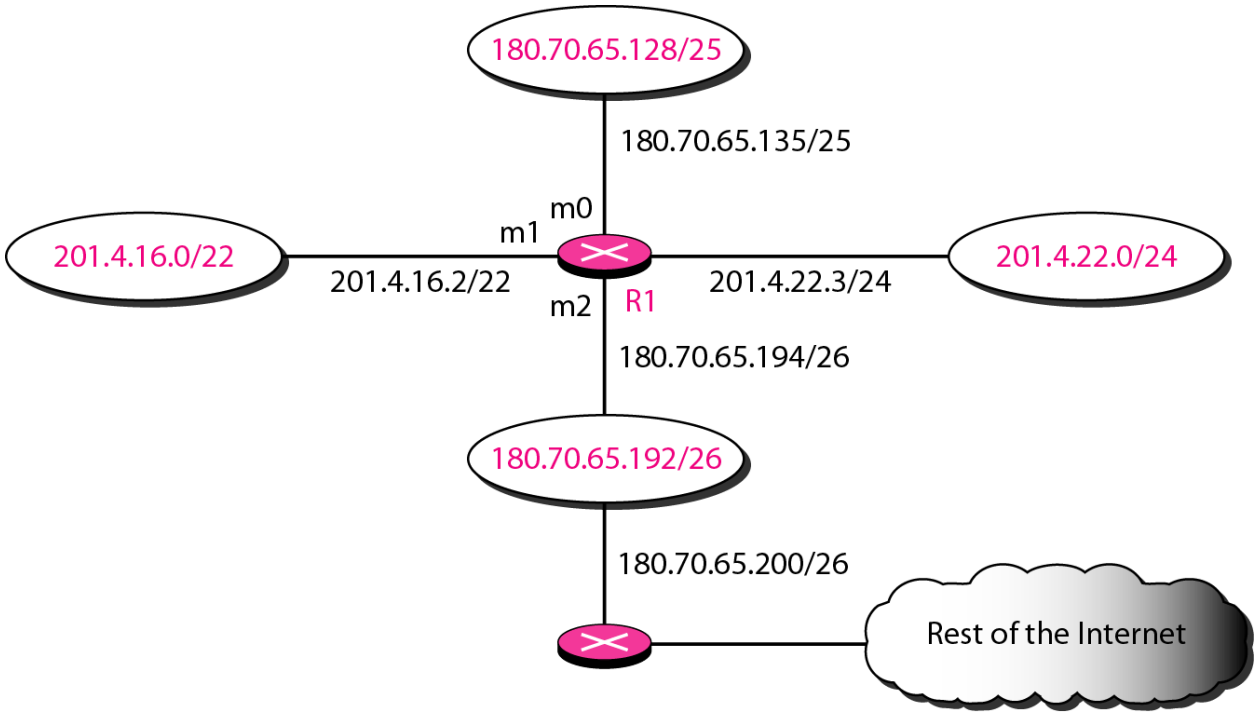
***Default method***




***Simplified forwarding module in classless address***



***Make a routing table for router R1, using the configuration in Figure***




***Routing table for router R1 in***

| *Mask* | *Network Address* | *Next Hop* | *Interface* |
|---|---|---|---|
| /26 | 180.70.65.192 | — | m2 |
| /25 | 180.70.65.128 | — | m0 |
| /24 | 201.4.22.0 | — | m3 |
| /22 | 201.4.16.0 | .... | m1 |
| Any | Any | 180.70.65.200 | m2 |

Show the forwarding process if a packet arrives at R1 in Figure 22.6 with the destination address 180.70.65.140

Solution

The router performs the following steps:

1. The first mask (/26) is applied to the destination address.
   The result is 180.70.65.128, which does not match the
   corresponding network address.
2. The second mask (/25) is applied to the destination address. The result is 180.70.65.128, which matches the corresponding network address. The next-hop address and the interface number m0 are passed to ARP for further processing.

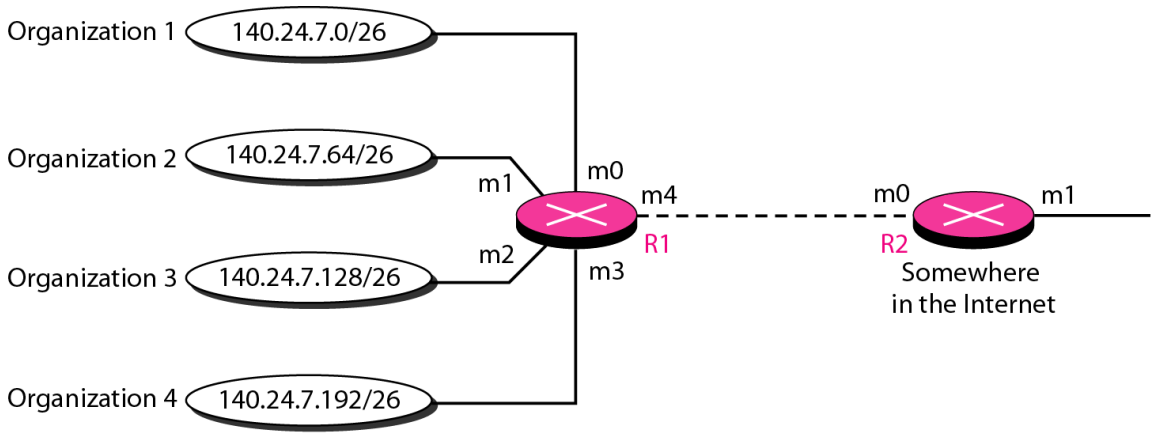
***Show the forwarding process if a packet arrives at R1 in Figure with the destination address 201.4.22.35.***

The router performs the following steps:

1. The first mask (/26) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address.
2. The second mask (/25) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address (row 2).

The third mask (/24) is applied to the destination address. The result is 201.4.22.0, which matches the corresponding network address. The destination address of the packet and the interface number m3 are passed to ARP.

Show the forwarding process if a packet arrives at R1 in Figure with the destination address 18.24.32.78 This time all masks are applied, one by one, to the destination address, but no matching network address is found. When it reaches the end of the table, the module gives the next-hop address 180.70.65.200 and interface number m2 to ARP. This is probably an outgoing package that needs to be sent, via the default router, to someplace else in the Internet.



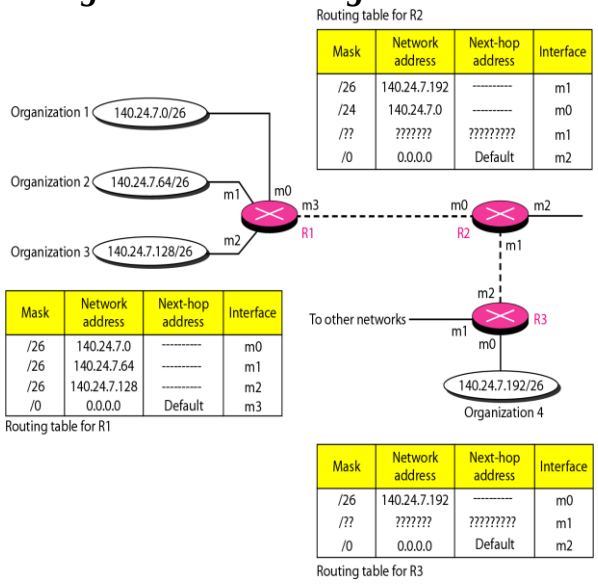

| Mask | Network address | Next-hop address | Interface |
|---|---|---|---|
| /26 | 140.24.7.0 | ---------- | m0 |
| /26 | 140.24.7.64 | ---------- | m1 |
| /26 | 140.24.7.128 | ---------- | m2 |
| /26 | 140.24.7.192 | ---------- | m3 |
| /0 | 0.0.0.0 | Default | m4 |

Routing table for R1

| Mask | Network address | Next-hop address | Interface |
|---|---|---|---|
| /24 | 140.24.7.0 | ---------- | m0 |
| /0 | 0.0.0.0 | Default | m1 |

Routing table for R2

## *Longest mask matching*

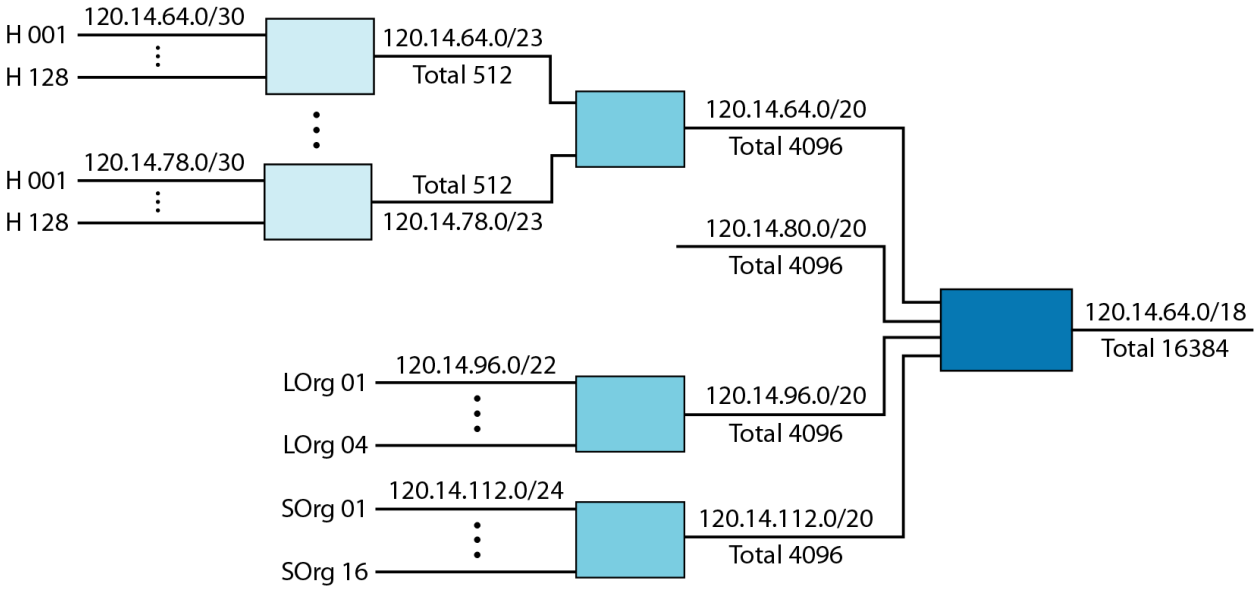



As an example of hierarchical routing, let us consider Figure A regional ISP is granted 16,384 addresses starting from 120.14.64.0. The regional ISP has decided to divide this block into four sub blocks, each with 4096 addresses. Three of these sub blocks are assigned to three local ISPs; the second sub block is reserved for future use. Note that the mask for each block is /20 because the original block with mask /18 is divided into 4 blocks.

The first local ISP has divided its assigned sub block into 8 smaller blocks and assigned each to a small ISP. Each small ISP provides services to 128 households, each using four addresses.

The second local ISP has divided its block into 4 blocks and has assigned the addresses to four large organizations.
The third local ISP has divided its block into 16 blocks and assigned each block to a small organization. Each small organization has 256 addresses, and the mask is /24.
There is a sense of hierarchy in this configuration. All routers in the Internet send a packet with destination address 120.14.64.0 to 120.14.127.255 to the regional ISP.

### *Hierarchical routing with ISPs*




### *Common fields in a routing table*

| Mask | Network address | Next-hop address | Interface | Flags | Reference count | Use |
|---|---|---|---|---|---|---|
| ..... | ..... | ..... | ..... | ..... | ..... | ..... |

## 7.3 Unicast Routing Protocols

A routing table can be either static or dynamic. A *static table* is one with manual entries. A *dynamic table,* on the other hand, is one that is updated automatically when there is a change somewhere in the internet. Today, an internet needs dynamic routing tables. The tables need to be updated as soon as there is a change in the internet. For instance, they need to be updated when a router is down, and they need to be updated whenever a better route has been found.

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 neighborhood. The sharing of information allows a router in San Francisco to know about the failure of a network in Texas. The routing protocols also include procedures for combining information received from other routers.

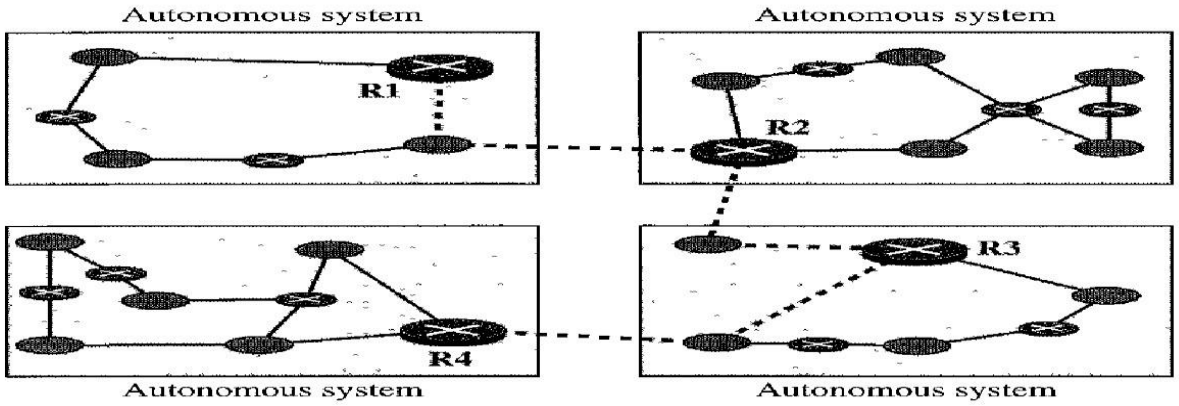
## Optimization

A router receives a packet from a network and passes it to another network. A router is usually attached to several networks. When it receives a packet, to which network should it pass the packet? The decision is based on optimization: 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.

## Intra- and Interdomain Routing

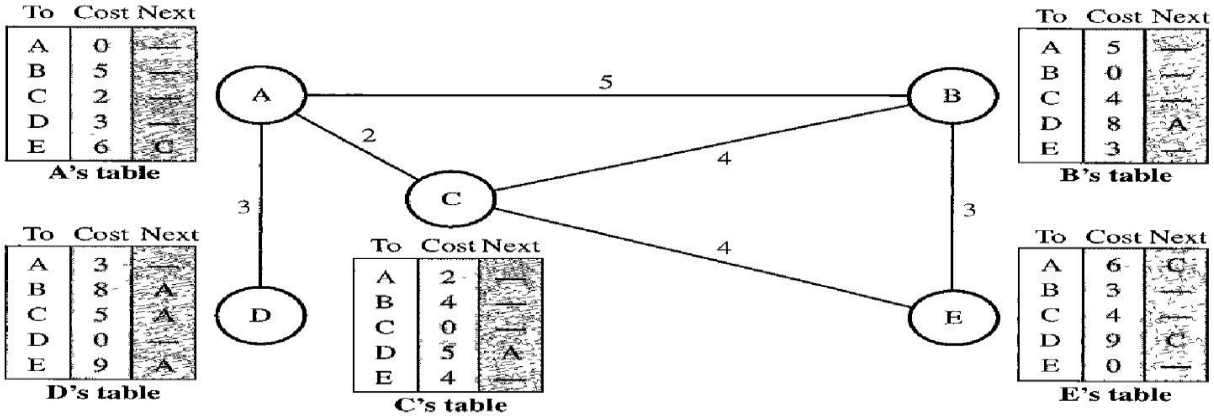
Today, an internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers. For this reason, an internet is divided into autonomous systems. An **autonomous system (AS)** is a group of networks and routers under the authority of a single administration. 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 (see



# 1. Distance Vector Routing

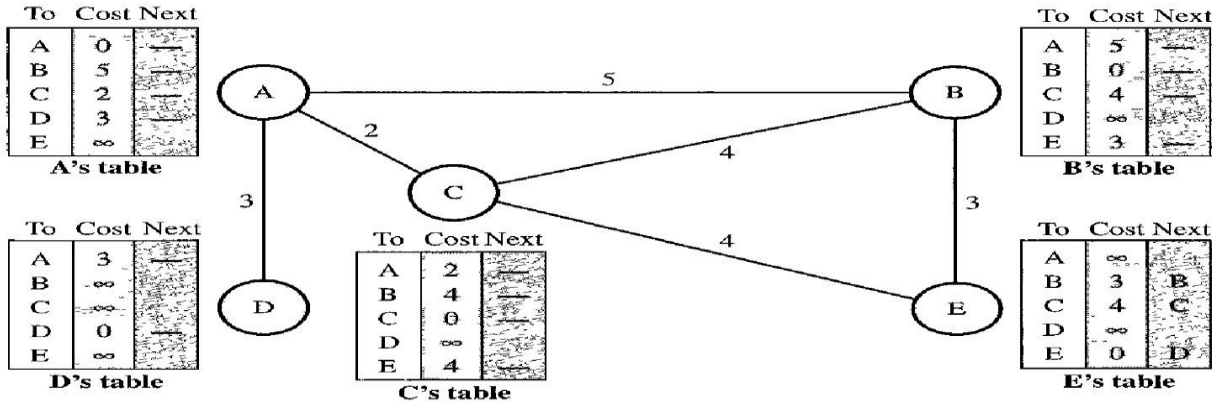
In **distance vector routing,** the least-cost route between any two nodes is the route with minimum distance. In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node. The table at each node also guides the packets to the desired node by showing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

We can think of nodes as the cities in an area and the lines as the roads connecting them. A table can show a tourist the minimum distance between cities.




*Sharing*

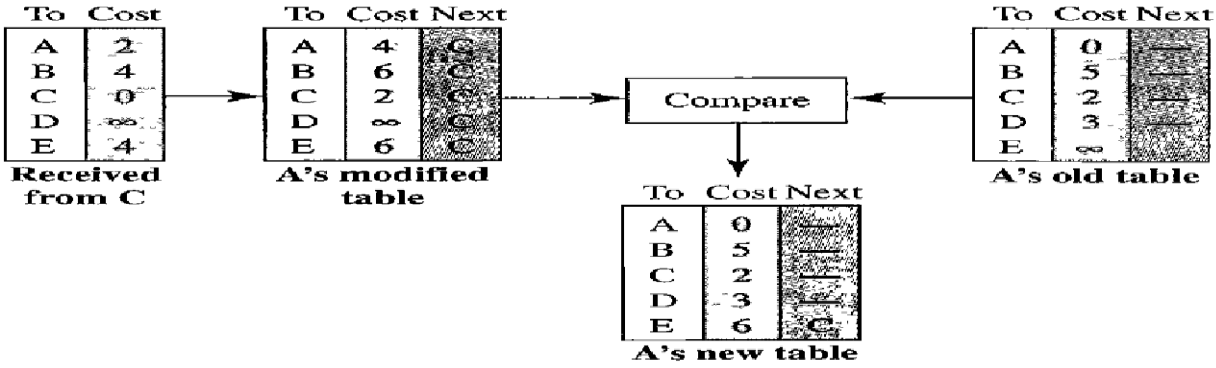
The whole idea of distance vector routing is the sharing of information between neighbors. Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E. On the other hand, node C does not know how to reach node D, but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



*Updating*

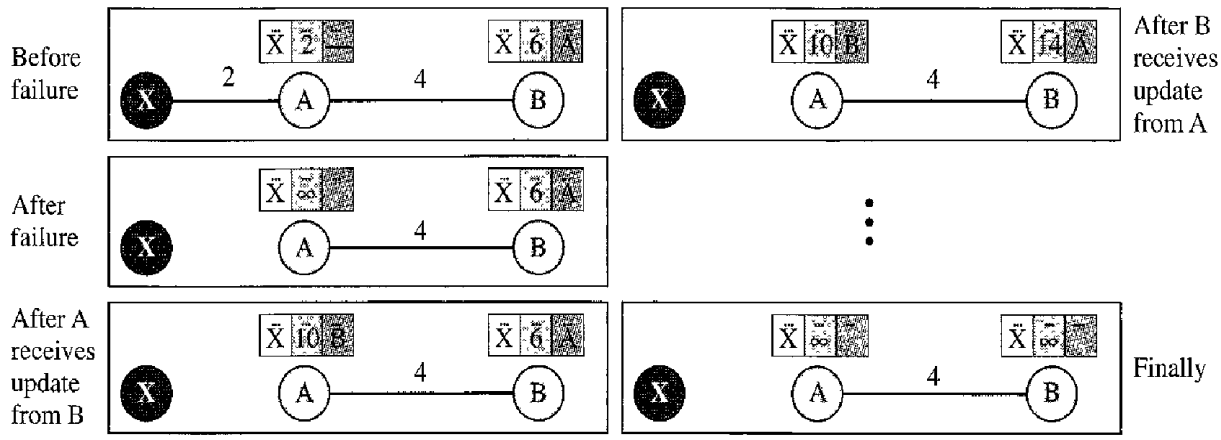
When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:

1. The receiving node needs to add the cost between itself and the sending node to each value in the second column. The logic is clear. If node C claims that its distance to a destination is $x$ mi, and the distance between A and C is $y$ mi, then the distance between A and that destination, via C, is $x + y$ mi.
2. The receiving node needs to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending node is the next node in the route.
3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.
   a. If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.
   b. If the next-node entry is the same, the receiving node chooses the new row. For example, suppose node C has previously advertised a route to node X with distance 3. Suppose that now there is no path between C and X; node C now advertises this route with a distance of infinity. Node A must not ignore this value even though its old entry is smaller. The old route does not exist any more. The new route has a distance of infinity.




*Two-Node Loop Instability*

A problem with distance vector routing is instability, which means that a network using this protocol can become unstable. To understand the problem, let us look at the scenario



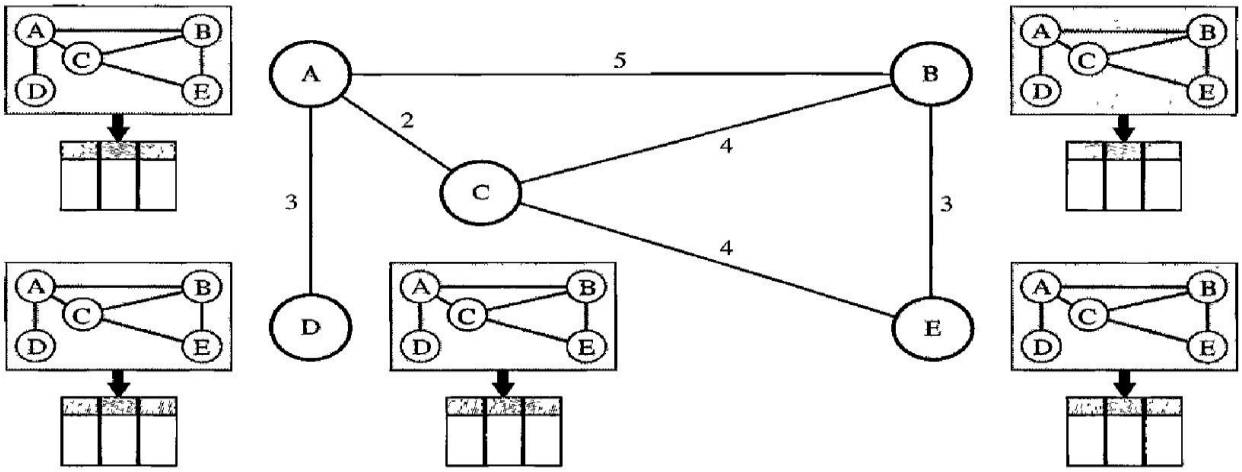

## 2. Routing Information Protocol (RIP)

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:

1. In an autonomous system, we are dealing with routers and networks (links). The routers have routing tables; networks do not.
2. The destination in a routing table is a network, which means the first column defines a network address.
3. 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.**
4. Infinity is defined as 16, which means that any route in an autonomous system using RIP cannot have more than 15 hops.
5. The next-node column defines the address of the router to which the packet is to be sent to reach its destination.

### 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




The figure shows a simple domain with five nodes. 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. This is analogous to a city map. While each person may have the same map, each needs to take a different route to reach her specific destination.

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.

## 4. Path Vector Routing

Distance vector and link state routing are both intradomain routing protocols. They can be used inside an autonomous system, but not between autonomous systems. These two protocols are not suitable for interdomain routing mostly because of scalability. Both of these routing protocols become intractable when the domain of operation becomes large. Distance vector routing is subject to instability if there are more than a few hops in the domain of operation. Link state routing needs a huge amount of resources to calculate routing tables. It also creates heavy traffic because of flooding. There is a need for a third routing protocol which we call **path vector routing.**

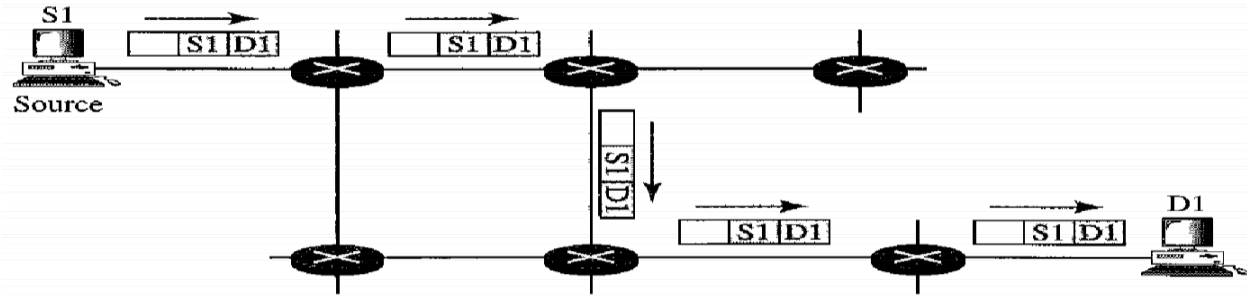
Node A1 is the speaker node for AS1, B1 for AS2, C1 for AS3, and D1 for AS4. Node A1 creates an initial table that shows A1 to A5 are located in AS1 and can be reached through it. Node B1 advertises that B1 to B4 are located in AS2 and can be reached through B1. And so on.

## 7.4 Multicast Routing Protocols

### *Unicasting*

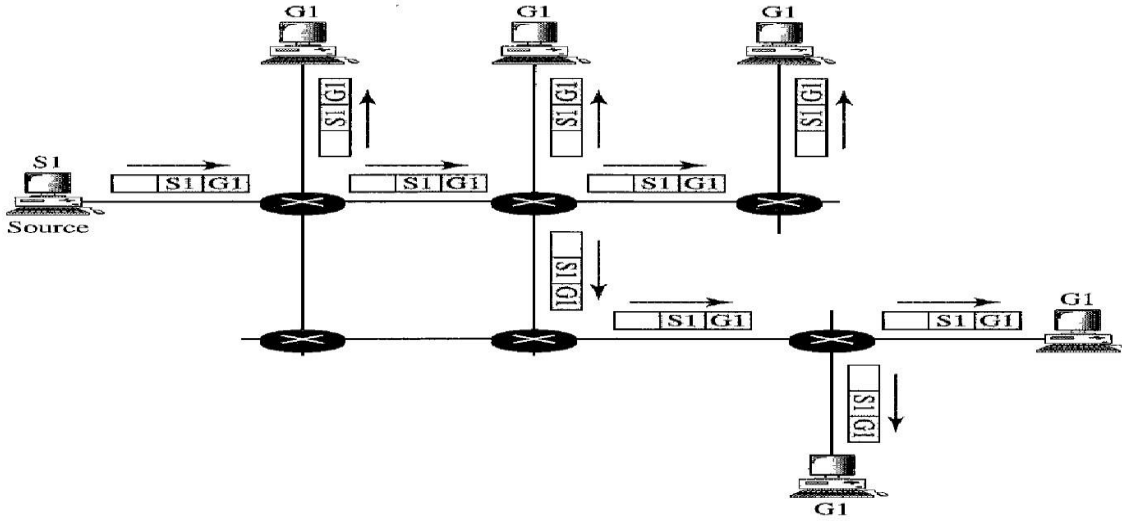
In unicast communication, there is one source and one destination. The relationshi between the source and the destination is one-to-one. In this type of communication both the source and destination addresses, in the IP datagram, are the unicast addresse assigned to the hosts (or host interfaces, to be more exact). In Figure 22.33, a unicas packet starts from the source S1 and passes through routers to reach the destination D1 We have shown the networks as a link between the routers to simplify the figure.

Note that in **unicasting,** when a router receives a packet, it forwards the packe through only one of its interfaces (the one belonging to the optimum path) as defined i the routing table. The router may discard the packet if it cannot find the destinatio address in its routing table.




### *Multicasting*

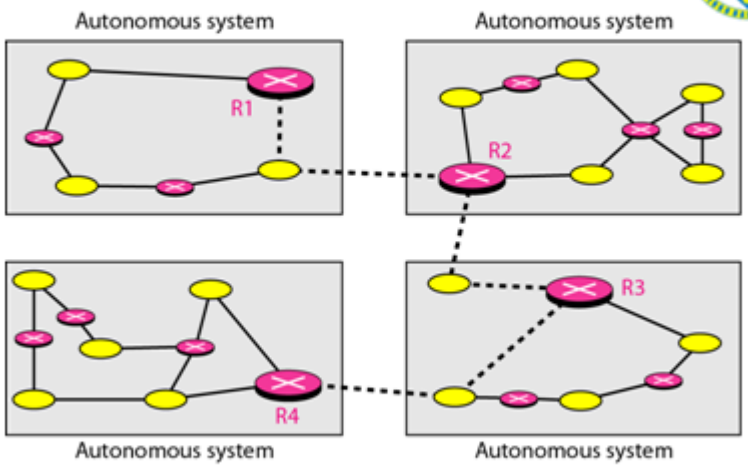
In multicast communication, there is one source and a group of destinations. The rela tionship is one-to-many. In this type of communication, the source address is a unicas address, but the destination address is a group address, which defines one or more desti nations. The group address identifies the members of the group. Figure 22.34 shows th idea behind **multicasting.**

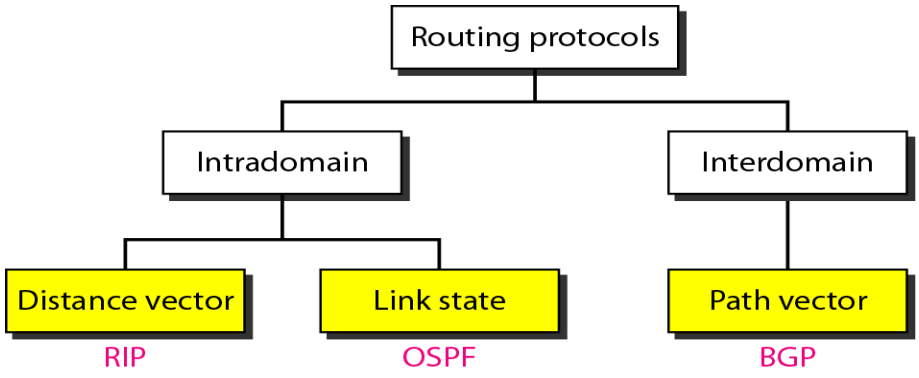


### *Broadcasting*

In broadcast communication, the relationship between the source and the destination is one-to-all. There is only one source, but all the other hosts are the destinations. The Internet does not explicitly support **broadcasting** because of the huge amount of traffic it would create and because of the bandwidth it would need. Imagine the traffic generated in the Internet if one person wanted to send a message to everyone else connected to the Internet.

**Figure** *Autonomous systems*



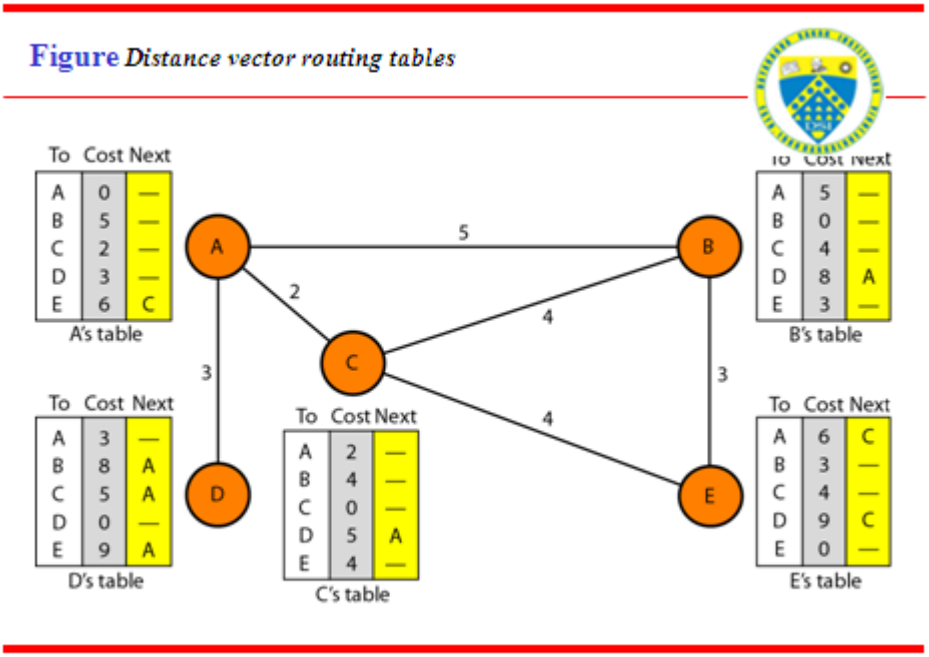
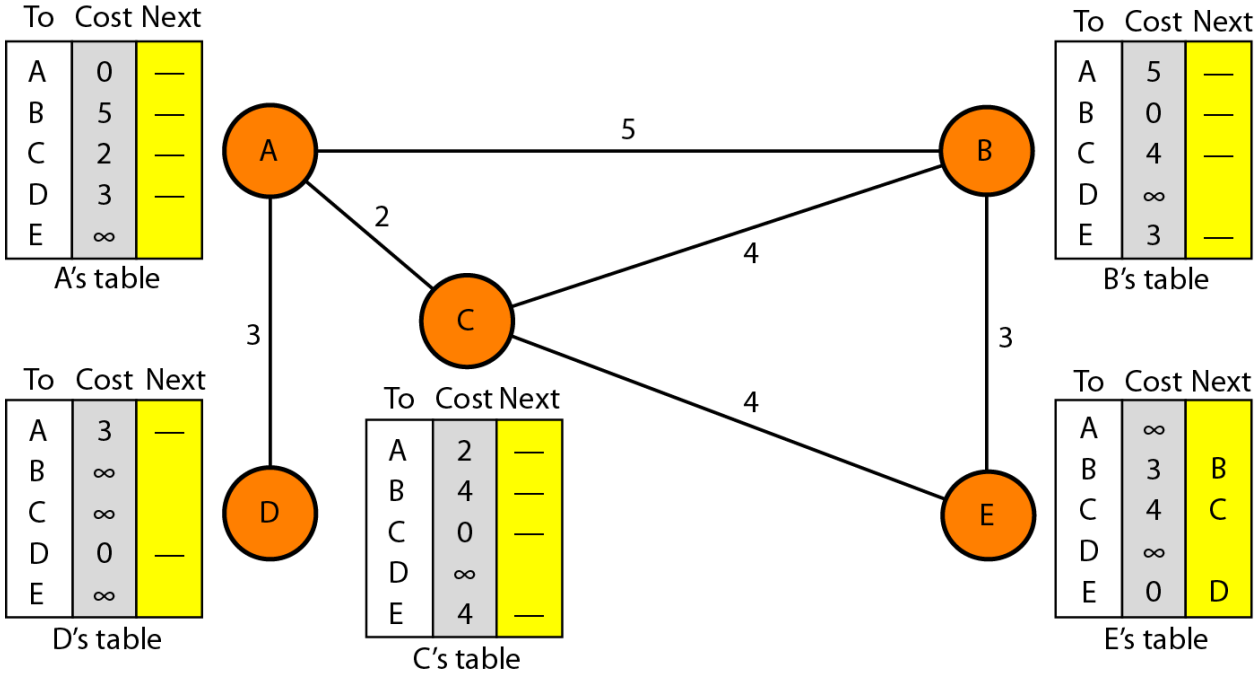
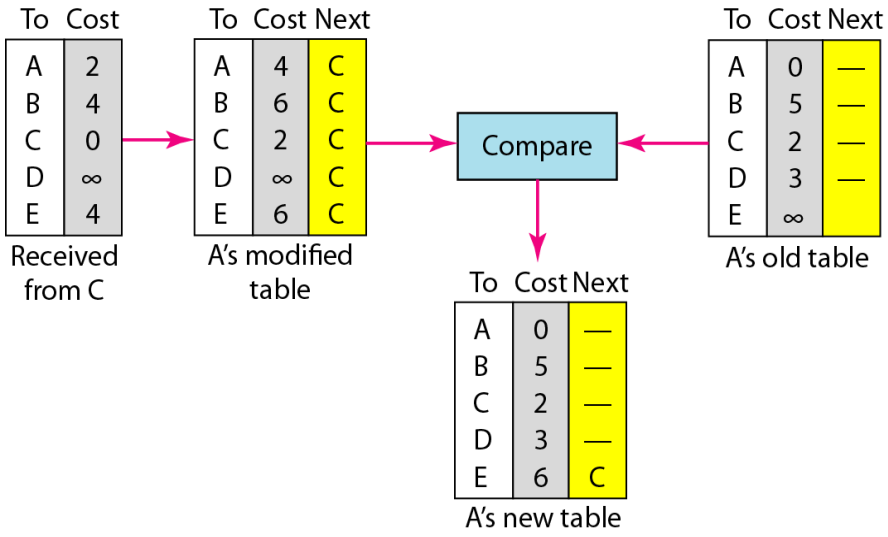


Figure Distance vector routing tables
To Cost Next
A 0 —
B 5 —
C 2 —
D 3 —
E 6 C
A's table
To Cost Next
A 5 —
B 0 —
C 4 —
D 8 A
E 3 —
B's table
To Cost Next
A 3 —
B 8 A
C 5 A
D 0 —
E 9 A
D's table
To Cost Next
A 2 —
B 4 —
C 0 —
D 5 A
E 4 —
C's table
To Cost Next
A 6 C
B 3 —
C 4 —
D 9 C
E 0 —
E's table
A
B
C
D
E
5
2
4
3
3
4




To Cost Next
A 0 —
B 5 —
C 2 —
D 3 —
E ∞
A's table
To Cost Next
A 5 —
B 0 —
C 4 —
D ∞
E 3 —
B's table
To Cost Next
A 3 —
B ∞
C ∞
D 0 —
E ∞
D's table
To Cost Next
A 2 —
B 4 —
C 0 —
D ∞
E 4 —
C's table
To Cost Next
A ∞
B 3 B
C 4 C
D ∞
E 0 D
E's table
A
B
C
D
E
5
2
4
3
3
4

***Updating in distance vector routing***




***Two-node instability***

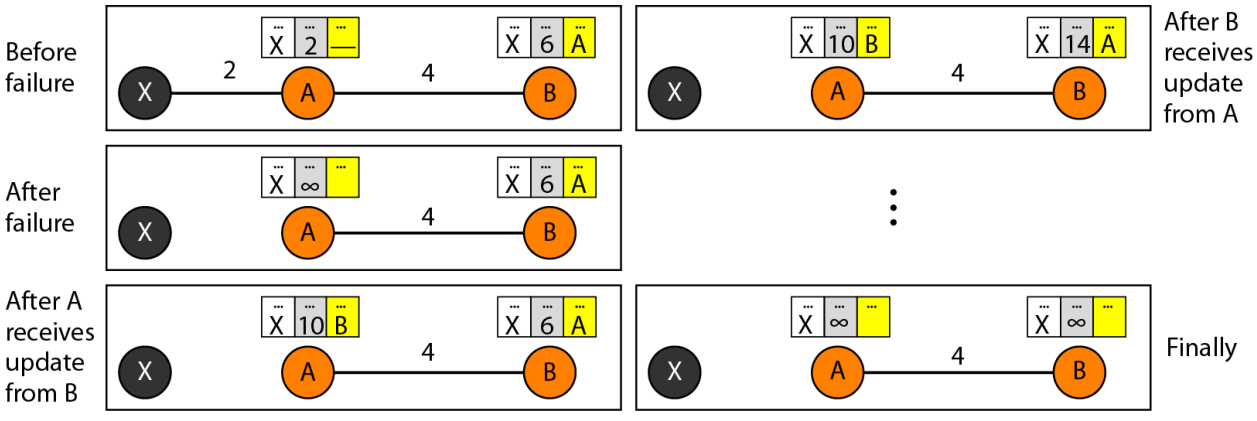
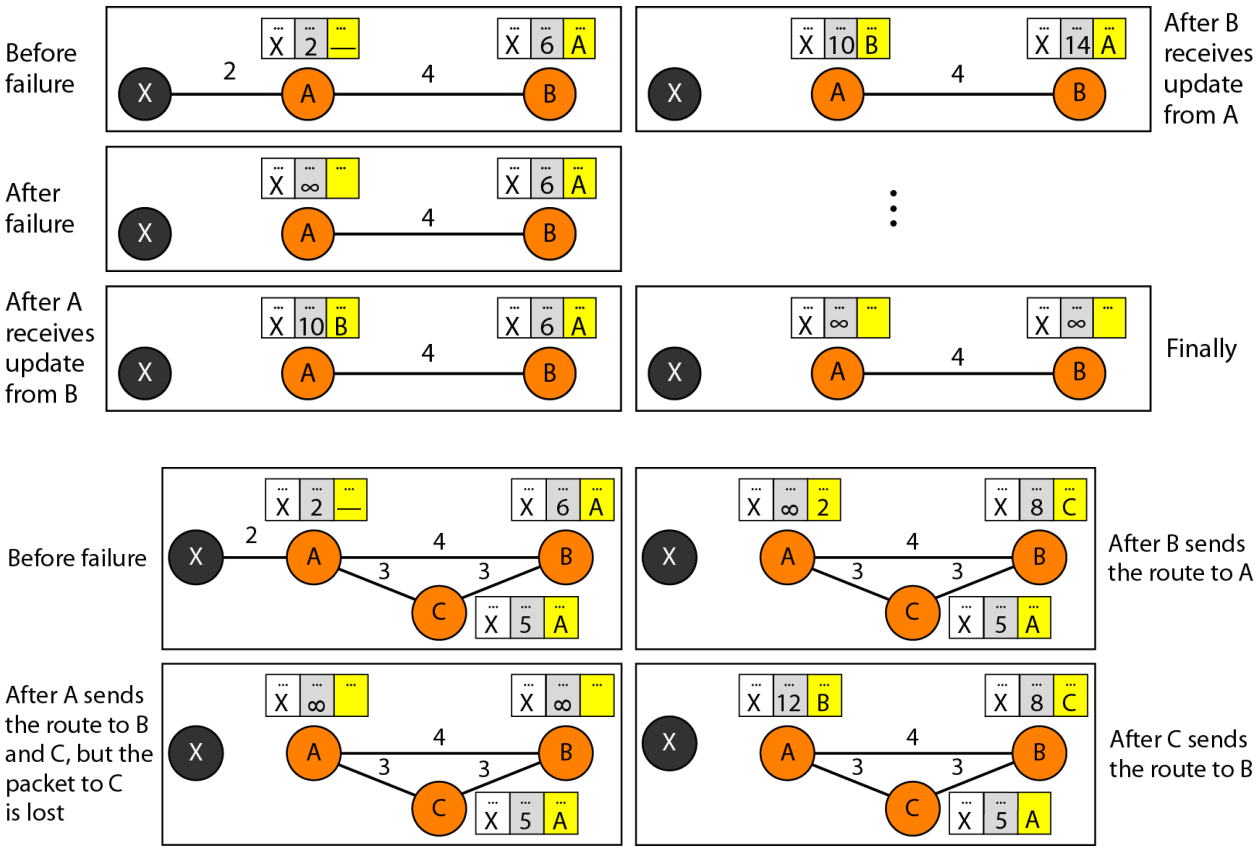
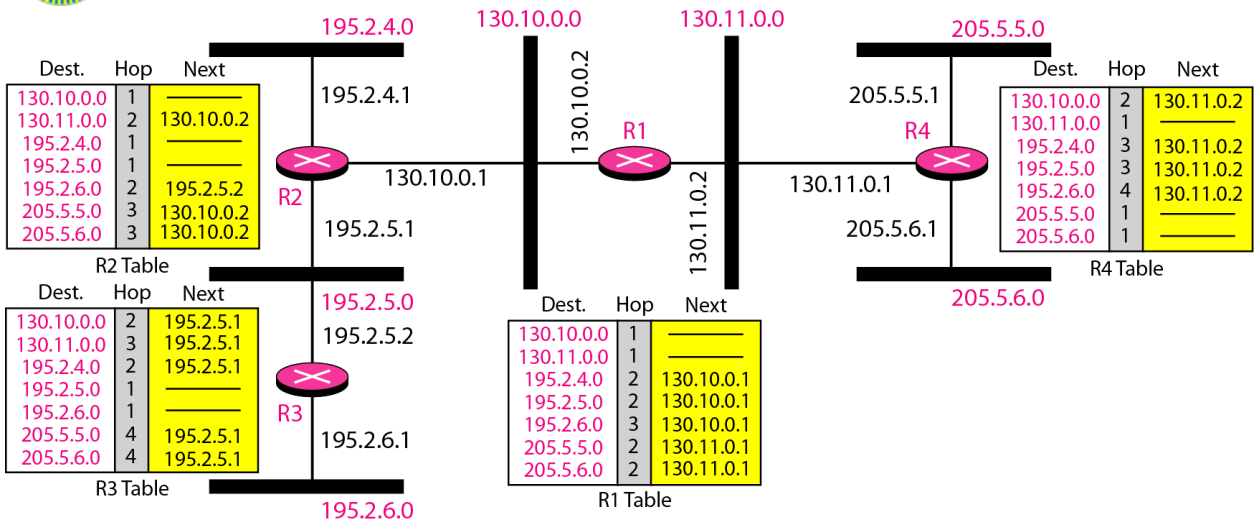


**Figure** ***Two-node instability***




Example of a domain using RIP




## Concept of link state routing

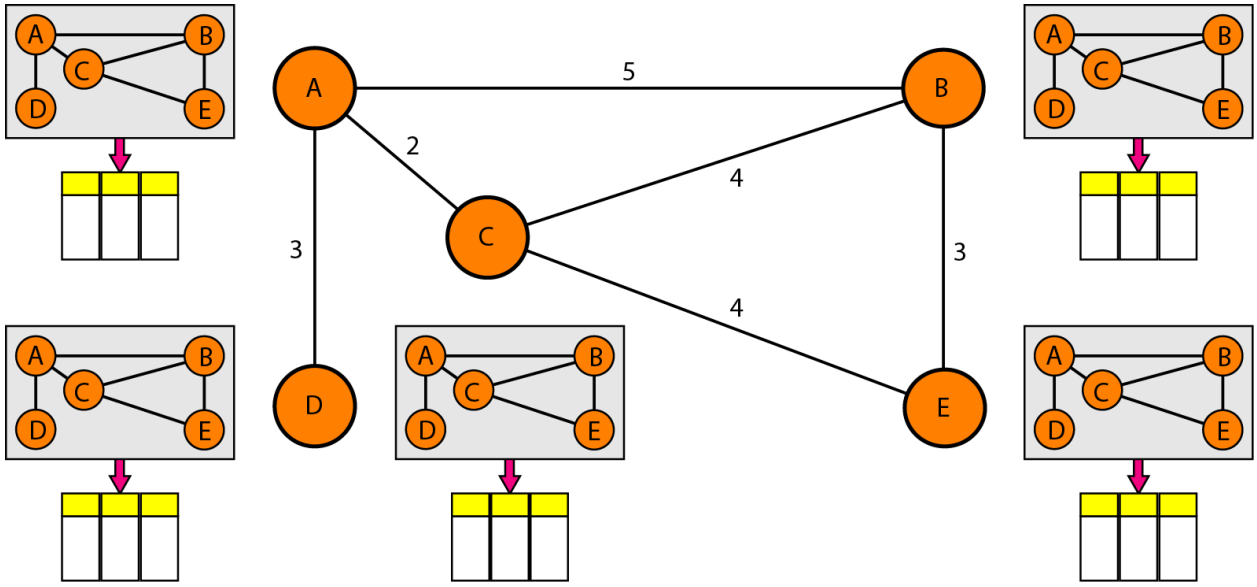
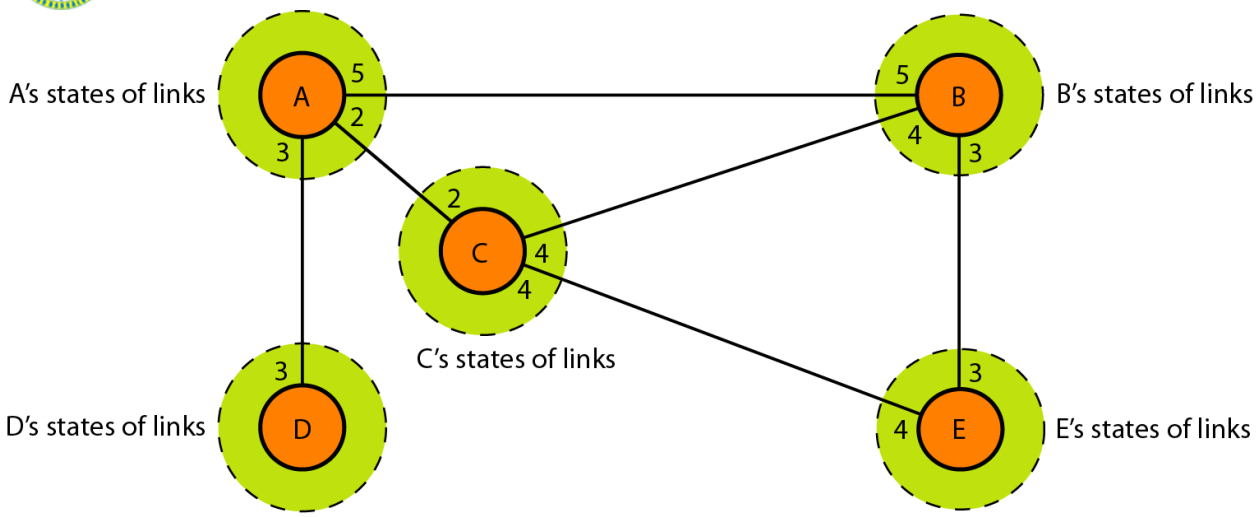
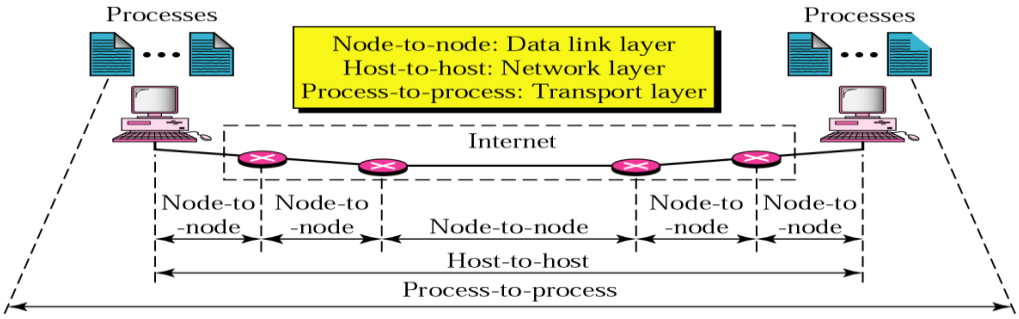



**Figure** ***Link state knowledge***


A's states of links
A
5
2
3
B
5
4
3
B's states of links
C
2
4
4
C's states of links
D
3
D's states of links
E
3
4
E's states of links

## UNIT VIII

### Process – to – process Delivery:

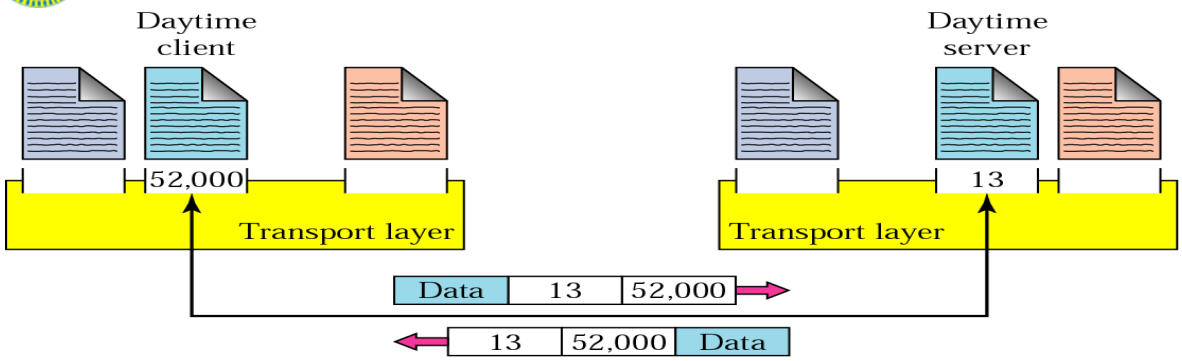



* Real communication in the Internet takes place between two processes.
* Several processes may be running on both the source & destination host.
* A mechanism is required to deliver data from one of these processes running on the source host to the corresponding process running on the destination host.
* The Transport Layer is responsible for process-to-process delivery.
* Two processes communicate in a client/server relationship.

### *Client/Server Paradigm*

* The most common method of achieving Process-to-Process delivery is Client/Server Paradigm.
* A process on the local host, called a client needs service from a process usually on the remote host, called a server.
* Both Client & Server have the same name.
* Local Host, Local Process, Remote Host, Remote Process must be defined for communication.

### *Addressing*




* To choose among the multiple processes running on the Host, Port numbers are required.

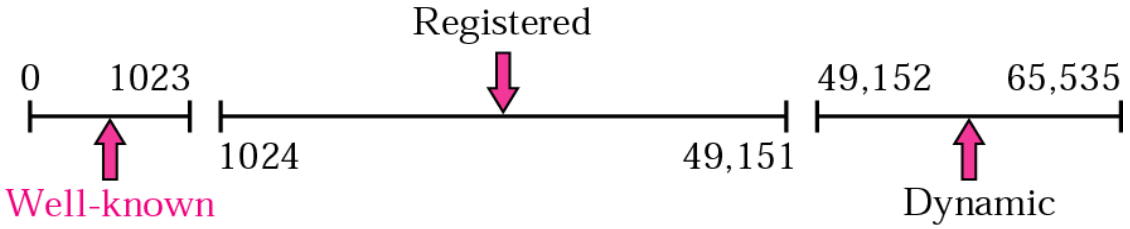
* The destination port number is needed for delivery, and source port number is needed for reply.

* In the Internet model, the port numbers are 16-bit integers: 0 to 65535.

* The client process defines itself with a port number, chosen randomly, called EPHEMERAL (temporary) port number.

* The server process must also define itself with a port number, but not chosen randomly. The server uses the Universal port numbers called WELL-KNOWN port numbers.

***IANA ranges***

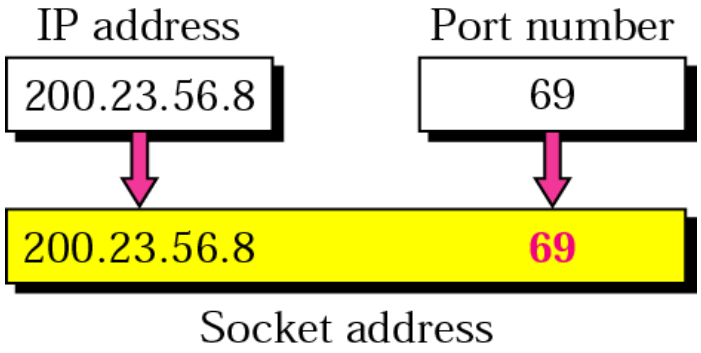



* WELL-KNOWN ports: The ports ranging from 0 to 1023 are assigned and controlled by IANA.

* 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 ports: The ports ranging from 49, 152 to 65,535 are neither controlled nor registered. They can be used by any process. These are the Ephemeral ports.

***Socket addresses***




* A Transport layer protocol needs a pair of socket addresses: the Client socket address and the Server socket address.
* The IP header contains the IP address & the UDP or TCP contains the port numbers.

***Multiplexing:***
* many-to-one relationship at the sender site.
* protocol accepts messages from different processes, differentiated by their assigned port numbers.
* adds the header, & passes the packet to the network layer.

***De-multiplexing:***
* one-to-many relationship at the receiver site.
* protocol accepts datagram from the network layer, checks errors, drops the header and delivers each message to the appropriate process based on the port number.

***Connectionless Vs Connection-oriented service***

***Connectionless:***
* no connection establishment prior to the data transfer.
* the packets may be delayed or lost or may arrive out of sequence.
* Ex: UDP

***connection-oriented:***
* a connection is first established b/w the sender and the receiver, data are transferred and then the connection is released.
* Ex: TCP, SCTP

***Reliable Vs Unreliable***

***Reliable:***
* Reliable protocol guarantees successful data transmission by implementing flow & error control mechanisms

***Unreliable:***

* doesn't guarantee successful data transmission. It doesn't implement any error or flow control mechanism

***USER DATAGRAM PROTOCOL (UDP)***

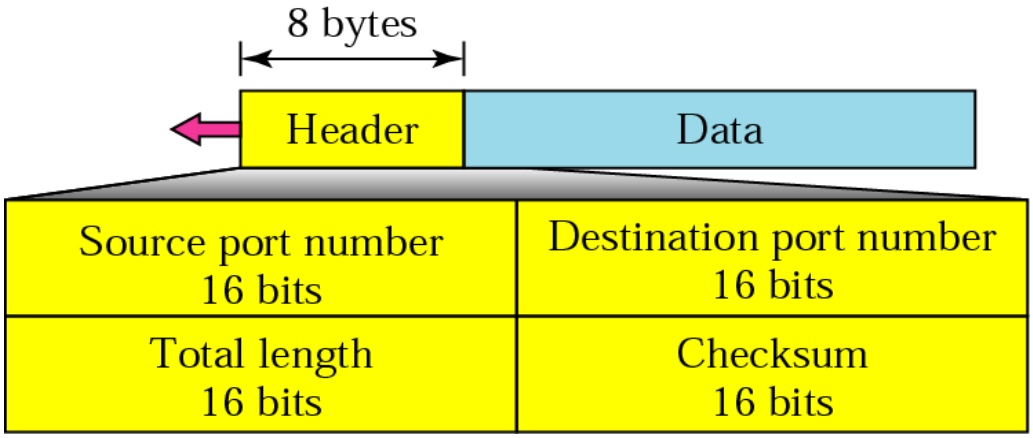
* Is called a connectionless, unreliable transport protocol.
* it doesn't add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication.
* it performs very limited error checking.
* if a process wants to send a small message and doesn't care much about reliability, it can use UDP.

***WELL-KNOWN port numbers for UDP***

| *Port* | *Protocol* | *Description* |
|---|---|---|
| **7** | **Echo** | **Echoes a received datagram back to the sender** |
| **9** | **Discard** | **Discards any datagram that is received** |
| **11** | **Users** | **Active users** |
| **13** | **Daytime** | **Returns the date and the time** |
| **17** | **Quote** | **Returns a quote of the day** |
| **19** | **Chargen** | **Returns a string of characters** |
| **53** | **Nameserver** | **Domain Name Service** |
| **67** | **Bootps** | **Server port to download bootstrap information** |
| **68** | **Bootpc** | **Client port to download bootstrap information** |
| **69** | **TFTP** | **Trivial File Transfer Protocol** |
| **111** | **RPC** | **Remote Procedure Call** |
| **123** | **NTP** | **Network Time Protocol** |
| **161** | **SNMP** | **Simple Network Management Protocol** |
| **162** | **SNMP** | **Simple Network Management Protocol (trap)** |

**User Datagram format**

* UDP packets, called user datagrams have a fixed-size header of 8 bytes.




***Source port number:***

* used by the processes running on the source host.
* it is 16-bit long: 0 to 65,535
* if the source host is client, the port number is an ephemeral port number requested by process & chosen by UDP software running on the host.
* if the source host is server, the port number is an well-known.

***Destination port number:***

* used by the processes running on the destination host.
* it is 16-bit long: 0 to 65,535
* if the destination host is client, the port number is an ephemeral port number requested by process & chosen by UDP software running on the host.
* if the destination host is server, the port number is an well-known.

***Total Length:***

* it is 16-bit long, defines the total length of the user datagram plus data.
* 16-bits can define a total length of 0 to 65,535 bytes, but the total length needs to be much less, because a UDP datagram is stored in a IP datagram with a total length of 65,535 bytes.

***Checksum:***

* used to detect the errors over the entire datagram.
* calculation & inclusion of the checksum in a user datagram is optional.
* if the checksum is not calculated, the field is filled with 1s.

***TRANSMISSION CONTROL PROTOCOL***

* TCP is a connection-oriented, reliable transport protocol.
* it adds connection-oriented and reliability features to the services of IP.

***TCP services:***

i) Process –to –process communication:

* TCP provides process-to-process communication using the port numbers.

ii) Stream Delivery Service:

* TCP is a stream-oriented protocol – allows sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.

***TCP Services***

**Sending and Receiving buffers:**

* because the sending & receiving processes may not write or read data at the same speed, TCP needs buffers for storage.
* two buffers for sending and receiving.
* one way to implement a buffer is to use a circular array of 1-byte locations.

**Segments:**

* the IP 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 and delivers to the IP layer.
* the segments are encapsulated in the IP datagram and transmitted.

***iii) Full-Duplex Communication:***

* TCP offers full-duplex service, in which data can flow in both directions at the same time.

***iv) Connection-oriented service:***

* when a process at site A wants to send and receive data from another process at site B, the following occurs.
  1. the 2 TCPs establish a connection between them.
  2. data are exchanged in both directions.
  3. the connection is terminated.

***v) Reliable service:***

it uses an acknowledgement mechanism to check the safe and sound arrival of data.

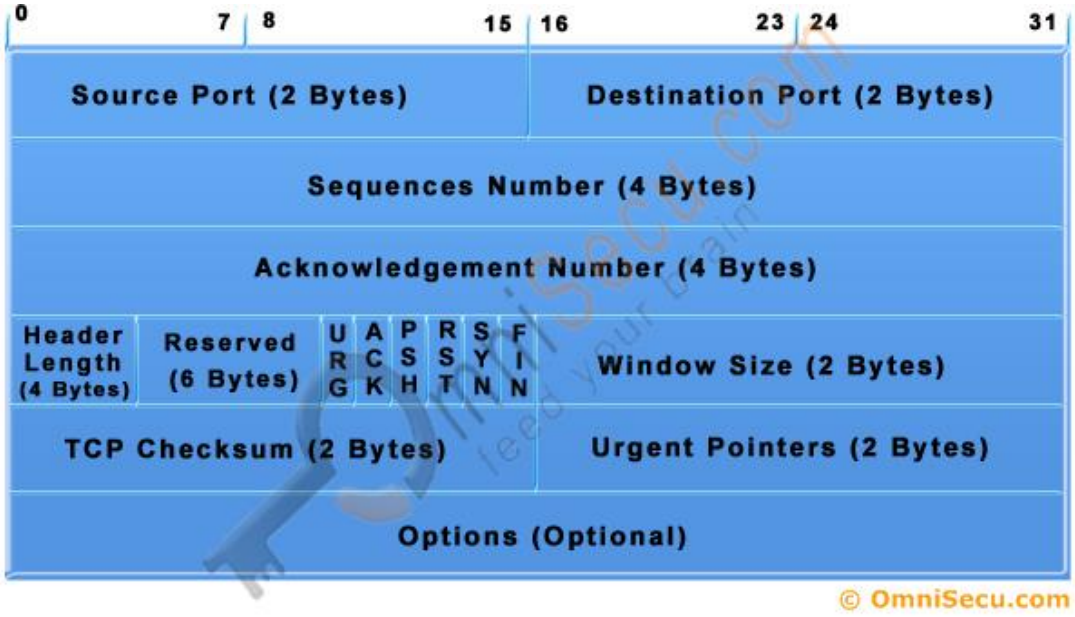
Transmission Control Protocol (TCP) is responsible for breaking up the message (Data from application layer) into TCP Segments and reassembling them at the receiving side. It

is not sure that the data reaching at the receiving device is in the same order as the sending side, because of the problems in network or different paths packets flow to the destination. TCP is responsible for keeping the unordered segments in the right order. TCP assures a reliable delivery by resending anything that gets lost while traveling the network.

*** haracteristics of Transmission Control Protocol (TCP).**

* Stream Data transfer: Applications working at the Application Layer transfers a contiguous stream of bytes to the bottom layers. It is the duty of TCP to pack this byte stream to packets, known as TCP segments, which are passed to the IP layer for transmission to the destination device. The application does not have to bother to chop the byte stream data packets.
* Reliability: The most important feature of TCP is reliable data delivery. In order to provide reliability, TCP must recover from data that is damaged, lost, duplicated, or delivered out of order by the Network Layer. TCP assigns asequence number to each byte transmitted, and expects a positive acknowledgment (ACK) from the receiving TCP layer. If the ACK is not received within a timeout interval, the data is retransmitted. The receiving TCP uses the sequence numbers to rearrange the TCP segments when they arrive out of order, and to eliminate duplicate TCP segments.
* Flow control: Network devices operate at different data rates because of various factors like CPU and available bandwidth. It may happen a sending device to send data at a much faster rate than the receiver can handle. TCP uses a sliding window mechanism for implementing flow control. The number assigned to a segment is called the sequence number and this numbering is actually done at the byte level. The TCP at the receiving device, when sending an ACK back to the sender, also indicates to the TCP at the sending device, the number of bytes it can receive (beyond the last received TCP segment) without causing serious problems in its internal buffers.
* Multiplexing: Multitasking achieved through the use of port numbers.
* Connections: Before application processes can send data by using TCP, the devices must establish a connection. The connections are made between the port numbers of the sender and the receiver devices. A TCP connection identifies the end points involved in the connection. A socket number is a combination of IP address and port number, which can uniquely identify a connection.
* Full duplex: TCP provides for concurrent data streams in both directions

***TCP Segment format***




source port: 16 Bit number which identifies the Source Port number (Sending Computer's TCP Port).

Destination port: 16 Bit number which identifies the Destination Port number (Receiving Port).

Sequence number: 32 Bit number used for byte level numbering of TCP segments. If you are using TCP, each byte of data is assigned a sequence number. If SYN flag is set (during the initial three way handshake connection initiation), then this is the initial sequence number. The sequence number of the actual first data byte will then be this sequence number plus 1. For example, let the first byte of data by a device in a particular TCP header will have its sequence number in this field 50000. If this packet has 500 bytes of data in it, then the next packet sent by this device will have the sequence number of 50000 + 500 + 1 = 50501.

Acknowledgment Number: 32 Bit number field which indicates the next sequence number that the sending device is expecting from the other device.

Header Length: 4 Bit field which shows the number of 32 Bit words in the header. Also known as the Data Offset field. The minimum size header is 5 words (binary pattern is 0101).Reserved: Always set to 0 (Size 6 bits).

Control Bit Flags: We have seen before that TCP is a Connection Oriented Protocol. The meaning of Connection Oriented Protocol is that, before any data can be transmitted, a reliable connection must

be obtained and acknowledged. Control Bits govern the entire process of connection establishment, data transmissions and connection termination. The control bits are listed as follows: They are:

URG: Urgent Pointer.

ACK: Acknowledgement.

PSH: This flag means Push function. Using this flag, TCP allows a sending application to specify that the data must be pushed immediately. When an application requests the TCP to push data, the TCP should send the data that has accumulated without waiting to fill the segment.

RST: Reset the connection. The RST bit is used to RESET the TCP connection due to unrecoverable errors. When an RST is received in a TCP segment, the receiver must respond by immediately terminating the connection. A RESET causes both sides immediately to release the connection and all its resources. As a result, transfer of data ceases in both directions, which can result in loss of data that is in transit. A TCP RST indicates an abnormal terminination of the connection.

SYN: This flag means synchronize sequence numbers. Source is beginning a new counting sequence. In other words, the TCP segment contains the sequence number of the first sent byte (ISN).

FIN: No more data from the sender. Receiving a TCP segment with the FIN flag does not mean that transferring data in the opposite direction is not possible. Because TCP is a fully duplex connection, the FIN flag will cause the closing of connection only in one direction. To close a TCP connection gracefully, applications use the FIN flag.

Window: indicates the size of the receive window, which specifies the number of bytes beyond the sequence number in the acknowledgment field that the receiver is currently willing to receive.

Checksum: The 16-bit checksum field is used for error-checking of the header and data.

Urgent Pointer: Shows the end of the urgent data so that interrupted data streams can continue. When the URG bit is set, the data is given priority over other data streams (Size 16 bits).

In this lesson, you have learned different fields in Transmission Control Protocol (TCP) Segment Header and the use of these fields. The fields in Transmission Control Protocol (TCP) Segment Header are Source Port, Destination Port, Sequence Number, Acknowledgement Number, Header Length, Flags, Window Size, TCP Checksum and Urgent pointer

| Flag | Description |
|---|---|
| URG | The value of the urgent pointer field is valid. |
| ACK | The value of the acknowledgment field is valid. |
| PSH | Push the data. |
| RST | The connection must be reset. |
| SYN | Synchronize sequence numbers during connection. |
| FIN | Terminate the connection. |

* In TCP connection-oriented transmission requires three phases:
  1. Connection Establishment
  2. Data Transfer
  3. Connection Termination

Connection Establishment

Before the sending device and the receiving device start the exchange of data, both devices need to be synchronized. During the TCP initialization process, the sending device and the receiving device exchange a few control packets for synchronization purposes. This exchange is known as Three-way handshake.

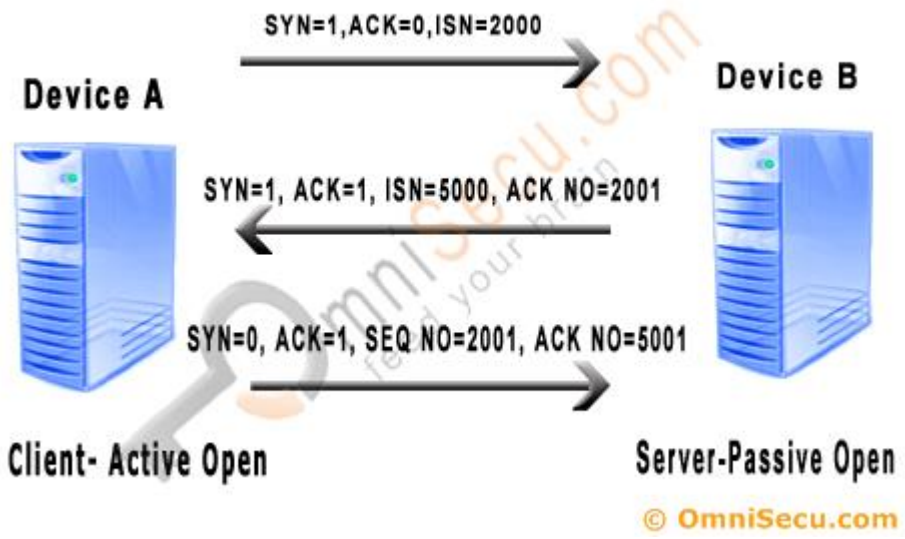
The Three-way handshake begins with the initiator sending a TCP segment with the SYN control bit flag set.

TCP allows one side to establish a connection. The other side may either accept the connection or refuse it. If we consider this from application layer point of view, the side that is establishing the connection is the client and the side waiting for a connection is the server.

TCP identifies two types of OPEN calls:

Active Open. In an Active Open call a device (client process) using TCP takes the active role and initiates the connection by sending a TCP SYN message to start the connection.

Passive Open A passive OPEN can specify that the device (server process) is waiting for an active OPEN from a specific client. It does not generate any TCP message segment. The server processes listening for the clients are in Passive Open mode.




**TCP Three-way Handshake**

Step 1. Device A (Client) sends a TCP segment with SYN = 1, ACK = 0, ISN (Initial Sequence Number) = 2000.

An Initial Sequence Number (ISN) is a random Sequence Number, allocated for the first packet in a new TCP connection.

The Active Open device (Device A) sends a segment with the SYN flag set to 1, ACK flag set to 0 and an Initial Sequence Number 2000 (For Example), which marks the beginning of the sequence numbers for data that device A will transmit. SYN is short for SYNchronize. SYN flag announces an attempt to open a connection.

Step 2. Device B (Server) receives Device A's TCP segment and returns a TCP segment with SYN = 1, ACK = 1, ISN = 5000 (Device B's Initial Sequence Number), Acknowledgment Number = 2001 (2000 + 1, the next sequence number Device B expecting from Device A).

Step 3. Device A sends a TCP segment to Device B that acknowledges receipt of Device B's ISN, With flags set as SYN = 0, ACK = 1, Sequence number = 2001, Acknowledgment

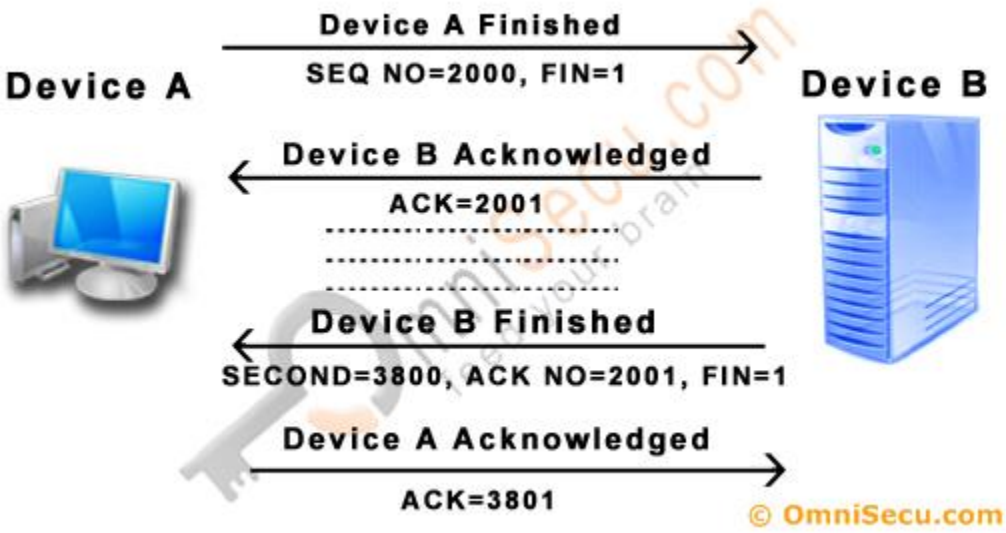
number = 5001 (5000 + 1, the next sequence number Device A expecting from Device B)

This handshaking technique is referred to as TCP Three-way handshake or SYN, SYN-ACK, ACK.

After the Three-way handshake, the connection is open and the participant computers start sending data using the agreed sequence and acknowledge numbers.

### *Connection Termination*

When the data transmission is complete and the device want to terminate the connection, the device initiating the termination, places a TCP segment (Segment is the name of the data packet at transport layer, if the protocol is TCP) with the FIN flag set to one. The purpose of FIN bit is to enable TCP to gracefully terminate an established session. The application then enters in a state called the FIN-WAIT state. When at FIN-WAIT state, Device A continues to receive TCP segments from Device B and processes the segments already in the queue, but no additional data is accepted from the application.




**TCP Connection Termination**

In the example shown above, assume Device A has completed its transmission and indicates this by sending a segment to Device B with the FIN bit set to 1. Device B will acknowledge the segment with an ACK. At this point in time, Device B will no longer accept data from Device A. Device B can continue to accept data from its application to transmit to Device A. If Device B does not have any more data to transmit, it will also terminate the connection by transmitting a segment to Device A with the FIN bit set to 1. Device A will then ACK that segment and terminates the connection.

# Difference between Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)

## Transmission Control Protocol (TCP)

1) Transmission Control Protocol (TCP) is a connection oriented protocol, which means the devices should open a connection before transmitting data and should close the connection gracefully after transmitting the data.

2) Transmission Control Protocol (TCP) assure reliable delivery of data to the destination.

3) Transmission Control Protocol (TCP) protocol provides extensive error checking mechanisms such as flow control and acknowledgment of data.

4) Sequencing of data is a feature of Transmission Control Protocol (TCP).

5) Delivery of data is guaranteed if you are using Transmission Control Protocol (TCP).

6) Transmission Control Protocol (TCP) is comparatively slow because of these extensive error checking mechanisms

7) Multiplexing and Demultiplexing is possible in Transmission Control Protocol (TCP) using TCP port numbers.

8) Retransmission of lost packets is possible in Transmission Control Protocol (TCP).

## User Datagram Protocol (UDP)

1) User Datagram Protocol (UDP) is Datagram oriented protocol with no overhead for opening a connection (using three-way handshake), maintaining a connection, and closing (terminating) a connection.

2) User Datagram Protocol (UDP) is efficient for broadcast/multicast type of network transmission.

3) User Datagram Protocol (UDP) has only the basic error checking mechanism using checksums.

4) There is no sequencing of data in User Datagram Protocol (UDP).

5) The delivery of data cannot be guaranteed in User Datagram Protocol (UDP).

6) User Datagram Protocol (UDP) is faster, simpler and more efficient than TCP. However, User Datagram Protocol (UDP) it is less robust then TCP

7) Multiplexing and Demultiplexing is possible in User Datagram Protocol (UDP) using UDP port numbers.

8) There is no retransmission of lost packets in User Datagram Protocol (UDP).

# UNIT – VIII

**1.** a)Explain connection establishment and connection termination in TCP **(July 2013/2011 10 marks)**

b)Write a note on DNS **(July 2013 10marks)**

**2.** a)Compare the TCP header and UDP header. List the fields in TCP header that are missing from UDP header. Give the reason for their absence **(Jan 2013 10 marks)**

b)What are the three domains of domain name space? Explain **(Jan 2013 10 marks)**

**3.** a)Explain in detail UDP **(July 2012 10 marks)**

b) Describe DNS in the internet **(July 2012/2011 10 marks)**

**4.** a)Explain the user datagram format. (**Dec 2011 05 marks)**

b)How does recursion resolution differ from iterative resolution **(Jan 2013 05marks)**

c) Explain the features of TCP. **(Dec 2011 10 marks)**

**5.** a)Suppose a TCP connection is transferring a file of 5000 bytes, the 1st byte is numbered 10,001. What are the sequence nos of each segment, if data are sent in 5 segments each carrying 1000 bytes? **(Dec 2011 10 marks)**

**b)**Explain TCP and UDP datagram (**Jun 2014 10 marks)**

**6.** Write a short note two of the following: UDP, TCP, (**Dec 2010 20 marks)**

b) Write short notes on any two of the following:

a. UDP

b. TCP segment format

c. IPV4 datagram format **(June 2010 20 marks)**

**7.** a)Expalin TCP and UDP datagram **(June 2014 10 marks)**

**b)**Describe TCP connection establishment using three way handshake **(June 2014 10 marks)**

**8.** **a)**How is TCP better than UDP .Explain services offered by TCP **(Dec 2014 10 marks)**

**b)**What is name space ?How it is classified?What is DNS **(Dec 2014 10 marks)**

**9.** Write a short note on any two of the following:DNS, TCP, (**Dec 2010 20 marks)**

**10.** **a)**Explain TCP connection establishment (**Jun 2014 10marks)**

b) Write a short note on source port number and destination port number in user datagram. **(July 2013 06 marks)**