DEPARTMENT OF CSE

II YEAR – IV SEMESTER

CS6551- COMPUTER NETWORKS

LECTURE NOTES

CS6551 COMPUTER NETWORKS **L T P C 3 0 0 3**

UNIT I FUNDAMENTALS & LINK LAYER **9**

Building a network – Requirements - Layering and protocols - Internet Architecture – Network software – Performance ; Link layer Services - Framing - Error Detection - Flow control

UNIT II MEDIA ACCESS & INTERNETWORKING **9**

Media access control - Ethernet (802.3) - Wireless LANs – 802.11 – Bluetooth - switching and bridging – Basic Internetworking (IP, CIDR, ARP, DHCP,ICMP)

UNIT III ROUTING **9**

Routing (RIP, OSPF, metrics) – Switch basics – Global Internet (Areas, BGP, IPv6), Multicast – addresses – multicast routing (DVMRP, PIM)

UNIT IV TRANSPORT LAYER **9**

Overview of Transport layer - UDP - Reliable byte stream (TCP) – Connection management – Flow control - Retransmission – TCP Congestion control - Congestion avoidance (DECbit, RED) – QoS – Application requirements

UNIT V APPLICATION LAYER 9

Traditional applications -Electronic Mail (SMTP, POP3, IMAP, MIME) – HTTP – Web Services – DNS - SNMP

TOTAL: 45 PERIODS

TEXT BOOK:

1. Larry L. Peterson, Bruce S. Davie, “Computer Networks: A Systems Approach”, Fifth Edition, Morgan Kaufmann Publishers, 2011.

REFERENCES:

1. James F. Kurose, Keith W. Ross, “Computer Networking - A Top-Down Approach Featuring the Internet”, Fifth Edition, Pearson Education, 2009.

2. Nader. F. Mir, “Computer and Communication Networks”, Pearson Prentice Hall Publishers, 2010.

3. Ying-Dar Lin, Ren-Hung Hwang, Fred Baker, “Computer Networks: An Open Source Approach”,Mc Graw Hill Publisher, 2011.

4. Behrouz A. Forouzan, “Data communication and Networking”, Fourth Edition, Tata McGraw – Hill,2011.

UNIT I

Introduction

Computer Network:

Collection of autonomous computers interconnected by single technology.

Connectivity:

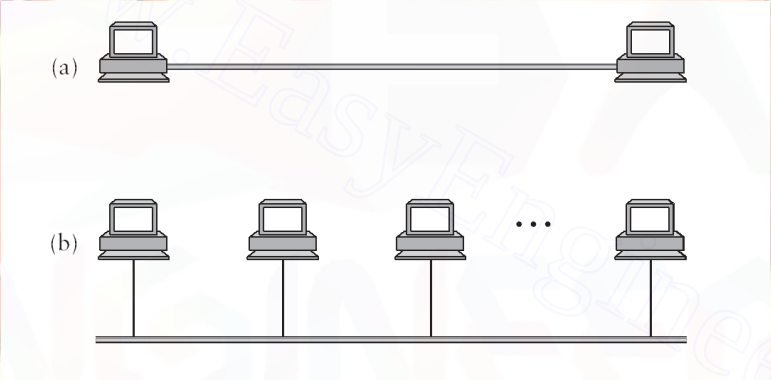
Connectivity occurs between two computers through physical medium like coaxial cable or an optical fiber.

Physical Medium - Link

Computers - Nodes

When a physical link occurs between a pair of nodes then it is referred as **point-to-point.**

When more than two nodes share a single physical link then it is referred as **Multiple access**.



(a) point-to-point (b) Multiple-access.

Data communication between the nodes is done by forwarding the data from one link to another. The systematic way of organizing these forwarding nodes form a ***switched network.***

Two common types of switched network are

- Circuit switched – e.g. Telephone System
- Packet switched – e.g. Postal System

Packet Switched Network

In this network nodes send discrete blocks of data to each other. These blocks can be called as ***packet or message.***

Store and forward strategy:

This network follows this technique. It means "Each node receives a complete packets over the link, stores in internal memory and then forwards to next node".

Circuit Switched Network

It first establishes a circuit across the links and allows source node to send stream of bits across this circuit to the destination node

The representation of network is given by cloud symbol

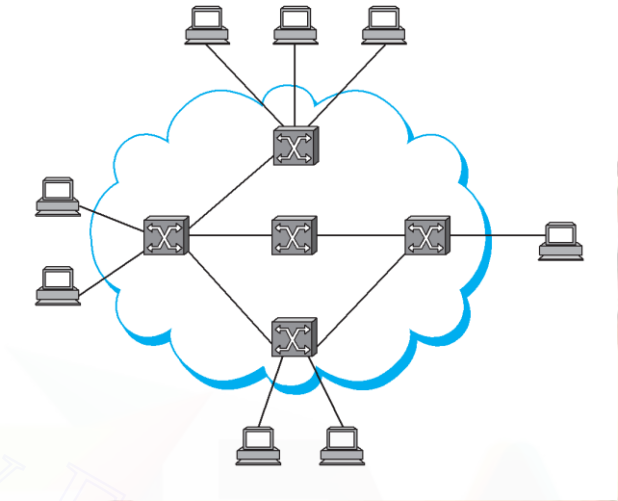


Fig: Switched network.

Cloud represents the network

Nodes inside the cloud (Switches) – Implement the network

Nodes outside the cloud (host) - Use the network

Internetwork

Set of independent network are interconnected to form inter network or **internet.** Node that is connected to two or more network is called **router or gateway.** It is responsible for forwarding data between the networks.

Applications

Most people know about the Internet (a computer network) through applications

- World Wide Web
- Email
- Online Social Network
- Streaming Audio Video
- File Sharing
- Instant Messaging

Application Protocol

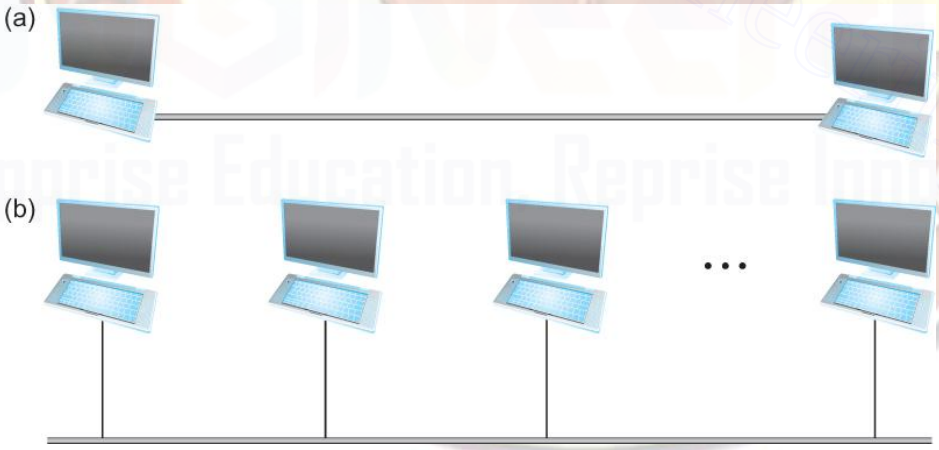
- URL
  - Uniform resource locater
  - http://www.cs.princeton.edu/~llp/index.html

- HTTP
  - Hyper Text Transfer Protocol
- TCP
  - Transmission Control Protocol
- 17 messages for one URL request
  - 6 to find the IP (Internet Protocol) address
  - 3 for connection establishment of TCP
  - 4 for HTTP request and acknowledgement
    - Request: I got your request and I will send the data
    - Reply: Here is the data you requested; I got the data
  - 4 messages for tearing down TCP connection

Requirements

- Application Programmer
  - List the services that his application needs: delay bounded delivery of data
- Network Designer
  - Design a cost-effective network with sharable resources
- Network Provider
  - List the characteristics of a system that is easy to manage

Connectivity



(a) Point-to-point

(b) Multiple access

Need to understand the following terminologies

(a) Scale: A system that is designed to support growth to an arbitrarily large size is said to be scale.

(b) Link: It is the physical and logical component used to interconnect hosts in the network.

(c) Nodes: It is any device connected to a network

(d) Point-to-point

(e) Multiple access

(f) Switched Network

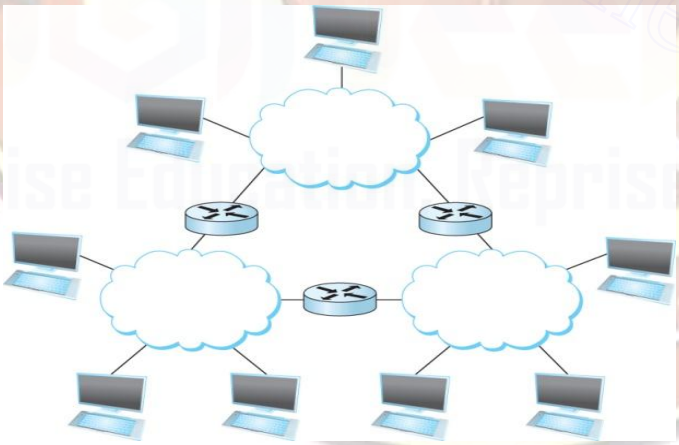
(a) Circuit Switched

(b) Packet Switched

(g) Packet, message

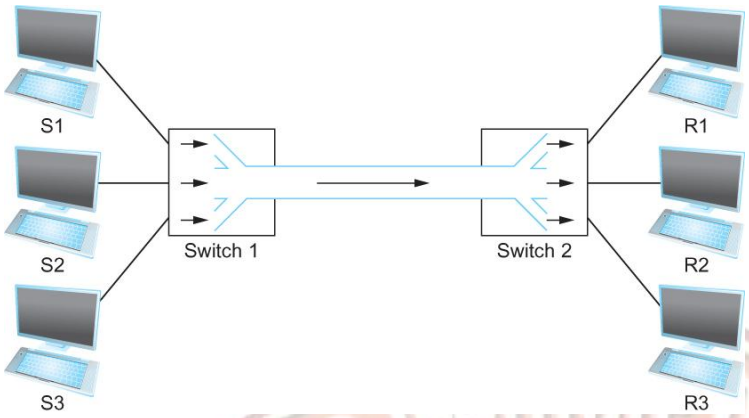
(h) Store-and-forward

(a) A switched network



(a) Interconnection of networks

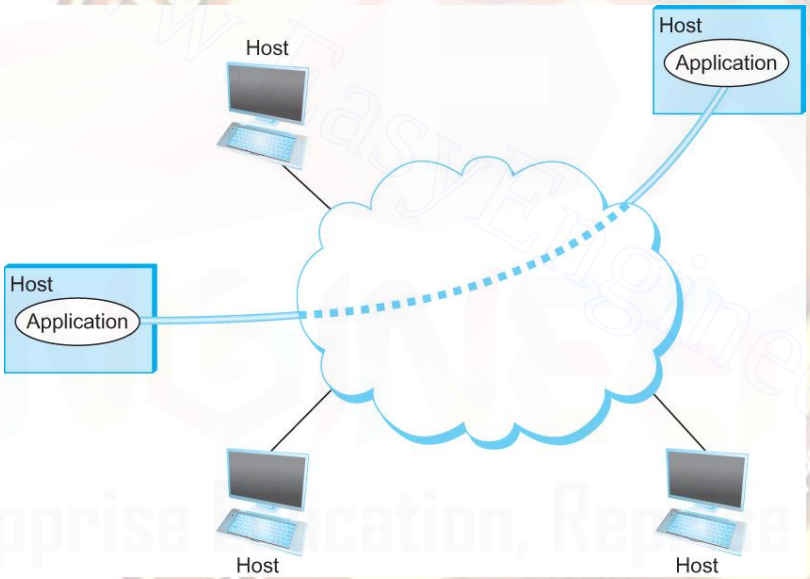
Cost-Effective Resource Sharing



Multiplexing multiple logical flows over a single physical link

Support for Common Services

- **Logical Channels**
  - **Application-to-Application communication path or a pipe**



Process communicating over an abstract channel

Common Communication Patterns

- **Client/Server**
- **Two types of communication channel**
  - **Request/Reply Channels**
  - **Message Stream Channels**

Reliability

- **Network should hide the errors**
- **Bits are lost**
  - **Bit errors (1 to a 0, and vice versa)**

- **Burst errors – several consecutive errors**
- **Packets are lost (Congestion)**
- **Links and Node failures**
- **Messages are delayed**
- **Messages are delivered out-of-order**
- **Third parties eavesdrop**

To get the reliable network, it is necessary to find how network fails.

Three classes of failures

- Bit error
- Packet loss
- Physical link and node failure

Addressing

The final requirement is that each node must be able to say which of the other node it wants to communicate with.

This is done by assigning address to each node. when a source node wants to deliver message to destination node, it specifies the address of destination node.

Switches and Routers use this address to decide how to forward the message. This process based on address is called **Routing**.

Unicast – sending message to single node.

Broadcast – Sending message to all the nodes on the network.

Multicast – Sending message to some subnet not to all.

Resource Sharing

Pblm: How do several hosts share the same link when they all want to use it at the same time?

Sol: Multiplexing – System resources are shared among multiple users

Methods: **1.** Synchronous Time Division Multiplexing(STDM)

Divide time into equal sized quanta

2.Frequency Division Multiplexing(FDM)

Transmit each flow at different frequency

3. Statistical Multiplexing

First two methods are limited in two ways

- If one flow does not have data to send then its time quantum remains idle, even the other flow has data to transmit.
- No of flows are fixed and known ahead of time, it cannot be resized.

Statistical methods combine the ideas of both STDM and FDM

- Data from each flow is transmitted on demand so no idle quantum
- It defines upper bound on size of data and it is referred as packet.

Common communication patterns

Communication between a pair of processes is done by request / reply basis. The process which sends request is referred as client and the one which honors the request is referred as server.

This can be done using channels. Two types of channels are

- Request / Reply channels
- Message stream Channels

1.1NETWORK ARCHITECTURE

Networks do not remain fixed at single point in time, but it must evolve to accommodate changes based on the technologies on which they are based and demands made by application programmer.

Network architecture guides the design and implementation of network. Two commonly used architecture are

- OSI Architecture
- Internet or TCP/IP architecture

Layering and Protocols

When the system gets complex, the system designer introduces another level of abstraction. It defines unifying model with important aspects of the system, encapsulated this model in interface objects and hide it from users

In network, abstraction leads to layering. Layering provides two nice features.

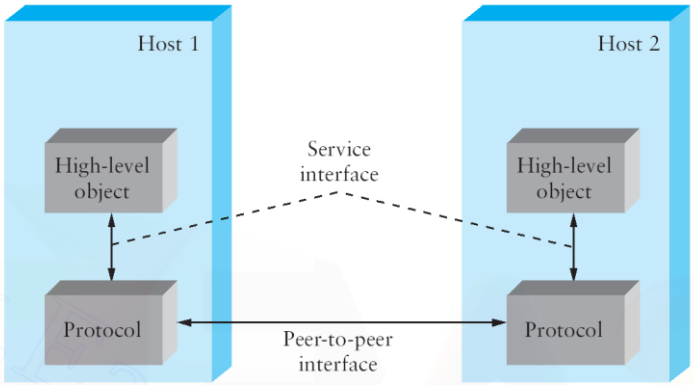
- It decomposes the problem of building a network into more manageable components. Rather than implementing a monolithic piece of software that does everything implement several layers, each of which solves one part of the problem.
- It provides more modular design. To add some new service, it is enough to modify the functionality at one layer, reusing the functions provided at all the other layers.

Protocols

A protocol is a set of rules that governs data communication. It defines what is communicated, how it is communicated, and when it is communicated. The key elements of a protocol are syntax, semantics and timing.

Each protocol defines two different interfaces.

- **Service interface** - to the other objects on the same computer that want to use its communication services. This service interface defines the operations that local objects can perform on the protocol.
- **Peer interface** - to its counterpart (peer) on another machine. It also defines the form and meaning of messages exchanged between protocol peers to implement the communication service.



Except at the hardware level, peer to peer communication is indirect.

We can represent these protocols as protocol graph. Nodes of the graph correspond to protocols, and the edges represent a depends-on relation.

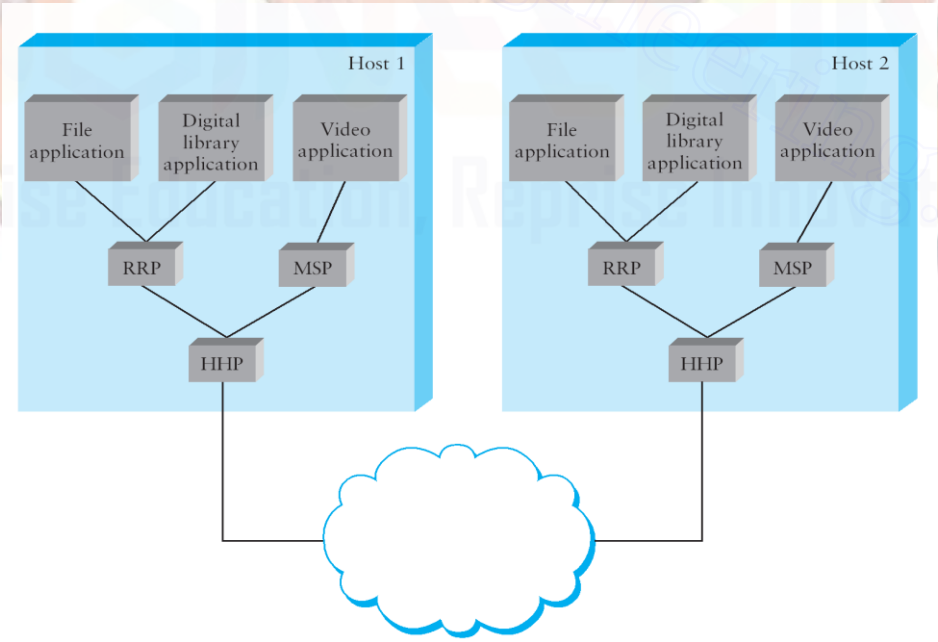


Fig: Example of a protocol graph.

For example, the file access program on host 1 wants to send a message to its peer on host 2 using the communication service offered by protocol RRP. In this case, the file

application asks RRP to send the message on its behalf. To communicate with its peer, RRP then invokes the services of HHP, which in turn transmits the message to its peer on the other machine. Once the message has arrived at protocol HHP on host 2, HHP passes the message up to RRP, which in turn delivers the message to the file application. In this particular case, the application is said to employ the services of the protocol stack RRP/HHP.

Encapsulation

Control information must be added with the data to instruct the peer how to handle with the received message. It will be added into the header or trailer.

Header - Small data structure from few bytes to few kilobytes attached to the front of message.

Trailer – Information will be added at the end of the message

Payload or message body – Data send by the program

In this case data is encapsulated with new message created by protocol at each level.

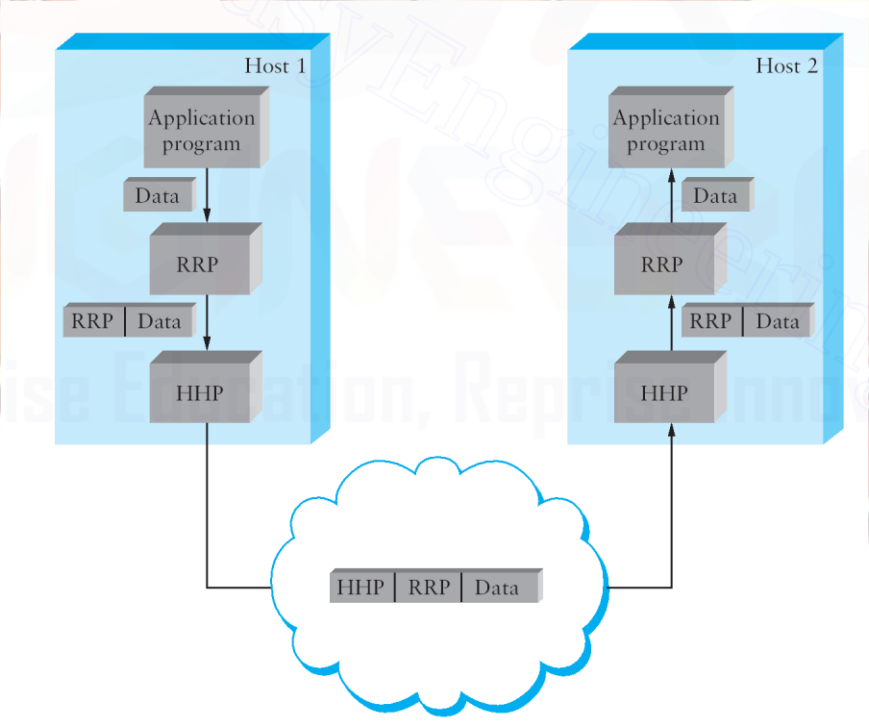


Fig: High-level messages are encapsulated inside of low-level messages.

In this example HHP encapsulates RRP's message by attaching a header of its own. Then HHP sends the message to its peer over some network, and then when the message arrives at the destination host, it is processed in the opposite order.

Multiplexing and De-Multiplexing

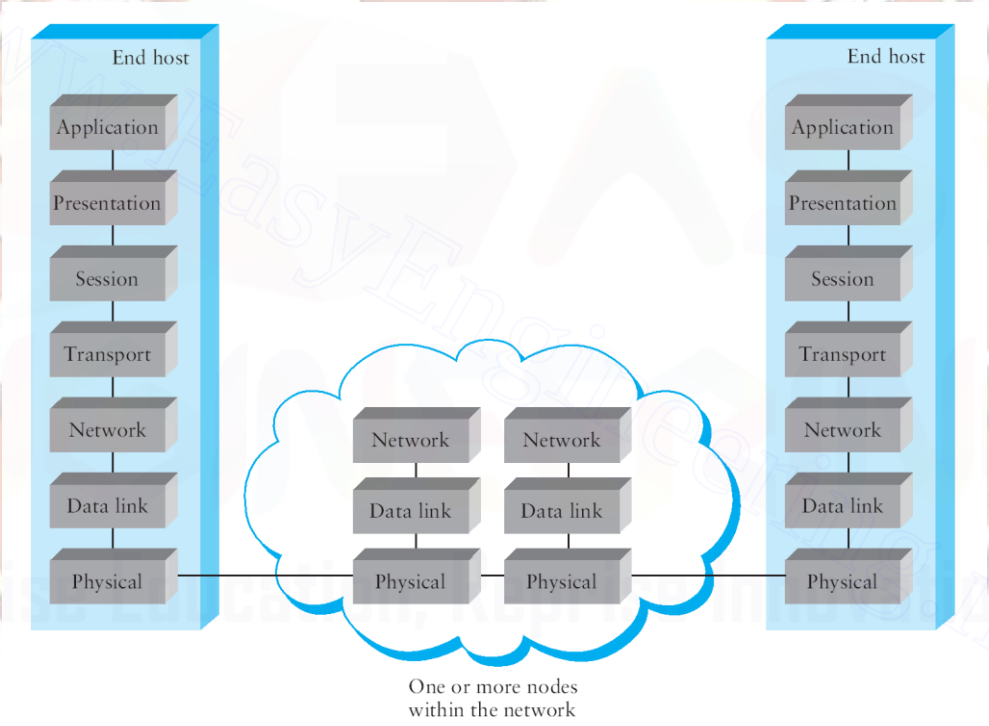
The fundamental idea of packet switching is to multiplex multiple flows of data over a single physical link. This can be achieved by adding identifier to the header message. It is known as **demultiplexing or demux key.** It gives the address to which it has to communicate.

The messages are demultiplexed at the destination side. In some cases same key is used on both sides and in some cases different keys are used.

1.2.OSI ARCHITECTURE

ISO defines a common way to connect computer by the architecture called Open System Interconnection (OSI) architecture.

Network functionality is divided into seven layers.



Organization of the layers

The 7 layers can be grouped into 3 subgroups

1. **Network Support Layers**

   Layers 1,2,3 - Physical, Data link and Network are the network support layers. They deal with the physical aspects of moving data from one device to another such as electrical specifications, physical addressing, transport timing and reliability.

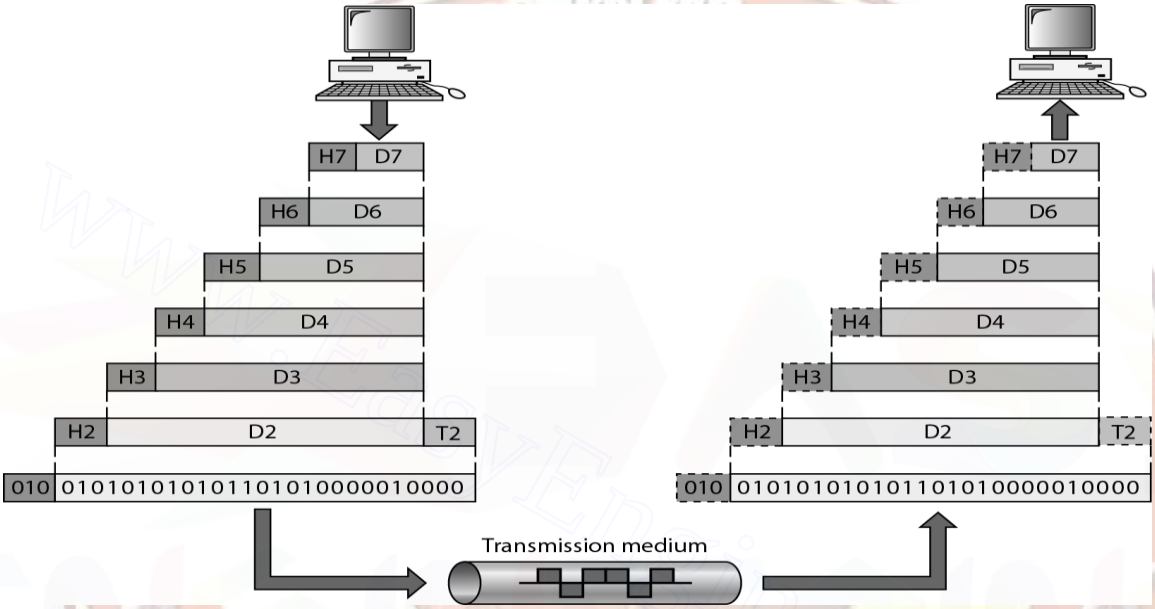
2. Transport Layer

Layer4, transport layer, ensures end-to-end reliable data transmission on a single link.

3. User Support Layers

Layers 5,6,7 – Session, presentation and application are the user support layers. They allow interoperability among unrelated software systems

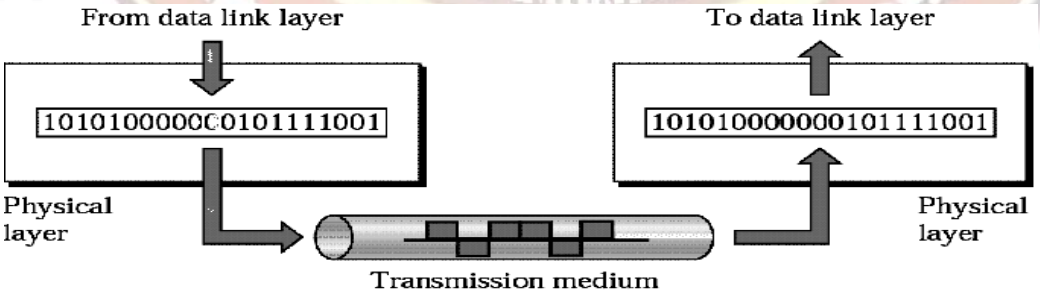
Data exchange using the OSI model



Functions of the Layers

1. Physical Layer

The physical layer coordinates the functions required to transmit a bit stream over a physical medium.



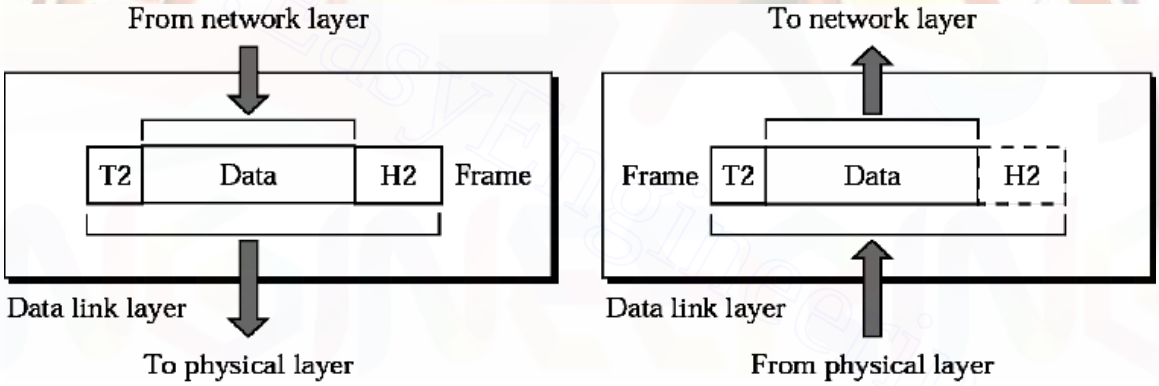
The physical layer is concerned with the following:

- **Physical characteristics of interfaces and media** - The physical layer defines the characteristics of the interface between the devices and the transmission medium.
- **Representation of bits** - To transmit the stream of bits, it must be encoded to signals. The physical layer defines the type of encoding.

- **Data Rate or Transmission rate -** The number of bits sent each second – is also defined by the physical layer.
- **Synchronization of bits -** The sender and receiver must be synchronized at the bit level. Their clocks must be synchronized.
- **Line Configuration -** In a point-to-point configuration, two devices are connected together through a dedicated link. In a multipoint configuration, a link is shared between several devices.
- **Physical Topology -** The physical topology defines how devices are connected to make a network. Devices can be connected using a mesh, bus, star or ring topology.
- **Transmission Mode** - The physical layer also defines the direction of transmission between two devices: simplex, half-duplex or full-duplex.

2. Data Link Layer

It is responsible for transmitting frames from one node to next node.

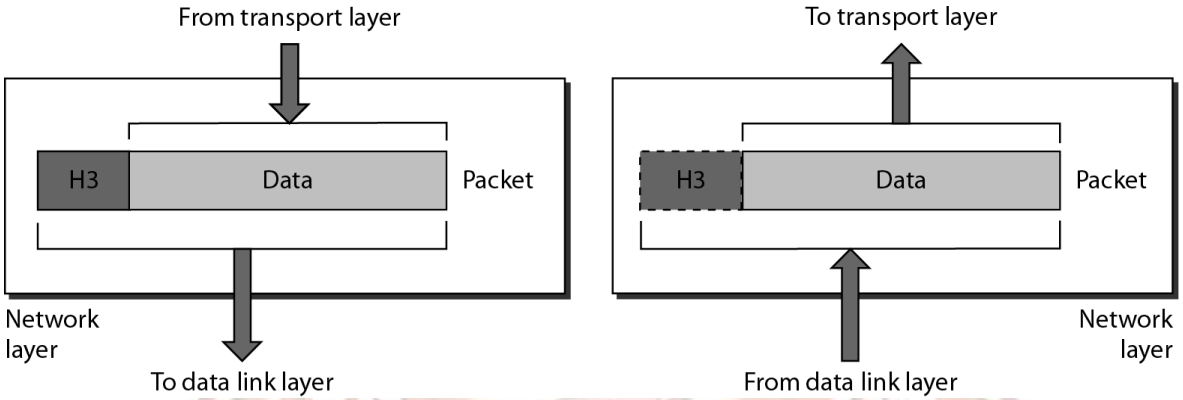


The other responsibilities of this layer are

- **Framing -** Divides the stream of bits received into data units called frames.
- **Physical addressing** – If frames are to be distributed to different systems on the n/w , data link layer adds a header to the frame to define the sender and receiver.
- **Flow control**- If the rate at which the data are absorbed by the receiver is less than the rate produced in the sender ,the Data link layer imposes a flow ctrl mechanism.
- **Error control**- Used for detecting and retransmitting damaged or lost frames and to prevent duplication of frames. This is achieved through a trailer added at the end of the frame.
- **Access control** -Used to determine which device has control over the link at any given time.

3. NETWORK LAYER

This layer is responsible for the delivery of packets from source to destination.



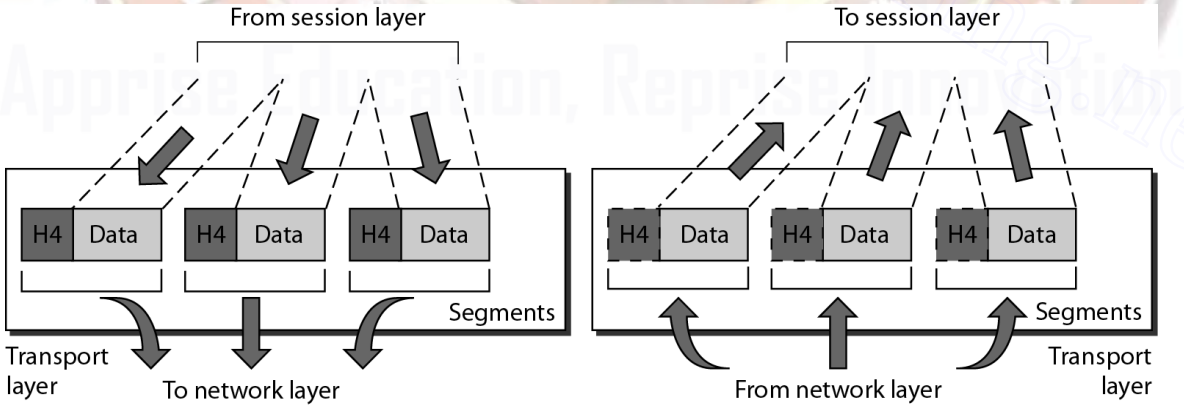
It is mainly required, when it is necessary to send information from one network to another.

The other responsibilities of this layer are

- **Logical addressing** - If a packet passes the n/w boundary, we need another addressing system for source and destination called logical address.
- **Routing** – The devices which connects various networks called routers are responsible for delivering packets to final destination.

4. TRANSPORT LAYER

- It is responsible for **Process to Process** delivery.
- It also ensures whether the message arrives in order or not.



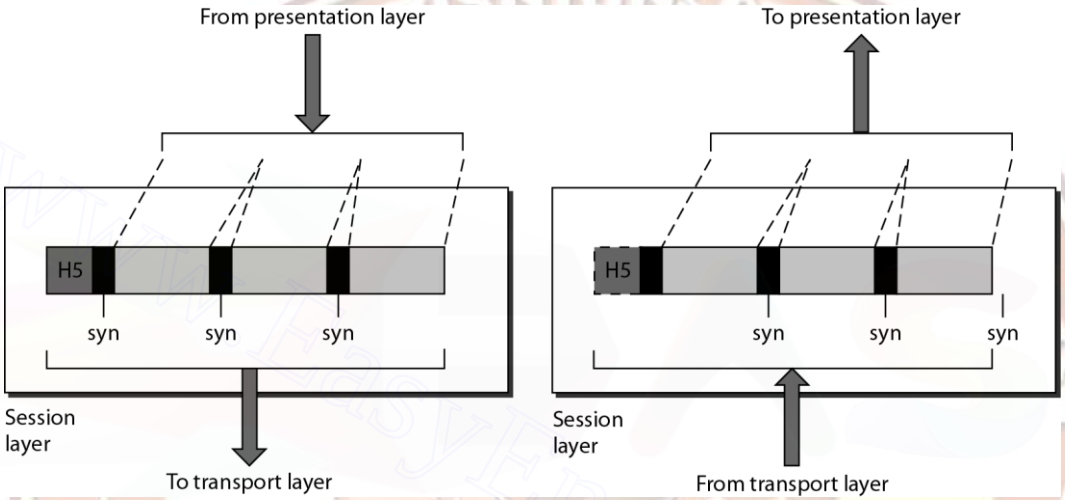
The other responsibilities of this layer are

- **Port addressing** - The header in this must therefore include a address called port address. This layer gets the entire message to the correct process on that computer.
- **Segmentation and reassembly** - The message is divided into segments and each segment is assigned a sequence number. These numbers are arranged correctly on the arrival side by this layer.

- **Connection control** - This can either be **connectionless or connection-oriented.** The connectionless treats each segment as a individual packet and delivers to the destination. The connection-oriented makes connection on the destination side before the delivery. After the delivery the termination will be terminated.
- **Flow and error control** - Similar to data link layer, but process to process take place.

5. SESSION LAYER

This layer establishes, manages and terminates connections between applications.

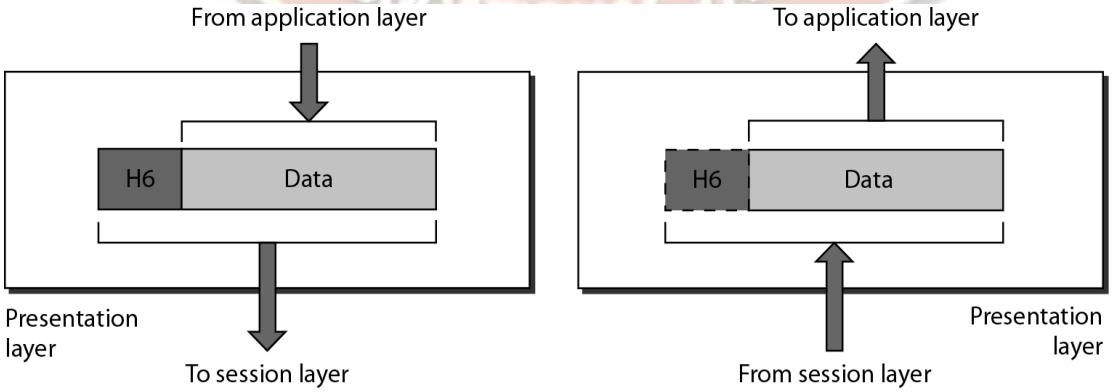


The other responsibilities of this layer are

- **Dialog control** - This session allows two systems to enter into a dialog either in half duplex or full duplex.
- **Synchronization**-This allows to add checkpoints into a stream of data.

6.PRESENTATION LAYER

It is concerned with the syntax and semantics of information exchanged between two systems.

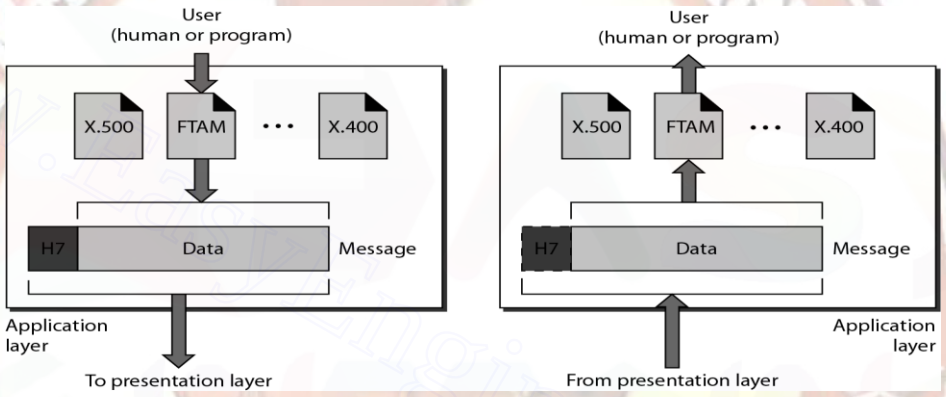


The other responsibilities of this layer are

- **Translation** – Different computers use different encoding system, this layer is responsible for interoperability between these different encoding methods. It will change the message into some common format.
- **Encryption and decryption**-It means that sender transforms the original information to another form and sends the resulting message over the n/w. and vice versa.
- **Compression and expansion**-Compression reduces the number of bits contained in the information particularly in text, audio and video.

7. APPLICATION LAYER

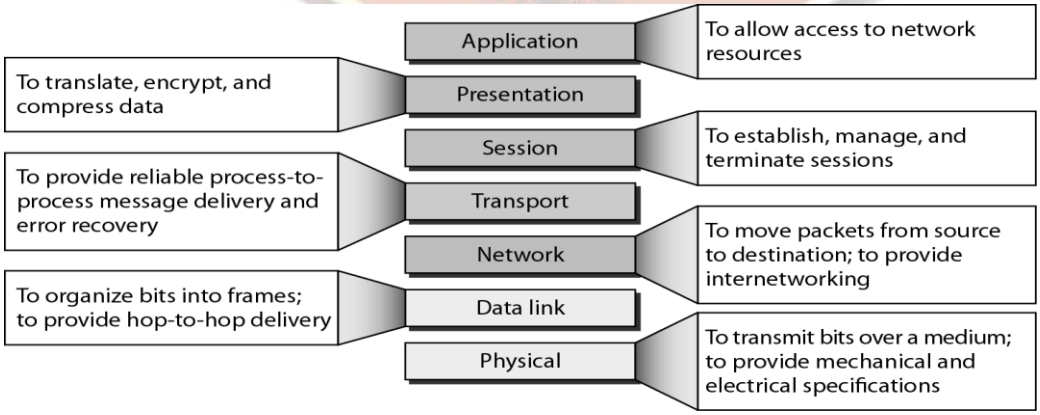
This layer enables the user to access the n/w. This allows the user to log on to remote user.



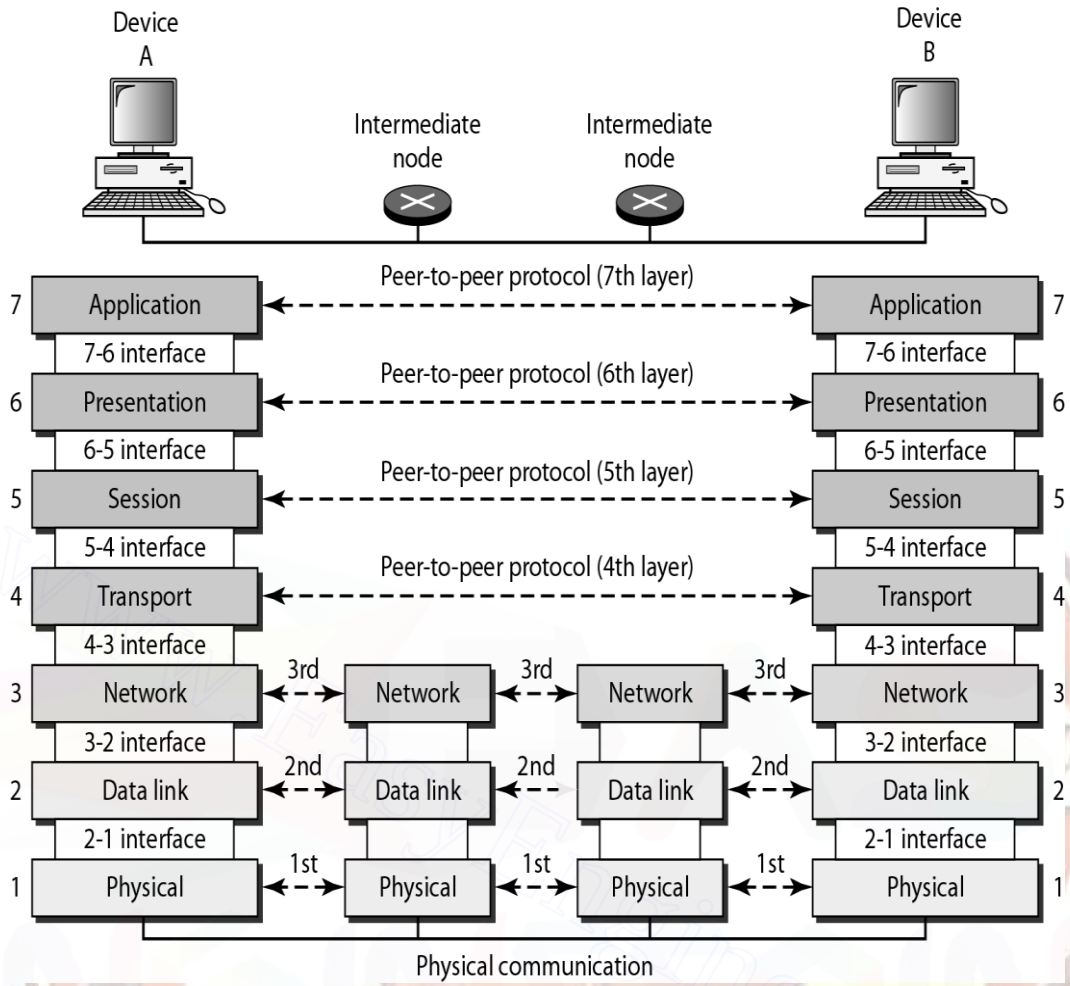
The other responsibilities of this layer are

- **FTAM(file transfer,access,management)** - Allows user to access files in a remote host.
- **Mail services** - Provides email forwarding and storage.
- **Directory services** - Provides database sources to access information about various sources and objects.

Summary of layers

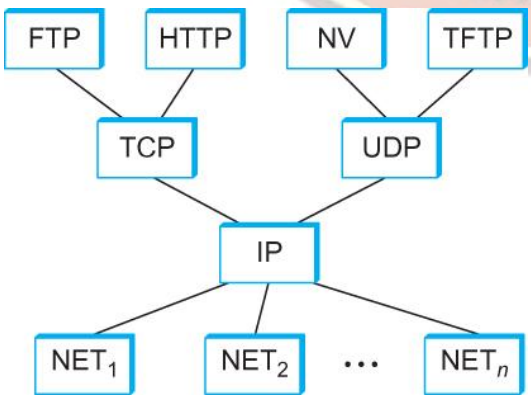


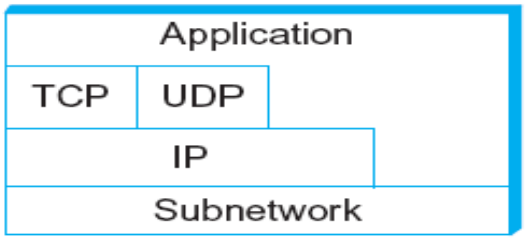
The interaction between layers in the OSI model



Internet Architecture

Internet Architecture is referred as TCP/IP model. Earlier packet switched network is called ARPANET. Internet Architecture is evolved based on that. Consider the Internet protocol graph. At the lowest level, network protocols are denoted by NET_1, NET_2, etc.,. These protocols are implemented by a combination of hardware (Network Adapter and software (network device driver).





Alternative view of the Internet architecture. The "Network" layer shown here is sometimes referred to as the "sub-network" or "link" layer.

Network Interface Layer

The *Network Interface layer* (also called the Network Access layer) is responsible for placing TCP/IP packets on the network medium and receiving TCP/IP packets off the network medium. TCP/IP was designed to be independent of the network access method, frame format, and medium. In this way, TCP/IP can be used to connect differing network types. These include LAN technologies such as Ethernet and Token Ring and WAN technologies such as X.25 and Frame Relay.

Internet Layer

The *Internet layer* is responsible for addressing, packaging, and routing functions. The core protocols of the Internet layer are IP, ARP, ICMP, and IGMP.

- The *Internet Protocol* (IP) is a routable protocol responsible for IP addressing, routing, and the fragmentation and reassembly of packets.
- The *Address Resolution Protocol* (ARP) is responsible for the resolution of the Internet layer address to the Network Interface layer address such as a hardware address.
- The *Internet Control Message Protocol* (ICMP) is responsible for providing diagnostic functions and reporting errors due to the unsuccessful delivery of IP packets.
- The *Internet Group Management Protocol* (IGMP) is responsible for the management of IP multicast gro

Transport Layer

The *Transport layer* (also known as the Host-to-Host Transport layer) is responsible for providing the Application layer with session and datagram communication services. The core protocols of the Transport layer are *Transmission Control Protocol* (TCP) and the *User Datagram Protocol* (UDP).

- TCP provides a one-to-one, connection-oriented, reliable communications service. TCP is responsible for the establishment of a TCP connection, the sequencing and acknowledgment of packets sent, and the recovery of packets lost during transmission.
- UDP provides a one-to-one or one-to-many, connectionless, unreliable communications service. UDP is used when the amount of data to be transferred is small (such as the data that would fit into a single packet), when the overhead of establishing a TCP connection is not desired or when the applications or upper layer protocols provide reliable delivery.

The Transport layer encompasses the responsibilities of the OSI Transport layer and some of the responsibilities of the OSI Session layer.

Application Layer

The *Application layer* provides applications the ability to access the services of the other layers and defines the protocols that applications use to exchange data. There are many Application layer protocols and new protocols are always being developed.

The most widely-known Application layer protocols are those used for the exchange of user information:

- The Hypertext Transfer Protocol (HTTP) is used to transfer files that make up the Web pages of the World Wide Web.
- The File Transfer Protocol (FTP) is used for interactive file transfer.
- The Simple Mail Transfer Protocol (SMTP) is used for the transfer of mail messages and attachments.
- Telnet, a terminal emulation protocol, is used for logging on remotely to network hosts.

- Three main features
  - Does not imply strict layering. The application is free to bypass the defined transport layers and to directly use IP or other underlying networks
  - An hour-glass shape – wide at the top, narrow in the middle and wide at the bottom. IP serves as the focal point for the architecture
  - In order for a new protocol to be officially included in the architecture, there needs to be both a protocol specification and at least one (and preferably two) representative implementations of the specification

Network software

- Application Programming Interface
- Interface exported by the network
- Since most network protocols are implemented (those in the high protocol stack) in software and nearly all computer systems implement their network protocols as part of the operating system, when we refer to the interface "*exported by the network*", we are generally referring to the interface that the OS provides to its networking subsystem
- The interface is called the network Application Programming Interface (API)

Socket

Socket Interface was originally provided by the Berkeley distribution of Unix

- Now supported in virtually all operating systems

- Each protocol provides a certain set of *services*, and the API provides a syntax by which those services can be invoked in this particular OS
- What is a socket?
- The point where a local application process attaches to the network
- An interface between an application and the network
- An application creates the socket

The interface defines operations for

- Creating a socket
- Attaching a socket to the network
- Sending and receiving messages through the socket
- Closing the socket

Socket Family

- PF_INET denotes the Internet family
- PF_UNIX denotes the Unix pipe facility
- PF_PACKET denotes direct access to the network interface (i.e., it bypasses the TCP/IP protocol stack)
- Socket Type
- SOCK_STREAM is used to denote a byte stream
- SOCK_DGRAM is an alternative that denotes a message oriented service, such as that provided by UDP

Creating a Socket

int sockfd = socket(address_family, type, protocol);

- The socket number returned is the socket descriptor for the newly created socket
- int sockfd = socket (PF_INET, SOCK_STREAM, 0);
- int sockfd = socket (PF_INET, SOCK_DGRAM, 0);

  The combination of PF_INET and SOCK_STREAM implies TCP

Client-Server Model with TCP

Server

- Passive open
- Prepares to accept connection, does not actually establish a connection

Server invokes

int bind (int socket, struct sockaddr *address, int addr_len)

int listen (int socket, int backlog)

int accept (int socket, struct sockaddr *address, int *addr_len)

Bind

- Binds the newly created socket to the specified address i.e. the network address of the local participant (the server)
- Address is a data structure which combines IP and port

Listen

- Defines how many connections can be pending on the specified socket

Accept

- Carries out the passive open
- Blocking operation
  - Does not return until a remote participant has established a connection
  - When it does, it returns a new socket that corresponds to the new established connection and the address argument contains the remote participant's address

Client

- Application performs active open
- It says who it wants to communicate with

Client invokes

int connect (int socket, struct sockaddr *address, int addr_len)

Connect

- Does not return until TCP has successfully established a connection at which application is free to begin sending data
- Address contains remote machine's address

- Once a connection is established, the application process invokes two operation

int send (int socket, char *msg, int msg_len, int flags)

int recv (int socket, char *buff, int buff_len, int flags)

Example Application: Client

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define SERVER_PORT 5432
#define MAX_LINE 256
int main(int argc, char * argv[])
{
        FILE *fp;
        struct hostent *hp;
        struct sockaddr_in sin;
        char *host;
        char buf[MAX_LINE];
        int s;
        int len;
        if (argc==2) {
                host = argv[1];
        }
        else {
                fprintf(stderr, "usage: simplex-talk host\n");
        exit(1);
        }
/* translate host name into peer's IP address */
        hp = gethostbyname(host);
```

```
        if (!hp) {
                fprintf(stderr, "simplex-talk: unknown host: %s\n", host);
                exit(1);
        }
        /* build address data structure */
        bzero((char *)&sin, sizeof(sin));
        sin.sin_family = AF_INET;
        bcopy(hp->h_addr, (char *)&sin.sin_addr, hp->h_length);
        sin.sin_port = htons(SERVER_PORT);
        /* active open */
        if ((s = socket(PF_INET, SOCK_STREAM, 0)) < 0) {
                perror("simplex-talk: socket");
                exit(1);
        }
        if (connect(s, (struct sockaddr *)&sin, sizeof(sin)) < 0) {
                perror("simplex-talk: connect");
                close(s);
                exit(1);
        }
        /* main loop: get and send lines of text */
        while (fgets(buf, sizeof(buf), stdin)) {
                buf[MAX_LINE-1] = '\0';
                len = strlen(buf) + 1;
                send(s, buf, len, 0);
        }
}
```

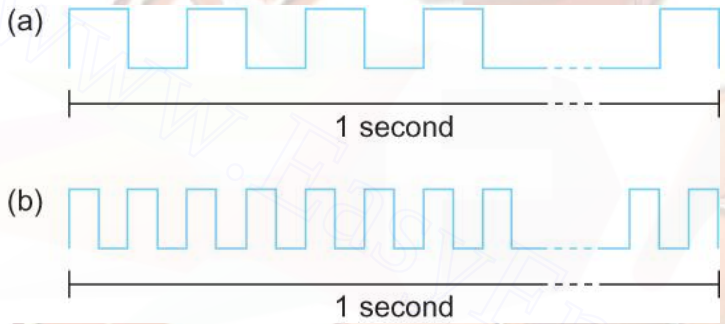
Example Application: Server

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define SERVER_PORT 5432
#define MAX_PENDING 5
```

```
#define MAX_LINE 256
int main()
{
        struct sockaddr_in sin;
        char buf[MAX_LINE];
        int len;
        int s, new_s;
        /* build address data structure */
        bzero((char *)&sin, sizeof(sin));
        sin.sin_family = AF_INET;
        sin.sin_addr.s_addr = INADDR_ANY;
        sin.sin_port = htons(SERVER_PORT);
        /* setup passive open */
        if ((s = socket(PF_INET, SOCK_STREAM, 0)) < 0) {
                perror("simplex-talk: socket");
                exit(1);
        }
if ((bind(s, (struct sockaddr *)&sin, sizeof(sin))) < 0) {
                perror("simplex-talk: bind");
                exit(1);
        }
        listen(s, MAX_PENDING);
        /* wait for connection, then receive and print text */
        while(1) {
                if ((new_s = accept(s, (struct sockaddr *)&sin, &len)) < 0) {
                perror("simplex-talk: accept");
                exit(1);
        }
        while (len = recv(new_s, buf, sizeof(buf), 0))
                fputs(buf, stdout);
                close(new_s);
        }
}
```

Network Performance

- Bandwidth
  - Width of the frequency band
  - Number of bits per second that can be transmitted over a communication link
- 1 Mbps: 1 x 10^6 bits/second = 1x2^{20} bits/sec
- 1 x 10^{-6} seconds to transmit each bit or imagine that a timeline, now each bit occupies 1 micro second space.
- On a 2 Mbps link the width is 0.5 micro second.
- Smaller the width more will be transmission per unit time.



Bits transmitted at a particular bandwidth can be regarded as having some width:
(a) bits transmitted at 1Mbps (each bit 1 μs wide);
(b) bits transmitted at 2Mbps (each bit 0.5 μs wide).

- Latency = Propagation + transmit + queue
- Propagation = distance/speed of light
- Transmit = size/bandwidth
- One bit transmission => propagation is important
- Large bytes transmission => bandwidth is important

Delay X Bandwidth

- We think the channel between a pair of processes as a hollow pipe
- Latency (delay) length of the pipe and bandwidth the width of the pipe
- Delay of 50 ms and bandwidth of 45 Mbps

⇒ 50 x 10^{-3} seconds x 45 x 10^6 bits/second

⇒ 2.25 x 10^6 bits = 280 KB data.

⇒ Relative importance of bandwidth and latency depends on application

⇒ For large file transfer, bandwidth is critical

⇒ For small messages (HTTP, NFS, etc.), latency is critical

⇒ Variance in latency (jitter) can also affect some applications (*e.g.*, audio/video conferencing)

Infinite bandwidth

⇒ RTT dominates

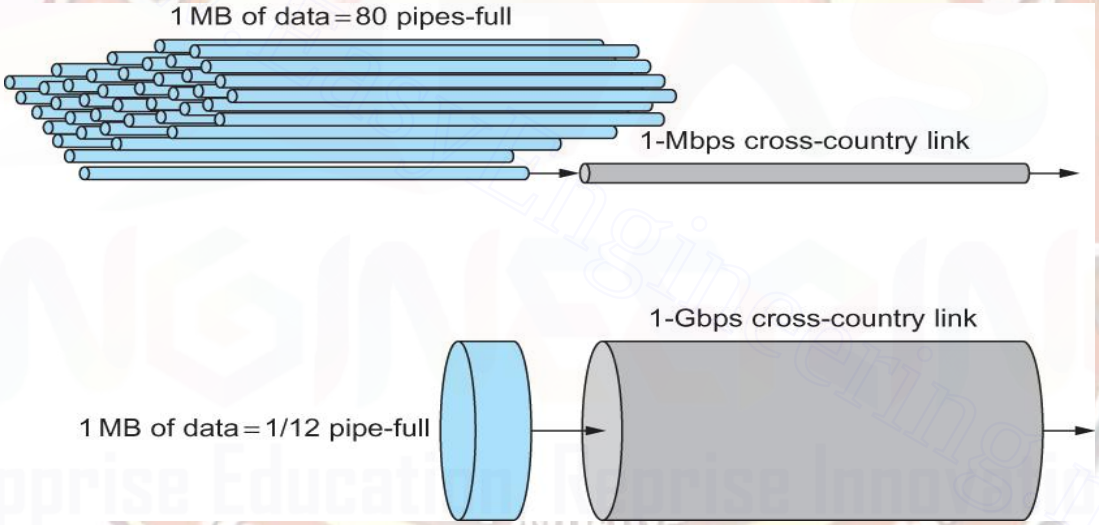
⇒ Throughput = TransferSize / TransferTime

⇒ TransferTime = RTT + 1/Bandwidth x TransferSize

⇒ Its all relative

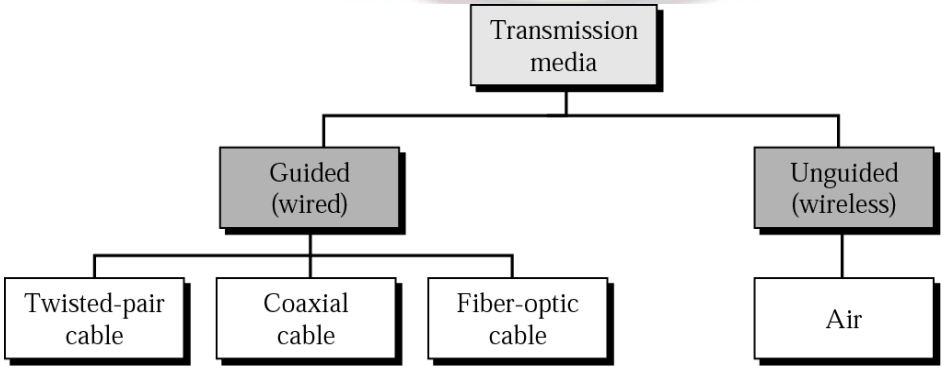
⇒ 1-MB file to 1-Gbps link looks like a 1-KB packet to 1-Mbps link

Relationship between bandwidth and latency



A 1-MB file would fill the 1-Mbps link 80 times, but only fill the 1-Gbps link 1/12 of one time

PHYSICAL LINKS



Guided Media

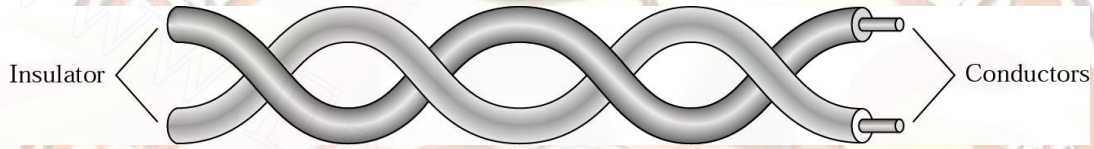
Guided media conduct signals from one device to another include Twisted-pair cable, Coaxial Cable and Fiber-optic cable. A signal traveling along any of these media is directed and contained by the physical limits of the medium.

Twisted-pair and coaxial cable use metallic (copper) conductors that accept and transport signals in the form of electric current. Optical fiber is a glass cable that accepts and transports signals in the form of light.

Twisted Pair Cable

A twisted pair consists of two conductors (normally copper) each with its own plastic insulation, twisted together.

- One of the wires is used to carry signals to the receiver
- Other is used as ground reference



Interference and cross talk may affect both the wires and create unwanted signals, if the two wires are parallel.

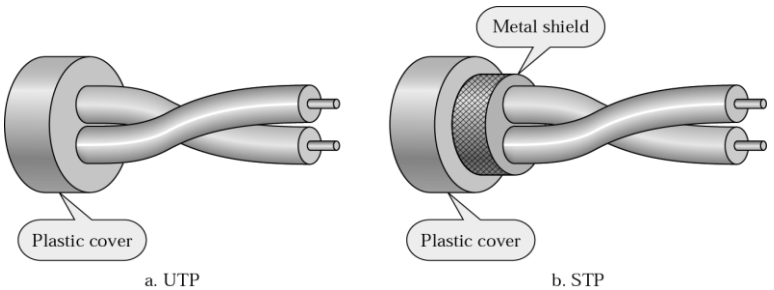
By twisting the pair, a balance is maintained. Suppose in one twist one wire is closer to noise and the other is farther in the next twist the reverse is true. Twisting makes it probable that both wires are equally affected by external influences.

Twisted Pair Cable comes into two forms:

- **Unshielded**
- **Shielded**

Unshielded versus shielded Twisted-Pair Cable

- Shielded Twisted-Pair (STP) Cable has a metal foil or braided-mesh covering that encases each pair of insulated conductors.
- Metal casing improves that quality of cable by preventing the penetration of noise or cross talk.
- It is more expensive. The following figure shows the difference between UTP and STP



Applications

- Twisted Pair cables are used in telephone lines to provide voice and data channels.
- Local area networks also use twisted pair cables.

Connectors

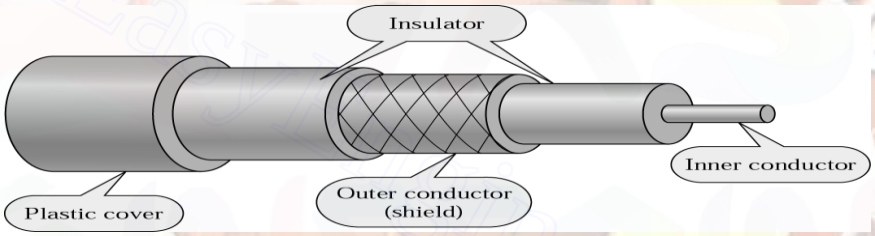
The most common UTP connector is RJ45.

Coaxial Cable

Coaxial cable (coax) carries signals of higher frequency ranges than twisted pair cable.

Instead of having two wires, coax has a central core conductor of solid or stranded wire (usually copper) enclosed in an insulating sheath, and with outer conductor of metal foil.

The outer metallic wrapping serves both as a shield against noise and as the second conductor and the whole cable is protected by a plastic cover.



Categories of coaxial cables

Category	Impedance	Use
RG-59	75	Cable TV
RG-58	50	Thin Ethernet
RG-11	50	Thick Ethernet

Applications

- It is used in analog and digital telephone networks
- It is also used in Cable TV networks
- It is used in Ethernet LAN

Connectors

- BNC connector – to connect the end of the cable to a device
- BNC T - to branch out network connection to computer
- BNC terminator - at the end of the cable to prevent the reflection of the signal.

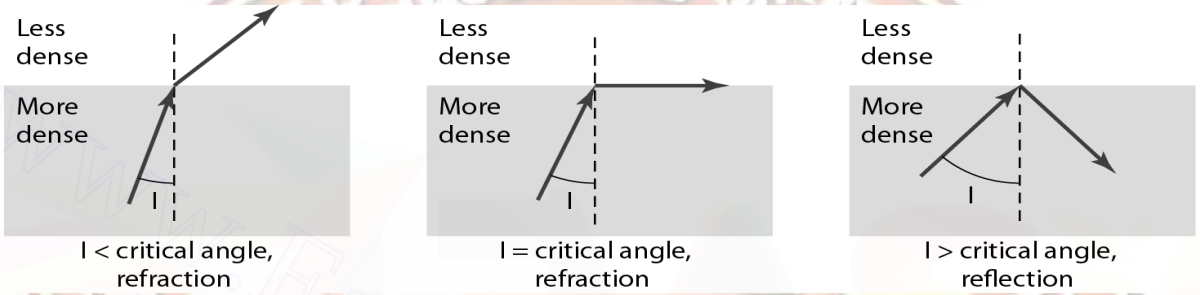
Fiber Optic Cable

A fiber-optic cable is made of glass or plastic and transmits signals in the form of light.

Properties of light

- Light travels in a straight line as long as it moves through a single uniform substance. If traveling through one substance suddenly enters another, ray changes its direction.

Bending of light ray

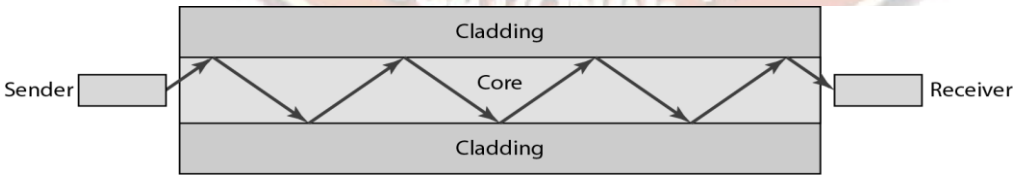


If the angle of incidence(the angle the ray makes with the line perpendicular to the interface between the two medium) is less than the critical angle the ray refracts and move closer to the surface.

If the angle of incidence is equal to the critical angle, the light bends along the interface.

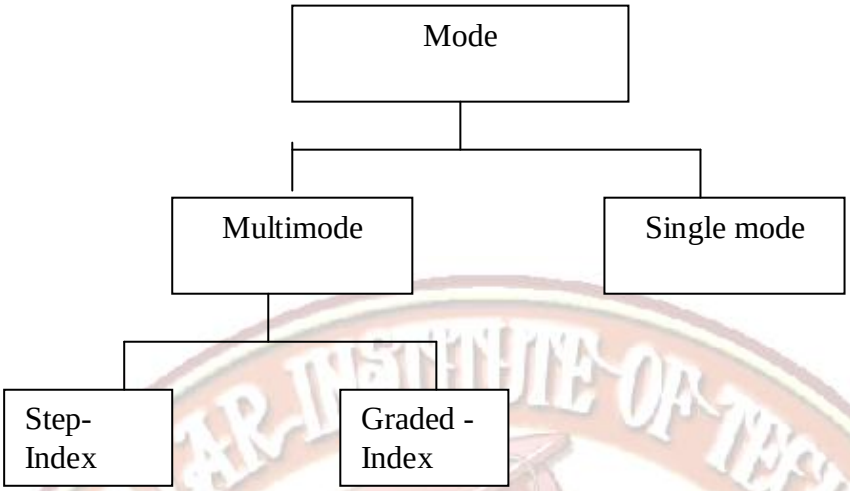
If the angle of incidence is greater than the critical angle, the ray reflects and travels again in the denser substance. Critical angle differs from one medium to another medium.

Optical fiber use reflection to guide light through a channel.



A Glass or plastic core is surrounded by a cladding of less dense glass or plastic.

Propagation Modes

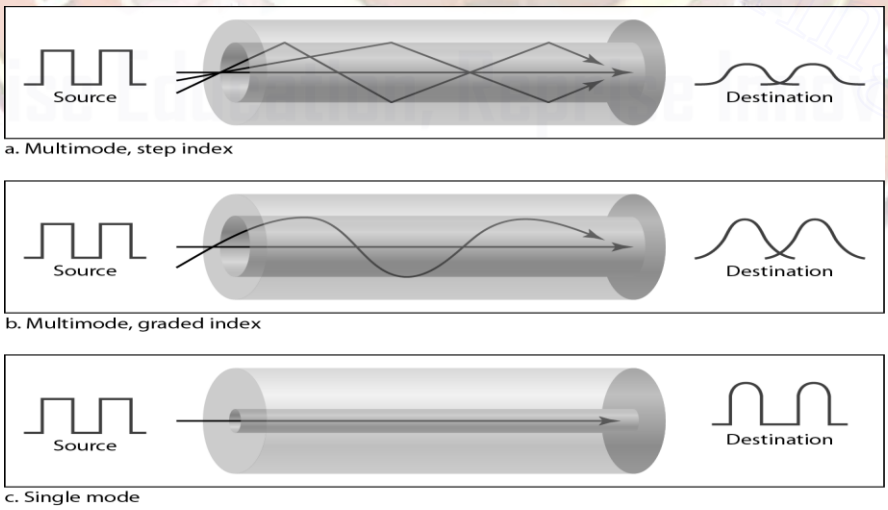


Multimode

In the multiple mode, multiple light beams from a source move through the core in different paths.

- **Multimode-Step-Index fiber:** The density of core remains constant from the centre to the edge.

  A ray of light moves through this constant density in a straight line until it reaches the interface of the core and the cladding. At the interface there is an abrupt change to a lower density that changes the angle of the beam's motion.



- **Multimode- Graded -Index fiber:** The density is varying. Density is highest at the centre of the core and decreases gradually to its lowest at the edge.

Single Mode

Single mode uses step-index fiber and a highly focused source of light that limits beams to a small range of angles, all close to the horizontal.

The single mode fiber itself is manufactured with a much smaller diameter than that of multimedia fiber.

Connectors

- **Subscriber channel** (SC) **connector** is used for cable TV.
- **Straight-tip** (ST) **connector** is used for connecting cable to networking devices.

Advantages of Optical Fiber

- Noise resistance
- Less signal attenuation
- Light weight

Disadvantages

- Cost
- Installation and maintenance
- Unidirectional
- Fragility (easily broken)

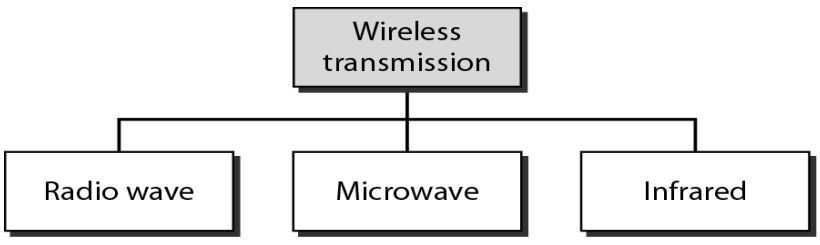
Unguided media

Unguided media transport electromagnetic waves without using a physical conductor. This type of communication is often referred to as wireless communication.

Signals are normally broadcast through air and thus available to anyone who has device capable of receiving them.

Unguided signals can travel from the source to destination in several ways:

- **Ground propagation** – waves travel through lowest portion on atmosphere.
- **Sky propagation** – High frequency waves radiate upward into ionosphere and reflected back to earth.



Line-of-sight propagation – Very high frequency signals travel in a straight line

Radio Waves

Electromagnetic waves ranging in frequencies between 3 kHz and 1 GHz are normally called radio waves.

Properties

- Radio waves are omnidirectional. When an antenna transmits radio waves, they are propagated in all directions. This means that the sending and receiving antennas do not have to be aligned.
- A sending antenna sends waves that can be received by any receiving antenna.
- Radio waves, particularly those of low and medium frequencies, can penetrate walls.

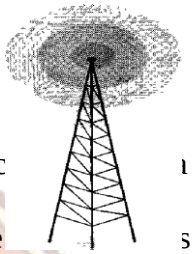


Fig: Omnidirec ı

Disadvantages

- The omnidirectional property has a disadvantage, that the s transmitted by one antenna are susceptible to interference by another antenna that may send signals using the same frequency or band.
- As Radio waves can penetrate through walls, we cannot isolate a communication to just inside or outside a building.

Applications

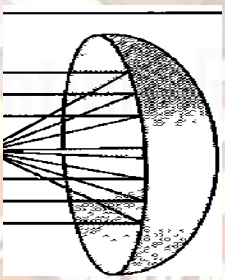
Radio waves are used for multicast communications, such as radio and television, and paging systems.

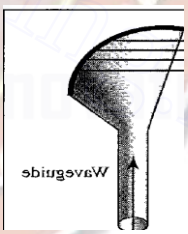
Microwaves

Electromagnetic waves having frequencies between 1 and 300 GHz are called microwaves.

Properties

- Microwaves are unidirectional.
- Sending and receiving antennas need to be aligned
- Microwave propagation is line-of-sight
- Very high-frequency microwaves cannot penetrate walls





a) ParabolicDish antenna b)Horn antenna

- Parabolic Dish antenna focus all incoming waves into single point
- Outgoing transmissions are broadcast through a horn aimed at the dish.

Disadvantage

- If receivers are inside buildings, they cannot receive these waves

Applications

- Microwaves are used for unicast communication such as cellular telephones, satellite networks, and wireless LANs.

Infrared

- Electromagnetic waves with frequencies from 300 GHz to 400 THz are called infrared rays
- Infrared waves, having high frequencies, cannot penetrate walls.

Applications

- Infrared signals can be used for short-range communication in a closed area using line-of-sight propagation.

CHANNEL ACCESS ON LINKS

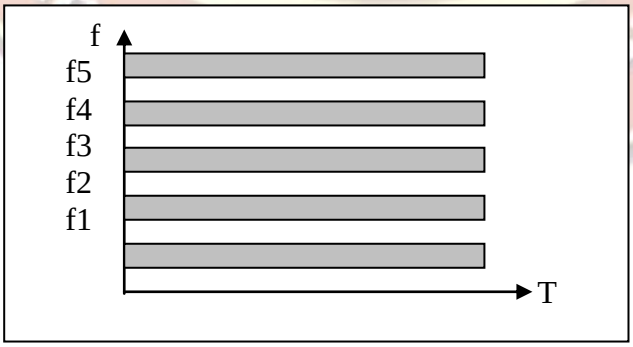
Multiple Access Techniques

Various multiple access techniques are

- Frequency Division Multiple Access(FDMA)
- Time Division Multiple Access (TDMA)
- Code Division Multiple Access(CDMA)

Frequency Division Multiple Access

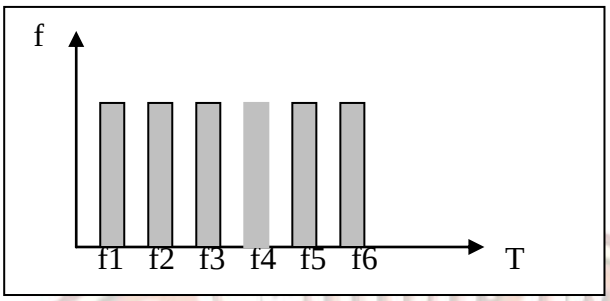
- In frequency-division multiple access (FDMA), the available bandwidth is divided into frequency bands.
- Each station is allocated a band to send its data.
- In this method when any one frequency level is kept idle and another is used frequently leads to inefficiency.



Time Division Multiple Access

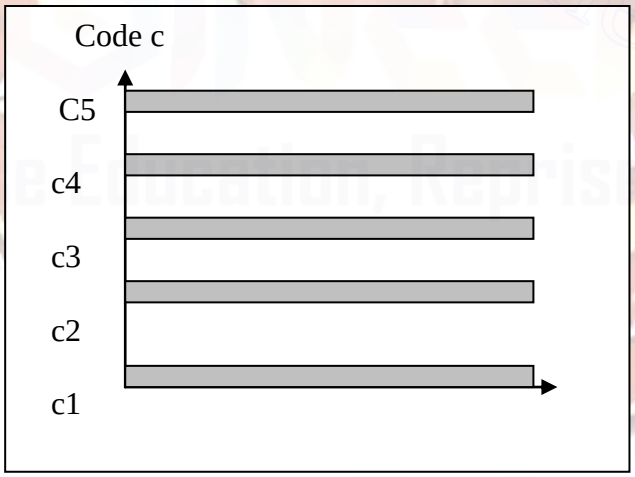
- In time-division multiple access (TDMA), the stations share the bandwidth of the channel in time.
- Each station is allocated a time slot during which it can send data.

- The main problem with TDMA lies in achieving synchronization between the different stations.
- Each station needs to know the beginning of its slot and the location of its slot.



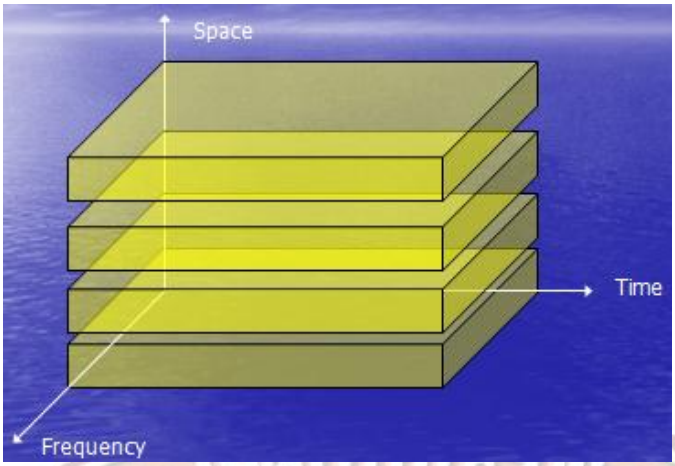
Code Division Multiple Access

- CDMA differs from FDMA because only one channel occupies the entire bandwidth of the link.
- It differs from TDMA because all stations can send data at the same time without timesharing.
- CDMA simply means communication with different codes.
- CDMA is based on coding theory. Each station is assigned a code, which is a sequence of numbers called chips.
- Chips will be added with the original data and it can be transmitted through same medium.



Space Division Multiple Access

- In SDMA, users are separated in a spatial way.
- Generally an adaptive array antenna technique is adopted.
- One disadvantage is the difficulty of separating two users who are placed near the base station.



ISSUES IN THE DATA LINK LAYER

a) **Services Provided to the Network Layer**
b) **Framing**
c) **Error Control**
d) **Flow Control**

a.Services Provided to the Network Layer

1. Provide service interface to the network layer
2. Dealing with transmission errors
3. Regulating data flow
   a. Slow receivers not swamped by fast senders

b.Framing

To transmit frames over the node it is necessary to mention start and end of each frame. There are three techniques to solve this frame

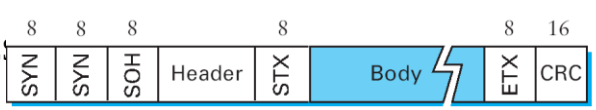
- Byte-Oriented Protocols (BISYNC, PPP, DDCMP)
- Bit-Oriented Protocols (HDLC)
- Clock-Based Framing (SONET)

A) Byte Oriented protocols

In this, view each frame as a collection of bytes (characters) rather than a collection of bits. Such a byte-oriented approach is exemplified by the **BISYNC** (Binary Synchronous Communication) protocol and the **DDCMP** (Digital Data Communication Message Protocol)

Sentinel Approach

The BISYNC protocol illustrates the sentinel approach to framing; its frame format is Fig: BISYNC Frame format



- The beginning of a frame is denoted by sending a special SYN (synchronization) character.
- The data portion of the frame is then contained between special sentinel characters: STX (start of text) and ETX (end of text).
- The SOH (start of header) field serves much the same purpose as the STX field.
- The frame format also includes a field labeled CRC (cyclic redundancy check) that is used to detect transmission errors.

The problem with the sentinel approach is that the ETX character might appear in the data portion of the frame. BISYNC overcomes this problem by "escaping" the ETX character by preceding it with a DLE (data-link-escape) character whenever it appears in the body of a frame; the DLE character is also escaped (by preceding it with an extra DLE) in the frame body. This approach is called **character stuffing**.

Point-to-Point Protocol (PPP)

The more recent Point-to-Point Protocol (PPP). The format of PPP frame is

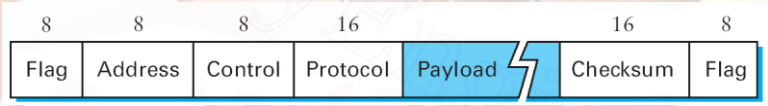
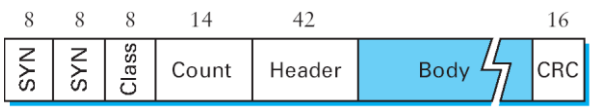


Fig: PPP Frame Format

- The Flag field has 01111110 as starting sequence.
- The Address and Control fields usually contain default values
- The Protocol field is used for demultiplexing.
- The frame payload size can he negotiated, but it is 1500 bytes by default.
- The PPP frame format is unusual in that several of the field sizes are negotiated rather than fixed.
- Negotiation is conducted by a protocol called LCP (Link Control Protocol).
- LCP sends control messages encapsulated in PPP frames—such messages are denoted by an LCP identifier in the PPP Protocol.

Byte-Counting Approach

The number of bytes contained in a frame can he included as a field in the frame header. DDCMP protocol is used for this approach. The frame format is



- COUNT Field specifies how many bytes are contained in the frame's body.
- Sometime count field will be corrupted during transmission, so the receiver will accumulate as many bytes as the COUNT field indicates. This is sometimes called a **framing error**.
- The receiver will then wait until it sees the next SYN character.

B)Bit-Oriented Protocols (HDLC)

In this, frames are viewed as collection of bits. High level data link protocol is used. The format is

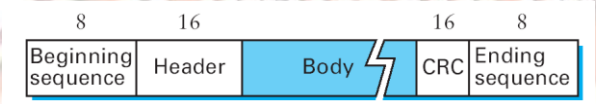


Fig: HDLC Frame Format

- HDLC denotes both the beginning and the end of a frame with the distinguished bit sequence 01111110.
- This sequence might appear anywhere in the body of the frame, it can be avoided by bit stuffing.
- On the sending side, any time five consecutive 1's have been transmitted from the body of the message (i.e., excluding when the sender is trying to transmit the distinguished 01111110 sequence), the sender inserts a 0 before transmitting the next bit.
- On the receiving side, five consecutive 1's arrived, the receiver makes its decision based on the next bit it sees (i.e., the bit following the five is).
- If the next bit is a 0, it must have been stuffed, and so the receiver removes it. If the next bit is a 1, then one of two things is true, either this is the end-of-frame marker or an error has been introduced into the bit stream.
- By looking at the next bit, the receiver can distinguish between these two cases:

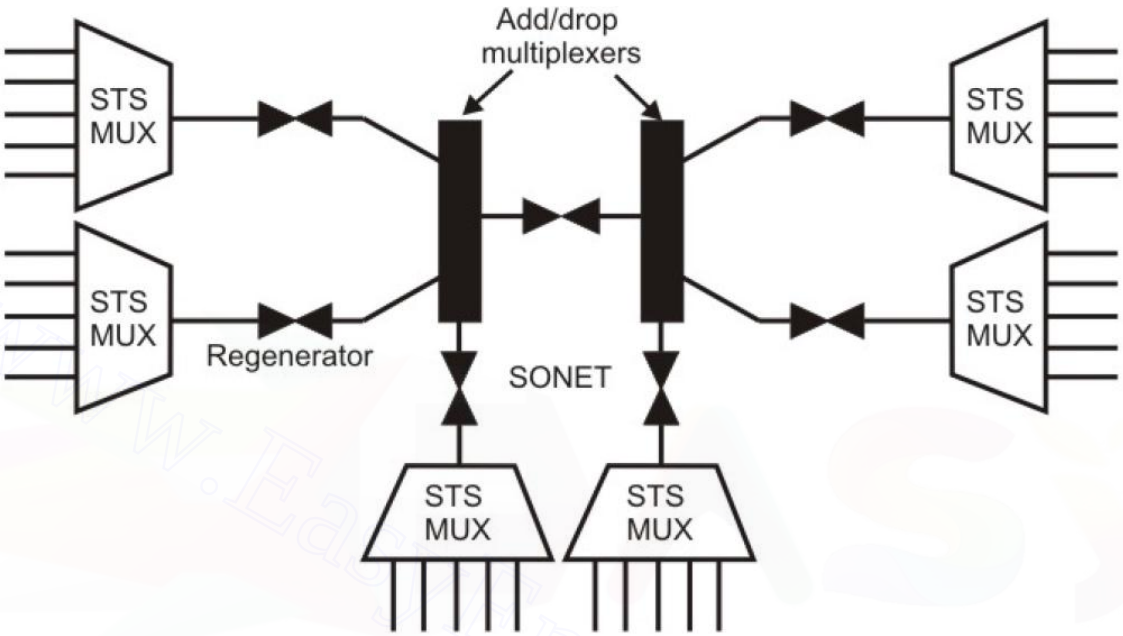
7 If it sees a 0 (i.e., the last eight bits it has looked at are 01111110), then it is the end-of- frame marker.

8 If it sees a 1 (i.e., the last eight bits it has looked at are 01111111), then there must have been an error and the whole frame is discarded.

C)Clock-Based Framing (SONET)

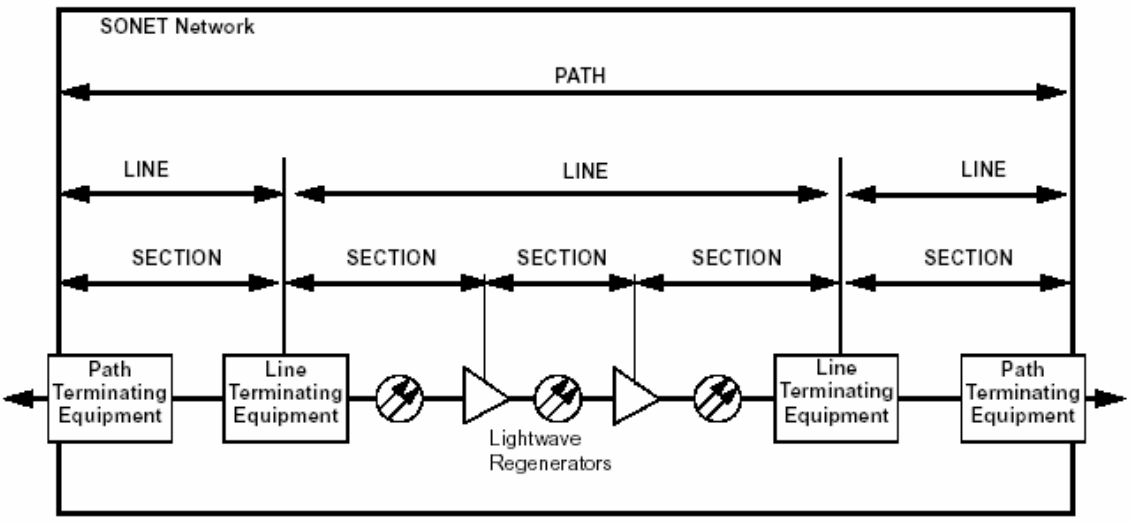
Physical Configuration and Network Elements
Three basic devices used in the SONET system are shown in Fig. 4.3.1. Functions of the three devices are mentioned below:

• Synchronous Transport Signal (STS) multiplexer/demultiplexer: It either multiplexes signal from multiple sources into a STS signal or demultiplexes an STS signal into different destination signals.
• Regenerator: It is a repeater that takes a received optical signal and regenerates it. It functions in the data link layer.
• Add/drop Multiplexer: Can add signals coming from different sources into a given path or remove a desired signal from a path and redirect it

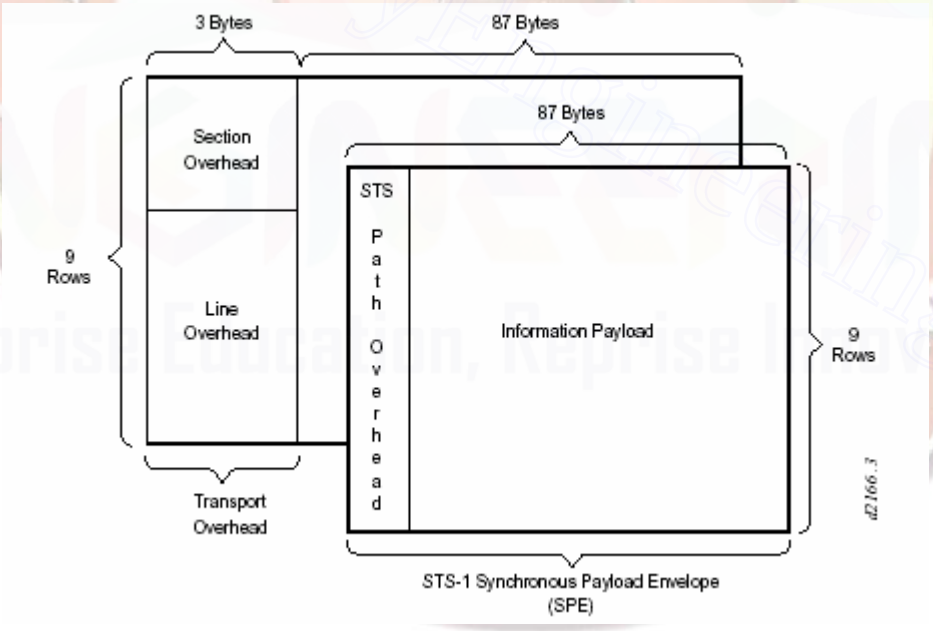


Section, Line and paths

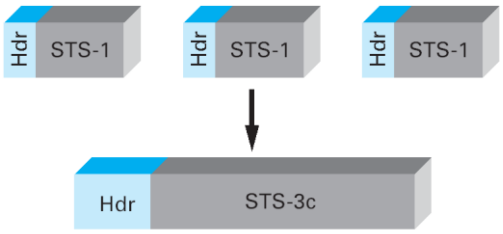
A number of electrical signals are fed into an STS multiplexer, where they are combined into a single optical signal. Regenerator recreates the optical signal without noise it has picked up in transit. Add/Drop multiplexer reorganize these signals. A section is an optical link, connecting two neighboring devices: multiplexer to multiplexer, multiplexer to regenerator, or regenerator to regenerator r. A line is a portion of network between two
multiplexers: STS to add/drop multiplexer, two add/drop multiplexer, or two STS multiplexer. A Path is the end-to-end portion of the network between two STS multiplexers,



- Synchronous Optical Network Standard is used for long distance transmission of data over optical network.
- It supports multiplexing of several low speed links into one high speed links.
- An STS-1 frame is used in this method.



- It is arranged as nine rows of 90 bytes each, and the first 3 bytes of each row are overhead, with the rest being available for data.
- The first 2 bytes of the frame contain a special bit pattern, and it is these bytes that enable the receiver to determine where the frame starts.
- The receiver looks for the special bit pattern consistently, once in every 810 bytes, since each frame is 9 x 90 = 810 bytes long.



- The STS-N frame can he thought of as consisting of N STS-1 frames, where the bytes from these frames are interleaved; that is, a byte from the first frame is transmitted, then a byte from the second frame is transmitted, and so on.
- Payload from these STS-1 frames can he linked together to form a larger STS-N payload, such a link is denoted STS-Nc. One of the bit in overhead is used for this purpose.

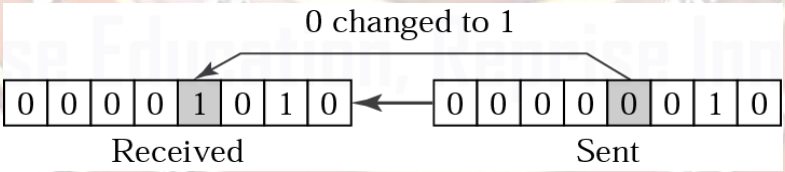
1.6.3. ERROR DETECTION AND CORRECTION

Data can be corrupted during transmission. For reliable communication, errors must be detected and corrected.

Types of Errors

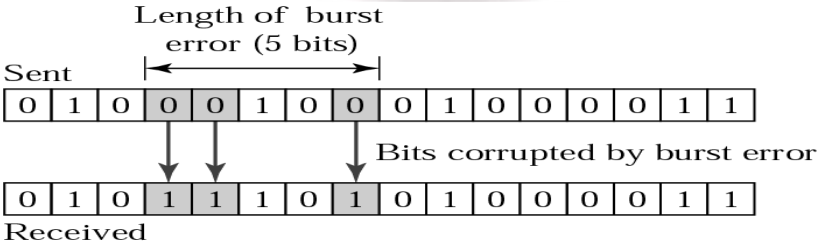
Single-bit error

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



Burst Error

The term Burst Error means that two or more bits in the data unit have changed from 1 to 0 or from 0 to 1.



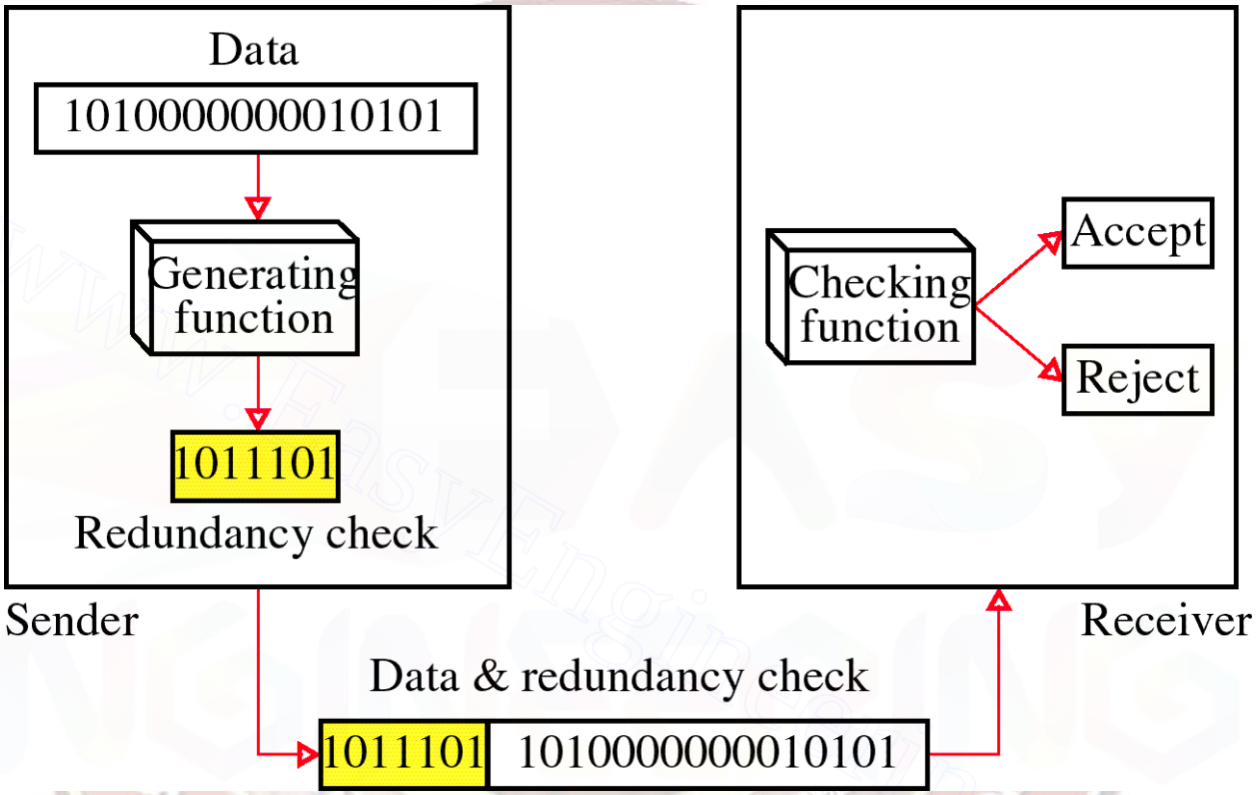
One method to detect error is to send every data twice, so that receiver checks every bit of two copies and detect error. This is not efficient.

Drawbacks

- Sends n-redundant bits for n-bit message.
- Many errors are undetected if both the copies are corrupted.

Redundancy

To overcome this drawback , Error detection **uses the concept of redundancy**. Here extra bits are added along with the original data for detecting errors at the destination.



Error Detection Methods

1.Vertical Redundancy Check(VRC)

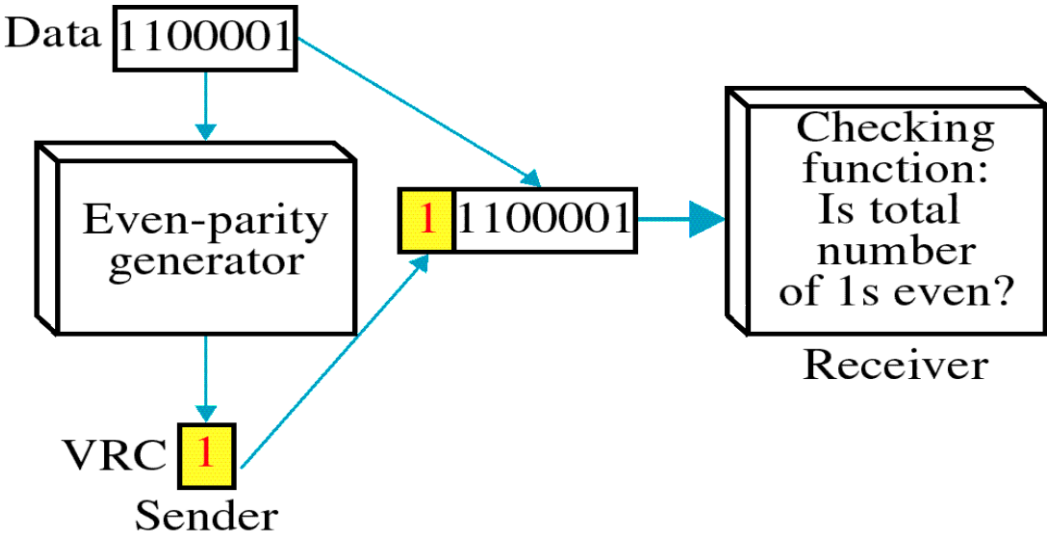
2.Longitudinal Redundancy Check(LRC)

3.Cyclic Redundancy Check(CRC)

4.Checksum

1.VRC (Simple parity check)

Only one redundant bit, called parity bit is added to every data unit so that the total number of 1's in unit become even (or odd)



Performance:

- ➔ It can detect single bit error
- ➔ It can detect burst errors only if the total number of errors is odd.

2. LRC (Two Dimensional Parity)

- ➢ It is based on simple parity.
- ➢ It performs calculation for each bit position across each byte in the frame.
- ➢ This adds extra parity byte for entire frame, in addition to a parity bit for each byte.

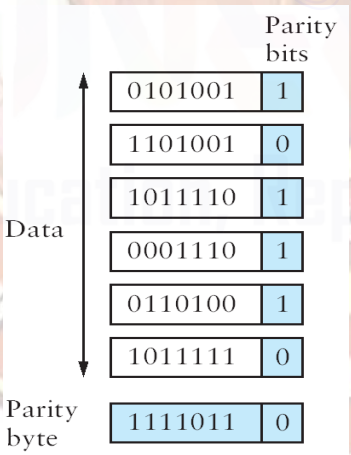


Fig: Two-dimensional parity

In this case, 14 bits of redundant information are added with original information.

Performance:

- ➔ LCR increases the likelihood of detecting burst errors.

➔ If two bits in one data units are damaged and two bits in exactly the same positions in another data unit are also damaged, the LRC checker will not detect an error.

➔

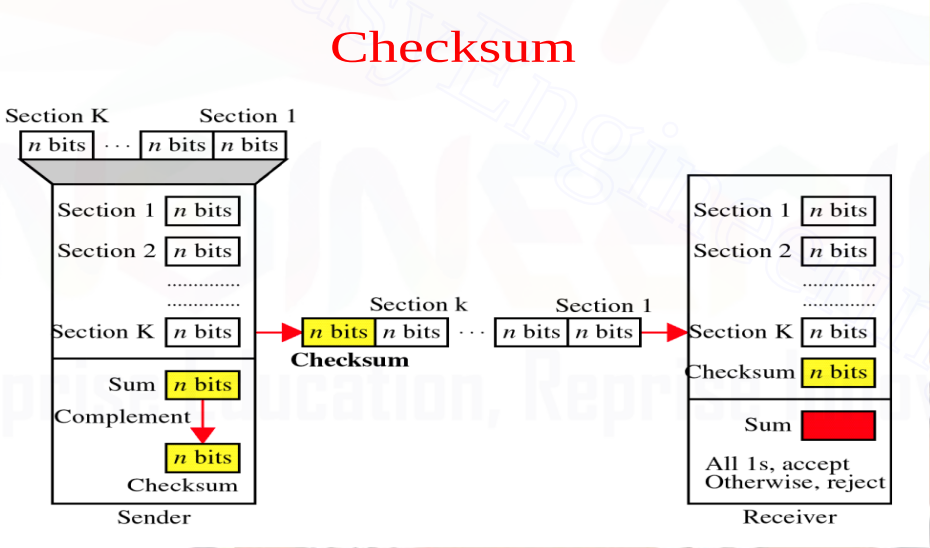
3. Check sum algorithm

Sender Side:

- The unit is divided into *k* sections, each of *n* bits.
- All sections are added together to get the sum.
- The sum is complemented and becomes the checksum.
- The checksum is sent with the data

Receiver Side:

- The unit is divided into *k* sections, each of *n* bits.
- All sections are added together to get the sum.
- The sum is complemented.
- If the result is zero, the data are accepted: otherwise, they are rejected.



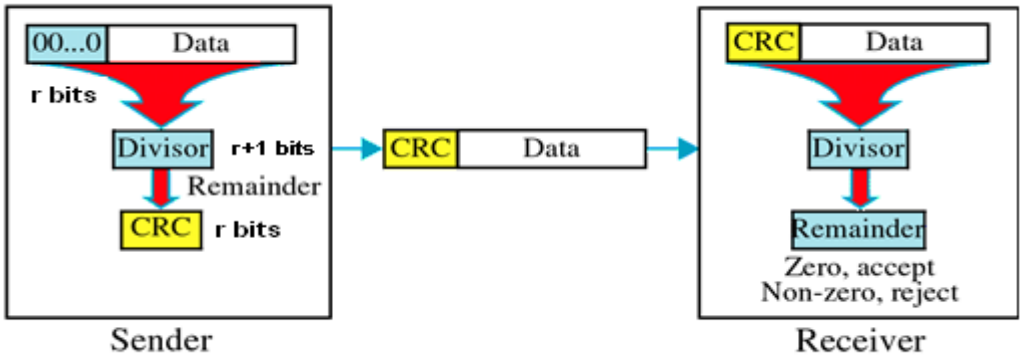
4. Cyclic Redundancy Check

- It uses small number of redundant bits to detect errors.
- Divisor is calculated by the polynomial functions under two conditions
  a. It should not be divisible by x
  b. It should be divisible by x+1

  Consider the original message as M(x) – n+1 bits

  Divisor C(x) – K bits

  Original sent message = M(x) + k-1 bits

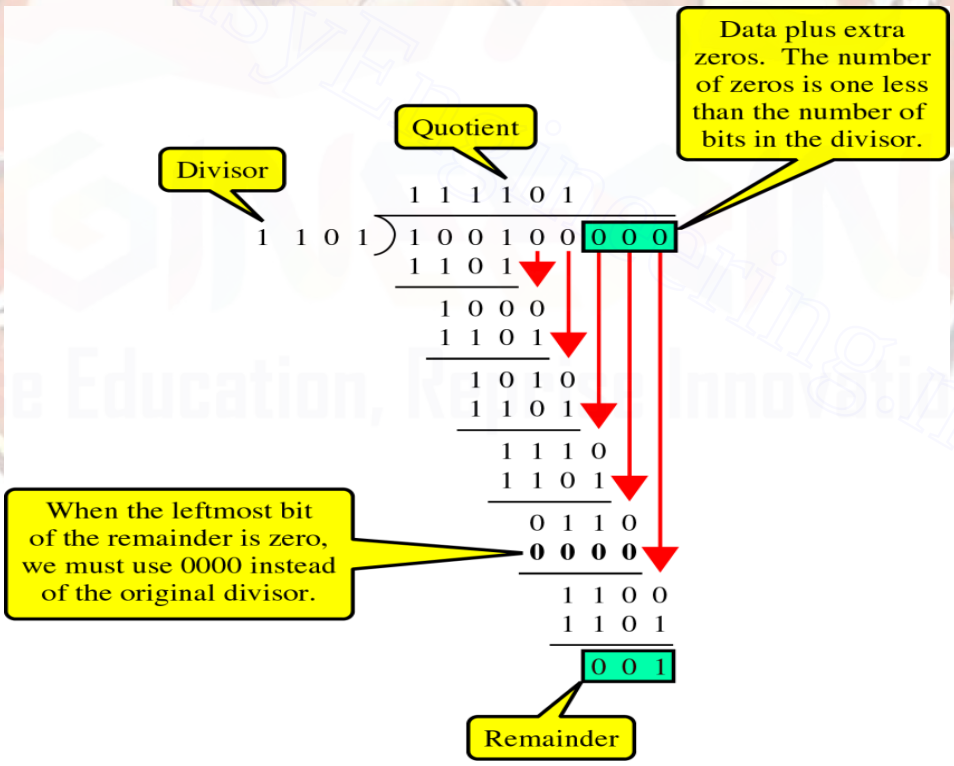


Steps

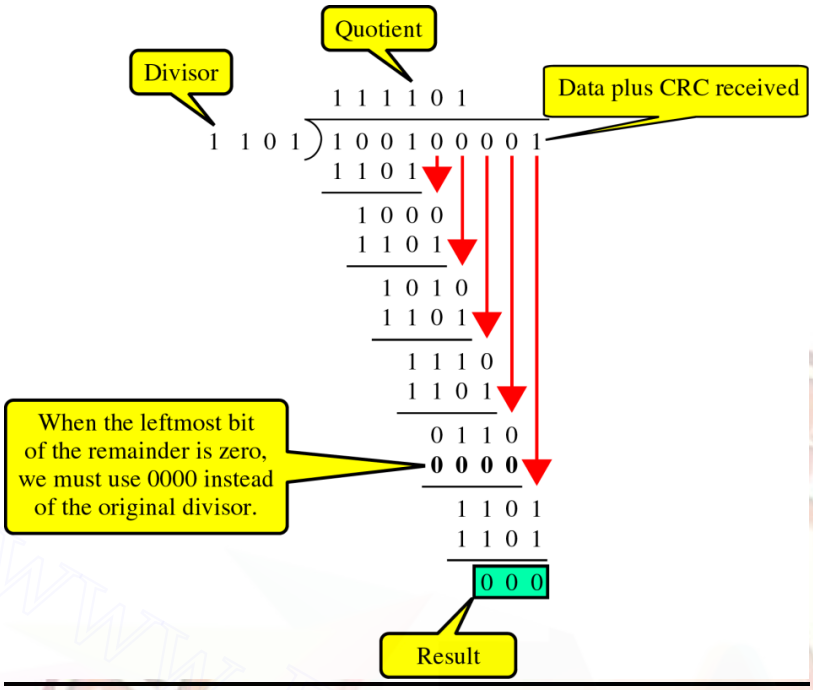
- Append k-1 zeros with M(x) – P(x)
- Divide P(x) by C(x)
- Subtract the remainder from T(x)
- Subtraction is made by making XOR operation

Eg: 100100 by 1101

Sender Side:



Receiver Side:



Error Correction

Error Correction can be handled in two ways

1. When an error is discovered, the receiver can have the sender to retransmit the entire data unit.
2. A receiver can use an error correcting code, which automatically correct certain errors.

Error correcting codes are more sophisticated than error-detection codes and require more redundancy bits.

In single bit error detection only two states are sufficient.

1) error

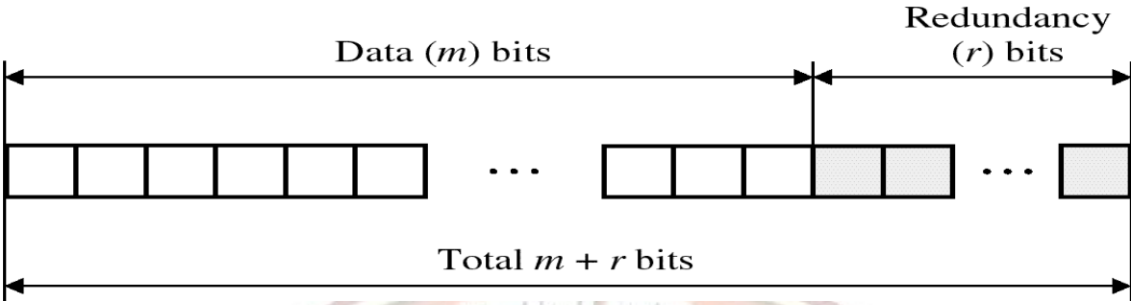
2) no error

Two states are not enough to detect an error but not to correct it.

Redundancy Bits

- To calculate the number of redundancy bit(r) required to correct a given number of data bits (m), we must find a relationship between m and r.
- Add m bits of data with r bits. The length of the resulting code is m+r.

Data and Redundancy bits



- If the total number of bits are m+r, then r must be able to indicate at least m+r+1 different states. r bits can indicate 2^r different states. Therefore, 2^r must be equal to or greater than m+r+1

$$2^r >= m+r+1$$

- For example if the value of m is 7 the smallest r value that can satisfy this equation is 4.

Relationship between data and redundancy bits

Number of Data Bits (m)	Number of redundancy Bits(r)	Total bits (m+r)
1	2	3
2	3	5
3	3	6
4	3	7
5	4	9
6	4	10
7	4	11

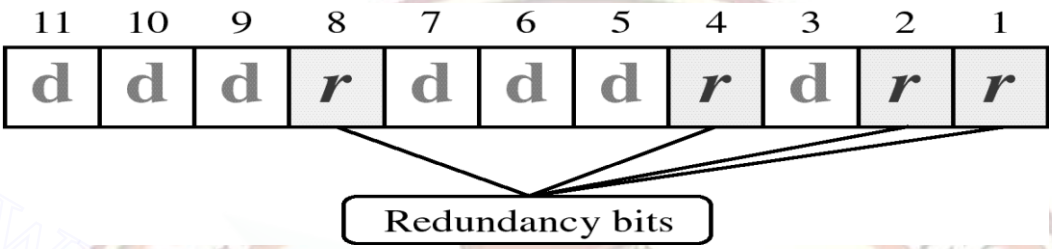
Hamming Code

R.W. Hamming provides a practical solution for the error correction.

Positioning the Redundancy Bits

For example, a seven-bit ASCII code requires four redundancy bits that can be added to the end of the data or intersperse with the original data bits. These redundancy bits are placed in positions 1, 2, 4 and 8. We refer these bits as r1, r2, r3 and r4

Position of redundancy bits in Hamming code



The combination used to calculate each of the four r values for a seven-bit data sequence are as follows

- The r1 bit is calculated using all bits positions whose binary representation include a 1 in the rightmost position
- r2 is calculated using all bit position with a 1 in the second position and so on

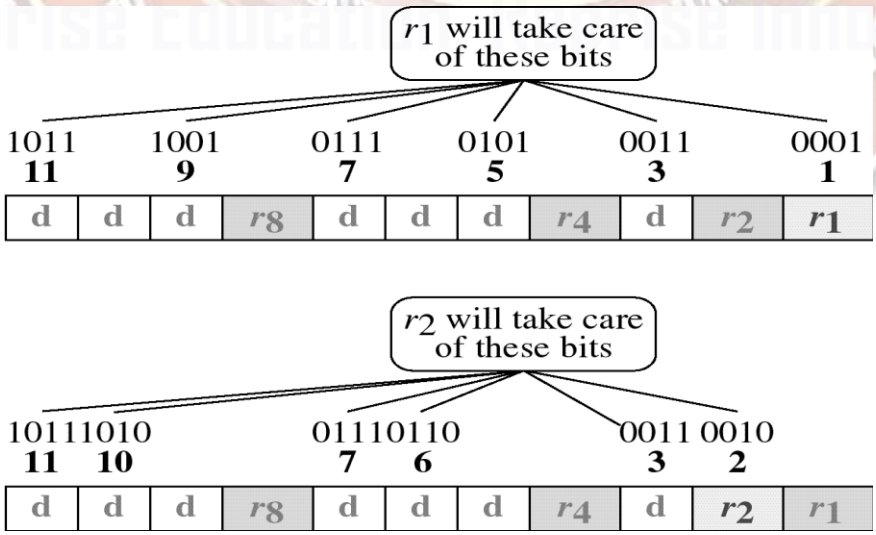
  r1: bits 1,3,5,7,9,11

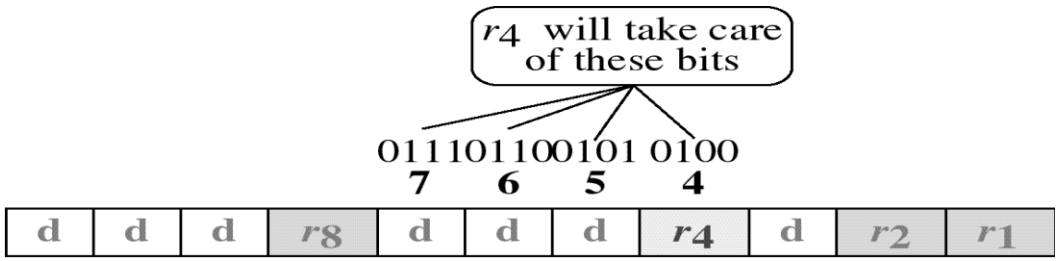
  r2: bits 2, 3, 6, 7, 10, 11

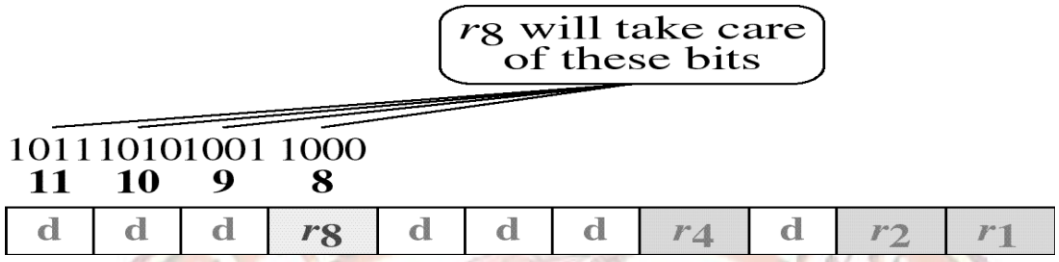
  r3: bits 4, 5, 6, 7

  r4: bits 8, 9, 10, 11

Redundancy bits calculation





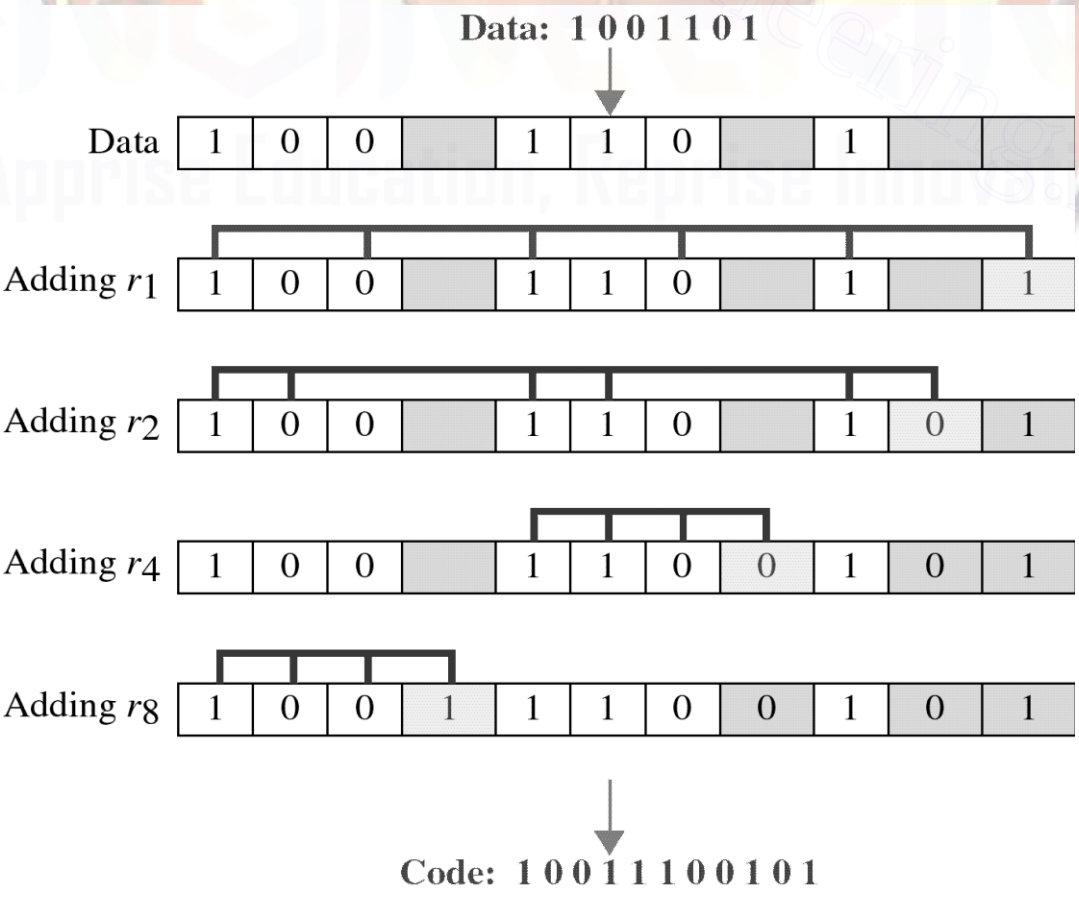


Calculating the r values

- Place each bit of the original character in its appropriate position in the 11-bit unit.
- Calculate the even parities for the various bit combination.
- The parity value for each combination is the value of the corresponding r bit.

For example,

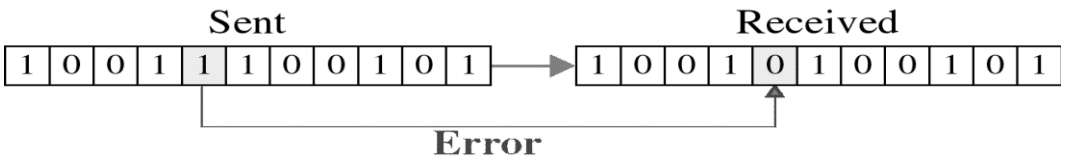
- The value of r1 is calculated to provide even parity for a combination of bits 3,5,7,9 and 11.
- The value of r2 is calculated to provide even parity with bits 3, 6, 7, 10 and 11.
- The value of r3 is calculated to provide even parity with bits 4,5,6 and 7.
- The value of r4 is calculated to provide even parity with bits 8,9,10 and 11.



Error Detection and Correction

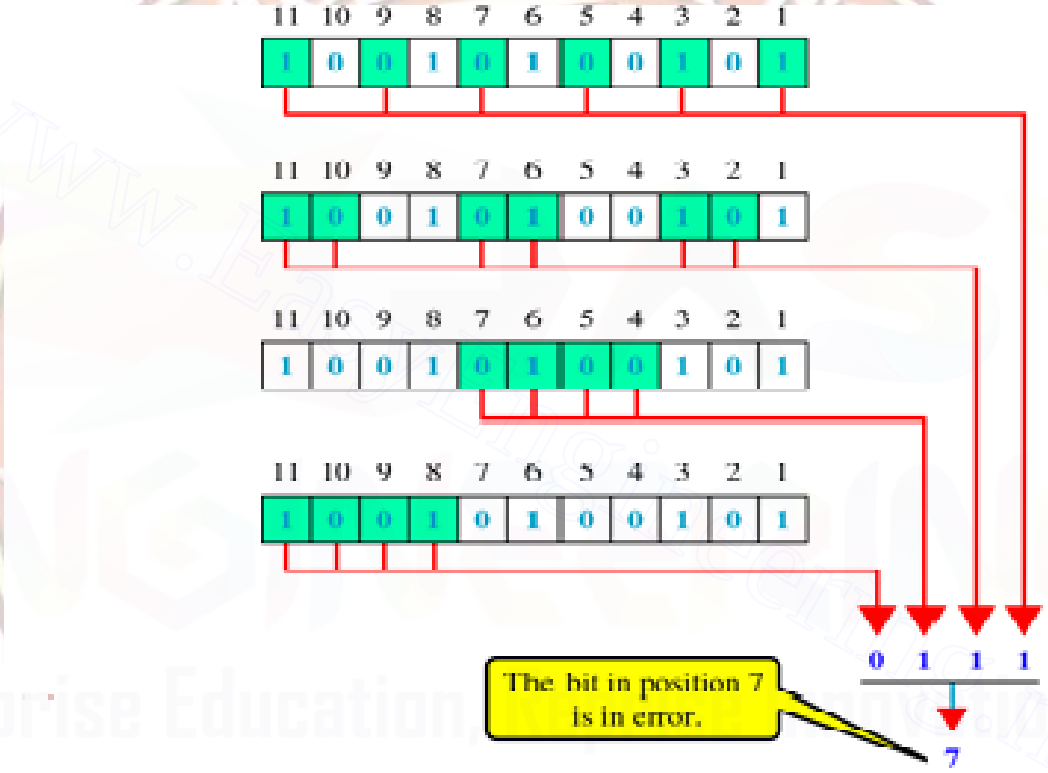
In the received data ,7th bit has been changed from 1 to 0.

Single-bit error



The receiver takes the transmission and recalculates four new data using the same set of bits used by the sender plus the relevant parity (r) bit for each set.

Error detection



- Then it assembles the new parity values into a binary number in order of r position (r8,r4,r2,r1).
- This step gives us the binary number 0111(7 in decimal) which is the precise location of the bit in error.
- Once the bit is identified, the receiver can reverse its value and correct the error.

Hamming Distance

One of the central concepts in coding for error control is the idea of the Hamming distance.

- The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits. The Hamming distance between two words *x* and *y* is *d(x, y).*
- The Hamming distance can be found by applying the XOR operation on the two words and count the number of 1's in the result.
- In a set of words, the minimum Hamming distance is the smallest Hamming distance between all possible pairs. We use dmin to define the minimum Hamming distance in a coding scheme.

Link Level Flow Control

Flow control is a technique that a transmitting entity does not conquer a receiving entity with data. Two fundamental mechanisms are acknowledgement and timeouts.

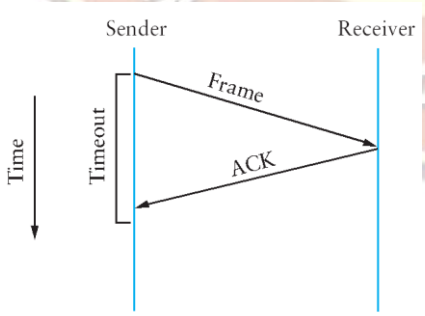
- After getting each frame the receiver will send ACK to sender.
- If the sender does not receive ACK up to reasonable amount of time then it retransmit the original frame waiting for reasonable amount of time is called timeout.

The two flow control mechanisms are

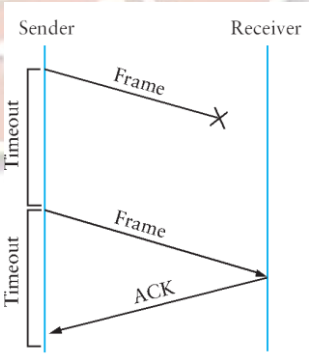
- Stop and wait Flow Control
- Sliding Window Flow Control

Stop and Wait Algorithm

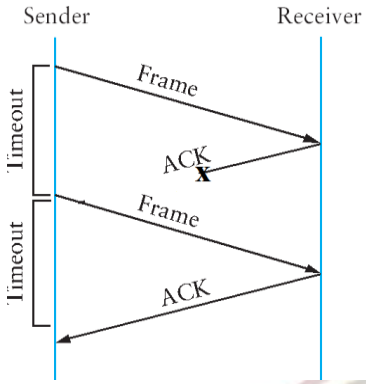
- After transmitting one frame, the sender waits for an acknowledgment before transmitting the next frame.
- If the acknowledgment does not arrive after a certain period of time, the sender times out and retransmit the original frame.



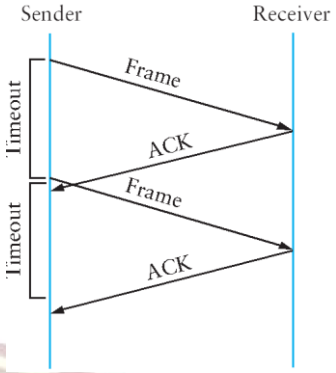
a) The ACK is received before the timer expires



b) The original frame is lost



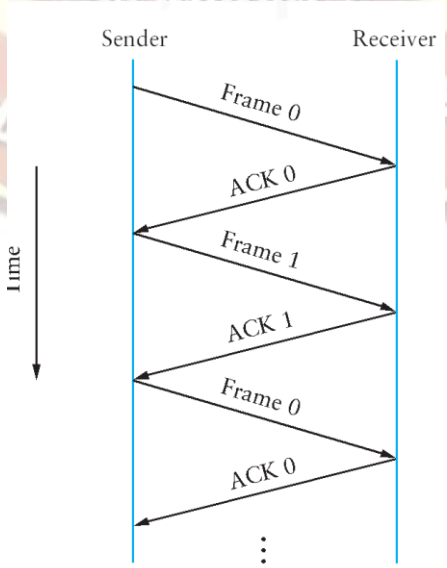
c) The ACK is lost



d) Delayed ACK

Fig: illustrates four different scenarios that result from this basic algorithm. The sending side is represented on the left, the receiving side is depicted on the right, and time flows from top to bottom.

- In Fig (a) ACK is received before the timer expires, (b) and (c) show the situation in which the original frame and the ACK, respectively, are lost, and (d) shows the situation in which the timeout fires too soon..
- Suppose the sender sends a frame and the receiver acknowledges it, but the acknowledgment is either lost or delayed in arriving. This situation is in (c) and (d). In both cases, the sender times out and retransmit the original frame, but the receiver will think that it is the next frame, since it correctly received and acknowledged the first frame.
- This makes the receiver to receive the duplicate copies. To avoid this two sequence numbers (0 and 1) must be used alternatively.



- The main drawback of the stop-and-wait algorithm is that it allows the sender have only one outstanding frame on the link at a time.

Sliding Window Algorithm

- The sender can transmit several frames before needing an acknowledgement.
- Frames can be sent one right after another meaning that the link can carry several frames at once and it s capacity can be used efficiently.
- The receiver acknowledges only some of the frames, using a single ACK to confirm the receipt of multiple data frames
- Sliding Window refers to imaginary boxes at both the sender and the receiver.
- Window can hold frames at either end and provides the upper limit on the number of frames that can be transmitted before requiring an acknowledgement.
- Frames are numbered modulo-n which means they are numbered from o to n-1
- For eg. If n=8 the frames are numbered 0,1,2,3,4,5,6,7. i.e the size of the window is n -1.
- When the receiver sends ACK it includes the number of the next frame it expects to receive.
- When the sender sees an ACK with the number 5, it knows that all frames up through number 4 have been received.

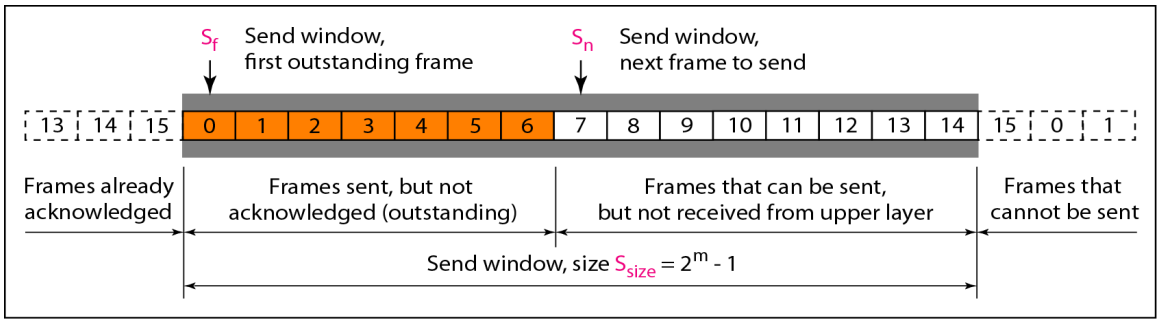
There are two methods to retransmit the lost frames

- GO-Back N
- Selective Repeat

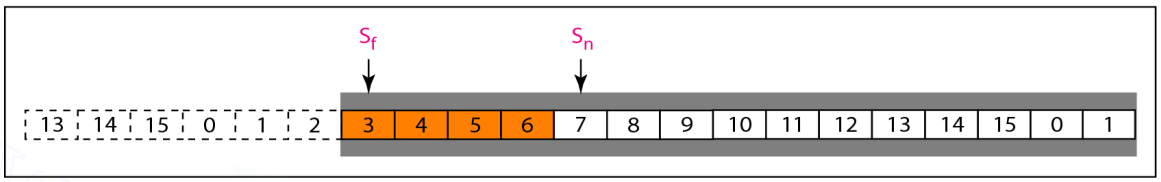
Go – Back N Method

Sender Window

- At the beginning of transmission, the sender window contains n-1 frames. As frames are sent out, the left boundary of the window moves inward, shrinking the size of the window
- If size of window is W if three frames have been transmitted since the last acknowledgement then the number of frames left in the window is w -3.
- Once an ACK arrives, the window expands to allow in a number of new frames equal to the number of frames acknowledged by that ACK.



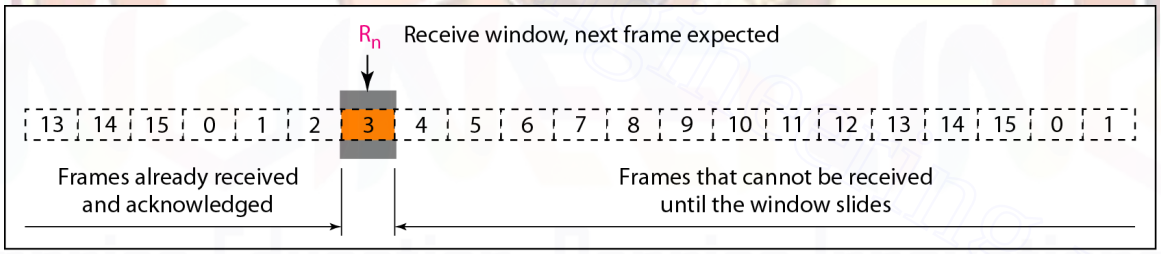
a. Send window before sliding



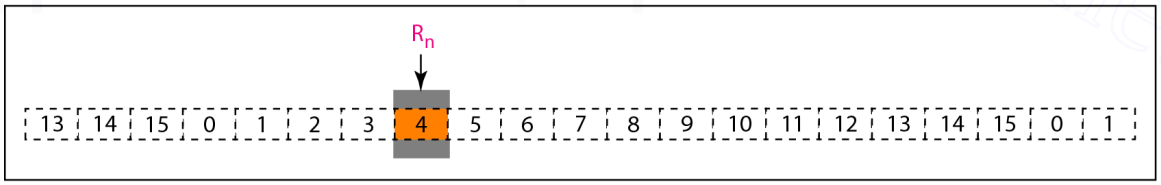
b. Send window after sliding

Receiver Window

- The receive window is an abstract concept defining an imaginary box of size 1 with one single variable Rn.
- The window slides when a correct frame has arrived, sliding occurs one slot at a time.



a. Receive window

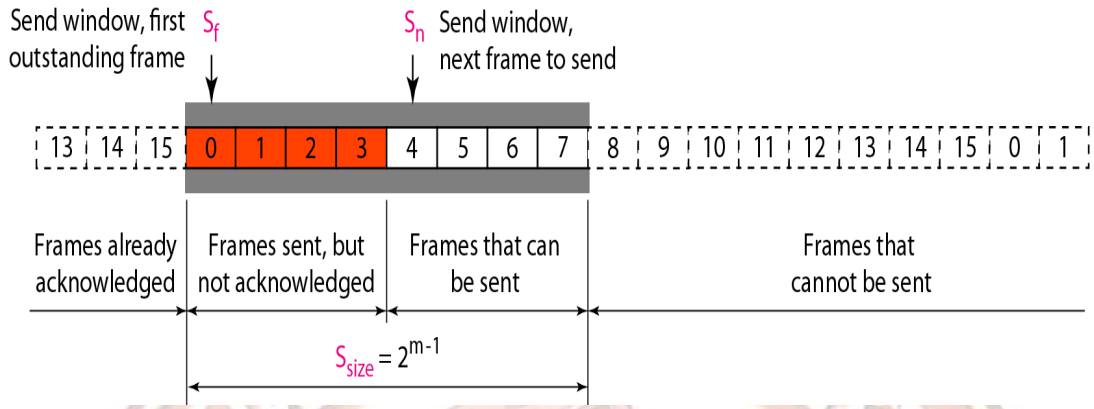


b. Window after sliding

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

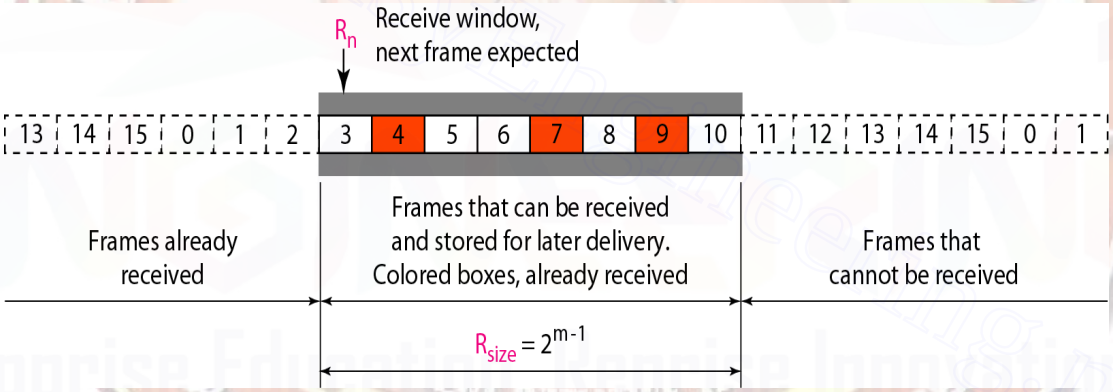
Selective Repeat

a. **Sender Window**



b.Receiver window

- The Selective Repeat Protocol allows as many frames as the size of the receive window to arrive out of order and be kept until there is a set of in-order frames to be delivered to the network layer.

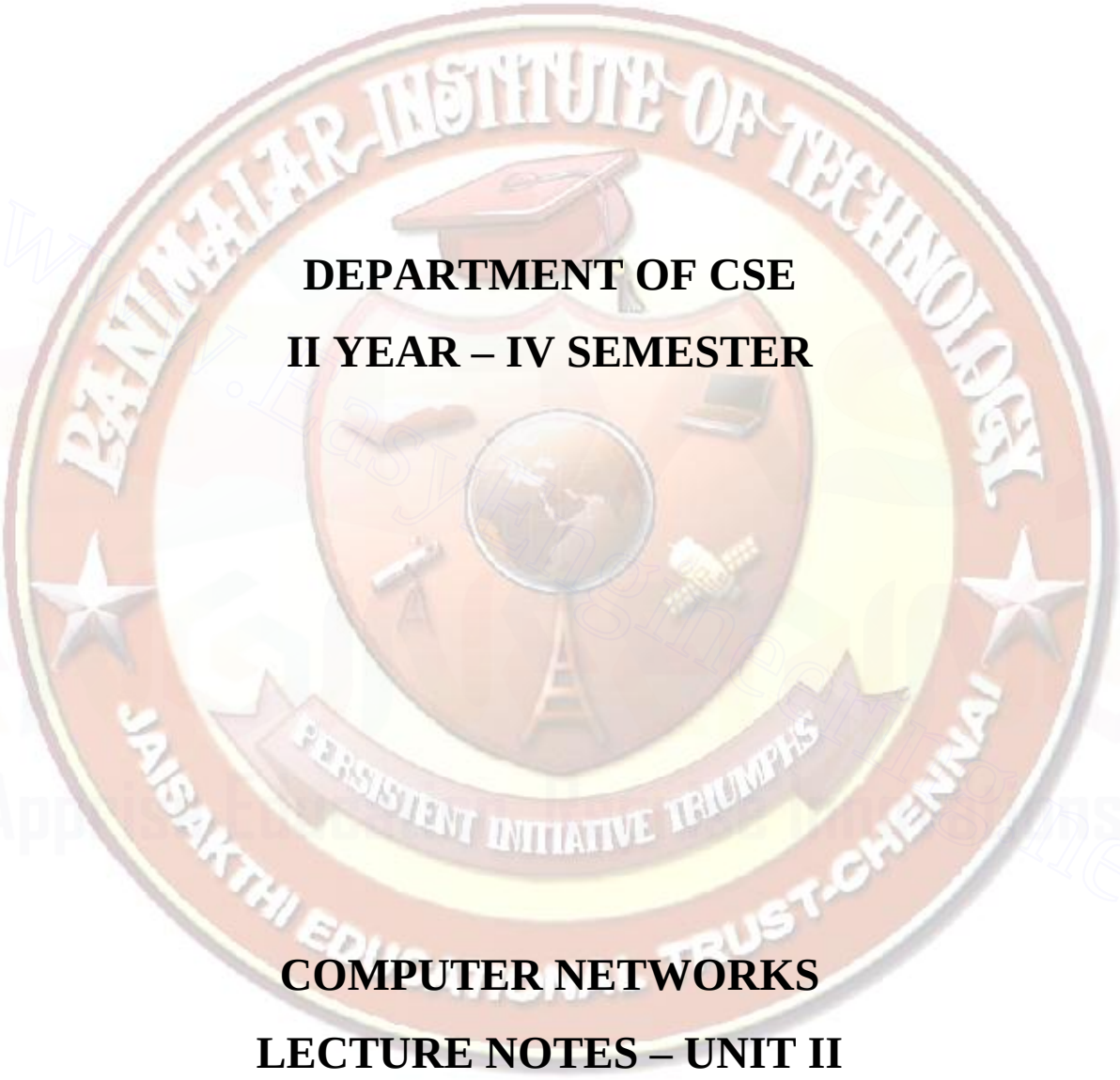


- Because the sizes of the send window and receive window are the same, all the frames in the send frame can arrive out of order and be stored until they can be delivered.
- If any frame lost, sender has to retransmit only that lost frames.

PANIMALAR INSTITUTE OF TECHNOLOGY

(JAISAKTHI EDUCATIONAL TRUST)

CHENNAI 602 103



DEPARTMENT OF CSE

II YEAR – IV SEMESTER

COMPUTER NETWORKS

LECTURE NOTES – UNIT II

UNIT II MEDIA ACCESS & INTERNET WORKING

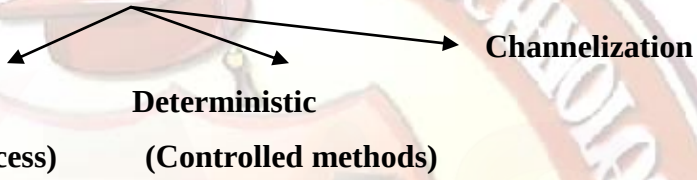
Media access control – Ethernet (802.3) – Wireless LAN' s – 802.11 – Bluetooth – Switching and bridging – Basic Internetworking (IP, CIDR, ARP, DHCP,ICMP)

2.1MEDIUM ACCESS:

One important feature of LAN is shared channel that provides multiple users to access the transmission facilities.

Thus there is possibility of conflict to occur. To resolve this, control mechanics are used. is used commonly.

Asynchronous TDM



Some examples of random access techniques are:

1) ALOHA
2) Carrier sense multiple access(CSMA)
3) CSMA/CD-Collision Detection
4) CSMA/CA-Collision Avoidance

Examples of controlled methods

1) Reservation
2) Polling
3) Token passing

Examples of Channelization methods

1) FDMA
2) TDMA
3) CDMA

RANDOM ACCESS METHODS:

Access to medium from many entry points is called contension. When several stations transmit at same line, there is possibility of collision to occur (ie) data collide with each other.

To overcome this, the following mechanisms are followed.

ALOHA:

This protocol was developed at university of Hawaii in early 1970's. it was developed for packet radio networks and it is applicable to any shared transmission medium.

Idea

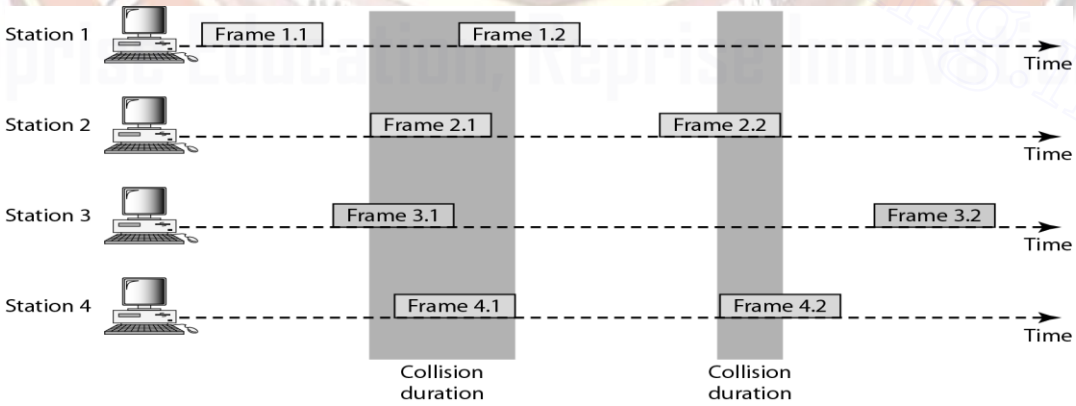
It is applicable to any system in which uncoordinated uses are competing for the use of single shared channel.

Types:

1) **Pure ALOHA:**

Original ALOHA is called pure ALOHA. The 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 channels.

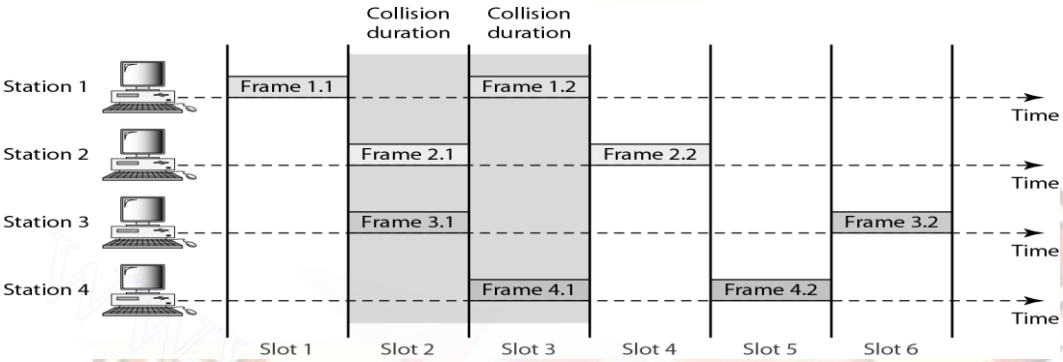
- They relay on acknowledgment from receiver
- Even if first bit new frame overlaps with last bit of old frame, both frames will be totally destroyed and will be retransmitted latter.



2) **Slotted ALOHA**

- The channel time is divided into time slots and stations are allowed to transmit at specific instances of time.

- These time slots are exactly equal to packet transmission time.
- All users are synchronized to these time slots, so that whatever a user generates a packet it must synchronize exactly with next possible channel slot.
- Thus, wasted time due to collision can be reduced to one packet time.



Carrier sense multiple access(CSMA)

Idea: Before sending information, every station listen the medium, thereby can reduce collision but cannot eliminate.

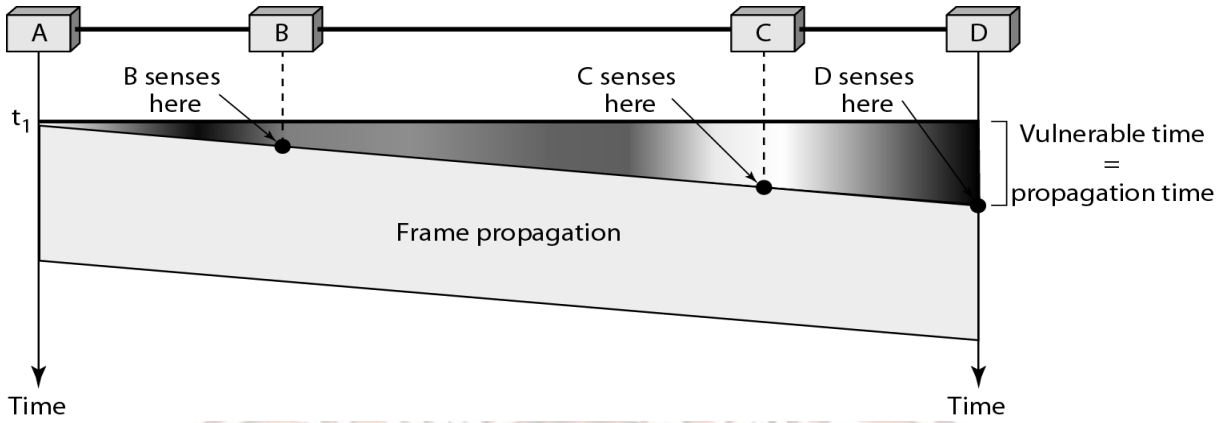
Collision occurs due to propagation delay.

Vulnerable time:

It is propagation time T_p (ie) the time needed for signal to propagate from one end to another.

After sensing the channel the actions to be taken are

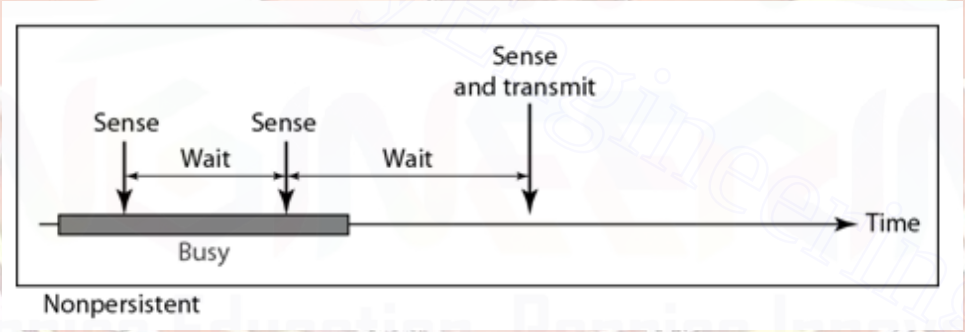
1) Non-Persistent CSMA
2) 1-Persistent CSMA
3) P-Persistent CSMA

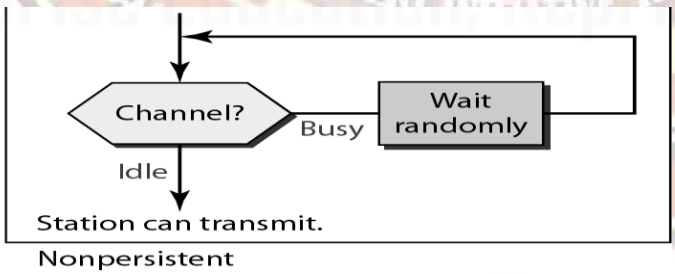


Non-Persistent CSMA

Here, when a station having a frame to transmit and finds the channel to be busy, it backs off for a fixed interval of time and checks the channel again and if the channel is free then it transmits.

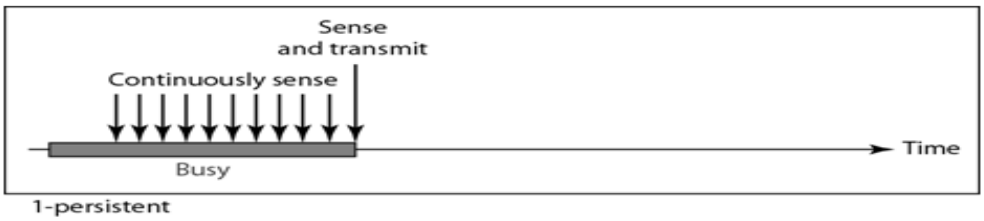
If channel is already in use, the station does not continuously senses, but it waits a random period of time and again checks for activity.

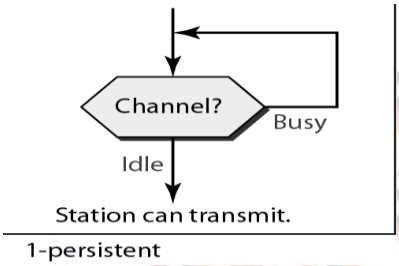




1-persistent CSMA:

Any station wishing to transmit, monitor the channel continuously until the channel is idle, and then transmits immediately with probability one, hence 1-persistent CSMA when two or more station are waiting to transmit a collision is guaranteed.



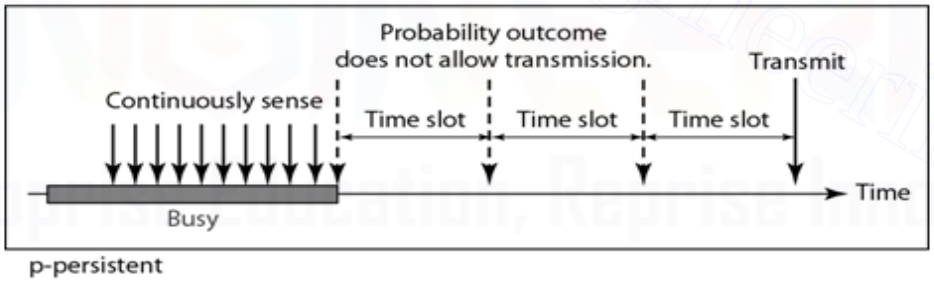


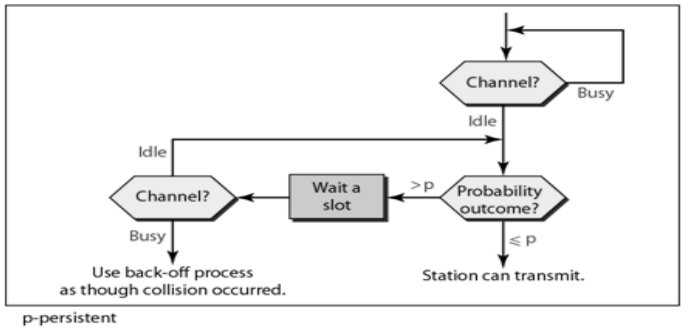
P-persistent CSMA:

To reduce the probability of collision in I-persistent CSMA, not all the waiting stations are allowed to transmit immediately, after the channel is idle.

When station becomes ready to send and it senses the channel to be idle, it either transmits with probability p or it delay transmission by one time slot with a probability q=1-P.

If the deferred slot is also idle, the station transmits with probability P or defers again with a probability q. This process is repeated.





Carrier Sense Multiple Access with Collision Detection CSMA/CD:

CSMA/CD is the commonly used protocol for LANs. CSMA/CD specifications were developed jointly by digital equipment corporation (DEC), Intel and Xerox. This network is called Ethernet IEEE 802.3 CSMA/CD standard for LAN is based on Ethernet specification.

Idea:

The station with message to send must monitor the channel to see if any station is sending. If other station is sending, the then second station must wait until it finishes

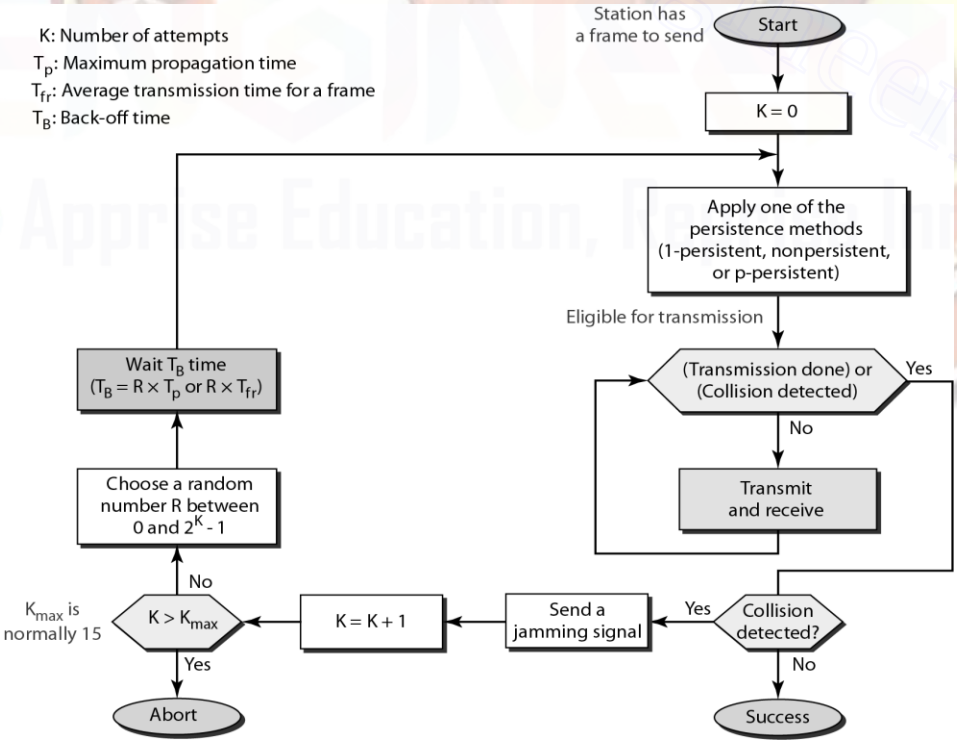
The term **Carrier Sense** indicates that listening before transmitting. If two or more station have to send, at same time and separated by distance on bus, then each will not be aware of other and hence transmit at same it and collide.

Here, a protocol is required for transmitting station to monitor the channel while sending its message and detect collisions.

When collision is detected, sending stations must cease transmitting, wait for random time and try again.

Because of quite termination of transmission, time and bandwidth is saved and hence it is more efficient than ALOHA and slotted ALOHA and also CSMA.

Flow chart for CSMA/CD procedure:



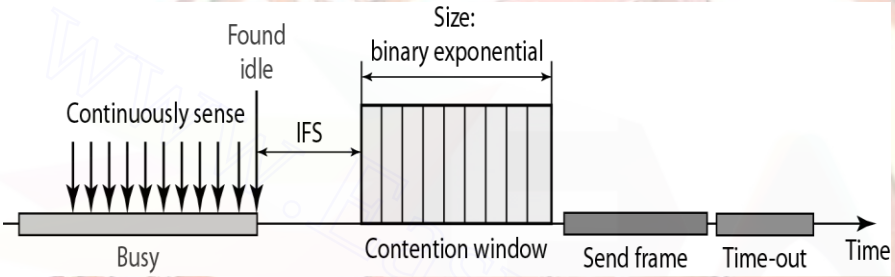
Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

Wireless networks cannot use CSMA/CD in the MAC sublayer, since this requires the ability to receive and transmit at the same time.

Collision needed to be avoided in wireless networks since they cannot be detected. Collision cannot be detected easily in wireless networks.

Collision are avoided by using 3 methods

1) Inter frame space
2) Contention window
3) Acknowledgments



Inter frame space

- Collisions are avoided by deferring the transmission ever if channel is idle.
- When a idle channel is found, the station does not send immediately but waits for a period of time called inter frame space.

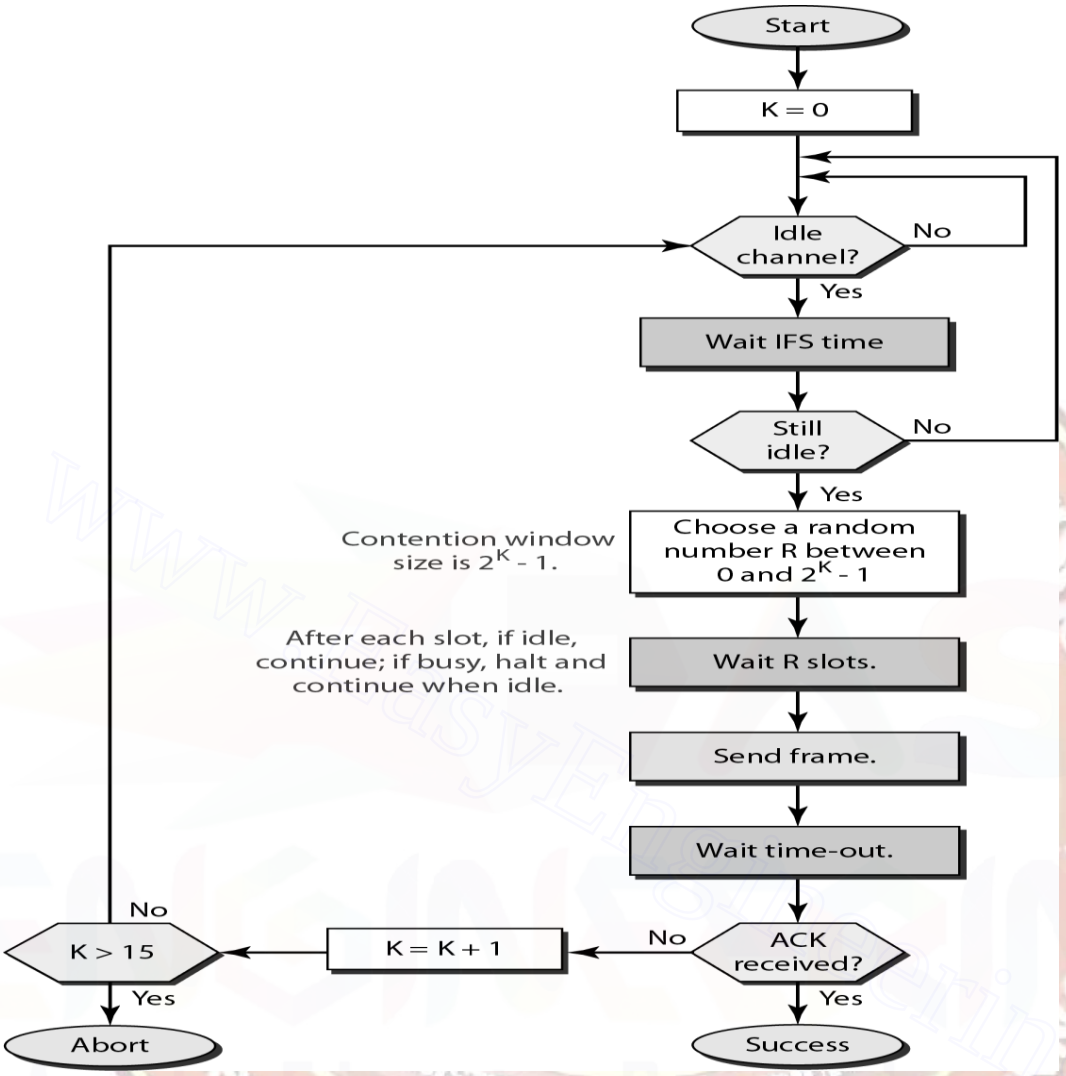
Contention window

They are an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time

- Station set one slot for first time and then double each time the station cannot detect an idle channel after the IFS time.
- Station needs to sense the channel after each time slot
- If station find channel busy. It does not restart the process it first stop the timer and restarts it when the channel is sensed as idle.
- This method gives priority to station with longest waiting time.

Acknowledgments

The positive acknowledgments and time out can help guarantee that receiver has received the frame.



CONTROLLED ACCESS:

In this method the stations consult one another to find which station has right to send. Each station cannot send unless it has been authorized by other stations.

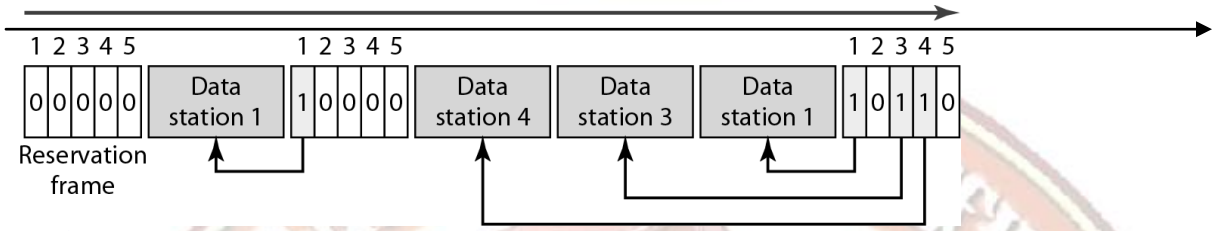
Methods:

1) Reservation
2) Polling
3) Token passing

Reservation

Before sending data, stations needs to make a reservation. Number of reservation are equal to number of stations. Each stations have their own minislot in reservation frame.

When station needs to send a data frame it makes a reservation in its own minislots.

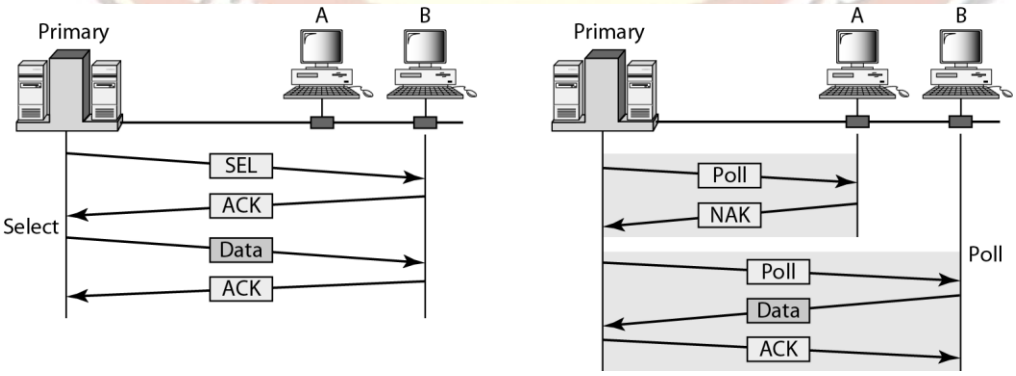


The stations that have made reservation can send data frames after the reservation time. In the first slot, only station 1,2,3 and 4 made reservation

Polling

One device is designed as primary station and other devices are secondary

- Link control done by primary
- All data exchange take place through primary
- Primary decides, which device is allowed to use channel at a given time.
- If primary wants to receive data, it asks the secondaries if they have anything to send (ie) polling
- Select mode and poll mode are two functions.
- In polling, primary receive the data.
- In select mode, primary sends data to secondary.

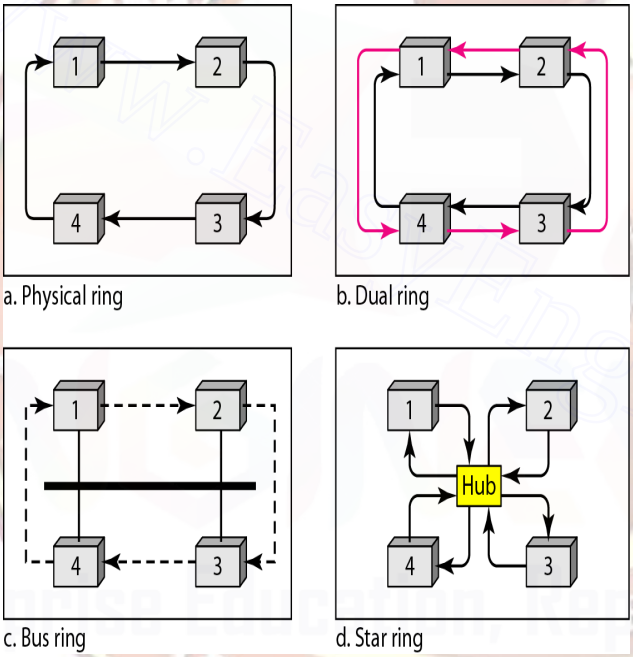


- Before sending datas primary creates and transmits a select (SEL) frame.

- SEL frame include address of intended secondary device.

Token passing :

- A station is allowed to send data when it receives a token.
- Ring topology is used for connecting devices
- Each station has a predecessor & successor
- Frames come from predecessor and go to successor
- Token circulates around the ring
- Any station can capture token if they want to send data



PROJECT 802

In 1985, the computer security of the IEEE started a project, called Project 802, to set standards to enable intercommunications between equipment from a variety of a manufacture.

The relationship of IEEE Project 802 to the OSI model is as follows

The IEEE has subdivided the data link into two sublayers

1) Logical Link Control layer(LLC)
2) Medium access control(MAC)

IEEE standards:

The institute of electrical and electronics engineers(IEEE) publish several accepted LAN recommended standards. These standards collectively known as IEEE 802. Eg:

1. IEEE 802.1 (internetworking)
2. IEEE 802.2 local link control
3. IEEE 802.3 CSMA/CD- ethernet
4. IEEE 802.4 token bus
5. IEEE 802.5 token ring
6. IEEE 802.6 metropolitan Area network
7. IEEE 802.7 broadband LANS
8. IEEE 802.8 fiber optics LANS IEEE 802.9 intergrated data and voice network
9. IEEE 802.10 security
10. IEEE 802.11 wireless network

ETHERNET

IEEE 802.3 supports a LAN standard originally developed by Xerox and this was called Ethernet. IEEE 802.3 defines two categories:

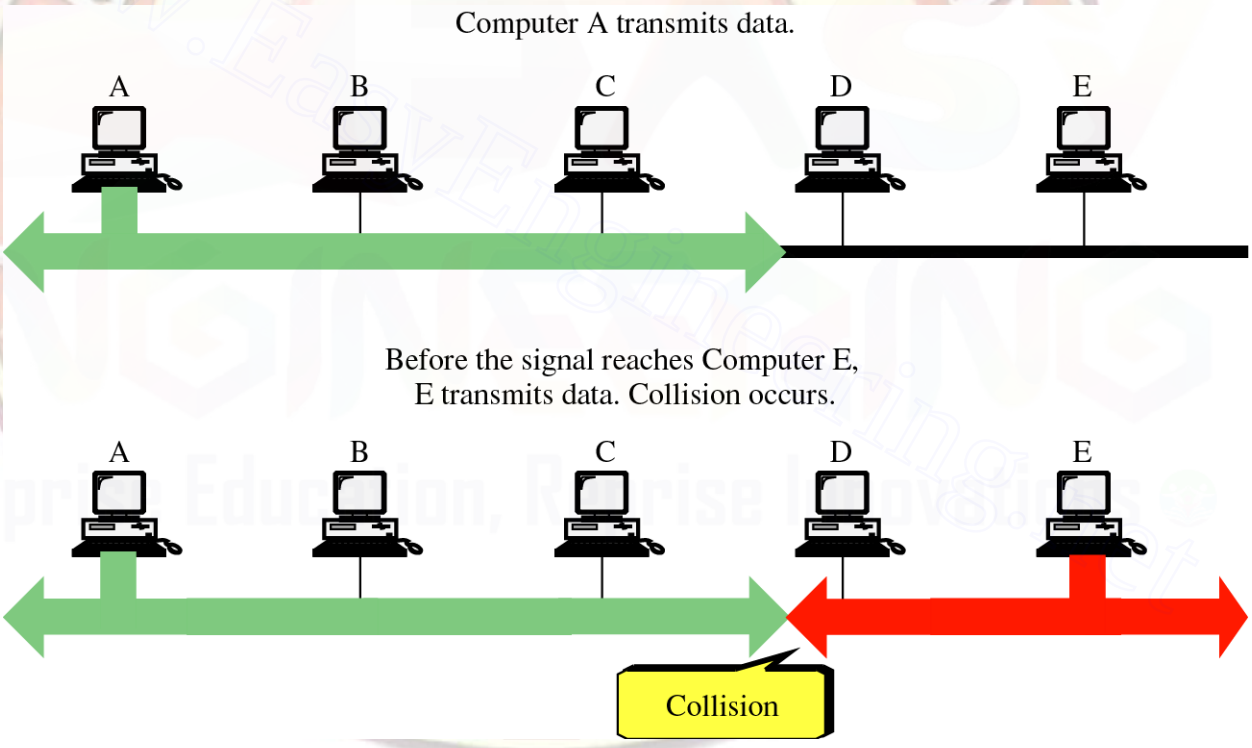
1. base band
2. broad band

The word base specifies a digital signal. It divides five different standard.**10base5, 10base2, 10base-T, 1base5 and 100base-T.** The first number indicate (10, 1, 100) the data rate in mbps. The last number indicates (5, 2, 1, T) indicate maximum length or type of cable.

ACCESS METHOD (CSMA/CD):

- The access mechanism used in an Ethernet is called carrier sense multiple access with collision detection (CSMA /CD).
- The original design was a multiple access method in which every workstation had equal access to a link. In MA, there was no provision for traffic coordination.
- Any station wishing to transmit did so, and then relied on acknowledgments to verify that the transmitted frame had not been destroyed by other traffic on the line.

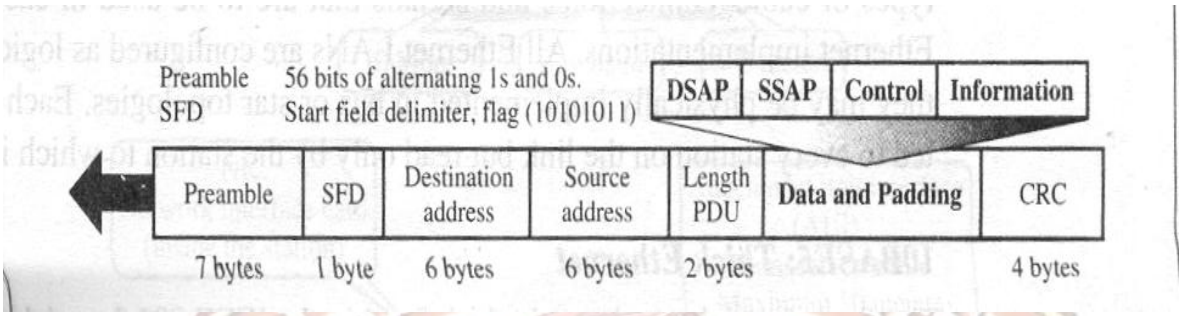
- In a CSMA system, any workstation wishing to transmit must first listen for existing traffic on the line. A device listens by checking for a voltage. If no voltage is detected, the line is considered idle and the transmission is initiated.
- CSMA cuts down on the number of collisions but does not eliminate them. Collisions can still occur.
- The final step is the addition of collision detection (CD). In CSMA/CD the station wishing to transmit first listens to make certain the link is free, and then transmits its data, then listens again.
- During the data transmission, the station checks the line for the extremely high voltages that indicate a collision. If a collision is detected, the station. Quits the current transmission and waits a predetermined amount of time for the line clear, then sends its data again.



Frame Format

IEEE 802.3 specifies one type of frame containing seven fields: preamble, SRI, DA, SA, length/type of PDU, 802.2 frames, and the CRC. Ethernet does not provide at 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 of the MAC frame in CSMA/CD is shown in Figure.



Preamble

The first field of the 802.3 frame, the preamble, contains seven bytes (56 bits) of alternating 0s and 1s that alert the receiving system to the coming frame and enable it to synchronize its input timing.

The pattern 1010101 provides only an alert and a timing pulse; it can be too easily aliased to be useful in indicating the beginning of the data stream. HDLC combined the alert, timing, and start synchronization into a single field.

Start frame delimiter (SFD).

The second field (one byte: 10101011) of the 802.3 frame signals the beginning of the frame. The SFD tells the receiver that every thing that follows is data, starting with the addresses.

Destination address (DA).

The destination address (DA) field is allotted six bytes and contains the physical address of the packet's next destination.

A system's physical address is a bit pattern encoded on its network interface card (NIC). Each NIC has a unique address that distinguishes it from any other NIC.

If the packet must cross from one LAN to another to reach its destination, the DA field contains the physical address of the router connecting the current LAN to the next one

When the packet reaches the target network, the DA field contains the physical address of the destination device.

Source address (SA). The source address (SA) field is also allotted six bytes and contains the physical address of the last device to forward the packet. That device can be the sending station or the most recent router to receive and forward the packet.

Length/type of PDU.

These next two bytes indicate the number of bytes in the coming PDU. If the length of the PDU is fixed, this field can be used to indicate type, or as a base for other protocols.

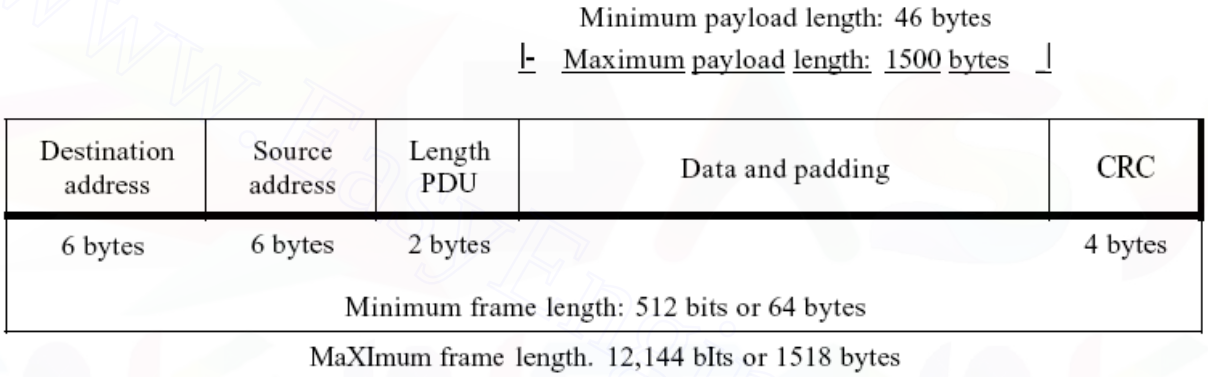
802.2 frame (PDU).

This field of the 802.3 frame contains the entire 802.2 frame as a modular, removable unit. The PDU can be anywhere from 46 to 1500 bytes long, depending on the type of frame and the length of the information field. The PDU is generated by the upper (LLC) sub layer.

CRC

The last field in the 802.3 frame contains the error detection information, in this case a CRC-32.

Frame length:



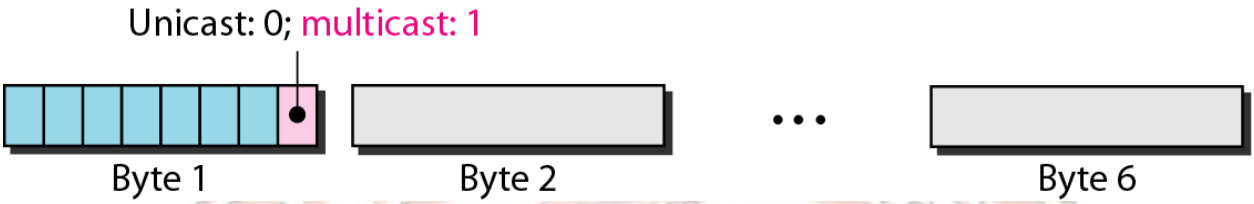
ADDRESSING

Each station on an Ethernet network (such as a PC, workstation, or printer) has its own network interface card (NIC). The NIC fits inside the station and provides the station with a 6-byte physical address. The Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes.

Example of an Ethernet address in hexadecimal notation

Unicast, Multicast, and Broadcast Addresses

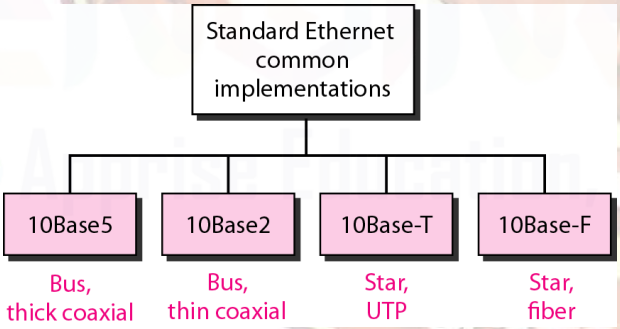
A source address is always a unicast address (i.e) the frame comes from only one station. The destination address, however, can be unicast, multicast or broadcast. If the least significant bit of the first byte in a destination address is 0, then the address is unicast; otherwise, it is multicast.



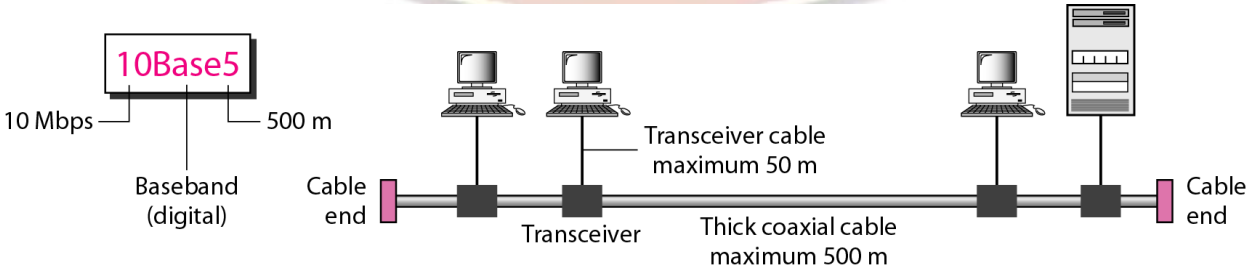
A unicast destination address defines only one recipient; the relationship between the sender and the receiver is one-to-one. A multicast destination address defines a group of addresses; the relationship between the sender and the receivers is one-to-many. The broadcast address is a special case of the multicast address; the recipients are all the stations on the LAN. A broadcast destination address is forty-eight 1's(11111111 1111111 11111111 11111111 11111111 11111111)

ETHERNET IMPLEMENTATION:

In the 802.3 standard, the IEEE defines the types of cable, connections, and signals that are to be used in each of different Ethernet implementations.



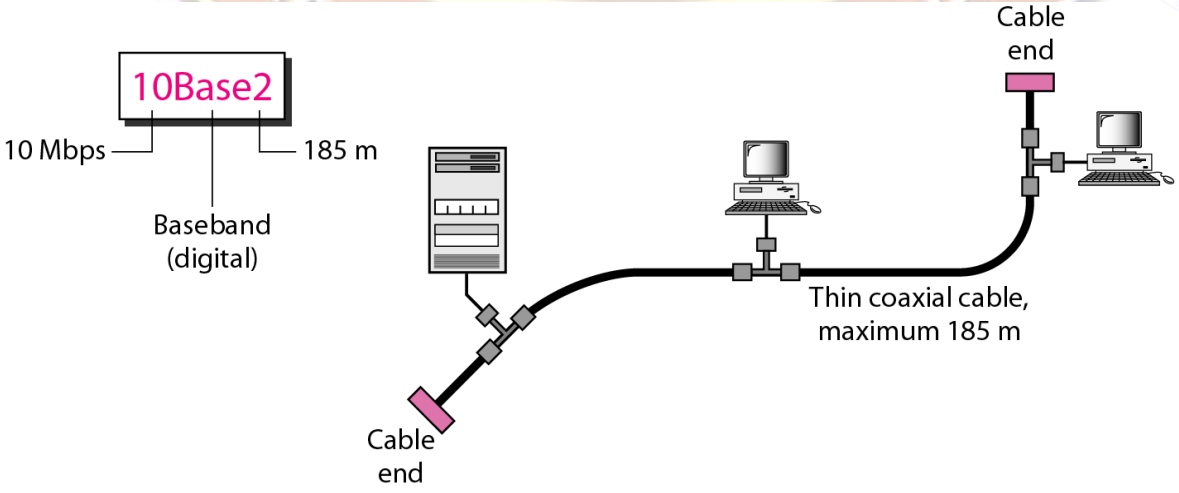
10 BASE 5: Thick Ethernet



- The first of the physical standards defined in the IEEE 802.3 model is called lOBase5 thick Ethernet, or Thicknet. The nickname derives from the size of the cable, which is roughly the size of garden hose and too stiff to bend with your hands.
- 10Base5 is a BUS topology LAN that uses baseband signaling and has a maximum segment length of 500 meters.
- In this case, the length of each segment is limited to 500 meters. However, to reduce collisions, the total length of the bus should not exceed 2500- meters (five segments). Also, the standard: demands that each station be separated from each neighbor by 2.5 meters (200 station per segment and 1000 stations total)
- The physical connectors and cables utilized by 10Base5 include coaxial cable, network interface cards, transceivers, and attachment unit interface (AUI) cables

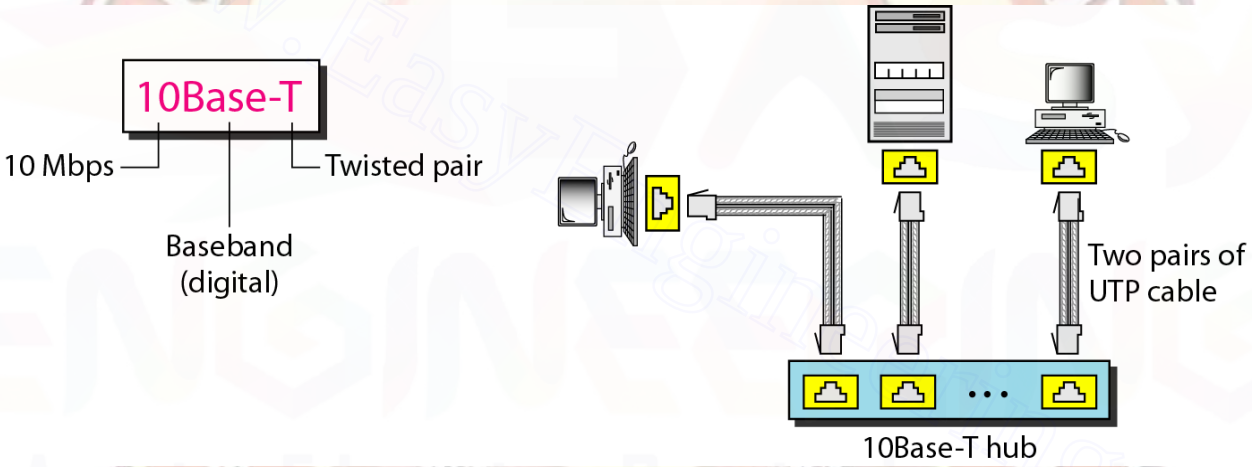
10BASE2: Thin Ethernet

- The second Ethernet implementation defined by the IEEE 802 series is called 10Base2 or thin Ethernet.
- Thin Ethernet (also called Thinnet, cheapnet, cheapernet, and thin wire Ethernet) provides an inexpensive alternative to 10Base5 Ethernet, with the same data rate. Like 10Base2, is a bus topology LAN.
- The advantages thin ether net is reduced cost and ease of installation. The disadvantages are shorter range and smaller capacity.
- The physical layout of lOBase2 is illustrated in Figure. The connect cables utilized are: NICs, thin coaxial cable, and BNC-T connectors.



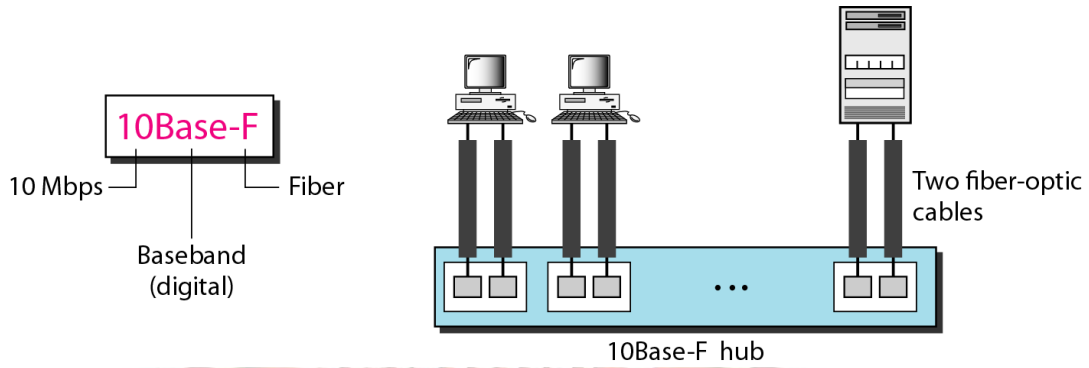
1OBASE-T: Twisted-Pair Ethernet

- The most popular standard defined in the IEEE 802.3 series is l0Base-T (also called twisted-pair Ethernet), a star-topology LAN using unshielded twisted pair (UTP) cab instead of coaxial cable.
- Instead of individual transceivers, l0Base-T Ethernet places all of its networking operations in an intelligent hub with a port for each station.
- Stations are linked into the hub by four-pair RJ-45 cable (eight-wire unshielded twisted-pair cable) terminating a each end in a male-type connector much like a telephone jack.
- The hub fans out any transmitted frame to all of its connected stations. Logic in the NIC assures that the only station to open and read a given frame is the station to which that frame is addressed.



1OBASE-F: Fiber Optic

- **10BaseF** is an implementation of Ethernet 802.3 over fiber optic cabling.
- 10BaseF offers only 10 Mbps, even though the fiber optic media has the capacity for much faster data rates.
- One of the implementations of 10BaseF is to connect two hubs as well as connecting hubs to workstations.
- The best time to use 10BaseF is in the rewiring of a network from copper to fiber optic, when you need an intermediate protocol using the new wiring.
- 10BaseF is not often a permanent solution because the data rate is so low and the cabling so expensive in comparison to using UTP.



Summary of Standard Ethernet implementations

Characteristics	*10Base5*	*10Base2*	*10Base-T*	*10Base-F*
Media	Thick coaxial cable	Thin coaxial cable	2 UTP	2 Fiber
Maximum length	500 m	185 m	100 m	2000 m
Line encoding	Manchester	Manchester	Manchester	Manchester

WIRELESS LAN (WLAN 802.11)

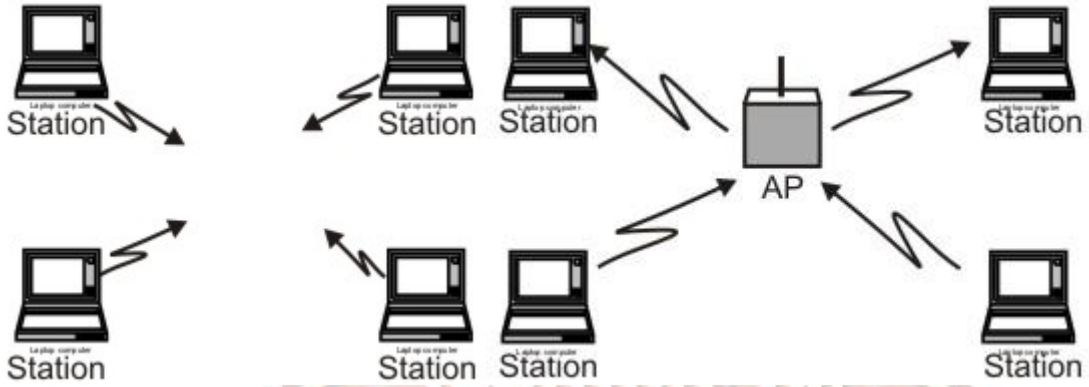
IEEE802.11 ARCHITECTURE AND SERVICES

The standard defines two kinds of services: the basic service set (BSS) and the extended service set (ESS).

Basic Service Set

IEEE 802.11 defines the basic service set (BSS) as the building block of a wireless LAN. A basic service set is made of stationary or mobile wireless stations and an optional central base station, known as the access point (AP).

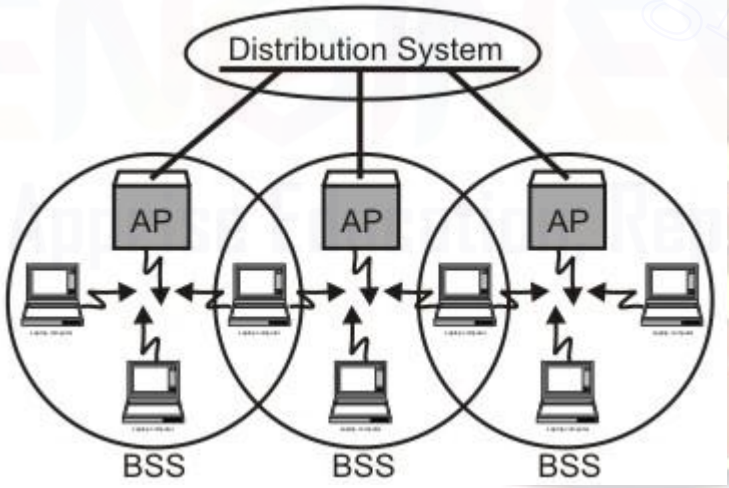
- The BSS without an AP is a stand-alone network and cannot send data to other BSSs. It is called an **ad hoc architecture**. In this architecture, stations can form a network without the need of an AP; they can locate one another and agree to be part of a BSS.
- A BSS with an AP is sometimes referred to as an ***infrastructure* network**.



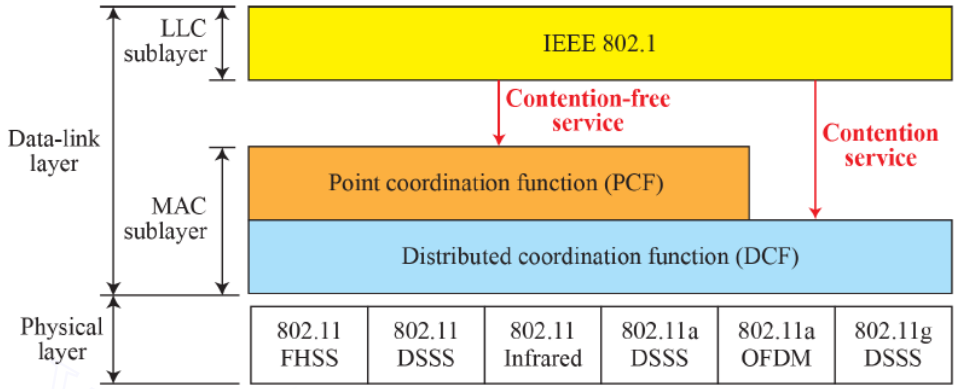
(a) BSS without AP (b) BSS with AP

Extended Service Set

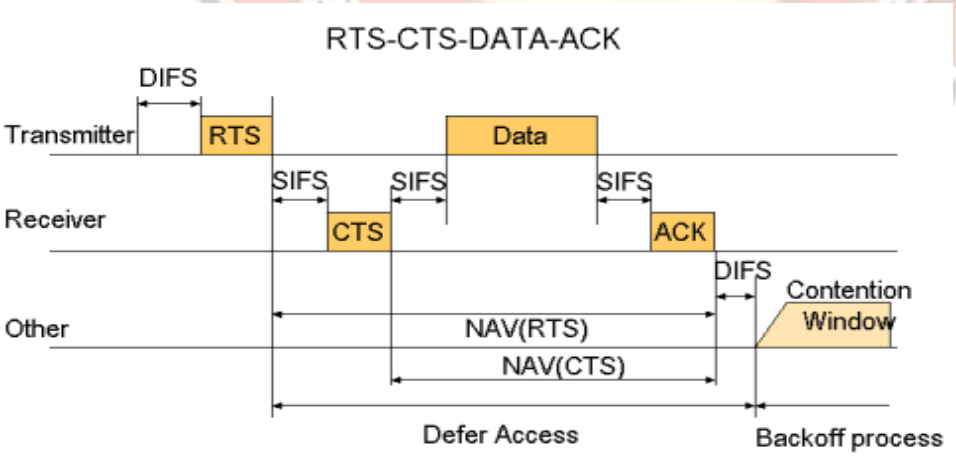
- An extended service set (ESS) is made up of two or more BSSs with APs. In this case, the BSSs are connected through a distribution system, which is usually a wired LAN. The distribution system connects the APs in the BSSs.
- IEEE 802.11 does not restrict the distribution system; it can be any IEEE LAN such as an Ethernet. Note that the extended service set uses two types of stations: mobile and stationary. The mobile stations are normal stations inside a BSS.
- The stationary stations are AP stations that are part of a wired LAN.



MAC layers in IEEE 802.11 standard



- **Distributed Coordination Function (DCF)**
  - Distributed access protocol
  - Contention-Based
  - Makes **use of CSMA/CA** rather than CSMA/CD
  - Suited for ad hoc network and ordinary asynchronous traffic
- **Point Coordination Function (PCF)**
  - Alternative access method on top of DCF
  - Centralized access protocol
  - Contention-Free
  - Works like **polling**
  - Suited for time bound services like voice or multimedia



DIFS: Distributed IFS

RTS: Request To Send

SIFS: Short IFS

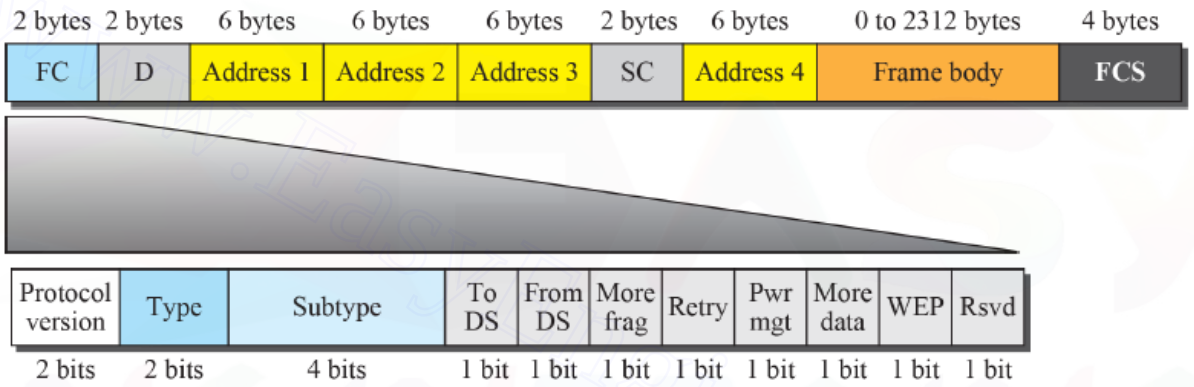
CTS: Clear To Send

ACK: Acknowledgement

NAV: Network Allocation Vector

DCF: Distributed Coordination Function

Frame format



- **Frame Control Field (in MAC header)**
  The protocol version field is 2 bits in length and will carry the version of the 802.11 standard. The initial value of 802.11 is 0; all other bit values are reserved.
- **Type** and **subtype** fields are 2 and 4 bits, respectively. They work together hierarchically to determine the function of the frame.
- The remaining 8 fields are all 1 bit in length.
- The **To DS** field is set to 1 if the frame is destined for the distribution system.
- **From DS** field is set to 1 when frames exit the distribution system. Note that frames which stay within their basic service set have both of these fields set to 0.
- The More **Frag field** is set to 1 if there is a following fragment of the current MSDU.
- **Retry** is set to 1 if this frame is a retransmission.
- **Power Management** field indicates if a station is in power save mode (set to 1) or active (set to 0).
- **More data** field is set to 1 if there is any MSDUs are buffered for that station.
- The **WEP** field is set to 1 if the information in the frame body was processed with the WEP algorithm.
- The **Order** field is set to 1 if the frames must be strictly ordered.

- **The Duration/ID field** is 2 bytes long. It contains the data on the duration value for each field and for control frames it carries the associated identity of the transmitting station.
- The **address fields** identify the basic service set, the destination address, the source address, and the receiver and transmitter addresses. Each address field is 6 bytes long.
- The **sequence control field** is 2 bytes and is split into 2 subfields, fragment number and sequence number.
- **Fragment number** is 4 bits and tells how many fragments the MSDU is broken into.
- The **sequence number field** is 12 bits that indicates the sequence number of the MSDU. The frame body is a variable length field from 0 - 2312. This is the payload.

Addressing Mechanism

- **The IEEE 802.11 addressing mechanism specifies four cases, defined by the value of the two flags in the FC field, To DS and From DS.**

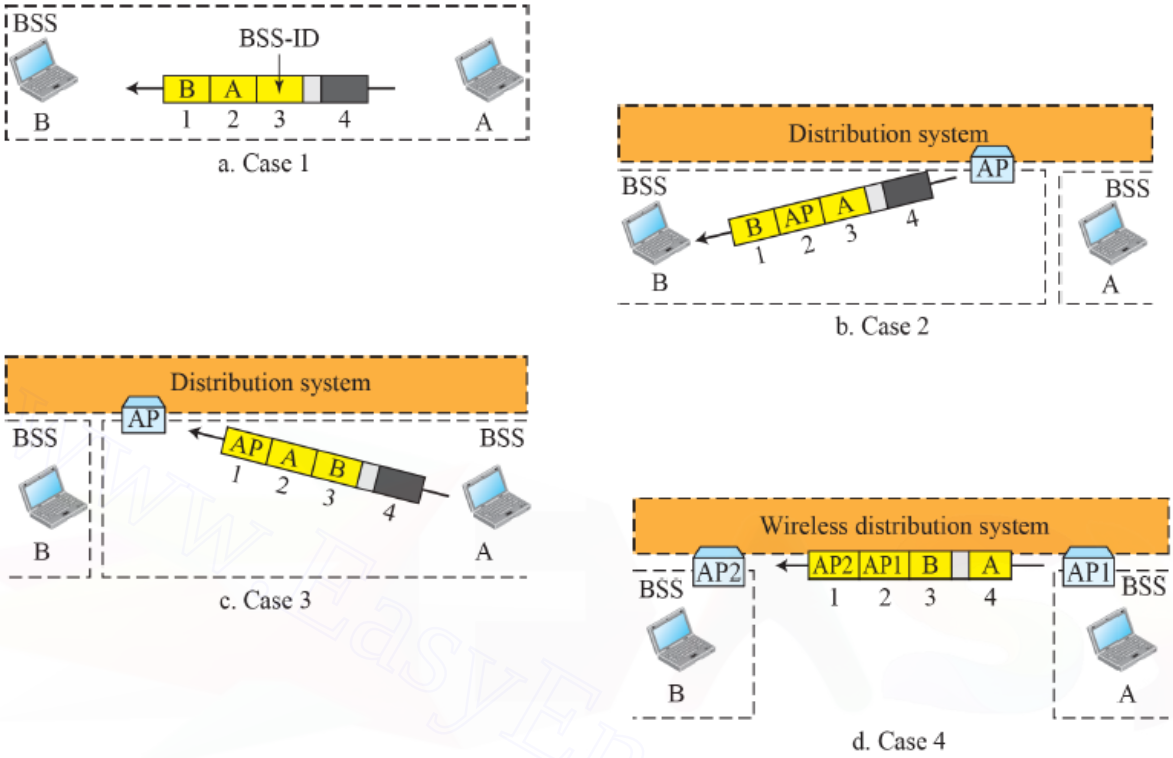
- **Each flag can be either 0 or 1, resulting in four different situations.**

- **The interpretation of the four addresses (address 1 to address 4) in the MAC frame depends on the value of these flags**

Addresses

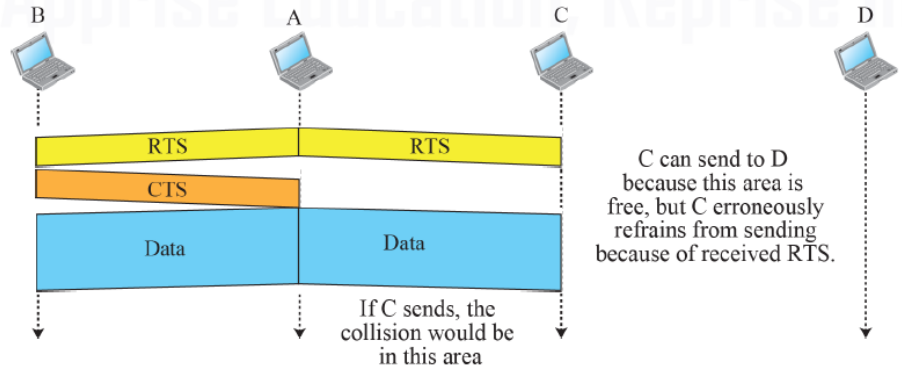
To DS	*From DS*	*Address 1*	*Address 2*	*Address 3*	*Address 4*
0	0	Destination	Source	BSS ID	N/A
0	1	Destination	Sending AP	Source	N/A
1	0	Receiving AP	Source	Destination	N/A
1	1	Receiving AP	Sending AP	Destination	Source

Addressing mechanisms



Problems in WLAN

Exposed station problem



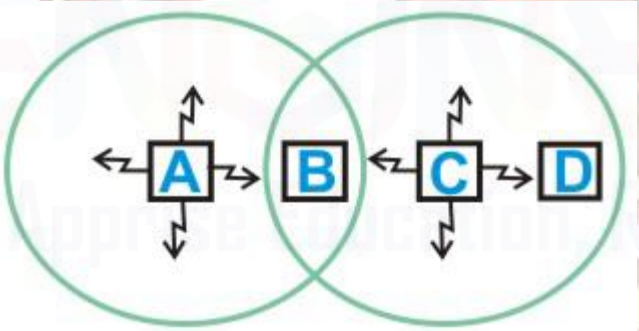
Most wired LANs products use Carrier Sense Multiple Access with Collision Detection (CSMA/CD) as the MAC protocol. Carrier Sense means that the station will listen before it transmits. If there is already someone transmitting, then the station waits and tries again later. If

no one is transmitting then the station goes ahead and sends what it has. But when more than one station tries to transmit, the transmissions will collide and the information will be lost. This is where Collision Detection comes into play. The station will listen to ensure that its transmission made it to the destination without collisions. If a collision occurred then the stations wait and try again later. The time the station waits is determined by the back off algorithm. This technique works great for wired LANs but wireless topologies can create a problem for CSMA/CD. However, the wireless medium presents some unique challenges not present in wired LANs that must be dealt with by the MAC used for IEEE 802.11. Some of the challenges are:

The wireless LAN is prone to more interference and is less reliable.
The wireless LAN is susceptible to unwanted interception leading to security problems.
There are so called *hidden station* and *exposed station* problems.

The Hidden Station Problem

Consider a situation when A is transmitting to B, as depicted in the Fig. 5.7.7. If C senses the media, it will not hear anything because it is out of range, and thus will falsely conclude that no transmission is going on and will start transmit to B. the transmission will interfere at B, wiping out the frame from A. The problem of a station not been able to detect a potential competitor for the medium because the competitor is too far away is referred as *Hidden Station Problem.* As in the described scenario C act as a hidden station to A, which is also competing for the medium.



BLUETOOTH

- Bluetooth is a wireless LAN technology designed to connect devices of different functions such as telephones, notebooks, computers, cameras, printers, coffee makers, and so on.
- It is an ad hoc network (i.e) network is formed spontaneously.
- The devices are sometimes called **Gadgets.**

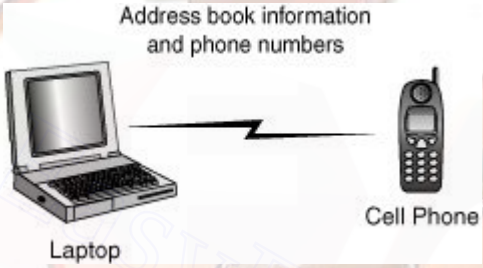
BLUETOOTH ARCHITECTURE

Bluetooth defines two types of networks:

1. Piconet
2. Scatternet

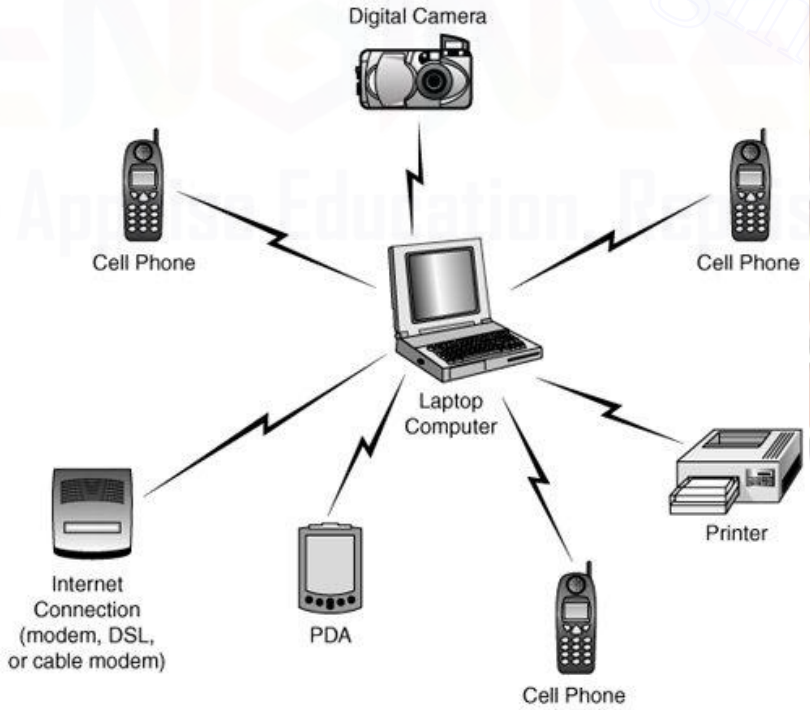
Piconet (Smallnet)

A piconet is formed when two or more devices discover each other and begin to communicate. A piconet can have up to eight devices, with one device acting as a master and the rest acting as slaves. The first device to initiate transmission becomes the master. A specific frequency-hopping sequence is used by all devices within each piconet. Figure shows the simplest example of a piconet:



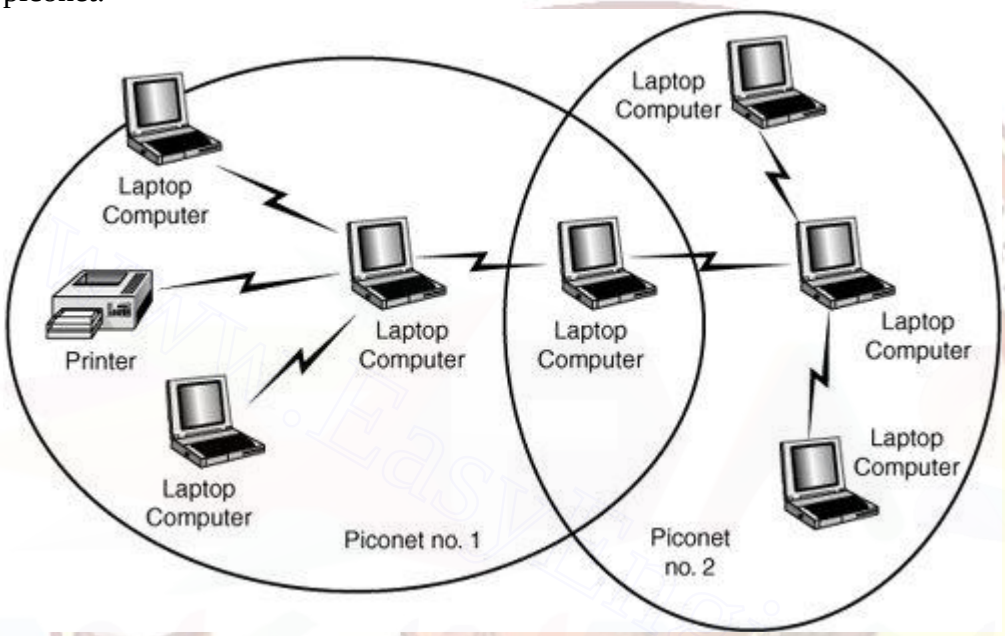
A piconet can have only one master and up to seven slave devices.

Photographic representation of seven slave devices in Piconet



Scatternet:

Group of piconet is called **Scatternet**. A device can be a master of only one piconet. The device can, at the same time, also be a slave in another piconet that is within range. A slave can also participate in two different piconets that are within its range. However, the master device determines the **hopping pattern** used for a piconet, a device cannot be a master of more than one piconet.

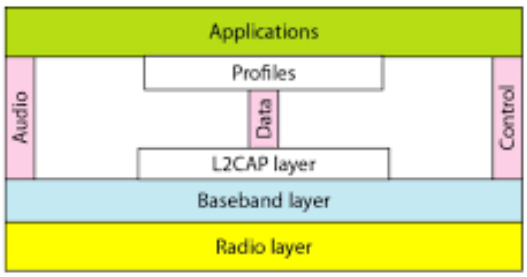


Photographic representation of Scatternet

BLUETOOTH LAYERS AND PROTOCOL STACK

Bluetooth is a wireless LAN technology designed to connect devices of different functions. Bluetooth standard has many protocols that are organized into different layers. The layer structure of Bluetooth does not follow OS1 model, TCP/IP model or any other known model.

The different layers and Bluetooth protocol architecture.



Radio Layer

1. The Bluetooth radio layer corresponds to the physical layer of OSI model.

2. It deals with ratio transmission and modulation.
3. The radio layer moves data from master to slave or vice versa.
4. It is a low power system that uses 2.4 GHz ISM band in a range of 10 meters.
5. Bluetooth uses the Frequency Hopping Spread Spectrum (FHSS) method in the physical layer to avoid interference from other devices

Baseband Layer

a. Baseband layer is equivalent to the MAC sublayer in LANs.
b. Bluetooth uses a form of TDMA called TDD-TDMA (Time Division Duplex TDMA).
c. Master and slave stations communicate with each other using time slots.
d. In TDD- TDMA, communication is half duplex in which receiver can send and receive data but not at the same time.
e. In Baseband layer, two types of links can be created between a master and slave. These are:

1. Asynchronous Connection-less (ACL)

It is used for packet switched data that is available at irregular intervals. ACL delivers traffic on a best effort basis. Frames can be lost & may have to be retransmitted. A slave can have only one ACL link to its master. Thus ACL link is used where correct delivery is preferred over fast delivery. The ACL can achieve a maximum data rate of 721 kbps by using one, three or more slots.

2. Synchronous Connection Oriented (SCO)

SCO is used for real time data such as sound. It is used where fast delivery is preferred over accurate delivery. In an SCO link, a physical link is created between the master and slave by
reserving specific slots at regular intervals. Damaged packet are not retransmitted over SCO links. A slave can have three SCO links with the master and can send data at 64 Kbps.

Logical Link, Control Adaptation Protocol Layer (L2CAP)

The logical unit link control adaptation protocol is equivalent to logical link control sublayer of LAN. The ACL link uses L2CAP for data exchange but SCO channel does not use it. The various function of L2CAP is:

1. Segmentation and reassembly

L2CAP receives the packets of upto 64 KB from upper layers and divides them into frames for transmission. It adds extra information to define the location of frame in the original
packet. The L2CAP reassembles the frame into packets again at the destination.

2. Multiplexing

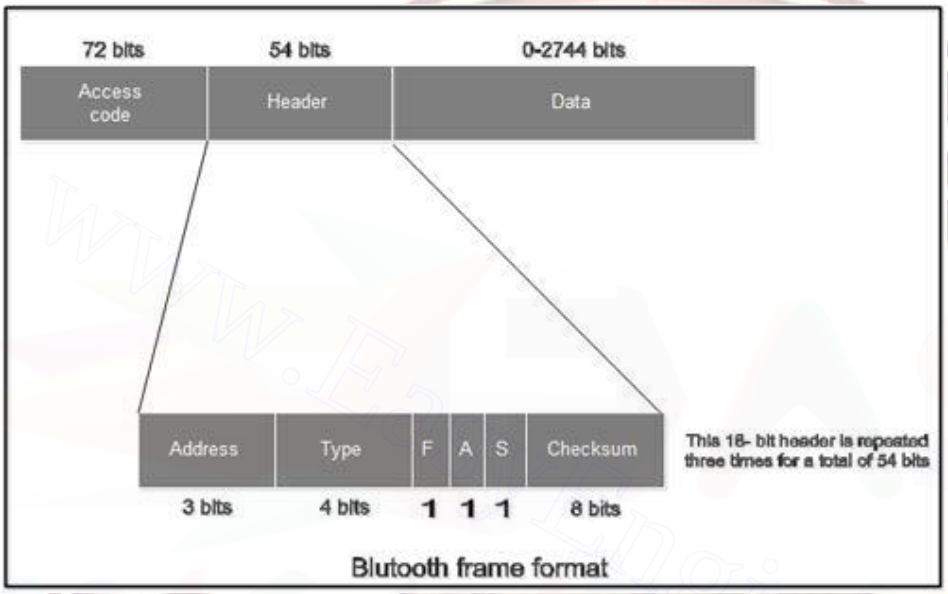
L2CAP performs multiplexing at sender side and demultiplexing at receiver side. At the sender site, it accepts data from one of the upper layer protocol frames and deliver them to the Baseband layer. At the receiver site, it accepts a frame from the baseband layer, extracts the data, and delivers them to the appropriate protocol layer.

3. Quality of Service (QOS)

L2CAP handles quality of service requirements, both when links are established and during normal operation. It also enables the devices to negotiate the maximum payload size during connection establishment.

BLUETOOTH FRAME FORMAT

The various fields of blue tooth frame format are:



Frame format representation of Bluetooth

1. Access Code: It is 72 bit field that contains synchronization bits. It identifies the master.

2. Header: This is 54-bit field. It contain 18 bit pattern that is repeated for 3 time.

The header field contains following subfields:

(i) Address: This is 3 bit field and can define upto seven slaves (1 to 7). If the address is zero, it is used for broadcast communication from primary to all secondaries.

(ii)Type: This 4 bit field identifies the type of data coming from upper layers.

(iii) F: This flow bit is used for flow control. When set to 1, it means the device is unable to receive more frames.

(iv) A: This bit is used for acknowledgment.

(v) S: This bit contains a sequence number of the frame to detect retransmission. As stop and wait protocol is used, one bit is sufficient.

(vi) Checksum: This 8 bit field contains checksum to detect errors in header.

Data: This field can be 0 to 2744 bits long. It contains data or control information coming from upper layers.

3.2 Internetwork Protocol (IP)

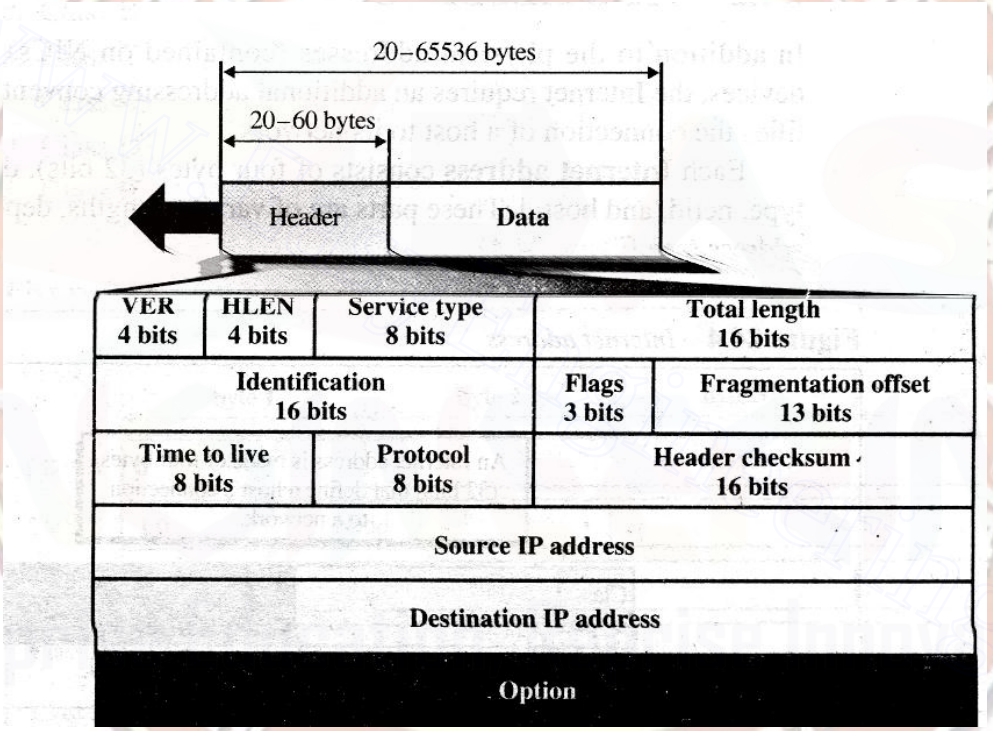
IP is the transmission mechanism used by the TCP/IP protocols. It is an **unreliable and connectionless datagram protocol—a best-effort delivery service.**

Noise can cause bit errors during transmission across a medium; a congested router may discard a datagram if it is unable to relay it before a time limit runs out.

Datagram

Packets in the IP layer are called **datagrams.** A datagram is a variable-length packet (up to 65,536 bytes).

IPv4 Datagram Format



• **Version**: The first field defines the version number of the IP. The current version is 4 (IPv4), with a binary value of 0100.

• **Header length (HLEN)**: The HLEN field defines the length of the header in multiples of four bytes. The four bits can represent a number between 0 and 15, when multiplied by 4, gives a maximum of 60 bytes.

• **Service type**: The service type field defines how the datagram should be handled. It includes bits that define the priority of the datagram. It also contains bits specify the type of service the sender desires such as the level of throughput, ability, and delay.

• **Total length**: The total length field defines the total length of the IP datagram a two-byte field (16 bits) and can define up to 65,535 bytes.

• **Identification**: The identification field is used in fragmentation. A datagram when passing through different networks may be divided into fragments to match the network frame size. When this happens, fragment is identified by sequence number in this field.

• **Flags**: The bits in the flags field deal with fragmentation (the datagram can or can not be fragmented; can be the first, middle, or last fragment; etc.).

• **Fragmentation offset:** The fragmentation offset is a pointer that shows the offset of the data in the original datagram (if it is fragmented).

• **Time to live:** The time-to-live field defines the number of hops a datagram can travel before it is discarded.

The source host, when it creates the datagram, sets this field to an initial value; Then, as the datagram travels through the Internet, router by router, each router decrements this value by 1.

If this value becomes 0 before the datagram reaches its final destination, the datagram is discarded. This prevents a datagram from going back and forth forever between routers.

• **Protocol:** The protocol field defines which upper-layer protocol data are encapsulated in the datagram (TCP, UDP, ICMP, etc.).

• **Header checksum**: This is a 16-bit field used to check the integrity of the header, not the rest of the packet.

• **Source address**: The source address field is a four-byte (32-bit) Internet address. It identifies the original source of the datagram.

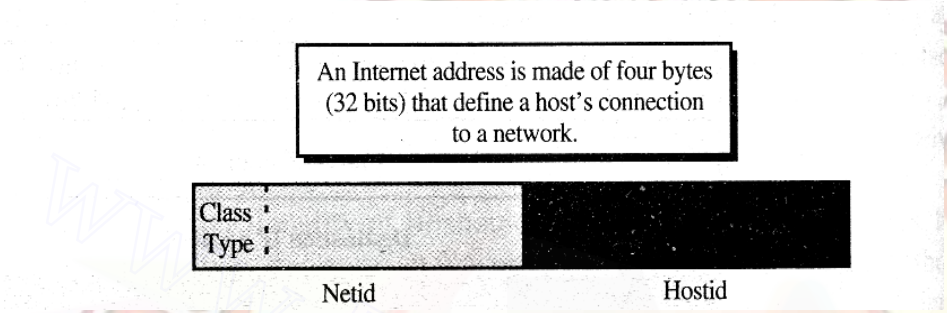
• **Destination address**: The destination address field is a four-byte (32-bit) Internet address. It identifies the final destination of the datagram.

• **Options**: The option field gives more functionality to the IP datagram. It can carry fields that control routing, timing, management, and alignment.

ADDRESSING

In addition to the physical addresses (contained on NICs) that identify individual devices, the Internet requires an additional addressing convention: an address that identifies the connection of a host to its network.

Each Internet address **(IP Address)** consists of four bytes (32 bits), defining three fields: Class type, netid, and hostid. These parts are of varying lengths, depending on the class address.

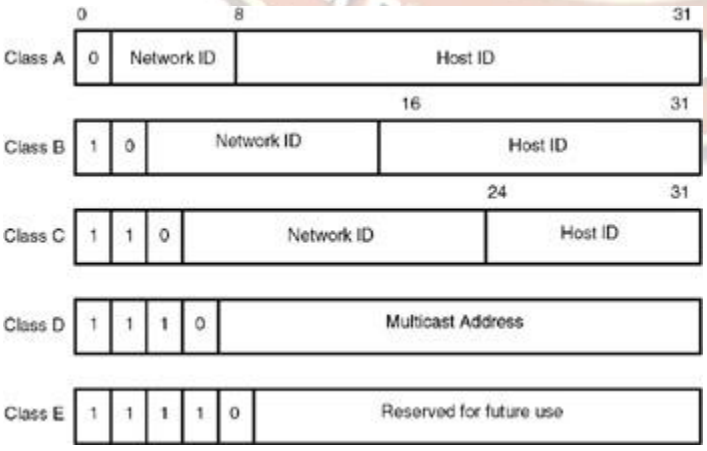


Classes

There are currently five different field-length patterns in use, each defining a class address. The different classes are designed to cover the needs of different type organizations.

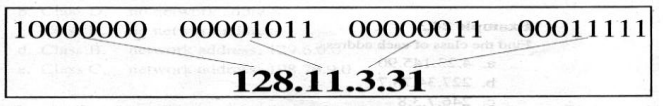
IP Address in Binary notation:

	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10			
Class C	110			
Class D	1110			
Class E	1111			



Dotted-Decimal Notation

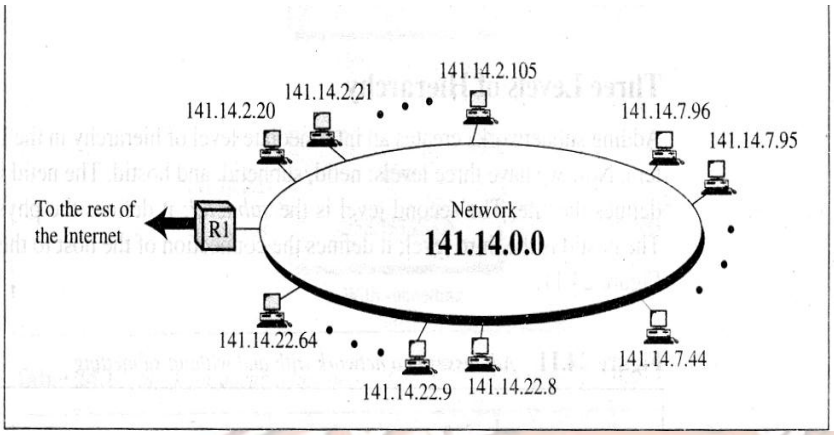
To make the 32-bit form shorter and easier to read, Internet addresses are usually written in decimal form with decimal points separating the bytes dotted-decimal notation.



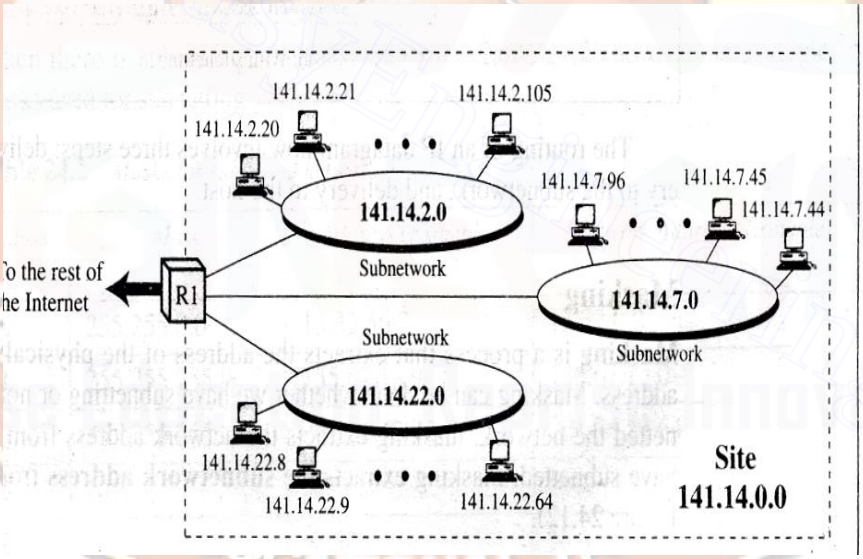
	First byte	Second byte	Third byte	Fourth byte
Class A	0 to 127			
Class B	128 to 191			
Class C	192 to 223			
Class D	224 to 239			
Class E	240 to 255			

SUBNETTING

- An IP address is 32 bits long. One portion of the address indicates a network (netid), and the other portion indicates the host (or router) on the network (hostid).
- This means that there is a sense of hierarchy in IP addressing. To reach a host on the Internet, we n first reach the network using the first portion of the address (netid).
- Then we must reach the host itself using the second portion (hostid). in other words, c1as A, B,-and C in IP addressing are designed with two levels of hierarchy.
- However, in many cases, these two levels of hierarchy are not enough For example,
- Imagine an organization with a class B address. The organization has two-level hierarchical addressing, but it cannot have more than one physical network



- With this scheme, the organization is limited to two levels of hierarchy. The hosts cannot be organized into groups, and all of the hosts are at the same level. The organization has one network with many hosts.
- One solution to this problem is sub netting, the further division of a network into smaller networks called sub networks. The following figure divided into three sub networks.



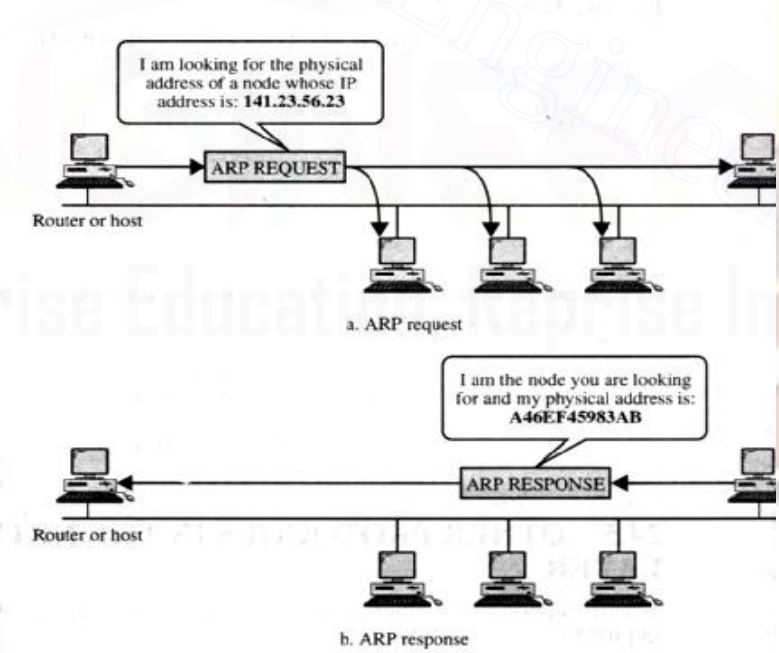
In this example, the rest of the Internet is not aware that the network is divided into three physical subnet works: the three sub networks still appear as a single network to the rest of the Internet. A packet destined for host 141.14.2.21 still reaches router R1. The destination address of the IP datagram is still a class B address where 141.14 defines the netid and 2.21 defines the hostid.

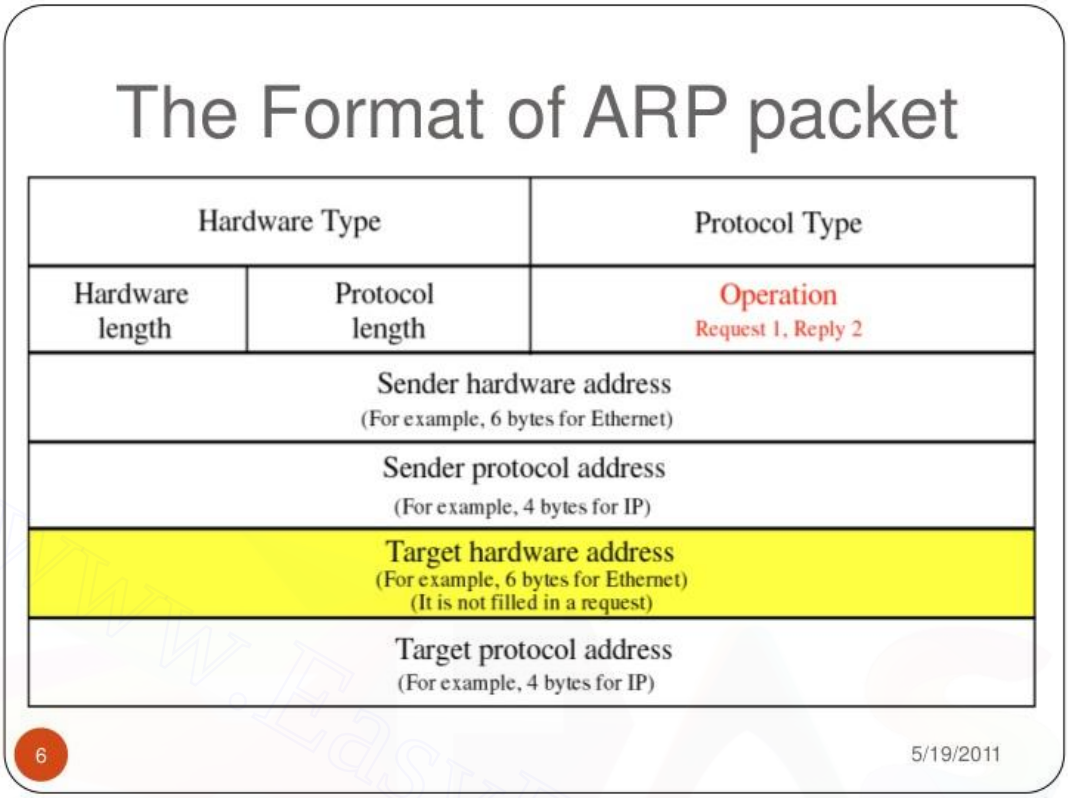
OTHER PROTOCOLS IN THE NETWORK LAYER

TCP/IP supports four other protocols in the network layer: ARP, RARP, ICMP, and IGMP.

ADDRESS RESOLUTION PROTOCOL (ARP)

- The address resolution protocol (ARP) associates an IP address with the physical address. On a typical physical network, such as a LAN, each device on a link is identified by a physical or station address usually imprinted on the network interface card (NIC).
- Physical addresses have local jurisdiction and can be changed easily. For ex ample, if the NIC on a particular machine fails the physical address changes.
- ARP is used to find the physical address of the node when its Internet address is known.
- Anytime a host, or a router, needs to find the physical address of another host network, it formats an ARP query packet that includes the IP address and broadcast over the network .Every host on the network receives and processes the ARP packet, but only the intended recipient recognizes its internet address sends back its physical address.
- The host holding the datagram adds the address target host both to its cache memory and to the datagram header, then sends the datagram on its way .





REVERSE ADDRESS RESOLUTION PROTOCOL (RARP)

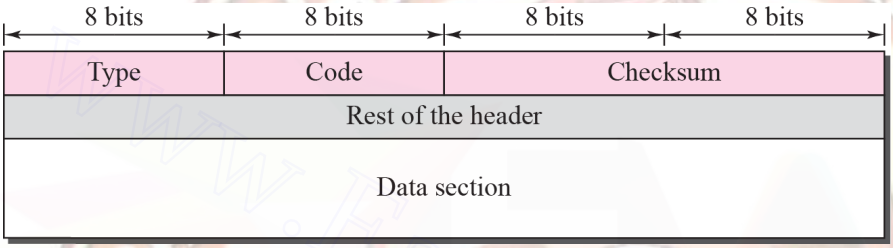
- The reverse address resolution protocol (RARP) allows a host to discover it's in address when it knows only its physical address. The question here is why do we RARP? A host is supposed to have its internet address stored on its hard disk!
- Answer; True, true. But what if the host is a diskless computer? Or what computer is being connected to the network for the first time (when it is being booted) or what if you get a new computer but decide to keep the old NIC?
- RARP works much like ARP. The host wishing to retrieve its internet address broadcasts an RARP query packet that contains its physical address to every host.

INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

- The internet control message protocol (ICMP) is a mechanism used by hosts and routers to send notification of datagram problems back to the sender
- As we saw above, IP is essentially an unreliable and connectionless protocol. ICMP, however, allows IP to inform a sender if a datagram is undeliverable. A datagram travels from router to router until it reaches one that can deliver it to its final destination.

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

GENERAL FORMAT OF ICMP MESSAGES



Note: ICMP always reports error messages to the original source.

Category	Type	Message
Error-reporting Message	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query Message	8 or 0	Echo request or reply
	13 or 14	Timestamp request and reply
	17 or 18	Address mask request and reply
	10 or 9	Router solicitation and advertisement

Error Reporting Messages:

Destination-unreachable messages : When the router cannot route the datagram to the destination due to network failure, this message is sent to the source node.

Source Quench: A source-quench message informs the source that a datagram has been discarded due to congestion in a router or the destination host. The source must slow down the sending of datagrams until the congestion is relieved.

Time Exceeded: Whenever a router decrements a datagram with a time-to-live value to zero, it discards the datagram and sends a time-exceeded message to the original source.
Parameter Problem: A parameter-problem message can be created by a router or the destination host whenever a parameter missing problem occurs in the datagram.
Redirection: Whenever a datagram is forwarded to the wrong router, a redirection message is sent from a router to the source.

Query Messages
Echo Request and Reply: Echo-request and echo-reply messages can test the reachability of a host. This is usually done by invoking the ping command**.**
Timestamp Request and Reply: Timestamp-request and timestamp-reply messages can be used to calculate the round-trip time between a source and a destination machine.
Address Mask Request and Reply: Allows a host to determine the subnet mask in use on the local network.
Router Solicitation and Advertisement: It enables a node to discover the existence of neighboring routers on the same link.

INTERNET GROUP MESSAGE PROTOCOL (IGMP)

- The IF protocol can be involved in two types of communication: unicasting and multi-casting.
- Unicasting is the communication between one sender and one receiver. It is a one-to-one communication.
- However, some processes sometimes need to send the same message to a large number of receivers simultaneously. This is called multicasting, which is a one-to-many communication.
- IP addressing supports multicasting. All 32-bit addresses that start with 1110 (class 0) are multicast addresses. With 28 bits remaining for the group address, more than 250 million addresses are available for assignment. Some of these addresses are permanently assigned.
- The internet group message protocol (IGMP) has been designed to help a multi- cast router identify the hosts in a LAN that are members of a multicast group.

DYNAMIC HOST CONFIGURATION PROTOCOL (DHCP)

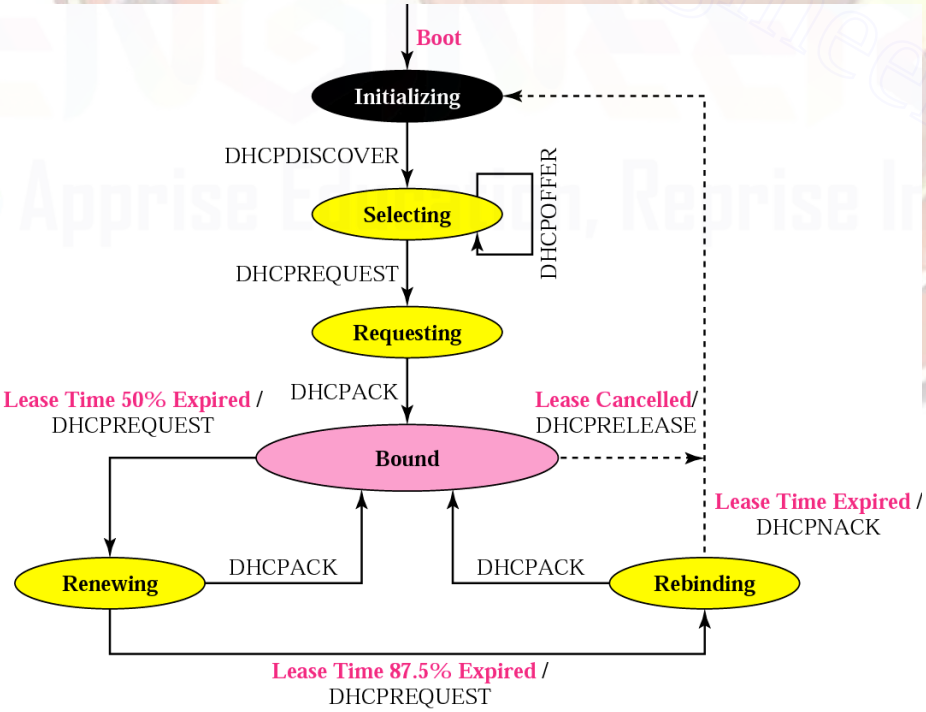
- DHCP provides a temporary IP address for a limited period of time.
- DHCP has two databases.
  First one has static bindings for physical addresses (MAC) with IP addresses.
  Second one has a list of available IP addresses that may be assigned for a period of time.

- Client request to DHCP server causes server to see if MAC is in static database. If so assign the static IP entry to client. If not, choose from available pool.
- Assigned addresses are temporary (leased).
- When client's lease expires, must renew or stop using.

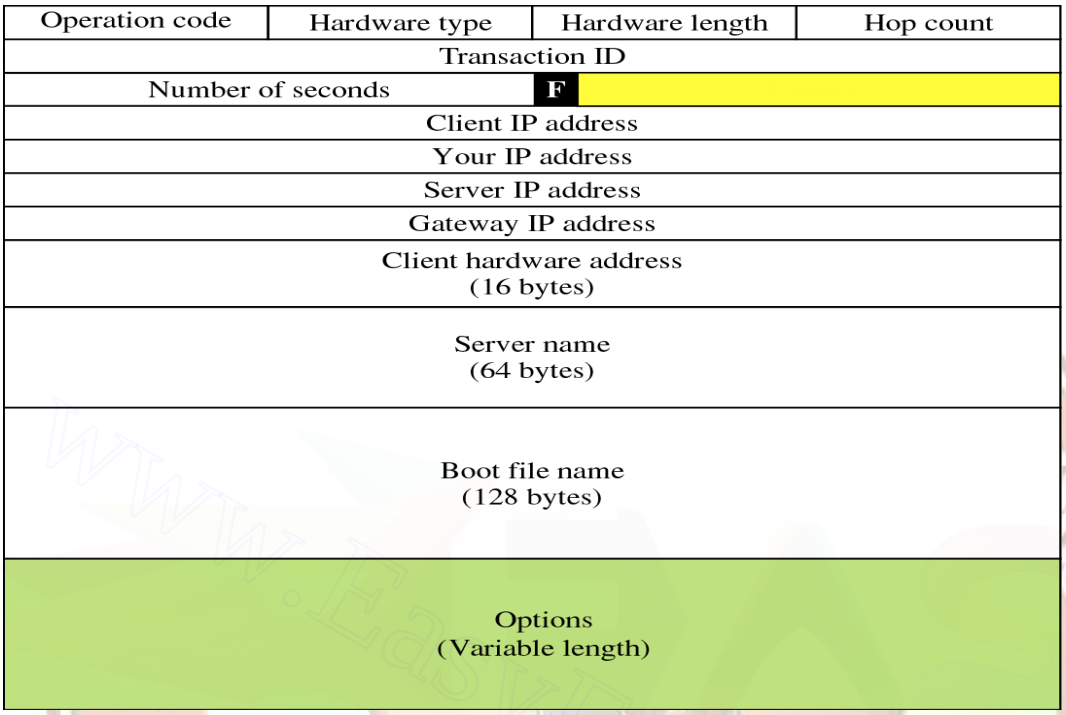
DHCP OPERATION

1. Client broadcasts on 255.255.255.255 a DHCPDISCOVER message using destination server port 67.
2. Server(s) respond with DHCPOFFER message. Contains IP address, duration of lease which by default is one hour.If client does not receive a DHCPOFFER, attempts again up to 4 more attempts in two second intervals, then waits 5 minutes to try again.
3. Client chooses one of the offers and sends DHCPREQUEST to the selected server.
4. Server responds with DHCPACK and creates a binding between MAC address and the IP address offered. Client has rights to that IP address until lease expires.
5. At the 50% of lease period expiration time, client sends a DHCPREQUEST to request renewal.
6. If server responds with DHCPACK, client is good to go and resets client timer. If server denies request with DHCPNACK, client must immediately stop using that IP address and try to find another server.
7. If no server responds with anything in step 6, client sends another DHCPREQUEST at 87.5% time of the original lease.
8. If no server response, client uses IP until lease time expires and then starts from scratch. Client sends DHCPRELEASE message to the mean server. DHCP transition diagram

DHCP TRANSITION DIAGRAM



DHCP PACKET



Operation Code: One byte field defines type of DHCP packet: Request = 1, Reply = 2
Hardware Type: One byte field defining physical network: Ethernet = 1
Hardware Length: One byte field specifying length of physical address: Ethernet = 6
Hop Count: One byte field maximum hops packet can go. Client sets this to 0
Transaction ID: Four Byte field used by client to make sure server is talking to this client and not another simultaneous request's response
Number of seconds: two byte field number of seconds since client became alive
Flag: One bit flag allows client to force server to broadcast reply instead of sending reply to a specific IP address. If client does not know its IP address yet, it wants a broadcast reply from server.
Client IP address: Four byte field of client's IP address. If unknown is zero.
Your IP address: Four byte field server fills in to tell client the clients IP address
Server IP address: four byte field. Server responding fills in it's own IP
Gateway IP Address: Four byte field containing IP address of router (filled in by server)
Client Hardware Address: In our case 6 byte Ethernet MAC of client sending. Can get this from Ethernet frame source MAC but this makes life easy for lazy server
Server Name: Optional 64 byte field filled in by server contains the domain name of the server
Boot File Name: Optional 128 byte field filled in by server containing full pathname for boot file when legacy BOOTP protocol is being used instead of DHCP. DHCP is backward compatible with BOOTP (Aside: Bootstrap Protocol provides IP address, subnet mask, IP address of a router, IP address of a name server to a diskless computer).

Option: Optional 64 byte field. Options consist of three fields: One byte Tag field, One byte length field for just this particular option, a variable length value field.

Tag	Length	Value

For example:

Tag	**Length**	**Value**	**Meaning**
53	1	1	DHCPDISCOVER
53	1	2	DHCPOFFER
53	1	3	DHCPREQUEST
53	1	4	DHCPDECLINE
53	1	5	DHCPACK
53	1	6	DHCPNACK
53	1	7	DHCPRELEASE

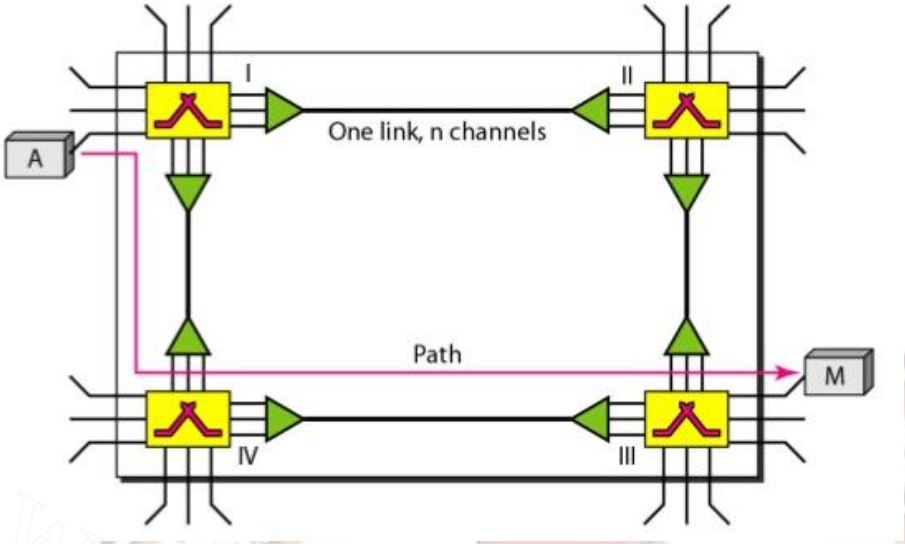
SWITCHING AND BRIDGING

Bridges and switches are data communications devices that operate principally at Layer 2 of the OSI reference model. As such, they are widely referred to as data link layer devices. Bridges became commercially available in the early 1980s. At the time of their introduction, bridges connected and enabled packet forwarding between homogeneous networks. More recently, bridging between different networks has also been defined and standardized. Several kinds of bridging have proven important as internetworking devices. Transparent bridging is found primarily in Ethernet environments, while source-route bridging occurs primarily in Token Ring environments. Translational bridging provides translation between the formats and transit principles of different media types (usually Ethernet and Token Ring). Finally, source-route transparent bridging combines the algorithms of transparent bridging and source-route bridging to enable communication in mixed Ethernet/Token Ring environments. Today, switching technology has emerged as the evolutionary heir to bridging-based inter-networking solutions. Switching implementations now dominate applications in which bridging technologies were implemented in prior network designs. Superior throughput performance, higher port density, lower per-port cost, and greater flexibility have contributed to the emergence of switches as replacement technology for bridges and as complements to routing technology.

SWITCHING TECHNIQUES

1. CIRCUIT-SWITCHED NETWORKS

A circuit-switched network consists of a set of switches connected by physical links. A connection between two stations is a dedicated path made of one or more links. However, each connection uses only one dedicated channel on each link. Each link is normally divided into *n*channels by using FDM or TDM. Figure shows a trivial circuit-switched network with four switches and four links. Each link is divided into *n (n* is 3 in the figure) channels by using FDM or TDM.



A trivial circuit-switched network We have explicitly shown the multiplexing symbols to emphasize the division of the link into channels even though multiplexing can be implicitly included in the switch fabric. The end systems, such as computers or telephones, are directly connected to a switch. We have shown only two end systems for simplicity. When end system A needs to communicate with end system M, system A needs to request a connection to M that must be accepted by all switches as well as by M itself. This is called the setup phase; a circuit (channel) is reserved on each link, and the combination of circuits or channels defines the dedicated path. After the dedicated path made of connected circuits (channels) is established, data transfer can take place. After all data have been transferred, the circuits are tom down. We need to emphasize several points here:

1.Circuit switching takes place at the physical layer.
2.Before starting communication, the stations must make a reservation for the resources to be used during the communication. These resources, such as channels (bandwidth in FDM and time slots in TDM), switch buffers, switch processing time, and switch input/output ports, must remain dedicated during the entire duration of data transfer until the **teardown phase.**
3.Data transferred between the two stations are not packetized (physical layer transfer of the signal). The data are a continuous flow sent by the source station and received by the destination station, although there may be periods of silence.
4.There is no addressing involved during data transfer. The switches route the data based on their occupied band (FDM) or time slot (TDM). Of course, there is end-to end addressing used during the setup phase, as we will see shortly.

Three Phases

The actual communication in a circuit-switched network requires three phases: connection setup, data transfer, and connection teardown.

Setup Phase

Before the two parties can communicate, a dedicated circuit needs to be established. The end systems are normally connected through dedicated lines to the switches, so connection setup means creating dedicated channels between the switches

Data Transfer Phase

After the establishment of the dedicated circuit (channels), the two parties can transfer data.

Teardown Phase

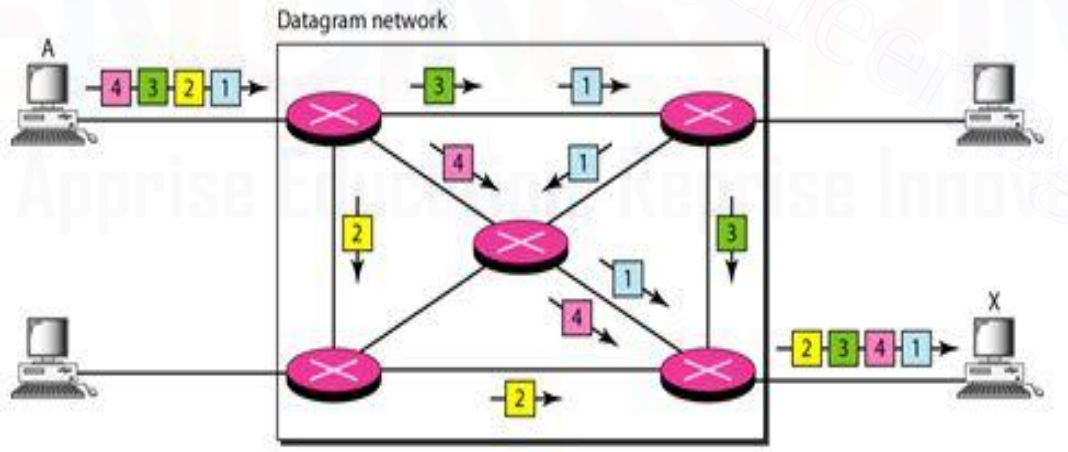
When one of the parties needs to disconnect, a signal is sent to each switch to release the resources.

Circuit-Switched Technology in Telephone Networks

The telephone companies have previously chosen the circuit switched approach to switching in the physical layer; today the tendency is moving toward other switching techniques. For example, the telephone number is used as the global address, and a signaling system (called SS7) is used for the setup and teardown phases.

2.DATAGRAM NETWORKS

In data communications, we need to send messages from one end system to another. If the message is going to pass through a packet-switched network, it needs to be divided into packets of fixed or variable size. The size of the packet is determined by the network and the governing protocol. In packet switching, there is no resource allocation for a packet. Resources are allocated on demand. The allocation is done on a first come, first-served basis. When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed. In a datagram network, each packet is treated independently of all others. Even if a packet is part of a multi packet transmission, the network treats it as though it existed alone. Packets in this approach are referred to as datagrams. Datagram switching is normally done at the network layer. Figure shows how the datagram approach is used to deliver four packets from station A to station X. The switches in a datagram network are traditionally referred to as routers. That is why we use a different symbol for the switches in the figure.

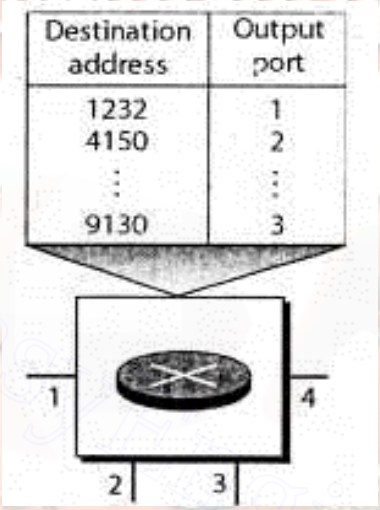


This approach can cause the datagrams of a transmission to arrive at their destination out of order with different delays between the packets. Packets may also be lost or dropped because of a lack of resources. In most protocols, it is the responsibility of an upper-layer protocol to reorder the datagrams or ask for lost datagrams before passing them on to the application. The datagram networks are sometimes referred to as connectionless networks. The term *connectionless* here means that the switch (packet switch) does not keep information about the connection state.

There are no setup or teardown phases. Each packet is treated the same by a switch regardless of its source or destination.

Routing Table

If there are no setup or teardown phases, how are the packets routed to their destinations in a datagram network? In this type of network, each switch (or packet switch) has a routing table which is based on the destination address. The routing tables are dynamic and are updated periodically. The destination addresses and the corresponding forwarding output ports are recorded in the tables. This is different from the table of a circuit switched network in which each entry is created when the setup phase is completed and deleted when the teardown phase is over. Figure shows the routing table for a switch.



Routing table in a datagram network

Destination Address

Every packet in a datagram network carries a header that contains, among other information, the destination address of the packet. When the switch receives the packet, this destination address is examined; the routing table is consulted to find the corresponding port through which the packet should be forwarded.

Datagram Networks in the Internet

The Internet has chosen the datagram approach to switching at the network layer. It uses the universal addresses defined in the network layer to route packets from the source to the destination.

3.VIRTUAL-CIRCUIT NETWORKS

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

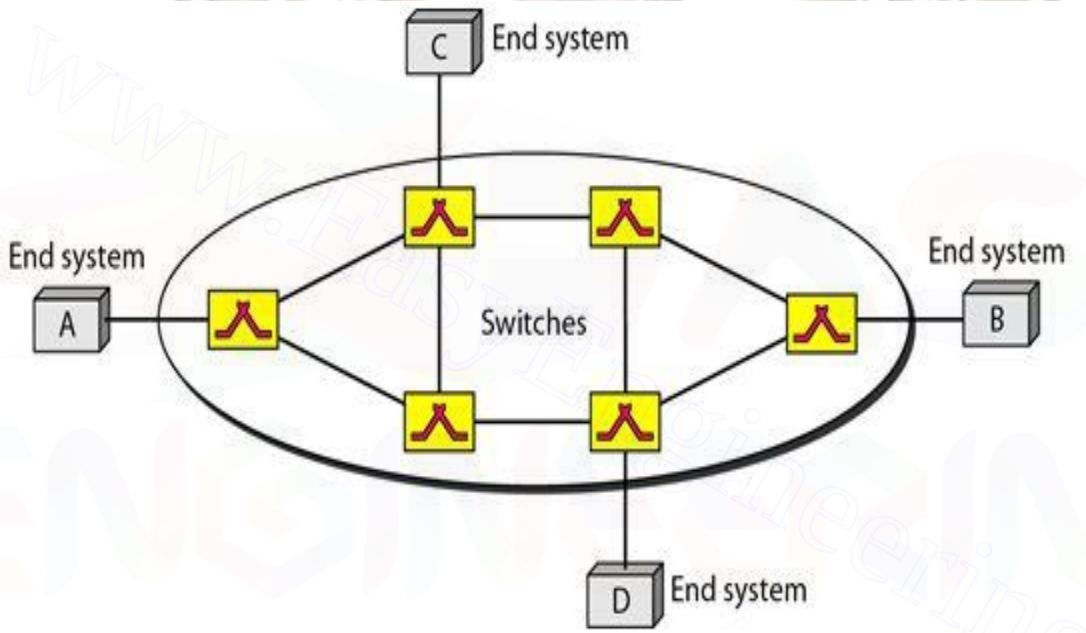
1.As in a circuit-switched network, there are setup and teardown phases in addition to the data transfer phase.

2.Resources can be allocated during the setup phase, as in a circuit-switched network, or on demand, as in a datagram network.

3.As in a datagram network, data are packetized and each packet carries an address in the header. However, the address in the header has local jurisdiction not end-to-end jurisdiction. The reader may ask how the intermediate switches know where to send the packet if there is no final destination address carried by a packet.

4.As in a circuit-switched network, all packets follow the same path established during the connection.

5.A virtual-circuit network is normally implemented in the data link layer, while a circuit-switched network is implemented in the physical layer and a datagram network in the network layer. Figure is an example of a virtual-circuit network. The network has switches that allow traffic from sources to destinations. A source or destination can be a computer, packet switch, bridge, or any other device that connects other networks.



Virtual-circuit network

Circuit-Switched Technology in WANs

Virtual-circuit networks are used in switched WANs such as Frame Relay and ATM networks. The data link layer of these technologies is well suited to the virtual-circuit technology.

Comparision of Communication Switching Network

Circuit Switching	Datagram Packet Switching	Virtual circuit Packet switching
Dedicate transmission path	No dedicate path	No dedicate path
Continuous transmission of data	Transmission of packets	Transmission of packets
Fast enough for interactive	Fast enough for interactive	Fast enough for interactive
Message are not stored	Packets may be stored until delivered	Packets stored until delivered
The path is established for entire conversation	Route established for each packet	Route established for entire conversation
Call setup delay; negligible transmission delay	Packet transmission delay	Call setup delay; Packet transmission delay
Busy signal if called party busy	Sender may be notified if packet not delivered	Sender notified of connection denial
Overload may block call setup; no delay for established calls	Overload increases packet delay	Overload may block call setup; increases packet delay
Electromechanical or computerized switching nodes	Small switching nodes	Small switching nodes
User responsible for message loss protection	Network may be responsible for individual packets	Network may be responsible for packet sequences
Usually no speed or code conversion	Speed and code conversion	Speed and code conversion
Fixed bandwidth	Dynamic use of bandwidth	Dynamic use of bandwidth
No overhead bits after call setup	Overhead bits in each packet	Overhead bits in each packet

BRIDGE

A **bridge** device filters data traffic at a network boundary. Bridges reduce the amount of traffic on a LAN by dividing it into two segments. Bridges operate at the data link layer (Layer 2) of the OSI model. Bridges inspect incoming traffic and decide whether to forward or discard it. An Ethernet bridge, for example, inspects each incoming Ethernet frame - including the source and destination MAC addresses, and sometimes the frame size - in making individual forwarding decisions. Bridges serve a similar function as switches, that also operate at Layer 2. Traditional bridges, though, support one network boundary, whereas switches usually offer four or more hardware ports. Switches are sometimes called “multi-port bridges” for this reason A network

bridge connects multiple network segments at the data link layer (layer 2) of the OSI model. Bridges broadcast to all ports except the port on which the broadcast was received.

However, bridges do not promiscuously copy traffic to all ports, as hubs do, but learn which MAC addresses are reachable through specific ports. Once the bridge associates a port and an address, it will send traffic for that address to that port only. Bridges learn the association of ports and addresses by examining the source address of frames that it sees on various ports. Once a frame arrives through a port, its source address is stored and the bridge assumes that MAC address is associated with that port. The first time that a previously unknown destination address is seen, the bridge will forward the frame to all ports other than the one on which the frame arrived. To select between segments, a bridge must have a look-up that contains the physical addresses of every station connected to it. The table indicates to which segment each station belongs.

Bridges come in three basic types:

1.Simple bridge
2.Multiport bridge
3.Transparent bridge

Simple bridge:

These are the most primitive and least expensive type of bridge. A simple bridge links two segments and contains a table that lists the addresses of all the stations included in each of them. Before a simple can be used, an operator must sit down and enter the addresses of every station: Whenever a new station is added, the table must be modified. If a station is removed, the newly invalid address must be deleted. The logic included in a simple bridge, is of the pass/ no pass variety, a configuration that makes a simple bridge straightforward and inexpensive to manufacture. Installation and maintenance of simple bridges are time consuming and potentially more trouble than the cost savings
are worth.

Multiport Bridge:

A multiport bridge can be used to connect more than two LANs. This type of bridge has three tables, one holding the physical addresses of stations reachable through the corresponding port.

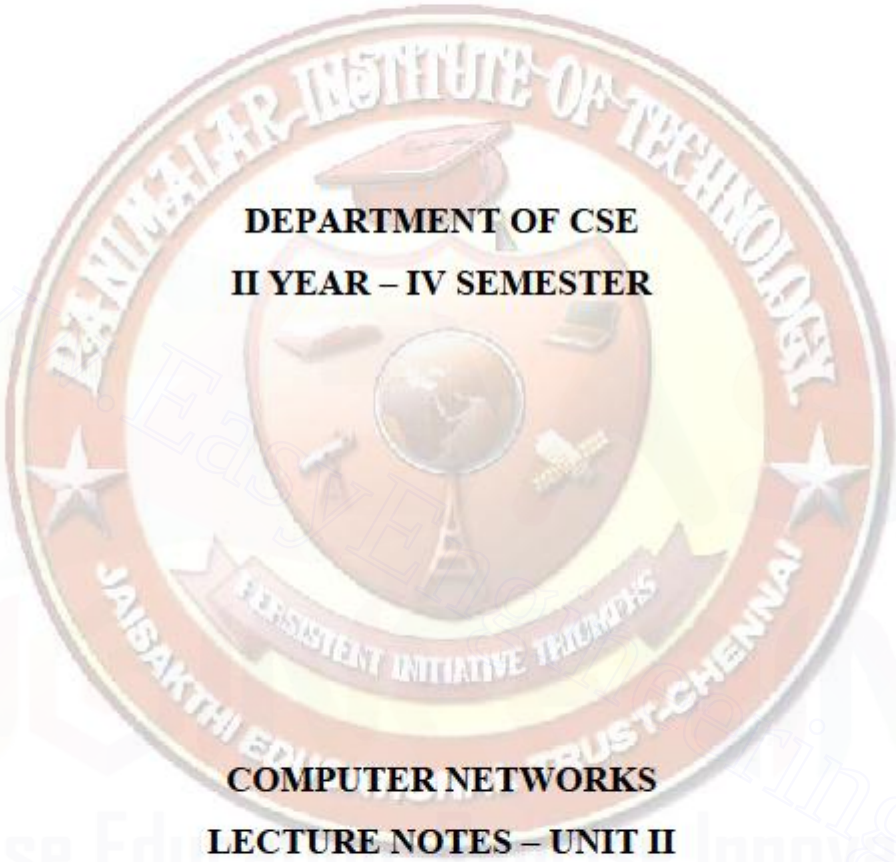
Transparent bridge:

A transparent or learning bridge builds its table of station addresses on its own as it performs its bridge function. When the transparent bridge is first installed, its table is empty. As it encounters each packet, it looks at both the destination and the source addresses. It checks the destination to decide where to send the packet. It does not recognize the destination address, it relays the packet to all of the stations on both segments. It uses the source address to build its table. As it reads the source address, it notes which side the packet came from and associates
that address with the segment to which it belongs.

PANIMALAR INSTITUTE OF TECHNOLOGY

(JAISAKTHI EDUCATIONAL TRUST)

CHENNAI 602 103



DEPARTMENT OF CSE

II YEAR – IV SEMESTER

COMPUTER NETWORKS

LECTURE NOTES – UNIT II

UNIT III

ROUTING

Routing (RIP, OSPF, metrics) – Switch basics – Global Internet (Areas, BGP, IPv6),Multicast – addresses – multicast routing (DVMRP, PIM).

Routing

It is the process of moving packets across a **network** from one host to another. It is usually performed by dedicated devices called **routers**.

Routing Table

- A host or a router has a routing table with an entry for each destination, or a combination of destinations, to route IP packets.
- The routing table can be either static or dynamic.

Static Routing Table

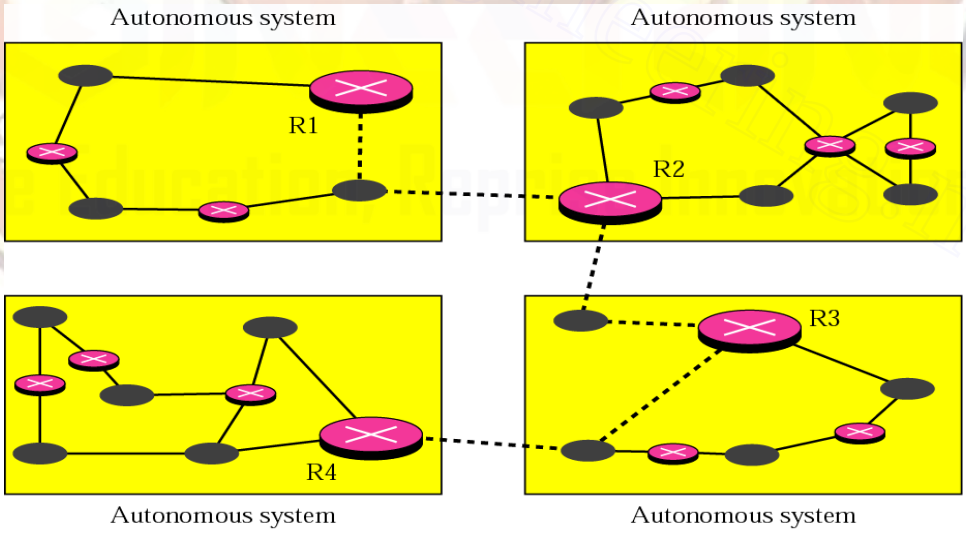
- A **static routing table** contains information entered manually.
- The administrator enters the route for each destination into the table.

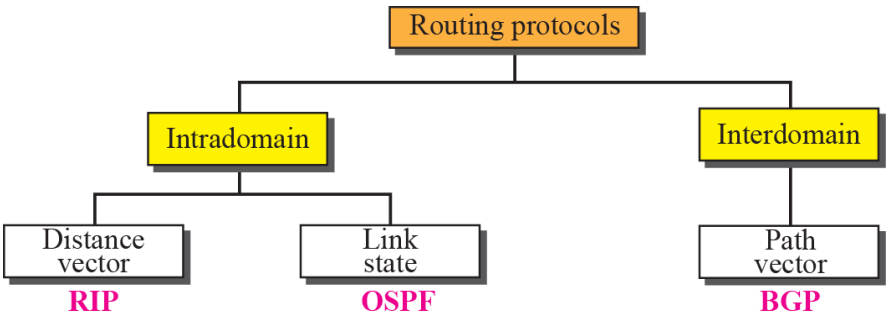
Dynamic Routing Table

- A dynamic routing table is updated periodically by using one of the dynamic routing Protocols such as **RIP, OSPF, or BGP.**
- Whenever there is a change in the Internet, such as a shutdown of a router or breaking of a link, the dynamic routing protocols update all the tables in the routers automatically.

Autonomous System

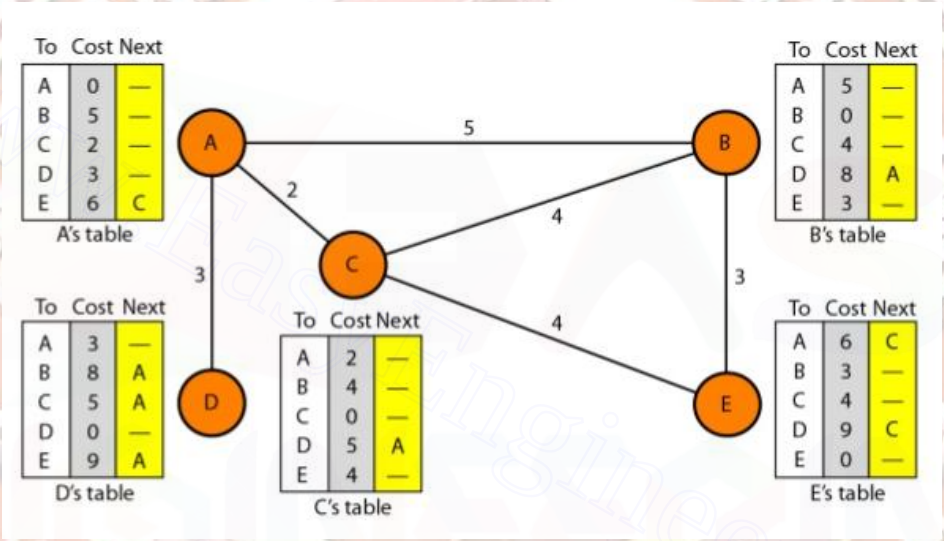
- An autonomous system is a set of networks and routers under the control of a single administrative authority.
- Routing within an autonomous system is **Intradomain routing.**
- Routing between autonomous systems is **Interdomain routing**.





Distance Vector Routing

In distance vector routing, the least-cost route between any two nodes is the route with minimum distance. In this protocol, each node maintains a vector (table) of minimum distances to every node.

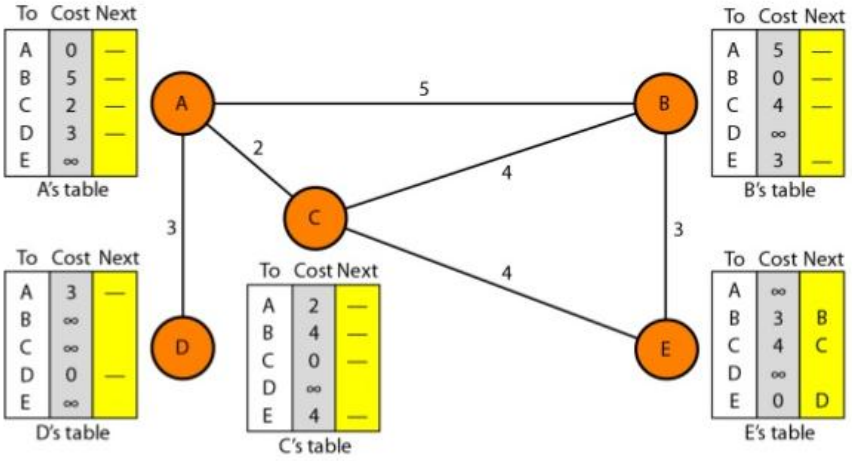


Distance vector routing tables

Initialization

- Each node can know only the distance between itself and its immediate neighbors, those directly connected to it.
- The distance for any entry that is not a neighbour is marked as infinite (unreachable).

Initialization of tables in distance vector routing



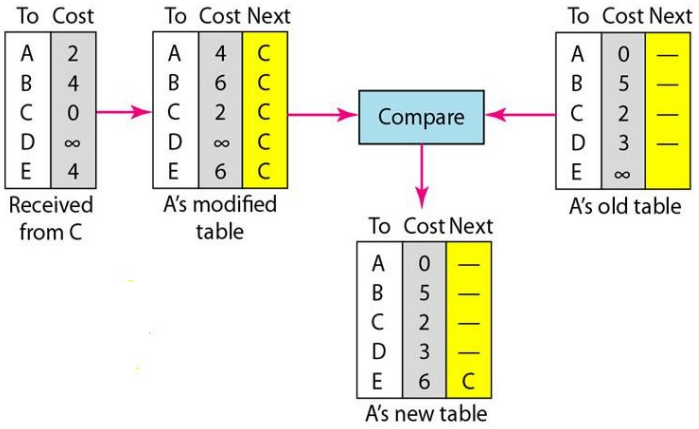
Sharing

- Each node shares its routing table with its immediate neighbours periodically and when there is a change.
- Eg. Node A does not know about node E, but 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.
- A node shares only the first two columns of its routing table to any neighbour.

Updating

When a node receives a two-column table from a neighbour, 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.
2. The receiving node needs to add the name of the sending node to each row as the third column.
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.



Step 1: Add cost (2) to table received from neighbor (C).
Step 2: Compare Modified Table with Old Table (row by row).

a. If Next node entry is different, select the row with the smaller cost. If tie, keep the old one.
b. If Next node entry the same, select the new row value (regard- less of whether new value is smaller or not).

When to Share

Periodic Update :

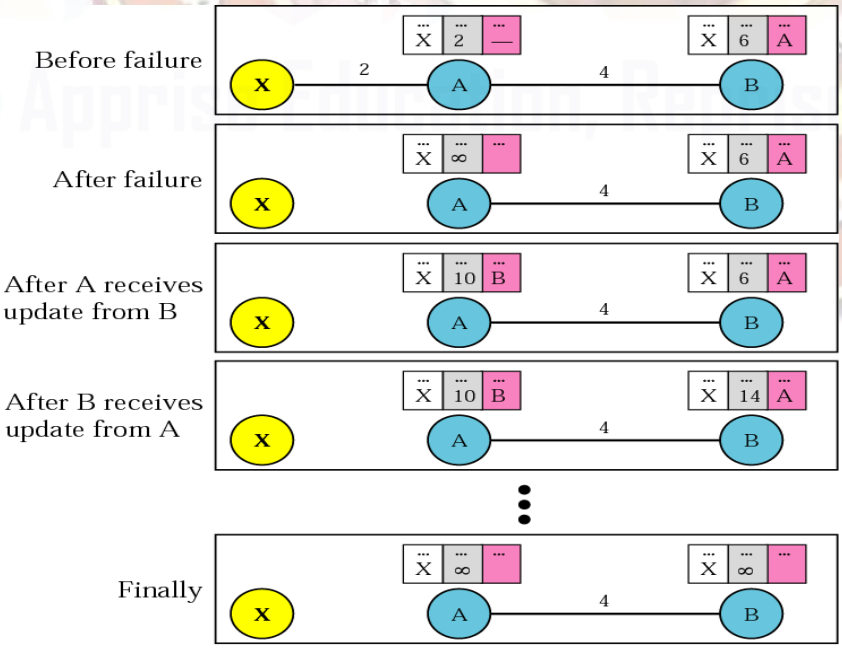
A node sends its routing table, normally every 30 s, in a periodic update. The period depends on the protocol that is using distance vector routing.

Triggered Update:

A node sends its two-column routing table to its neighbours anytime there is a change in its routing table. This is called a triggered update.

Problems in Distance Vector Routing Method:

1. **Two node Instability Problem**



Both A and B know where X is.

Link between A and X fails. A updates its table immediately.

But before A can tell B, B sends its info to A!

A, using B's info, updates its table (error!).

Then A send its table to B and B updates its table (more error).

Both routers keep updating tables, eventually hitting infinity.

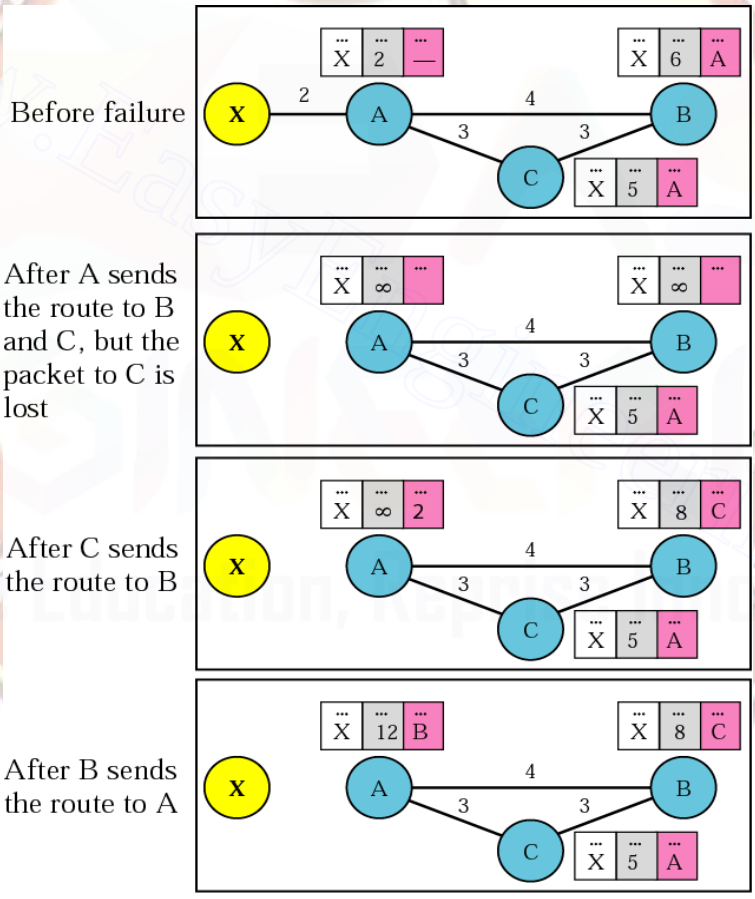
Solutions to two-node instability:

1. Define infinity to be a much smaller value, such as 100. Then it doesn't take too long to become stable. But now you can't use distance vector routing in large networks.

2. Split Horizon – instead of flooding entire table to each node, only part of its table is sent. More precisely, if node B thinks that the optimum router to reach X is via A, then B does not need to advertise this piece of info to A – the info has already come from A.

3.Split Horizon and Poison Reverse – Normally, the distance vector protocol uses a timer. If there is no news about a route, the node deletes the route from its table. So when A never hears from B about the route to X, it deletes it. Instead, Node B still advertises the value for X, but if the source of info is A, it replaces the distance with infinity, saying "Do not use this value; what I know about this route comes from you."

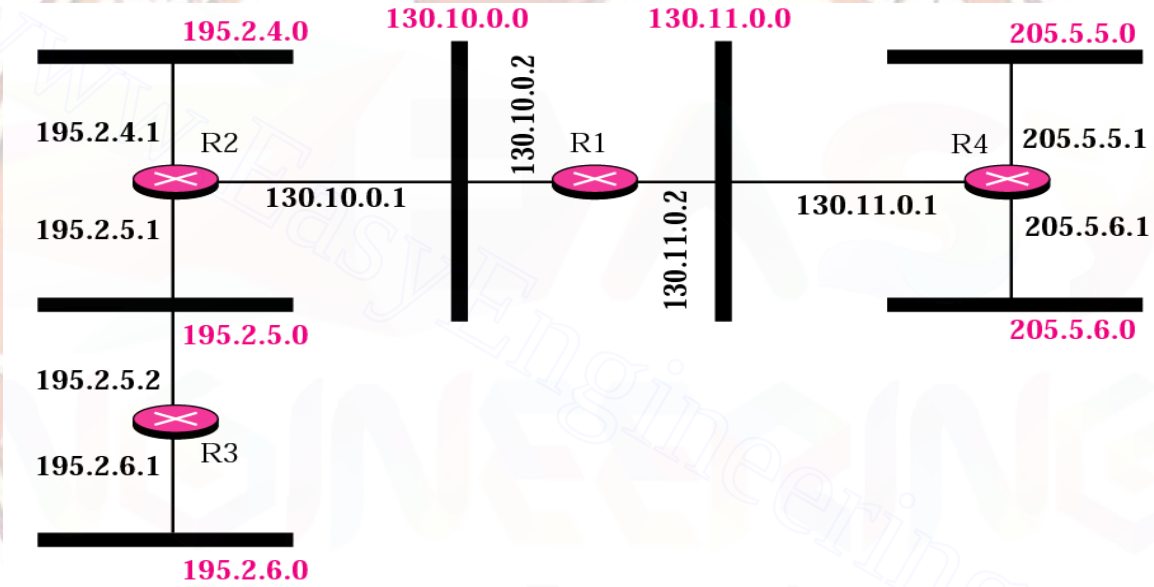
2.Three node Instability Problem



ROUTING INFORMATION PROTOCOL (RIP)

- It is an intra-domain (interior) 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. 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.



R1 Table

Dest.	Hop	Next
130.10.0.0	1	———
130.11.0.0	1	———
195.2.4.0	2	130.10.0.1
195.2.5.0	2	130.10.0.1
195.2.6.0	3	130.10.0.1
205.5.5.0	2	130.11.0.1
205.5.6.0	2	130.11.0.1

R2 Table

Dest.	Hop	Next
130.10.0.0	1	———
130.11.0.0	2	130.10.0.2
195.2.4.0	1	———
195.2.5.0	1	———
195.2.6.0	2	195.2.5.2
205.5.5.0	3	130.10.0.2
205.5.6.0	3	130.10.0.2

R3 Table

Dest.	Hop	Next
130.10.0.0	2	195.2.5.1
130.11.0.0	3	195.2.5.1
195.2.4.0	2	195.2.5.1
195.2.5.0	1	———
195.2.6.0	1	———
205.5.5.0	4	195.2.5.1
205.5.6.0	4	195.2.5.1

R4 Table

Dest.	Hop	Next
130.10.0.0	2	130.11.0.2
130.11.0.0	1	———
195.2.4.0	3	130.11.0.2
195.2.5.0	3	130.11.0.2
195.2.6.0	4	130.11.0.2
205.5.5.0	1	———
205.5.6.0	1	———

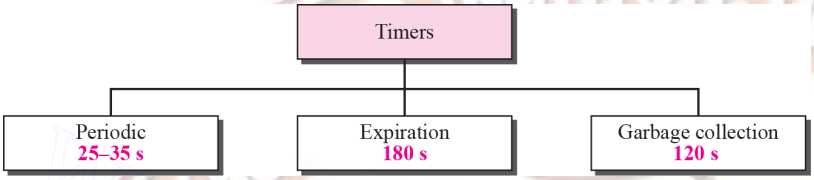
Above figure shows an autonomous system with seven networks and four routers. The table of each router is also shown. Let us look at the routing table for Rl. The table has seven entries to show how to reach each network in the autonomous system. Router Rl is directly connected to networks 130.10.0.0 and 130.11.0.0, which means that there are no next-hop entries for these two networks. To send a packet to one of the three networks at the far left, router Rl needs to deliver the packet to R2. The next-node entry for these three networks is the interface of router R2 with IP address 130.10.0.1. To send a packet to the two networks at the far right, router

Rl needs to send the packet to the interface of router R4 with IP address 130.11.0.1. The other tables can be explained similarly.

RIP messages

- Request
  - A request message is sent by a router that has just come up or by a router that has some time-out entries
  - A request can ask about specific entries or all entries
- Response
  - A response can be either solicited or unsolicited (30s or when there is a change in the routing table)

RIP Timers

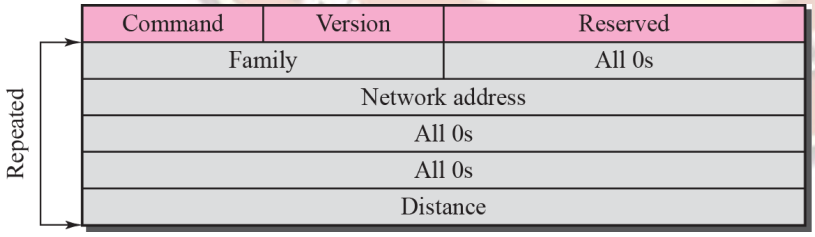


- Periodic timer
  - It controls the advertising of regular update message (25 ~ 30 sec)
- Expiration timer
  - It governs the validity of a route (180 sec)
  - The route is considered expired and the hop count of the route is set to 16
- Garbage collection timer
  - A invalid route is not purged from the routing table until this timer expires (120 sec)

RIPv2 vs. RIPv1:

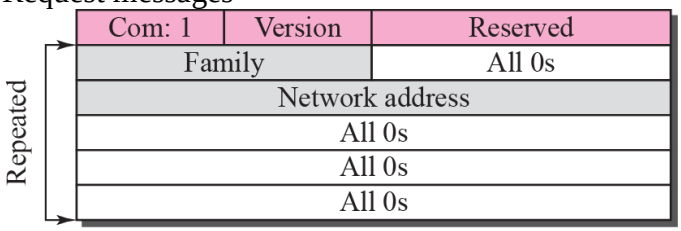
- RIPv1 uses broadcasting to send RIP messages to every neighbors. Routers as well as hosts receive the packets
- RIPv2 uses the all-router multicast address to send the RIP messages only to RIP routers in the network

RIP message format

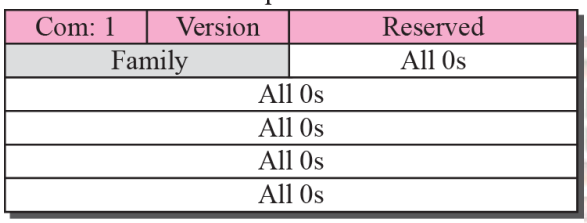


Command: request (1) or response (2)
Version: 1 or 2 (version 2 shown in a couple slides)
Family: TCP/IP has value 2
Network address: address of the destination network
Distance: hop count from the advertising router to the destination network

Request messages

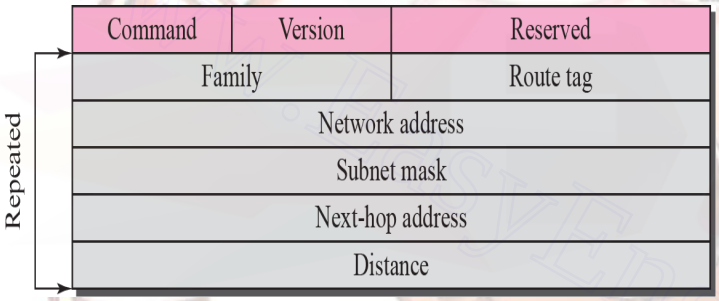


a. Request for some



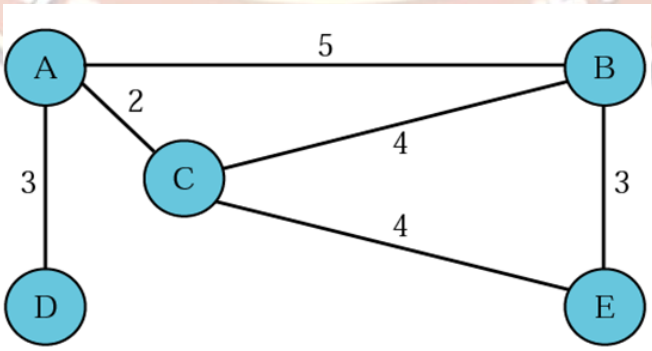
b. Request for all

RIP version 2 format



Link State Routing

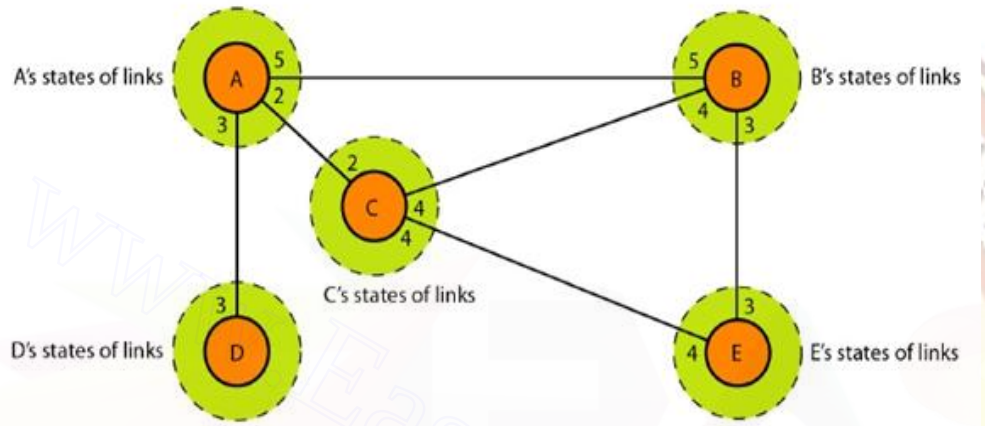
In link state routing, if each node in the domain has the entire topology of the domain such as the list of nodes and links, how they are connected including the type, cost (metric), and condition of the links (up or down). Node can use Dijkstra's algorithm to build a routing table.



Link state knowledge

Each router knows (maintains) its states of its links. Each router floods this info (via a Link State Packet) to other routers periodically (when there is a change in the topology, or every 60 to 120 minutes).

Each router takes in this data and, using Dijkstra's algorithm, creates the shortest path tree and corresponding routing table.



Node A knows that it is connected to node B with metric 5, to node C with metric 2, and to node D with metric 3. Node C knows that it is connected to node A with metric 2, to node B with metric 4, and to node E with metric 4. Node D knows that it is connected only to node A with metric 3. And so on. Although there is an overlap in the knowledge, the overlap guarantees the creation of a common topology-a picture of the whole domain for each node.

Building Routing Tables

Four Operations are involved

1. Creation of the link state packet (LSP).
2. Dissemination of LSPs to every other router, called **flooding,** in an efficient and reliable way.
3. Formation of a shortest path tree for each node using Dijkstras Algorithm.
4. Calculation of a routing table based on the shortest path tree.

1. Creation of Link State Packet (LSP):

A link state packet can carry a large amount of information. For the moment, however, we assume that it carries a minimum amount

LSPs are generated on two occasions:

a. When there is a change in the topology of the domain.

b. On a periodic basis.

Link State Packet (LSP) contains following fields

1. Node Identity
2. List of links
3. Sequence number
4. Time-To-Live (TTL) or Age for this packet

***2.* Flooding of LSPs**

(i). The creating node sends a copy of the LSP out of each interface.

(ii). A node that receives an LSP compares it with the copy it may already have. If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP. If it is newer, the node does the following:

a. It discards the old LSP and keeps the new one.

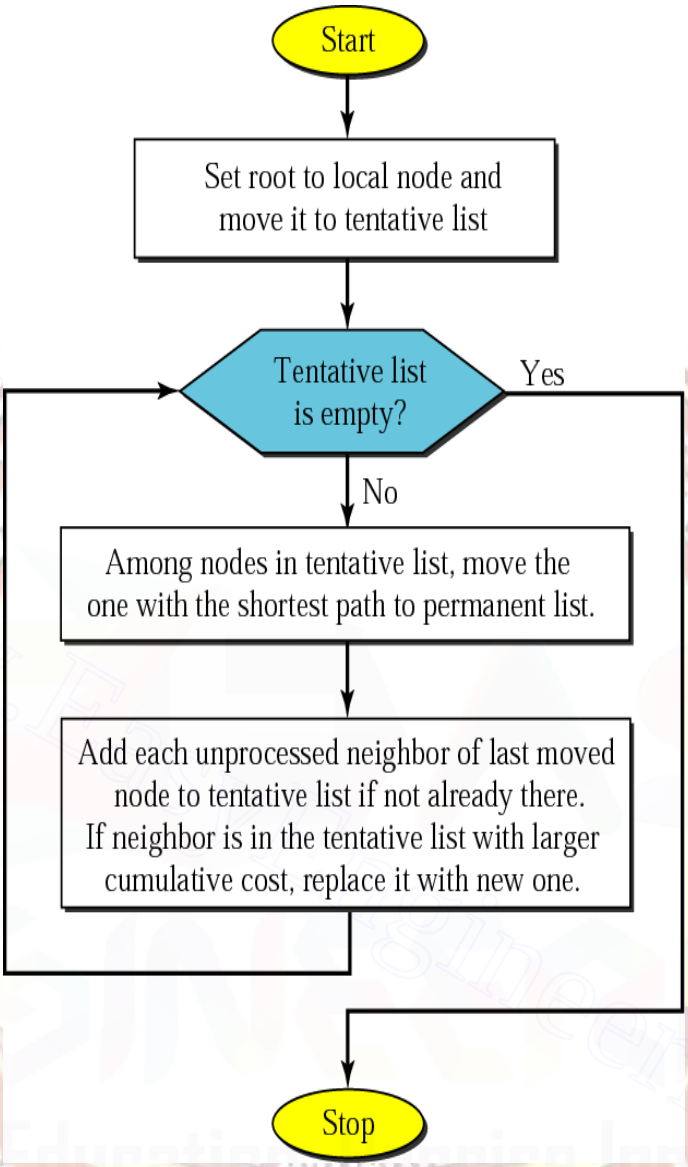
b. It sends a copy of it out of each interface except the one from which the packet arrived.

3. Formation of Shortest Path Tree:

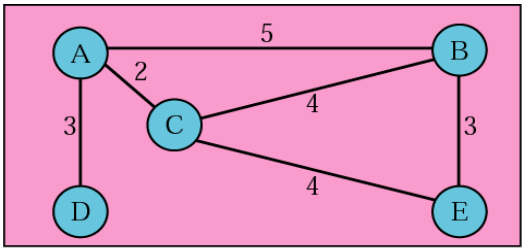
After receiving all LSPs, each node will have a copy of the whole topology. To find the shortest path to every other node; a shortest path tree is needed. A tree is a graph of nodes and links; one node is called the root. All other nodes can be reached from the root through only one single route. A shortest path tree is a tree in which the path between the root and every other node is the shortest.

The Dijkstra algorithm creates a shortest path tree from a graph. The algorithm divides the nodes into two sets: **tentative and permanent**.

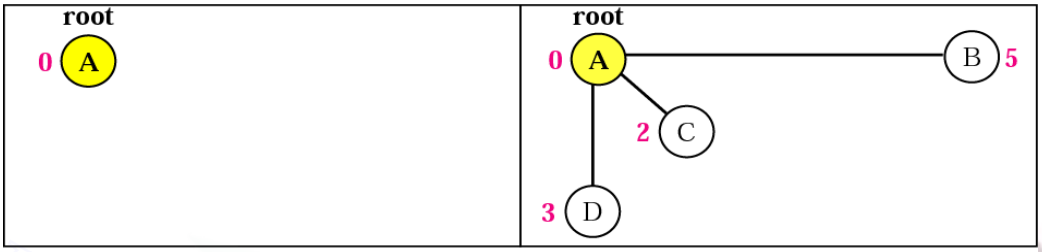
Dijkstra algorithm



Example of formation of shortest path tree

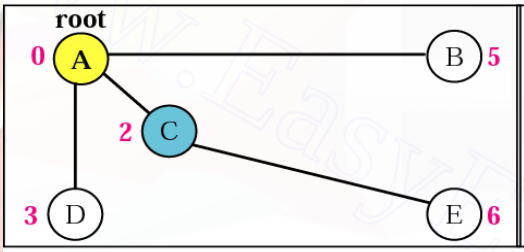


Topology

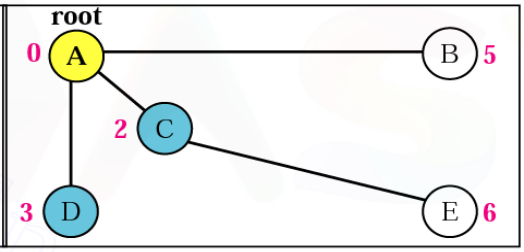


1. Set root to A and move A to tentative list

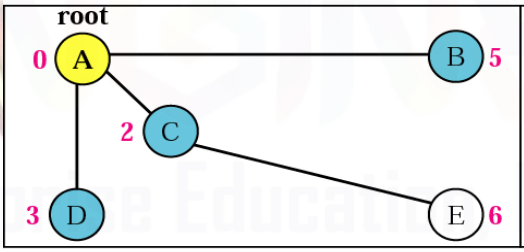
2. Move A to permanent list and add B, C, and D to tentative list



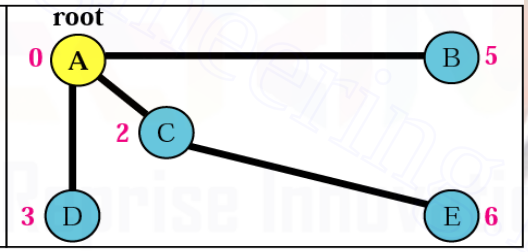
3. Move C to permanent and add E to tentative list



4. Move D to permanent list.



5. Move B to permanent list



6. Move E to permanent list (tentative list is empty)

1. We make node A the root of the tree and move it to the tentative list. Our two lists are
   Permanent list: empty **Tentative list: A(O)**
2. Node A has the shortest cumulative cost from all nodes in the tentative list. We move A to the permanent list and add all neighbors ofA to the tentative list. Our new lists are
   Permanent list: A(O) **Tentative list: B(5), C(2), D(3)**
3. Node C has the shortest cumulative cost from all nodes in the tentative list. We move C to the permanent list. Node C has three neighbors, but node A is already processed, which makes the unprocessed neighbors just B and E. However, B is already in the tentative list with a cumulative cost of 5. Node A could also reach node B through C with a cumulative cost of 6. Since 5 is less than 6, we keep node B with a cumulative cost of 5 in the tentative list and do not replace it. Our new lists are
   Permanent list: A(O), e(2) **Tentative list: B(5), 0(3), E(6)**

4. Node D has the shortest cumulative cost of all the nodes in the tentative list. We move D to the permanent list. Node D has no unprocessed neighbor to be added to the tentative list. Our new lists are

Permanent list: A(O), C(2), 0(3) **Tentative list: B(5), E(6)**

5. Node B has the shortest cumulative cost of all the nodes in the tentative list. We move B to the permanent list. We need to add all unprocessed neighbors of B to the tentative list (this is just node E).

However, E(6) is already in the list with a smaller cumulative cost. The cumulative cost to node E, as

the neighbor of B, is 8. We keep node E(6) in the tentative list. Our new lists are

Permanent list: A(O), B(5), C(2), 0(3) **Tentative list: E(6)**

6. Node E has the shortest cumulative cost from all nodes in the tentative list. We move E to the permanent list. Node E has no neighbor. Now the tentative list is empty. We stop; our shortest path

tree is ready. The final lists are

Permanent list: A(O), B(5), C(2), D(3), E(6) **Tentative list: empty**

4. Calculation of Routing Table from Shortest Path Tree

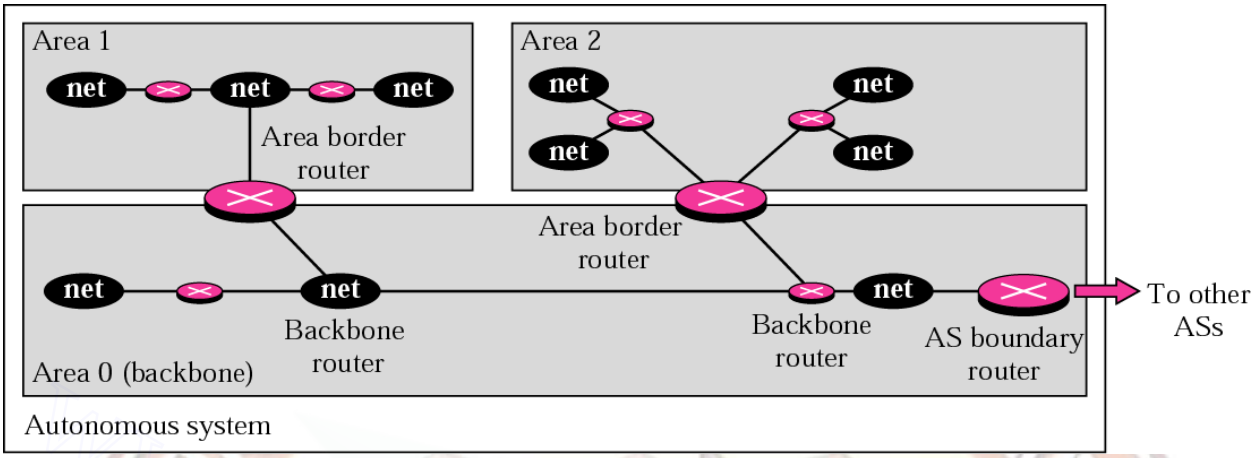
Each node uses the shortest path tree protocol to construct its routing table. The routing table shows the cost of reaching each node from the root.

Routing Table for node A

Node	*Cost*	*Next Router*
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

Open Shortest Path First(OSPF)

The Open Shortest Path First (OSPF) protocol is an Intradomain routing protocol based on link state routing. Its domain is also an autonomous system.



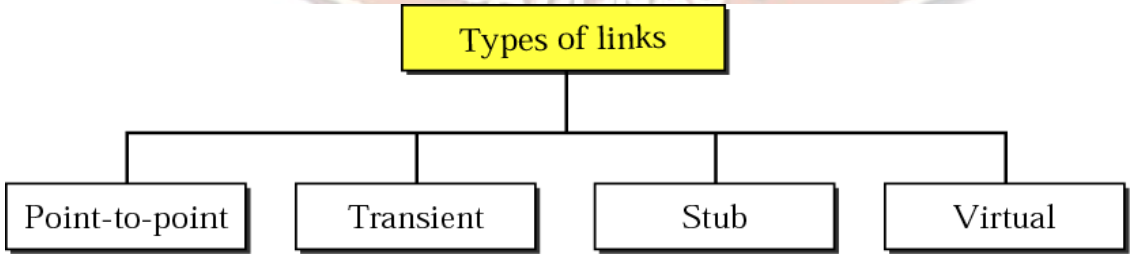
AREA:

- OSPF divides an autonomous system into areas.
- All networks inside an area must be connected.
- Routers inside an area flood the area with routing information.
- **Area border routers**: Summarize the information about the area and send it to other routers.
- **Backbone area [Primary area]:** All the areas inside an autonomous system must be connected to the backbone.
- Routers in the backbone area are called **backbone routers**. This area identification number is 0.
- If due to some problem the connectivity between a backbone and an area is broken, a **virtual link** between routers must be created by the administration to allow continuity of the functions of the backbone as the primary area.

METRIC:

- The **cost** associated with a route is called the metric.
- Metric could be min delay, max throughput, etc.

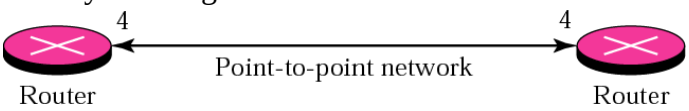
TYPES OF LINKS (CONNECTIONS)



Point-to-Point

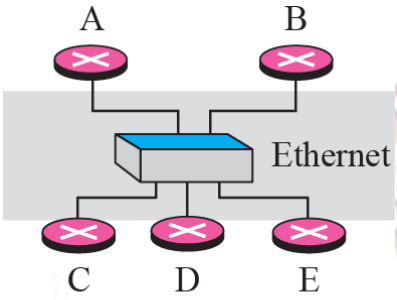
- Connects two routers without any other router or host in between.
- Directly connected routers using serial line.

- Only one neighbour.



Transient link

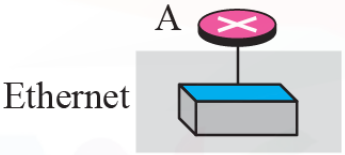
- A network with several routers attached to it.
- Each router has many neighbours.
- Lot of advertisements about their neighbours.



a. Transient network

Stub Link

- A network that is connected to only one router.
- The data packets enter the network through this single router and leave the network through this same router.
- Unidirectional from the router to the network

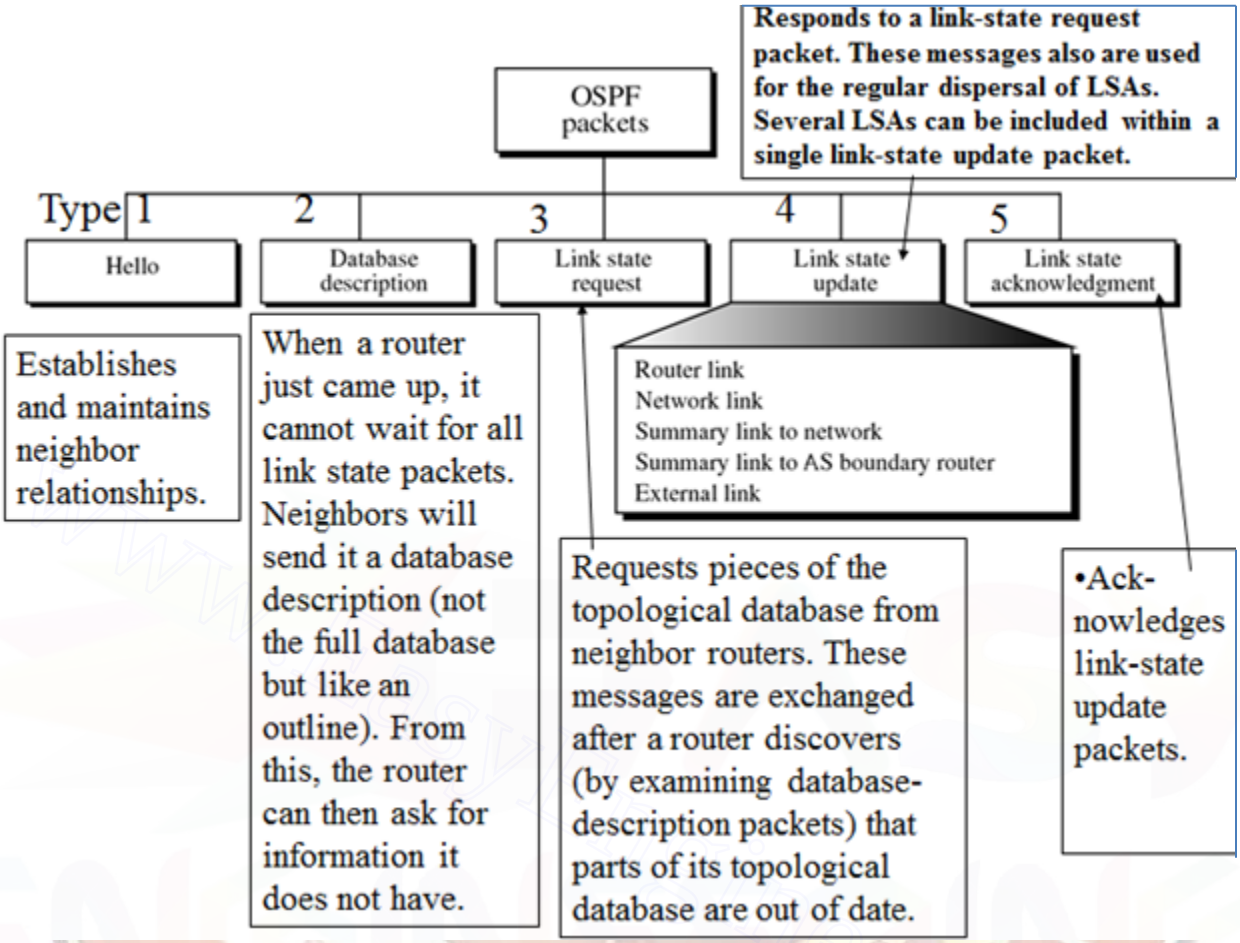


a. Stub network

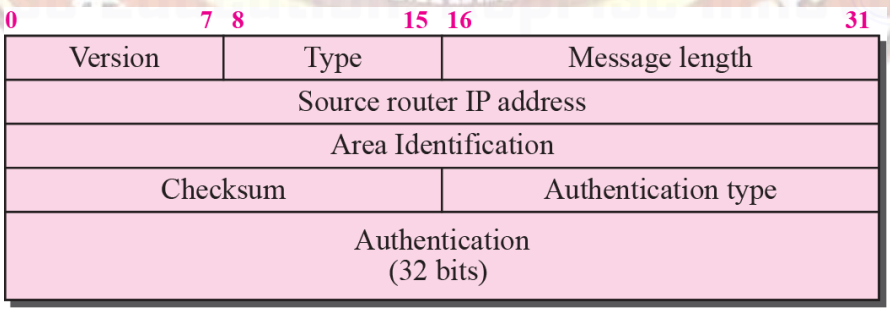
Virtual Link

- When the link between two routers is broken, the administration may create a virtual link between them, using a longer path that probably goes through several routers.

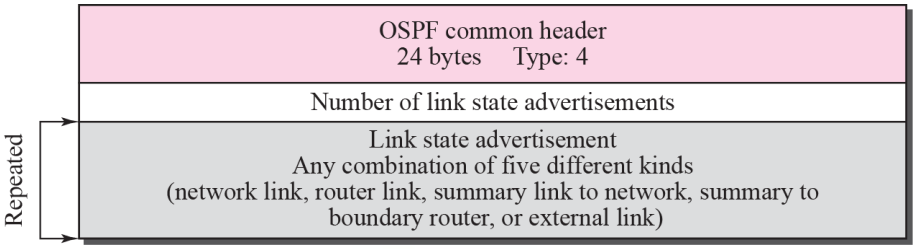
OSPF PACKETS



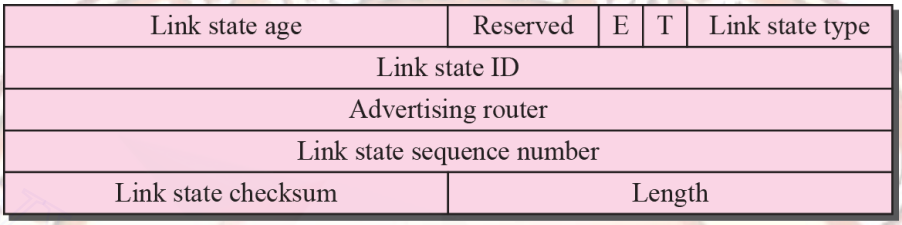
OSPF common header



Link state update packet



LSA general header



- Link state age
  - When a router creates the message, the value of this field is 0
  - When each successive router forwards this message, it estimates the transit time and adds it to the cumulative value of this field
- E flag
  - If this flag is set to 1, it means the area is a stub area (an area that is connected to the backbone area by only one path
- T flag
  - If this flag is set to 1, it means the router can handle multiple types of services
- Advertising router
  - The IP address of the router advertising this message
- Link state sequence number
  - A sequence number assigned to each link state update message LS Type and LS ID

PATH VECTOR ROUTING

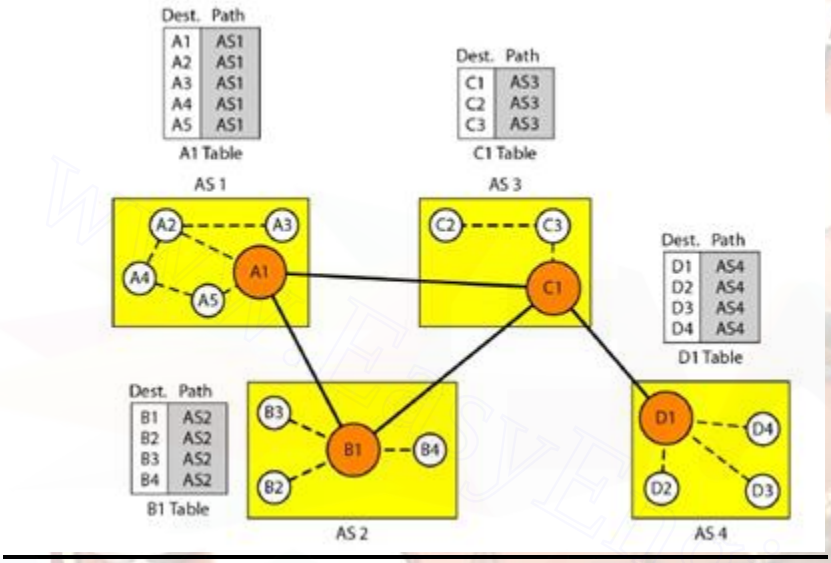
- Path vector routing is useful for Interdomain routing.
- The principle of path vector routing is similar to that of distance vector routing.
- In path vector routing, there is one node in each autonomous system that acts on behalf of the entire autonomous system, called **speaker node**.
- The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring ASs.
- The idea is the same as for distance vector routing except that only speaker nodes in each AS can communicate with each other.
- A speaker node advertises the path, not the metric of the nodes, in its autonomous system or other autonomous systems.

Initialization

At the beginning, each speaker node can know only the reachability of nodes inside its autonomous system. Following Figure shows the initial tables for each speaker node in a system made of four ASs.

- Node Al is the speaker node for ASl, Bl for AS2, Cl for AS3, and Dl for AS4.
- Node Al creates an initial table that shows Al to A5 are located in ASI and can be reached through it.
- Node Bl advertises that Bl to B4 are located in AS2 and can be reached through Bl and so on.

Initial routing tables in path vector routing



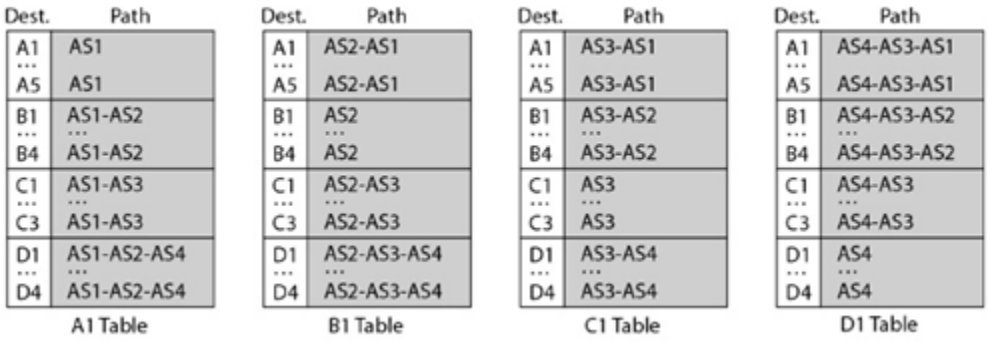
Sharing

- A speaker in an autonomous system shares its table with immediate neighbors.
- Node Al shares its table with nodes Bl and Cl.
- Node Cl shares its table with nodes Dl, Bl, and Al.
- Node Bl shares its table with Cl and Al.
- Node Dl shares its table with Cl.

Updating

- When a speaker node receives a two-column table from a neighbor, it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table.
- After a while each speaker has a table and knows how to reach each node in other ASs.
- If router Al receives a packet for nodes A3, it knows that the path is in ASI (the packet is at home); but if it receives a packet for Dl, it knows that the packet should go from ASl, to AS2, and then to AS3.
- The routing table shows the path completely.
- On the other hand, if node Dl in AS4 receives a packet for node A2, it knows it should go through AS4, AS3, and AS1.

Stabilized tables for three autonomous systems



Dest.	Path
A1 … A5	AS1
B1 … B4	AS1-AS2
C1 … C3	AS1-AS3
D1 … D4	AS1-AS2-AS4

A1 Table

Dest.	Path
A1 … A5	AS2-AS1
B1 … B4	AS2
C1 … C3	AS2-AS3
D1 … D4	AS2-AS3-AS4

B1 Table

Dest.	Path
A1 … A5	AS3-AS1
B1 … B4	AS3-AS2
C1 … C3	AS3
D1 … D4	AS3-AS4

C1 Table

Dest.	Path
A1 … A5	AS4-AS3-AS1
B1 … B4	AS4-AS3-AS2
C1 … C3	AS4-AS3
D1 … D4	AS4

D1 Table

BGP

- Border Gateway Protocol (BGP) is an Inter-domain routing protocol using path vector routing.
- The Internet is divided into hierarchical domains called **autonomous systems.**
- Types **of Autonomous** Systems
    1. Stub AS
    2. Multihomed AS
    3. Transit AS.

Stub AS

- A stub AS has only one connection to another AS.
- The hosts in the AS can send data traffic to other ASs.
- The hosts in the AS can receive data coming from hosts in other ASs.
- Data traffic cannot pass through a stub AS.
- A stub AS is either a source or a sink.
- A good example of a stub AS is a small corporation or a small local ISP.

Multihomed AS

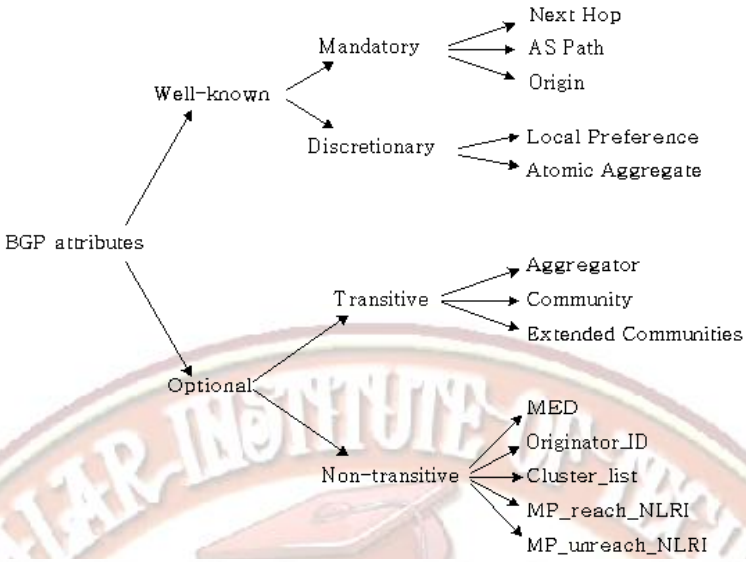
- A multihomed AS has more than one connection to other Ass.
- It can send data traffic to more than one AS.
- It can receive data traffic from more than one AS.
- It does not allow data coming from one AS and going to another AS to pass through.
- A good example of a multihomed AS is a large corporation that is connected to more than one regional or national AS that does not allow transient traffic.

Transit AS

- A transit AS is a multihomed AS that also allows transient traffic.
- Good examples of transit ASs are national and international ISPs (Internet backbones).

Path Attributes

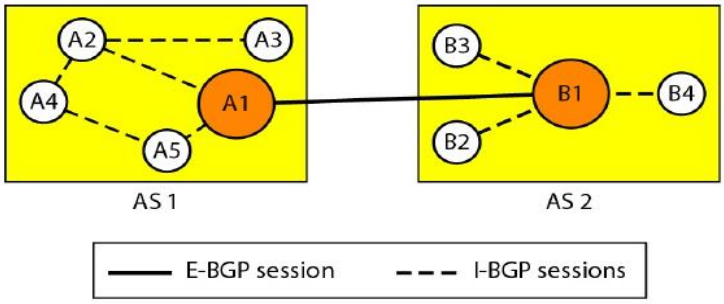
- The path was presented as a list of autonomous systems, but is, in fact, a list of attributes.
- Each attribute gives some information about the path.
- The list of attributes helps the receiving router make a more-informed decision when applying its policy.



- **Well-Known Mandatory:** Must be supported and recognised by all BGP routers. These attributes must be included in UPDATE messages. They must be passed on to other BGP routers.
- **Well-Known Discretionary:** Must be supported and recognised by all BGP routers. They must be passed on to other BGP routers. But these attributes may/may not be included in UPDATE messages, it's not mandatory.
- **Optional Transitive:** May be recognised/not recognised by BGP routers. But they must be passed on to other BGP routers. If these type of attributes aren't recognised, they're marked as "partial".
- **Optional Non-transitive:** May be recognised/not recognised by BGP routers and isn't passed on to other BGP routers.
- Also some vendors may use additional attribute to manipulate best path selection algoritm such as Cisco Systems, they use **weight** attribute which is locally significant, higher is better.

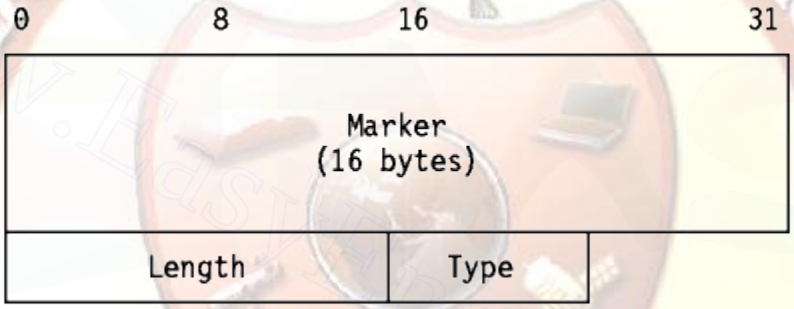
BGP Sessions

- The exchange of routing information between two routers using BGP takes place in a session.
- A session is a connection that is established between two BGP routers only for the sake of exchanging routing information.
- BGP can have two types of sessions: **External BGP (E-BGP)** and **Internal BGP (I-BGP)** sessions.
- The E-BGP session is used to exchange information between two speaker nodes belonging to two different autonomous systems.
- The I-BGP session is used to exchange routing information between two routers inside an autonomous system.



- The session established between AS1 and AS2 is an **E-BGP session**.
- The two speaker routers exchange information they know about networks in the Internet.
- However, these two routers need to collect information from other routers in the autonomous systems. This is done using **I-BGP sessions**.

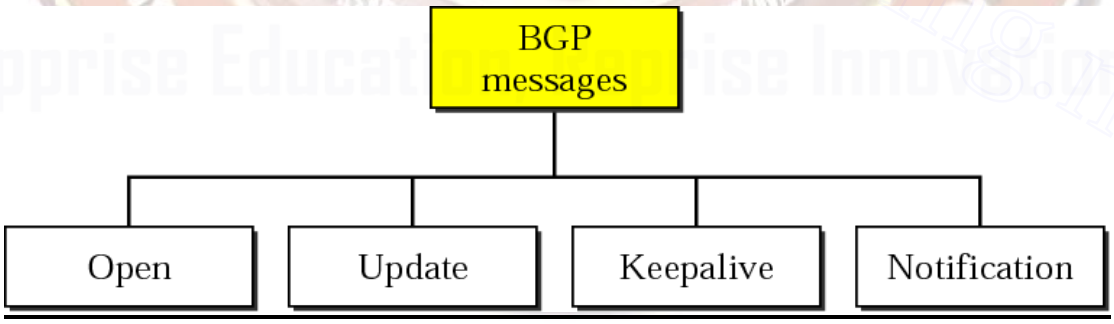
BGP Message Format



Marker: all ones in most cases; can be used for MD5 authentication.

Length: 19-4096 bytes

Type: one of four values (open, update, notification, keepalive)

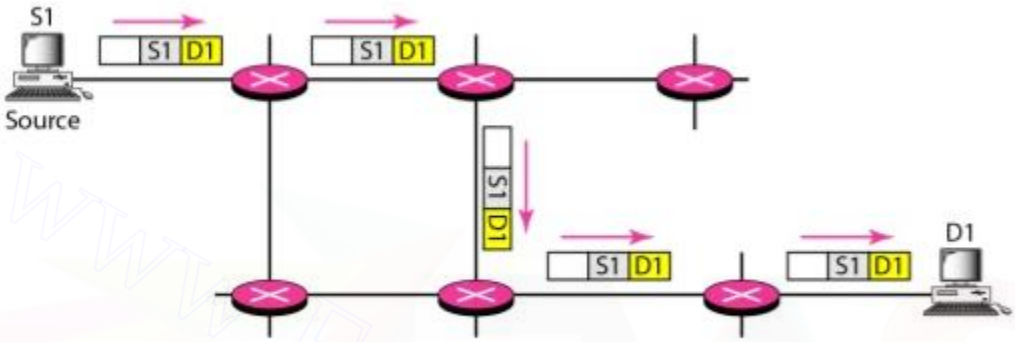


- OPEN
  - Establish BGP session with peer; negotiate hold time, advise ASN.
- KEEPALIVE
  - Periodic message sent so a router knows a peer is still up in absence of updates
- UPDATE
  - Routes added or withdrawn
- NOTIFICATION
  - Error condition encountered

MULTICASTING

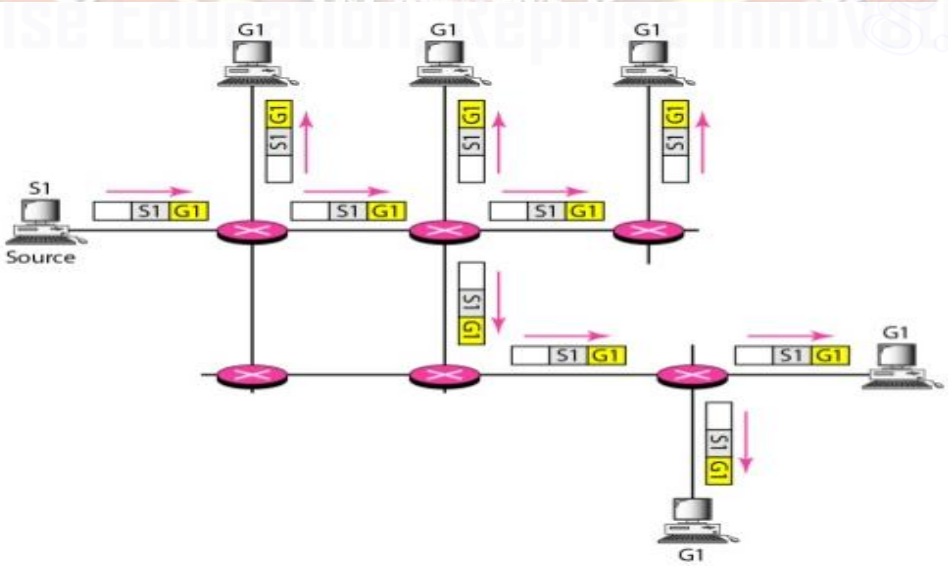
Unicasting

- In unicast communication, there is one source and one destination.
- The relationship 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 addresses assigned to the hosts.
- In unicast routing, the router forwards the received packet through only one of its interfaces.



Multicasting

- In multicast communication, there is one source and a group of destinations.
- The relationship is **one-to-many**.
- Here the source address is a unicast address, but the destination address is a group address.
- The group address identifies the members of the group.
- A multicast packet starts from the source S1 and goes to all destinations that belong to group G1.
- In multicasting, when a router receives a packet, it may forward it through several of its interface.



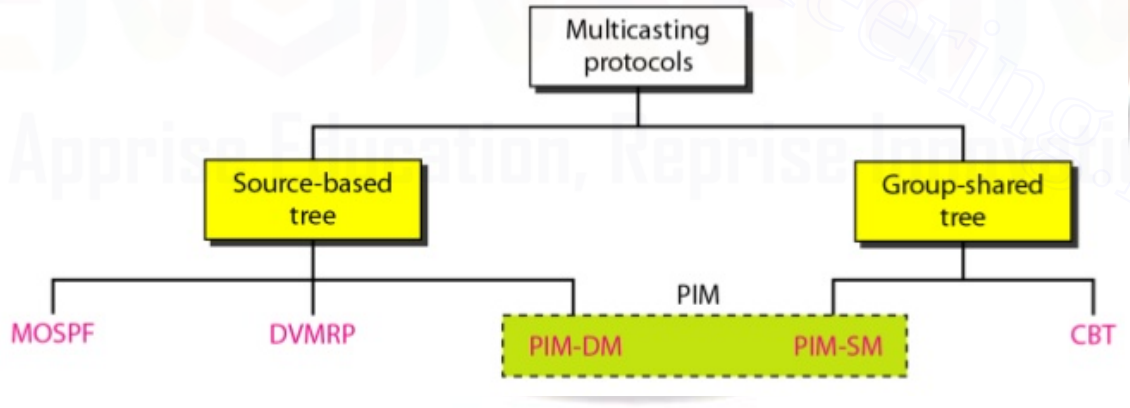
Multicast Addresses

- A multicast address is a destination address for a group of hosts that have joined a multicast group
- A packet that uses a multicast address as a destination can reach all members of the group unless there are some filtering restriction by the receiver
  - In classful addressing, multicast addresses occupied the only single block in class D
    - In classless addressing, the same block has been used for this purpose
      - In other words, the block assigned for multicasting is 224.0.0.0/4
- This large block, or the multicast address space as sometimes it is referred to, is divided into some smaller ranges,

Multicast Address Range

CIDR	*Range*	*Assignment*
224.0.0.0/24	224.0.0.0 → 224.0.0.255	Local Network Control Block
224.0.1.0/24	224.0.1.0 → 224.0.1.255	Internetwork Control Block
	224.0.2.0 → 224.0.255.255	AD HOC Block
224.1.0.0/16	224.1.0.0 → 224.1.255.255	ST Multicast Group Block
224.2.0.0/16	224.2.0.0 → 224.2.255.255	SDP/SAP Block
	224.3.0.0 → 231.255.255.255	Reserved
232.0.0.0/8	232.0.0.0 → 224.255.255.255	Source Specific Multicast (SSM)
233.0.0.0/8	233.0.0.0 → 233.255.255.255	GLOP Block
	234.0.0.0 → 238.255.255.255	Reserved
239.0.0.0/8	239.0.0.0 → 239.255.255.255	Administratively Scoped Block

MULTICAST ROUTING PROTOCOLS

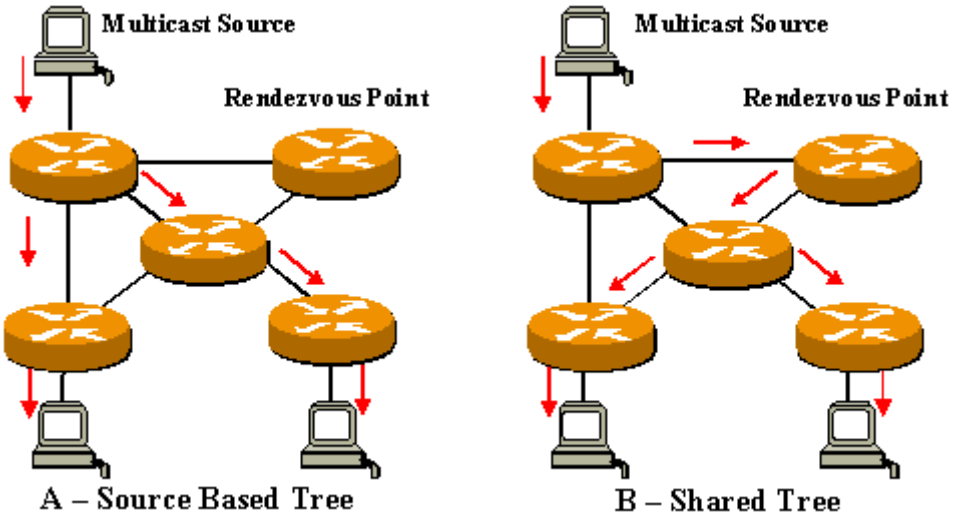


Source-Based Tree.

In the source-based tree approach, **each router** needs to have one shortest path tree for each group.

Group-Shared Tree.

In the group-shared tree approach, instead of each router having *m* shortest path trees, only one designated router, called the center core, or **rendezvous router**, takes the responsibility of distributing multicast traffic.



DVMRP (Distance Vector Multicast Routing Protocol)

- It is an implementation of **multicast distance vector routing**.
- It is a source-based routing protocol, based on RIP.

Multicast Distance Vector Routing:

- Unicast distance vector routing is very simple.
- Whereas multicast routing does not allow a router to send its routing table to its neighbors.
- The idea is to create a table from scratch by using the information from the unicast distance vector tables.
- The router never actually makes a routing table.
- When a router receives a multicast packet, it forwards the packet as though it is consulting a routing table.
- To accomplish this, the multicast distance vector algorithm uses a process based on **four decision-making strategies.**
    1. Flooding
    2. Reverse Path Forwarding (RPF).
    3. Reverse Path Broadcasting (RPB).
    4. Reverse Path Multicasting (RPM).

1. **Flooding**:

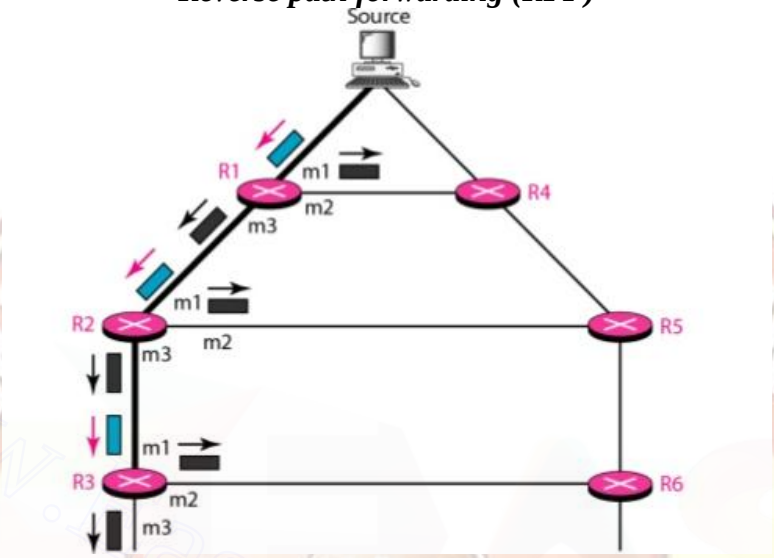
- A router receives a packet and, without even looking at the destination group address, sends it out from every interface except the one from which it was received
- Every network with active members receives the packet.
- This is a broadcast, not a multicast.
- **Problem**: It creates **loops.** A packet that has left the router may come back again from another interface or the same interface and be forwarded again.
- Reverse path forwarding, corrects this defect.

2. **Reverse Path Forwarding (RPF):**

- To prevent loops, only one copy is forwarded; the other copies are dropped.
- A router forwards only the copy that has traveled the shortest path from the source to the router.
- To find this copy, RPF uses the unicast routing table.

- The router receives a packet and extracts the source address (a unicast address).
- If the path from A to B is the shortest, then it is also the shortest from B to A.
- The router forwards the packet if it has traveled from the shortest path; it discards it otherwise.
- This strategy **prevents loops** because there is always one shortest path from the source to the router.

Reverse path forwarding (RPF)

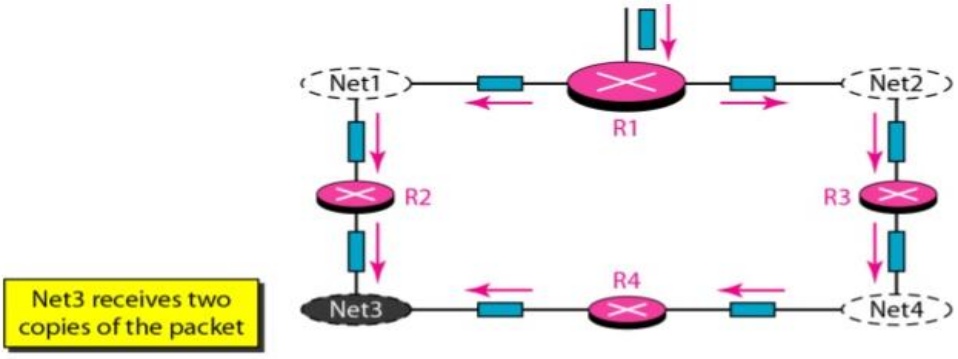


The shortest path tree as calculated by routers RI, R2, and R3 is shown by a thick line. When RI receives a packet from the source through the interface rnl, it consults its routing table and finds that the shortest path from RI to the source is through interface mI. The packet is forwarded. However, if a copy of the packet has arrived through interface m2, it is discarded because m2 does not define the shortest path from RI to the source. In reverse path forwarding (RPF), the router forwards only the packets that have traveled the shortest path from the source to the router; all other copies are discarded.

3. **Reverse path broadcasting**

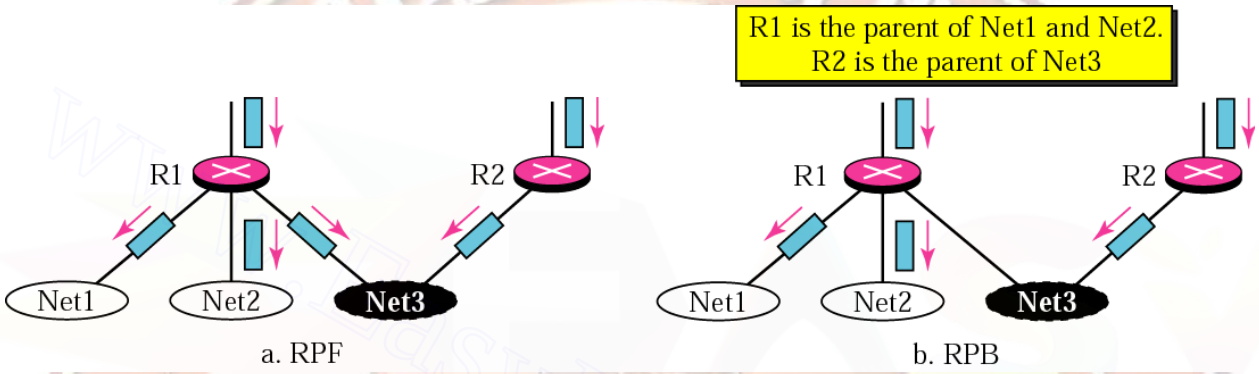
- RPF guarantees that each network receives a copy of the multicast packet without formation of loops.
- However, RPF does not guarantee that each network receives only one copy; a network may receive two or more copies.
- The reason is that RPF is not based on the destination address (a group address); forwarding is based on the source address.

Problem with RPF



- Net3 in this figure receives two copies of the packet even though each router just sends out one copy from each interface.
- To eliminate duplication, we must define only one parent router for each network.
- A network can receive a multicast packet from a particular source only through a designated parent router.
- For each source, the router sends the packet only out of those interfaces for which it is the designated parent.
- This policy is called **reverse path broadcasting (RPB).** RPB guarantees that the packet reaches every network and that every network receives only one copy.

RPF versus RPB

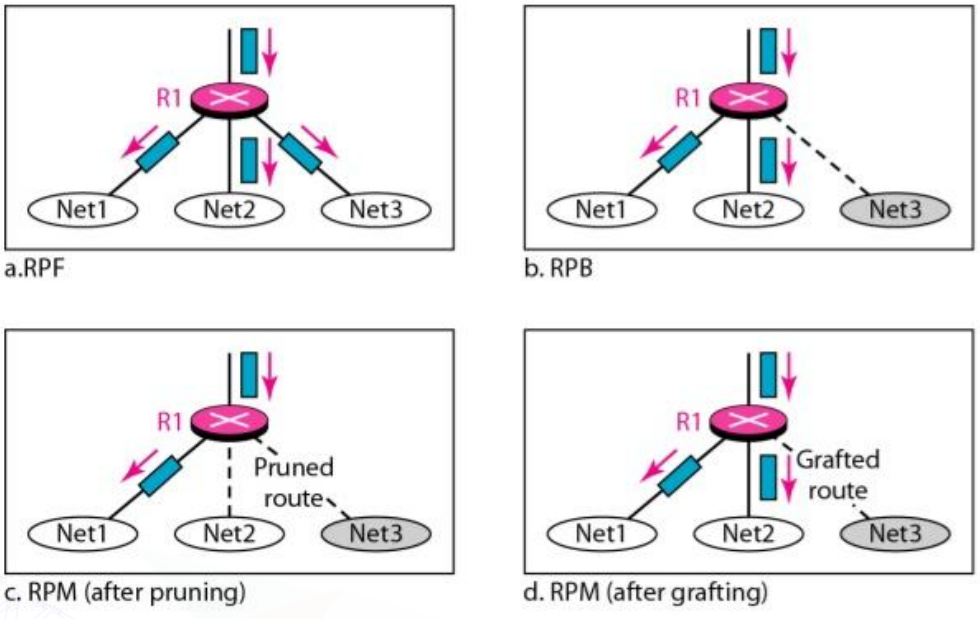


4. Reverse Path Multicasting (RPM).

- RPB does not multicast the packet, it broadcasts it. This is not efficient.
- To increase efficiency, the multicast packet must reach only those networks that have active members for that particular group.
- This is called **reverse path multicasting (RPM)**.
- To convert broadcasting to multicasting, the protocol uses two procedures, **pruning and grafting.**
- The router sends a **prune message** to the upstream router so that it can exclude the corresponding interface.

- That is, the upstream router can stop sending multicast messages for this group through that interface.
- The **graft message** forces the upstream router to resume sending the multicast messages.

RPF, RPB, and RPM



RPM adds pruning and grafting to RPB to create a multicast shortest path tree that supports dynamic membership changes.

PIM

Protocol Independent Multicast (PIM) is the name given to two independent multicast routing protocols: Protocol Independent Multicast, **Dense Mode** (PIM-DM) and Protocol Independent Multicast, **Sparse Mode** (PIM-SM).

PIM-DM

- PIM-DM is used when there is a possibility that each router is involved in multicasting (dense mode). In this environment, the use of a protocol that broadcasts the packet is justified because almost all routers are involved in the process.
- **PIM-DM is used in a dense multicast environment, such as a LAN.**
- It is a **source-based tree routing protocol** that uses RPF and pruning and grafting strategies for multicasting.
- Its operation is like that of DVMRP; however, unlike DVMRP, it does not depend on a specific unicasting protocol.
- It assumes that the autonomous system is using a unicast protocol and each router has a table that can find the outgoing interface that has an optimal path to a destination.
- This unicast protocol can be a distance vector protocol (RIP) or link state protocol (OSPF).

PIM-SM

- PIM-SM is used when there is a slight possibility that each router is involved in multicasting (sparse mode).
- In this environment, the use of a protocol that broadcasts the packet is not justified; a protocol such as CBT that uses a group-shared tree is more appropriate.
- **PIM-SM is used in a sparse multicast environment such as a WAN.**
- It is a **group-shared tree routing protocol** that has a rendezvous point (RP) as the source of the tree.
- Its operation is like CBT; however, it is simpler because it does not require acknowledgment from a join message.
- In addition, it creates a backup set of RPs for each region to cover RP failures.
- One of the characteristics of PIM-SM is that it can switch from a group-shared tree strategy to a source-based tree strategy when necessary.

IPv6

IPv4 has some deficiencies (listed below) that make it unsuitable for the fast-growing Internet.

- Address depletion is still a long-term problem in the Internet.
- Mminimum delay strategies and reservation of resources not provided in the IPv4 design.
- No encryption or authentication is provided by IPv4.

To overcome these deficiencies, IPv6 (**Internetworking Protocol, version 6**), also known as **IPng** (Internetworking Protocol, next generation), was proposed.

Advantages

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

o **Larger address space**. An IPv6 address is 128 bits long. Compared with the 32-bit address of IPv4, this is a huge increase in the address space.

o **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.

o **New options**. IPv6 has new options to allow for additional functionalities.

o Allowance for extension. IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.

o **Support for resource allocation**. In IPv6, the type-of-service field has been removed, but a mechanism (calledjlow *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.

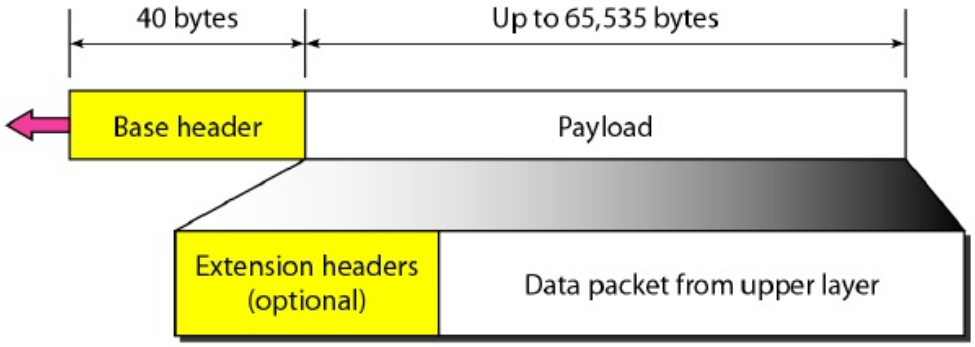
o **Support for more security**. The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

Packet Format

- The IPv6 packet is composed of a **mandatory base header** followed by the payload. The payload consists of two parts: **optional extension headers** and **data** from an upper layer.

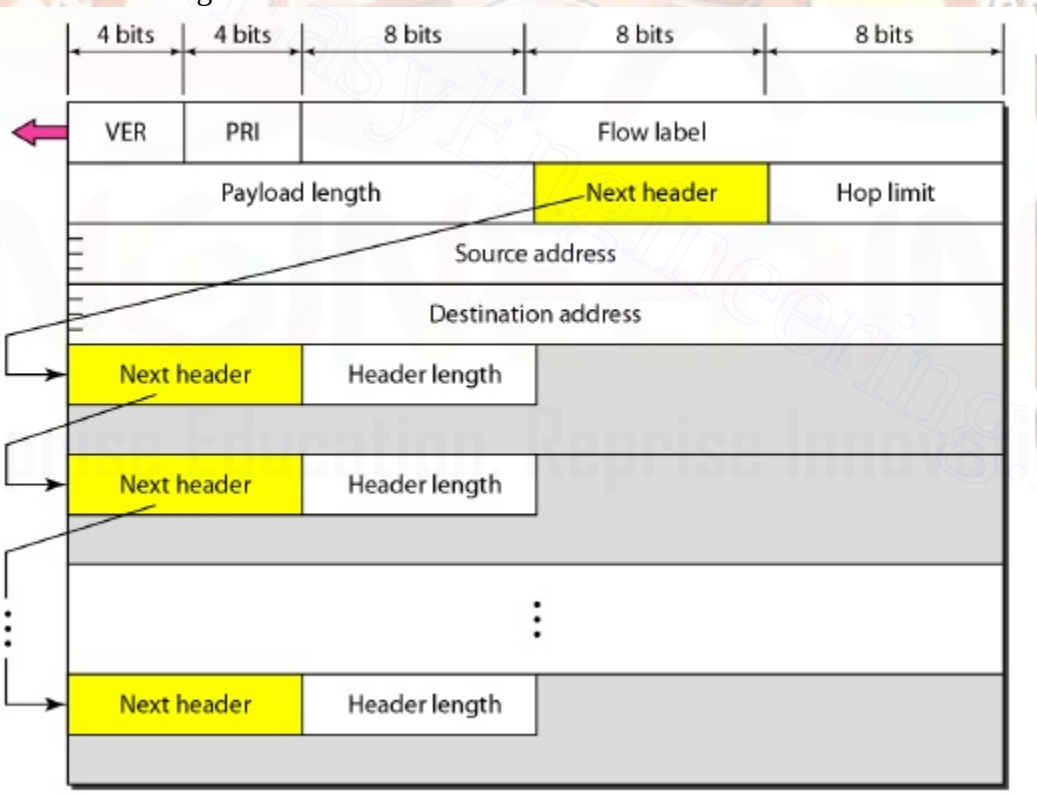
- The base header occupies 40 bytes, whereas the extension headers and data from the upper layer contain up to 65,535 bytes of information.

IPv6 datagram header and payload



Base Header

The base header has eight fields.



o **Version.** This 4-bit field defines the version number of the IP. For IPv6, the value is 6.
o **Priority.** The 4-bit priority field defines the priority of the packet with respect to traffic congestion.

Priorities for congestion-controlled traffic

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

Priorities for noncongestion-controlled traffic

Priority	*Meaning*
8	Data with greatest redundancy
. . .	. . .
15	Data with least redundancy

- **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.

O 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.

o **Payload length**. The 2-byte payload length field defines the length of the IP datagram excluding the base header.

o **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 shows the values of next headers. Note that this field in version 4 is called the *protocol.*

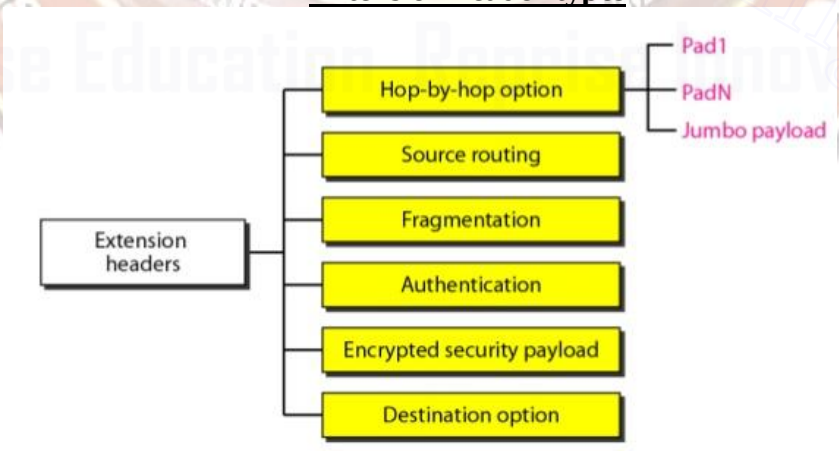
Code	Next Header
0	Hop-by-hop option
2	ICMP
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null (no next header)
60	Destination option

o **Hop limit**. This 8-bit hop limit field serves the same purpose as the TTL field in IPv4.
o **Source address**. The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram.
O 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.

Extension Headers

- To give greater functionality to the IP datagram, the base header can be followed by up to six extension headers.
- Six types of extension headers have been defined, as shown in Figure

Extension header types



Hop-by-Hop Option

- The hop-by-hop option is used when the source needs to pass information to all routers visited by the datagram.
- So far, only three options have been defined: Padl, PadN, and jumbo payload.

- The Padl option is 1 byte long and is designed for alignment purposes.
- PadN is used when 2 or more bytes is needed for alignment.
- The jumbo payload option is used to define a payload longer than 65,535 bytes.

Source Routing

- It combines the concepts of the strict source route and the loose source route options of IPv4.

Fragmentation

- The concept of fragmentation is the same as that in IPv4. However, the place where fragmentation occurs differs.
- In IPv4, the source or a router is required to fragment if the size of the datagram is larger than the MTU of the network over which the datagram travels.
- In Ipv6, only the original source can fragment.

Authentication

- It has a dual purpose:
  1. It validates the message sender .
  2. It ensures the integrity of data.

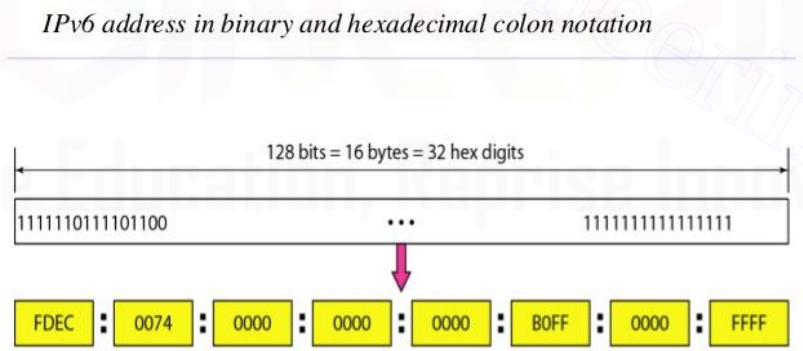
Encrypted Security Payload

- It provides confidentiality and guards against eavesdropping.

Destination Option

- It is used when the source needs to pass information to the destination only. Intermediate routers are not permitted to access this information.

IPv6 Address



IPv4	IPv6
IPv4 addresses are 32 bit length.	IPv6 addresses are 128 bit length.
IPv4 addresses are binary numbers represented in decimals.	IPv6 addresses are binary numbers represented in hexadecimals.
IPSec support is only optional.	Inbuilt IPSec support.
Fragmentation is done by sender and forwarding routers.	Fragmentation is done only by sender.
No packet flow identification.	Packet flow identification is available within the

	IPv6 header using the Flow Label field.
Checksum field is available in IPv4 header	No checksum field in IPv6 header.
Options fields are available in IPv4 header.	No option fields, but IPv6 Extension headers are available.
Address Resolution Protocol (ARP) is available to map IPv4 addresses to MAC addresses.	Address Resolution Protocol (ARP) is replaced with a function of Neighbor Discovery Protocol (NDP).
Internet Group Management Protocol (IGMP) is used to manage multicast group membership.	IGMP is replaced with Multicast Listener Discovery (MLD) messages.
Broadcast messages are available.	Broadcast messages are not available. Instead a link-local scope "All nodes" multicast IPv6 address (FF02::1) is used for broadcast similar functionality.
Manual configuration (Static) of IPv4 addresses or DHCP (Dynamic configuration) is required to configure IPv4 addresses.	Auto-configuration of addresses is available.

UNIT IV
TRANSPORT LAYER

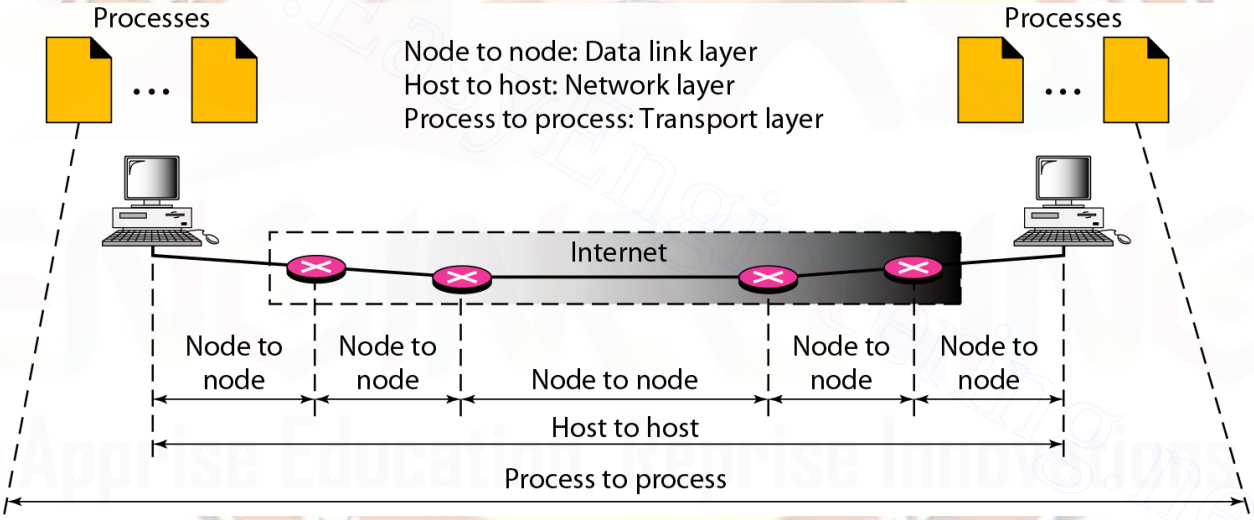
Overview of Transport layer - UDP - Reliable byte stream (TCP) - Connection management - Flow control - Retransmission – TCP Congestion control - Congestion avoidance (DECbit, RED) – QoS – Application requirements

The Internet model has three protocols at the transport layer: **UDP, TCP, and SCTP**.

PROCESS-TO-PROCESS DELIVERY

The **data link layer** is responsible for delivery of frames between two neighboring nodes over a link. This is called node-to-node delivery. The **network layer** is responsible for delivery of datagrams between two hosts. This is called host-to-host delivery. Communication on the Internet is not defined as the exchange of data between two nodes or between two hosts. Real communication takes place between two processes.

The **transport layer** is responsible for process-to-process delivery-the delivery of a packet, part of a message, from one process to another.



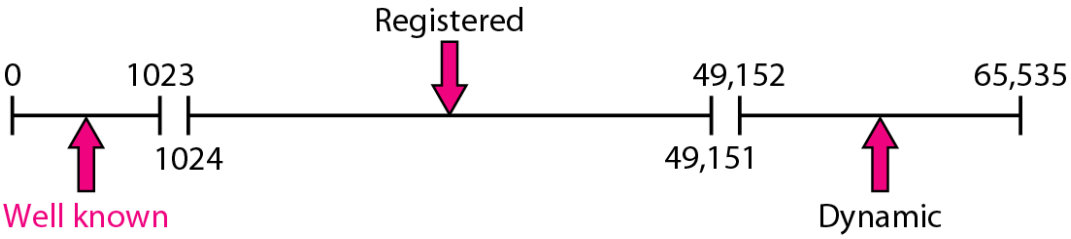
lANA Ranges

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

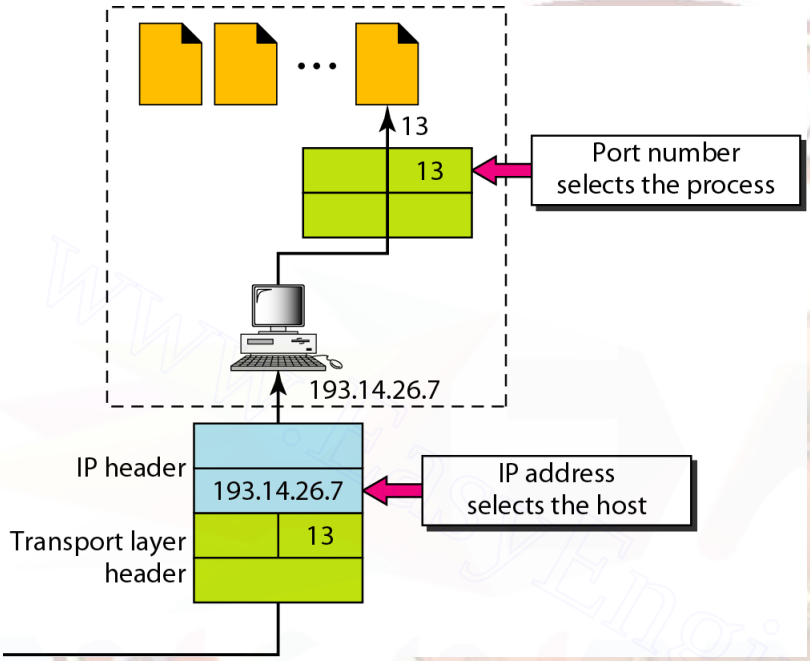
Well-known ports. The ports ranging from 0 to 1023 are assigned and controlled by IANA. These are the well-known ports.

Registered ports. The ports ranging from 1024 to 49,151 are not assigned or controlled by IANA. They can only be registered with lANA 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.



IP addresses versus port numbers



Socket Addresses

Process-to-process delivery needs two identifiers,
IP address and the port number.

The combination of an IP address and a port number is called a socket address. The client socket address defines the client process uniquely just as the server socket address defines the server process uniquely.

A transport layer protocol needs a pair of socket addresses:
Client socket address
Server socket address.

UNRELIABLE PROTOCOL - USER DATAGRAM PROTOCOL (UDP)

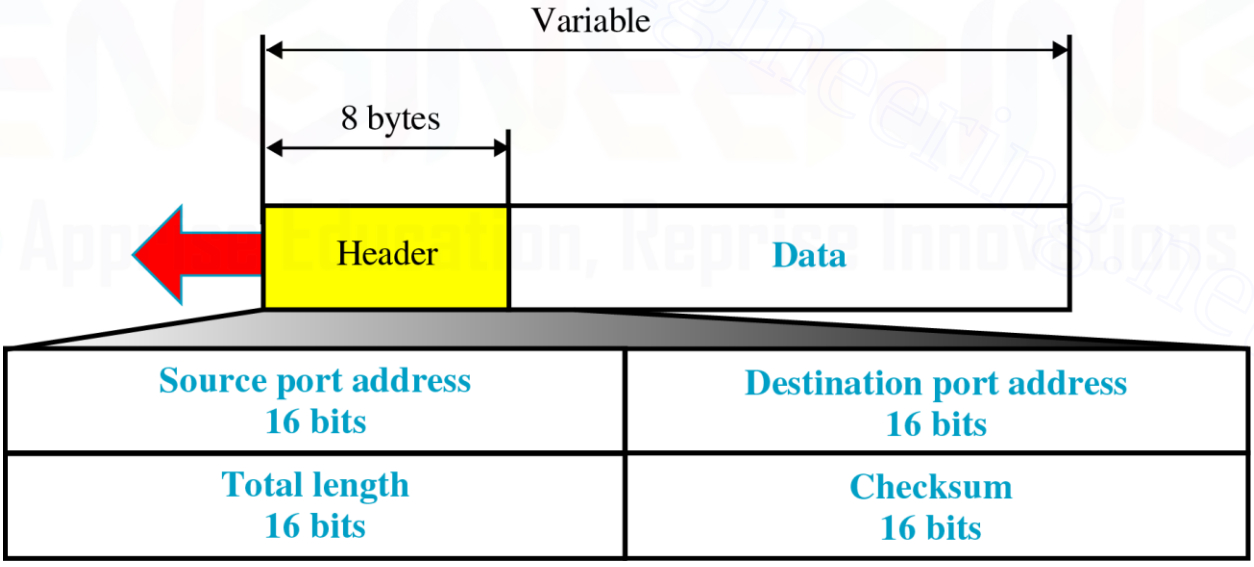
The User Datagram Protocol (UDP) is called a **connectionless, unreliable transport protocol**. It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication.

Well-Known Ports 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

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



Source port number. This is the port number used by the process running on the source host. It is 16 bits long, which means that the port number can range from 0 to 65,535.
Destination port number. This is the port number used by the process running on the destination host. It is also 16 bits long.

Length. This is a 16-bit field that defines the total length of the user datagram, header plus data. The 16 bits can define a total length of 0 to 65,535 bytes.

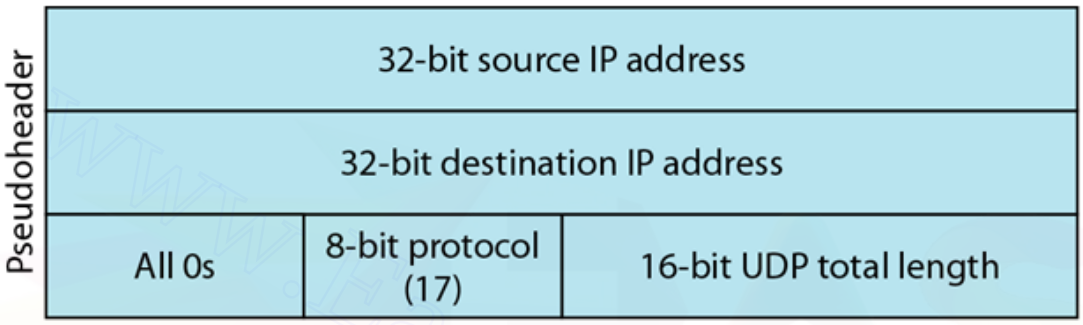
UDP length = IP length - IP header's length

Checksum. This field is used to detect errors over the entire user datagram.

The UDP checksum calculation is different from the one for IP and ICMP. Here the checksum includes three sections: a pseudoheader, the UDP header, and the data coming from the application layer. The pseudoheader is the part of the header of the IP packet in which the user datagram is to be encapsulated with some fields filled with Os.

If the checksum does not include the pseudoheader, a user datagram may arrive safe and sound. However, if the IP header is corrupted, it may be delivered to the wrong host.

Pseudoheader for checksum calculation



UDP OPERATION

Connectionless Services

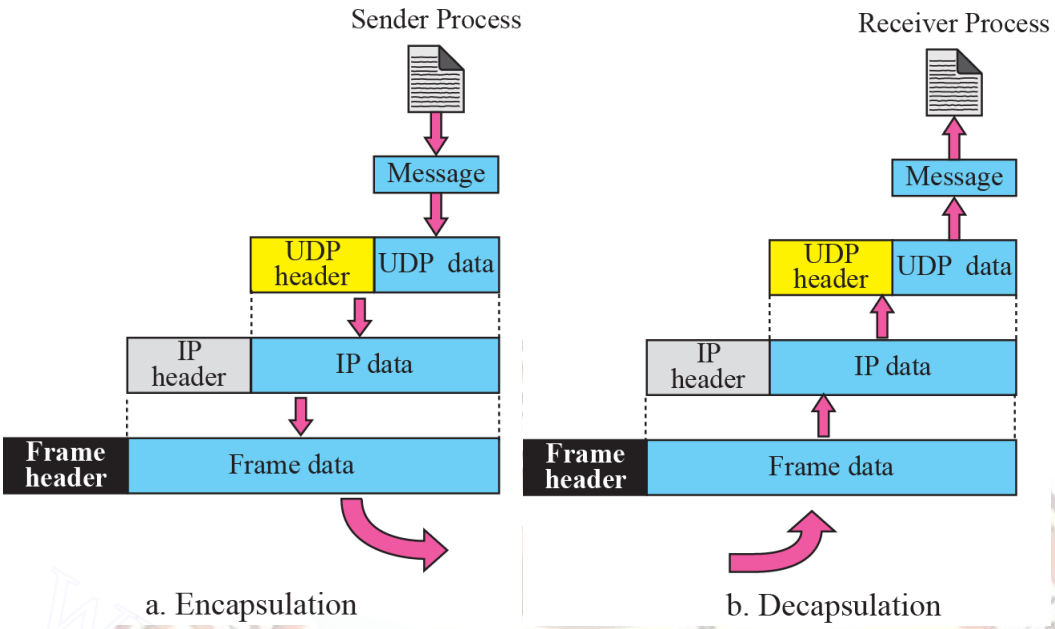
- There is no relationship between the different user datagrams even if they are coming from the same source process and going to the same destination program.
- The user datagrams are not numbered.
- There is no connection establishment and no connection termination.
- Each user datagram can travel on a different path.

Flow and Error Control

- UDP is a very simple, unreliable transport protocol.
- No flow control and hence no window mechanism.
- There is no error control mechanism in UDP except for the checksum.
- Sender does not know if a message has been lost or duplicated.
- When the receiver detects an error through the checksum, the user datagram is silently discarded.

Encapsulation and Decapsulation

To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages in an IP datagram.



Queuing

- Queues are opened for server / client processes
- Two queues for each process
  - **Incoming queue:** receive messages
  - **Outgoing queue:** send messages
- The queues function as long as the process is running
- The queues are destroyed when the process terminates

Queues on the client side

The client process requests a port number from the operating system
The process opens incoming and outgoing queues with the requested port number

Queues on the server side

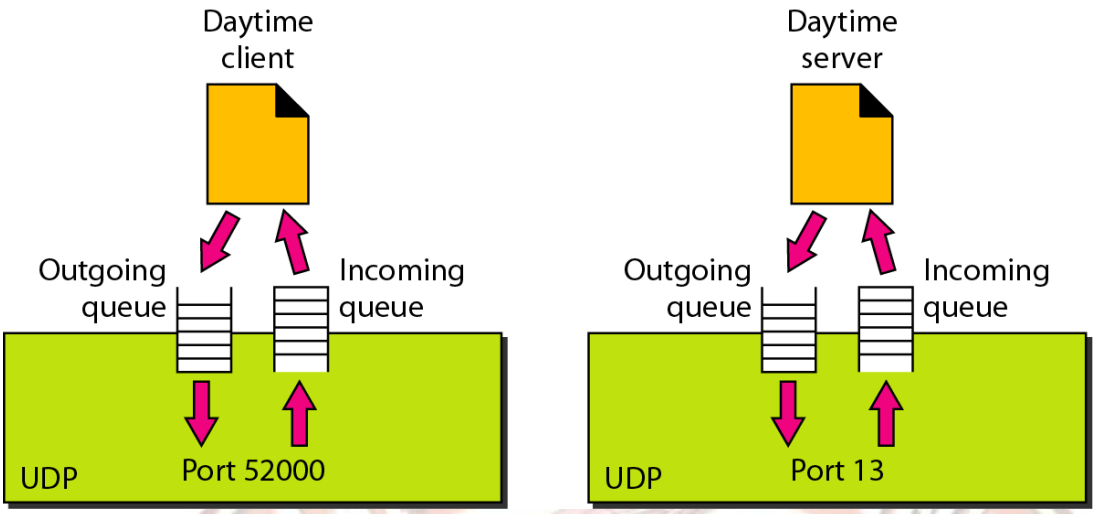
The server asks for incoming and outgoing queues using its well-known port number.

Outgoing queue overflow

The operating system asks the server / client to wait before sending any more messages

Incoming queue overflow

UDP drops the datagram and asks the ICMP protocol to send port unreachable message to the datagram sender



USE OF UDP

- o UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control.
- o UDP is suitable for a process with internal flow and error control mechanisms
- o UDP is a suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software but not in the TCP software.
- o UDP is used for management processes such as SNMP.
- o UDP is used for some route updating protocols such as Routing Information Protocol (RIP)

RELIABLE BYTE STREAM : TCP(TRANSMISSION CONTROL PROTOCOL)

TCP is a **connection-oriented** protocol; it creates a virtual connection between two TCPs to send data. In addition, TCP uses **flow and error control** mechanisms at the transport level**.**

TCP is called a ***connection-oriented, reliable*** transport protocol

TCP SERVICES

1.Process-to-Process Communication

Like UDP, TCP provides process-to-process communication using port numbers

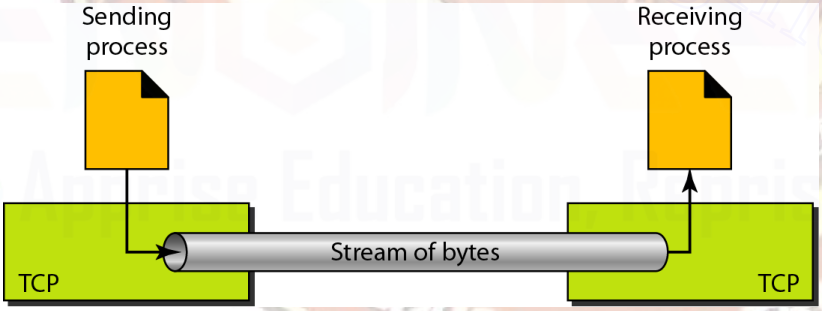
Ports used by TCP

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
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

2.Stream Delivery Service

TCP is a stream-oriented protocol.

The sending process produces the stream of bytes, and the receiving process consumes them.



Sending and receiving buffers

The sending and the receiving processes may not write or read data at the same speed, hence TCP needs buffers for storage.
There are two buffers, the sending buffer and the receiving buffer, one for each direction.
One way to implement a buffer is to use a circular array of 1-byte locations

Sending Buffer:

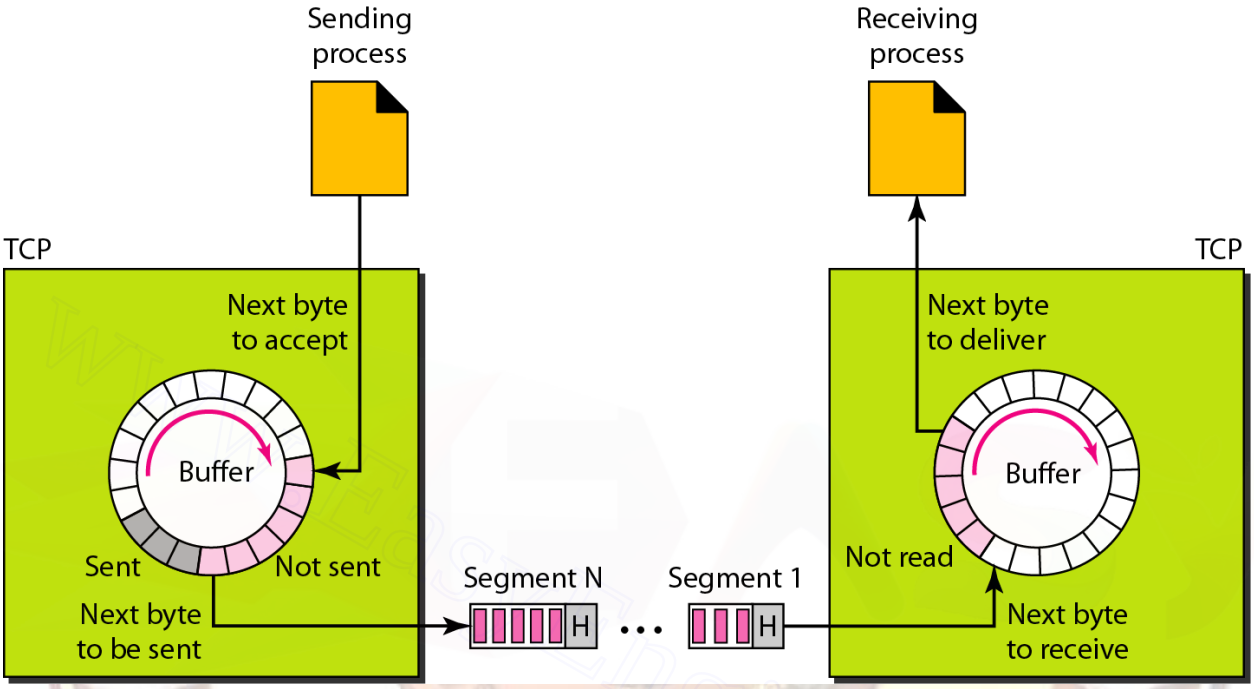
- The buffer has three types of chambers.
- The **white section** contains empty chambers that can be filled by the sending process.
- The **gray area** holds bytes that have been sent but not yet acknowledged.

- TCP keeps these bytes in the buffer until it receives an acknowledgment.
- The **colored area** contains bytes to be sent by the sending TCP.

Receiving Buffer:

- The circular buffer is divided into two areas (shown as white and colored).
- The white area contains empty chambers to be filled by bytes received from the network.
- The colored sections contain received bytes that can be read by the receiving process.

chambers.



3.Full-Duplex Communication:

TCP offers full-duplex service, in which data can flow in both directions at the same time. Each TCP then has a sending and receiving buffer, and segments move in both directions.

4.Connection-Oriented Service

TCP, unlike UDP, is a connection-oriented protocol. When a process at site A wants to send and receive data from another process at site B, the following occurs:

1. The two TCPs establish a connection between them.
2. Data are exchanged in both directions.
3. The connection is terminated.

5.Reliable Service

TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data. We will discuss this feature further in the section on error control.

TCP FEATURES

1. Numbering System:

- **Byte Number**: The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number.
- **Sequence Number:** The value in the sequence number field of a segment defines the number of the first data byte contained in that segment.
- **Acknowledgment Number:** The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive.

2.Flow Control

- TCP provides *flow control*
- The receiver of the data controls the amount of data that are to be sent by the sender. This is done to prevent the receiver from being overwhelmed with data.
- TCP uses byte-oriented flow control.

3.Error Control

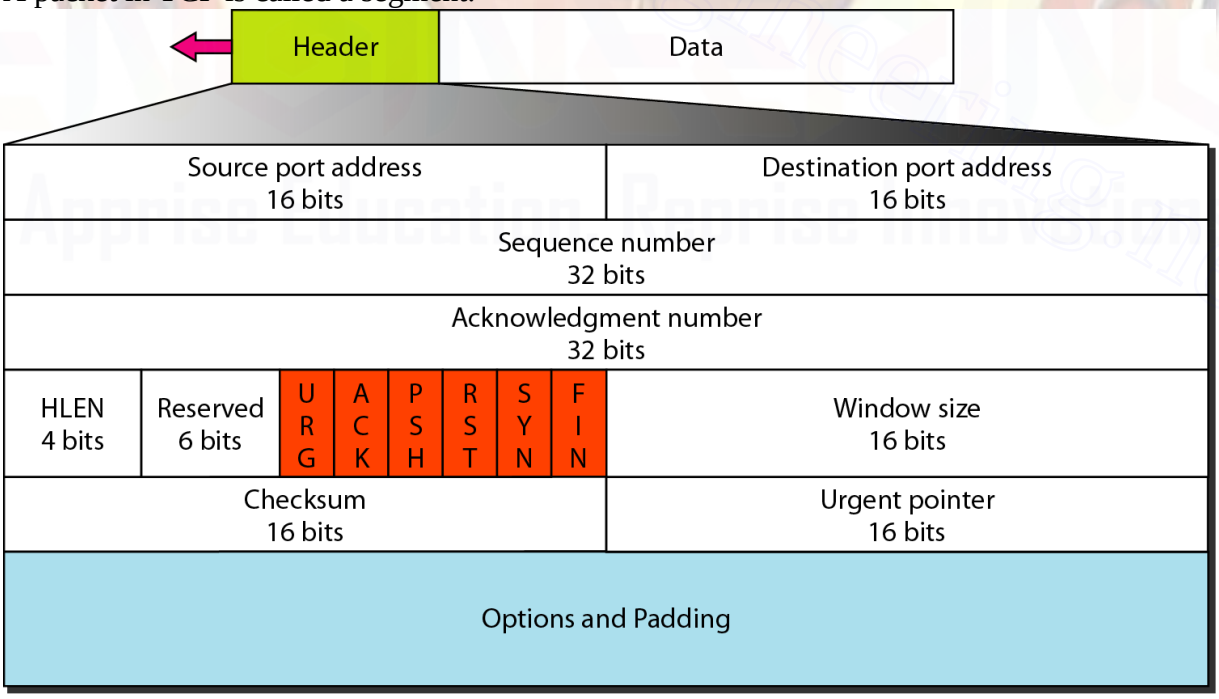
- To provide reliable service, TCP implements an error control mechanism.
- Error control considers a segment as the unit of data for error detection (loss or corrupted
- segments).
- Error control is byte-oriented.

4. Congestion Control

TCP takes into account congestion in the network. The amount of data sent by a sender is not only controlled by the receiver (flow control), but is also determined by the level of congestion in the network.

TCP SEGMENT FORMAT

A packet in TCP is called a segment.



The segment consists of a 20- to 60-byte header, followed by data from the application program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options.

Source port address. This is a 16-bit field that defines the port number of the application program in the host that is sending the segment. This serves the same purpose as the source port address in the UDP header.

Destination port address. This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment. This serves the same purpose as the destination port address in the UDP header.

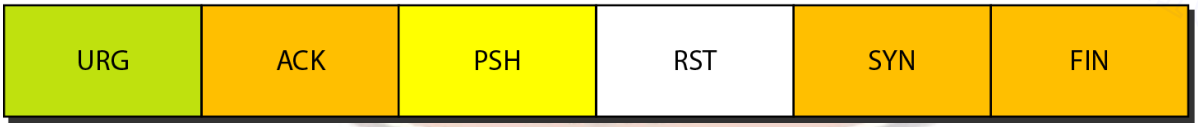
Sequence number. This 32-bit field defines the number assigned to the first byte of data contained in this segment. As we said before, TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence comprises the first byte in the segment. During connection establishment, each party uses a random number generator to create an initial sequence number (ISN), which is usually different in each direction.

Acknowledgment number. This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number *x* from the other party, it defines *x* + I as the acknowledgment number. Acknowledgment and data can be piggybacked together.

Header length. This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field can be between 5 (5 x 4 =20) and 15 (15 x 4 =60).
Reserved. This is a 6-bit field reserved for future use.

Control. This field defines 6 different control bits or flags as shown in Figure 23.17.
One or more of these bits can be set at a time.

URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection



TCP CONNECTION MANAGEMENT

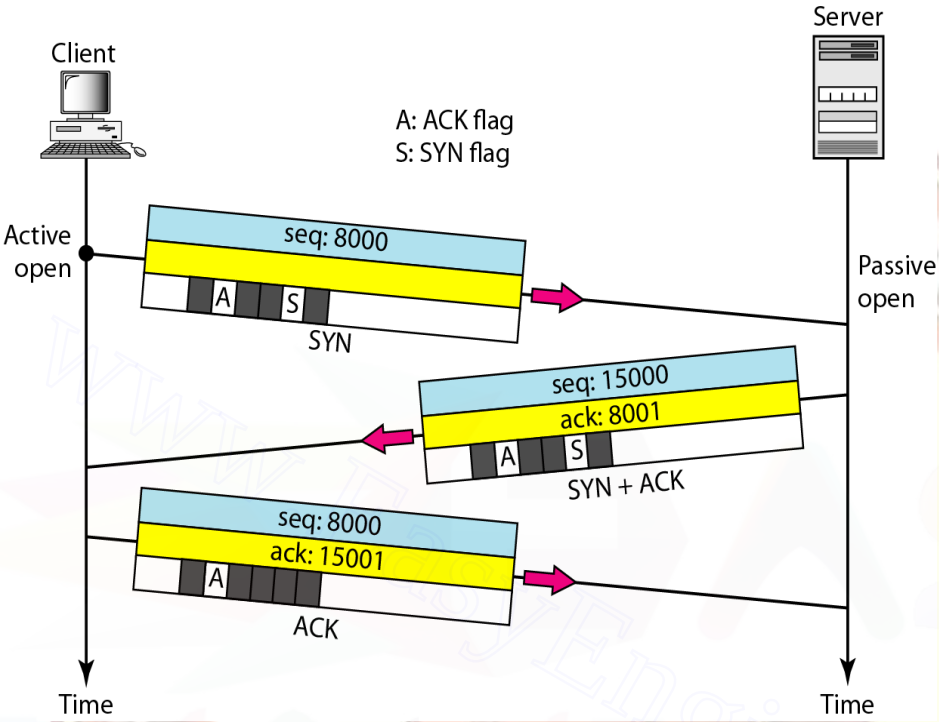
TCP is connection-oriented. A connection-oriented transport protocol establishes a virtual path between the source and destination. All the segments belonging to a message are then sent over this virtual path.
Connection-oriented transmission requires **three phases:**

- **Connection establishment**
- **Data transfer**
- **Connection termination**

1. CONNECTION ESTABLISHMENT

TCP transmits data in full-duplex mode. When two TCPs in two machines are connected, they are able to send segments to each other simultaneously.

Three-Way Handshaking:



The connection establishment in TCP is called **three-way handshaking**. An application program, called the client, wants to make a connection with another application program, called the server, using TCP as the transport layer protocol.

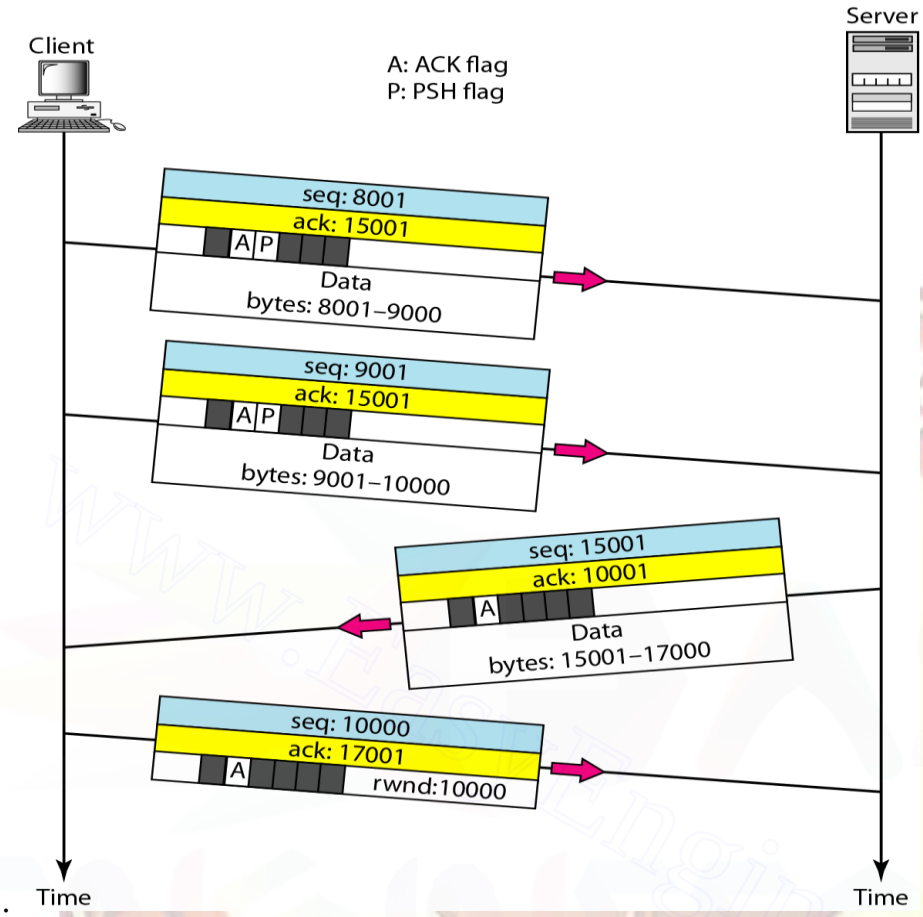
1. The client sends the first segment, a SYN segment, in which only the **SYN flag** is set.This segment is for synchronization of sequence numbers. It consumes one sequence number. When the data transfer starts, the sequence number is incremented by 1. The SYN segment carries no real data.

2. The server sends the second segment, a SYN +ACK segment, with 2 flag bits set: **SYN and ACK.** This segment has a dual purpose. It is a SYN segment for communication in the other direction and serves as the acknowledgment for the SYN segment. It consumes one sequence number.

3. The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the **ACK flag** and acknowledgment number field. Note that the sequence number in this segment is the same as the one in the SYN segment; the ACK segment does not consume any sequence numbers.

2.DATA TRANSFER:

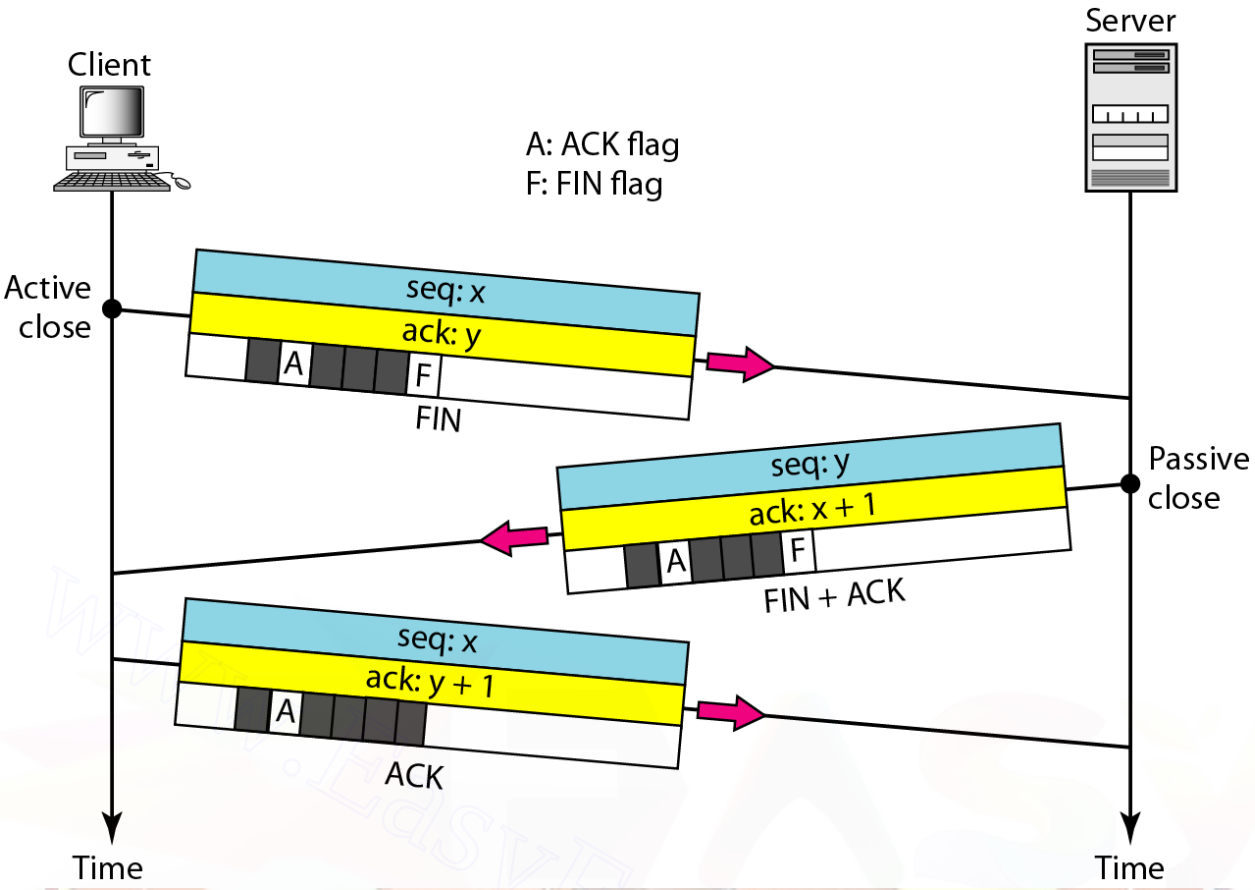
After connection is established, bidirectional data transfer can take place.



- The client and server can both send data and acknowledgments.
- The acknowledgment is piggybacked with the data.
- The data segments sent by the client have the **PSH (push) flag** set so that the server TCP knows to deliver data to the server process as soon as they are received.

3.CONNECTION TERMINATION:

Any of the two parties involved in exchanging data (client or server) can close the connection



Three-Way Handshaking for connection termination

1. The client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the **FIN flag** is set.
2. The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a **FIN +ACK segment**, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction.
3. The client TCP sends the last segment, an **ACK segment**, to confirm the receipt of the FIN segment from the TCP server.

STATE TRANSITION DIAGRAM FOR TCP

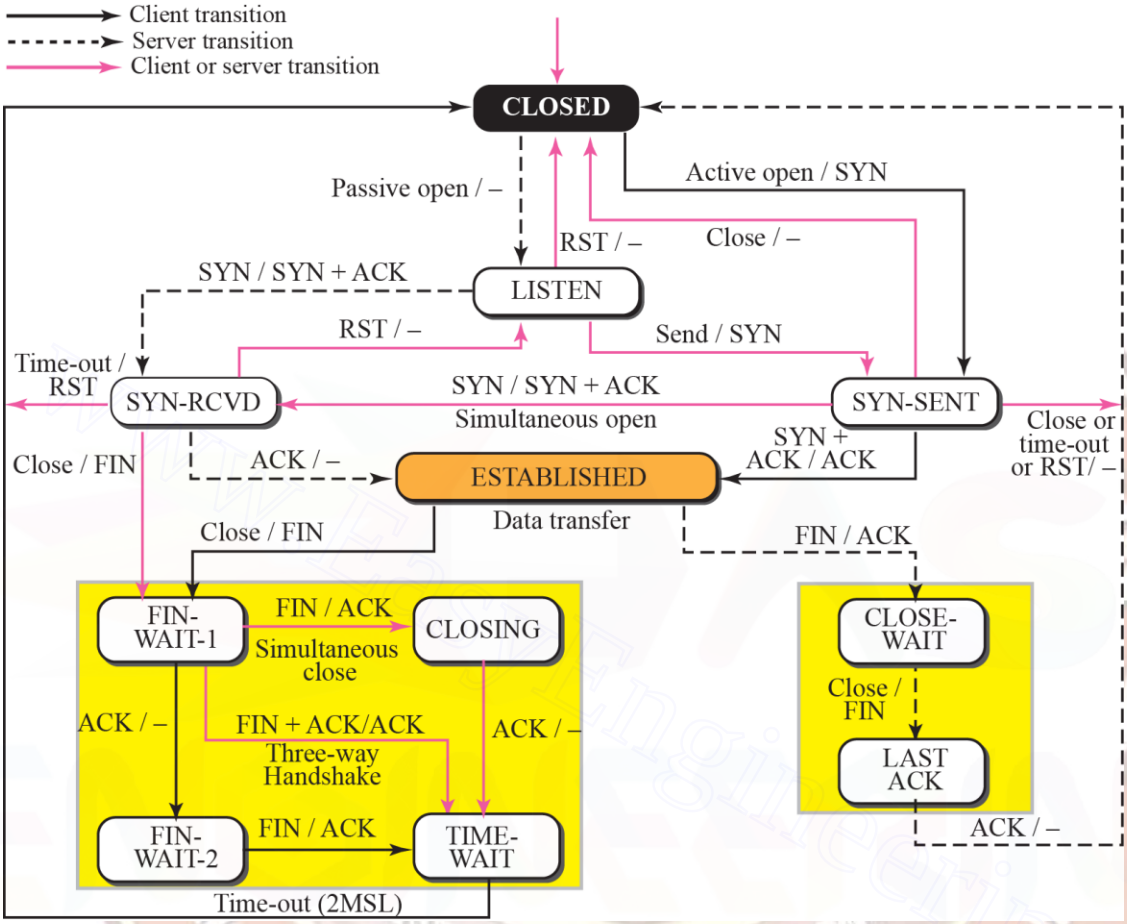
A TCP connection progresses through a series of states during its lifetime. All connections start in the CLOSED state. As the connection progresses, the connection moves from state to state according to the arcs. Each arc is labeled with a tag of the form event/action. Thus, if a connection is in the LISTEN state and a SYN segment arrives (i.e., a segment with the SYN flag set), the connection makes a transition to the SYN RCVD state and takes the action of replying with an ACK+SYN segment.

There are three combinations of transitions that get a connection from the ESTABLISHED state to the CLOSED state:

This side closes first: ESTABLISHED→FIN WAIT 1→FIN WAIT 2→TIME WAIT→CLOSED.

The other side closes first: ESTABLISHED→CLOSE WAIT→LAST ACK→CLOSED.

Both sides close at the same time: ESTABLISHED→FIN WAIT 1→CLOSING→TIME WAIT→CLOSED



State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIMED WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

TCP FLOW CONTROL

Flow control defines the amount of data a source can send before receiving an acknowledgement from receiver

– The flow control protocol must not be too slow

– The flow control protocol must make sure that receiver does not get overwhelmed with data

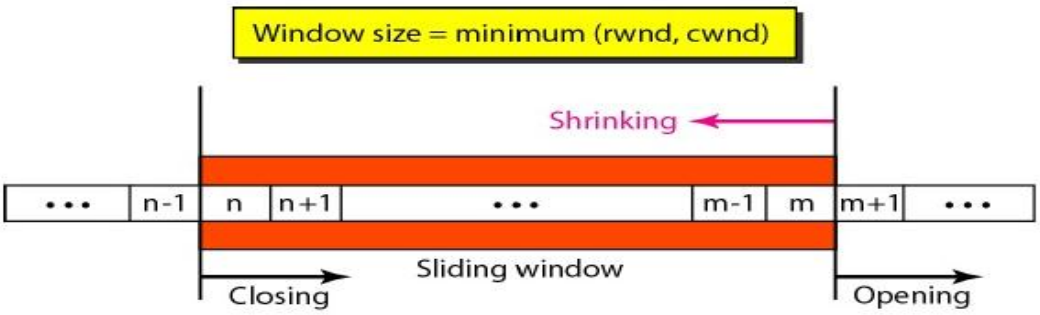
- Flow Control guarantees the reliable delivery of data.
- It ensures that data is delivered in order.

TCP uses 2 techniques to handle flow control.

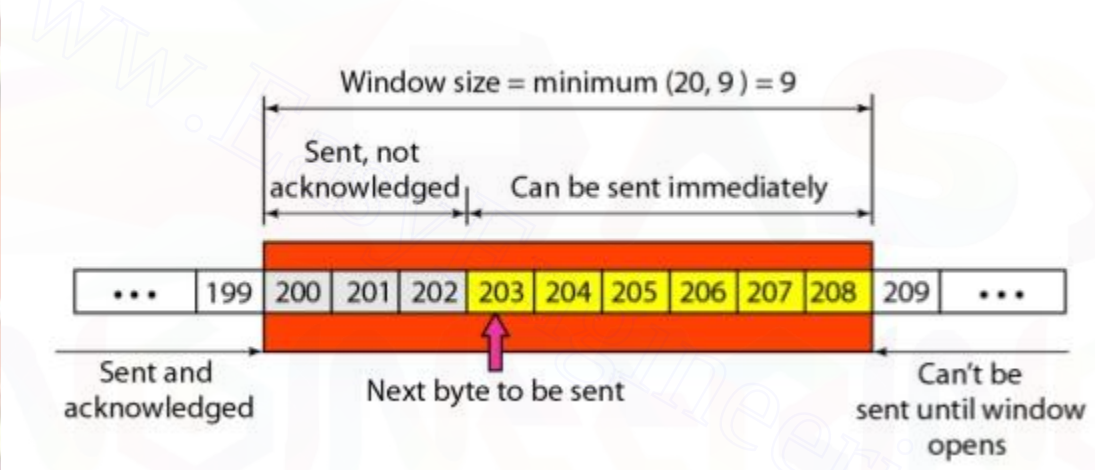
1. Sliding Window
2. Silly Window

Sliding Window:

- The sliding window protocol used by TCP has two walls: one left and one right.
- The window is *opened, closed,* or *shrunk.*
- **Opening a window** means moving the right wall to the right. This allows more new bytes in the buffer that are eligible for sending.
- **Closing the window** means moving the left wall to the right. This means that some bytes have been acknowledged.
- **Shrinking the window** means moving the right wall to the left. This means revoking the eligibility of some bytes for sending.



- **The size of the window** at one end is determined by the lesser of two values: *receiver window (rwnd)* or *congestion window (cwnd).*
- The **receiver window** is the value advertised by the opposite end in a segment containing acknowledgment. It is the number of bytes the other end can accept before its buffer overflows and data are discarded.
- The **congestion window** is a value determined by the network to avoid congestion.



Problem with the Sliding Window:

- Sender will send the data based on the Receiver's window Size.
- If the Receiver's Window size is Zero, then the sender can't send the data.

SILLY WINDOW SYNDROME

It is a situation in which a small window size is advertised by the receiver(1 Byte) and a small segment is sent by the sender.

A serious problem in sliding window operation

1. Sending application program creates data slowly
2. Receiving application program consumes data slowly.

In either case, data may be sent in small segments – inefficient use of bandwidth – increased processing by TCP

- In TCP "full" containers (data segments) going in one direction and empty containers (ACKs) going in the reverse direction, then MSS-sized segments correspond to large containers and 1-byte segments correspond to very small containers.
- If the sender aggressively fills an empty container as soon as it arrives, then any small container introduced into the system remains in the system indefinitely.
- That is, it is immediately filled and emptied at each end.

Silly Window Solution
Nagle's Algorithm

- How long does sender delay sending data?
  - too long: hurts interactive applications
  - too short: poor network utilization
  - strategies: timer-based vs self-clocking (ack)
- When application generates additional data
  - if fills a max segment (and window open): send it
  - else
    - if there is unack'ed data in transit: buffer it until ACK arrives
    - else: send it

Nagle Algorithm

```
When the application produces data to send
        if available data & window both >= MSS
                send a full segment
        else
                if there unACKed data in flight
                        buffer new data until Ack arrives
                else
                        send new data now
```

ADAPTIVE RETRANSMISSION

- Remember, TCP has to ensure reliability.
- So bytes need to be resent if there is no "timely" acknowledgement.
- How long should the sender wait ?
- It should be adaptive -- fluctuation in load on the network.
  - If too short, false time-outs
  - If too long, then poor rate of sending.
- Depends on round trip time**(RTT)** estimation

RTT Estimation (Original Algorithm)

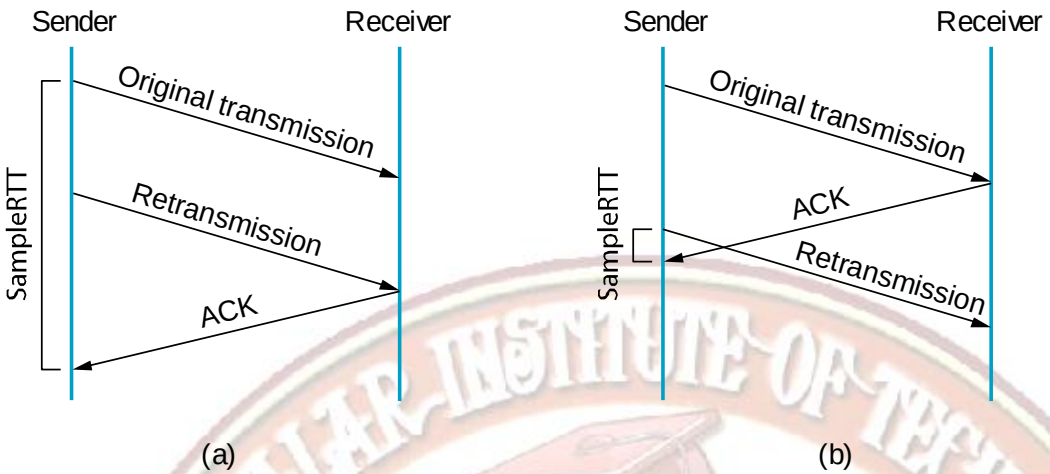
- Simple mechanism could be:
  - Send packet, record time T1
  - When ACK is returned, record time = T2.

    Estimated RTT = T2 – T1
- To avoid fluctuations, estimated RTT is a weighted average of previous time and current sample

  Estimated RTT = (1-a) Estimated RTT + a SampleRTT
- In the original specification a = 0.125

  Time out = 2 * RTT.

A problem here is



(a) (b)

- When there are retransmissions, it is unclear if the ACK is for the original transmission or for a retransmission.
  - Two algorithms are proposed to overcome this problem.

The Karn Patridge Algorithm

- Take SampleRTT measurements only for segments that have been sent once !
- This eliminates the possibility that wrong RTT estimates are factored into the estimation.

Another change -- Each time TCP retransmits, it sets the next timeout = 2 X Last timeout --> This is called the Exponential Back-off (primarily for avoiding congestion).

Jacobson Karels Algorithm

- The main problem with the Karn/Patridge scheme is that it does not take into account the variation between RTT samples.
- New method proposed -- the Jacobson Karels Algorithm.
- Estimated RTT = Estimated RTT + d X Difference
- Difference = Sample RTT - Estimated RTT
- Deviation = Deviation + d (|Difference| - deviation)
- Timeout = m Estimated RTT + f deviation.
- The values of m and f are computed based on experience.

  Typically m = 1 and f = 4.

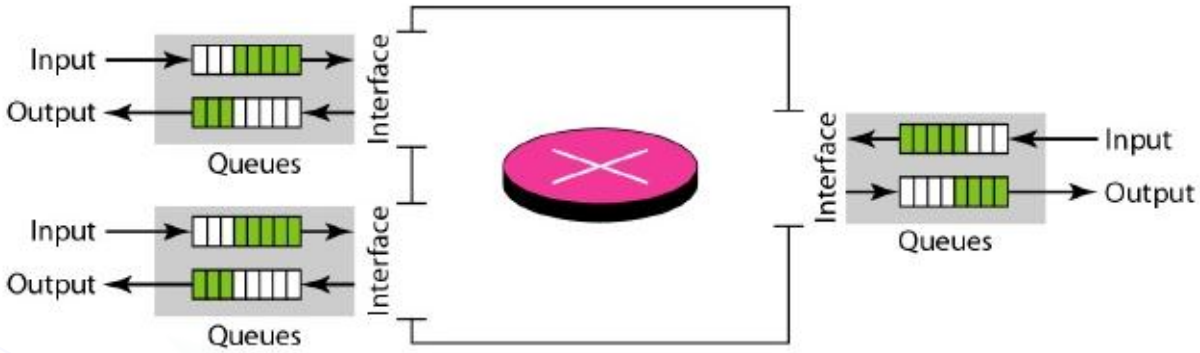
TCP CONGESTION CONTROL

- Congestion in a network may occur if the **load** on the network-the number of packets sent to the network-is greater than the ***capacity*** of the network-the number of packets a network can handle.
- **Congestion control** refers to the mechanisms and techniques to control the congestion and keep the load below the capacity.
- Congestion in a network or internetwork occurs because routers and switches have queues-buffers that hold the packets before and after processing.
- A router, for example, has an input queue and an output queue for each interface. When a packet arrives at the incoming interface, it undergoes **three steps** before departing.

  1. The packet is put at the end of the input queue while waiting to be checked.

2. The processing module of the router removes the packet from the input queue once it reaches the front of the queue and uses its routing table and the destination address to find the route.
3. The packet is put in the appropriate output queue and waits its tum to be sent.

Queues in a router



Two Issues

1. If the rate of packet arrival is higher than the packet processing rate, the input queues become longer and longer.
2. If the packet departure rate is less than the packet processing rate, the output queues become longer and longer.

Network Performance

Congestion control involves two factors that measure the performance of a network

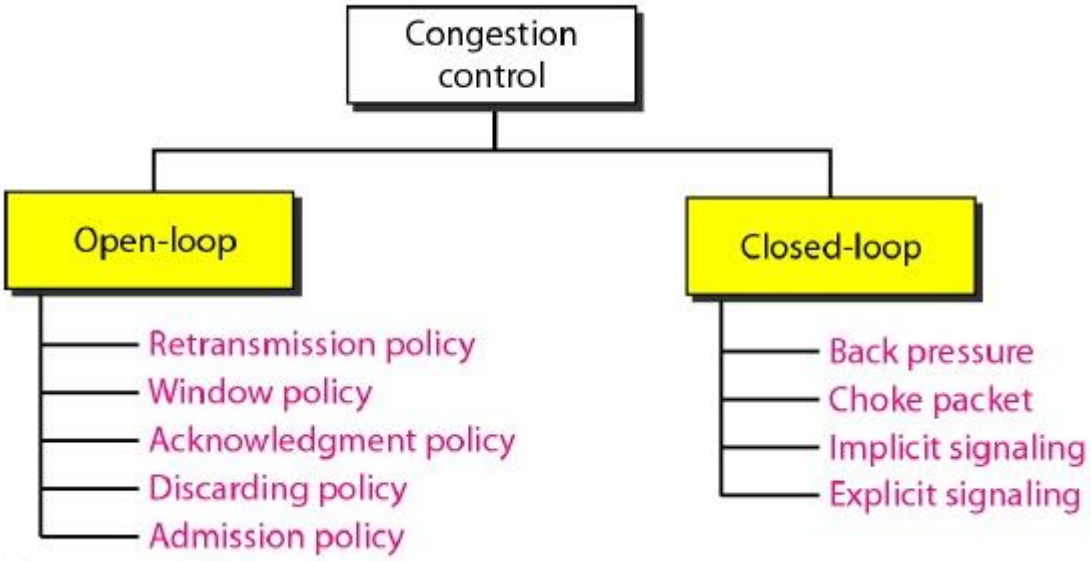
1. **Delay**
2. **Throughput.**

Delay Versus Load:

- when the load is much less than the capacity of the network, the delay is at a minimum.

Throughput Versus Load:

- Throughput in a network is defined as the number of packets passing through the network in a unit of time.
- When the load is below the capacity of the network, the throughput increases proportionally with the *load.*



Open-Loop Congestion Control

In open-loop congestion control, policies are applied to prevent congestion before it happens. In these mechanisms, congestion control is handled by either the source or the destination.

Retransmission Policy

If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion.

Window Policy

The Selective Repeat window is better than the Go-Back-N window for congestion control.

Acknowledgment Policy

Sending fewer acknowledgments means imposing less load on the network.

Discarding Policy

The policy is to discard less sensitive packets when congestion is likely to happen

Admission Policy

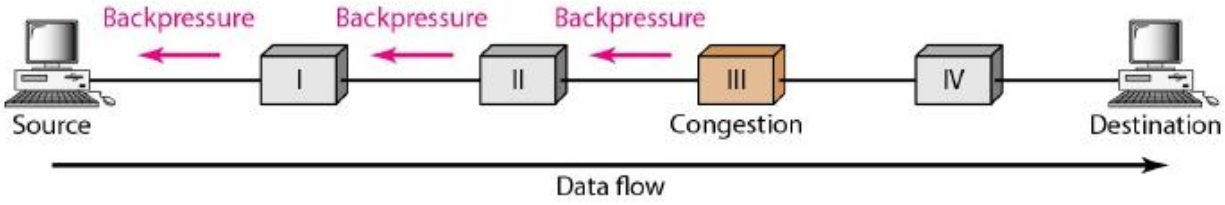
A router can deny establishing a connection if there is congestion in the network or if there is a possibility of future congestion.

Closed-Loop Congestion Control

Closed-loop congestion control mechanisms try to alleviate congestion after it happens.

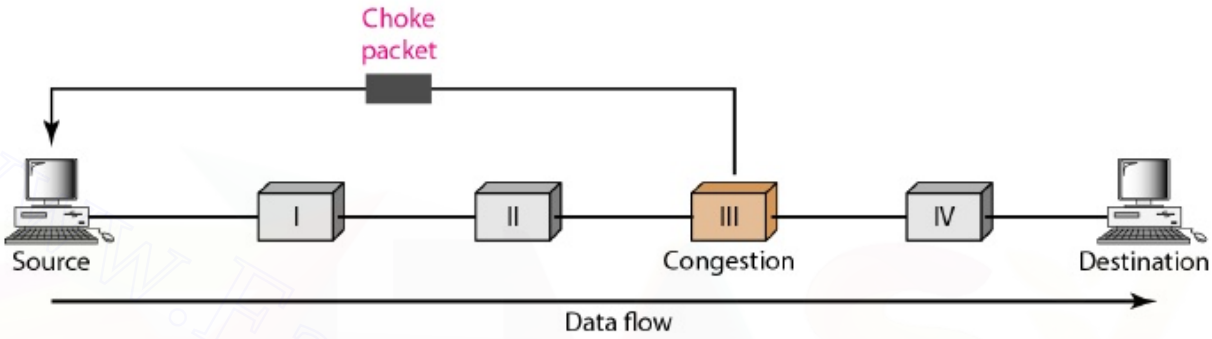
Backpressure

Congested node stops receiving data from the immediate upstream node or nodes



Choke Packet

- In backpressure, the warning is from one node to its upstream node
- In the choke packet method, the warning is **from the router**, which has encountered congestion, **to the source station directly.**



Implicit Signaling

- The source guesses that there is a congestion somewhere in the network from other symptoms.
- For example, when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested.

Explicit Signaling

- The node that experiences congestion can explicitly send a signal to the source or destination.
- In the explicit signaling method, the signal is included in the packets that carry data.

CONGESTION CONTROL IN TCP

Congestion Window

The sender's window size is determined not only by the receiver but also by congestion in the network.

Actual window size = minimum (rwnd, cwnd)

Congestion Policy

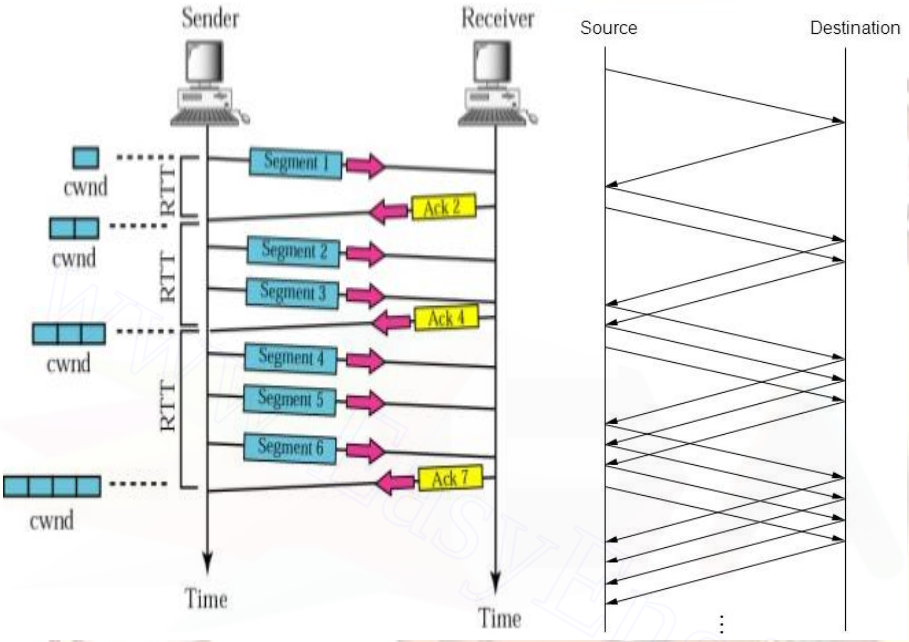
TCP's general policy for handling congestion is based on **three phases**

1. Additive Increase and Multiplicative Decrease (AIMD)
2. Slow start (Exponential Increase)
3. Fast Retransmission and Fast Recovery

ADDITIVE INCREASE AND MULTIPLICATIVE DECREASE (AIMD)

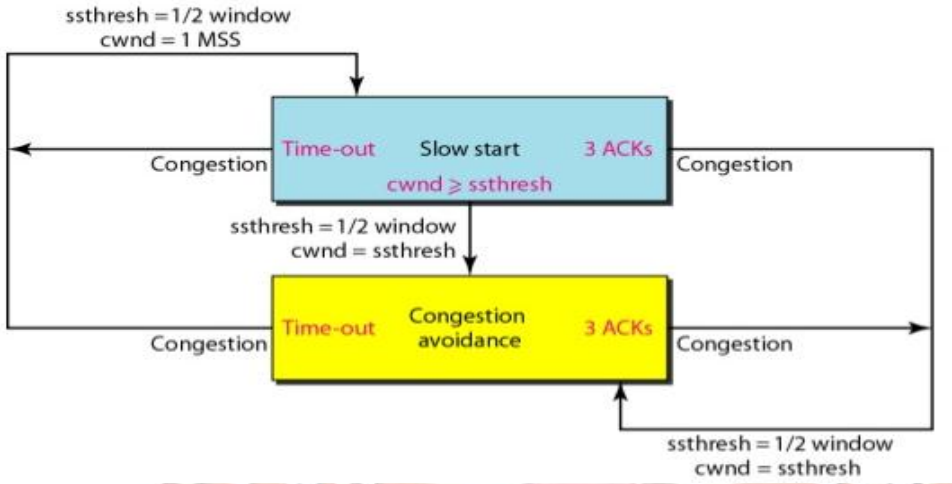
Congestion Avoidance: **Additive Increase**

In the congestion avoidance algorithm, the size of the congestion window increases additively until congestion is detected.



Congestion Detection: **Multiplicative Decrease**

- If congestion occurs, the congestion window size must be decreased.
- The only way the sender can guess that congestion has occurred is by the need to retransmit a segment.
- Retransmission can occur in one of two cases:
  1. when a timer times out
  2. when three ACKs are received.
- In both cases, the **size of the threshold is dropped to one-half**, a multiplicative decrease.
- An implementations reacts to congestion detection in one of the following ways:
  - If detection is by time-out, a new ***slow-start* phase** starts.
  - If detection is by three ACKs, a new ***congestion avoidance* phase** starts.

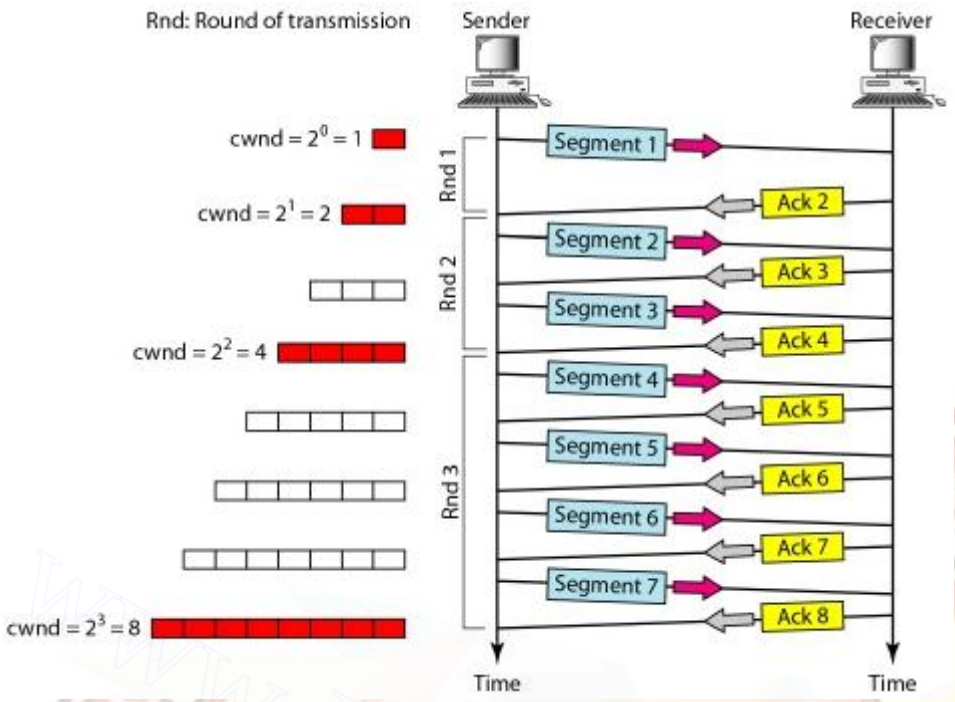


- Lets consider the maximum window size is 32 segments.
- The threshold is set to 16 segments (one-half of the maximum window size).
- In the *slow-start* phase the window size starts from 1 and grows exponentially until it reaches the threshold.
- After it reaches the threshold, the *congestion avoidance (additive increase)* procedure allows the window size to increase linearly until a timeout occurs or the maximum window size is reached. In the Figure, the time-out occurs when the window size is 20.
- At this moment, the *multiplicative decrease* procedure takes over and reduces the threshold to one-half of the previous window size.
- The previous window size was 20 when the time-out happened so the new threshold is now 10.
- TCP moves to slow start again and starts with a window size of 1, and TCP moves to additive increase when the new threshold is reached.
- When the window size is 12, a three-ACKs event happens.
- The multiplicative decrease procedure takes over again.
- The threshold is set to 6 and TCP goes to the additive increase phase this time.
- It remains in this phase until another time-out or another three ACKs happen.

SLOW START / EXPONENTIAL INCREASE

In the slow-start algorithm, the size of the congestion window increases exponentially until it reaches a threshold.

- The size of the **congestion window *(cwnd)*** starts with one **maximum segment size (MSS).**
- The size of the window increases one MSS each time an acknowledgment is received.
- As the name implies, the window starts slowly, but grows exponentially.
- The sender starts with ***cwnd* =1 MSS.**
- This means that the sender can send only one segment.
- After receipt of the acknowledgment for segment 1, the size of the congestion window is increased by 1, which means that *cwnd* is now 2.
- Now two more segments can be sent.
- When each acknowledgment is received, the size of the window is increased by 1 MSS.
- When all seven segments are acknowledged, *cwnd* = 8.



- We find that the rate is exponential as shown below:

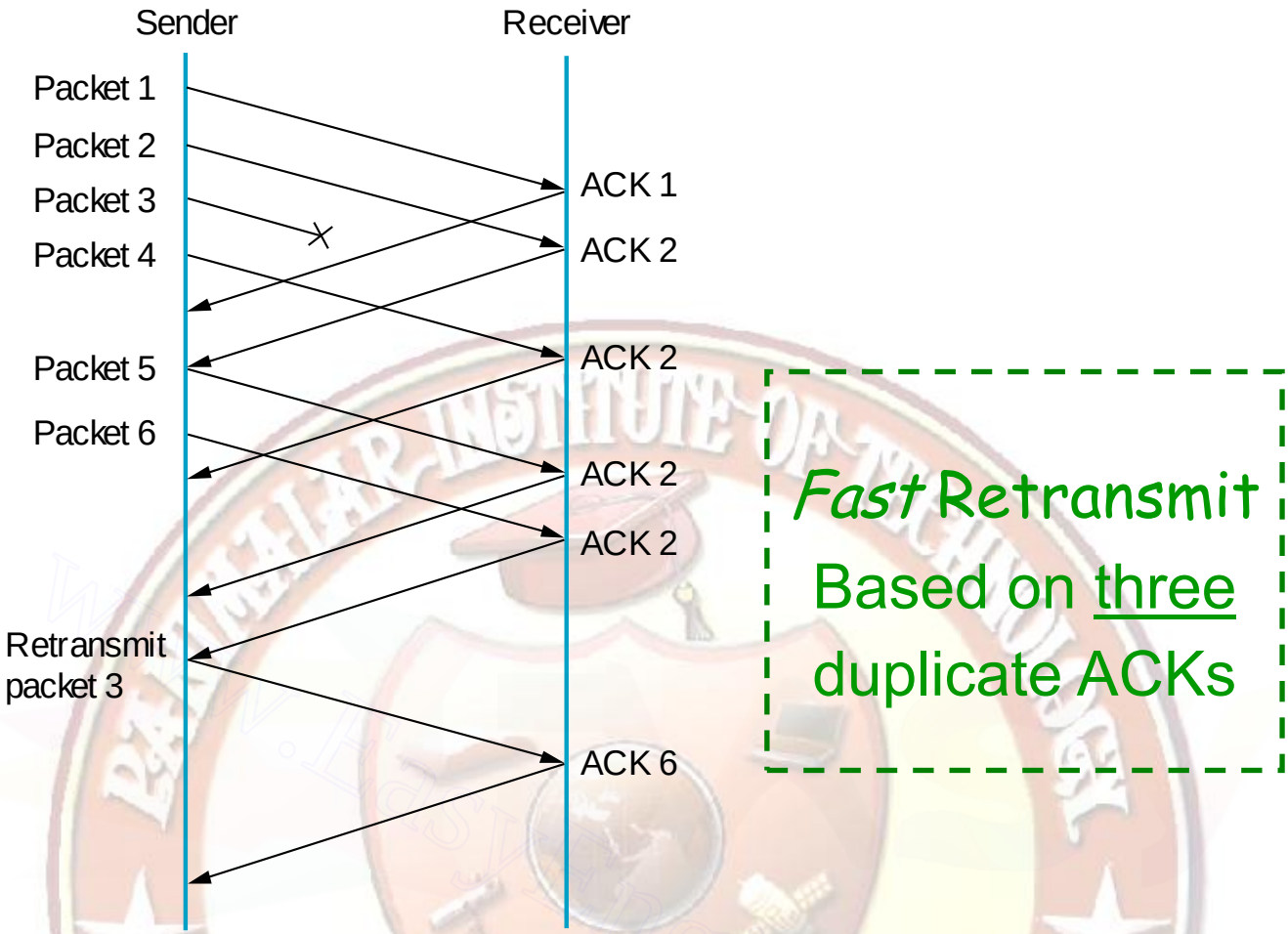
  Start → *cwnd=l*
  After round 1 → *cwnd=2^1 =2*
  After round 2 → *cwnd=2^2 =4*
  After round 3 → *cwnd=2^3 =8*
- Slow start cannot continue indefinitely.
- There must be a threshold to stop this phase.
- The sender keeps track of a variable named ***ssthresh*** **(slow-start threshold).**
- When the size of window in bytes reaches this threshold, slow start stops and the next phase starts.
- In most implementations the value of *ssthresh* is 65,535 bytes.

FAST RETRANSMISSION AND FAST RECOVERY

Fast Retransmit :

Fast retransmit is a heuristic that sometimes triggers the retransmission of a dropped packet sooner than the regular timeout mechanism. The fast retransmit mechanism does not replace regular timeouts.

Upon receipt of *three* duplicate ACKs, the TCP Sender retransmits the lost packet.Every time a data packet arrives at the receiving side, the receiver responds with an acknowledgment,even if this sequence number has already been acknowledged. TCP resends the same acknowledgment it sent the last time



Fast Recovery:

When fast retransmit detects three duplicate ACKs, start the recovery process from congestion avoidance region and use ACKs in the pipe to pace the sending of packets.

TCP CONGESTION AVOIDANCE

It is a mechanism to avoid producing congestion.

Approaches to congestion avoidance

1. **DECbit (Destination Experiencing Congestion Bit)**
   - Developed for the Digital Network Architecture

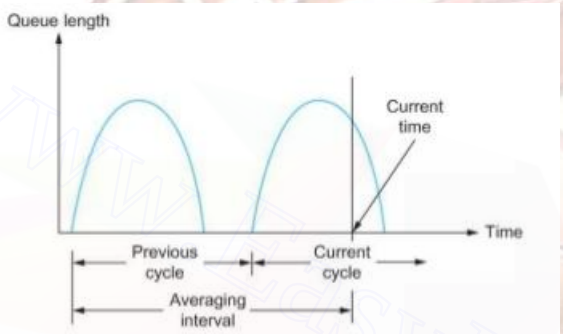
   Basic idea
   - More evenly split the responsibility for congestion control between the routers and the end nodes.
   - Each router monitors the load it is experiencing and explicitly notifies the end nodes when congestion is about to occur.
   - implemented by setting a binary congestion bit in the packets that flow through the router, hence the name *DECbit*.
   - The destination host then copies this congestion bit into the ACK it sends back to the source.
   - the source adjusts its sending rate so as to avoid congestion

- One bit allocated in packet header
- Any router experiencing congestion sets bit.
- Destination returns bit to source.
- Source adjusts rate based on bits.
- Note that responsibility is shared.

- **Routers identify congestion and Hosts act to avoid congestion.**

Router

- Monitors length over last busy + idle cycle
- Sets congestion bit if average queue length is greater than 1, when packet arrives
- Attempts to balance throughput against delay
  - smaller values result in more idle time
  - larger values result in more queueing delay



Computing average queue length at a router

End Hosts

- Destination echoes congestion bit back to source
- Source records how many packets resulted in set bit
- If less than 50% of last window had bit set

  Increase CongestionWindow by 1 packet
- If 50% or more of last window had bit set

  Decrease CongestionWindow by 0.875 percent

2.Random Early Detection (RED) gateways

- Developed for use with TCP
- **Basic idea:**

  Implicitly notifies the source of congestion by dropping one of its packets.
- The source is effectively notified by the subsequent timeout or duplicate ACK.
- RED is designed to be used in conjunction with TCP, which currently detects congestion by means of timeouts
- Router drops a few packets before it has exhausted its buffer space completely, so as to cause the source to slow down,
- drop each arriving packet with some *drop probability* whenever the queue length exceeds some *drop level*. This idea is called *early random drop*.

RED computes an average queue length using a weighted running average similar to the one used in the original TCP timeout computation.

$$AvgLen = (1-Weight)\times AvgLen + Weight \times SampleLen$$

where 0 < Weight < 1 and SampleLen is the length of the queue when a sample measurement is made.

RED has two queue length thresholds that trigger certain activity: MinThreshold and MaxThreshold. When a packet arrives at the gateway, RED compares the current AvgLen with these two thresholds, according to the following rules:

```
if AvgLen _ MinThreshold
    !queue the packet
if MinThreshold < AvgLen < MaxThreshold
    !calculate probability P
    !drop the arriving packet with probability P
if MaxThreshold _ AvgLen
    !drop the arriving packet
```

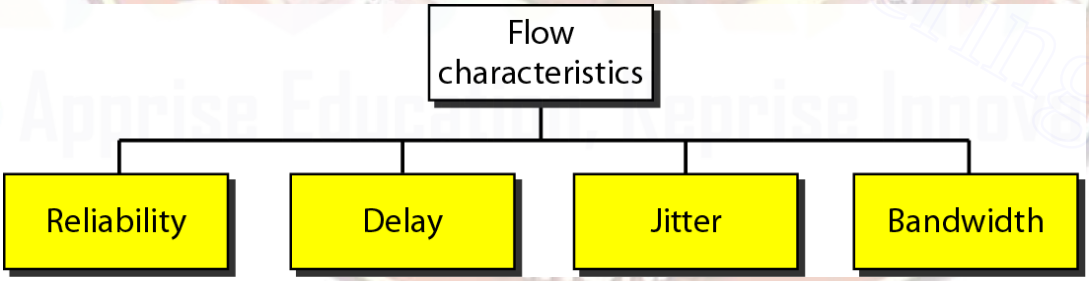
If the average queue length is smaller than the lower threshold, no action is taken, and if the average queue length is larger than the upper threshold, then the packet is always dropped. If the average queue length is between the two thresholds, then the newly arriving packet is dropped with some probability P

QUALITY OF SERVICE

QoS can be defined in terms of its flow characteristics.

Quality of service (QoS) refers to a network's ability to achieve maximum bandwidth and deal with other network performance elements like Reliability, Delay and Jitter.

Flow characteristics



Reliability

Reliability is a characteristic that a flow needs. Lack of reliability means losing a packet or acknowledgment, which entails retransmission. However, the sensitivity of application programs to reliability is not the same. For example, it is more important that electronic mail, file transfer, and Internet access have reliable transmissions than telephony or audio conferencing.

Delay

Source-to-destination delay is another flow characteristic. Again applications can tolerate delay in different degrees. In this case, telephony, audio conferencing, video conferencing, and remote log-in need minimum delay, while delay in file transfer or e-mail is less important.

Jitter

Jitter is the variation in delay for packets belonging to the same flow.

Bandwidth

It refers to the amount of data transmitted per second through the transmission medium.

TECHNIQUES TO IMPROVE QoS

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

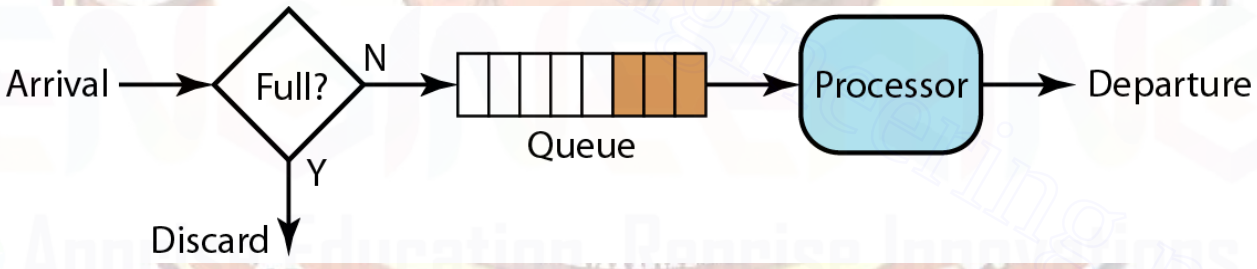
SCHEDULING

Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner.

- ***FIFO queue***
- ***Priority queuing***
- ***Weighted fair queuing***

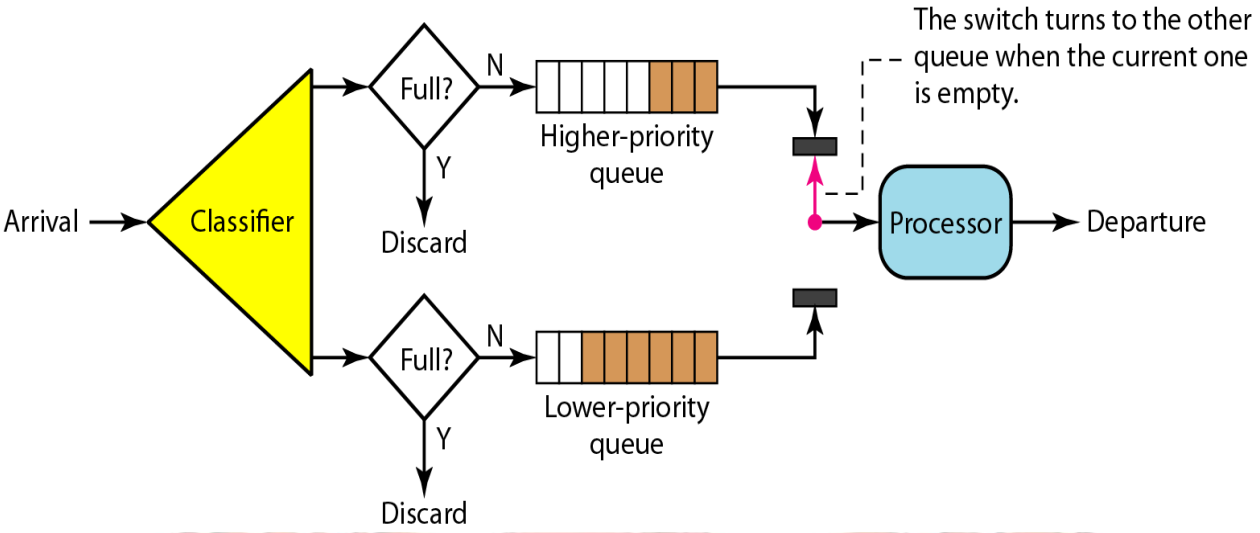
FIFO queue

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



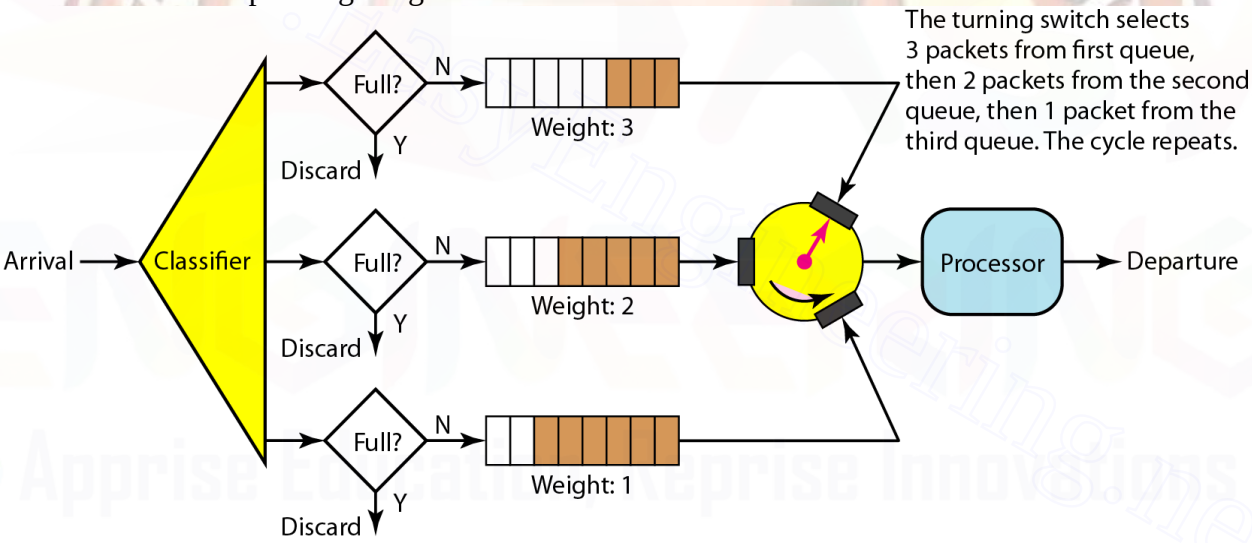
Priority queuing

In priority queuing, packets are first assigned to a priority class. Each priority class has its own queue. The packets in the highest-priority queue are processed first. Packets in the lowest-priority queue are processed last.



Weighted fair queuing

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



TRAFFIC SHAPING

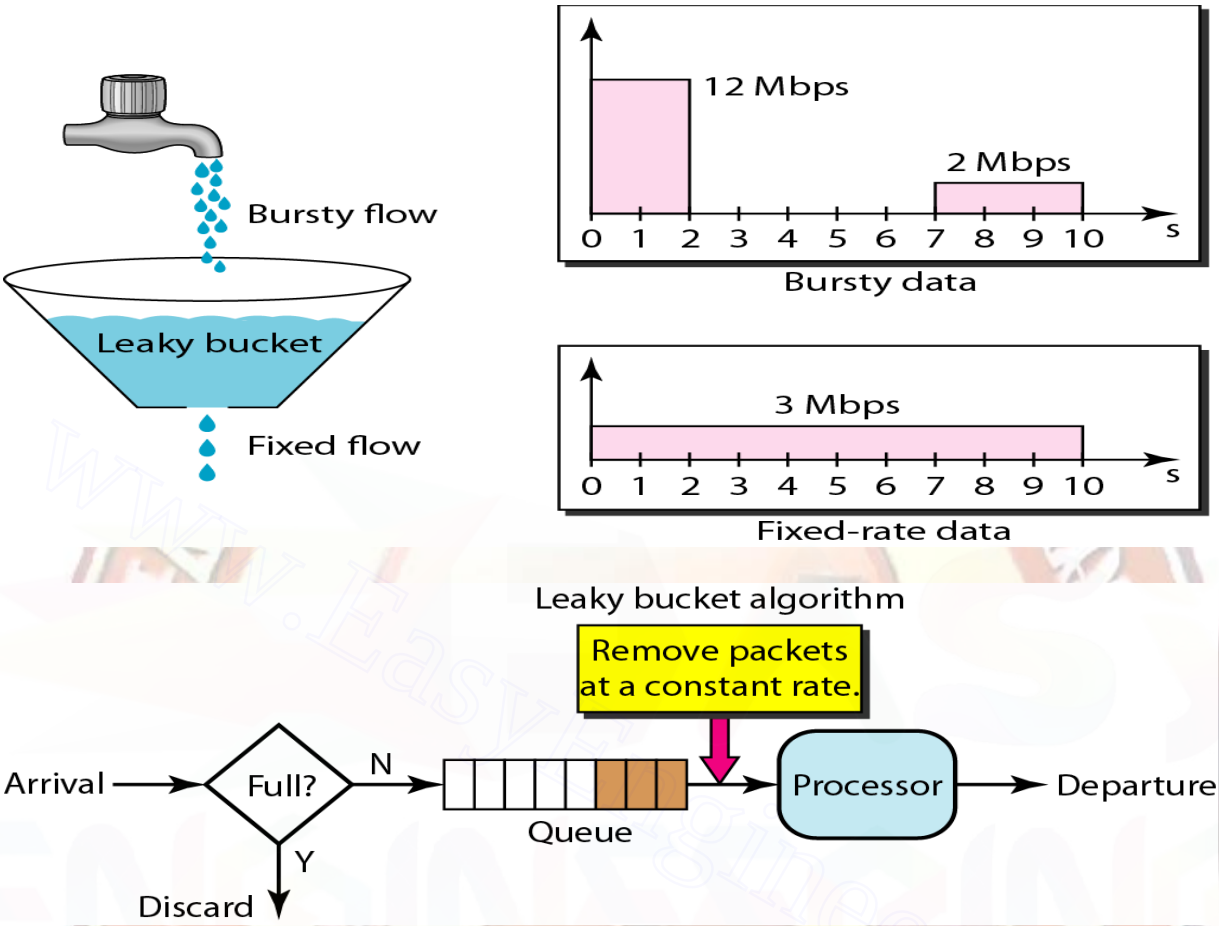
It is a mechanism to control the amount and the rate of the traffic sent to the network. Two techniques can shape traffic: **leaky bucket and token bucket.**

Leaky bucket

A leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packets if the bucket is full.

- If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket.
- The rate at which the water leaks does not depend on the rate at which the water is input to the bucket unless the bucket is empty.

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



Leaky bucket implementation

The following is an algorithm for variable-length packets:

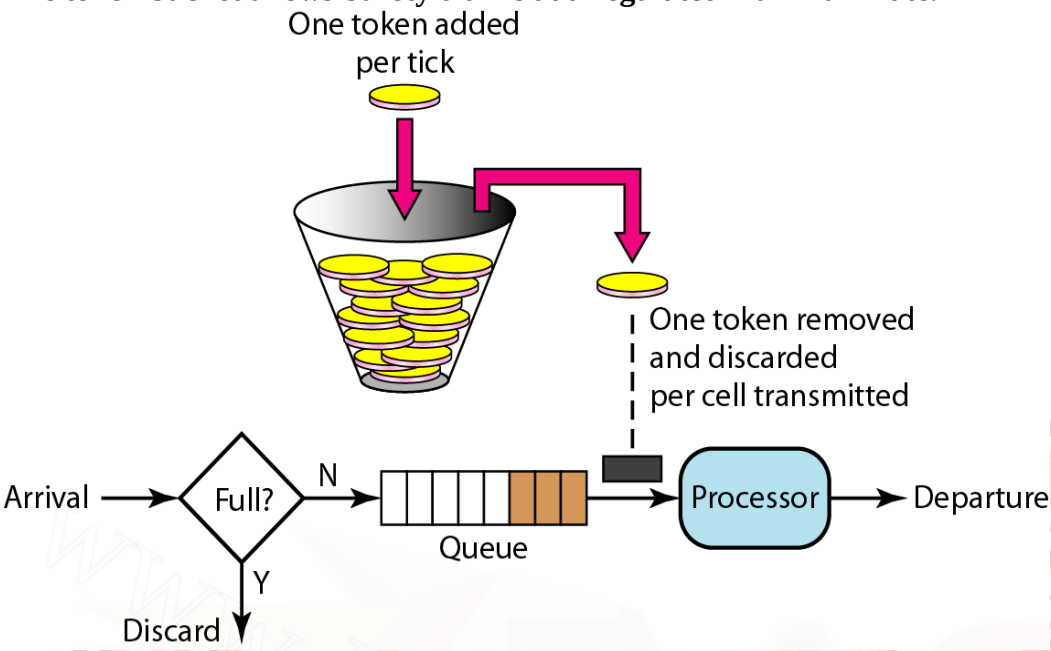
1. Initialize a counter to *n* at the tick of the clock.
2. If *n* is greater than the size of the packet, send the packet and decrement the counter by the packet size. Repeat this step until *n* is smaller than the packet size.
3. Reset the counter and go to step 1.

Token bucket

- If the host has bursty data, the leaky bucket allows only an average rate. The time when the host was idle is not taken into account.
- In Token Bucket technique, for each tick of the clock, the system sends *n* tokens to the bucket.
- The system removes one token for every cell (or byte) of data sent.
- The token bucket can easily be **implemented with a counter**.
- The token is initialized to zero.
- Each time a token is added, the counter is incremented by 1.
- Each time a unit of data is sent, the counter is decremented by 1.

- When the counter is zero, the host cannot send data.

The token bucket allows bursty traffic at a regulated maximum rate.



RESOURCE RESERVATION

- A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on.
- The quality of service is improved if these resources are reserved beforehand.

ADMISSION CONTROL

- Admission control refers to the mechanism used by a router, or a switch, to accept or reject a flow based on predefined parameters called flow specifications.
- Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow.

QoS Models

- Integrated Services (Intserv)
- Differentiated Services (Diffserv)
- Resource ReSerVation Protocol (RSVP)

Integrated Services (Intserv)

- Intserv is a framework developed by the IETF to provide individualized QoS guarantees to individual application sessions.
- Two key features lie at the heart of Intserv:

Reserved resources:

A router is supposed to know what amounts of its resources (buffers, link b/w) are already reserved for ongoing sessions

Call setup:

A session requiring QoS guarantees must first be able to reserve sufficient resources at each network router on its source-to-destination path to ensure that its end-to-end QoS requirement is met.

- The Intserv architecture defines two major service classes
- **Guaranteed service**
  - Targets hard real-time applications.
  - User specifies traffic characteristics and a service requirement.
  - Requires admission control at each of the routers.
  - Can mathematically guarantee bandwidth, delay, and jitter.
- **Controlled load.**
  - Targets applications that can adapt to network conditions within a certain performance window.
  - User specifies traffic characteristics and bandwidth.
  - Requires admission control at each of the routers.
  - Guarantee not as strong as with the guaranteed service.
    - e.g., measurement-based admission control.
  - Best effort

Session must first declare its QoS requirement and characterize the traffic it will send through the network

- **R-spec**: defines the QoS being requested by receiver (e.g., rate *r*)
- **T-spec**: defines the traffic characteristics of sender (e.g., leaky bucket with rate *r* and buffer size *b*).
- A signaling protocol is needed to carry the R-spec and T-spec to the routers where reservation is required; RSVP is a leading candidate for such signaling protocol.

Call Admission: routers will admit calls based on their R-spec and T-spec and base on the current resource allocated at the routers to other calls.

Differentiated Services: Diffserv

Difficulties associated with the Intserv model of per-flow reservation of resources:

Intended to address the following difficulties with Intserv and RSVP;

- **Scalability**: maintaining states by routers in high speed networks is difficult due to the very large number of flows
- **Flexible Service Models**: Intserv has only two classes, want to provide more qualitative service classes; want to provide 'relative' service distinction (Platinum, Gold, Silver, …)
- **Simpler signaling**: (than RSVP) many applications and users may only want to specify a more qualitative notion of service

- Do fine-grained enforcement only at the edge of the network.
  - Typically slower links at edges
  - E.g., mail sorting in post office
- Label packets with a field.
  - E.g., a priority stamp
- The core of the network uses only the type field for QoS management.

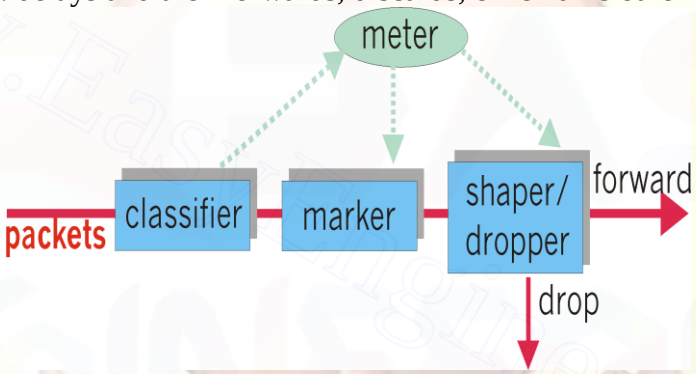
- Small number of types with well defined forwarding behavior
- Can be handled fast

Example: expedited service versus best effort

- Diffserv defines an architecture and a set of forwarding behaviors.
  - It is up to the service providers to define and implement end-to-end services on top of this architecture.
  - Offers a more flexible service model; different providers can offer different service.
- One of the main motivations for Diffserv is scalability.
  - Keep the core of the network simple.
- Focus of Diffserv is on supporting QoS for flow aggregates.

Edge Router/Host Functions

- **Classification**: marks packets according to classification rules to be specified.
- **Metering:** checks whether the traffic falls within the negotiated profile.
- **Marking**: marks traffic that falls within profile.
- **Conditioning**: delays and then forwards, discards, or remarks other traffic.



RSVP – Reservation protocol

- Used on connectionless networks.
  - Should not replicate routing functionality.
  - Should co-exist with route changes.
- Support for multicast.
  - Different receivers have different capabilities and want different QOS.
  - Changes in group membership should not be expensive.
  - Reservations should be aggregate – I.e. each receiver in group should not have to reserve.
  - Should be able to switch allocated resource to different senders.
- Limit control overhead.
- Modular design – should be generic "signaling" protocol.
- Result:
  - Receiver-oriented
  - Soft-state

Receiver-Initiated Reservation

- Receiver initiates reservation by sending a reservation over the sink tree.
  - Assumes multicast tree has been set up previously.
    - Also uses receiver-initiated mechanism.
  - Hooks up with the reserved part of the tree.
  - How far the request has to travel to the source depends on the level of service requested.
  - Uses existing routing protocol, but routers have to store the sink tree (reverse path from forwarding path).
- **Properties:**
  - Scales well: can have parallel independent connect and disconnect actions – single shared resource required.
  - Supports receiver heterogeneity: reservation specifies receiver requirements and capabilities.

RSVP Service Model

- Make reservations for simplex data streams.
- Receiver decides whether to make reservation
  - Control messages in IP datagrams (proto #46).
- PATH/RESV messages sent periodically to refresh soft state.
- One pass:
  - Failed requests return error messages - receiver must try again.
  - No end to end ack for success.

Basic Message Types:

- PATH message
- RESV message
- CONFIRMATION message
  - Generated only upon request.
  - Unicast to receiver when RESV reaches node with established state.
- TEARDOWN message
- ERROR message (if PATH or RESV fails)

UNIT V APPLICATION LAYER

Traditional applications -Electronic Mail (SMTP, POP3, IMAP, MIME) – HTTP – Web Services – DNS – SNMP

Application Layer

The application layer enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as electronic mail, file access and transfer, access to system resources, surfing the World Wide Web, and network management. The application layer is responsible for providing services to the user.

ELECTRONIC MAIL

Electronic mail allows a message to include text, audio, and video. It also allows one message to be sent to one or more recipients.
E-mail system includes three main components:

- User Agent (UA)
- Message Transfer Agent (MTA)
- Message Access Agent (MAA)

Architecture

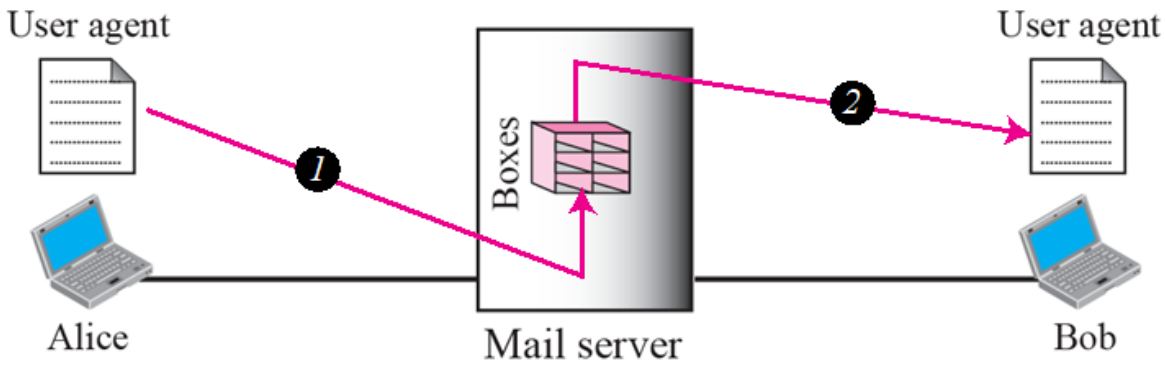
To explain the architecture of e-mail, there are four scenarios.
The fourth scenario is the most common in the exchange of email.

First Scenario

In the first scenario, the sender and the receiver of the e-mail are users (or application programs) on the same system; they are directly connected to a shared system. The administrator has created one mailbox for each user where the received messages are stored. A *mailbox* is part of a local hard drive, a special file with permission restrictions.

Only the owner of the mailbox has access to it. When Alice, a user, needs to send a message to Bob, another user, Alice runs a *user agent (UA)* program to prepare the message and store it in Bob's mailbox. The message has the sender and recipient mailbox addresses (names of files).

Bob can retrieve and read the contents of his mailbox at his convenience, using a user agent. This is similar to the traditional memo exchange between employees in an office. There is a mailroom where each employee has a mailbox with his or her name on it.

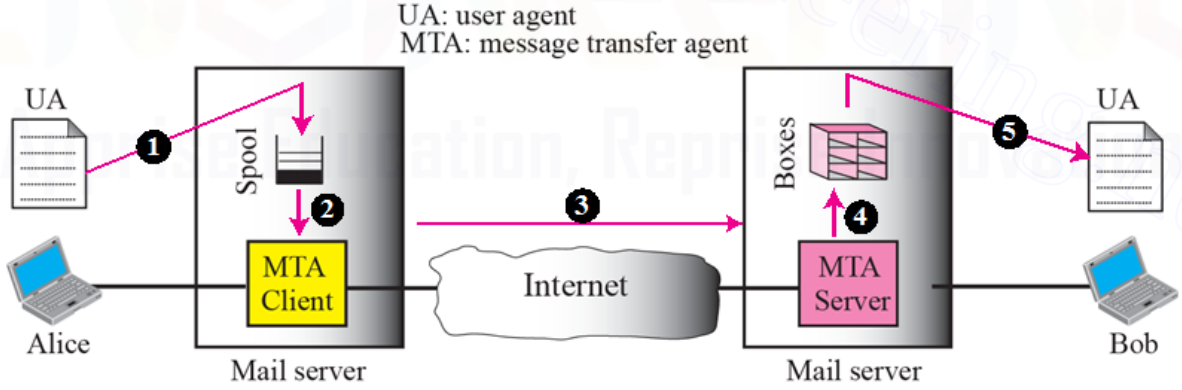


When Alice needs to send a memo to Bob, she writes the memo and inserts it into Bob's mailbox. When Bob checks his mailbox, he finds Alice's memo and reads it.

When the sender and the receiver of an e-mail are on the same system, we need only two user agents.

Second Scenario

In the second scenario, the sender and the receiver of the e-mail are users (or application programs) on two different systems. The message needs to be sent over the Internet. Here we need user agents (UAs) and message transfer agents (MTAs).

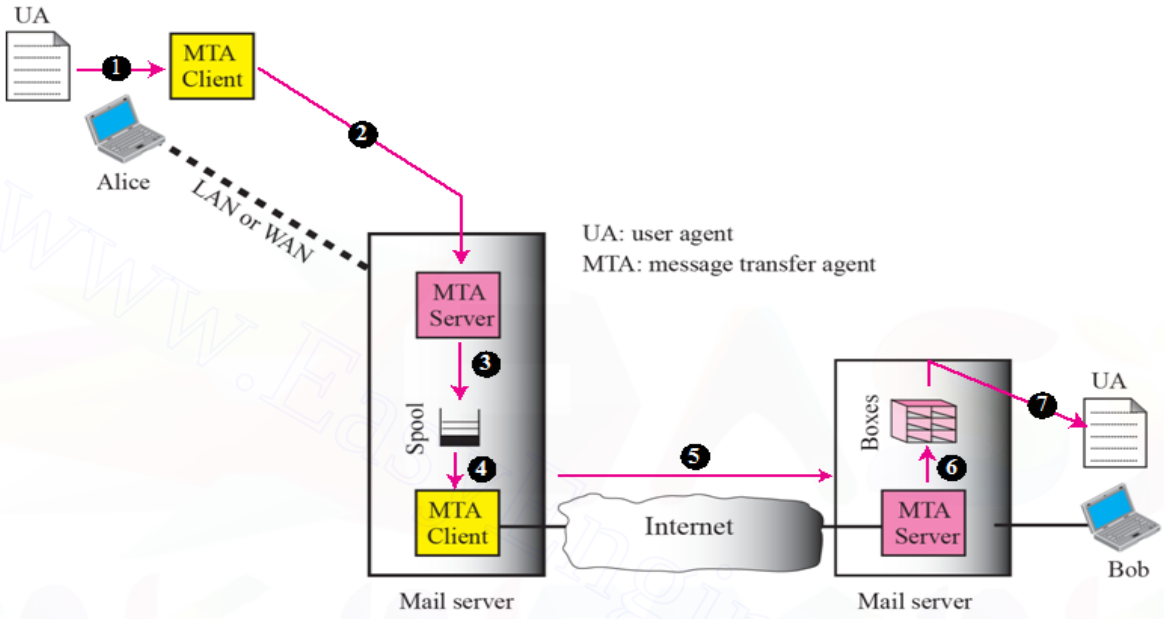


Alice needs to use a user agent program to send her message to the system at her own site. The system (sometimes called the mail server) at her site uses a queue to store messages waiting to be sent. Bob also needs a user agent program to retrieve messages stored in the mailbox of the system at his site. The message, however, needs to be sent through the Internet from Alice's site to Bob's site. Here two message transfer agents are needed: one 'client and one server.

When the sender and the receiver of an e-mail are on different systems, we need two UAs and a pair of MTAs (client and server).

Third Scenario

In the third scenario, Bob is directly connected to his system. Alice is separated from her system. Either Alice is connected to the system via a point-to-point WAN, such as a dial-up modem, a DSL, or a cable modem; or she is connected to a LAN in an organization that uses one mail server for handling e-mails-all users need to send their messages to this mail server.



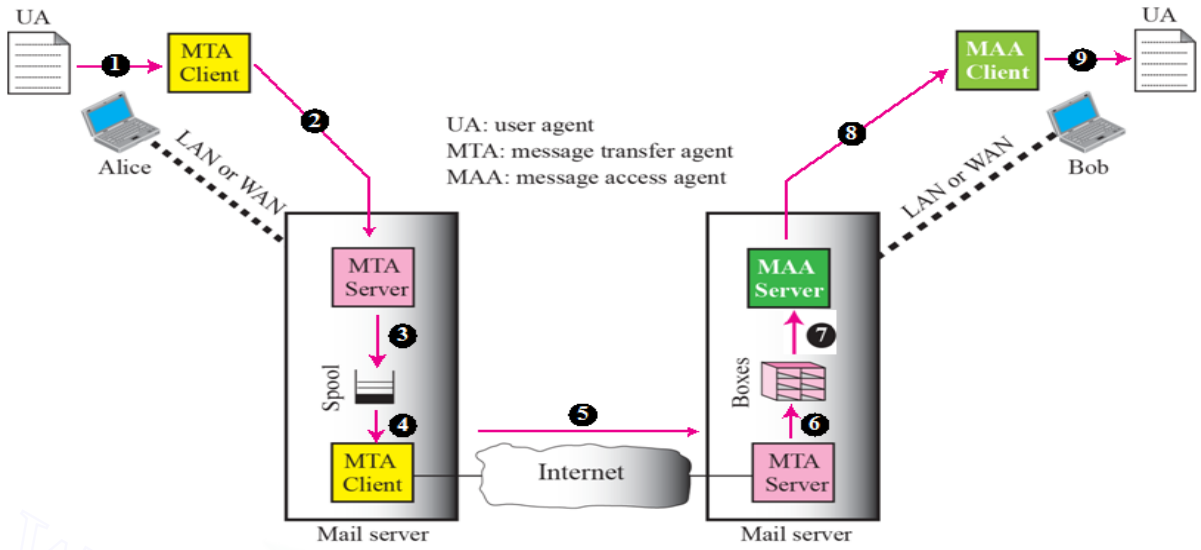
Alice still needs a user agent to prepare her message. She then needs to send the message through the LAN or WAN. This can be done through a pair of message transfer agents (client and server). Whenever Alice has a message to send, she calls the user agent which, in tum, calls the MTA client. The MTA client establishes a connection with the MTA server on the system, which is running all the time.

The system at Alice's site queues all messages received. It then uses an MTA client to send the messages to the system at Bob's site; the system receives the message and stores it in Bob's mailbox.

When the sender is connected to the mail server via a LAN or a WAN, we need two UA*s* and two pairs of MTAs (client and server).

Fourth Scenario

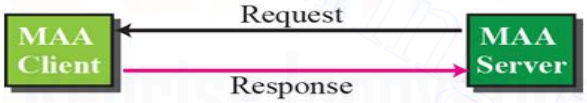
In the fourth and most common scenario, Bob is also connected to his mail server by a WAN or a LAN. After the message has arrived at Bob's mail server, Bob needs to retrieve it. Here, we need another set of client/server agents, which we call message access agents (MAAs). Bob uses an MAA client to retrieve his messages. The client sends a request to the MAA server, which is running all the time, and requests the transfer of the messages.



There are two important points here. First, Bob cannot bypass the mail server and use the MTA server directly. To use MTA server directly, Bob would need to run the MTA server all the time because he does not know when a message will arrive.

Second, note that Bob needs another pair of client/server programs: message access programs. MTA client/server program is a *push* program: the client pushes the message to the server. Bob needs a *pull* program. The client needs to pull the message from the server.

a. Client pushes messages



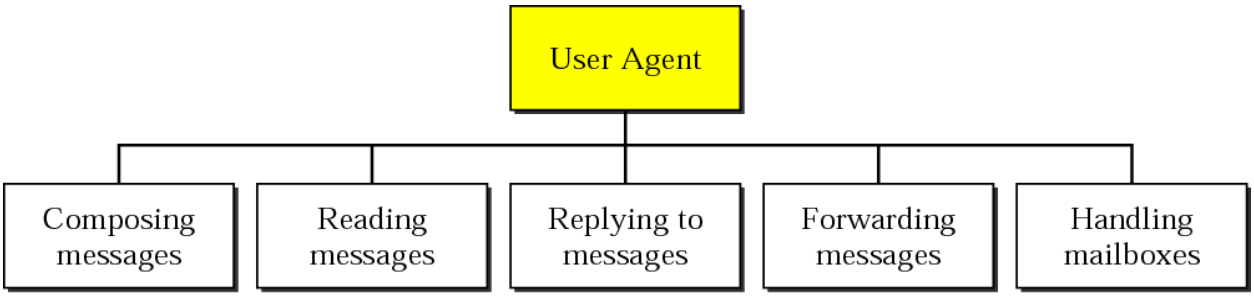
b. Client pulls messages

User Agent

The first component of an electronic mail system is the user agent *(VA).* It provides service to the user to make the process of sending and receiving a message easier.

Services Provided by a User Agent

A user agent is a software package (program) that composes, reads, replies to, and forwards messages. It also handles mailboxes.



Composing Messages:
A user agent helps the user compose the e-mail message to be sent out. Most user agents provide a template on the screen to be filled in by the user.

Reading Messages:
The second duty of the user agent is to read the incoming messages. When a user invokes a user agent, it first checks the mail in the incoming mailbox. Most user agents show a one-line summary of each received mail. Each e-mail contains the following fields.

1. A number field.
2. A flag field that shows the status of the mail such as new, already read but not replied to, or read and replied to.
3. The size of the message.
4. The sender.
5. The optional subject field.

Replying to Messages:
After reading a message, a user can use the user agent to reply to a message. A user agent usually allows the user to reply to the original sender or to reply to all recipients of the message. The reply message may contain the original message (for quick reference) and the new message.

Forwarding Messages*:*
Replying is defined as sending a message to the sender or recipients of the copy. *Forwarding* is defined as sending the message to a third party. A user agent allows the receiver to forward the message, with or without extra comments, to a third party.

Handling Mailboxes:
A user agent normally creates two mailboxes: an inbox and an out box. Each box is a file with a special format that can be handled by the user agent. The inbox keeps all the received e-mails until they are deleted by the user. The out box keeps all the sent e-mails until the user deletes them. Most user agents today are capable of creating customized mailboxes.

User Agent Types
There are two types of user agents:
command-driven and GUI-based.

Command-Driven: A command-driven user agent normally accepts a one-character command from the keyboard to perform its task. For example, a user can type the character r, at the command prompt, to reply to the sender of the message, or type the character R to reply to the sender and all recipients. Some examples of command-driven user agents are *mail*, *pine*, and *elm*

GUI-Based: Modern user agents are GUI-based. They contain graphical-user interface (GUI) components that allow the user to interact with the software by using both the keyboard and the mouse. They have graphical components such as icons, menu bars, and windows that make the services easy to access. Some examples of GUI-based user agents are Eudora, Microsoft's Outlook, and Netscape.

Sending Mail:

To send mail, the user, through the UA, creates mail that looks very similar to postal mail. It has an *envelope* and a *message.*

(i)Envelope: The envelope usually contains the sender and the receiver addresses.

(ii)Message: The message contains the header and the body. The header of the message defines the sender, the receiver, the subject of the message, and some other information (such as encoding type). The body of the message contains the actual information to be read by the recipient.

Receiving Mail

The user agent is triggered by the user (or a timer). If a user has mail, the *UA* informs the user with a notice. If the user is ready to read the mail, a list is displayed in which each line contains a summary of the information about a particular message in the mailbox. The summary usually includes the sender mail address, the subject, and the time the mail was sent or received. The user can select any of the messages and display its contents on the screen.

Addresses

To deliver mail, a mail handling system must use an addressing system with unique addresses. In the Internet,

the address consists of two parts: a local part and a domain name, separated by an @ sign.

Local Part: The local part defines the name of a special file, called the user mailbox, where all the mail received for a user is stored for retrieval by the message access agent.

Domain Name: The second part of the address is the domain name. An organization usually selects one or more hosts to receive and send e-mail; the hosts are sometimes called *mail servers* or *exchangers.* The domain name assigned to each mail exchanger either comes from the DNS database or is a logical name (for example, the name of the organization).

Mailing List:

Electronic mail allows one name, an alias, to represent several different e-mail addresses; this is called a mailing list. Every time a message is to be sent, the system checks the recipient's name against the alias database; if there is a mailing list for the defined alias, separate messages, one for each entry in the list, must be prepared and handed to the MTA. If there is no mailing list for the alias, the name itself is the receiving address and a single message is delivered to the mail transfer entity.

****

MIME- MULTIPURPOSE INTERNET MAIL EXTENSIONS

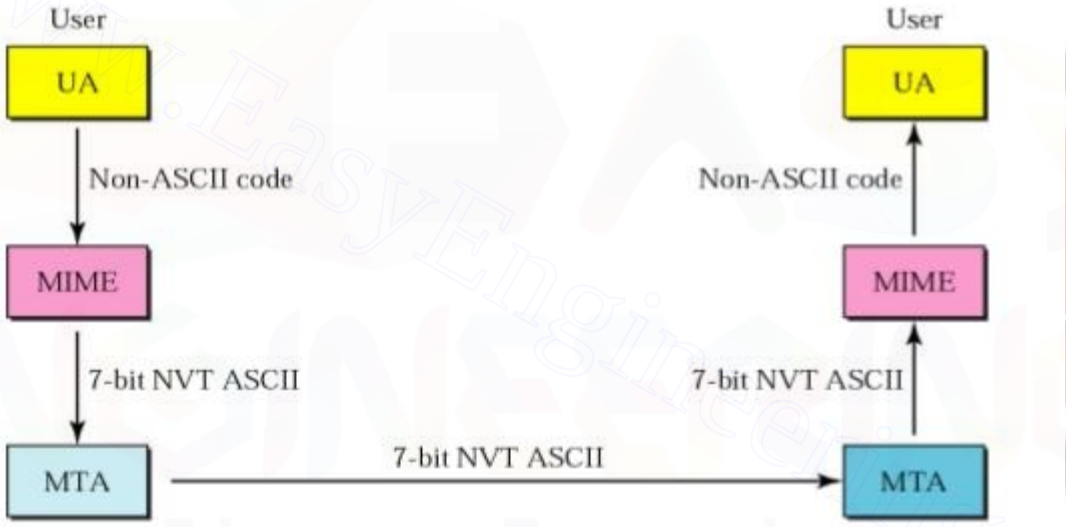
Electronic mail has a simple structure. Its simplicity, however, comes at a price. It can send messages only in NVT 7-bit ASCII format.

Limitations of E-mail:

- Cannot be used for languages that are not supported by 7-bit ASCII characters (such as French, German, Hebrew, Russian, Chinese, and Japanese).
- Cannot be used to send binary files or video or audio data.

Functions:

1. Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that allows non-ASCII data to be sent through e-mail.
2. MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers them to the client MTA to be sent through the Internet.
3. The message at the receiving side is transformed back to the original data.
4. MIME as a set of software functions that transforms non-ASCII data (stream of bits) to ASCII data and vice versa.



MIME defines five headers that can be added to the original e-mail header section to define the transformation parameters:

1. MIME-Version
2. Content-Type
3. Content-Transfer-Encoding
4. Content-Id
5. Content-Description

MIME headers

E-mail header
MIME-Version: 1.1 Content-Type: type/subtype Content-Transfer-Encoding: encoding type Content-Id: message id Content-Description: textual explanation of nontextual contents
E-mail body

MIME-Version: This header defines the version of MIME used. The current version is 1.1.
MIME-Version: 1.1

Content-Type: This header defines the type of data used in the body of the message. The content type and the content subtype are separated by a slash. Depending on the subtype, the header may contain other parameters.

Content-Type:<type J subtype; parameters>

Data Types and Subtypes in MIME

Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format (see Appendix E)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to Mixed, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

Content-Transfer-Encoding: This header defines the method used to encode the messages into 0 s and 1s for transport:

Content-Transfer-Encoding : <type>

Content-Transfer-Encoding

Type	*Description*
7bit	NVT ASCII characters and short lines
8bit	Non-ASCII characters and short lines
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data are encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters are encoded as an equal sign plus an ASCII code

Content-Id: This header uniquely identifies the whole message in a multiple-message environment.

Content-Id : id=<content-id>

Content-Description: This header defines whether the body is image, audio, or video.

Content-Description: <description>

****

SMTP – SIMPLE MAIL TRANSFER PROTOCOL
(Message Transfer Agent-MTA)

The actual mail transfer is done through message transfer agents. To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA. The formal protocol that defines the MTA client and server in the Internet is called the **Simple Mail Transfer Protocol (SMTP).**

Two pairs of MTA client/server programs are used in the most common situation. SMTP is used two times, between the sender and the sender's mail server and between the two mail servers. Another protocol is needed between the mail server and the receiver. SMTP defines how commands and responses must be sent back and forth. Each network is free to choose a software package for implementation.

Commands and Responses

SMTP uses commands and responses to transfer messages between an MTA client and an MTA server.

Each command or reply is terminated by a two-character (carriage return and line feed) end-of-line token.

Commands: Commands are sent from the client to the server. It consists of a keyword followed by zero or more arguments. SMTP defines 14 commands. The first five are mandatory; every implementation must support these five commands. The next three are often used and highly recommended. The last six are seldom used.

Keyword	*Argument(s)*
HELO	Sender's host name
MAIL FROM	Sender of the message
RCPTTO	Intended recipient of the message
DATA	Body of the mail
QUIT	
RSET	
VRFY	Name of recipient to be verified
NOOP	
TURN	
EXPN	Mailing list to be expanded
HELP	Command name

Keyword	*Argument(s)*
SEND FROM	Intended recipient of the message
SMOLFROM	Intended recipient of the message
SMALFROM	Intended recipient of the message

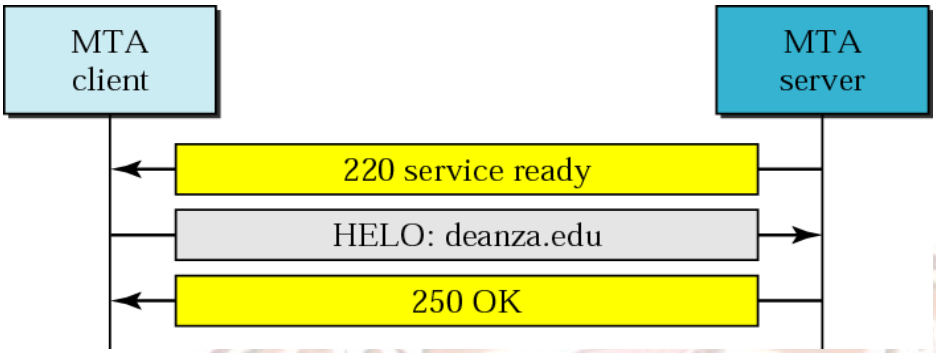
Responses are sent from the server to the client. A response is a three digit code that may be followed by additional textual information.

Code	*Description*
Positive Completion Reply	
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
Positive Intermediate Reply	
354	Start mail input
Transient Negative Completion Reply	
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted: insufficient storage
Permanent Negative Completion Reply	
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

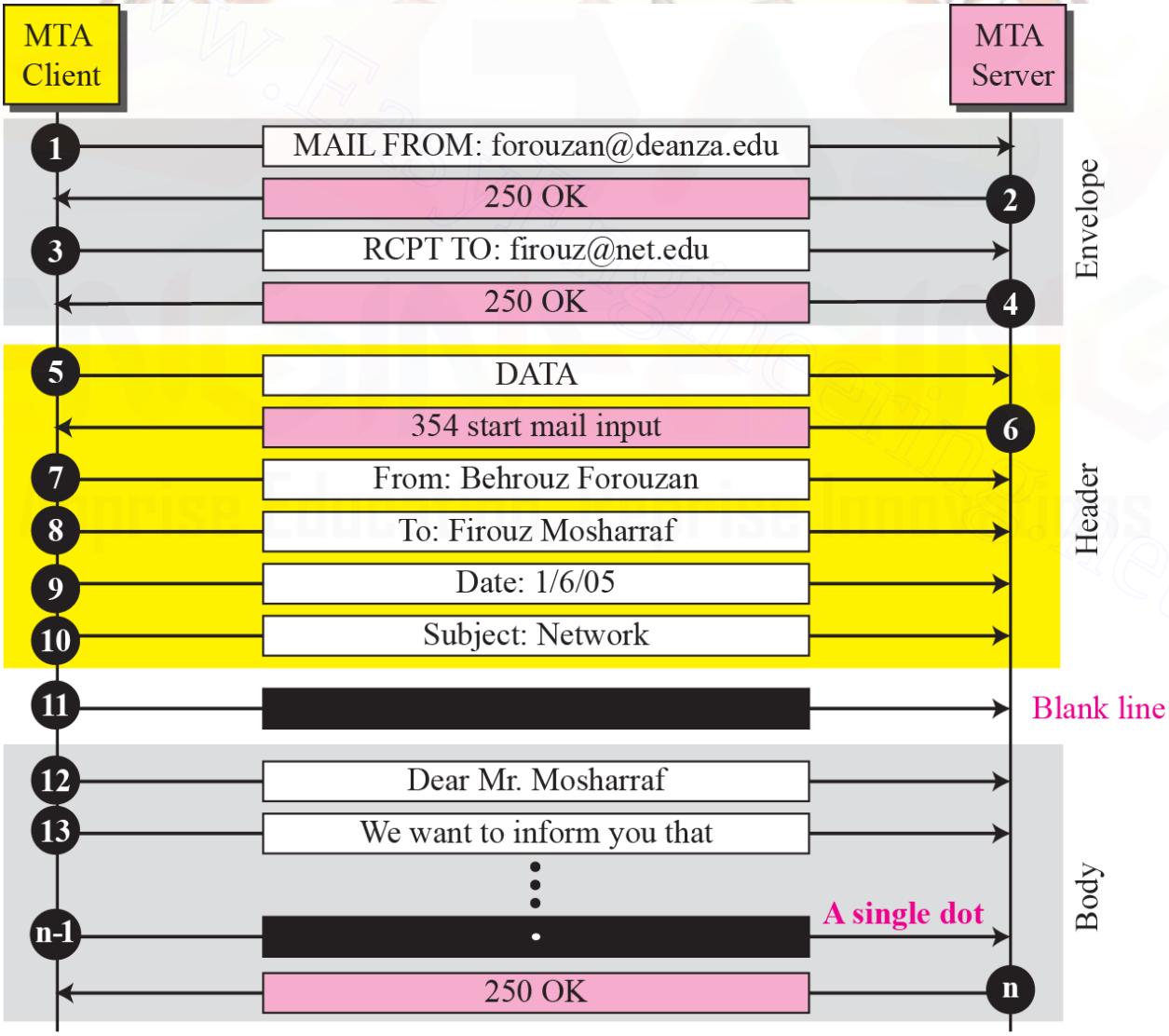
Mail Transfer Phases

The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination.

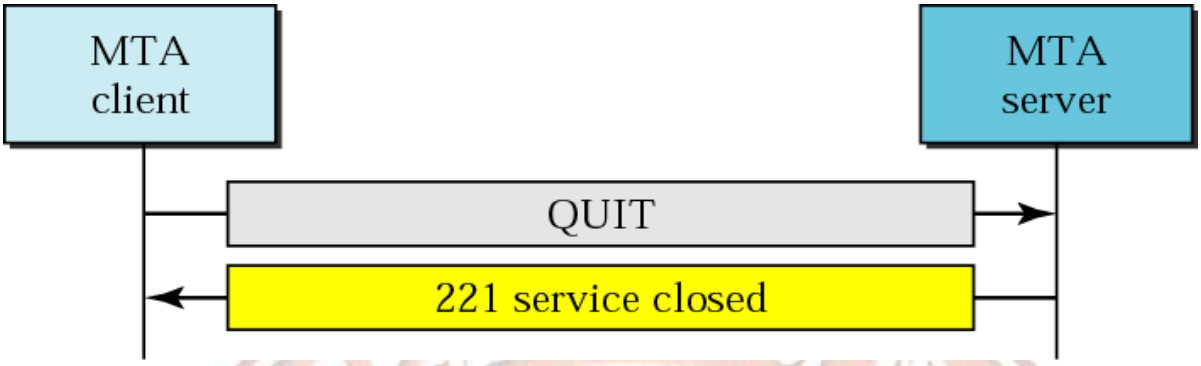
Connection Establishment



Message Transfer



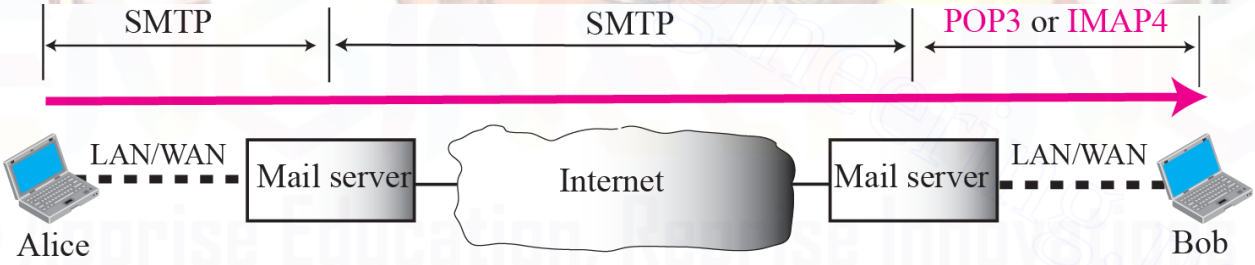
Connection Termination:



POP3 –POST OFFICE PROTOCOL

SMTP is a *push* protocol; it pushes the message from the client to the server. The direction of the bulk: data (messages) is from the client to the server. The third stage needs a *pull* protocol; the client must pull messages from the server. The direction of the bulk data is from the server to the client. The third stage uses a message access agent.
Two message access protocols

- Post Office Protocol, version 3 (POP3)
- Internet Mail Access Protocol, version 4 (IMAP4).



Post Office Protocol, version 3 (POP3) is simple and limited in functionality. The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.

- Mail access starts with the client to download e-mail from the mailbox on the mail server.
- The client opens a connection to the server on TCP port 110.
- It then sends its user name and password to access the mailbox.
- The user can then list and retrieve the mail messages, one by one.

Post Office Protocol version 3 (POP3) is a standard mail protocol used to **receive emails** from a remote server to a local email client. POP3 allows you to download email messages on your local computer and read them even when you are offline.

POP3 protocol works on two ports:

- **Port 110** - this is the default POP3 non-encrypted port

- **Port 995** - this is the port you need to use if you want to connect using POP3 securely

POP3 has two modes:

- delete mode
- keep mode.

In the delete mode, the mail is deleted from the mailbox after each retrieval. The delete mode is normally used when the user is working at her permanent computer and can save and organize the received mail after reading or replying.

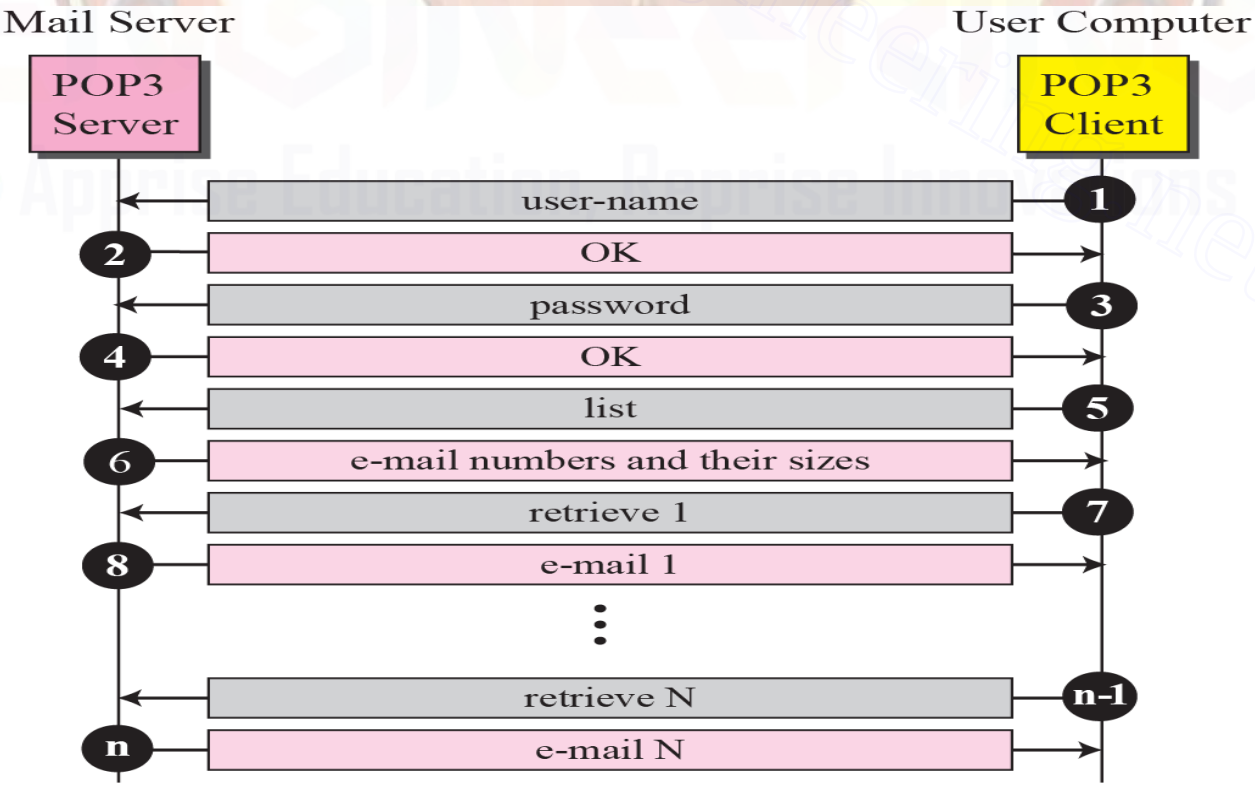
In the keep mode, the mail remains in the mailbox after retrieval. The keep mode is normally used when the user accesses her mail away from her primary computer (e.g., a laptop). The mail is read but kept in the system for later retrieval and organizing.

POP3 is designed to delete mail on the server as soon as the user has downloaded it. POP can be thought of as a "store-and-forward" service.

An alternative protocol is Internet Message Access Protocol (IMAP). IMAP provides the user more capabilities for retaining e-mail on the server and for organizing it in folders on the server. IMAP can be thought of as a remote file server.

POP and IMAP deal with the receiving of e-mail. But Simple Mail Transfer Protocol (SMTP), a protocol for transferring e-mail across the Internet. You send e-mail with SMTP and a mail handler receives it on your recipient's behalf. Then the mail is read using POP or IMAP.

POP3 is the current version of this particular style of email protocol. Since POP3 creates local copies of emails and deletes the originals from the server, the emails are tied to that specific machine. It cannot be accessed via any webmail or any separate client on other computers.



POP3 does not allow the user to organize her mail on the server; the user cannot have different folders on the server. POP3 does not allow the user to partially check the contents of the mail before downloading.

POP3 Commands	
Command	Description
USER identification	This command makes it possible to be authenticated. It must be followed by the user name, i.e. a character string identifying the user on the server. The USER command must precede the *PASS* command.
PASS password	The *PASS* command makes it possible to specify the user's password where the name has been specified by a prior *USER* command.
STAT	Information on the messages contained on the server
RETR	Number of the message to be picked up
DELE	Number of the message to be deleted
LIST [msg]	Number of the message to be displayed
NOOP	Allows the connection to be kept open in the event of inactivity
TOP <messageID> <n>	Command displaying *n* lines of the message, where the number is given in the argument. In the event of a positive response from the server, it will send back the message headers, then a blank line and finally the first *n* lines of the message.
UIDL [msg]	Request to the server to send back a line containing information about the message possibly given in the argument. This line contains a character string called a *unique identifier listing*, making it possible to uniquely identify the message on the server, independently of the session. The optional argument is a number relating to a message existing on the POP server, i.e. an undeleted message).
QUIT	The *QUIT* command requests exit from the POP3 server. It leads to the deletion of all messages marked as deleted and sends back the status of this action.

****

IMAP4 - Internet Mail Access Protocol, version 4

Another mail access protocol is Internet Mail Access Protocol, version 4 (IMAP4). IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex.

- It defines an abstraction known as a MAILBOX. Mailboxes are located on the same computer as a server.
- IMAP4 is a method for accessing electronic mail messages that are kept on a mail server. It permits a client e-mail program to view and manipulate those messages.
- Electronic mail stored on an IMAP server can be viewed or manipulated from a desktop computer at home, a notebook computer, or at a workstation.
- Mail messages can be accessed from multiple locations.

The POP3 protocol is very simple. It allows you to have a collection of messages stored in a text file on the server. Your e-mail client (e.g. Outlook Express) can connect to your POP3 e-mail server and download the messages from the POP3 text file onto your PC.

With POP3, once you download your e-mail it's stuck on the machine to which you downloaded it. If you want to read your e-mail both on your desktop machine and your laptop POP3 makes life difficult.

IMAP is a more advanced protocol that solves these problems. With IMAP, your mail stays on the e-mail server. You can organize your mail into folders, and all the folders live on the server as well. When you search your e-mail, the search occurs on the server machine, rather than on your machine.

This approach makes it extremely easy for you to access your e-mail from any machine, and regardless of which machine you use; you have access to all of your mail in all of your folders.

Internet Message Access Protocol, version 4 is a more complex protocol, which provides more extensive functionality than is available through POP3. With IMAP4, clients can not only retrieve messages from a server, but they can also manipulate the remote message folders (or mailboxes) in which the messages are stored.

Whereas POP3 simply allows messages to be downloaded to a local mailbox, IMAP4 includes operations for creating, deleting and removing remote mailboxes as well as for checking for new messages and selective fetching of message attributes. Of particular appeal to many users is the fact that it also provides facilities for offline support and synchronization.

IMAP4 provides the following extra functions:

- A user can check the e-mail header prior to downloading.
- A user can search the contents of the e-mail for a specific string of characters prior to downloading.
- A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
- A user can create, delete, or rename mailboxes on the mail server.
- A user can create a hierarchy of mailboxes in a folder for e-mail storage.
- IMAP allows several simultaneous accesses to be managed
- IMAP makes it possible to manage several inboxes
- IMAP provides more criteria which can be used to sort emails

To retrieve a message from an IMAP4 server, an IMAP4 client first establishes a Transmission Control Protocol (TCP) session using TCP port 143. The client then identifies itself to the server and issues a series of IMAP4 commands:

•LIST:

Retrieves a list of folders in the client’s mailbox

•SELECT:

Selects a particular folder to access its messages

•FETCH:

Retrieves individual messages

•LOGOUT:

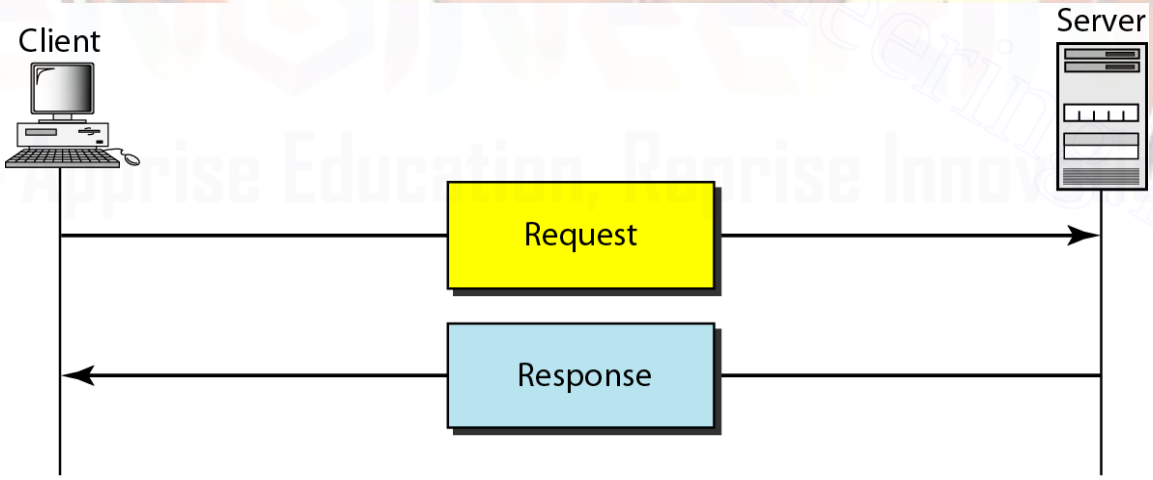
****

HTTP - Hypertext Transfer Protocol

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web. HTTP functions as a combination of FTP and SMTP. It is similar to FTP because it transfers files and uses the services of TCP. It is much simpler than FTP because it uses only one TCP connection. There is no separate control connection; only data are transferred between the client and the server.

HTTP is like SMTP because the data transferred between the client and the server look like SMTP messages. In addition, the format of the messages is controlled by MIME-like headers. Unlike SMTP, the HTTP messages are not destined to be read by humans; they are read and interpreted by the HTTP server and HTTP client (browser).

SMTP messages are stored and forwarded, but HTTP messages are delivered immediately. The commands from the client to the server are embedded in a request message. The contents of the requested file or other information are embedded in a response message. HTTP uses the services of TCP on well-known port 80.

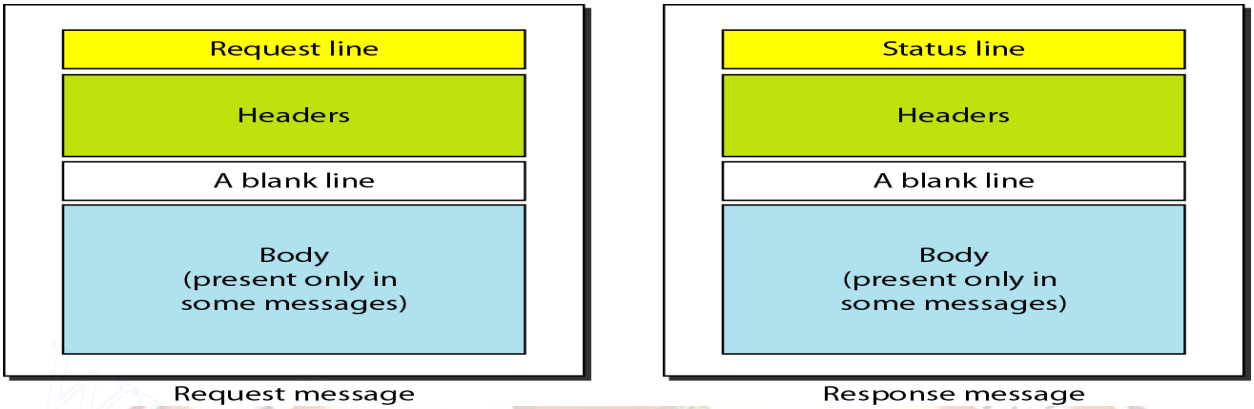


HTTP Transaction

HTTP uses the services of TCP; HTTP itself is a stateless protocol. The client initializes the transaction by sending a request message. The server replies by sending a response messages.

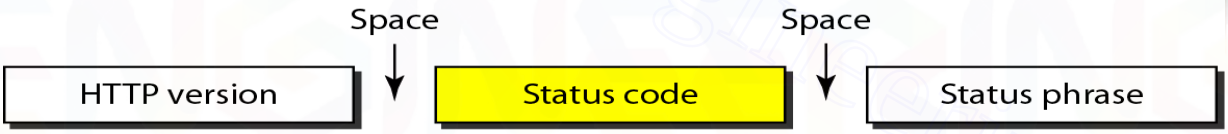
The format of the request and response messages are similar. A request message consists of a request line, a header, and sometimes a body.

A response message consists of a status line, a header, and sometimes a body.



Request and Status Lines The first line in a request message is called a request line; the first line in the response message is called the status line. There is one common field

a. Request line



b. Status line

Request type. This field is used in the request message. In version 1.1 of HTTP, several request types are defined. The request type is categorized into *methods*

Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
OPTION	Inquires about available options

- **Version.** The most current version of HTTP is 1.1.
- **Status code.** This field is used in the response message. The status code field is similar to those in the FTP and the SMTP protocols. It consists of three digits.

  Whereas the codes in the 100 range are only informational, the codes in the 200range indicate a successful request. The codes in the 300 range redirect the client to another URL, and the codes in the 400 range indicate an error at the client site. Finally, the codes in the 500 range indicate an error at the server site.

- Status phrase. This field is used in the response message. It explains the status code in text form.

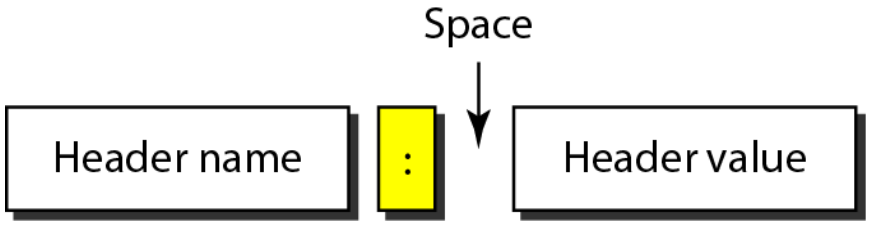
Status codes

Code	*Phrase*	*Description*
	Informational	
100	Continue	The initial part of the request has been received, and the client may continue with its request.
101	Switching	The server is complying with a client request to switch protocols defined in the upgrade header.
	Success	
200	OK	The request is successful.
201	Created	A new URL is created.
202	Accepted	The request is accepted, but it is not immediately acted upon.
204	No content	There is no content in the body.

Code	*Phrase*	*Description*
	Redirection	
301	Moved permanently	The requested URL is no longer used by the server.
302	Moved temporarily	The requested URL has moved temporarily.
304	Not modified	The document has not been modified.
	Client Error	
400	Bad request	There is a syntax error in the request.
401	Unauthorized	The request lacks proper authorization.
403	Forbidden	Service is denied.
404	Not found	The document is not found.
405	Method not allowed	The method is not supported in this URL.
406	Not acceptable	The format requested is not acceptable.
	Server Error	
500	Internal server error	There is an error, such as a crash, at the server site.
501	Not implemented	The action requested cannot be performed.
503	Service unavailable	The service is temporarily unavailable, but may be requested in the future.

Header: The header exchanges additional information between the client and the server. For example, the client can request that the document be sent in a special format, or the server can send extra information about the document.

- The header can consist of one or more header lines.
- Each header line has a header name, a colon, a space, and a header value.
- A header line belongs to one of four categories: general header, request header, response header, and entity header.
- A request message can contain only general, request, and entity headers.
- A response message, on the other hand, can contain only general, response, and entity headers.



General header The general header gives general information about the message and can be present in both a request and a response.

GENERAL HEADERS

Header	*Description*
Cache-control	Specifies information about caching
Connection	Shows whether the connection should be closed or not
Date	Shows the current date
MIME-version	Shows the MIME version used
Upgrade	Specifies the preferred communication protocol

Request header. The request header can be present only in a request message. It specifies the client's configuration and the client's preferred document format.

Request headers

Header	*Description*
Accept	Shows the medium format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
From	Shows the e-mail address of the user
Host	Shows the host and port number of the server
If-modified-since	Sends the document if newer than specified date
If-match	Sends the document only if it matches given tag
If-non-match	Sends the document only if it does not match given tag
If-range	Sends only the portion of the document that is missing
If-unmodified-since	Sends the document if not changed since specified date
Referrer	Specifies the URL of the linked document
User-agent	Identifies the client program

Response header The response header can be present only in a response message.It specifies the server's configuration and special information about the request.

Header	*Description*
Accept-range	Shows if server accepts the range requested by client
Age	Shows the age of the document
Public	Shows the supported list of methods
Retry-after	Specifies the date after which the server is available
Server	Shows the server name and version number

Entity header: The entity header gives information about the body of the document. It is mostly present in response messages, some request messages, such as POST or PUT methods, that contain a body also use this type of header.

Header	*Description*
Allow	Lists valid methods that can be used with a URL
Content-encoding	Specifies the encoding scheme
Content-language	Specifies the language
Content-length	Shows the length of the document
Content-range	Specifies the range of the document
Content-type	Specifies the medium type
Etag	Gives an entity tag
Expires	Gives the date and time when contents may change
Last-modified	Gives the date and time of the last change
Location	Specifies the location of the created or moved document

Body The body can be present in a request or response message. Usually, it contains the document to be sent or received.

****

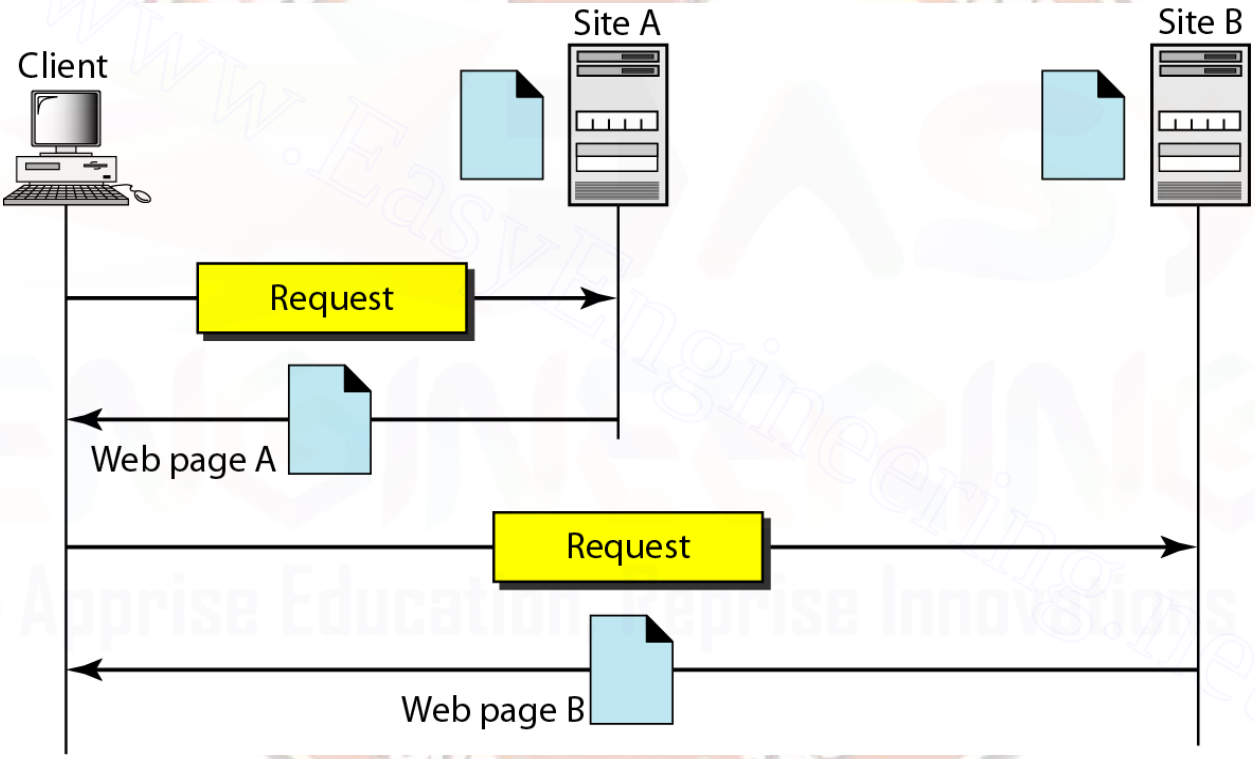
WEB SERVICES

WWW - World Wide Web:

The **World Wide Web** (WWW) is a repository of information linked together from Points all over the world. The WWW has a unique combination of flexibility, portability, and user-friendly features that distinguish it from other services provided by the Internet.

ARCHITECTURE

The WWW today is a distributed client J server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called *sites.*

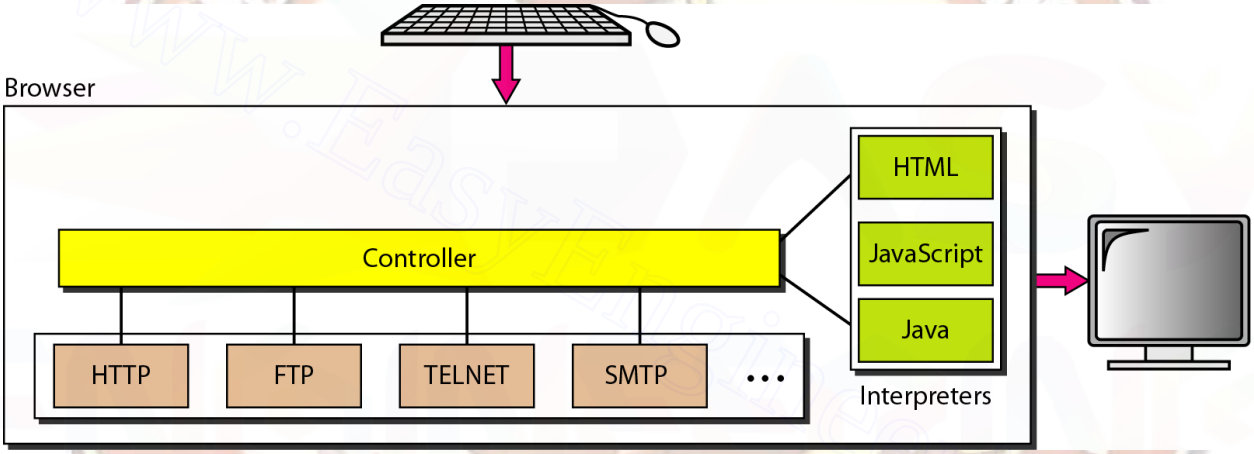


Each site holds one or more documents, referred to as *Web pages.* Each Web page can contain a link to other pages in the same site or at other sites. The pages can be retrieved and viewed by using browsers. The client needs to see some information that it knows belongs to site A. It sends a request through its browser, a program that is designed to fetch Web documents. The request, among other information, includes the address of the site and the Web page, called the URL, which we will discuss shortly. The server at site A finds the document and sends it to the client. When the user views the document, she finds some references to other documents, including a

Web page at site B. The reference has the URL for the new site. The user is also interested in seeing this document. The client sends another request to the new site, and the new page is retrieved.

Client (Browser)

A variety of vendors offer commercial browsers that interpret and display a Web document, and all use nearly the same architecture. Each browser usually consists of three parts: a controller, client protocol, and interpreters. The controller receives input from the keyboard or the mouse and uses the client programs to access the document. After the document has been accessed, the controller uses one of the interpreters to display the document on the screen. The client protocol can be one of the protocols described previously such as FTP or HTIP (described later in the chapter). The interpreter can be HTML, Java, or JavaScript, depending on the type of document.

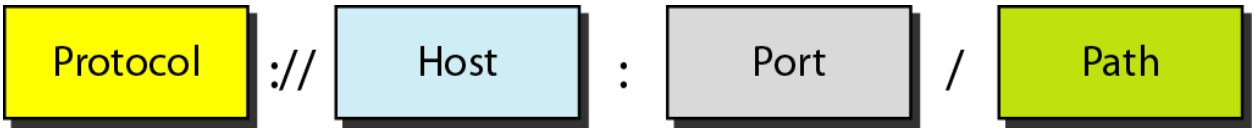


Server

The Web page is stored at the server. Each time a client request arrives, the corresponding document is sent to the client. To improve efficiency, servers normally store requested files in a cache in memory; memory is faster to access than disk. A server can also become more efficient through multithreading or multiprocessing. In this case, a server can answer more than one request at a time.

Uniform Resource Locator

A client that wants to access a Web page needs the address. To facilitate the access of documents distributed throughout the world, HTTP uses locators. The uniform resource locator (URL) is a standard for specifying any kind of information on the Internet. The URL defines four things: protocol, host computer, port, and path



The *protocol* is the client/server program used to retrieve the document. Many different protocols can retrieve a document; among them are FTP or HTTP. The most common today is HTTP.

The host is the computer on which the information is located, although the name of the computer can be an alias. Web pages are usually stored in computers, and computers are given alias names that usually begin with the characters "www". This is not mandatory, however, as the host can be any name given to the computer that hosts the Web page.

The URL can optionally contain the port number of the server. If the *port* is included, it is inserted between the host and the path, and it is separated from the host by a colon.

Path is the pathname of the file where the information is located. Note that the path can itself contain slashes that, in the UNIX operating system, separate the directories from the subdirectories and files.

Cookies

The World Wide Web was originally designed as a stateless entity. A client sends a request; a server responds. Their relationship is over. The original design of WWW, retrieving publicly available documents, exactly fits this purpose. Today the Web has other functions; some are listed here.

I. Some websites need to allow access to registered clients only.

2. Websites are being used as electronic stores that allow users to browse through the store, select wanted items, put them in an electronic cart, and pay at the end with a credit card.

3. Some websites are used as portals: the user selects the Web pages he wants to see.

4. Some websites are just advertising.

For these purposes, the cookie mechanism was devised. We discussed the use of cookies at the transport layer in Chapter 23; we now discuss their use in Web pages.

Creation and Storage of Cookies

The creation and storage of cookies depend on the implementation; however, the principle is the same.

When a server receives a request from a client, it stores information about the client in a file or a string. The information may include the domain name of the client, the contents of the cookie

(information the server has gathered about the client such as name, registration number, and so on), a timestamp, and other information depending on the implementation.

2. The server includes the cookie in the response that it sends to the client.

3. When the client receives the response, the browser stores the cookie in the cookie directory, which is sorted by the domain server name.

Using Cookies

When a client sends a request to a server, the browser looks in the cookie directory to see if it can find a cookie sent by that server. If found, the cookie is included in the request. When the server receives the request, it knows that this is an old client, not a new one. Note that the contents of the cookie are never read by the browser or disclosed to the user. It is a cookie *made* by the server and *eaten* by the server. Now let us see how a cookie is used for the four previously mentioned purposes:

1. The site that restricts access to registered clients only sends a cookie to the client when the client registers for the first time. For any repeated access, only those clients that send the appropriate cookie are allowed.

2. An electronic store (e-commerce) can use a cookie for its client shoppers. When a client selects an item and inserts it into a cart, a cookie that contains information about the item, such as its number and unit price, is sent to the browser. If the client selects a second item, the cookie is updated with the new selection information. And so on. When the client finishes shopping and wants to check out, the last cookie is retrieved and the total charge is calculated.

3. A Web portal uses the cookie in a similar way. When a user selects her favorite pages, a cookie is made and sent. If the site is accessed again, the cookie is sent to the server to show what the client is looking for.

4. A cookie is also used by advertising agencies. An advertising agency can place banner ads on some main website that is often visited by users. The advertising agency supplies only a URL that gives the banner address instead of the banner itself.

When a user visits the main website and clicks on the icon of an advertised corporation, a request is sent to the advertising agency. The advertising agency sends the banner, a GIF file, for example, but it also includes a cookie with the ill of the user.

Any future use of the banners adds to the database that profiles the Web behavior of the user. The advertising agency has compiled the interests of the user and can sell this information to

other parties. This use of cookies has made them very controversial. Hopefully, some new regulations will be devised to preserve the privacy of users.

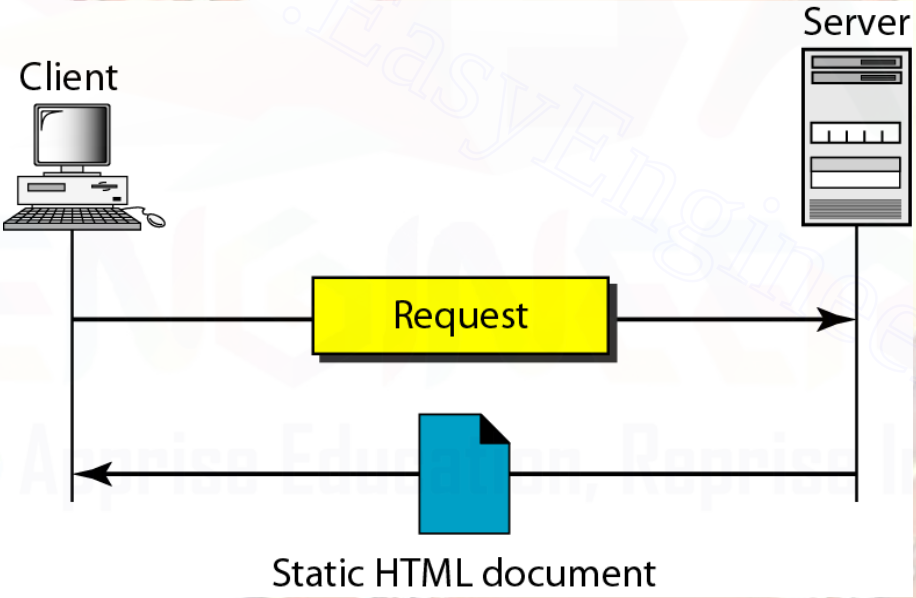
****

WEB DOCUMENTS

The documents in the WWW can be grouped into three broad categories: static, dynamic, and active. The category is based on the time at which the contents of the document are determined.

Static Documents Static documents are fixed-content documents that are created and stored in a server.

The client can get only a copy of the document. In other words, the contents of the file are determined when the file is created, not when it is used. Of course, the contents in the server can be changed, but the user cannot change them. When a client accesses the document, a copy of the document is sent. The user can then use a browsing program to display the document.

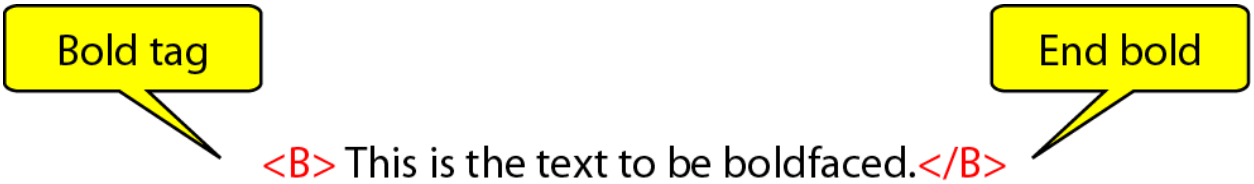


HTML

Hypertext Markup Language (HTML) is a language for creating Web pages. The term *markup language* comes from the book publishing industry. Before a book is typeset and printed, a copy editor reads the manuscript and puts marks on it. These marks tell the compositor how to format the text. For example, if the copy editor wants part of a line to be printed in boldface, he or she draws a wavy line under that part.

In the same way, data for a Web page are formatted for interpretation by a browser.

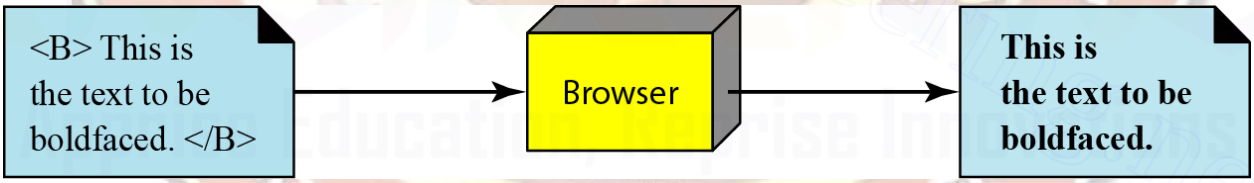
Let us clarify the idea with an example. To make part of a text displayed in boldface with HTML, we put beginning and ending boldface tags (marks) in the text.



The two tags <B> and </B> are instructions for the browser. When the browser sees these two marks, it knows that the text must be boldfaced.

A markup language such as HTML allows us to embed formatting instructions in the file itself. The instructions are included with the text. In this way, any browser can read the instructions and format the text according to the specific workstation. One might ask why we do not use the fonnatting capabilities of word processors to create and save formatted text. The answer is that different word processors use different techniques or procedures for formatting text. For example, imagine that a user creates formatted text on a Macintosh computer and stores it in a Web page. Another user who is on an IBM computer would not be able to receive the Web page because the two computers use different formatting procedures.

HTML use only ASCII characters for both the main text and formatting instructions. In this way, every computer can receive the whole document as an ASCII document. The main text is the data, and the formatting instructions can be used by the browser to format the data.



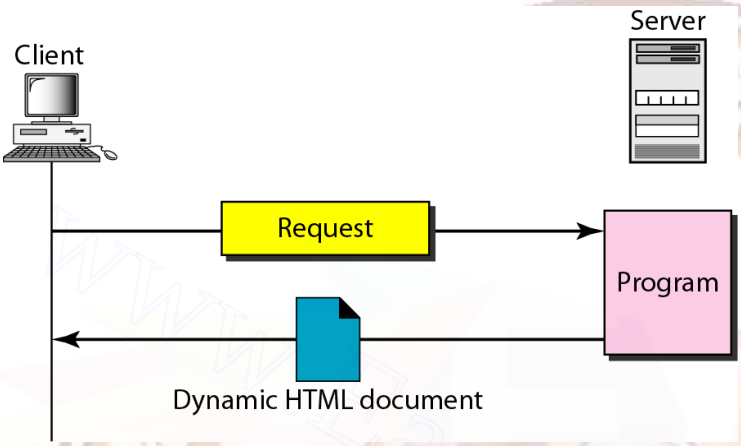
Dynamic Documents

A **dynamic document** is created by a Web server whenever a browser requests the document. When a request arrives, the Web server runs an application program or a script that creates the dynamic document. The server returns the output of the program or script as a response to the browser that requested the document. Because a fresh document is created for each request, the contents of a dynamic document can vary from one request to another. A very simple example of a dynamic document is the retrieval of the time and date from a server. Time and date are kinds of information that are dynamic in that they change from

moment to moment. The client can ask the server to run a program such as the *date* program in UNIX and send the result of the program to the client.

Common Gateway Interface (CGI)

The **Common** Gateway **Interface** (CGI) is a technology that creates and handles dynamic documents. CGI is a set of standards that defines how a dynamic document is written, how data are input to the program, and how the output result is used.



Active Documents

For many applications, we need a program or a script to be run at the client site. These are called active documents. For example, suppose we want to run a program that creates animated graphics on the screen or a program that interacts with the user.

The program definitely needs to be run at the client site where the animation or interaction takes place. When a browser requests an active document, the server sends a copy of the document or a script. The document is then run at the client (browser) site.

Input In traditional programming, when a program is executed, parameters can be passed to the program. Parameter passing allows the programmer to write a generic program that can be used in different situations. For example, a generic copy program can be written to copy any file to another. A user can use the program to copy a file named *x* to another file named *y* by passing *x* and *y* as parameters.

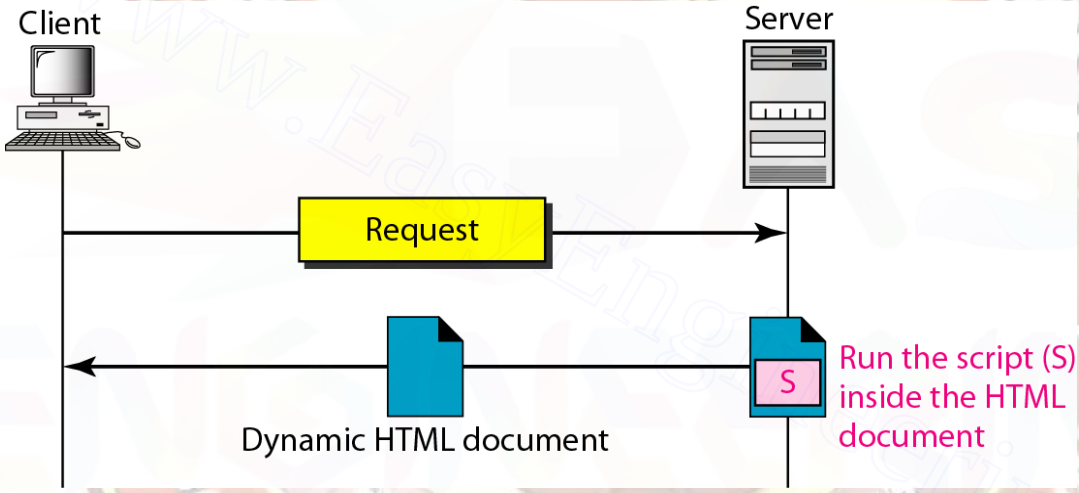
The input from a browser to a server is sent by using *a form.* If the infonnation in a fonn is small (such as a word), it can be appended to the URL after a question mark. For example, the following URL is carrying fonn infonnation (23, a value):

http://www.deanzalcgi-binlprog.pl?23

Output The whole idea of CGI is to execute a CGI program at the server site and send the output to the client (browser). The output is usually plain text or a text with HTML structures; however, the output can be a variety of other things. It can be graphics or binary data, a status code, instructions to the browser to cache the result, or instructions to the server to send an existing document instead of the actual output.

To let the client know about the type of document sent, a CGI program creates headers. As a matter of fact, the output of the CGI program always consists of two parts: a header and a body. The header is separated by a blank line from the body. This means any CGI program creates first the header, then a blank line, and then the body.

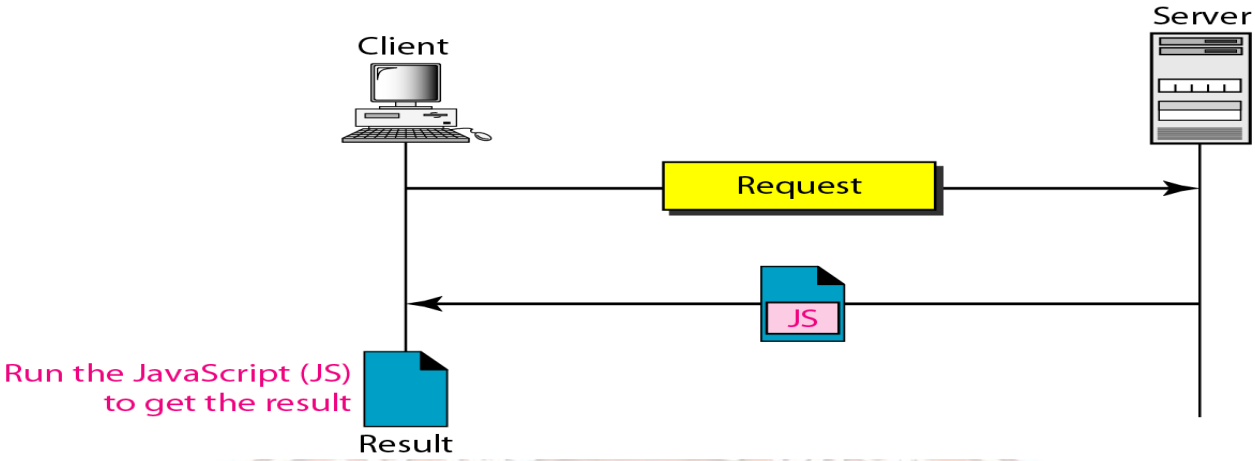
Although the header and the blank line are not shown on the browser screen, the header is used by the browser to interpret the body.



JavaScript

The idea of scripts in dynamic documents can also be used for active documents. If the active part of the document is small, it can be written in a scripting language; then it can be interpreted and run by the client at the same time. The script is in source code (text) and not in binary form. The scripting technology used in this case is usually JavaScript.

JavaScript, which bears a small resemblance to Java, is a very high level scripting language developed for this purpose.

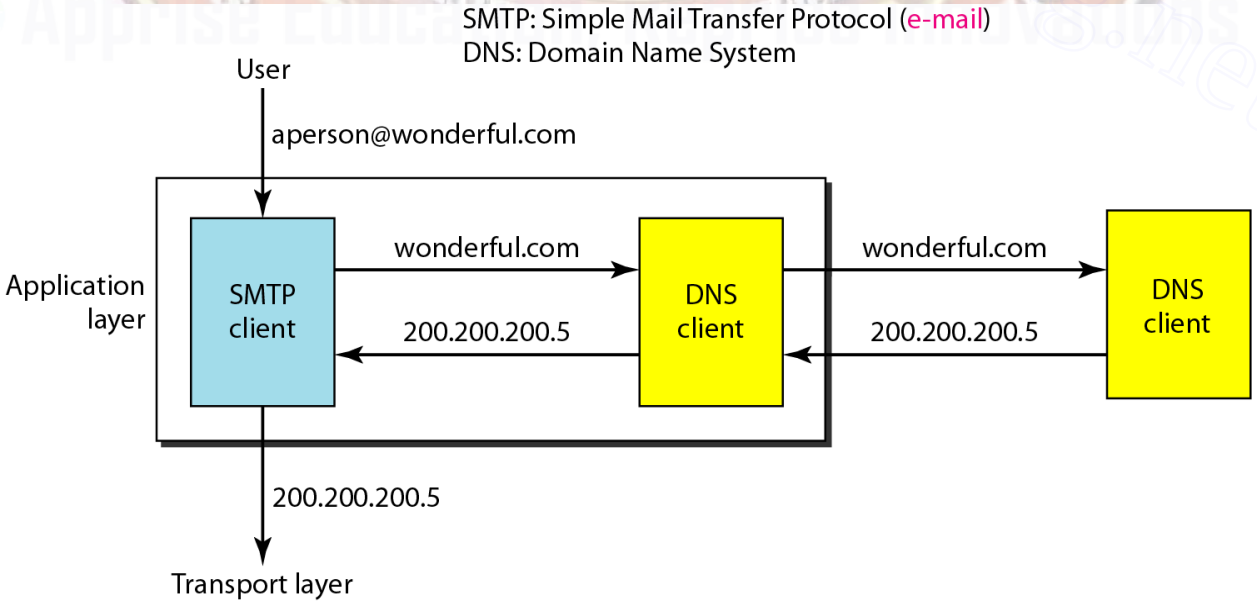


****

Domain** Name **System(DNS)

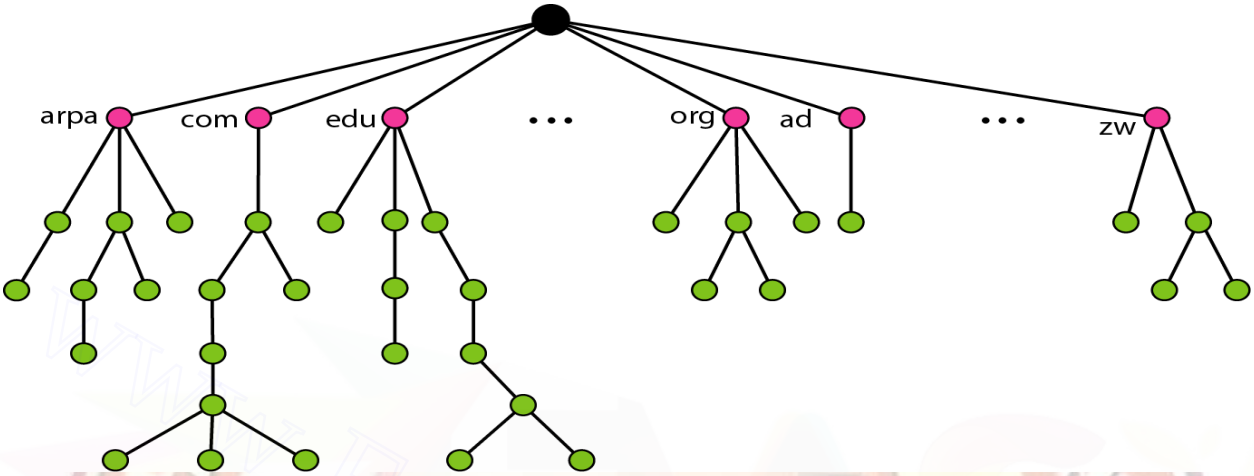
There are several applications in the application layer of the Internet model that follow the client/server paradigm. The client/server programs can be divided into two categories: those that can be directly used by the user, such as e-mail, and those that support other application programs. The Domain Name System (DNS) is a supporting program that is used by other programs such as e-mail.

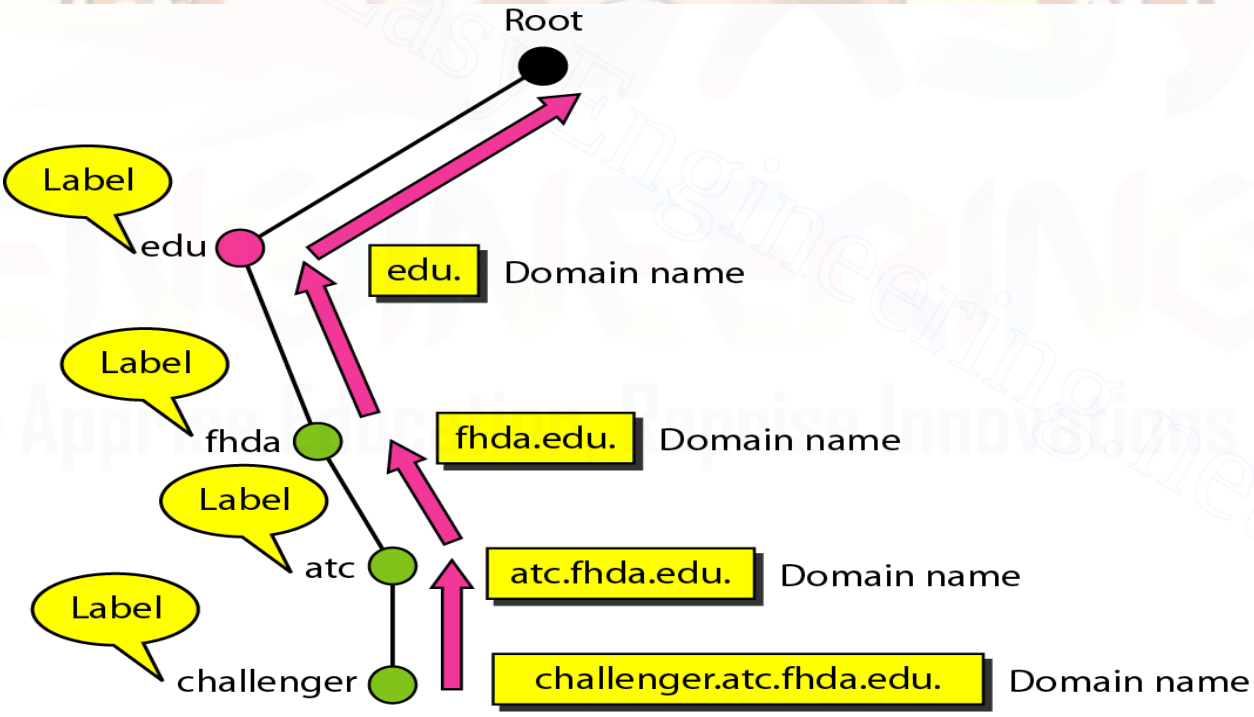
The Figure shows an example of how a DNS client/server program can support an e-mail program to find the IP address of an e-mail recipient. A user of an e-mail program may know the e-mail address of the recipient; however, the IP protocol needs the IP address. The DNS client program sends a request to a DNS server to map the e-mail address to the corresponding IP address.



DOMAIN NAME SPACE

To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127.





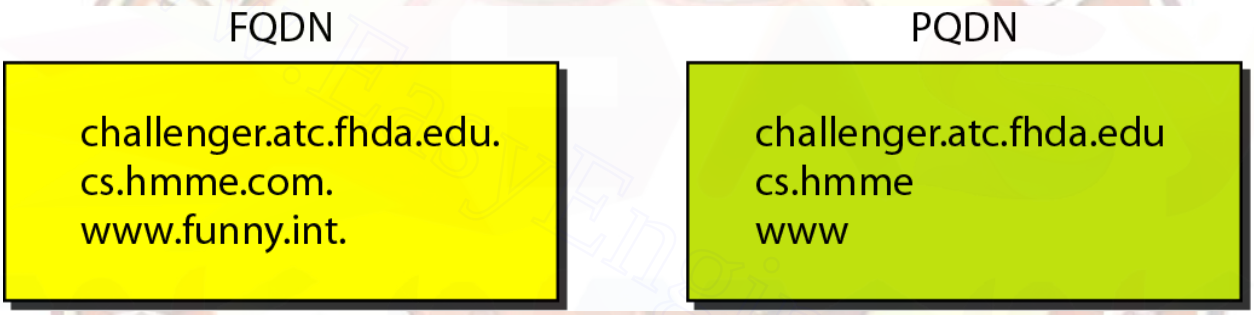
Fully Qualified Domain Name

If a label is terminated by a null string, it is called a fully qualified domain name (FQDN). An FQDN is a domain name that contains the full name of a host. It contains all labels, from the most specific to the most general, that uniquely define the name of the host.

For example, the domain name challenger.ate.tbda.edu. is the FQDN of a computer named *challenger* installed at the Advanced Technology Center (ATC) at De Anza College. A DNS server can only match an FQDN to an address. Note that the name must end with a null label, but because null means nothing, the label ends with a dot (.).

Partially Qualified Domain Name

If a label is not terminated by a null string, it is called a partially qualified domain name (PQDN). A PQDN starts from a node, but it does not reach the root. It is used when the name to be resolved belongs to the same site as the client. Here the resolver can supply the missing part, called the suffix, to create an FQDN. For example, if a user at the *jhda.edu.* site wants to get the IP address of the challenger computer, he or she can define the partial name challenger.The DNS client adds the suffix *atc.jhda.edu.* before passing the address to the DNS server.

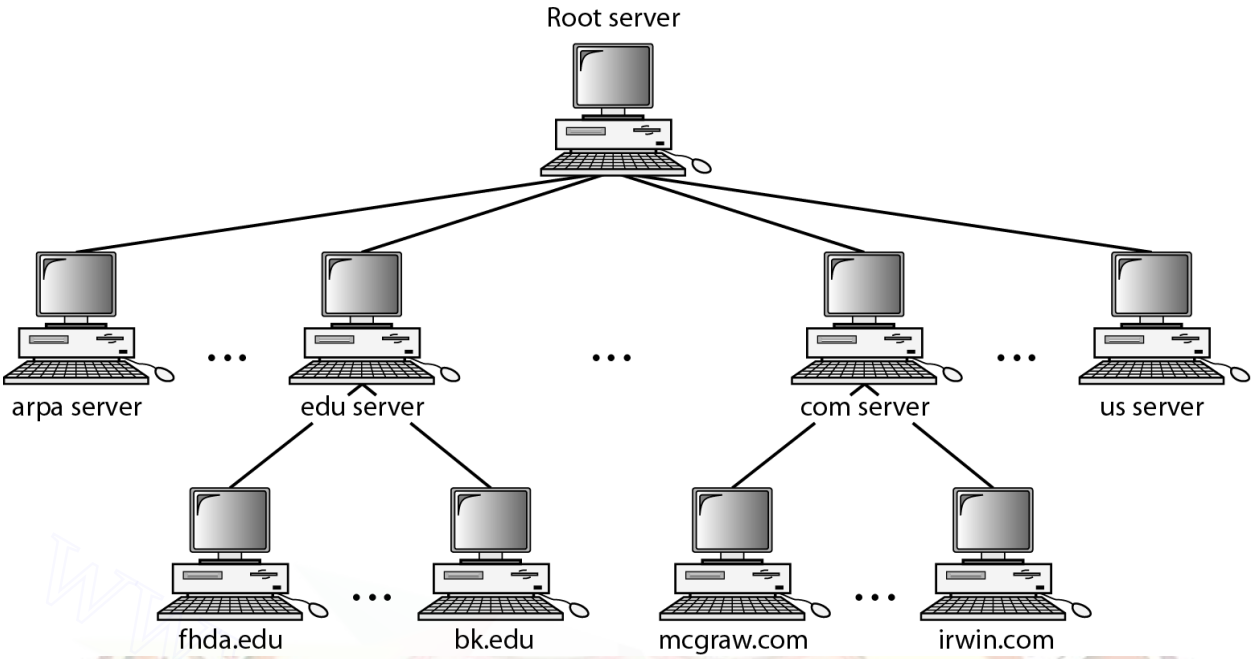


DISTRIBUTION OF NAME SPACE

The information contained in the domain name space must be stored. However, it is very inefficient and also unreliable to have just one computer store such a huge amount of information. It is inefficient because responding to requests from all over the world places a heavy load on the system. It is not unreliable because any failure makes the data inaccessible.

Hierarchy of Name Servers

The solution to these problems is to distribute the information among many computers called DNS servers. One way to do this is to divide the whole space into many domains based on the first level. In other words, we let the root stand alone and create as many domains (subtrees) as there are first-level nodes. Because a domain created in this way could be very large, DNS allows domains to be divided further into smaller domains (subdomains). Each server can be responsible (authoritative) for either a large or a small domain. In other words, we have a hierarchy of servers in the same way that we have a hierarchy of names.



Country Domains

The country domains section uses two-character country abbreviations (e.g., us for United States). Second labels can be organizational, or they can be more specific, national designations. The United States, for example, uses state abbreviations as a subdivision of us (e.g., ca.us.).

Figure shows the country domains section. The address *anza.cup.ca.us* can be translated to De Anza College in Cupertino, California, in the United States.

Inverse Domain

The inverse domain is used to map an address to a name. This may happen, for example, when a server has received a request from a client to do a task. Although the server has a file that contains a list of authorized clients, only the IP address of the client (extracted from the received IP packet) is listed. The server asks its resolver to send a query to the DNS server to map an address to a name to determine if the client is on the authorized list.

DNS MESSAGES

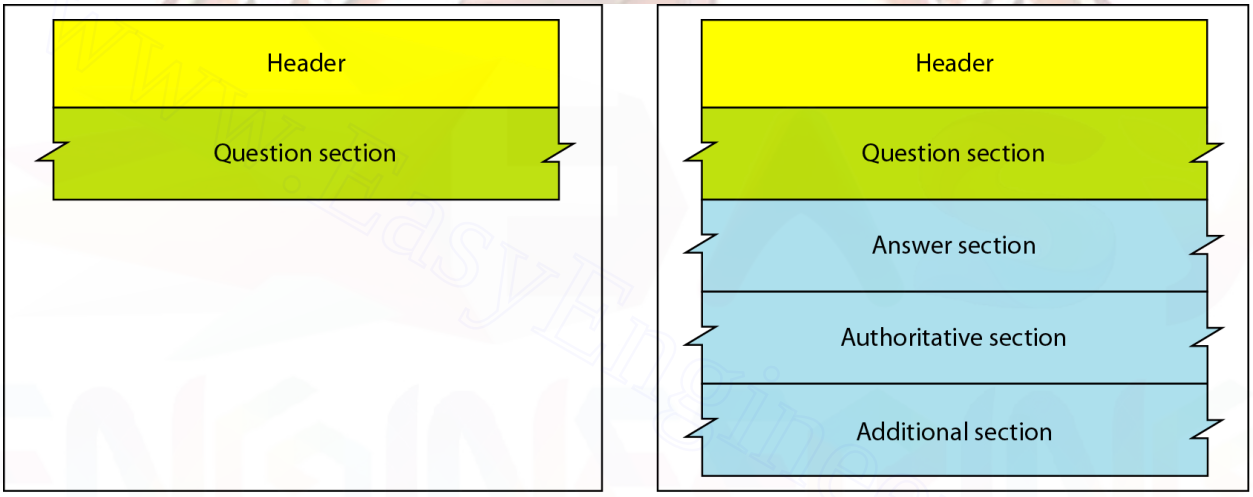
DNS has two types of messages: query and response. Both types have the same format.

The query message consists of a header and question records; the response message consists of a header, question records, answer records, authoritative records, and additional records.

Header

Both query and response messages have the same header format with some fields set to zero for the query messages. The header is 12 bytes. The *identification* subfield is used by the client to match the response with the query.

The client uses a different identification number each time it sends a query. The server duplicates this number in the corresponding response. The *flags* subfield is a collection of subfields that define the type of the message, the type of answer requested, the type of desired resolution (recursive or iterative), and so on. The *number of question records* subfield contains the number of queries in the question section of the message. The *number of answer records* subfield contains the number of answer records in the answer section of the response message.



a. Query b. Response

Question Section

This is a section consisting of one or more question records. It is present on both query and response messages. We will discuss the question records in a following section.

Answer Section

This is a section consisting of one or more resource records. It is present only on response messages. This section includes the answer from the server to the client (resolver).

Authoritative Section

This is a section consisting of one or more resource records. It is present only on response messages. This section gives infonnation (domain name) about one or more authoritative servers for the query.

Additional Information Section

This is a section consisting of one or more resource records. It is present only on response messages. This section provides additional infonnation that may help the resolver. For example, a server may give the domain name of an authoritative server to the resolver in the authoritative section, and include the IP address of the same authoritative server in the additional information section.

TYPES OF RECORDS

There are two types of records are used in DNS. The question records are used in the question section of the query and response messages. The resource records are used in the answer, authoritative, and additional information sections of the response message.

Question Record

A question record is used by the client to get information from a server. This contains the domain name.

Resource Record

Each domain name (each node on the tree) is associated with a record called the resource record. The server database consists of resource records. Resource records are also what is returned by the server to the client.

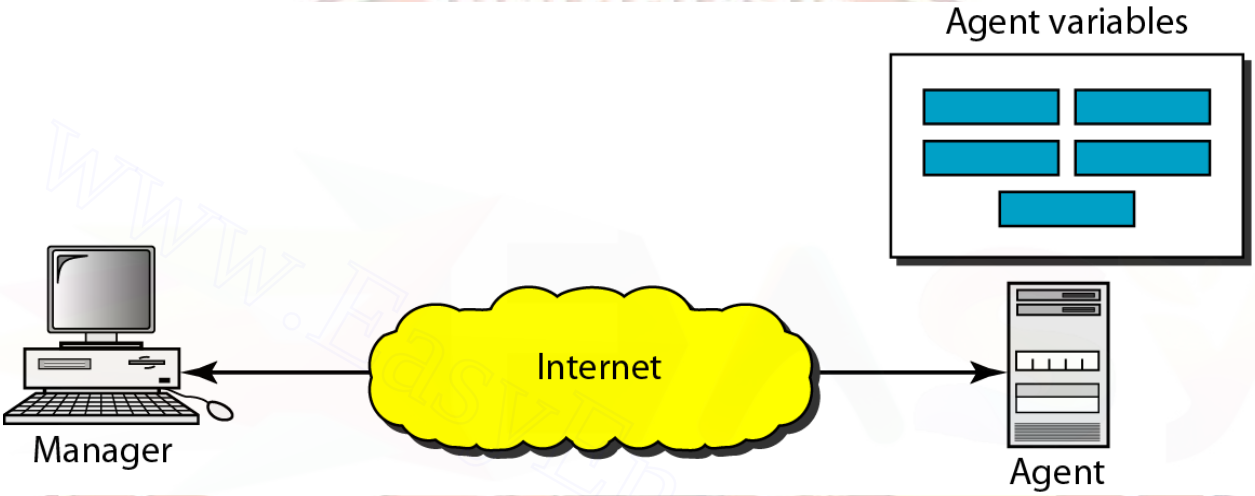
****

SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)

The Simple Network Management Protocol (SNMP) is a framework for managing devices in an internet using the TCP/IP protocol suite. It provides a set of fundamental operations for monitoring and maintaining an internet.

Concept

SNMP uses the concept of manager and agent. That is, a manager, usually a host, controls and monitors a set of agents, usually routers.



SNMP is an application-level protocol in which a few manager stations control a set of agents. The protocol is designed at the application level so that it can monitor devices made by different manufacturers and installed on different physical networks. In other words, SNMP frees management tasks from both the physical characteristics of the managed devices and the underlying networking technology. It can be used in a heterogeneous internet made of different LANs and WANs connected by routers made by different manufacturers.

Managers and Agents

A management station, called a manager, is a host that runs the SNMP client program. A managed station, called an agent, is a router (or a host) that runs the SNMP server program. Management is achieved through simple interaction between a manager and an agent. The agent keeps performance information in a database. The manager has access to the values in the database.

For example, a router can store in appropriate variables. the number of packets received and forwarded. The manager can fetch and compare the values of these two variables to see if the router is congested or not.

The manager can also make the router perform certain actions. For example, a router periodically checks the value of a reboot counter to see when it should reboot itself. It reboots itself, for example, if the value of the counter is O. The manager can use this feature to reboot the agent remotely at any time. It simply sends a packet to force a 0 value in the counter.

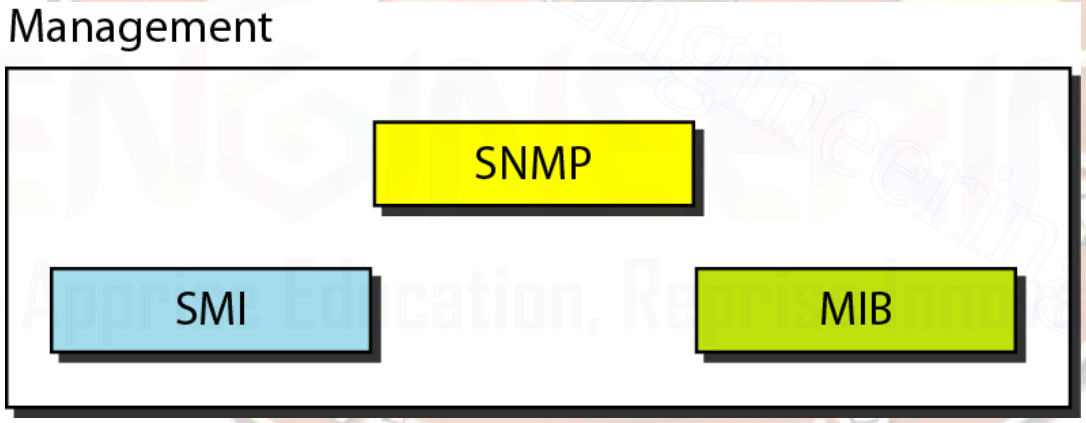
Agents can also contribute to the management process. The server program running on the agent can check the environment, and if it notices something unusual, it can send a warning message, called a trap, to the manager.

In other words, management with SNMP is based on three basic ideas:

1. A manager checks an agent by requesting information that reflects the behavior of the agent.
2. A manager forces an agent to perform a task by resetting values in the agent database.
3. An agent contributes to the management process by warning the manager of an unusual situation.

Management Components

To do management tasks, SNMP uses two other protocols: Structure of Management Information (SMI) and Management Information Base (MIB). In other words, management on the Internet is done through the cooperation of the three protocols SNMP, SMI, and MIB.



Role of SNMP

SNMP has some very specific roles in network management. It defines the format of the packet to be sent from a manager to an agent and vice versa. It also interprets the result and creates statistics (often with the help of other management software). The packets exchanged contain the object (variable) names and their status (values).

SNMP is responsible for reading and changing these values. SNMP defines the format of packets exchanged between a manager and an agent. It reads and changes the status (values) of objects (variables) in 8NMP packets.

Role of SMI

To use SNMP, we need rules. We need rules for naming objects. This is particularly important because the objects in SNMP form a hierarchical structure (an object may have a parent object and some children objects). Part of a name can be inherited from the parent. The sender may be a powerful computer in which an integer is stored as 8-byte data; the receiver may be a small computer that stores an integer as 4-byte data.

SMI is a protocol that defines these rules. However, we must understand that SMI only defines the rules; it does not define how many objects are managed in an entity or which object uses which type. SMI is a collection of general rules to name objects and to list their types. The association of an object with the type is not done by SMI.

8MI defines the general rules for naming objects, defining object types (including range and length), and showing how to encode objects and values.

8M1 does not define the number of objects an entity should manage or name the objects to be managed or define the association between the objects and their values.

Role of MIB

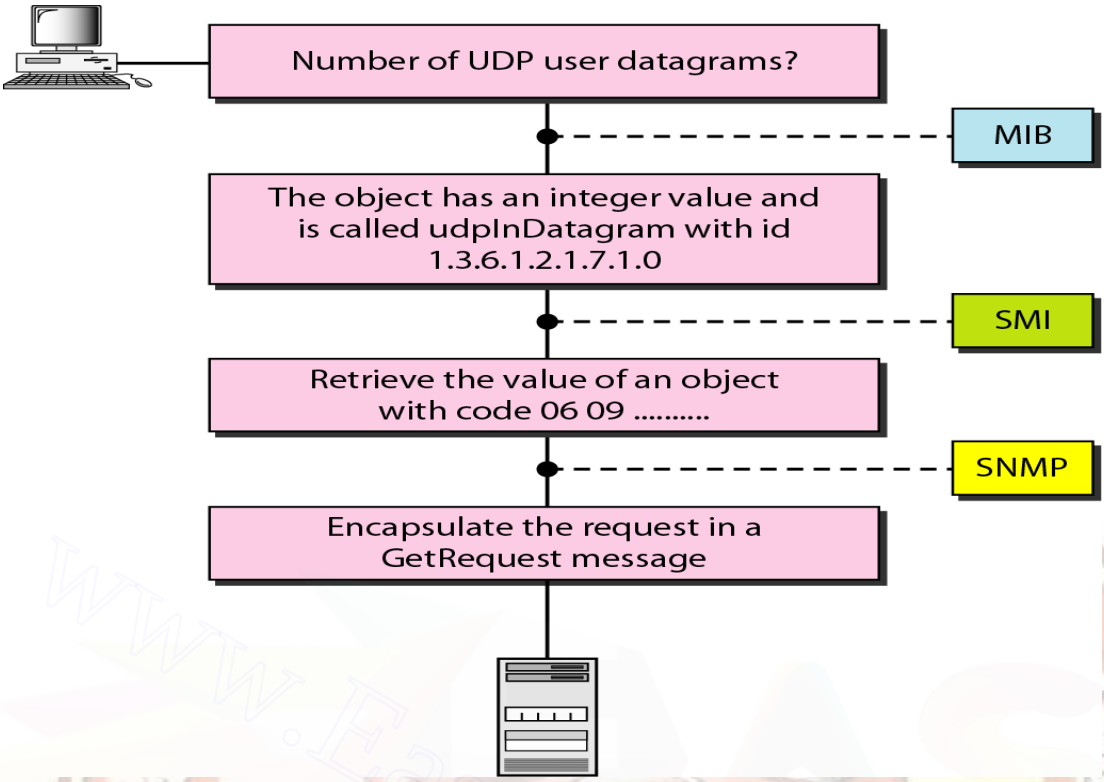
For each entity to be managed, this protocol must define the number of objects, name them according to the rules defined by SMI, and associate a type to each named object. This protocol is MIB. MIB creates a set of objects defined for each entity similar to a database (mostly metadata in a database, names and types without values).

MIB creates a collection of named objects, their types, and their relationships to each other in an entity to be managed.

An Analogy

Before discussing each of these protocols in greater detail, we give an analogy. The three network management components are similar to what we need when we write a program in a computer language to solve a problem.

MIB is responsible for finding the object that holds the number of the UDP user datagrams received. SMI, with the help of another embedded protocol, is responsible for encoding the name of the object. SNMP is responsible for creating a message, called a Get Request message, and encapsulating the encoded message. Of course, things are more complicated than this simple overview, but we first need more details of each protocol.

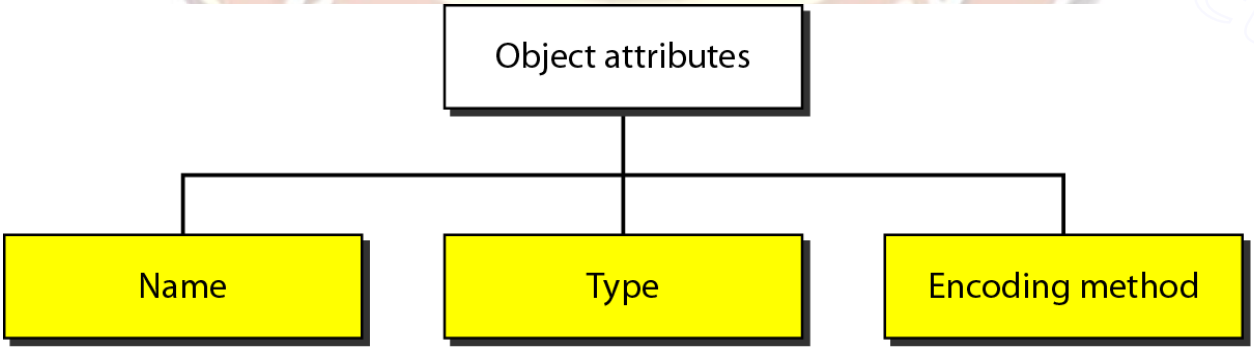


Structure of Management Information

The Structure of Management Information, version 2 (SMIv2) is a component for network management. Its functions are

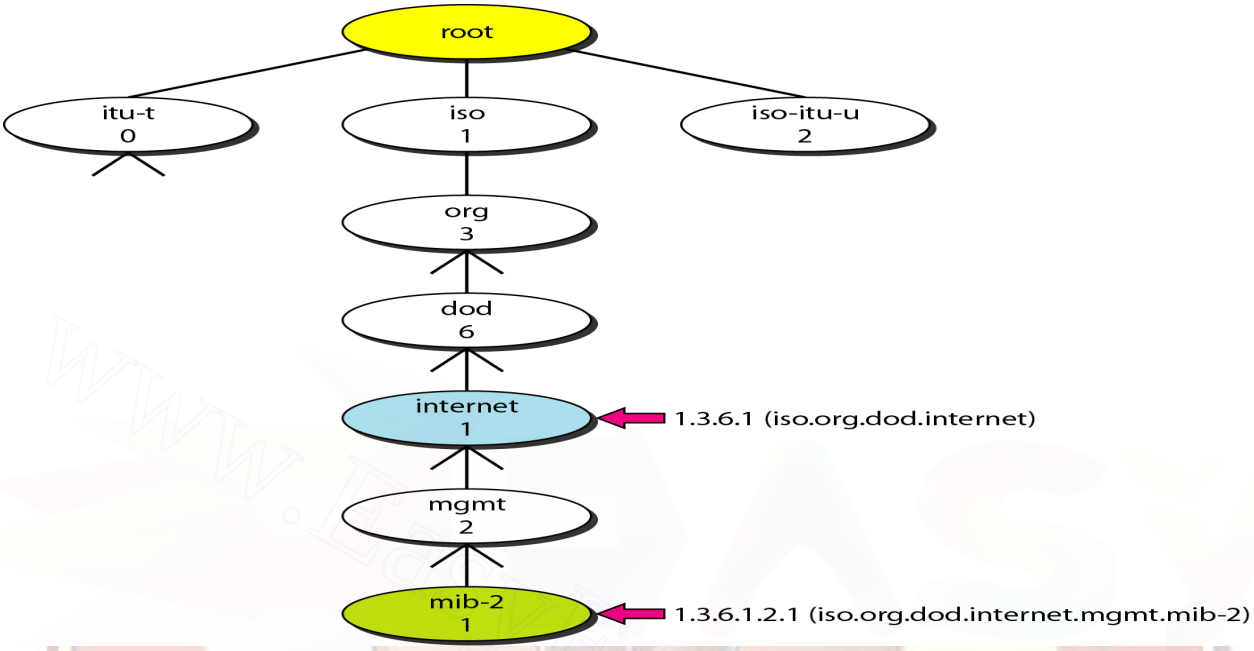
1. To name objects
2. To define the type of data that can be stored in an object
3. To show how to encode data for transmission over the network

SMI is a guideline for SNMP. It emphasizes three attributes to handle an object: name, data type, and encoding method.



Name

SMI requires that each managed object (such as a router, a variable in a router, a value) have a unique name. To name objects globally, SMI uses an object identifier, which is a hierarchical identifier based on a tree structure.



The tree structure starts with an unnamed root. Each object can be defined by using a sequence of integers separated by dots. The tree structure can also define an object by using a sequence of textual names separated by dots. The integer-dot representation is used in SNMP.

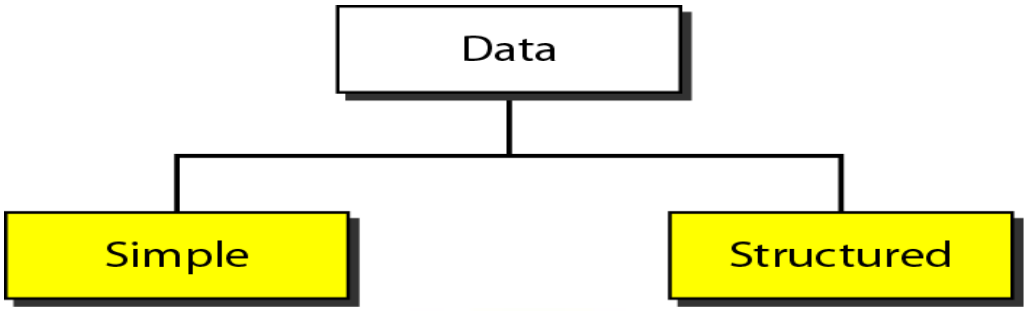
The name-dot notation is used by people. For example, the following shows the same object in two different notations:

iso.org.dod.internet.mgmt.mib-2 ... 1.3.6.1.2.1.

Type

The second attribute of an object is the type of data stored in it. To define the data type, SMI uses fundamental Abstract Syntax Notation 1 (ASN.l) definitions and adds some new definitions. In other words, SMI is both a subset and a superset of ASN.1.

SMI has two broad categories of data type: *simple* and *structured.*

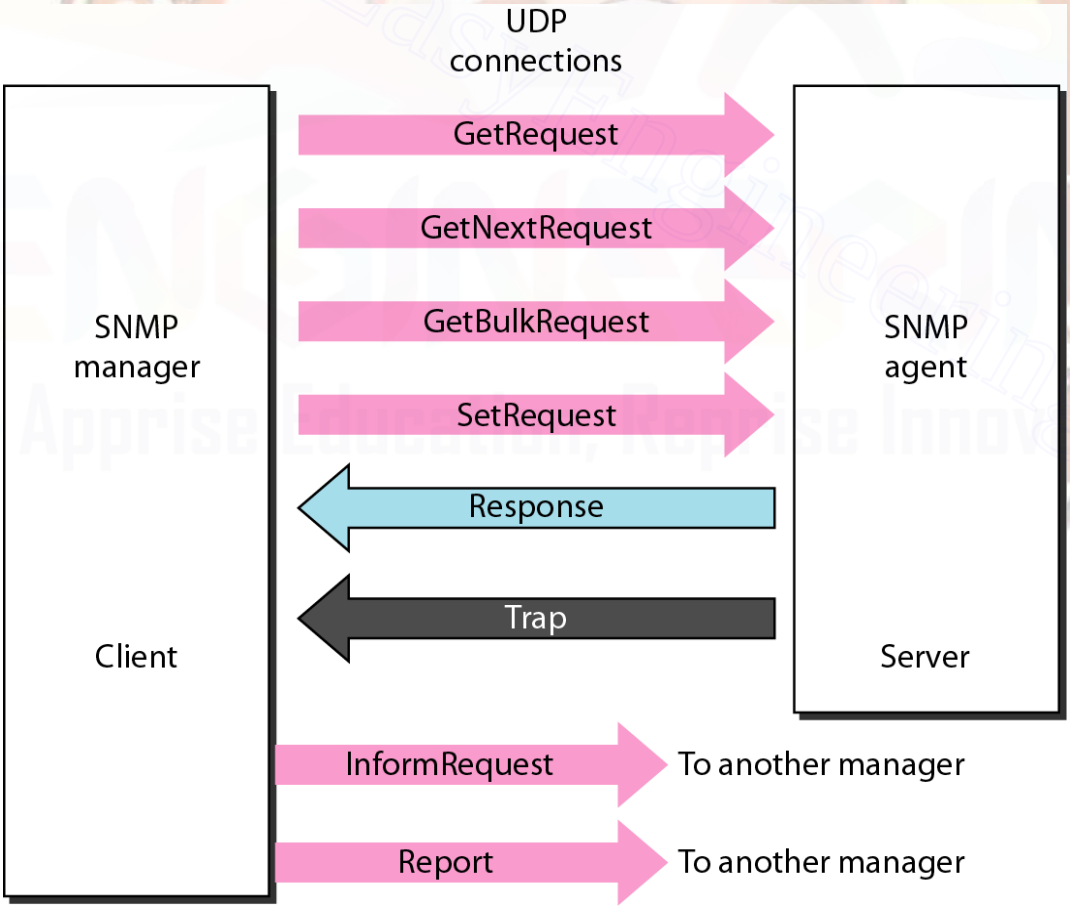


SNMP uses both SMI and MIB in Internet network management. It is an application program that allows

1. A manager to retrieve the value of an object defined in an agent
2. A manager to store a value in an object defined in an agent
3. An agent to send an alarm message about an abnormal situation to the manager

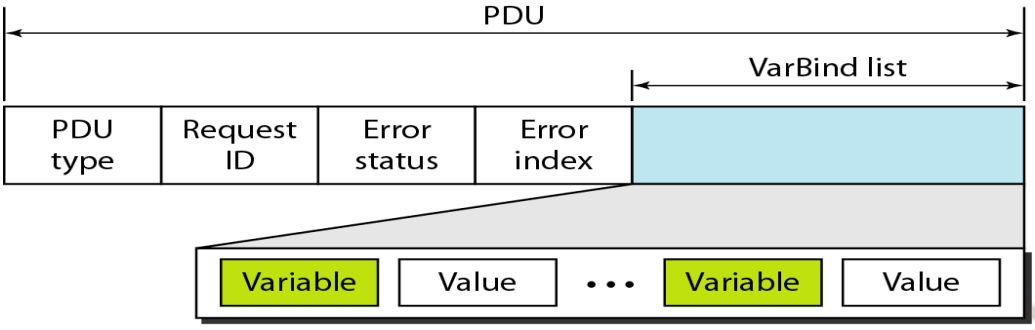
PDUs

SNMPv3 defines eight types of packets (or PDUs): Get Request, Get Next Request, Get Bulk Request, Set Request, Response, Trap, Inform Request, and Report.



Get Request:The Get Request PDU is sent from the manager (client) to the agent (server) to retrieve the value of a variable or a set of variables.

Get Next Request: The Get Next Request PDU is sent from the manager to the agent to retrieve the value of a variable. The retrieved value is the value of the object following the defined Object id in the PDD. It is mostly used to retrieve the values of the entries in a table.



Differences:

1. Error status and error index values are zeros for all request messages except GetBulkRequest.
2. Error status field is replaced by nonrepeater field and error index field is replaced by max-repetitions field in GetBulkRequest.

If the manager does not know the indexes of the entries, it cannot retrieve the values. However, it can use Get Next Request and define the Object Id of the table. Because the first entry has the Object Id immediately after the Object Id of the table, the value of the first entry is returned. The manager can use this Object Id to get the value of the next one, and so on.

The fields are listed below:

o PDU type. This field defines the type of the POD (see Table 28.4).

o **Request ID.** This field is a sequence number used by the manager in a Request POD and repeated by the agent in a response. It is used to match a request to a response.

o Error status. This is an integer that is used only in Response PDUs to show the types of errors reported by the agent. Its value is 0 in Request PDUs. Table 28.3 lists the types of errors that can occur.

Status	*Name*	*Meaning*
0	noError	No error
1	tooBig	Response too big to fit in one message
2	noSuchName	Variable does not exist
3	badValue	The value to be stored is invalid
4	readOnly	The value cannot be modified
5	genErr	Other errors

Non repeaters. This field IS used only in Get Bulk Request and replaces the error status field, which is empty in Request PDUs.

o **Error index.** The error index is an offset that tells the manager which variable caused the error.

o **Max~repetition**. This field is also used only in Get Bulk Request and replaces the error index field, which is empty in Request PDUs.

o **Var Bind list**. This is a set of variables with the corresponding values the manager wants to retrieve or set. The values are null in Get Request and Get Next Request. In a Trap PDU, it shows the variables and values related to a specific PDU.

****