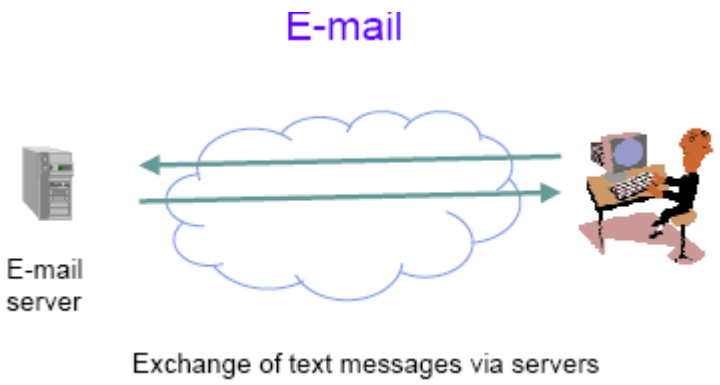
CS1302 COMPUTER NETWORKS
UNIT – I
DATA COMMUNICATION

Need for Communication

A communication service enables the exchange of information between users at different locations.
Communication services & applications are everywhere. Some examples are given below



Need for Computer Communication over Distances

Computer communication has become essential for the following reasons:
(a) Computers can send data at a very fast speed over long distances using satellite and microwave links. Therefore, the cost of transfer of documents using computers is cheaper than Other conventional means like telegrams.
(b) Computers can have fax system integrated with them. This allows them to send pictures along with the text. Thus the newspaper reports can be prepared and sent all over the world at a very high speed by composing and publishing them from different centers.
(c) The time taken to transmit the messages is very small. Hence different computers can be connected together and the users can work together as a group. Software packages have been developed for group working in Data Base Management (DBMS) and graphic works.
(d) Different departments of an organization may be separated physically being at distant places but their data could be stored on a central computer. This data is accessed by computers

located in different departments. The data at the central computer may be updated from time to time and accessed by all users. This prevents any bottlenecks in the smooth functioning of the organization. The latest data (say for inventory) will be easily available at all times to all the users.

(e) Fluctuations of prices in foreign exchange and shares/equities can be communicated instantaneously

using the medium of computer communications only. The transfer can be accelerated and verified at any instant of time.

1.1 Data Communication:

Data Communication is defined as the exchange of data between two devices via some form of transmission medium such as a wire cable. The communicating devices must be a part of a communication system made up of a combination of hardware (physical equipment) and software (programs).

Characteristics of data Communication:

The effectiveness of a data communication depends on three characteristics

1.Delivery
2.Accuracy
3.Timeliness

Delivery : The system must deliver data to correct destination.

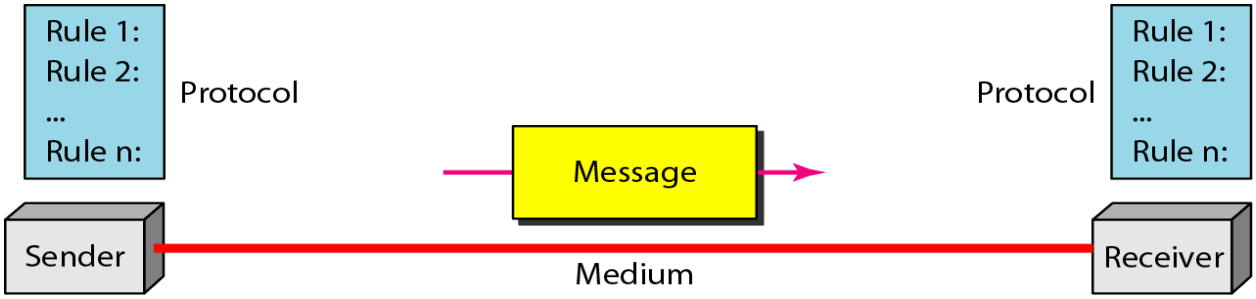
Accuracy: The system must deliver data accurately.

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

Components:

The components of a data communication are

Message
Sender
Receiver
Medium
Protocol

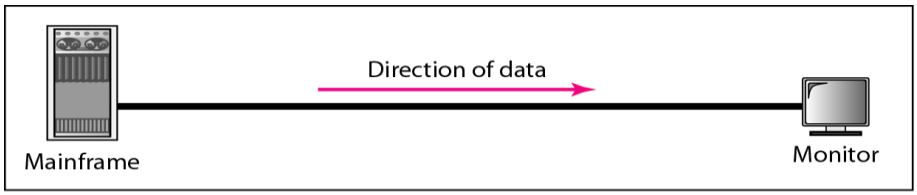


Message : The message is the information to be communicated. It can consist of text ,pictures, numbers, sound, video or audio .
Sender. The sender is the device that sends the data message. It can be a computer or workstation telephone handset, video camera and so on..
Receiver. The receiver is the device that receives the message. It can be a computer or workstation telephone handset, video camera and so on..
Medium. The transmission medium is the physical path by which a message travels from sender to receiver. It could be a twisted pair wire , coaxial cable, fiber optic cable, or radio waves.
Protocol. A protocol is a set of rules that governs data communications. It represents an agreement between the communicating devices.

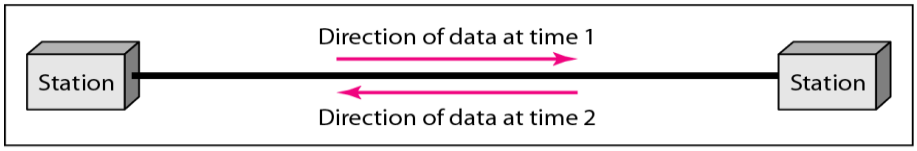
Data representation.
Information comes in different forms such as text, numbers, images, audio and video.

Text.
Text is represented as a bit pattern,
The number of bits in a pattern depends on the number of symbols in the language.
Different sets of bit patterns have been designed to represent text symbols. Each set is called a code. The process of representing the symbols is called coding.
ASCII : The American National Standards Institute developed a code called the American Standard code for Information Interchange .This code uses 7 bits for each symbol.
Extended ASCII : To make the size of each pattern 1 byte(8 bits),the ASCII bit patterns are augmented with an extra 0 at the left.
Unicode : To represent symbols belonging to languages other than English,a code with much greater capacity is needed. Unicode uses 16 bits and can represent up to 65,536 symbols.
ISO:The international organization for standardization known as ISO has designed a code using a 32 – bit pattern. This code can represent up to 4,294,967,296 symbols.
Numbers
Numbers are also represented by using bit patterns. ASCII is not used to represent numbers. The number is directly converted to a binary number.
Images
Images are also represented by bit patterns. An image is divided into a matrix of pixels,where each pixel is a small dot. Each pixel is assigned a bit pattern. The size and value of the pattern depends on the image.The size of the pixel depends on what is called the resolution.
Audio
Audio is a representation of sound. Audio is by nature different from text, numbers or images. It is continuous not discrete
Video
Video can be produced either a continuous entity or it can be a combination of images.

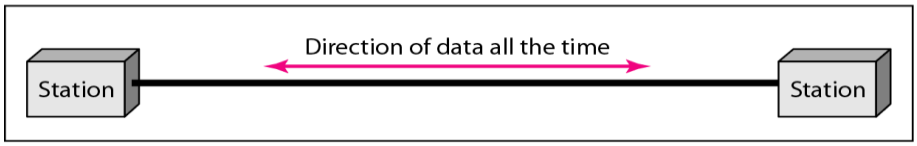
Direction of data flow
Communication between two devices can be simplex, half-duplex or full-duplex



a. Simplex



b. Half-duplex



c. Full-duplex

Simplex

In simplex mode, the communication is unidirectional. Only one of the devices on a link can transmit; the other can only receive.
Ex.Keyboards and monitors. The keyboard can only introduce input. The monitor can only accept output.

Half-duplex

In half-duplex mode ,each station can both transmit and receive but not at the same time. When one device is sending ,the other can only receive.
The half-duplex mode is like a one-lane road with two directional traffic. The entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time.
Ex. Walkie-talkies and CB(citizen band radios.

Full-duplex

In full-duplex mode, both stations can transmit and receive simultaneously. It is like a two-way street with traffic flowing in both directions at the same time. Signals going in either direction share the capacity of the link.
Ex. Telephone network
When two people are communicating by a telephone line, both can listen and talk at the same time.

1.2 Network:

Definition :

An interconnected collection of autonomous computers" interconnected = able to exchange information

A set of nodes connected by communication links .A node can be any device capable of sending &/or receiving data to &/or from other nodes in the network

A connected collection of hardware and software that permits information exchange and resource sharing.
Information = data, text, audio, video, images, …
Resources = printers, memory, link bandwidth

Uses of networks

•companies & organizations •resource sharing: programs, equipment, data…
•high reliability: multiple processors/links/file copies/...
•scalability: gradually improve system performance
•rapid communications & remote cooperation •saving money
•access to remote & diverse information sources •communicating with other people
•entertainment •education, healthcare, access to government...

Distributed Processing

Networks use distributed processing which is termed as a task divided among multiple computers. Instead of a single machine responsible for all aspects of a process, separate computers handle a subset.

Network Criteria

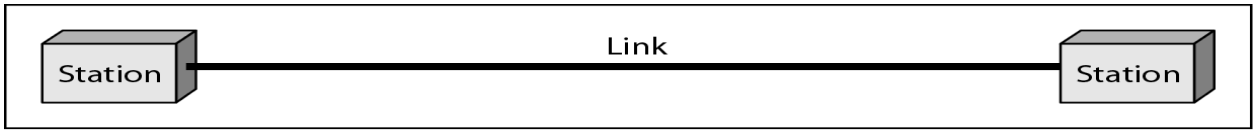
Performance

Performance can be measured by means of transit time, response time, number of users, type of transmission medium, and capabilities of the connected hardware and the efficiency of the software.
Transit time The amount of time required for a message to travel from one device to another.
Response time: The elapsed time between an inquiry and a response.
Reliability: Reliability is measured by the frequency of failure ,the time it takes a link to recover from a failure.
Security: Network security is protecting data from unauthorized access.

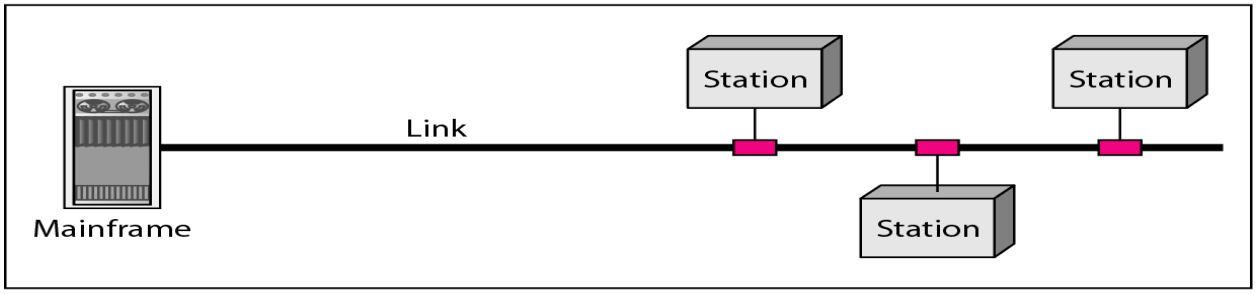
Physical Structures

Type of connection

There are two possible type of connections
Point-to-point
Multipoint
A ***point-to-point*** connection provides a dedicated link between two devices. The entire link is reserved for transmission between those two devices.
Ex. Change of television channel by infrared remote control. A point-to-point connection is established between the remote control and the televisions control system.
A ***multipoint*** (also called multidrop) connection is one in which more than two specific devices share a single link. The capacity of the channel is shared either spatially or temporally.



a. Point-to-point



b. Multipoint

Physical Topology

Physical Topology refers to the way in which network is laid out physically. Two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and the linking devices to one another.

The basic topologies are

Mesh

Star

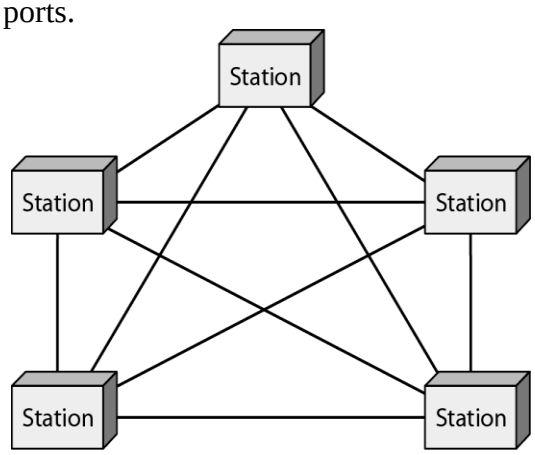
Bus and

Ring

Mesh

In a mesh topology each device has a dedicated point to point link to every other device. The term dedicated means that the link carries traffic only between the two devices it connects.

A fully connected mash network therefore has **n(n-1)/2** physical channels to link n devices. To accommodate that many links every device on the network has **(n-1) I/O** ports.



Merits.

- Dedicated link guarantees that each connection can carry its own data load. This eliminates the traffic problems that occur when links shared by multiple devices.
- If one link becomes unusable, it does not incapacitate the entire system.
- Privacy or security: When every message travels along a dedicated line only the intended recipient

Demerits

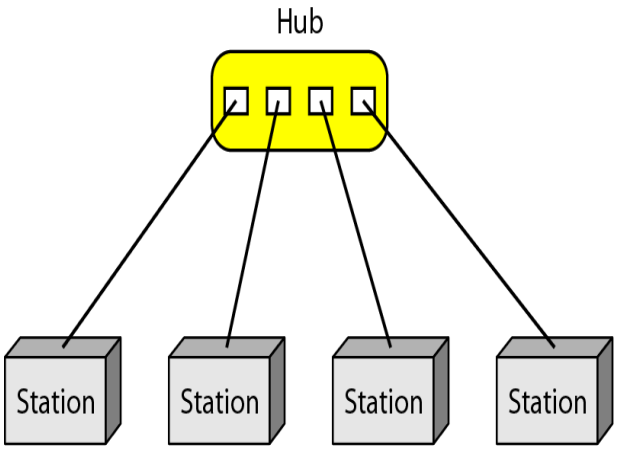
- The amount of cabling and the I/O ports required
- Installation and reconnection are difficult
- The sheer bulk of the wires accommodate more space than available.

The hardware required to connect each link can be prohibitively expensive.

(Mesh)

Star topology

Each device has a dedicated point to point link only to a central controller usually called a hub. If one device has to send data to another it sends the data to the controller, which then relays the data to the other connected device.



Merits

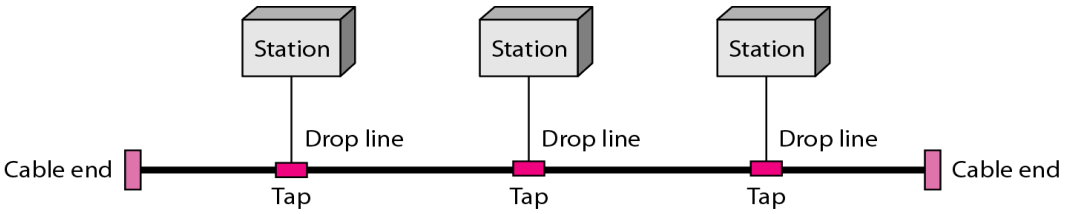
- Less expensive than a mesh topology. Each device needs only one link and I/O port to connect it to any number of others.
- Installation and reconfigure is easy.
- Robustness. If one link fails only that link is affected.
- Requires less cable than a mesh.

Demerits

- Require more cable compared to bus and ring topologies.

Bus

One long cable acts as a backbone to link all the devices in a network Nodes are connected to the bus cable by drop lines and taps. A drop line is a connection running between the device and the main cable. A tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a contact with a metallic core. As the signal travels farther and farther, it becomes weaker .So there is limitation in the number of taps a bus can support and on the distance between those taps.



Merits

- Ease of installation.
- Bus uses less cabling than mesh or star topologies.

Demerits

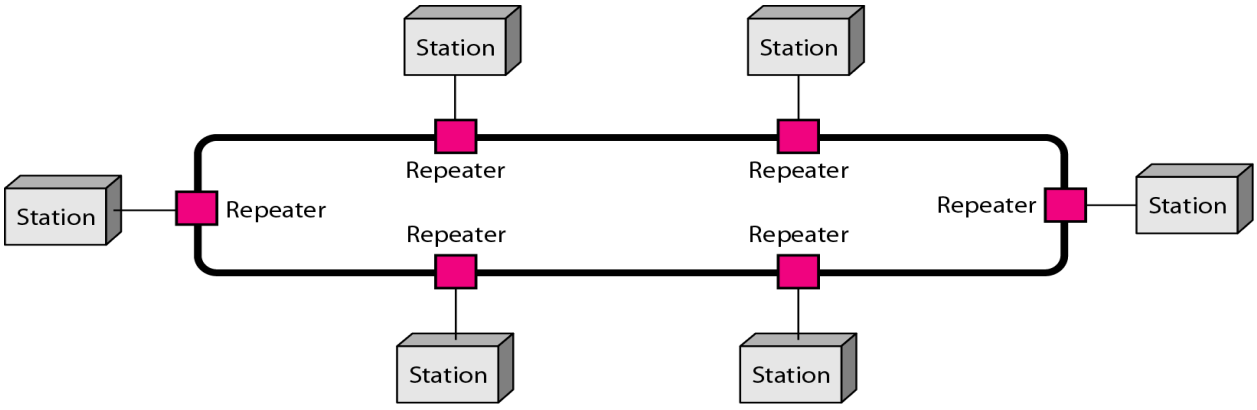
- Difficult reconnection and isolation.
- Signal reflection at the taps can cause degradation in quality.

- A fault or break in the bus cable stops all transmission. It also reflects signals back in the direction of origin creating noise in both directions.

Ring

Each device has a dedicated point to point connection only with the two devices on either side of it.

A signal is passed along the ring in one direction from device to device until it reaches the destination
Each device in the ring incorporates a repeater. It regenerates the bits and passes them along, when it receives the signal intended for another device.



Merits:

- Easy to install and reconfigure.
- To add or delete a device requires changing only two connections.
- The constraints are maximum ring length and the number of devices.
- If one device does not receive the signal within a specified period, it issue an alarm that alerts the network operator to the problem and its location

Demerits

- A break in the ring disables the entire network. It can be solved by using a dual ring or a switch capable of closing off the break.

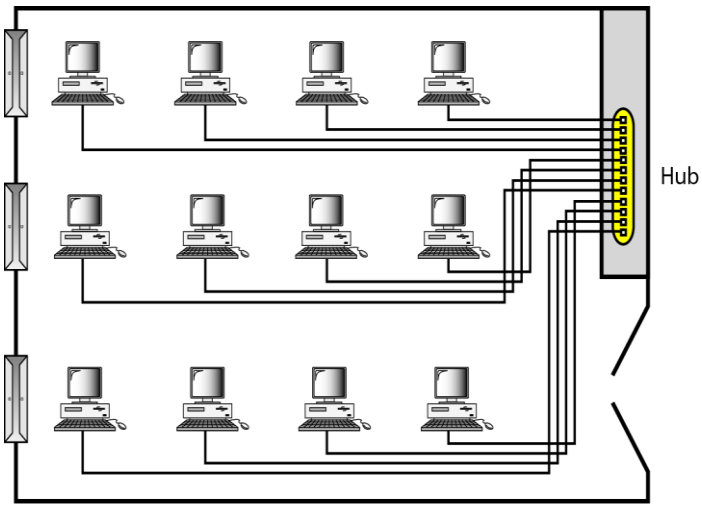
Networks Models

Categories of Network

The three primary categories are of network are Local Area Network (LAN), Metropolitan Area Network (MAN), and Wide Area Network (WAN). The category into which a network fall is determined by its size, ownership, the distance it covers and its physical architecture.

LAN

- A LAN is usually privately owned and links the devices in a single office, building or campus.
- A LAN can be as simple as two PCs or it can extend throughout a company. LAN size is limited to a few kilometers. The most widely used LAN system is the Ethernet system developed by the Xerox Corporation.
- It is designed to allow resources (hardware, software or data) to be shared between PC's or workstations.



- It may be used to provide a (shared) access to remote organizations through a router connected to a Metropolitan Area Network (MAN) or a Wide Area Network (WAN)
- One of the computers may be given a large capacity disk drive and may become a server to other clients. Software can be stored on this server and used by the whole group.
  - The size of the LAN may be determined by the licensing restrictions on the numbers per copy of software or the number of users licensed to access the operating system.
- Also differentiated from other types of network by transmission media and topology.LAN use only one type of transmission medium. The common LAN topologies are bus, ring and star.

- LANs have data rates in the 4 to 10 megabits per second. Can also reach 100 Mbps with gigabit systems in development.

- Intermediate nodes (i.e. repeaters, bridges and switches) allow LANs to be connected together to form larger LANs. A LAN may also be connected to another LAN or to WANs and MAN's using a "router"

Metropolitan Area Network

A MAN is designed to extend over an entire city.

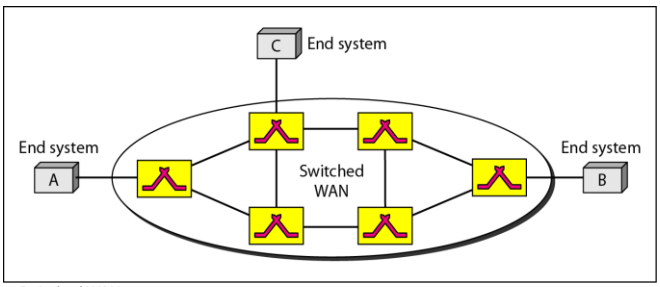
- May be a single network such as cable TV network
- May be a means of connecting a number of LANs into a larger network
- Resources may be shared LAN to LAN as well as device to device

Example: Company can use a MAN to connect the LANs in all its offices throughout a city.

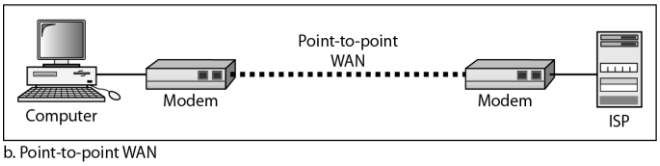
- A MAN can be owned by a private company or it may be a service provided by a public company ,such as local telephone company
- Telephone companies provide a popular MAN service called (SMDS) Switched Multi-megabit Data Services.

Wide Area Network (WAN)

- A WAN provides long distance transmission of data, voice, image and video information over large geographic areas.
- It may comprise a country, continent or even the whole world. Transmission rates are typically 2 Mbps, 34 Mbps, 45 Mbps, 155 Mbps, 625 Mbps (or sometimes considerably more).
- WAN utilize public, leased, or private communication equipment usually in combinations and therefore span an unlimited number of miles.
- A WAN that is wholly owned and used by a single company is referred to as an Enterprise Network. The figure represents the comparison of the different types of networks



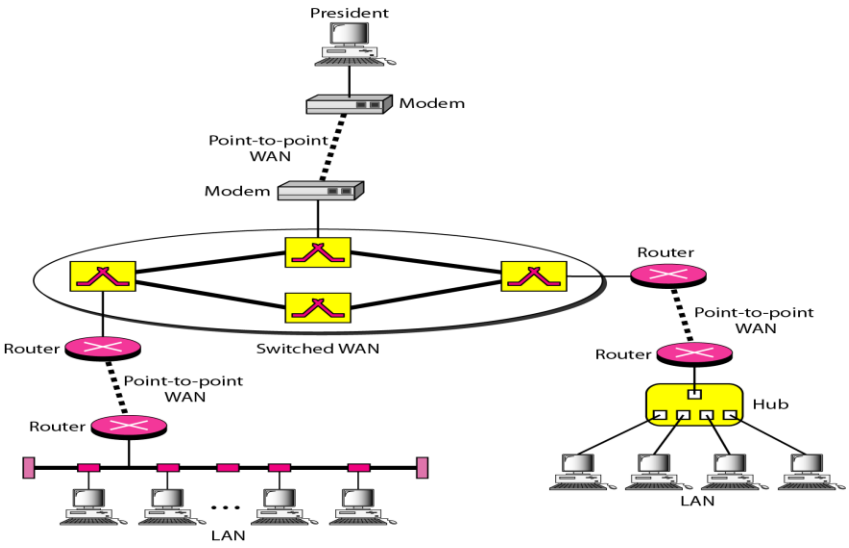
a. Switched WAN



b. Point-to-point WAN

Internetwork

When two or more networks are connected they become an internetwork or internet



1.3 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

Syntax: It refers to the structure or format of the data. This refers the order in which the data are presented.

Example

- The first 8 bits of data to be the address of the sender.
- The second 8 bits to be the address of the receiver.
- The rest of the stream may be the message itself

Semantics: It refers to the meaning of each section of bits. How a particular pattern to be interpreted? What action is to be taken based on that interpretation?

Example

An address specifies the route to be taken or the final destination of the message.

Timing: It refers to two characteristics

When data should be sent and how fast they can be sent.

Example

If a sender produces data at 100 Mbps and the receiver process data at only 1 Mbps, it will overload the receiver and data will be lost.

Standards

Why do we need standards?

- To create and maintain an open and competitive market for equipment manufacturers
- To guarantee national and international interoperability of data, telecommunication technology and process
- To give a fixed quality and product to the customer
- To allow the same product to be re used again elsewhere
- To aid the design and implementation of ideas

- To provide guidelines to manufacturers, vendors, government agencies and other service providers to ensure kind of interconnectivity.

Data communication standards are divided into two categories

De facto(from the fact):

- Standards that have not been approved by an organized body.
- It have been adopted as standards through widespread use.
- This is often established originally by manufacturers to define the functionality of a new product or technology.

De jure (by law):

- Those that have been legislated by an officially recognized body.

Standards organizations

Standards are developed through the cooperation of standards creation committees, forums, and government regulatory agencies.

Standards Creation Committees

ITU - International Telecommunications Union formerly the (CCITT):

- It a standard for telecommunication in general and data systems in particular.

ISO - International Standards Organization:

- It is active in developing cooperation in the realms of scientific, technological and economic activity.

ANSI - American National Standards Institute:

- It is a private nonprofit corporation and affiliated with the U.S federal government.

IEEE - Institute of Electrical and Electronics Engineers:

- It aims to advance theory, creativity, and product quality in the fields of electrical engineering , electronics radio and in all related branches of Engineering.
- It oversees the development and adoption of international standards for computing and communications.

EIA - Electronic Industries Association:

- It is a nonprofit organization devoted to the promotion of electronics manufacturing concerns.
- Its activities include public awareness education and lobbying efforts in addition to standards development.
- It also made significant contributions by defining physical connection interfaces and electronic signaling specifications for data communication.

Forums

- It work with universities and users to test, evaluate ,and standardize new technologies.
- The forums are able to speed acceptance and use of those technologies in the telecommunications community.
- It presents their conclusions to standard bodies.

Regulatory Agencies:

- Its purpose is to protect the public interest by regulating radio, television and wire cable communications.
- It has authority over interstate and international commerce as it relates to communication.

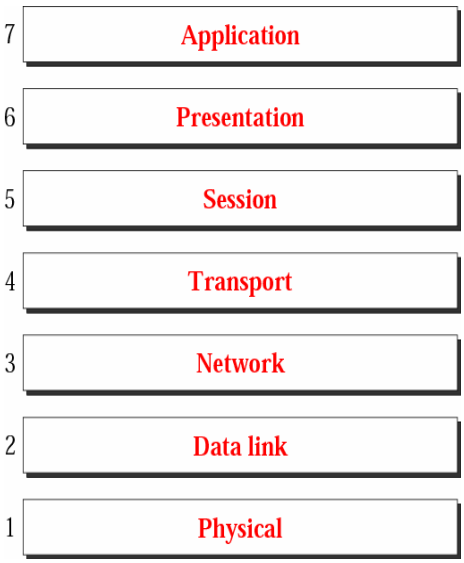
Internet Standards

- It is a thoroughly tested specification that is useful to and adhered to by those who work with the internet.
- It is a formalized regulation that must be followed.
- A specification begins as an internet draft and attains Internet standard status.
- An Internet draft is a working document and it may be published as Request for Comment (RFC).RFC is edited, assigned a number, and made available to all interested parties.

2.2OSI

The Open Systems Interconnection (OSI) architecture has been developed by the International Organization for Standardization (ISO) to describe the operation and design of layered protocol architectures. This forms a valuable reference model and defines much of the language used in data communications.

Layered Architecture



>Each layer provides services to the next higher layer and shields the upper layer from the details implemented in the lower layers.

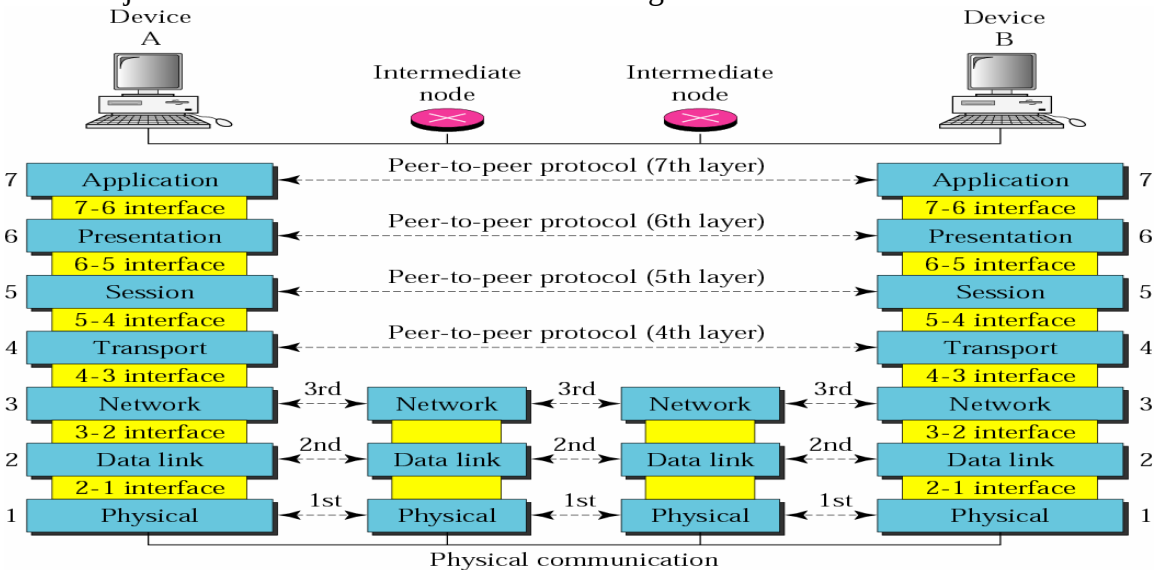
>Each layer appears to be in direct (virtual) communication with its associated layer on the other computer. Actual communication between adjacent layers takes place on one computer only.

>Layering simplifies design, implementation, and testing. Only the lowest level (Physical layer) can directly communicate with its peer communications process into parts.

Peer-to-Peer Processes

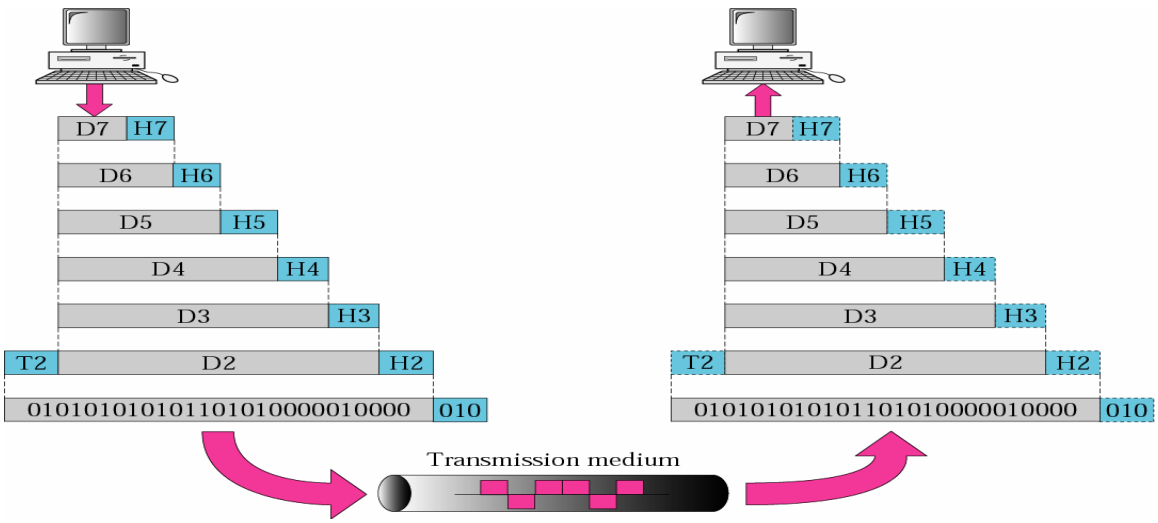
- The processes on each machine that communicate at a given layer are called peer-to-peer processes.
- At higher layers communication must move down through the layers on device A aver to device B and then back up through the layers.

- Each layer in the sending device adds its own information to the message it receives from the layer just above it and passes the whole package to the layer just below and transferred to the receiving device.



Interfaces between layers

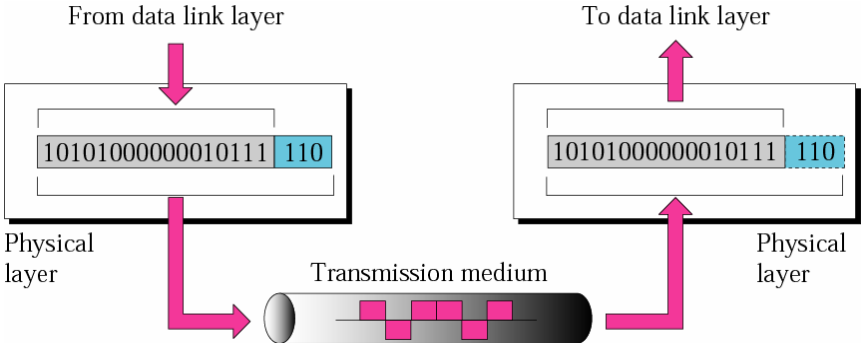
- The passing of data and network information down through the layers of the sending device and back up through the layers of the receiving device is made possible by an interface between each pair of adjacent layers.
- Each interface defines what information and services a layer must provide for the layer above it.
- Well defined interfaces and functions provide modularity to a network.



2.3 Layers in the OSI model

Physical Layer

- It coordinates the functions required to transmit a bit stream over a physical medium.
- It deals with the mechanical and electrical specifications of the interface and transmission media.
  Mechanical: cable, plugs, pins...
  Electrical/optical: modulation, signal strength, voltage levels, bit times
- It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to occur.

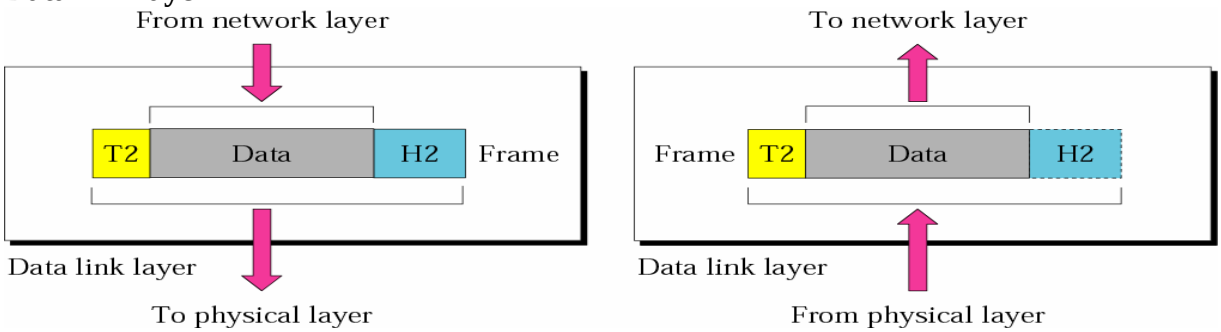


(Information flows from top to bottom at the sender and bottom to top at the receiver.)

Major responsibilities of Physical layer are

- **Physical characteristics of interfaces and media:** It defines the characteristics of the interface between the devices and the transmission media. Also defines the type of transmission medium.
- **Representation of bits**: To transmit the bits, it must be encoded into electrical or optical signals. It defines the type of representation how 0s and 1s are changed to signals.
- **Data rate:** The number of bits sent each second is also defined by the physical layer.
- **Synchronization of bits:** Sender and the receiver must be synchronized at the bit level .i.e the sender and the receiver clocks must be synchronized.

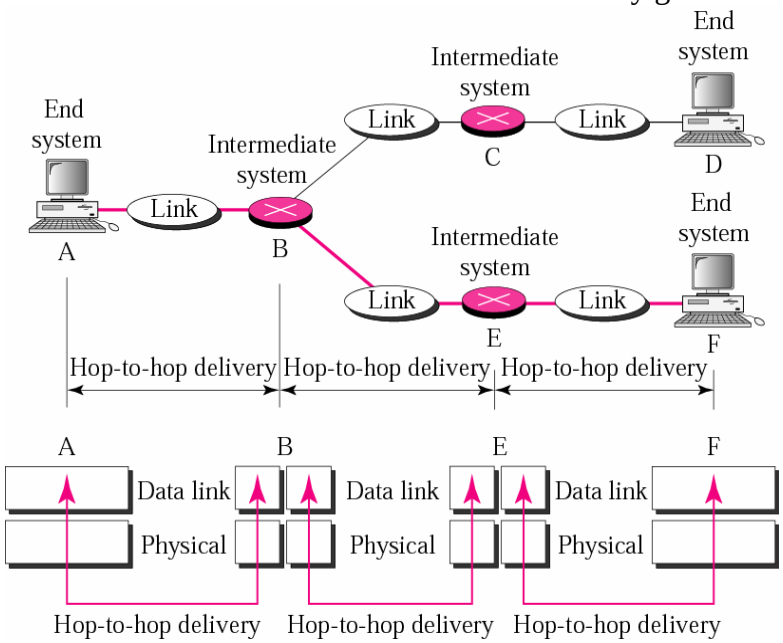
Data link layer



The data link layer is responsible for hop-to-hop (node-to-node) delivery. It transforms the physical layer a raw transmission facility to a reliable link. It makes physical layer appear error free to the network layer.

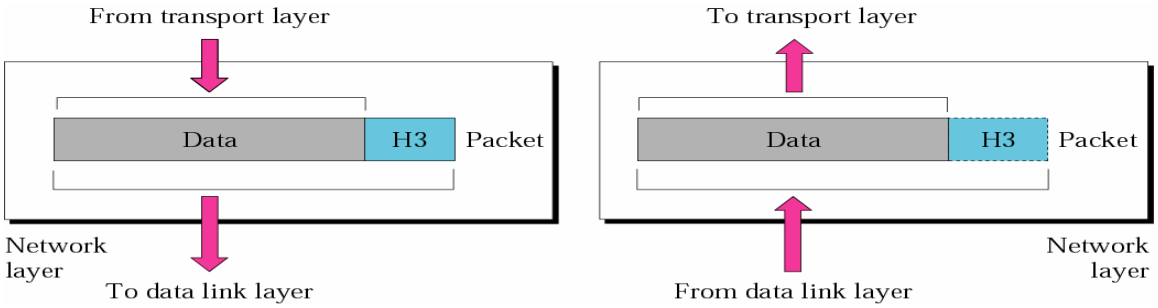
The duties of the data link layer are

- **Framing:** The data link layer divides the stream of bits received from the network layer into manageable data units called frames.
- **Physical Addressing:** If the frames are to be distributed to different systems on the network the data link layer adds a header to the frame to define the receiver or sender of the frame. If the frame is intended for a system located outside the senders network then the receiver address is the address of the connecting device that connects the network to the next one.
- **Flow Control:** If the rate at which the data absorbed by the receiver is less than the rate produced in the sender, the data link layer imposes a flow control mechanism to overwhelming the receiver.
- **Error control** Reliability is added to the physical layer by data link layer to detect and retransmit loss or damaged frames. and also to prevent duplication of frames. This is achieved through a trailer added to the end of the frame
- **Access control** When two or more devices are connected to the same link it determines which device has control over the link at any given time.



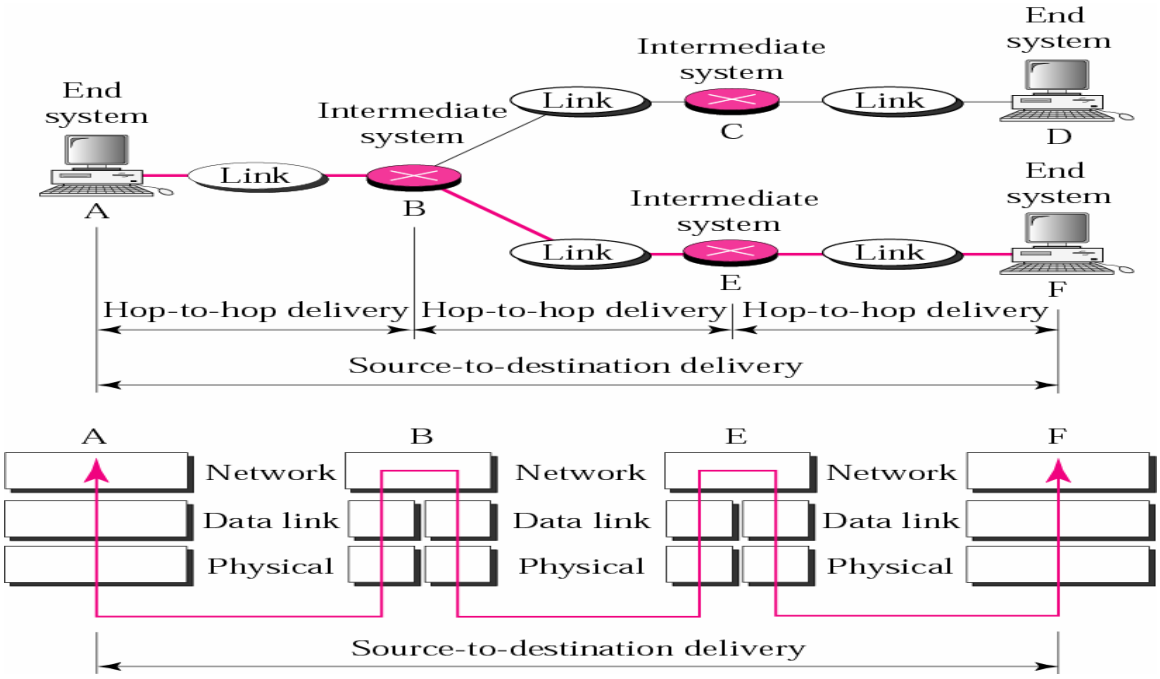
Network Layer

The network layer is responsible for source-to-destination delivery of a packet across multiple networks. It ensures that each packet gets from its point of origin to its final destination. It does not recognize any relationship between those packets. It treats each one independently as though each belong to separate message.



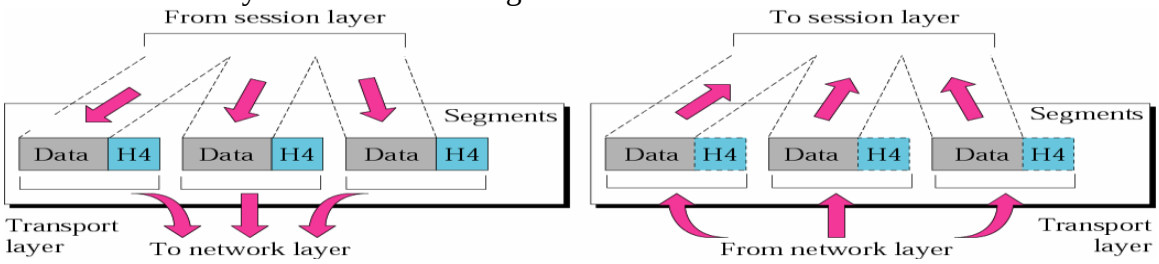
The functions of the network layer are

- **Logical Addressing** If a packet has to cross the network boundary then the header contains information of the logical addresses of the sender and the receiver.
- **Routing** when independent networks or links are connected to create an internetwork or a large network the connective devices route the packet to the final destination.



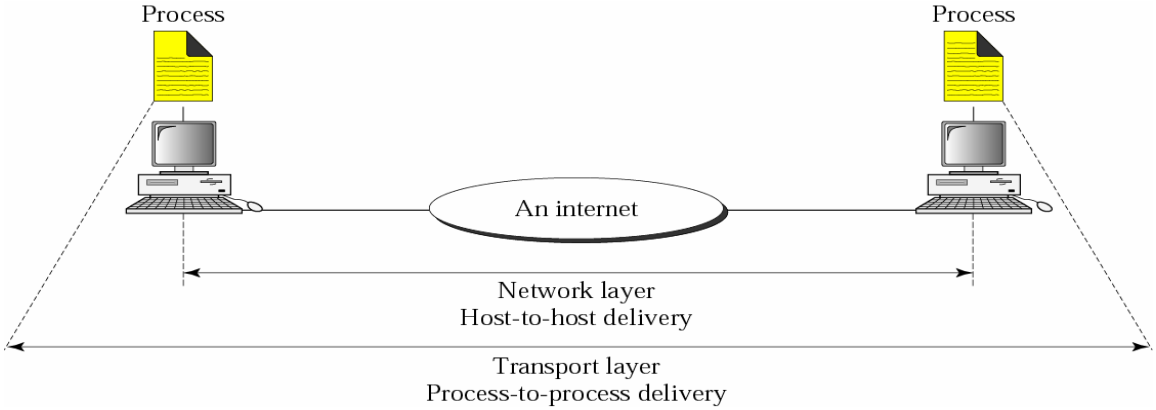
Transport Layer

The network layer is responsible for process-to-process delivery that is source to destination delivery of the entire message.



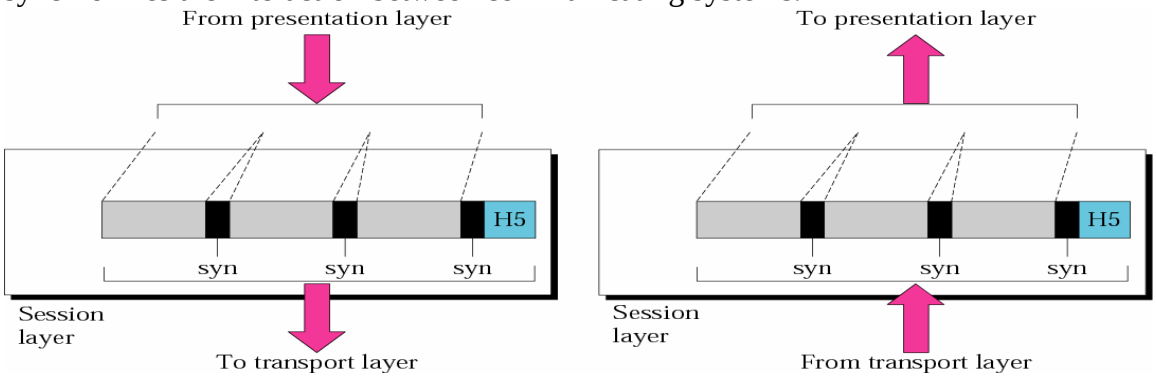
The responsibilities of Transport layer are

- **Service-point (port) addressing**: Computers run several programs at the same time. Source-to-destination delivery means delivery from a specific process on one computer to a specific process on the other. The transport layer header therefore includes a type of address called a service – point address.
- **Segmentation and reassembly**: A message is divided into segments and each segment contains a sequence number. These numbers enable the Transport layer to reassemble the message correctly upon arriving at the destination. The packets lost in the transmission is identified and replaced.
- **Connection control:** The transport layer can be either connectionless or connection-oriented. A connectionless transport layer treats segment as an independent packet and delivers it to the transport layer. A connection-oriented transport layer makes a connection with the transport layer at the destination machine and delivers the packets. After all the data are transferred the connection is terminated.
- **Flow control:** Flow control at this layer is performed end to end .
- **Error Control**: Error control is performed end to end. At the sending side ,the transport layer makes sure that the entire message arrives at the receiving transport layer with out error. Error correction is achieved through retransmission.



Session Layer

Session layer is the network dialog controller. It establishes, maintains, and synchronizes the interaction between communicating systems.

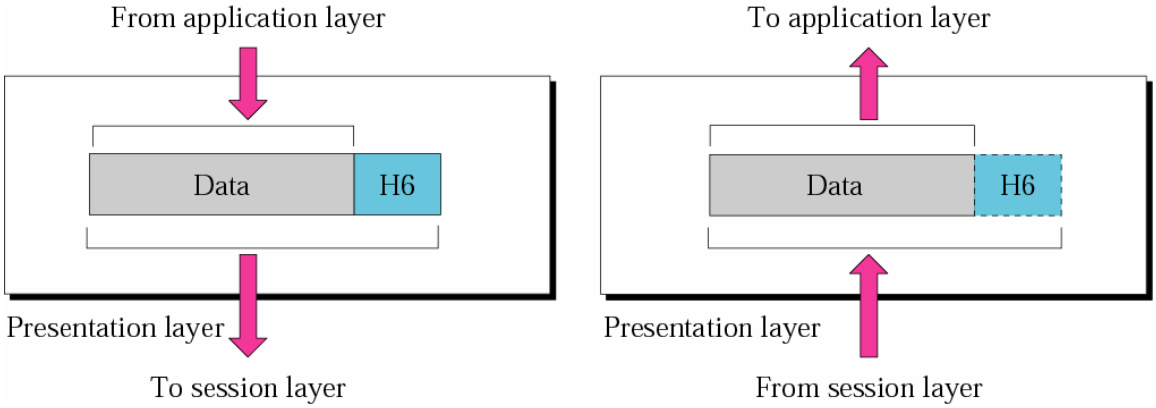


Specific responsibilities of the layer are

- **Dialog Control:** Session layer allows two systems to enter in to a dialog. Communication between two processes takes place either in half-duplex or full-duplex. Example: The dialog between terminals connected to a mainframe. Can be half-duplex.
- **Synchronization.** The session layer allows a process to add checkpoints into a stream of data. Example If a system is sending a file of 2000 pages, check points may be inserted after every 100 pages to ensure that each 100 page unit is advised and acknowledged independently. So if a crash happens during the transmission of page 523, retransmission begins at page 501, pages 1 to 500 need not be retransmitted.

Presentation layer

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

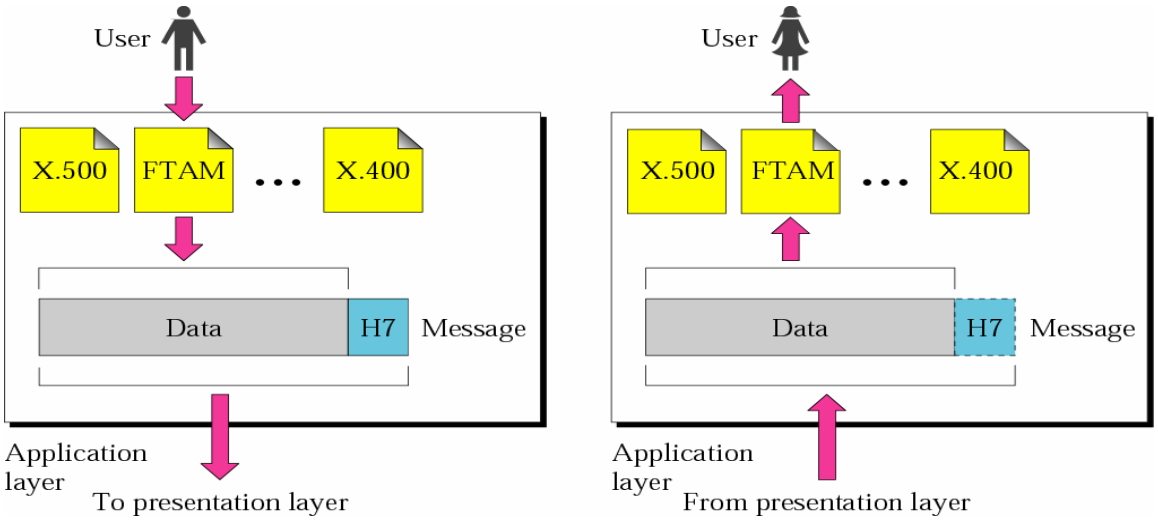


Responsibilities of the presentation layer are

- **Translation:** The processes in two systems are usually exchanging information in the form of character strings, numbers, and so on. The since different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. At the sender, the presentation layer changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver dependent format.
- **Encryption:** The sender transforms the original information from to another form and sends the resulting message over the entire network. Decryption reverses the original process to transform the message back to its original form.
- **Compression**: It reduces the number of bits to be transmitted. It is important in the transmission of text, audio and video.

Application Layer

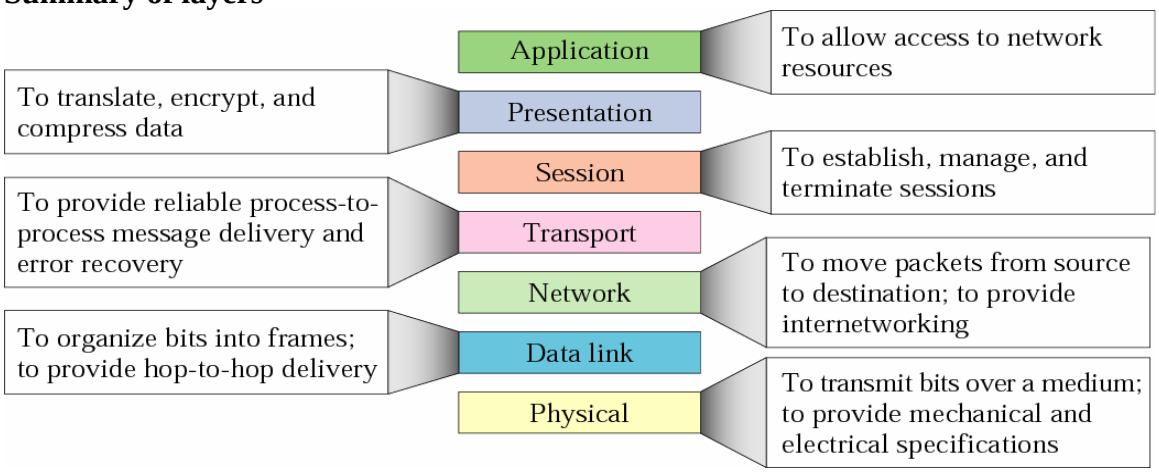
It enables the user (human/software) to access the network. It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management and other types of distributed information services.



Services provided by the application layer are

- **Network Virtual terminal**: It is a software version of a physical terminal and allows a user to log on to a remote host.
- **File transfer, access and management**. This application allows a user to access files in a remote computer, to retrieve files from a remote computer and to manage or control files in a remote computer.
- **Mail services**. This application provides the basis for e-mail forwarding and storage.
- **Directory services.** It provides distributed database sources and access for global information about various objects and services.

Summary of layers

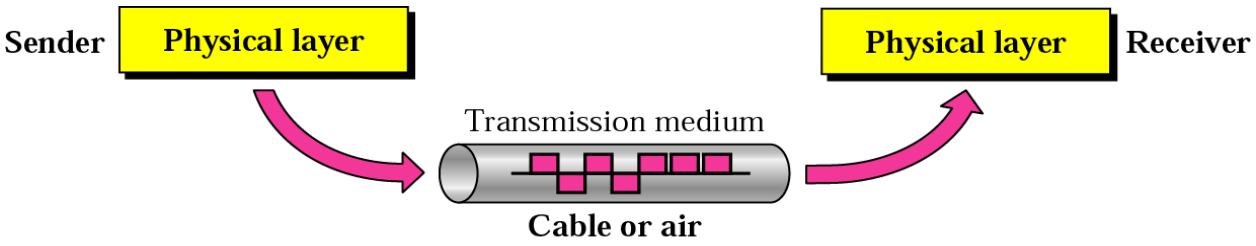


7 Transmission Media

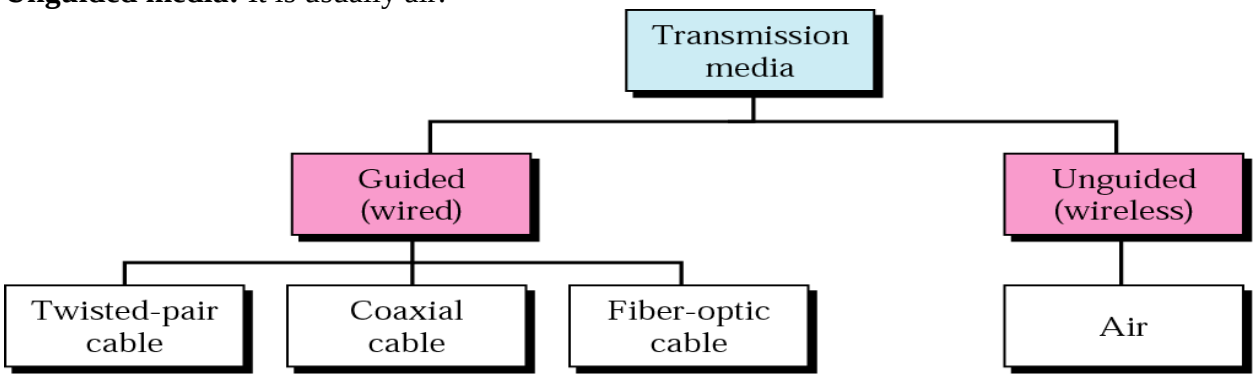
Transmission media are actually located below the physical layer and directly controlled by the physical layer.

Transmission media can be divided into two broad categories

- Guided &
- Unguided



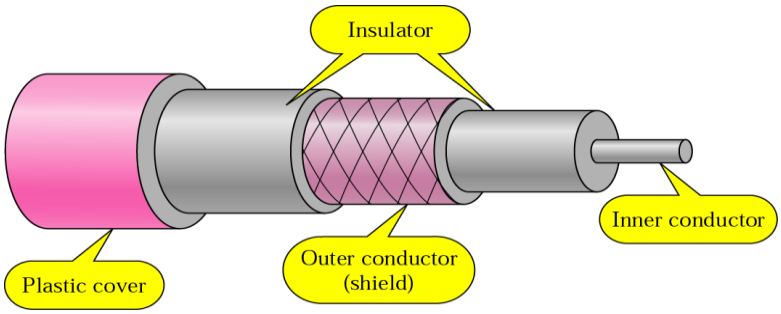
Guided media: It includes twisted-pair cable, coaxial cable, and fiber-optic cable
Unguided media: It is usually air.



7.1 Guided media

Guided media, which are those that provide a conduit from one device to another, include twisted-pair cable, coaxial cable, and fiber-optical cable.

Coaxial cable



- Coaxial cable carries signals of higher frequency ranges than twisted pair cable.

- It has a central core conductor of solid or stranded wire enclosed in an insulating sheath. This in turn encased in an outer conductor of metal foil, braid or a combination of the two.
- The metallic wrapping serves both as a shield against noise and as the second conductor completes the circuit.
- The outer conductor is also enclosed in an insulating sheath and the whole cable is protected by a plastic cover.

Coaxial cable Standards

Coaxial cables are categorized by their radio government (RG) ratings.Each RG number denotes a set of physical specifications such as,

- wire gauge of the inner conductor

- thickness and type of the inner insulator
- the construction of the shield
- the size and type of outer casing

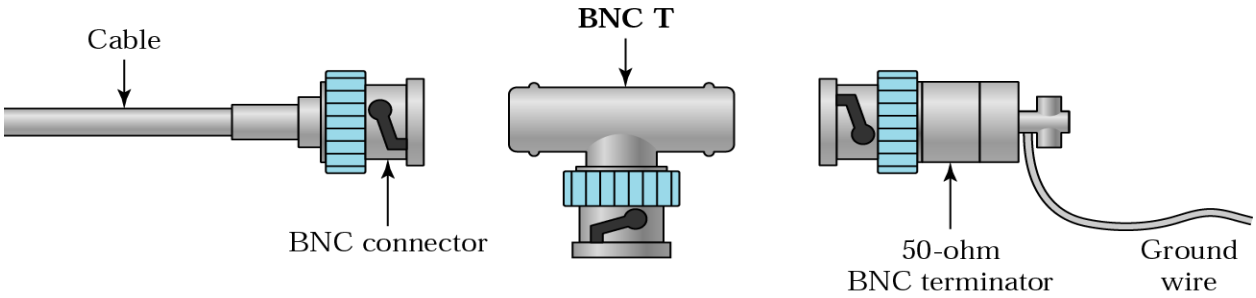
Categories of coaxial cables

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

Coaxial Cable Connectors

Coaxial Cable Connectors are used to connect coaxial cable to devices. The most common type of connector is the Bayone Neill-concelman or BNC connectors. There are three popular types of connectors

- BNC connector
- BNC T connector &
- BNC terminator



BNC connector	**BNC T connector**
It is used to connect the end of the cable to a device such as a TV set.	It is used in Ethernet networks to branch out a cable for connection to a computer or other devices.
BNC terminator It is used at the end of the cable to prevent the reflection of the signal. Performance • Attenuation is much higher in coaxial cables than in twisted pair cable. • Coaxial cable has a much higher bandwidth the signal weakens rapidly and needs the frequent use of repeaters.	

Basic definitions

- Signal Attenuation is the phenomenon whereby the amplitude of a signal decreases as it propagates along a transmission line.
- Attenuation is a function of distance and frequency of signal
- Repeaters are used to increase the power of the signal at appropriate intervals
- Skin effect, which increases attenuation as the bit rate of the transmitted signal increases

Applications

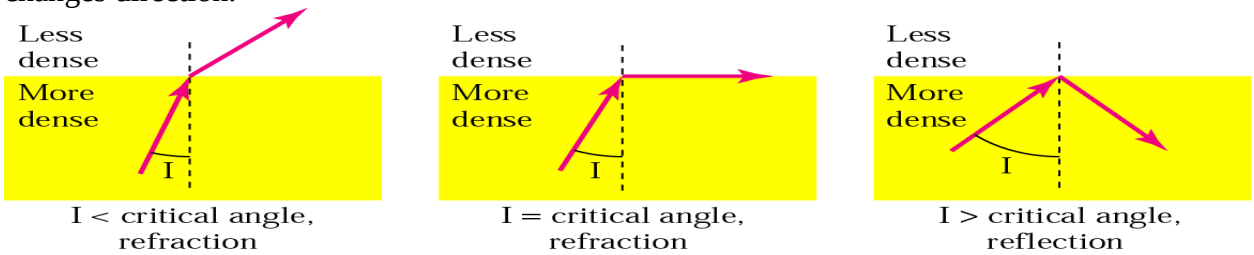
- Coaxial cable is used in analog telephone network where a single coaxial cable could carry 10,000 voice signals.
- It is also used in digital telephone network where a cable could carry digital data up to 600 Mbps.
- Cable TV networks also used RG-59 coaxial cables.
- It is also used in traditional Ethernets.

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 array traveling through one substance suddenly enters another the ray changes direction.



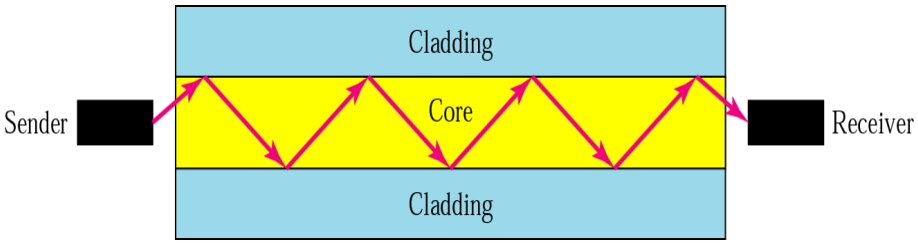
Refraction:

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

Reflection:

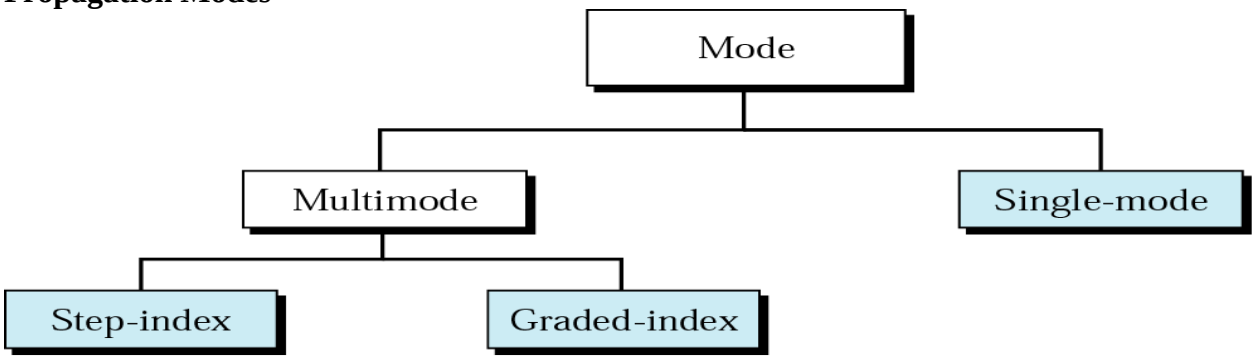
If the angle of incidence is greater than the critical angle the ray reflects and travels again in the denser substance.

Optical fibers use reflection to guide light through a channel.



A glass or plastic core is surrounded by a cladding of less dense glass or plastic. The difference in the density of the two materials must be such that a beam of light moving through the core is reflected off the cladding.

Propagation Modes



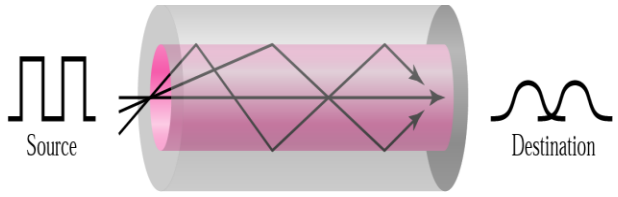
There are two modes for propagating light along optical channels; each requires fiber with different physical characteristics

- Multimode
- Single mode

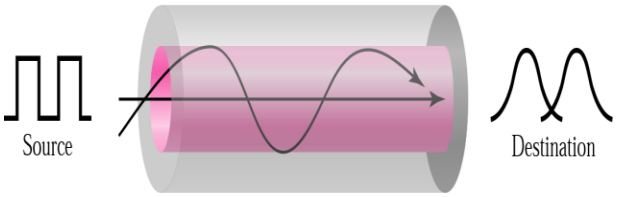
Multimode

Multiple beams from a light source move through the core in different paths.
Multimode can be implemented in two forms

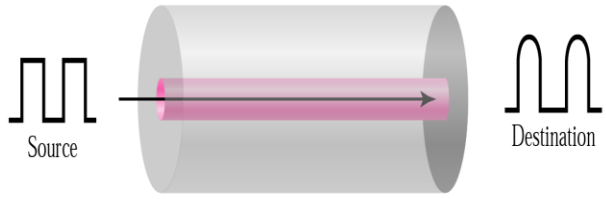
- Step-index
- Graded index



a. Multimode, step-index



b. Multimode, graded-index



c. Single-mode

Multimode Step –index fiber

- In Multimode Step –index fiber the density of the fiber remains constant from the center to the edges
- A beam of light moves through this constant density in a straight line.
- When it reaches the interface of the core and the cladding, there is an abrupt change to a lower density that alters the angle of the beams motion.
- Step-index -> the suddenness of this change.

Multimode Graded-index fiber

- It decreases the distortion of the signal through the cable.
- Density is highest at the center of the core and decreases gradually to its lowest at the edge.

Single-Mode

- It 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

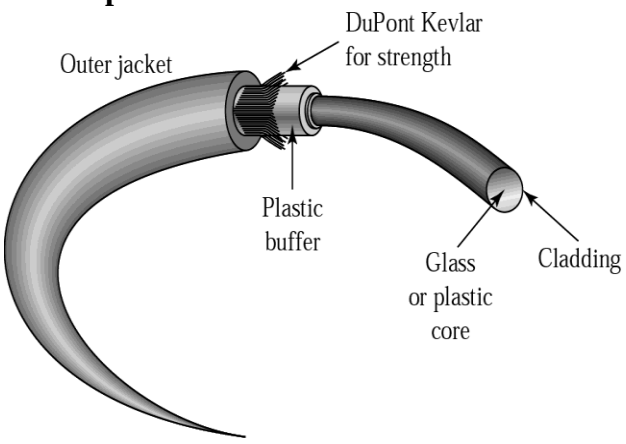
- The Single-Mode fiber itself is manufactured with a smaller diameter than that of multimode fiber and with lower density.
- This results in a critical angle that is close enough to 90· To make it horizontal.
- All the beams arrive at the destination together and can be recombined with little distortion to the signal.

Fiber Sizes

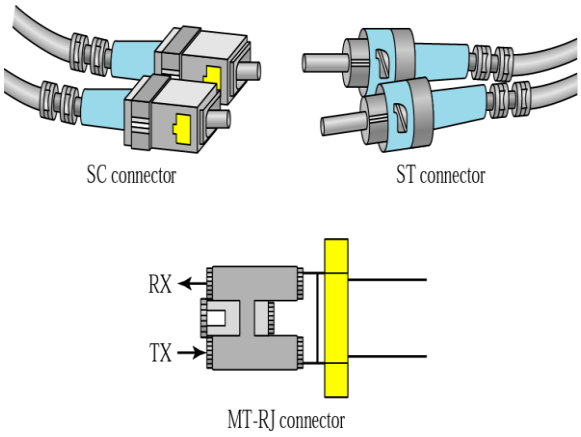
Optical fibers are defined by the ratio of the diameter of their core to the diameter of their cladding expressed in micrometers.

Type	Core	Cladding	Mode
50/125	**50**	**125**	**Multimode, graded-index**
62.5/125	**62.5**	**125**	**Multimode, graded-index**
100/125	**100**	**125**	**Multimode, graded-index**
7/125	**7**	**125**	**Single-mode**

Fiber-optic cable connectors.



Fiber construction



Fiber-optic cable connectors

Three different types of connectors are used by fiber –optic cable.

SC (subscriber channel) Connector:

It is used in cable TV.

ST(Straight-tip) Connector:

It is used for connecting cable to networking devices .

Performance :

- Attenuation is flatter than in the case of twisted pair cable and coaxial cable.
- Few repeaters are needed when we use fiber optic cable.

Application

It is used in cable TV and LAN (Fast Ethernet and 100Base –X.

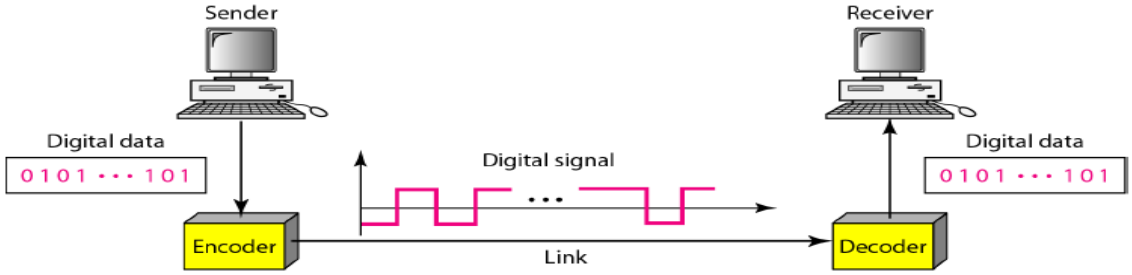
Advantages

Higher bandwidth: It can support higher bandwidth than twisted pair or coaxial cable.

Less signal attenuation: Transmission distance is greater than that of other guided media. Signals can be transmitted for 50 km without requiring regeneration.
Immunity to electromagnetic Interference : Electromagnetic noise can not affect fiber-optic cables
Resistance to corrosive materials: glass is more resistant to corrosive materials.
Light-weight: It is of less weight than the copper cables.
More Immune to taping: Fiber-optic cables are more immune to taping than copper cables.
Disadvantages :
Installation/Maintenance. Installation/Maintenance need expertise since it is a new technology.
Unidirectional: Propagation of light is unidirectional. Bidirectional communication is achieved by means of two optical fibers.
Cost: It is more expensive and the use of optical fiber cannot be justified if the need for bandwidth is not high.

4.1 Line Coding

Line Coding is the process of converting binary data , a sequence of bits , to a digital signal.

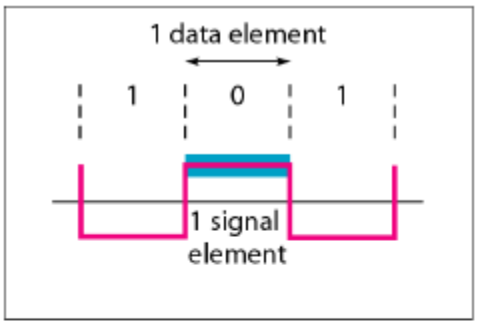


Characteristics of Line coding
Some characteristics of line coding are

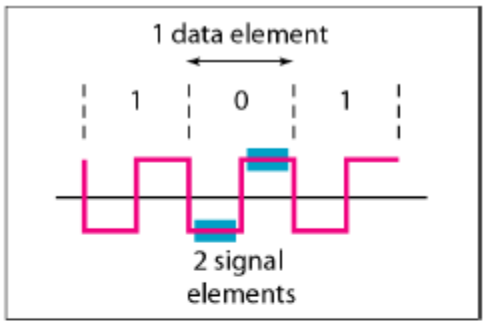
- Signal level versus data level
- Pulse rate vs. bit rate
- Dc components and
- Self-synchronization

Signal level: The number of values allowed in a particular signal is termed as signal level.
Data level: The number of values used to represent data are termed as data level.
DC component (zero frequency):
If the positive voltages are not get cancelled by the negative voltages then it is called a dc component. This component is undesirable for 2 reasons They are

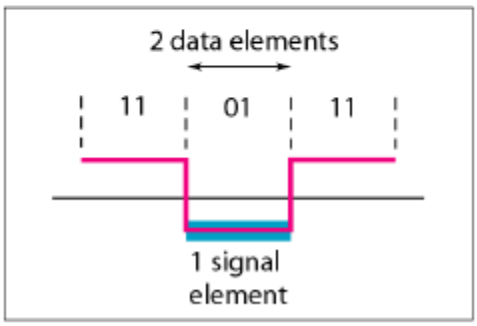
- If the signal is to pass through a system that does not allow the passage of a dc component, the signal is distorted and may create errors in the output.
- This component is an extra energy residing on the line and is useless.



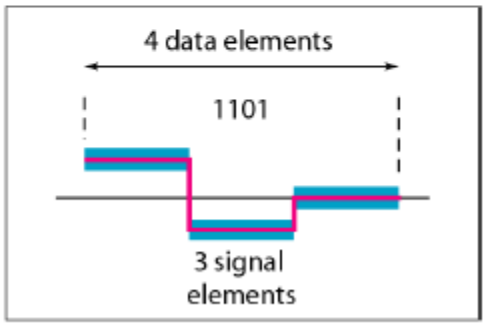
a. One data element per one signal element (r = 1)



b. One data element per two signal elements $\left(r = \frac{1}{2}\right)$



c. Two data elements per one signal element (r = 2)

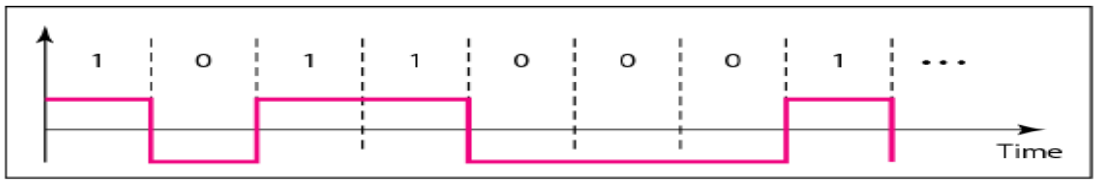


d. Four data elements per three signal elements $\left(r = \frac{4}{3}\right)$

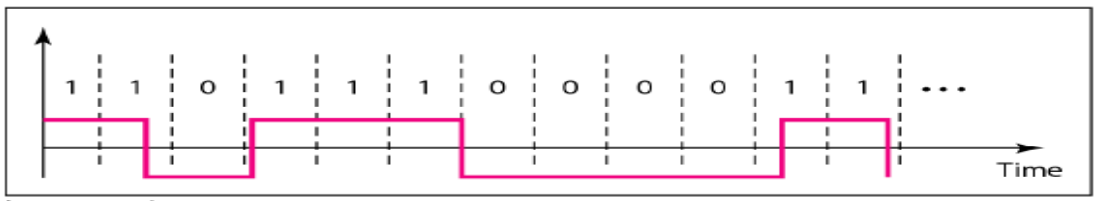
Self-synchronization:

Need: To correctly interpret the signals received from the sender, the receivers bit intervals must correspond exactly to the senders bit intervals. If the receiver clock is faster or slower, the bit intervals are not matched and the receiver might interpret the signals differently than the sender intended.

A Self-synchronizing digital signal includes timing information in the data being transmitted. This can be achieved if there are transitions in the signal that alert the receiver to the beginning, middle or end of the pulse. If the receiver's clock is out of synchronization, these alerting points can reset the clock



a. Sent

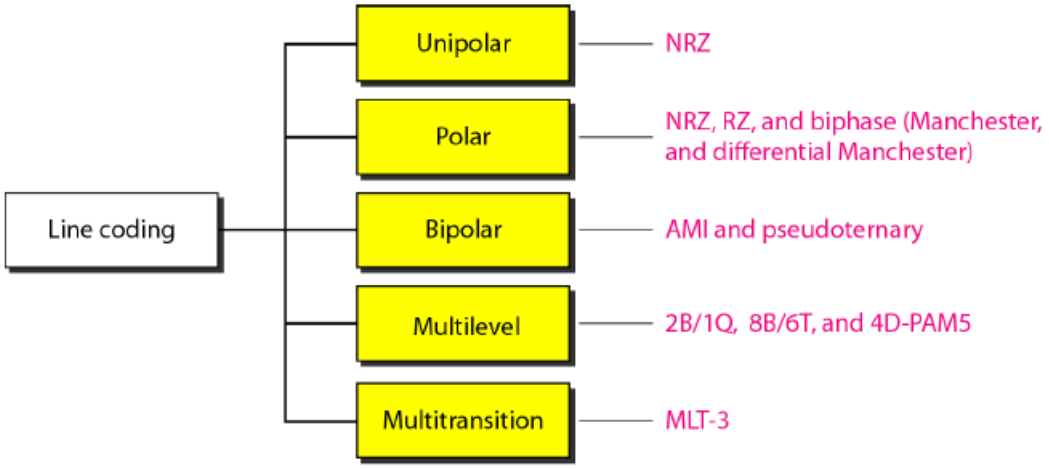


b. Received

Line coding schemes(digital to digital transmission)

Line coding schemes are divided in to three categories.

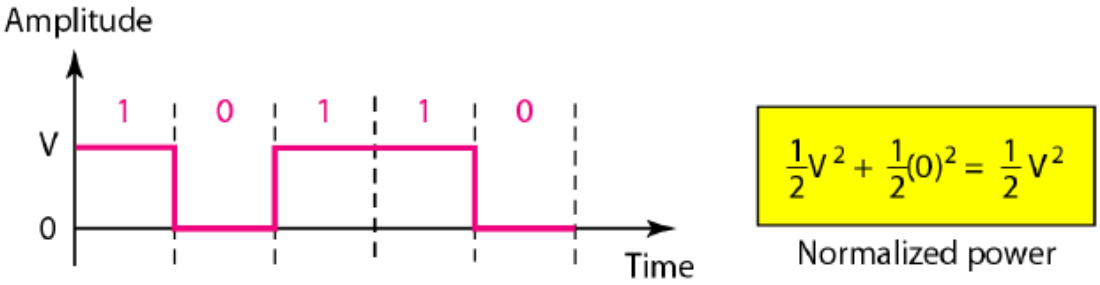
1. Unipolar
2. Polar
3. Bipolar



Unipolar

Unipolar encoding uses only one polarity.0 is represented by zero voltage and 1 is represented by positive voltage.It are inexpensive to implement. Unipolar encoding has two problems

1. Lack of synchronization
2. A dc component



Polar encoding:

It uses two voltage levels

1. Positive
2. Negative

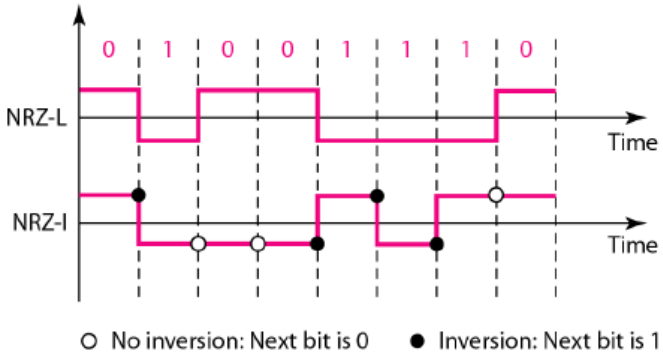
The types of polar encoding are

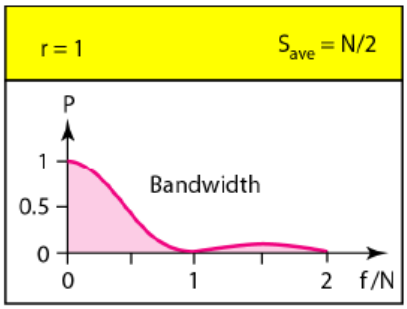
1. Non return to zero(NRZ)
2. Return to zero(RZ)
3. Biphase

NRZ

The level of the signal is always either positive or negative.

NRZ-L
The level of the signal depends on the type of bit it represents.
The bit 0 is represented by positive voltage
The bit 1 is represented by negative voltage.
Demerits
Problem arises when there is a long stream of 0s or 1s in the data.
If the receiver receives a continuous voltage ,it should determine how many bits are sent by relying on its clock.
The receiver may or may not be synchronized with the sender clock





NRZ-I
The 1 bit is represented by an inversion (transition between a positive and a negative voltage) of the voltage level.
The existence of 1's in the data stream allows the receiver to resynchronize its timer to the actual arrival of the transmission.
A string of 0's can still cause problems.

RZ
It uses three values

- Positive
- Negative &
- Zero

In RZ the signal changes during each bit.

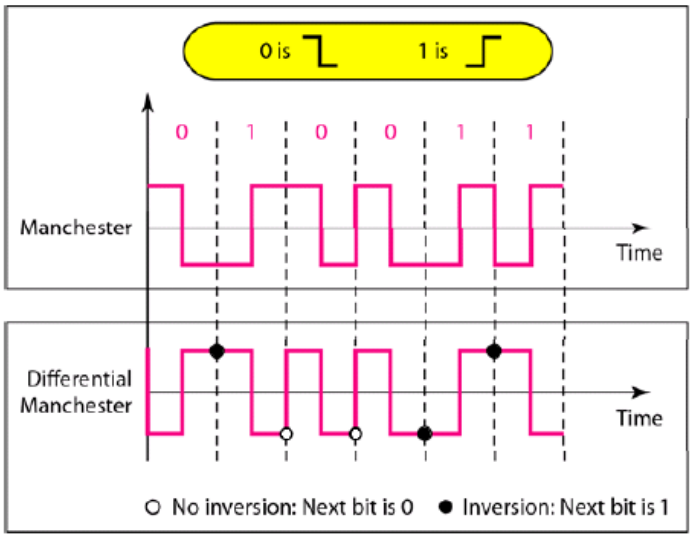
1. A 1 bit is actually represented by positive-to-zero and
2. A 0 bit is actually represented by negative-to-zero

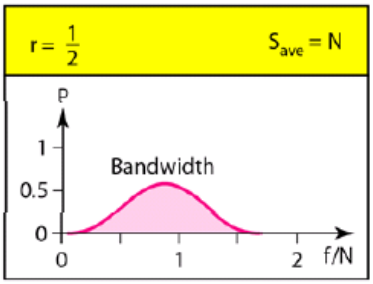
Demerits

- It requires two signal changes to encode one bit.
- It occupies more bandwidth.

Biphase
The signal changes at the middle of the bit interval and does not return to zero.
There are two types of biphase encoding

- Manchester
- Differential Manchester



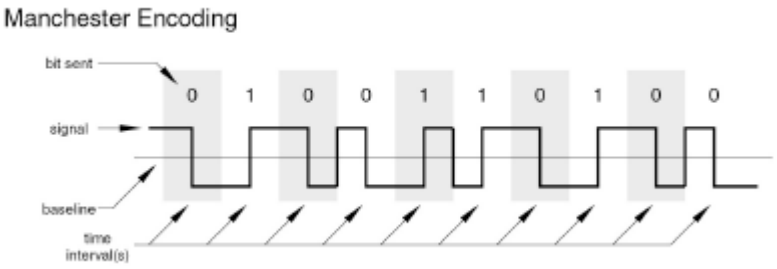


Manchester

- It uses the inversion at the middle of each bit interval for both synchronization and bit representation.
- The bit 1 is represented by negative -to-positive transition.
- The bit 0 is represented by positive-to-negative transition.

Merits

A single transition achieves the level of synchronization but with only two levels of amplitude



Differential Manchester

Inversion at the middle of the bit interval is used for synchronization.
Presence or absence of additional transition at the beginning of the interval is used to identify the bit.
A bit 0 is represented by a transition.
A bit 1 means no transition.
It requires two signal changes to represent binary 0,but only one to represent binary 1.

Bipolar
It uses three voltage levels
Positive

Negative and
Zero

- The bit 0 is represented by zero level
- The 1s are represented by alternate positive and negative voltages.If the first 1 bit is represented by positive amplitude,the second will be represented by the negative amplitude, and so on.

There are three types of bipolar encoding

1. AMI
2. B8ZS
3. HDB3

Bipolar Alternate Mark Inversion

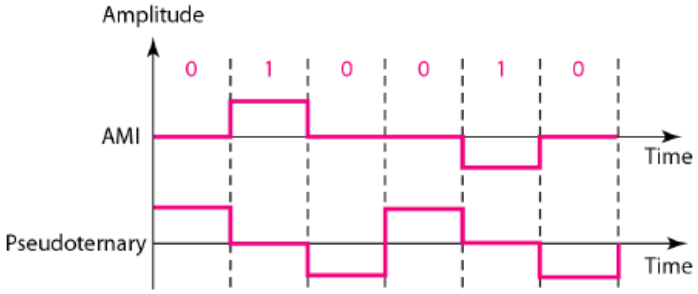
A binary 0 is represented by zero voltage.
A binary 1s are represented by alternate positive and negative voltages.

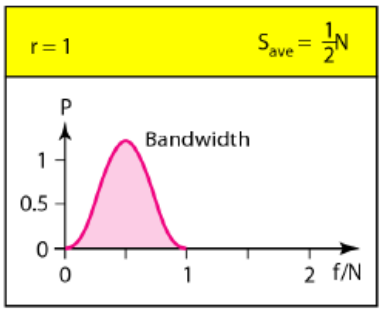
Merits

By inverting on each occurrence of 1,
The dc component is zero
A long sequence of 1s stays synchronized.

Pseudoternary

A binary 0 alternate between positive and negative voltages.





Comparison

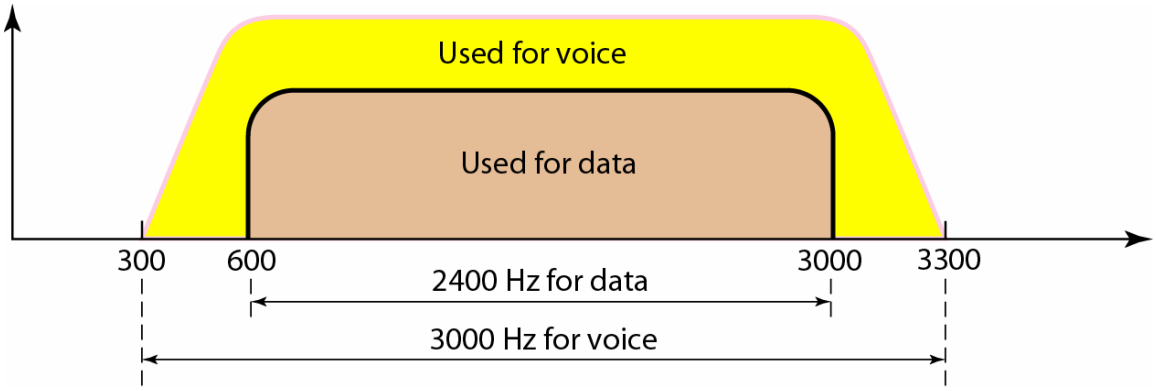
The comparison of the different encoding schemes of the following NRZ, Polar NRZ,NRZ Inverted, Bipolar, Manchester, Differential Manchester are given.

Category	*Scheme*	*Bandwidth (average)*	*Characteristics*
Unipolar	NRZ	$B = N/2$	Costly, no self-synchronization if long 0s or 1s, DC
Unipolar	NRZ-L	$B = N/2$	No self-synchronization if long 0s or 1s, DC
	NRZ-I	$B = N/2$	No self-synchronization for long 0s, DC
	Biphase	$B = N$	Self-synchronization, no DC, high bandwidth
Bipolar	AMI	$B = N/2$	No self-synchronization for long 0s, DC
Multilevel	2B1Q	$B = N/4$	No self-synchronization for long same double bits
	8B6T	$B = 3N/4$	Self-synchronization, no DC
	4D-PAM5	$B = N/8$	Self-synchronization, no DC
Multiline	MLT-3	$B = N/3$	No self-synchronization for long 0s

MODEMS

The term modem is a composite word that refers to the two functional entities that make up the device; a signal modulator and a signal demodulator. A modulator creates a band-pass analog signal from binary data. A demodulator recovers the binary data from the modulated signal.

Modem stands for modulator and demodulator.



TELEPHONE MODEMS

Traditional telephone lines can carry frequencies between 300 and 3300 HZ, giving them BW of 3000 Hz; All this range is used for transmitting voice, where a great deal of interference and distortion can be accepted without loss of intelligibility.

The effective BW of a telephone line being used for data Transmission is 2400 Hz, covering the range from 600 to 3000 Hz.

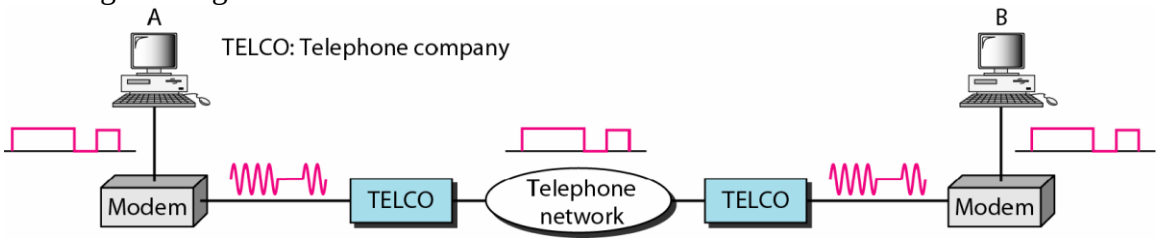


Figure shows the relationship of modems to a communication link. The computer on the left sends binary data to the modulator portion of the modem; the data is sent as an analog signal on the telephone lines. The modem on the right receives the analog signal, demodulates it through its demodulator, and delivers data to the computer on the right.

The communication can be bidirectional, which means the computer on the right can also send data to the computer on the left using the same modulation and demodulation processes.

Modem standards

V-series standards published by the ITU-T.

V.32 V.32bis V.34bis

V.90 V.92

V.32

This modem uses a combined modulation and demodulation encoding technique called trellis-coded modulation. Trellis is essentially QAM plus a redundant bit. The Data stream is divided into 4-bit sections. Instead of a quad bit, however, a pentabit is transmitted. The value of the extra bit is calculated from the values of the data bits.

In any QAM system, the receiver compares each received signal point to all valid points in the constellation and selects the closest point as the intended value. A signal distorted by transmission noise can arrive closer in value to an adjacent point than to the intended point, resulting in a misidentification of the point and an error in the received data.

By adding a redundant bit to each quad bit, trellis-coded modulation increases the amount of information used to identify each bit pattern thereby reduces the number of possible matches.

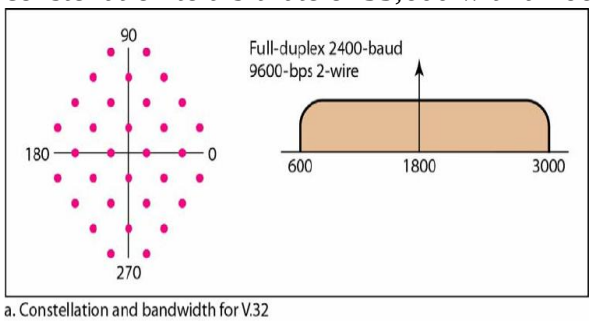
The V.32 calls for 32-QAM with a baud rate of 2400. Because only 4 bits of each pentabit represents data, the resulting speed is 4*2400=9600.

V.32 bis

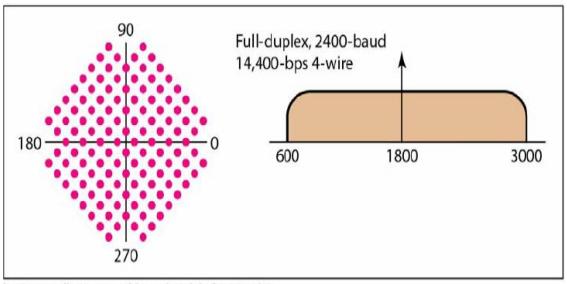
The V.32 bis modem support 14,400-bps transmission. The V.32 uses 128-QAM transmission.

V.34 bis

The V.34 bis modem support 28,800-bps transmission with a 960-point constellation to a bit rate of 33,600 with a 1664-point constellation.



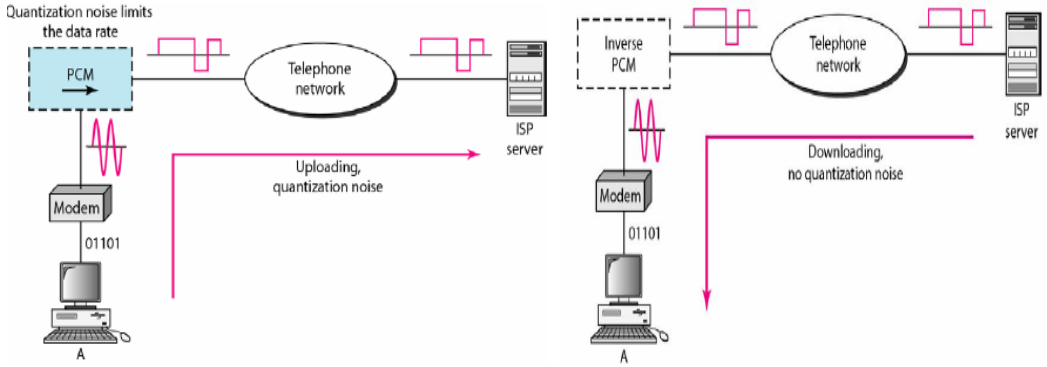
a. Constellation and bandwidth for V.32



b. Constellation and bandwidth for V.32bis

V.90

Traditional modems have a limitations on the data rate.V.90 modems with a bit rate of 56,000 bps, called 56Kmodems, are available. Downloading rate is 56K, while the uploading rate is a maximum of 33.6 kbps.



After modulation by the modem, an analog signal reaches the telephone company switching station. Where it is sampled and digitized to be passed through the digital network. The quantization noise introduced in the signal at the sampling point limits the data rate according to the capacity. This limit is 33.6 Kbps.

V.92

The standard above V.92 is called V.92. These modems can adjust their speed, and if the noise allows, they can upload data at the rate of 48 Kbps. The modem has additional features. For example, the modem can interrupt the internet connection when there is an incoming call if the line has call-waiting service.

RS 232 INTERFACE

- RS 232 is a standard interface by EIA and RS232C is the latest version of this interface.

INTERFACING WITH RS232

- It expects a modem to be connected to both receiving and transmitting end.
- The modem is termed as DCE (Data Communication Equipment) And the computer with which modem is interfaced is called DTE (Data Terminal Equipment).
- The DCE and DTE are linked via a cable whose length does not exceed 50 feet. The DTE has 35 pins male connector and DCE has 25 pins Female connector.

FEATURES OF RS232 INTERFACE

1. **RS232 Signal LEVEL**
   - RS232 standard follows –ve logic, Logic1 is represented by negative voltage. Logic0 is represented by +ve voltage.
   - Level 1 varies from -3 to -15v and level 0 varies from 3 to 15v

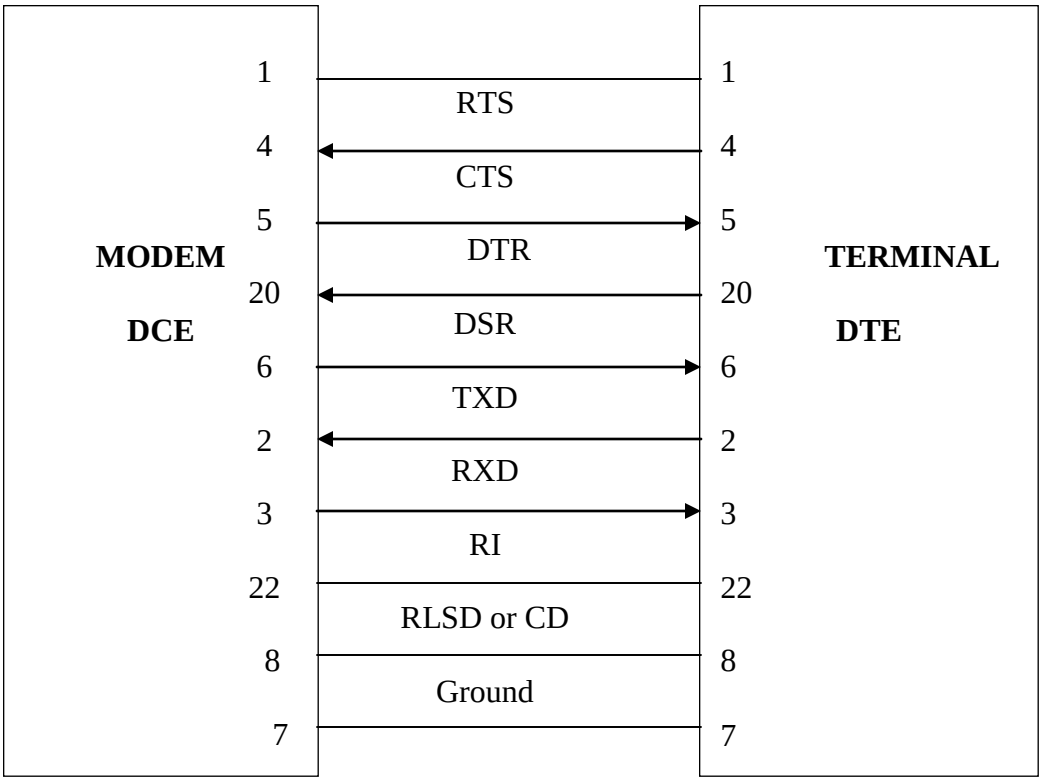
2. **RS232 SIGNALS**

SL NO	PIN NUMBER	SIGNAL	SIGNAL NAME
1	1	---	Frame ground
2	2	TXD	Transmit data
3	3	RXD	Receive data
4	4	RTS	Request to send
5	5	CTS	Clear to send
6	6	DSR	Data Set Ready
7	7	SG	Signal Ground
8	8	RLSD or CD	Received line signal detect or carrier detect
9	20	DTR	Data Terminal Ready
10	22	RI	Ring Indicator

COMMUNICATION BETWEEN DCE AND DTE

- Before sending data to the other end the DTE requests the permission from the modem by issuing RTS signal.
- The modem has a method to find out if any telephone line is free and if the other end of modem is ready.
- When the modem finds the communication path is ready for communication it issues CTS signal to DTE as an acknowledgement.
- The DTE issues DTR signal when it is powered on, error free and ready for logical connection through the modem.
- The modem issues a DSR signal to indicate that it is powered on and it is error free.
- The data is transferred by TXD signal from DTE to DCE and RXD signal receives data from DCE to DTE.
- The RI and RLSD signals are used with the dialed modem, when the telephone link is shared.

Communication



25 pin female connector **25 pin male connector**

UNIT II

DATA LINK LAYER

ERROR DETECTION AND CORRECTION

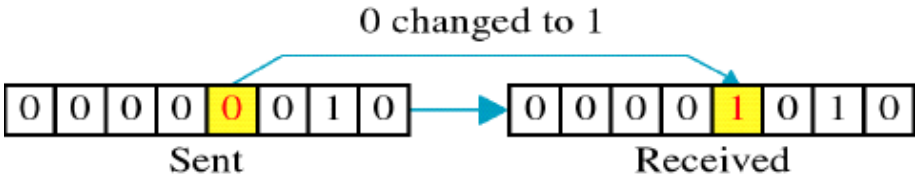
ERROR:

Data can be corrupted during transmission. For reliable communication, errors must be detected and corrected. Signals flows from one point to another. This is subjected to unpredictable interferences from heat, magnetism and other forms of electricity.

TYPES OF ERRORS:

- **Single bit Error:**

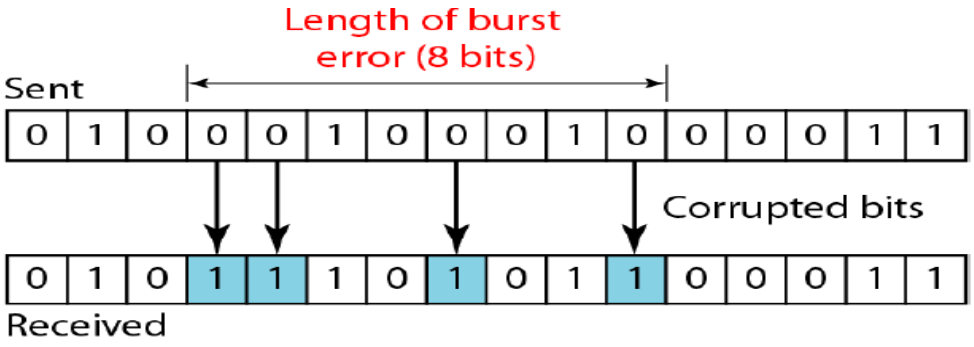
  The term single bit error means that only one bit of a given data unit is changed from 1 to 0 or 0 to 1. 010101 is changed to 110101 here only one bit is changed by single bit error.



- **Burst Error:**

  A burst error means that 2 or more bits in the data unit have changed.

  Example:



Three kinds of errors can occur:

- the bits in the frame can be inverted, anywhere within the frame including the data bits or the frame's control bits,
- additional bits can be inserted into the frame, before the frame or after the frame and
- Bits can be deleted from the frame.

DETECTION

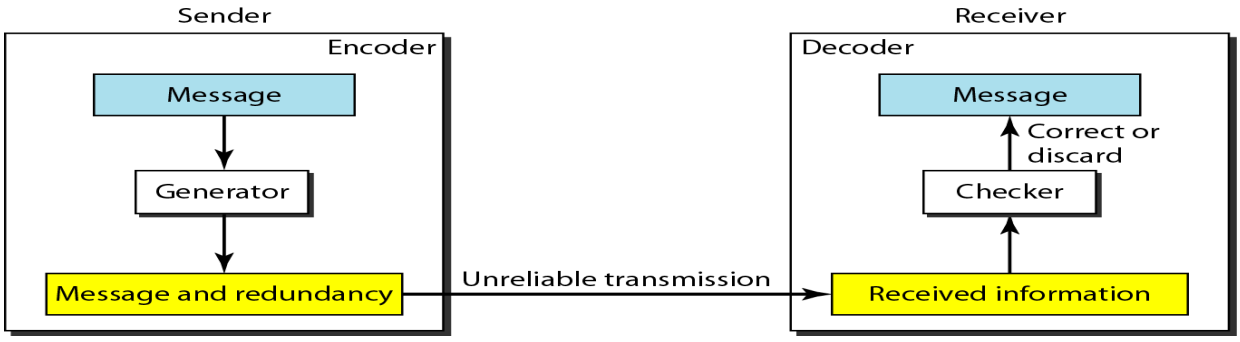
Redundancy

Error detection use the concept of redundancy, which means adding extra bits for detecting errors at the destination .i.e., instead of repeating the entire data stream, a shorter group of bits may be appended to the end of each unit.

- To detect or correct errors, we need to send extra (redundant) bits with data.
- The receiver will be able to detect or correct the error using the extra information.
- Detection
  - Looking at the existence of any error, as YES or NO.
  - Retransmission if yes. (ARQ)
- Correction
  - Looking at both the number of errors and the location of the errors in a message.
  - Forward error correction. (FEC)

Coding

- Encoder vs. decoder
- Both encoder and decoder have agreed on a detection/correct method in priori.

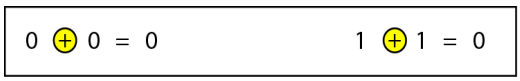


Modulo Arithmetic

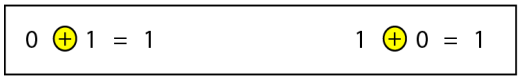
- In modulo-*N* arithmetic, we use only the integers in the range 0 to *N*−1, inclusive.
- Calculation
  - If a number is greater than *N*−1, it is divided by *N* and the remainder is the result.
  - If it is negative, as many N's as needed are added to make it positive.
- Example in Modulo-12
  - $15_{12} = 3_{12}$
  - $-3_{12} = 9_{12}$

Modulo-2 Arithmetic

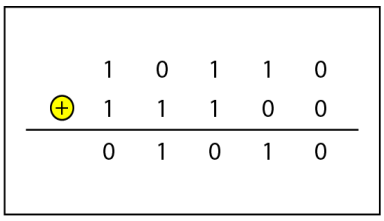
- Possible numbers are {0, 1}
- Arithmetic
  - Addition
    - 0+0=0, 0+1=1, 1+0= 1, 1+1=2=0
  - Subtraction
    - 0-0=0, 0-1=-1=1, 1-0=1, 1-1=0
  - Surprisingly, the addition and subtraction give the same result.
  - XOR (exclusively OR) can replace both addition and subtraction.



a. Two bits are the same, the result is 0.



b. Two bits are different, the result is 1.



c. Result of XORing two patterns

Detection methods

- **Parity check**
- **Cyclic redundancy check**
- **checksum**

Parity check

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

SIMPLE PARITY CHECK

- A simple parity-check code is a single-bit error-detecting code in which $n = k + 1$ with $d_{min} = 2$.
- A simple parity-check code can detect an odd number of errors.

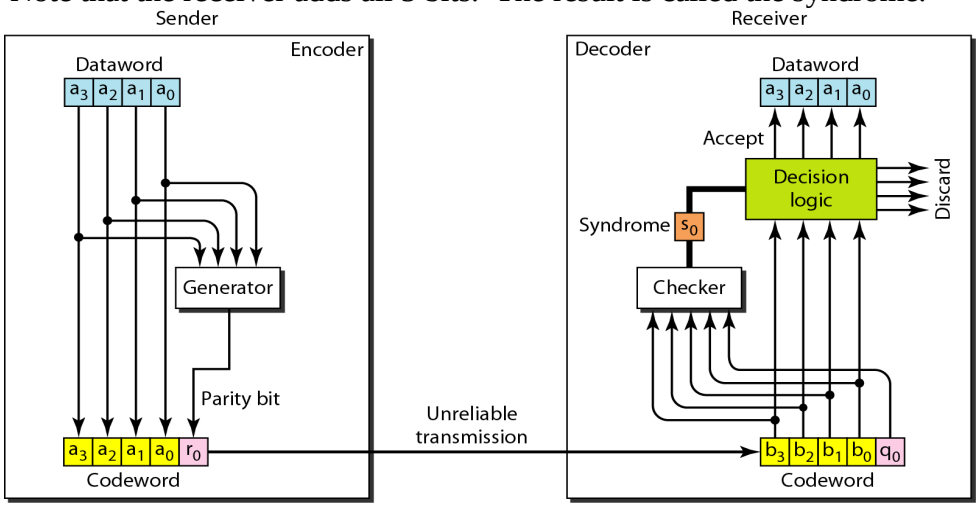
In a simple parity check a redundant bit is added to a string of data so that total number of 1's in the data become even or odd.

The total data bit is then passed through parity checking function. For even parity, it checks for even number of 1's and for odd parity it checks even number of 1's. If an error is detected the data is rejected.

Datawords	*Codewords*	*Datawords*	*Codewords*
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111
0100	01001	1100	11000
0101	01010	1101	11011
0110	01100	1110	11101
0111	01111	1111	11110

Encoder and decoder for simple parity-check code

- In modulo,
  - $r_0 = a_3+a_2+a_1+a_0$
  - $s_0 = b_3+b_2+b_1+b_0+q_0$
- Note that the receiver adds all 5 bits. The result is called the syndrome.



Example 1: data to be transmitted = 10110101

- 5 1's in the data
- Parity bit is 1
- Transmitted codeword = 101101011
- If receiver gets 101101011, parity check ok ---accept (OK)
- If receiver gets 101100011, parity check fails ---reject (OK), ask for frame to be re-transmitted

- If receiver gets 101110011, parity check ok ---accept (NOT OK: even number of errors undetected)
- If receiver gets 001100011, parity check ok ---accept (NOT OK: even number of errors undetected)

Let us look at some transmission scenarios. Assume the sender sends the dataword 1011. The codeword created from this dataword is 10111, which is sent to the receiver. We examine five cases:

1. No error occurs; the received codeword is 10111. The syndrome is 0. The dataword 1011 is created.
2. One single-bit error changes a_1. The received codeword is 10011. The syndrome is 1. No dataword is created.
3. One single-bit error changes r_0. The received codeword is 10110. The syndrome is 1. No dataword is created.
4. An error changes r_0 and a second error changes a_3 . The received codeword is 00110. The syndrome is 0. The dataword 0011 is created at the receiver. Note that here the dataword is wrongly created due to the syndrome value.
5. Three bits—a_3, a_2, and a_1—are changed by errors. The received codeword is 01011. The syndrome is 1. The dataword is not created. This shows that the simple parity check, guaranteed to detect one single error, can also find any odd number of errors.

CYCLIC REDUNDANCY CHECK

CRC is based on binary division. In CRC, instead of adding bits to achieve the desired parity, a sequence of redundant bits, called the CRC or the CRC remainder, is appended to the end of the data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number. At its destination, the incoming data unit is assumed to be intact and is therefore accepted. A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

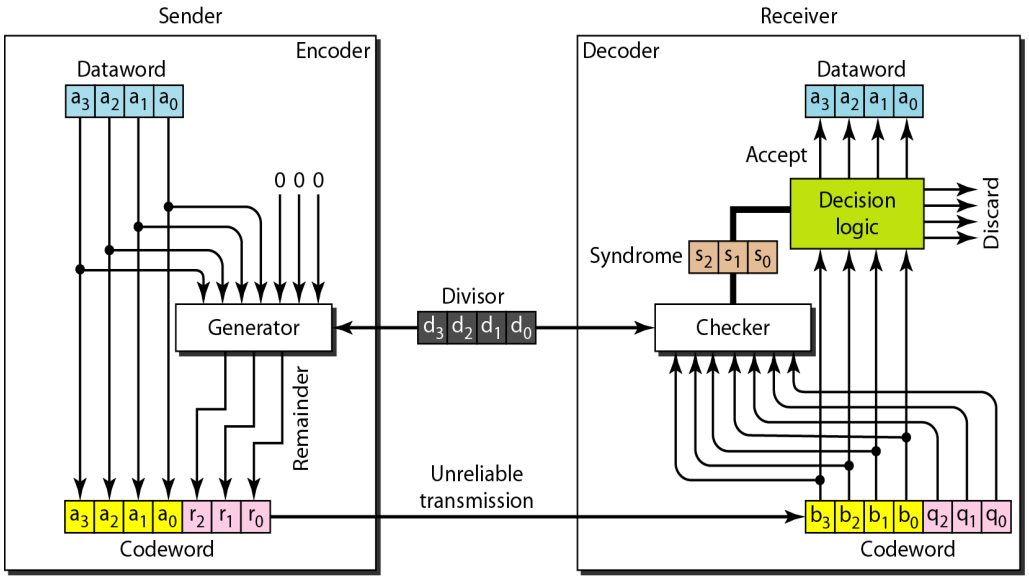
STEP BY STEP PROCEDURE

- Dividing the data unit by a predetermined divisor derives the redundancy bits used by CRC; the remainder is CRC.
- First a starting of n 0's is appended to the data unit. The number n is one less than the number of bits in the predetermined divisor, which is n+1 bits.
- The newly elongated data unit is divided by the divisor, using a process called binary division. The remainder resulting from this division is the CRC.
- The CRC of n bits derived in step 2 replaces the appended 0s at the end of the data unit. Note that the CRC may consist of all 0s.
- The data unit arrives at the receiver data first, followed by the CRC. The receiver treats the whole string as unit and divides it by the same divisor that was used to find the CRC remainder.

- If the string arrives without error, the CRC checker yields a remainder of zero ad the data unit passes. If the string has been changed in transit, the division yields a non zero remainder and the data does not pass.

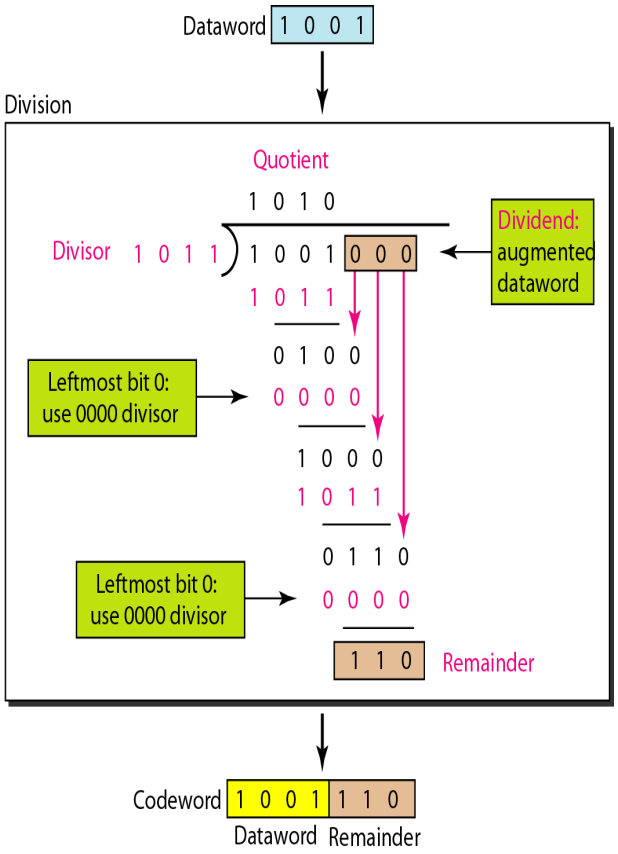
Dataword	*Codeword*	*Dataword*	*Codeword*
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

Architecture of CRC

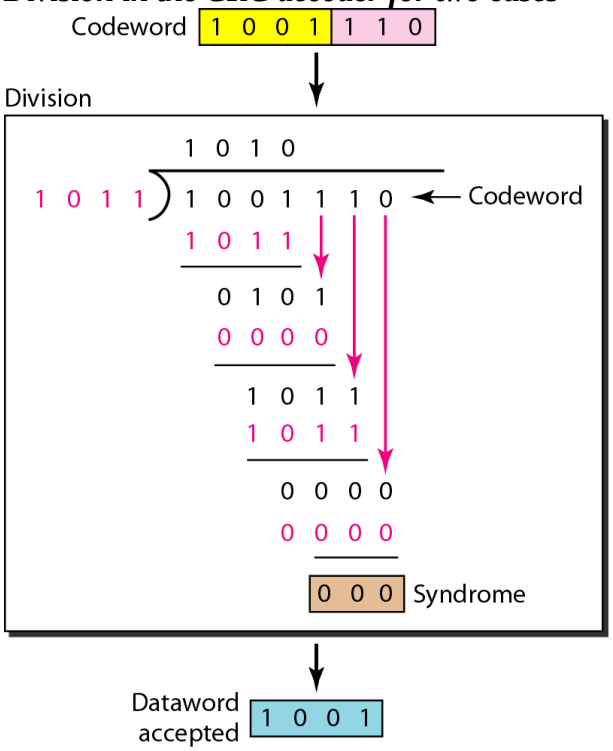


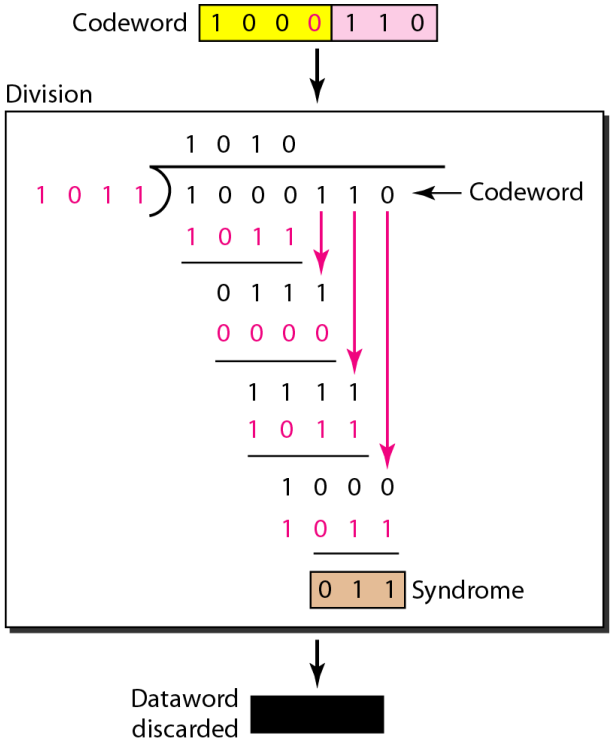
CRC GENERATOR AND CHECKER

CRC GENERATOR

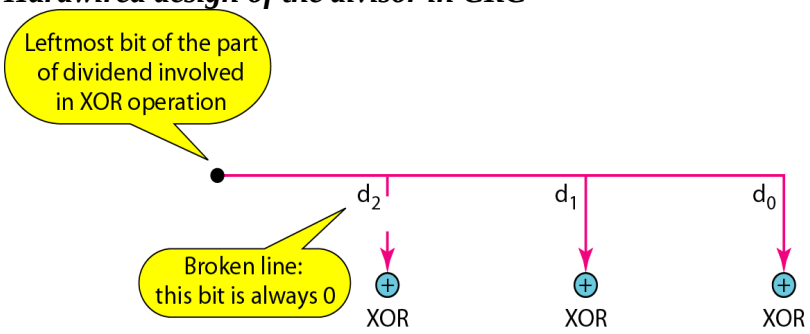


Division in the CRC decoder for two cases

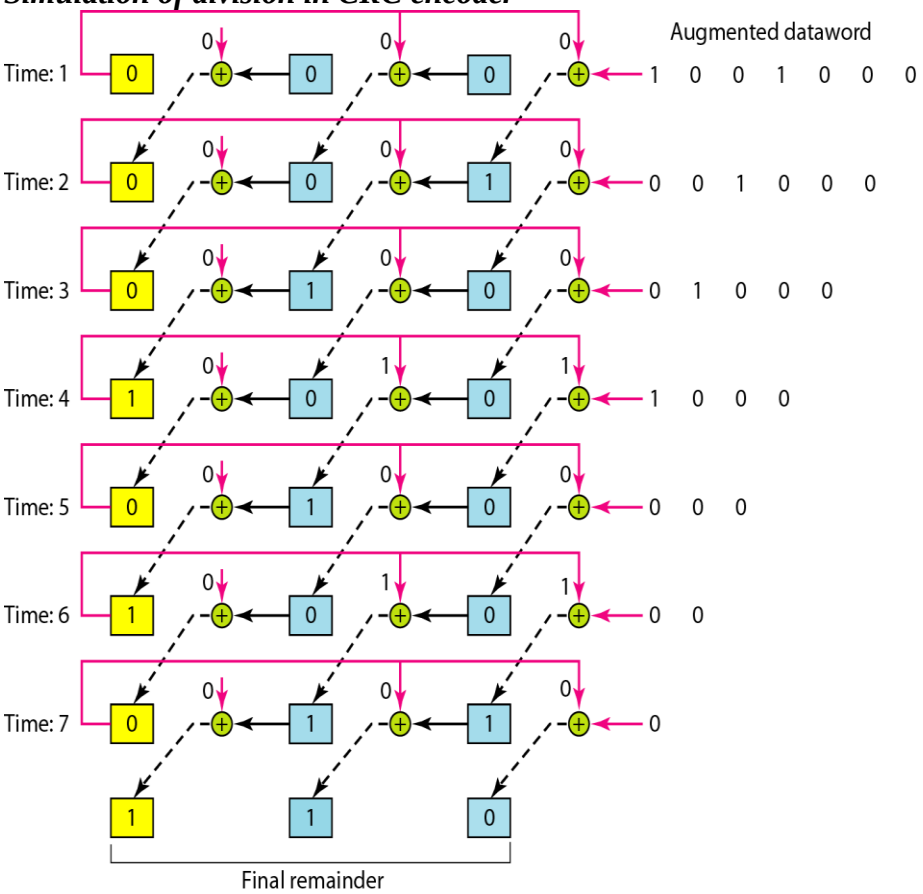




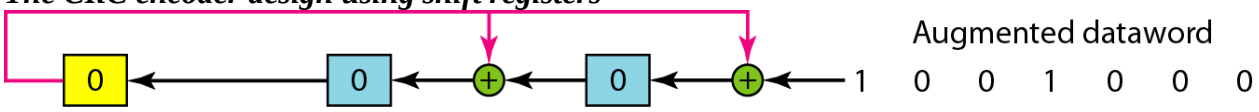
Hardwired design of the divisor in CRC



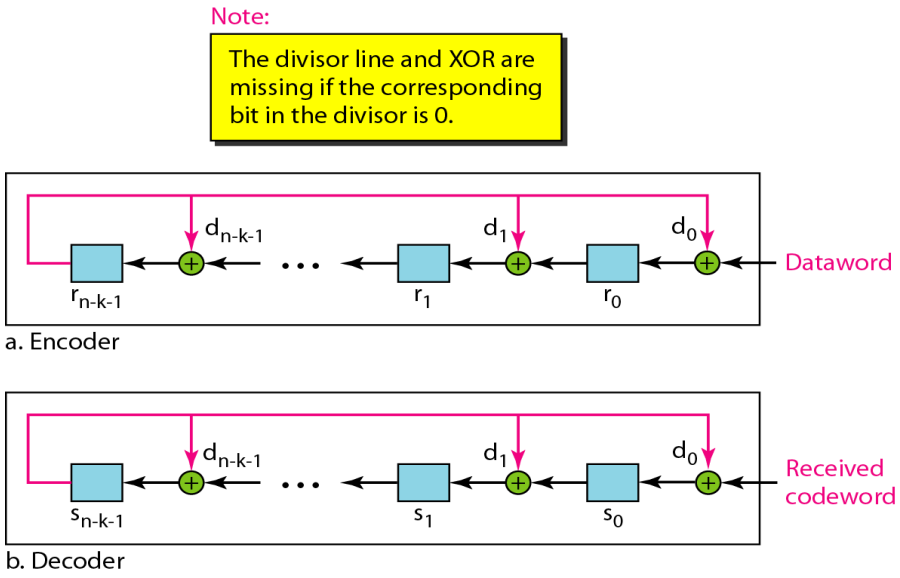
Simulation of division in CRC encoder



The CRC encoder design using shift registers



General design of encoder and decoder of a CRC code



HAMMING CODE:

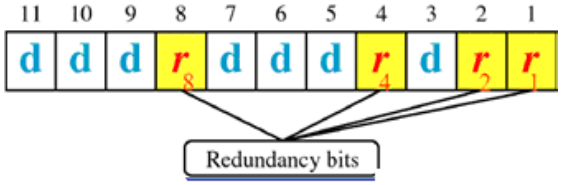
- A ***minimum number of redundancy bits*** needed to correct any single bit error in the data

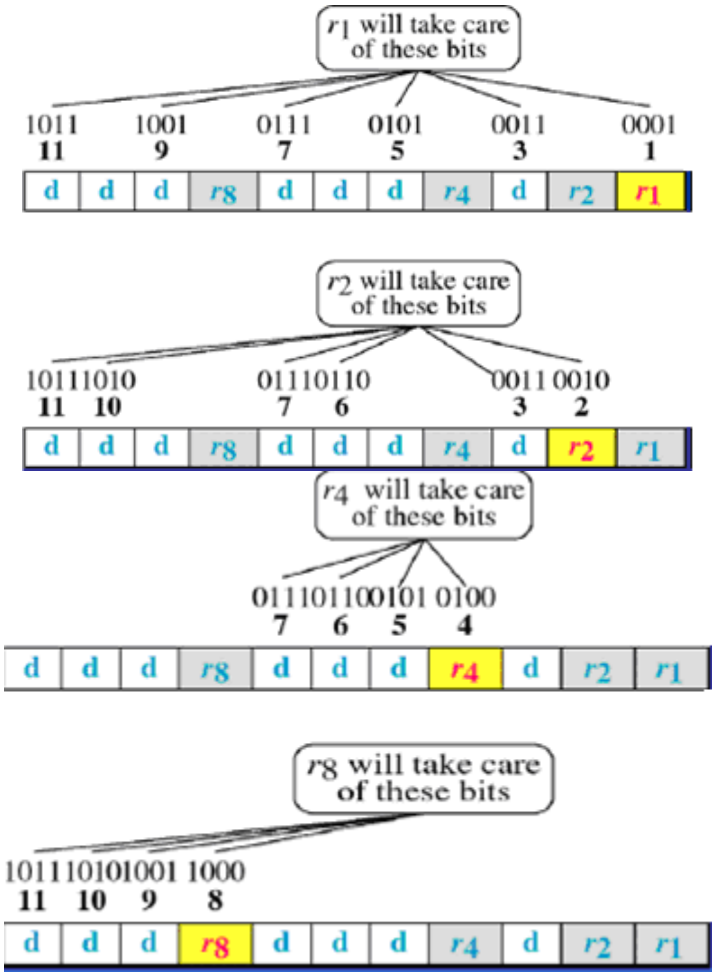
- A minimum of 4 redundancy bits is needed if the number of data bits is 4.

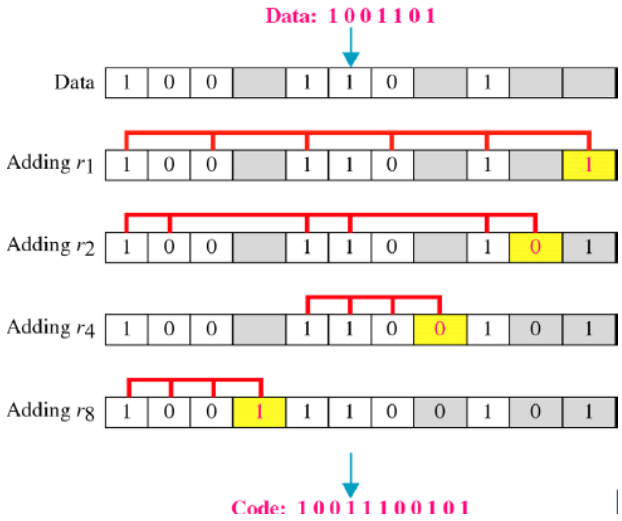
- Redundancy bits in the Hamming code are placed in the codeword bit positions that are a power of 2

- Each redundancy bit is the parity bit for a different combination of data bits

- Each data bit may be included in more than one parity check.




r1 will take care of these bits
1011 1001 0111 0101 0011 0001
11 9 7 5 3 1
d d d r8 d d d r4 d r2 r1
r2 will take care of these bits
10111010 01110110 0011 0010
11 10 7 6 3 2
d d d r8 d d d r4 d r2 r1
r4 will take care of these bits
0111011000101 0100
7 6 5 4
d d d r8 d d d r4 d r2 r1
r8 will take care of these bits
101110101001 1000
11 10 9 8
d d d r8 d d d r4 d r2 r1


Data: 1 0 0 1 1 0 1
Data 1 0 0 1 1 0 1
Adding r1 1 0 0 1 1 0 1 1
Adding r2 1 0 0 1 1 0 1 0 1
Adding r4 1 0 0 1 1 0 0 1 0 1
Adding r8 1 0 0 1 1 1 0 0 1 0 1
Code: 1 0 0 1 1 1 0 0 1 0 1

- Easy way to compute the redundancy bit values: write down binary representations for positions of data bits which contain a 1; compute parity bits for each "column"; put parity bits into codeword in correct order.

- Here: data is 1001101 so codeword will look like 100x110x1xx (where x denotes redundancy bits) ⇒1's in positions 3, 6, 7, and 11

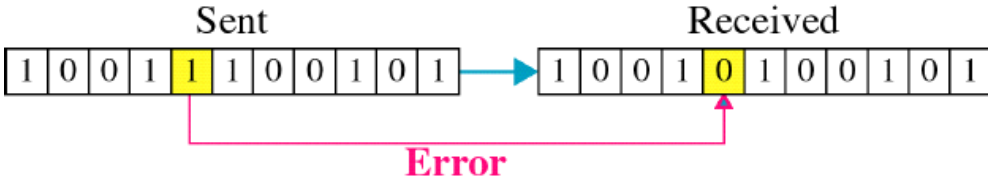
11: 1 0 1 1
7: 0 1 1 1
6: 0 1 1 0
3: 0 0 1 1

1 0 0 1

parity bits

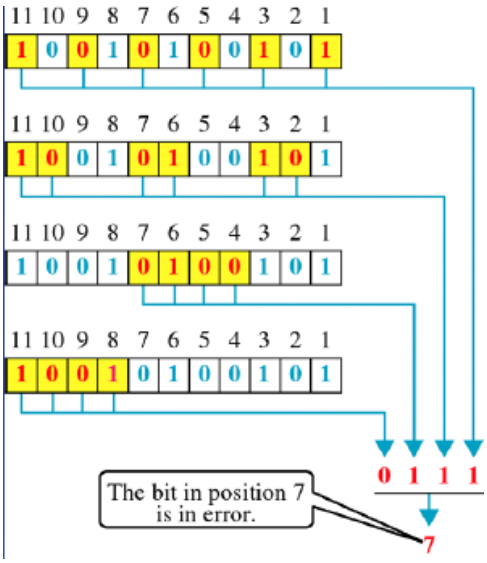
r1 r2 r4 r8

So codeword is 100*1*1100*1*0*1*(as before)
suppose that the bit in position 7 is received in error:



1 • If the transmitted codeword is received error-free, the "new" parity bits the receiver computes will all be 0 ,the receiver knows no bit errors occurred.

1 • This simple form of Hamming code can be used to provide some protection against burst errors, by transmitting 1st bit from every codeword to be transmitted, then 2nd bit from every one of these codeword, and so on…In some cases, burst errors can be corrected



1

FLOW CONTROL AND ERROR CONTROL

The two main features of data kink layer are flow control and error control.

FLOW CONTROL

Flow control coordinates that amount of data that can be sent before receiving ACK It is one of the most important duties of the data link layer.

ERROR CONTROL

- Error control in the data link layer is based on ARQ (automatic repeat request), which is the retransmission of data.
- The term error control refers to methods of error detection and retransmission.
- Anytime an error is detected in an exchange, specified frames are retransmitted. This process is called ARQ.

FLOW AND ERROR CONTROL MECHANISMS

1. STOP-AND WAIT ARQ.
2. GO-BACK-N ARQ.
3. SELECTIVE-REPEAT ARQ.

STOP-AND- WAIT ARQ

This is the simplest flow and error control mechanism. It has the following features.

- The sending devise keeps the copy of the last frame transmitted until it receives an acknowledgement for that frame. Keeping a copy allows the sender to re-transmit lost or damaged frames until they are received correctly.
- Both data and acknowledgement frames are numbered alternately 0 and 1. A data frame 0 is acknowledged by an ACK 1.
- A damaged or lost frame is treated in the same manner by the receiver. If the receiver detects an error in the received frame, it simply discards the frame and sends no acknowledgement.
- The sender has a control variable, which we call S, that holds the number of recently sent frame. The receiver has a control variable, which we call R that holds the number of the next frame expected.
- The sender starts a timer when it sends a frame. If an ACK is not received within an allotted time period the sender assumes that the frame was lost or damaged and resends it.
- The receivers send only positive ACK for frames received safe and sound; it is silent about the frames damaged or lost.

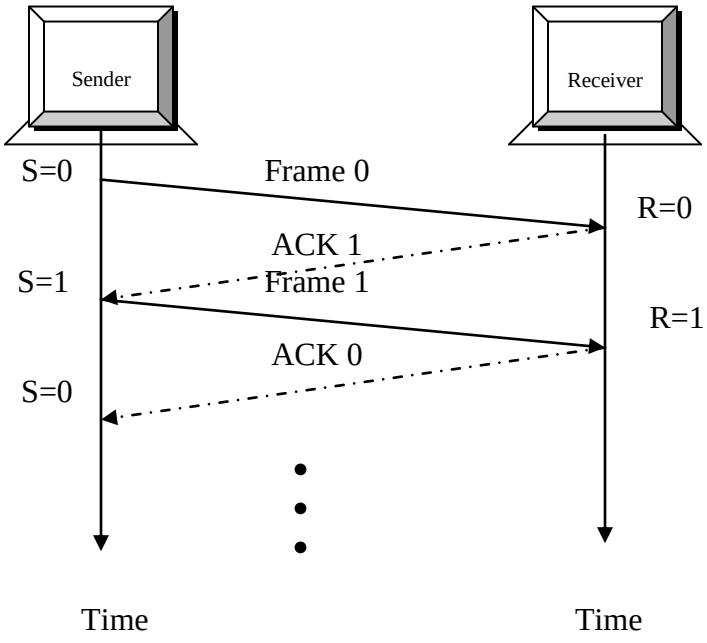
OPERATION:

The possible operations are

Normal operation
lost frame
ACK lost
delayed ACK.

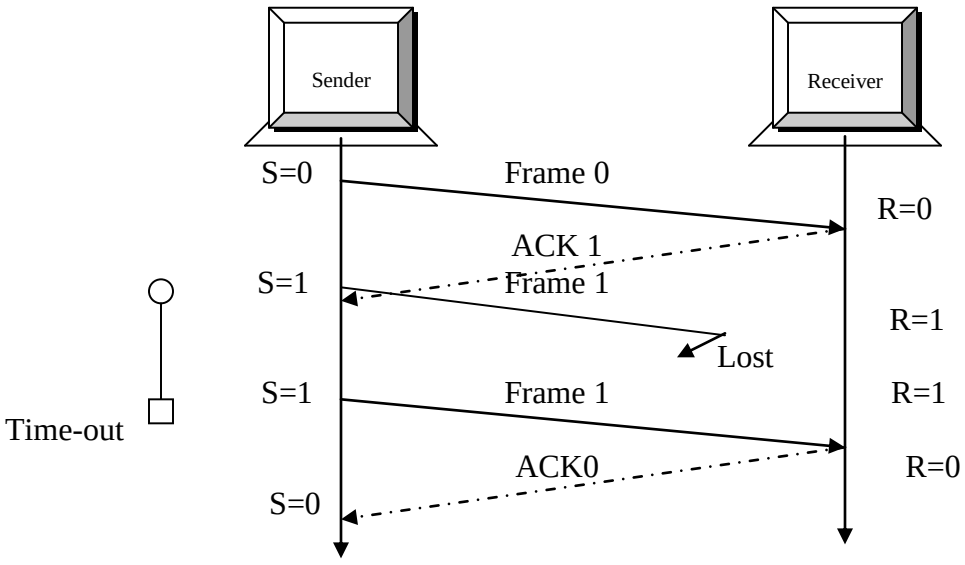
The sender sends frame 0 and wait to receive ACK 1. when ACK 1 is received it sends frame 1 and then waits to receive ACK 0, and so on.

The ACK must be received before the time out that is set expires. The following figure shows successful frame transmission.



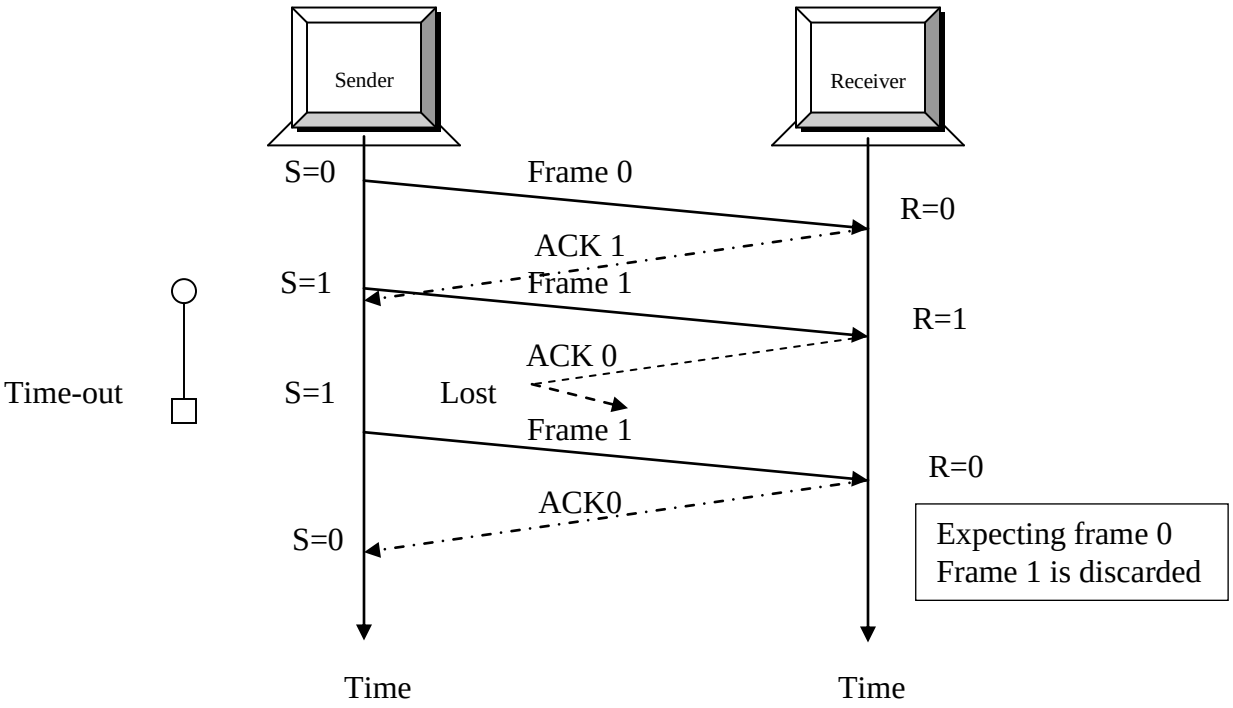
Lost or damaged acknowledgement

- When the receiver receives the damaged frame it discards it, which essentially means the frame is lost. The receiver remains silent about a lost frame and keeps its value of R.
- For example in the following figure the sender transmits frame 1, but it is lost. The receiver does nothing, retaining the value of R (1). After the timer at the sender site expires, another copy of frame 1 is sent.

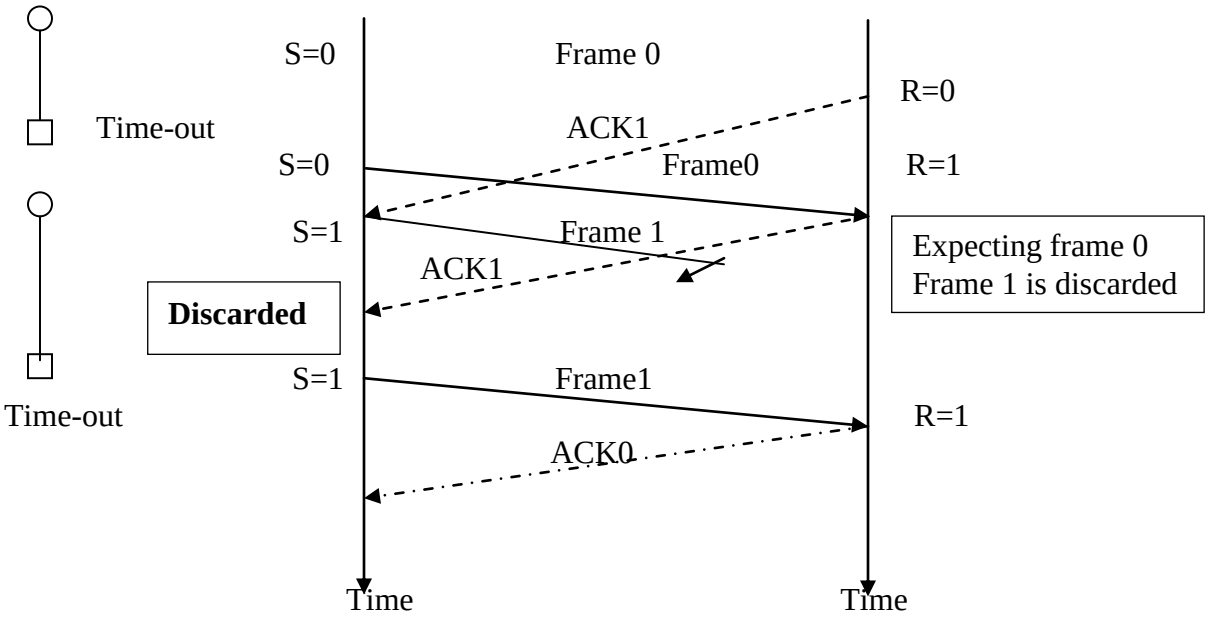


Lost acknowledgement

- A lost or damaged ACK is handle in the same by the sender; if the sender receives a damaged ACK, it discards it.
- The following figure shows a lost ACK 0.the waiting sender does not know if frame 1 has been received. When the timer for frame 1 expires the sender retransmits frame 1.
- Note that the receiver has already received frame 1 and is expecting to receive frame 0. Therefore, its silently discards the second copy of frame 1.



- **Delayed acknowledgement**
  - An ACK can be delayed at the receiver or by some problem with the link. The following figure shows the delay of ACK 1; it ids received after the timer for frame 0 as already expired.
  - The sender has already retransmitted a copy of frame 0. The receiver expects frame 1 so its simply discards the duplicate frame 0.
  - The sender has now received two ACK's, one that was delayed and one that was sent after the duplicate frame 0 arrived. The second ACK 1 is discarded.

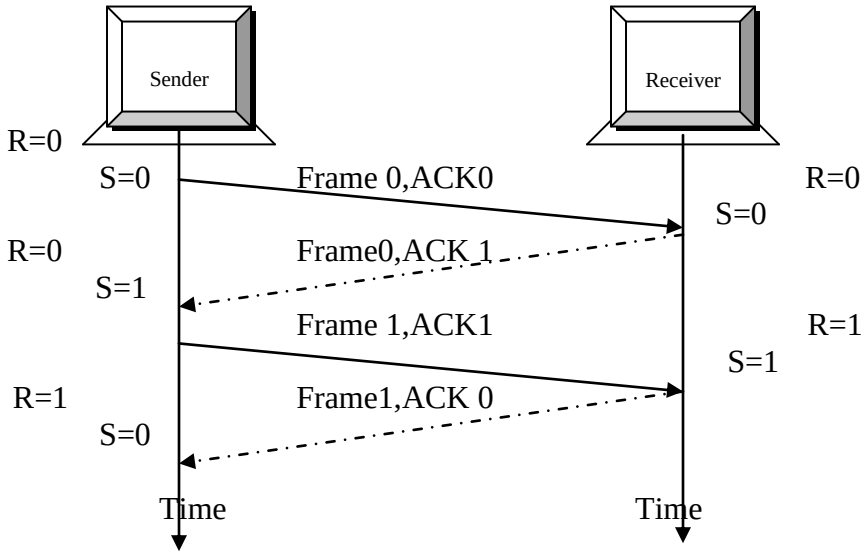


BIDIRECTIONAL TRANSMISSION

The stop – and – wait mechanism is unidirectional. We can have bi-directional transmission if the two parties have two separate channels for full duplex communication or share the same channel for off duplex transmission. In this case, each party needs both S and R variables to track frames sent and expected.

PIGGYBACKING

It's a method to combine a data frame with an ACK. In following figure both the sender and the receiver have data to send. Instead of sending separate data and ACK frames. It can save bandwidth because the overhead from a data frame and an ACK frame can be combined into just one frame



GO-BACK-N ARQ

- As in Stop-and-wait protocol senders has to wait for every ACK then next frame is transmitted. But in GO-BACK-N ARQ number of frames can be transmitted without waiting for ACK. A copy of each transmitted frame is maintained until the respective ACK is received.

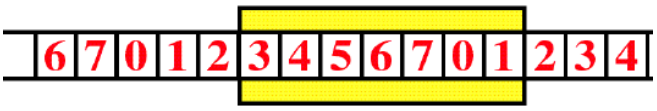
Features of GO-BACK-N ARQ

1.**sequence numbers.**

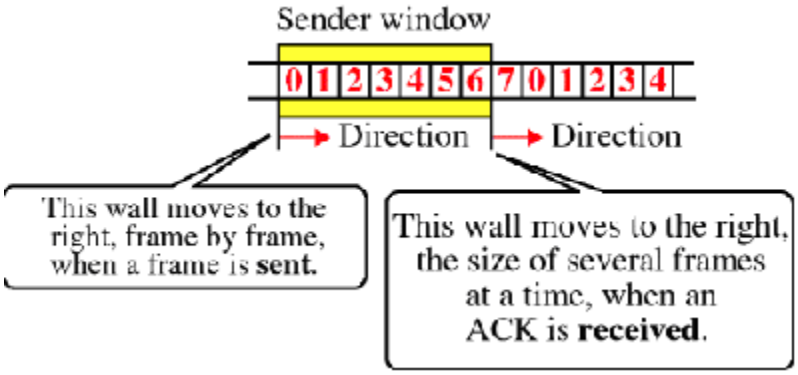
Sequence numbers of transmitted frames are maintained in the header of frame. If k is the number of bits for sequence number, then the numbering can range from 0 to 2k-1. Example: if k=3 means sequence numbers are 0 to 7.

2. sender sliding window:

- Window is a set of frames in a buffer waiting for ACK. This window keeps on sliding in forward direction, the window size is fixed. As the ACK is received, the respective frame goes out of window and new frame to sent come into window. Figure illustrates the sliding window.
- If Sender receives. ACK 4, then it ***knows Frames upto*** and including Frame 3 were ***correctly received***

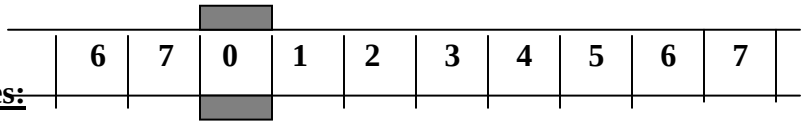


Window size=7



3. Receiver sliding window:

In the receiver side size of the window is always one. The receiver is expecting to arrive frame in specifies sequence. Any other frame is received which is out of order is discarded. The receiver slides over after receiving the expected frame. The following figure shows the receiver side-sliding window.



4. Control variables:

Sender variables and Receiver variables:
Sender deals with three different variables
S -> sequence number of recently sent frame
S_F -> sequence number of first frame in the window.
S_L -> sequence number of last frame in the window.
The receiver deals with only one variable
R -> sequence number of frame expected.

5. Timers

The sender has a timer for each transmitted frame. The receivers don't have any timer.

6. Acknowledgement:

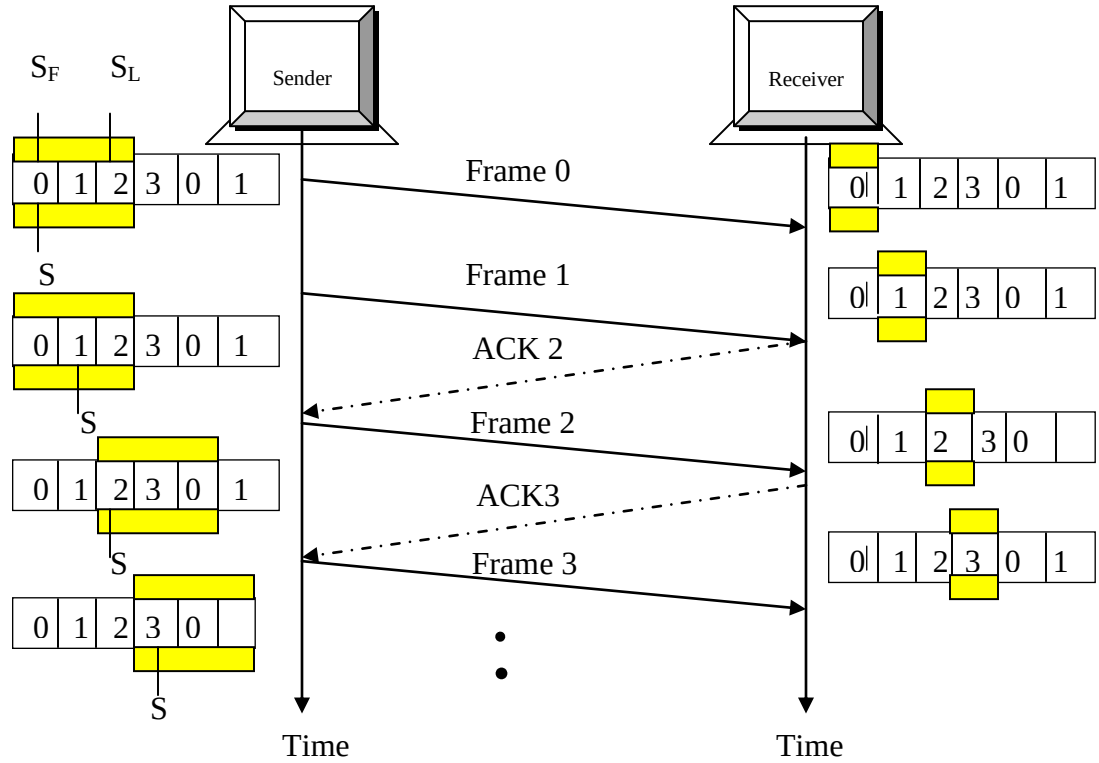
The receiver responds for frame arriving safely by positive ACK. For damaged or lost frames receiver doesn't reply, the sender has to retransmit it when timer of that frame elapsed. The receiver may ACK once for several frames.

7. resending frames:

if the timer for any frame expires, the sender sender has to resend that frame and the subsequent frame also, hence the protocol is called GO-BACK-N ARQ.

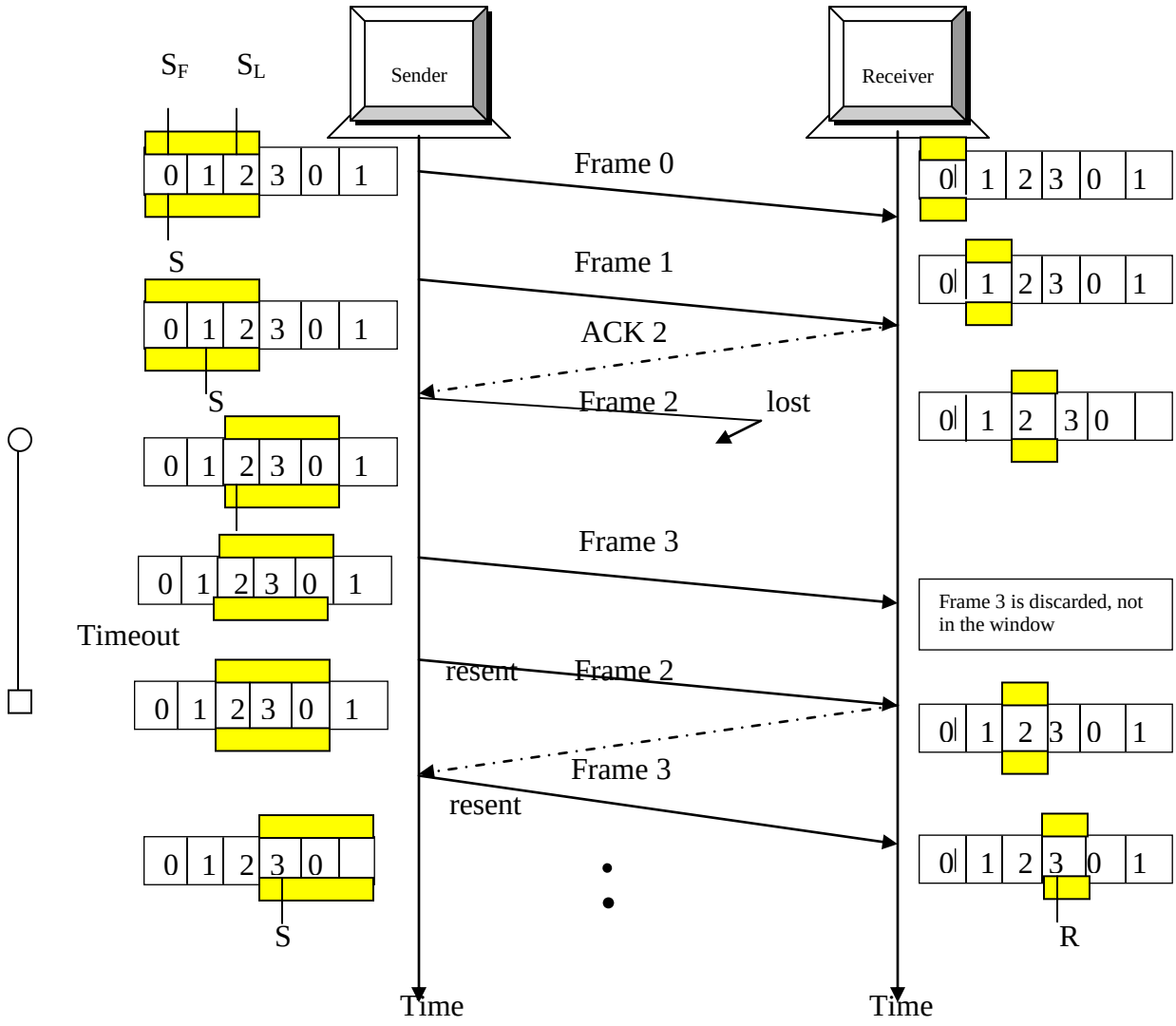
Operation

Normal operation: Following diagram shows this mechanism. The sender keeps track of the outstanding frames and updates the variables and windows as acknowledgements arrive.



Damaged or lost frame:

Figure shows that frame 2 is lost. Note that when the receiver receives frame 3, it is discarded because the receiver is expecting frame 2, not frame3. after the timer for frame 2 expires at the sender site, the sender sends frame 2 and 3.



Damaged or lost acknowledgement:

If an ACK is lost, we can have two situations. If the next ACK arrives before the expiration of timer, there is no need for retransmission of frames because ACK are cumulative in this protocol.. if the next ACK arrives after the timeout, the frame and all the frames after that are resent. The receiver never resends an ACK.
For diagrams refer your class work notes.

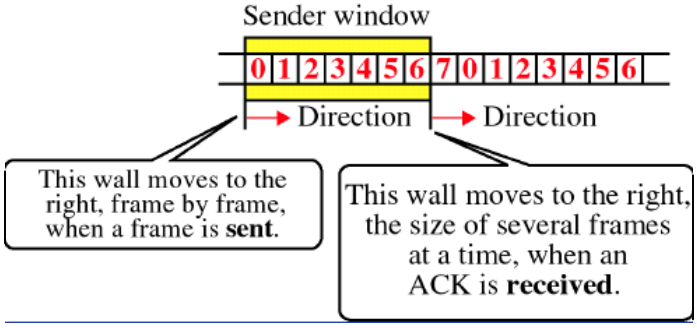
Delayed Acknowledgement:

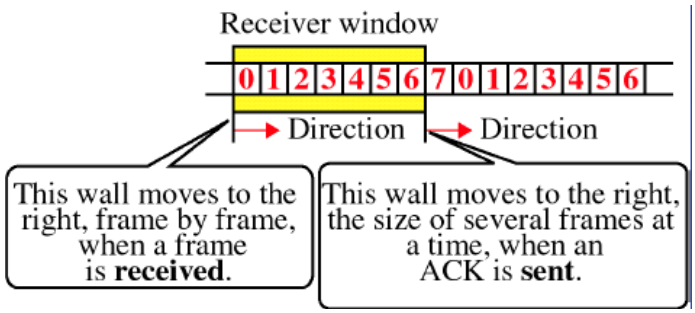
A delayed ACK also triggers the resending of frames.

SELECTIVE REPEAT ARQ:

- The configuration and its control variables for this are same as those selective repeat ARQ.

- The size of the window should be one half of the value 2^m.
- The receiver window size must also be the size. In this the receiver is looking for a range of sequence numbers.
- The receiver has control variables R_F and R_L to denote the boundaries of the window.





selective repeat also defines a negative ACK NAK that reports the sequence number of a damaged frame before the timer expires.

Operation

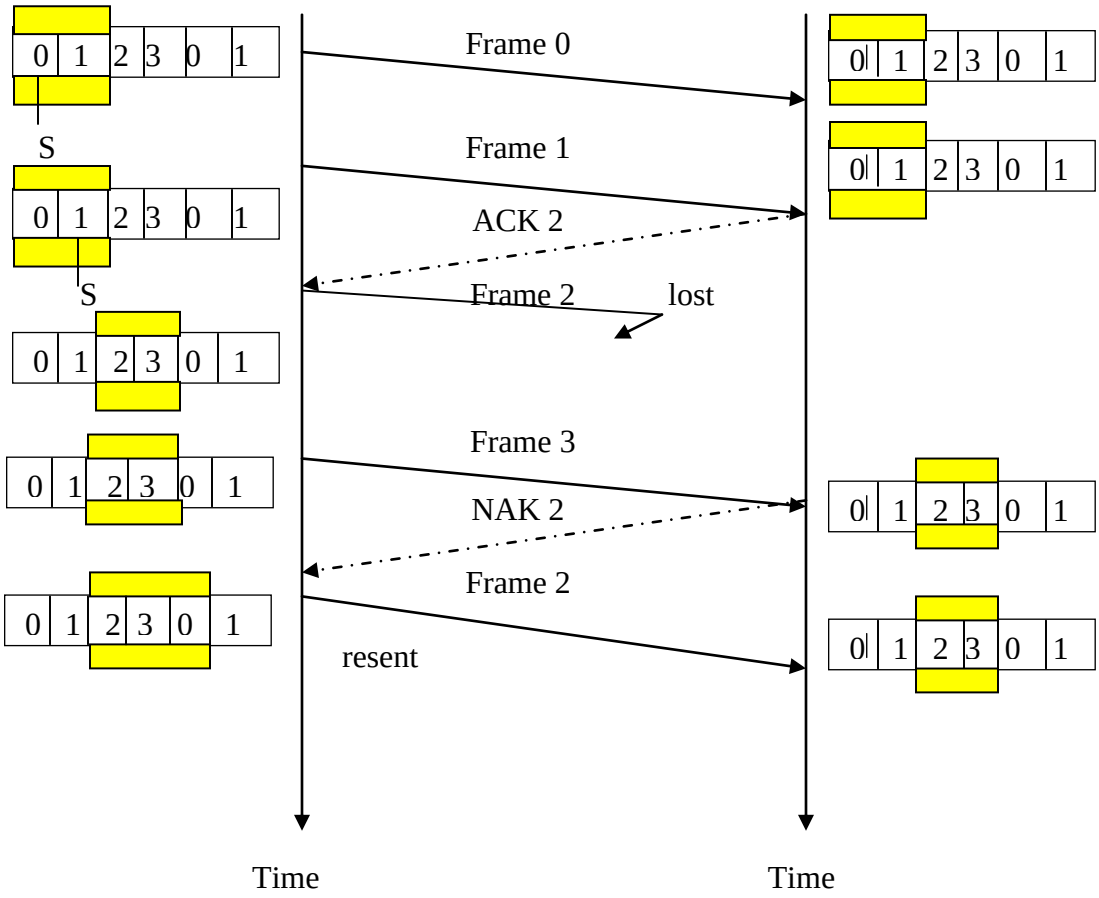
Normal operation

Normal operations of the selective repeat ARQ is same as GO-BACK-N ARQ mechanism.

Lost or damaged frame

The following figure shows operation of the mechanism with an example of a lost frame**.**

Frame 0 and 1 are accepted when received because they are in the range specified by the receiver window. When frame 3 is received, it is also accepted for the same reason. However the receiver sends a NAK 2 to show that frame 2 has not been received. When the sender receives the NAK 2, it resends only frame 2, which is then accepted because it is in the range of the window.



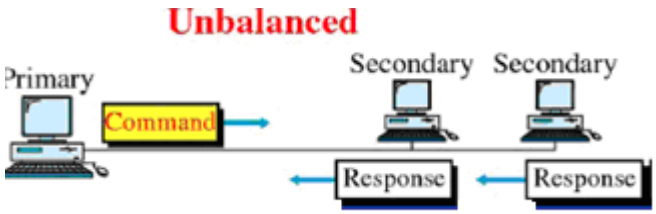
Lost and delayed ACKs and NAKs

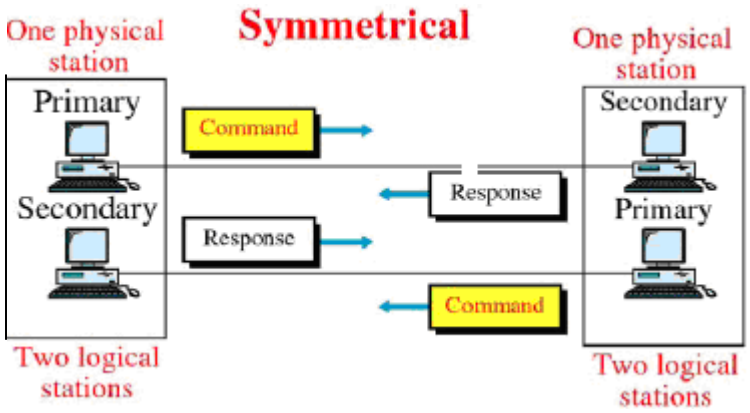
In this sender also sets a timer for each frame sent. The remaining operations are same as GO-BACK-N ARQ.

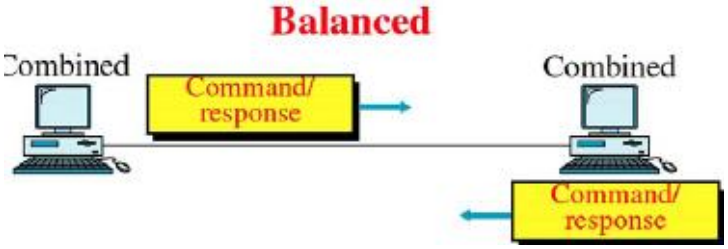
High-level Data Link Control (HDLC) protocol

- HDLC standardized ISO in 1979 and accepted by most other standards bodies (ITU-T, ANSI)
- 3 types of end-stations:
  - *Primary*–sends commands
  - *Secondary*–can only respond to Primary's commands
  - *Combined*–can both command and respond
- 3 types of configuration

(Note: no balanced multipoint)







TRANSFER MODE

- Mode = relationship between 2 communicating devices;
- Describes who controls the link
  - NRM = Normal Response Mode
  - ABM = Asynchronous Balanced Mode

NRM:

only difference is that secondary needs permission from the Primary in NRM, but doesn't need permission from the Primary in ARM.

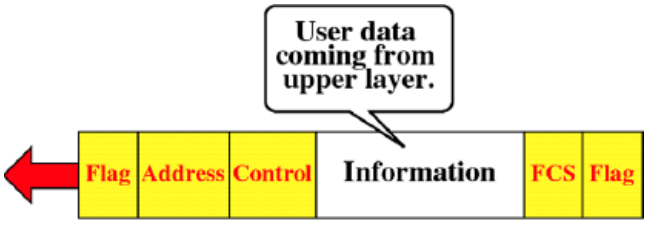
FRAMES:

3 types of Frames are

1 ***I-Frame* –** transports user data and control info about user data**.**

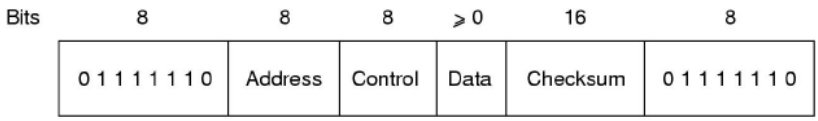
1 ***S-Frame* –** supervisory Frame, only used for transporting control information

1 ***U-Frame*** **–** unnumbered Frame, reserved for system management(managing the link itself)

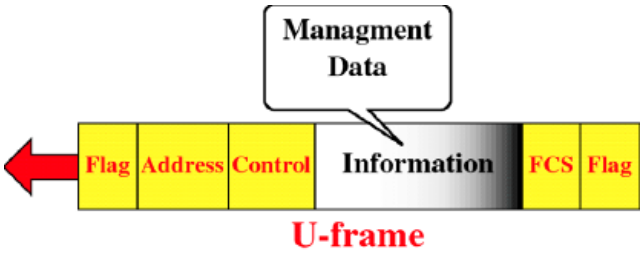


2 **I-frame**

FRAME FORMAT



U-Frames:



U-frame

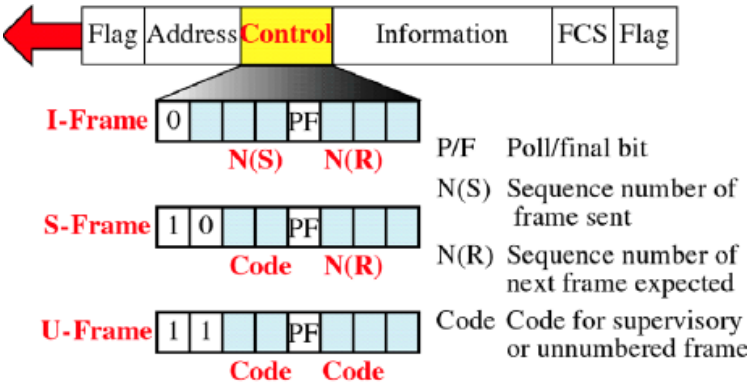
- U-frames are used for functions such as link setup. They do not contain any sequence numbers.
- Five code bits denote the frame type (but there are not 32 different possibilities):
- Set Asynchronous Balanced Mode (SABM).Used in the link set up to indicate ABM mode will be used.
- Set Normal Response Mode (SNRM).Used for asymmetric mode (master/slave).
- SABME and SNMRE—extended format.
- Disconnect (DISC).Used to disconnect the logical connection.
- Frame Reject (FRMR)—reject frame with incorrect semantics.
- Unnumbered Acknowledgement (UA).Used to acknowledge other frames in this class.
- Unnumbered Information (UI)–initialisation, poling and status information needed by the data link layer.
- U-frames may carry data when unreliable connectionless service is called for.

S-Frames:

- S-frames are similar to unnumbered frames, the main difference being that they do carry sequence information.
- Some supervisory frames function as positive and negative acknowledgements, they therefore play a very important role in error and flow control.
- Two bits indicate the frame type, so that there are four possibilities.

    - Receiver Ready -RR(Positive Acknowledgement)
    - Receiver Not Ready -RNR
    - Reject -REJ(NAK go-back-N)
    - Selective Reject -SREJ(NAK selective retransmit)

Control Field:



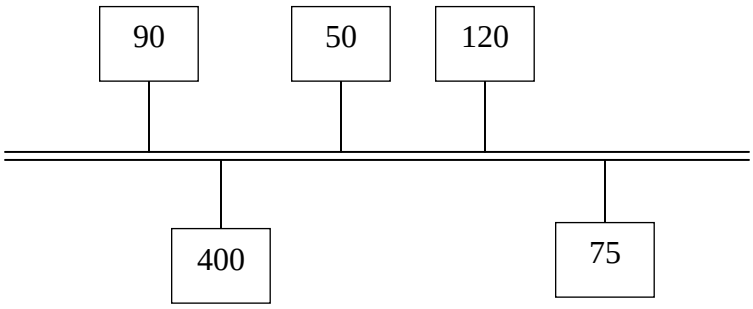
IEEE 802.4 TOKEN BUS

- IEEE 802.4 describes a token bus LAN standard.
- In token passing method stations, connected on a bus are arranged in a logical ring. When the logical ring is initiated, the highest number station may send the first frame. After this it passes permission to its immediate neighbor by sending a special frame called a token.

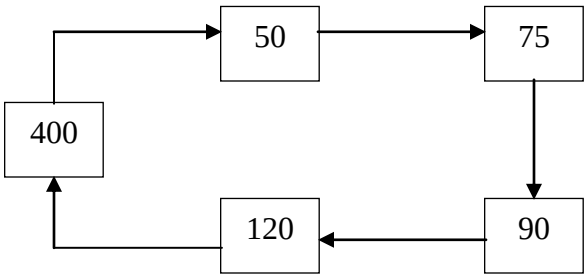
- The token propagates around the logical ring, with only the token holder being permitted to transmit frames. Since only one station at a time holds the token, collision do not occur.
- There is no relation between the physical location of the station on the bus and its logical sequence number..

The following figure shows the operation of the token bus.

Physical topology



Logical sequence of token passing



Token passing in a bus

802.4 cable standards

- The token bus standard specifies three physical layer options in terms of transmission medium, signaling technique, data rate and maximum electrical cable segment length.

Medium options

1. Broadband: Transmission medium is co-axial cable and its uses AM/PSK as a signaling techniques, data rate is 1,5,10 mbps.

2. Carrier band: Transmission medium is co-axial cable and its uses KSK as a signaling techniques, data rate is 1,5,10Mbps.

3. Optical fiber: Transmission medium is optical fiber and its uses ASK with Manchester encoding as a signaling techniques, data rate is 5,10,20Mbps.

IEEE 802.4 Frame format

- Token bus frame format is shown in the following figure.

1	1	1	2-6	2-6	0-8182	4	
Preamble	SD	FC	DA	SA	DATA	FCS	ED

- Preamble: the preamble is an at least one byte long pattern to establish bit synchronization
- SD: Start frame delimiter: Its also one byte unique bit pattern, which marks the start of the frame.
- FC: Frame control: The frame control field is used to distinguish data frames from control frames. For data frame, it carries the frames priority. The frame control field indicates the type of the frame data frame or control frame.
- DA: Destination address: The destination address field is 2 or 6 bytes long.
- SA: Source address: The destination address field is 2 or 6 bytes long.
- DATA: Data field
- FCS: Frame check sequence: frame check sequence is 4 bytes long and contains CRC code. It is used to detect transmission errors on DA, SA, FC and data fields.
- ED: End delimiter: It is a unique bit pattern, which marks the end of the frame. It is one byte long.
- The total length of the frame is 8191 bytes.

Performance:

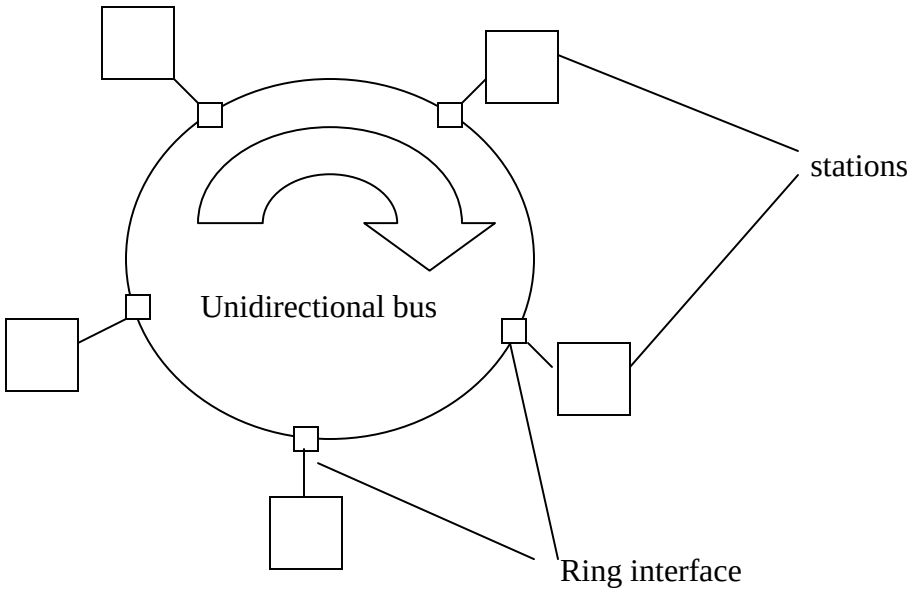
For token ring, the slightly higher delay compared to CSMS/CD bus occurs. For higher transmission loads the token ring performs well.

IEEE 802.5 TOKEN RING

- IEEE 802.4 describes a token ring LAN standard.
- In a token ring a special bit pattern, called the token circulates around the ring when all stations are idle.
- When a station transmits, it breaks the ring and inserts its own frame with source and destination address.

- When the frame eventually returns to the originating station after completing the round, the station removes the frame and closes the ring. Because there is only one token, only one station can transmit at a given instant, thus solving the channel access problem.
- Each station is connected to the ring through a Ring Interface Unit (RIU). The sequence of token is determined by the physical locations of the stations on the ring.

The following figure shows the operation and arrangement of the Token Ring.



802.5 cable standards

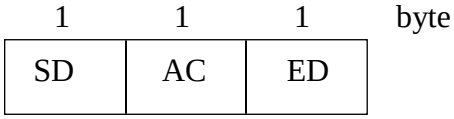
Its uses two types of transmission medium.

1. Shielded twisted pair cable: (STP)
   It uses differential Manchester encoding technique. Data rate is 4 or 16 Mbps. Maximum number of repeaters allowed is 250.
2. Unshielded twisted pair cable: (UTP)
   It uses differential Manchester encoding technique. Data rate is 4Mbps. Maximum number of repeaters allowed is 250.

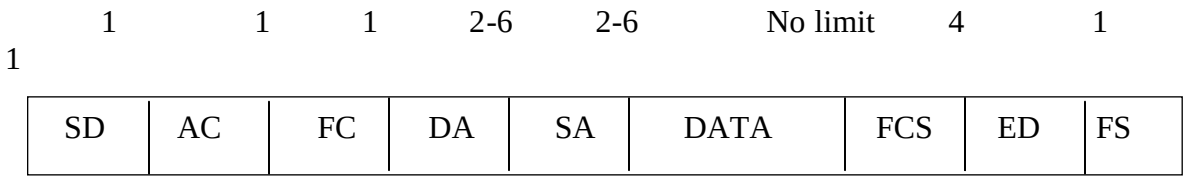
IEEE 802.5 Frame format

- Token ring frame format is shown in the following figure.

Token frame format



Data Frame



- SD: Start frame delimiter: Its also one byte unique bit pattern, which marks the start of the frame.
- AC: Access control: It is one byte long field containing priority bits(P), Token bit(T), monitoring bit(M), and reservation bir(R).
- FC: Frame control: The frame control field is used to distinguish data frames from control frames. For data frame, it carries the frames priority. The frame control field indicates the type of the frame data frame or control frame.
- DA: Destination address: The destination address field is 2 or 6 bytes long.
- SA: Source address: The destination address field is 2 or 6 bytes long.
- DATA: Data field
- FCS: Frame check sequence: frame check sequence is 4 bytes long and contains CRC code. It is used to detect transmission errors on DA, SA, FC and data fields.
- ED: End delimiter: It is a unique bit pattern, which marks the end of the frame. It is one byte long.
- FS: Frame status: This field is none byte long and contains a unique bit pattern marking the end of a token or a data frame.

Performance:

When traffic is light, the token will spend most of its time idly circulating around the ring. When traffic is heavy, there is a queue at each station. Network efficiency is more.

Disadvantages:

- A break in a link or repeater failures disturbs the entire network.
- Installation of new repeaters requires identification of two topologically adjacent repeaters.
- Since the ring is closed loop, a packet will circulate indefinitely unless it is removed.
- Each repeater adds an increment of delay.
- There is practical limit to the number of repeaters.

Fiber Distributed Data Interface

Introduction

The Fiber Distributed Data Interface (FDDI) specifies a 100-Mbps token-passing, dual-ring LAN using fiber-optic cable. FDDI is frequently used as high-speed backbone technology because of its support for high bandwidth and greater distances than copper. It should be noted that relatively recently, a related copper specification, called Copper Distributed Data Interface (CDDI), has emerged to provide 100-Mbps service over copper. CDDI is the implementation of FDDI protocols over twisted-pair copper wire. This chapter focuses mainly on FDDI specifications and operations, but it also provides a high-level overview of CDDI.

FDDI uses dual-ring architecture with traffic on each ring flowing in opposite directions (called counter-rotating). The dual rings consist of a primary and a secondary ring. During normal operation, the primary ring is used for data transmission, and the secondary ring remains idle. As will be discussed in detail later in this chapter, the primary purpose of the dual rings is to provide superior reliability and robustness. Figure 8-1 shows the counter-rotating primary and secondary FDDI rings.

FDDI Specifications

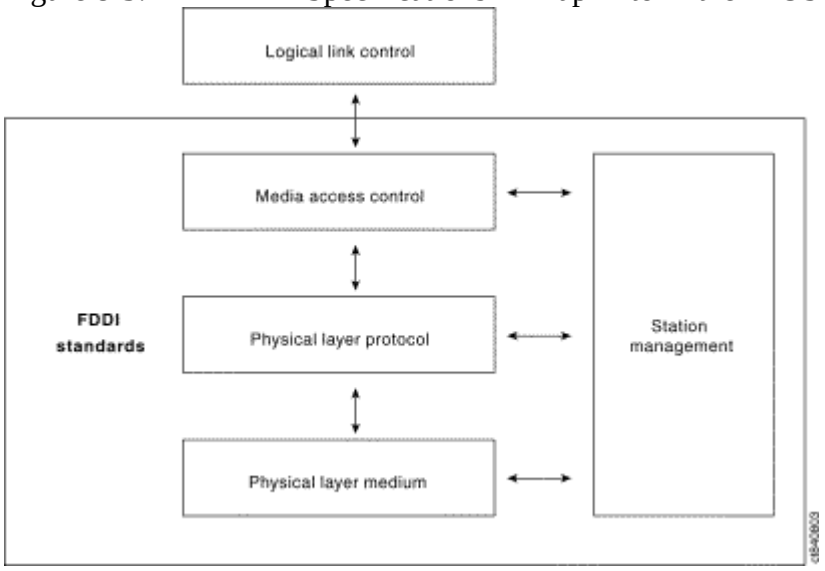
FDDI specifies the physical and media-access portions of the OSI reference model. FDDI is not actually a single specification, but it is a collection of four separate specifications, each with a specific function. Combined, these specifications have the capability to provide high-speed connectivity between upper-layer protocols such as TCP/IP and IPX, and media such as fiber-optic cabling.

FDDI's four specifications are the Media Access Control (MAC), Physical Layer Protocol (PHY), Physical-Medium Dependent (PMD), and Station Management (SMT) specifications. The MAC specification defines how the medium is accessed, including frame format, token handling, addressing, algorithms for calculating cyclic redundancy check (CRC) value, and error-recovery mechanisms. The PHY specification defines data encoding/decoding procedures, clocking requirements, and framing, among other functions. The PMD specification defines the characteristics of the transmission medium, including fiber-optic links, power levels, bit-error rates, optical components, and connectors. The SMT specification defines FDDI station configuration, ring

configuration, and ring control features, including station insertion and removal, initialization, fault isolation and recovery, scheduling, and statistics collection.

FDDI is similar to IEEE 802.3 Ethernet and IEEE 802.5 Token Ring in its relationship with the OSI model. Its primary purpose is to provide connectivity between upper OSI layers of common protocols and the media used to connect network devices. Figure 8-3 illustrates the four FDDI specifications and their relationship to each other and to the IEEE-defined Logical Link Control (LLC) sublayer. The LLC sublayer is a component of Layer 2, the MAC layer, of the OSI reference model.

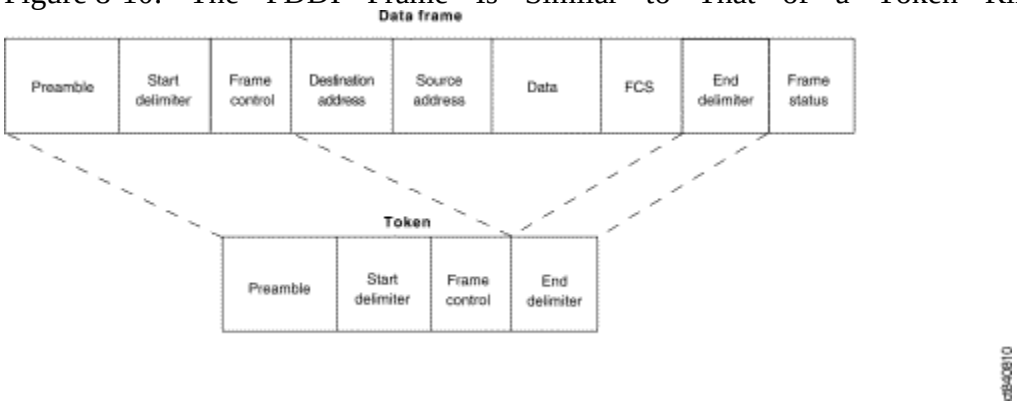
Figure 8-3: FDDI Specifications Map to the OSI Hierarchical Model



FDDI Frame Format

The FDDI frame format is similar to the format of a Token Ring frame. This is one of the areas in which FDDI borrows heavily from earlier LAN technologies, such as Token Ring. FDDI frames can be as large as 4,500 bytes. Figure 8-10 shows the frame format of an FDDI data frame and token.

Figure 8-10: The FDDI Frame Is Similar to That of a Token Ring Frame



FDDI Frame Fields

The following descriptions summarize the FDDI data frame and token fields illustrated in Figure 8-10.

- Preamble—Gives a unique sequence that prepares each station for an upcoming frame.
- Start delimiter—Indicates the beginning of a frame by employing a signaling pattern that differentiates it from the rest of the frame.
- Frame control—Indicates the size of the address fields and whether the frame contains asynchronous or synchronous data, among other control information.
- Destination address—Contains a unicast (singular), multicast (group), or broadcast (every station) address. As with Ethernet and Token Ring addresses, FDDI destination addresses are 6 bytes long.
- Source address—Identifies the single station that sent the frame. As with Ethernet and Token Ring addresses, FDDI source addresses are 6 bytes long.
- Data—Contains either information destined for an upper-layer protocol or control information.
- Frame check sequence (FCS)—Is filed by the source station with a calculated cyclic redundancy check value dependent on frame contents (as with Token Ring and Ethernet). The destination address recalculates the value to determine whether the frame was damaged in transit. If so, the frame is discarded.
- End delimiter—Contains unique symbols; cannot be data symbols that indicate the end of the frame.
- Frame status—Allows the source station to determine whether an error occurred; identifies whether the frame was recognized and copied by a receiving station.

Dual Ring

FDDI's primary fault-tolerant feature is the dual ring. If a station on the dual ring fails or is powered down, or if the cable is damaged, the dual ring is automatically wrapped (doubled back onto itself) into a single ring. When the ring is wrapped, the dual-ring topology becomes a single-ring topology. Data continues to be transmitted on the FDDI

ring without performance impact during the wrap condition. Figure 8-6 and Figure 8-7 illustrate the effect of a ring wrapping in FDDI.

Figure 8-6: A Ring Recovers from a Station Failure by Wrapping

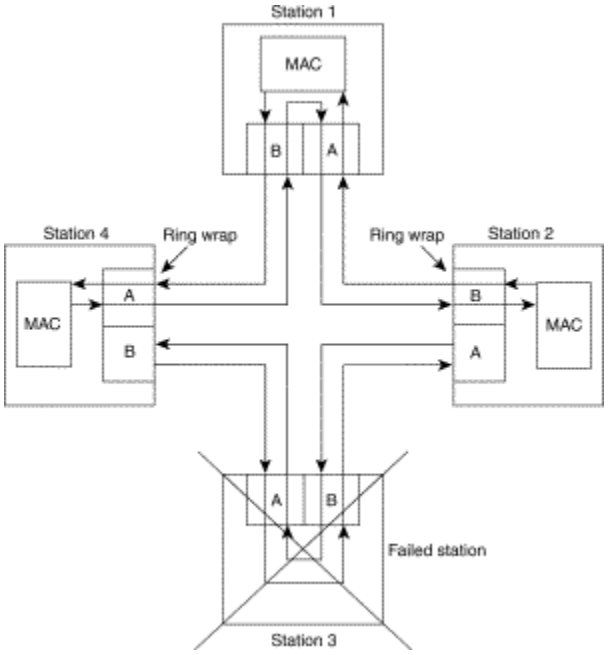
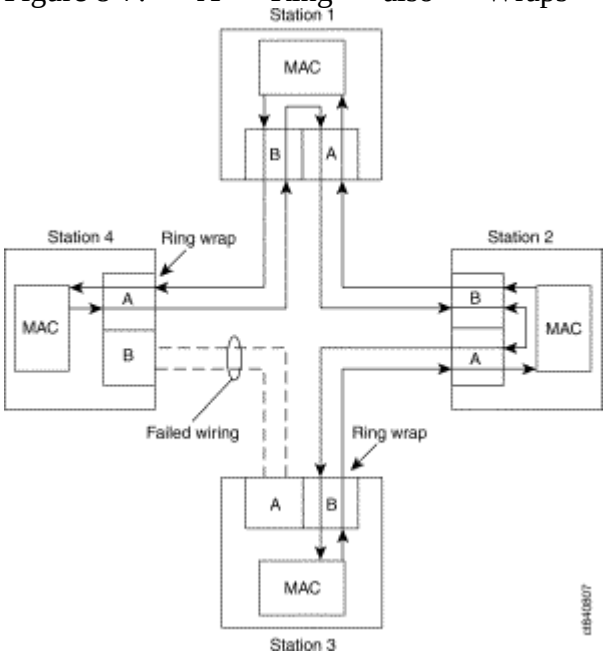


Figure 8-7: A Ring also Wraps to Withstand a Cable Failure



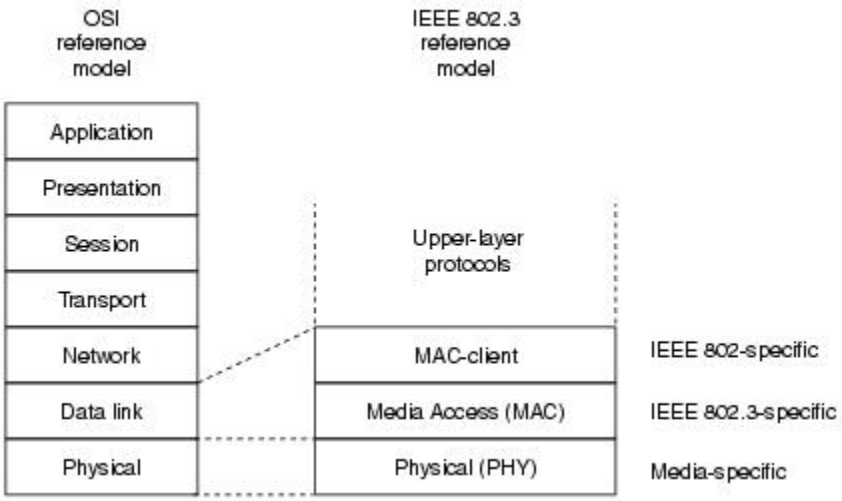
When a single station fails, as shown in Figure 8-6, devices on either side of the failed (or powered-down) station wrap, forming a single ring. Network operation continues for the remaining stations on the ring. When a cable failure occurs, as shown in Figure 8-7, devices on either side of the cable fault wrap. Network operation continues for all stations.

It should be noted that FDDI truly provides fault tolerance against a single failure only. When two or more failures occur, the FDDI ring segments into two or more independent rings that are incapable of communicating with each other.

The IEEE 802.3 Logical Relationship to the ISO Reference Model

Figure 7-4 shows the IEEE 802.3 logical layers and their relationship to the OSI reference model. As with all IEEE 802 protocols, the ISO data link layer is divided into two IEEE 802 sublayers, the Media Access Control (MAC) sublayer and the MAC-client sublayer. The IEEE 802.3 physical layer corresponds to the ISO physical layer.

Figure 7-4 Ethernet's Logical Relationship to the ISO Reference Model



The MAC-client sublayer may be one of the following:

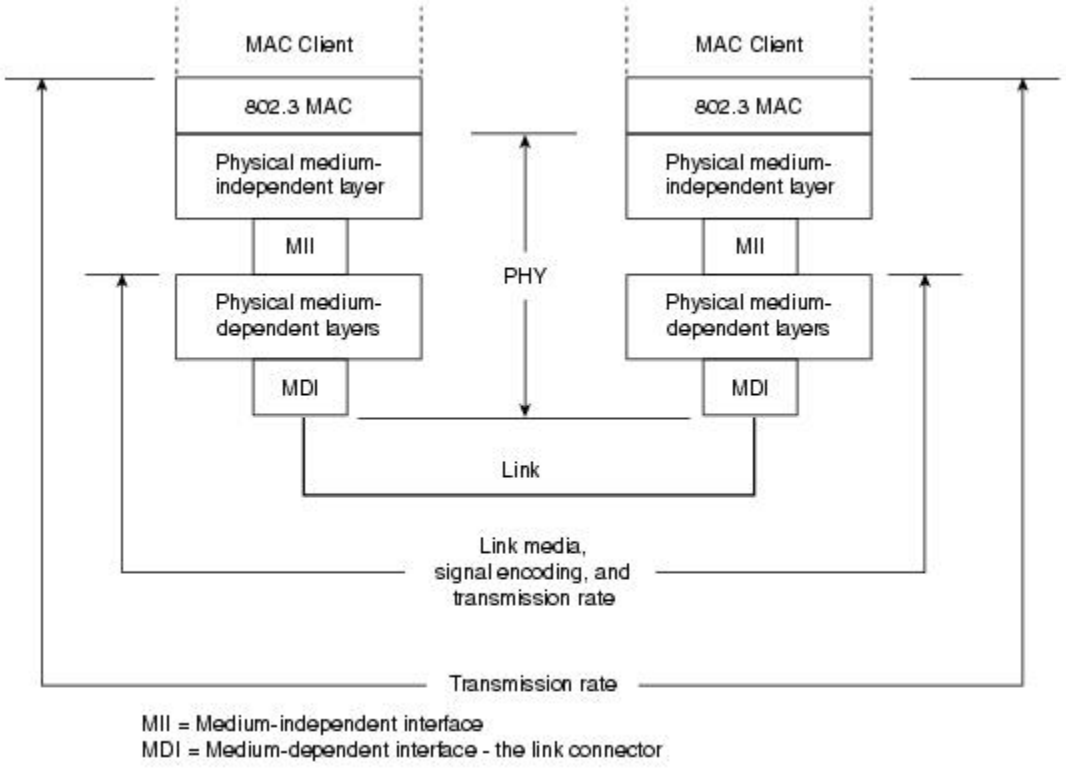
- Logical Link Control (LLC), if the unit is a DTE. This sublayer provides the interface between the Ethernet MAC and the upper layers in the protocol stack of the end station. The LLC sublayer is defined by IEEE 802.2 standards.

- Bridge entity, if the unit is a DCE. Bridge entities provide LAN-to-LAN interfaces between LANs that use the same protocol (for example, Ethernet to Ethernet) and also between different protocols (for example, Ethernet to Token Ring). Bridge entities are defined by IEEE 802.1 standards.

Because specifications for LLC and bridge entities are common for all IEEE 802 LAN protocols, network compatibility becomes the primary responsibility of the particular

network protocol. Figure 7-5 shows different compatibility requirements imposed by the MAC and physical levels for basic data communication over an Ethernet link.

Figure 7-5 MAC and Physical Layer Compatibility Requirements for Basic Data Communication



The MAC layer controls the node's access to the network media and is specific to the individual protocol. All IEEE 802.3 MACs must meet the same basic set of logical requirements, regardless of whether they include one or more of the defined optional protocol extensions. The only requirement for basic communication (communication that does not require optional protocol extensions) between two network nodes is that both MACs must support the same transmission rate.

The 802.3 physical layer is specific to the transmission data rate, the signal encoding, and the type of media interconnecting the two nodes. Gigabit Ethernet, for example, is defined to operate over either twisted-pair or optical fiber cable, but each specific type of cable or signal-encoding procedure requires a different physical layer implementation.

The Ethernet MAC Sublayer

The MAC sub layer has two primary responsibilities:

- Data encapsulation, including frame assembly before transmission, and frame parsing/error detection during and after reception

- Media access control, including initiation of frame transmission and recovery from transmission failure

The Basic Ethernet Frame Format

The IEEE 802.3 standard defines a basic data frame format that is required for all MAC implementations, plus several additional optional formats that are used to extend the protocol's basic capability. The basic data frame format contains the seven fields shown in Figure 7-6.

- Preamble (PRE)—Consists of 7 bytes. The PRE is an alternating pattern of ones and zeros that tells receiving stations that a frame is coming, and that provides a means to synchronize the frame-reception portions of receiving physical layers with the incoming bit stream.

- Start-of-frame delimiter (SOF)—Consists of 1 byte. The SOF is an alternating pattern of ones and zeros, ending with two consecutive 1-bits indicating that the next bit is the left-most bit in the left-most byte of the destination address.

- Destination address (DA)—Consists of 6 bytes. The DA field identifies which station(s) should receive the frame. The left-most bit in the DA field indicates whether the address is an individual address (indicated by a 0) or a group address (indicated by a 1). The second bit from the left indicates whether the DA is globally administered (indicated by a 0) or locally administered (indicated by a 1). The remaining 46 bits are a uniquely assigned value that identifies a single station, a defined group of stations, or all stations on the network.

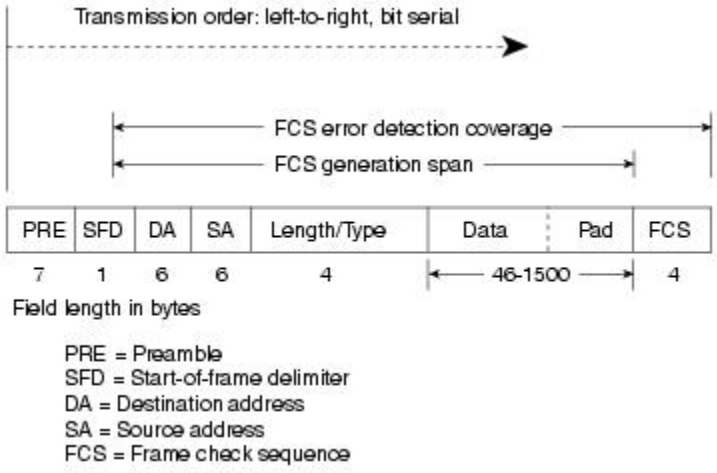
- Source addresses (SA)—Consists of 6 bytes. The SA field identifies the sending station. The SA is always an individual address and the left-most bit in the SA field is always 0.

- Length/Type—Consists of 2 bytes. This field indicates either the number of MAC-client data bytes that are contained in the data field of the frame, or the frame type ID if the frame is assembled using an optional format. If the Length/Type field value is less than or equal to 1500, the number of LLC bytes in the Data field is equal to the Length/Type field value. If the Length/Type field value is greater than 1536, the frame is an optional type frame, and the Length/Type field value identifies the particular type of frame being sent or received.

- Data—Is a sequence of n bytes of any value, where n is less than or equal to 1500. If the length of the Data field is less than 46, the Data field must be extended by adding a filler (a pad) sufficient to bring the Data field length to 46 bytes.

• Frame check sequence (FCS)—Consists of 4 bytes. This sequence contains a 32-bit cyclic redundancy check (CRC) value, which is created by the sending MAC and is recalculated by the receiving MAC to check for damaged frames. The FCS is generated over the DA, SA, Length/Type, and Data fields.

Figure 7-6 The Basic IEEE 802.3 MAC Data Frame Format



Note Individual addresses are also known as unicast addresses because they refer to a single MAC and are assigned by the NIC manufacturer from a block of addresses allocated by the IEEE. Group addresses (a.k.a. multicast addresses) identify the end stations in a workgroup and are assigned by the network manager. A special group address (all 1s—the broadcast address) indicates all stations on the network.

Frame Transmission

Whenever an end station MAC receives a transmit-frame request with the accompanying address and data information from the LLC sublayer, the MAC begins the transmission sequence by transferring the LLC information into the MAC frame buffer.

• The preamble and start-of-frame delimiter are inserted in the PRE and SOF fields.

• The destination and source addresses are inserted into the address fields.

• The LLC data bytes are counted, and the number of bytes is inserted into the Length/Type field.

• The LLC data bytes are inserted into the Data field. If the number of LLC data bytes is less than 46, a pad is added to bring the Data field length up to 46.

• An FCS value is generated over the DA, SA, Length/Type, and Data fields and is appended to the end of the Data field.

After the frame is assembled, actual frame transmission will depend on whether the MAC is operating in half-duplex or full-duplex mode.

The IEEE 802.3 standard currently requires that all Ethernet MACs support half-duplex operation, in which the MAC can be either transmitting or receiving a frame, but it cannot be doing both simultaneously. Full-duplex operation is an optional MAC capability that allows the MAC to transmit and receive frames simultaneously.

UNIT III

Network Layer

• Transport segment from sending to receiving host.
• On sending side encapsulates segments into datagrams.
• On receiving side, delivers segments to transport layer.
• network layer protocols in every host, router.
• Router examines header fields in all IP datagrams passing through it.

Network-Layer Functions

• forwarding: move packets from router's input to appropriate router Output.
• routing: determine route taken by packets from source to destination.

Internetworking

- Internetworking is a scheme for interconnecting multiple networks of dissimilar technologies
- Uses both hardware and software
  - Extra hardware positioned between networks
  - Software on each attached computer
- System of interconnected networks is called an internetwork or an internet

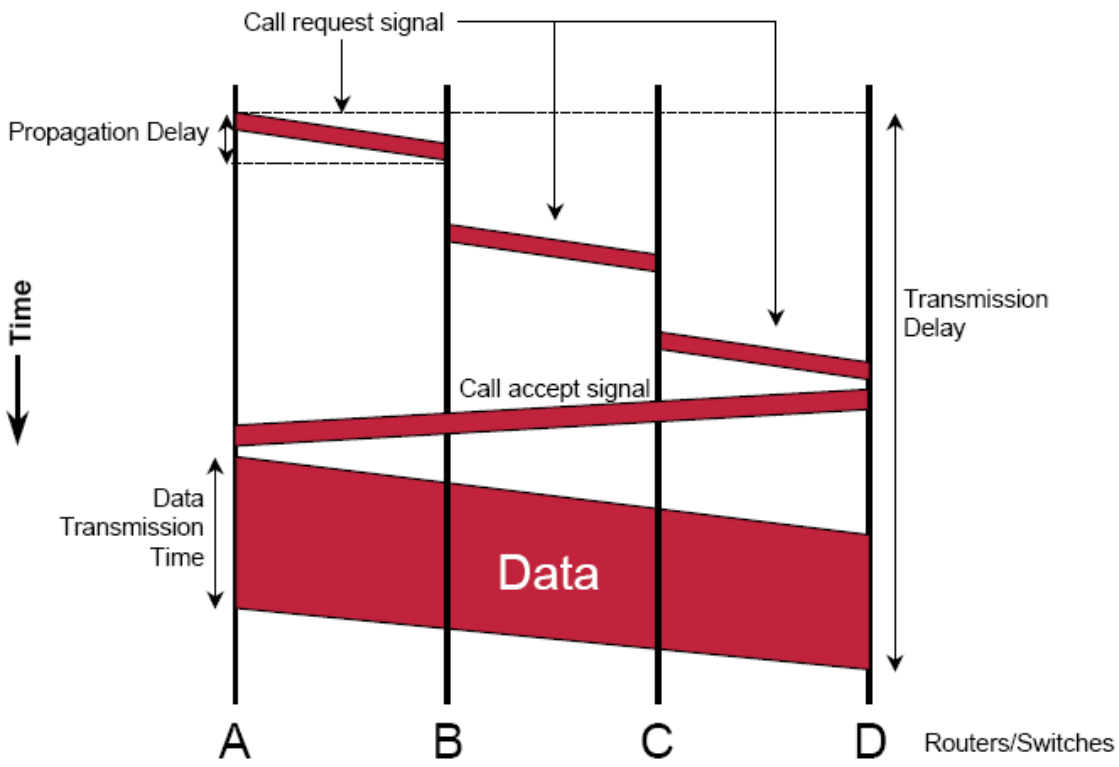
Switching Schemes

(1) Circuit Switching
(2) Message Switching (Store-and-Forward)
(3) Packet Switching (Store-and-Forward)

Circuit Switching

• Provides service by setting up the total path of connected lines hop-by-hop from the origin to the destination
• Example: Telephone network

1. Control message sets up a path from origin to destination
2. Return signal informs source that data transmission may proceed
3. Data transmission begins
4. Entire path remains allocated to the transmission (whether used or not)
5. When transmission is complete, source releases the circuit.



VC implementation

A VC consists of:

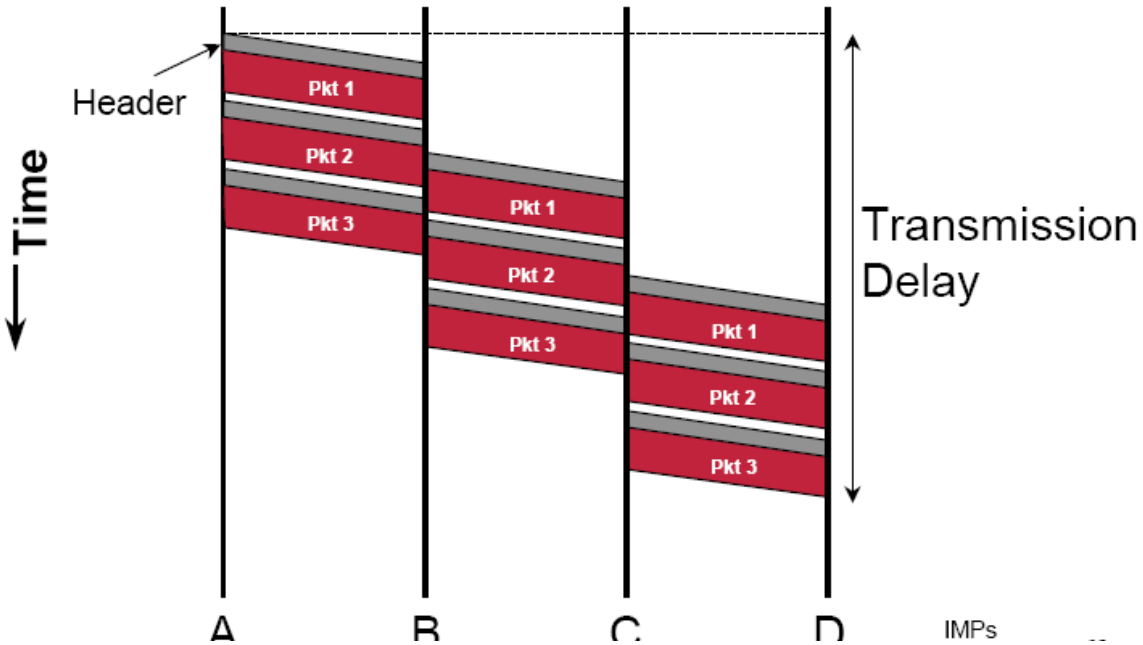
1. Path from source to destination
2. VC numbers, one number for each link along path
3. Entries in forwarding tables in routers along path

- Packet belonging to VC carries a VC number.
- VC number must be changed on each link.
- New VC number comes from forwarding table

Packet Switching

• Messages are split into smaller pieces called packets.
• These packets are numbered and addressed and sent through the network one at a time.
• Allows Pipelining

– Overlap sending and receiving of packets on multiple links.



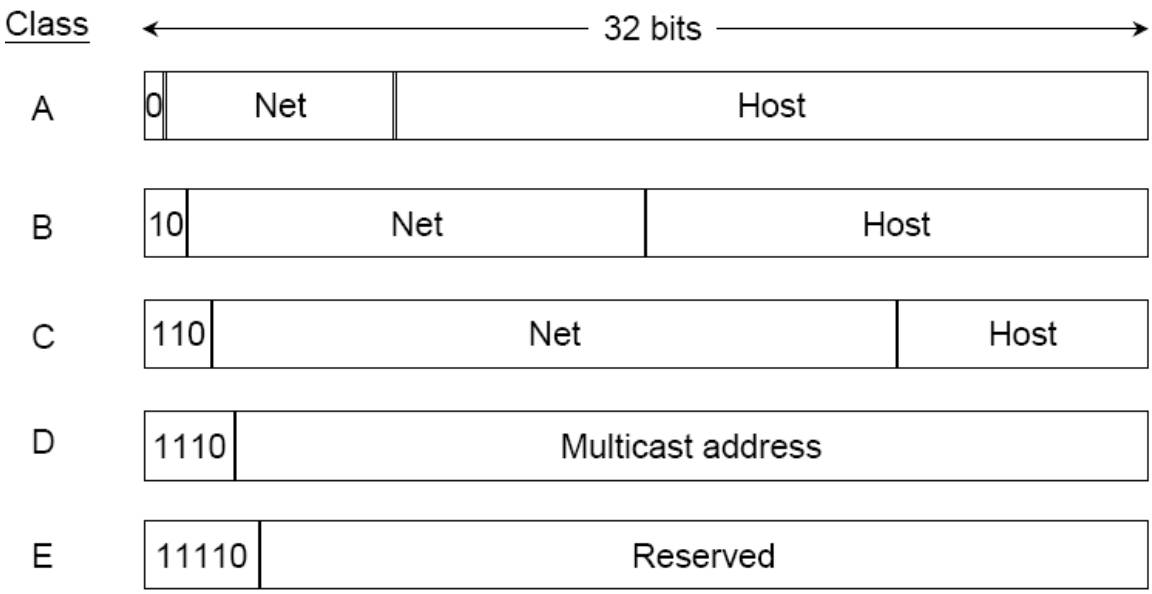
IP Addresses

Each network interface on the Internet as a unique global address, called the IP address. An IP address- is 32 bits long. It encodes a network number and a host number.IP addresses are written in a dotted decimal notation:
128.238.42.112 means

10000000 in 1st Byte
11101110 in 2nd Byte
00101010 in 3rd Byte
01110000 in 4th Byte

Internet Address Classes:

IP distinguishes 5 classes of addresses.



IP Address classes

- Class A:
  - For very large organizations
  - 16 million hosts allowed
- Class B:
  - For large organizations
  - 65 thousand hosts allowed
- Class C
  - For small organizations
  - 255 hosts allowed
- Class D
  - Multicast addresses
  - No network/host hierarchy

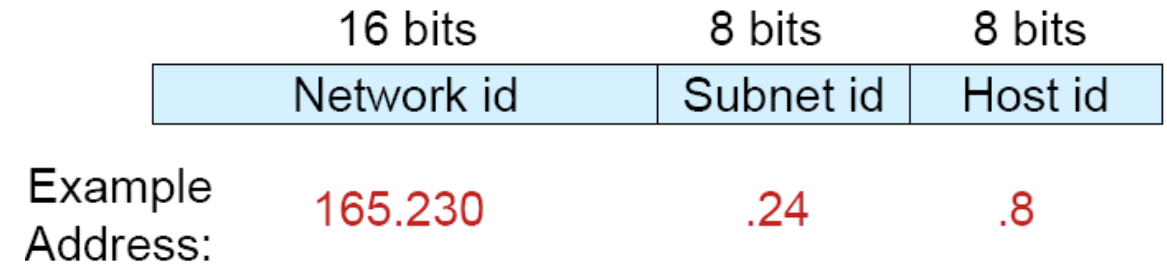
Internet= a collection of connected networks which share a common set of rules for communication
IP Address Hierarchy

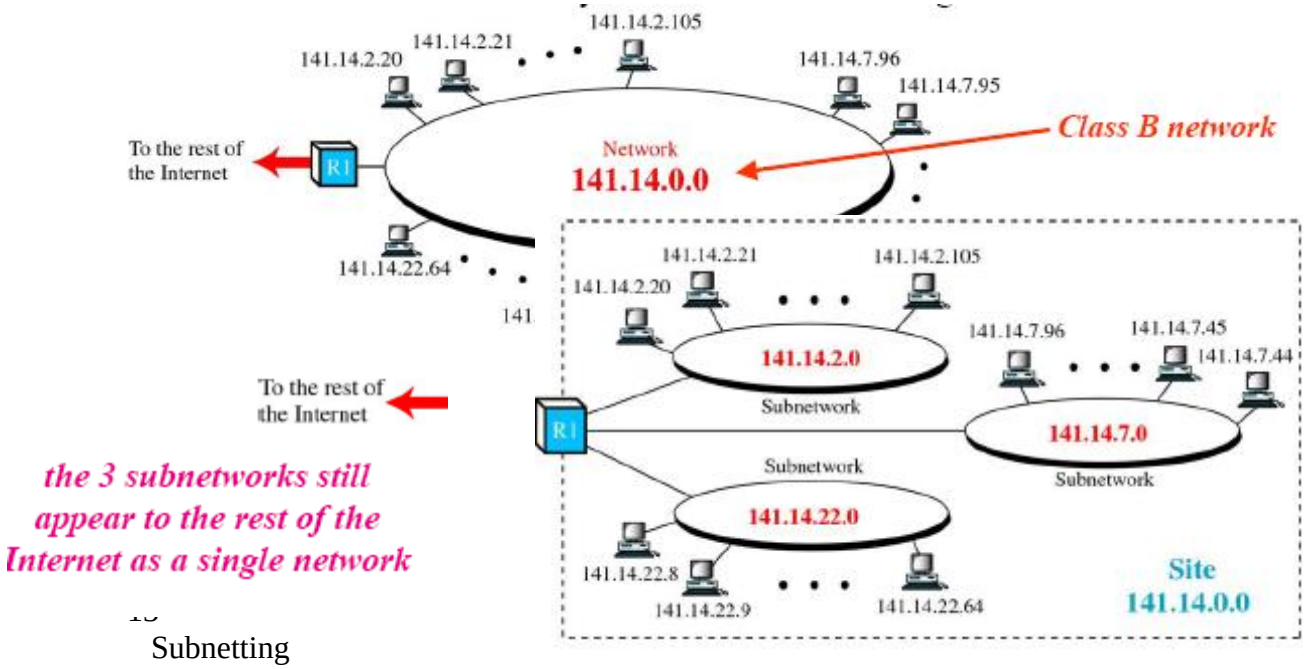
- Note that Class A, Class B, and Class C addresses only support two levels of hierarchy
- Each address contains a network and a host portion, meaning two levels of Hierarchy.
- However, the host portion can be further split into “subnets” by the address class owner
- This allows for more than 2 levels of hierarchy.

IP Subnetting

- Subnetting is a technique used to allow a single IP network address to Span multiple physical networks
- IP hosts should support subnetting.
- Subnetting is done by using some of the bits of the host-id part of the IP address physical layer network identifier
- The subnet mask is used to determine the bits of the network identifier.
- All hosts on the same network should have the same subnet mask.
- IP address is composed of a Netid part and a Hostid part ⇒2-level hierarchy.

Sometimes a 2-level hierarchy is insufficient for an organisation's needs.





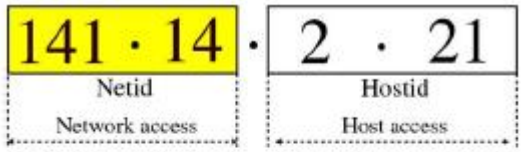
Subnetting

- An IP packet from some other network destined for host 141.14.2.21 still reaches router R1, since the destination address is still a Class B address with Netid141.14 and Hostid 2.21 as far as the rest of the Internet is concerned.
- when the packet reaches router R1, the interpretation of the IP address changes

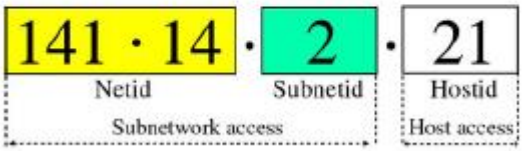
- R1 knows that there are 3 levels of hierarchy within the organization, and that in this case, the Netid is 141.14,the Subnetid is 2, and the Hostid is 21.
- How is this knowledge of the internal network hierarchy implemented in the organization's routers?
- Masking of IP addresses during the packet-forwarding process.
- Masking is done whether or not subnetting is being used with subnetting,

the Netid defines the site, the Subnetid defines the physical network, and the Hostid defines the actual machine.

14



a. Without subnetting



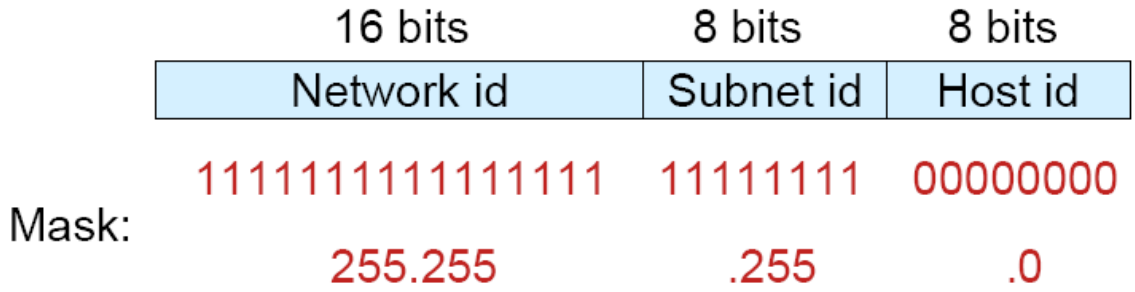
b. With subnetting

15
16
17

Subnet Masks

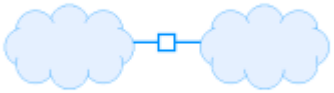
Subnet masks allow hosts to determine if another IP address is on the same subnet or the same network.

18
19



20
21
22
23 **Router**
24

- A router is a hardware component used to interconnect networks
- A router has interfaces on multiple networks



- Networks can use different technologies
- Router forwards packets between networks
- Transforms packets as necessary to meet standards for each network

25
26

Routing issues:

Scalability: must be able to support large numbers of hosts, routers, networks

Adapt to changes in topology or significant changes in traffic, quickly and efficiently

self-healing: little or not human intervention

Route selection may depend on different criteria

Performance: "choose route with smallest delay"

Policy : "choose a route that doesn't cross a government network" (equivalently: "let no non-government traffic cross this network")

27

Classification of Routing algorithms

Centralized versus decentralized

Centralized: central site computes and distributed routes (equivalently: information for computing routes known globally, each router makes same computation)

Decentralized: each router sees only local information (itself and physically-connected neighbors) and computes routes on this basis. pros and cons?

Static versus adaptive

Static: routing tables change very slowly, often in response to human intervention
Dynamic: routing tables change as network traffic or topology change

Two basic approaches adopted in practice:

Link-state routing: centralized, dynamic (periodically run)
Distance vector: distributed, dynamic (in direct response to changes)
28
29
30 Routers
31
Routers are distinguished by the functions they perform
Internal routers
- Only route packets within one area

Area border routers
- Connect to areas together

Backbone routers
- Reside only in the backbone area

AS boundary routers
- Routers that connect to a router outside the AS

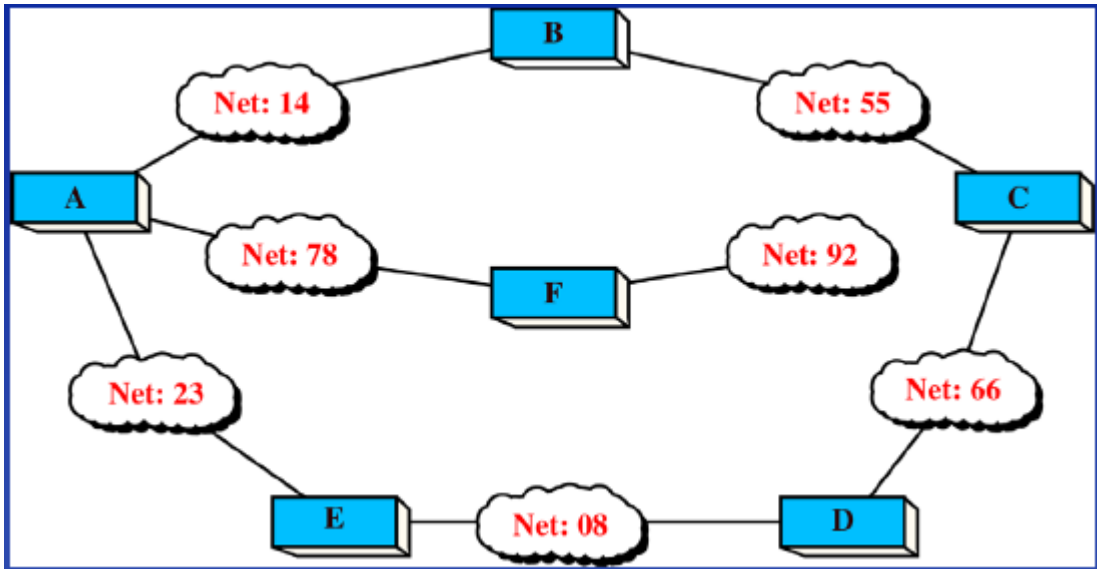
32
33
34 Routing
35 The most common routing algorithms are distance-vector and link-state routing.

Distance-vector:
- Each router exchanges information about the entire network with neighboring routers at regular intervals.
- Neighboring routers = connected by a direct link (e.g. a LAN)
- Regular intervals: e.g. every 30 seconds

Link-state:
- Each router exchanges information about its neighborhood with all routers in the network when there is a change.
- Neighborhood of a router = set of neighbor routers for this router.
- Each router's neighborhood information is flooded through the network.

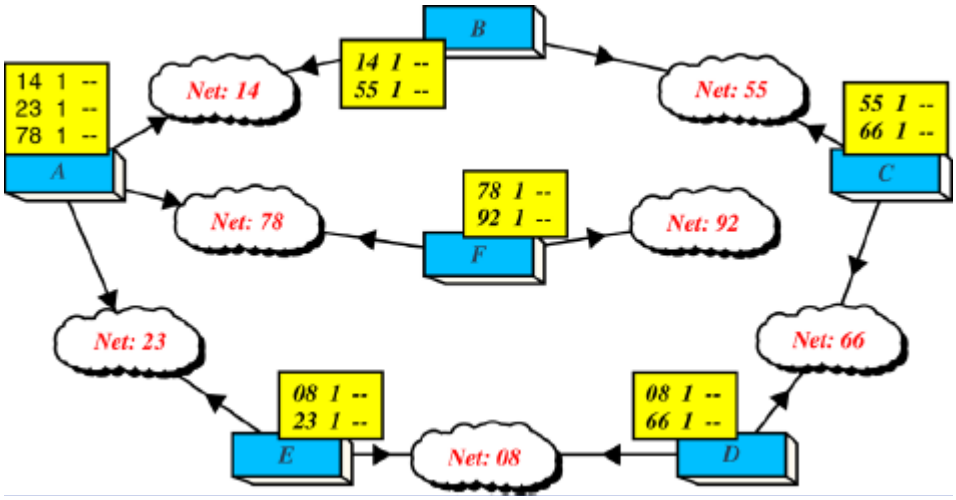
- Change: e.g. if a neighboring router does not reply to a status message.

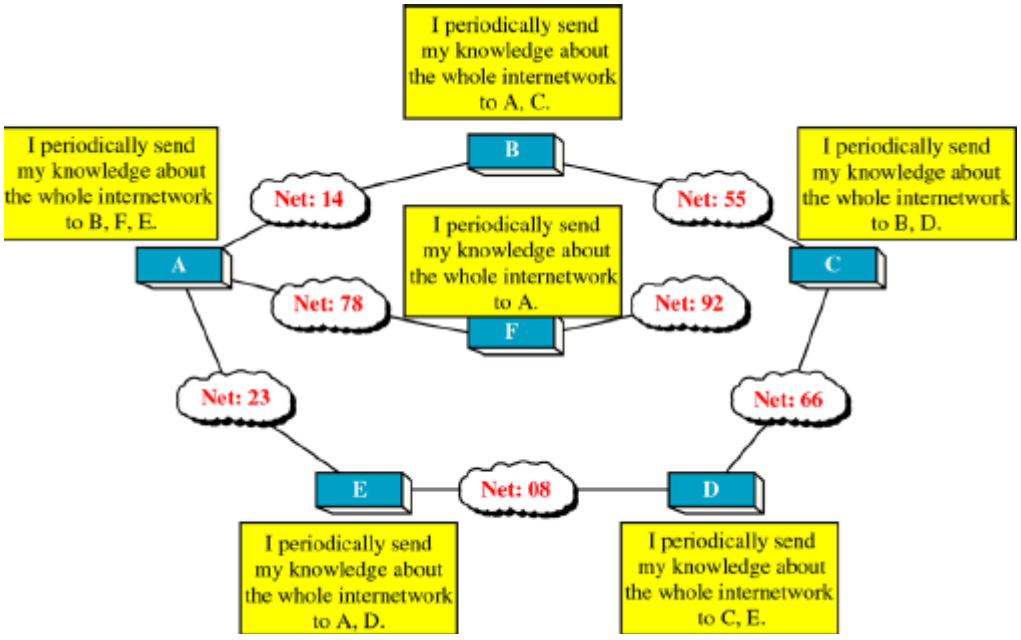
1



- "Clouds" represent LANs; number in cloud represents network ID
- A, B, C, D, E, F are routers (or gateways)

- Each router sends its information about the entire network only to its neighbors





How do non-neighboring routers learn about each other and share information?

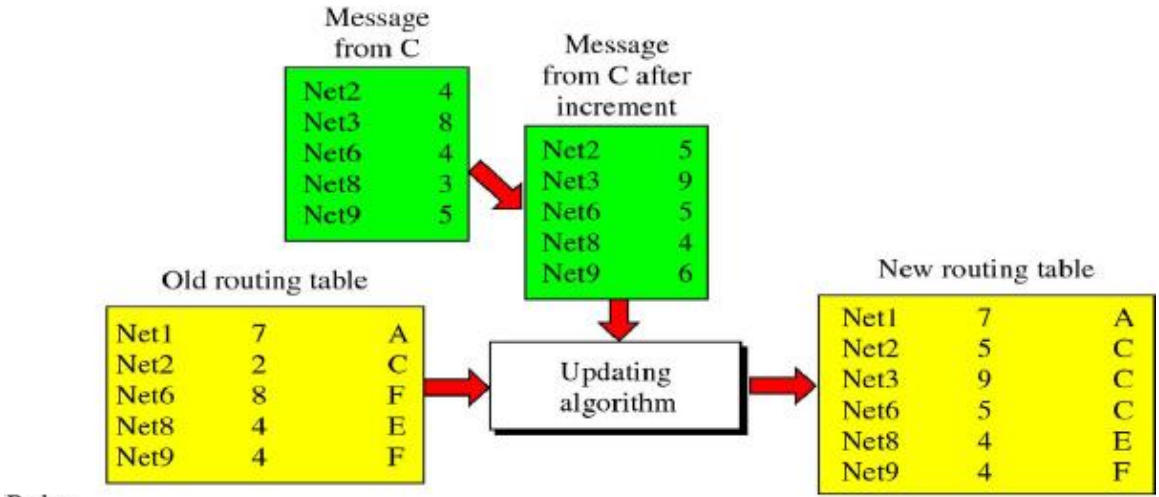
•A router sends its information to its neighbors

•Each neighbor router adds this information to its own, and sends the updated information to its neighbors; the first router learns about its neighbors' neighbors.

Routing table update algorithm (distributed Bellman-Ford algorithm):

- Add 1 to cost of each incoming route (since each neighbor is 1hop away)
- If a new destination is learned, add its information to the routing table
- If new information received on an existing destination:
- If Next Hop field is the same, replace existing entry with the new information even if the cost is greater("new information invalidates old")
- If Next Hop field is not the same, only replace existing entry with the new information if the cost is lower

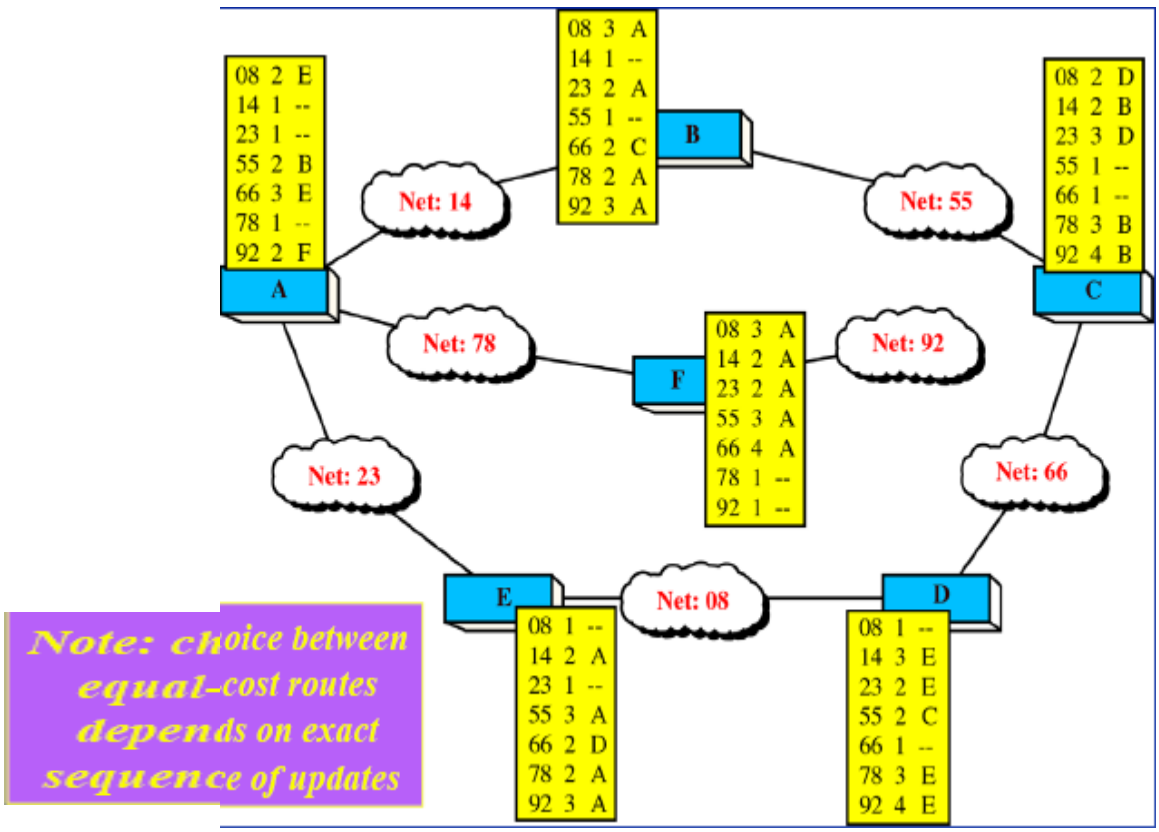
Example of routing table update algorithm



Rules
Net2: Replace **(Rule 2.a)**
Net3: Add **(Rule 1)**
Net6: Replace **(Rule 2.b.i)**
Net8: No change **(Rule 2.b.ii)**
Net9: No change **(Rule 2.b.ii)**

Note that there is no news about Net1 in the advertised message, so none of the rules apply to this entry.

Final routing tables :

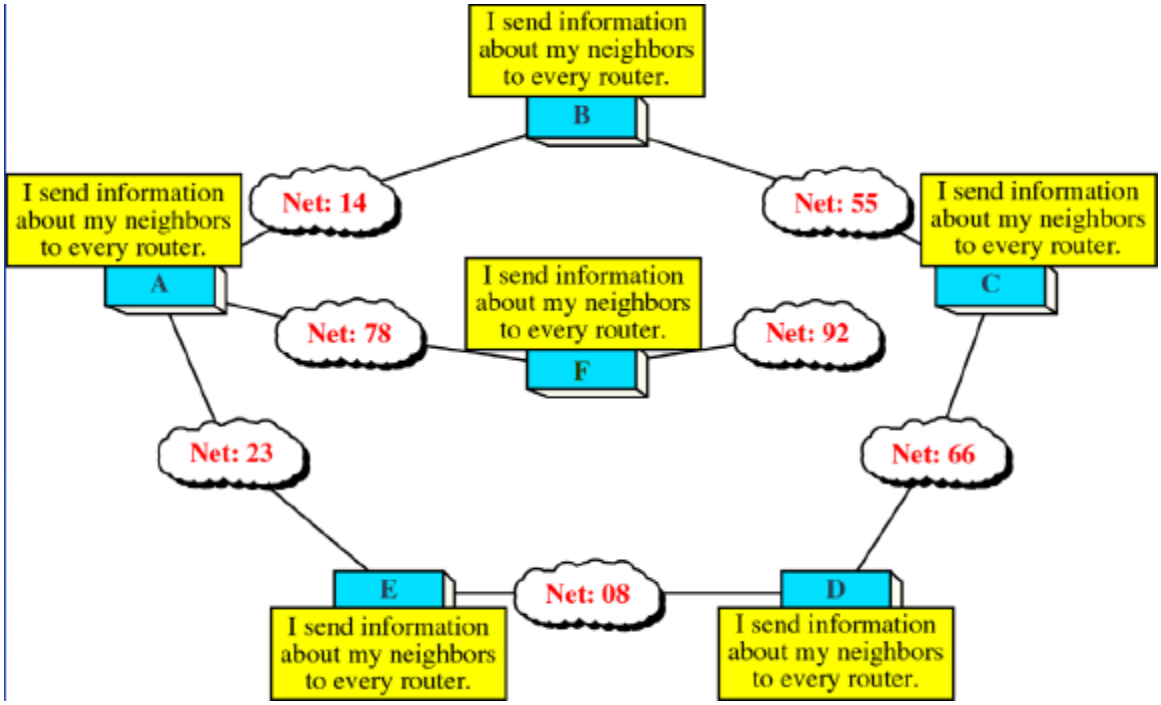


Problem with distance-vector routing:

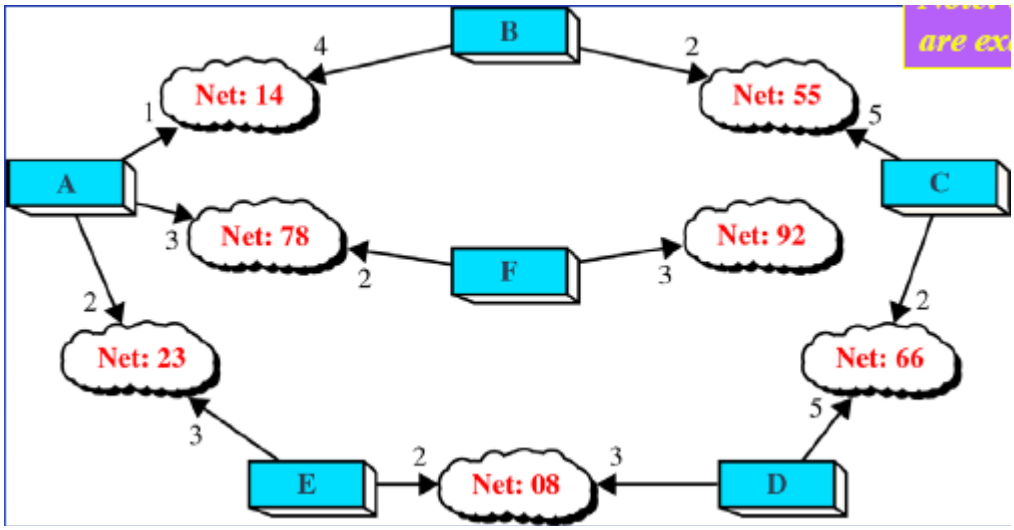
- Slow convergence of distance vector routing algorithms under some conditions
- Slow reaction to link/router failure because information only comes from neighboring routers and it may be out-of-date (e.g. it may not properly reflect the impact of the failure on route costs)

Link-State routing

•Each router sends information about its neighborhood to every other router



- Link cost is usually a weighted sum of various factors
- e.g. traffic level, security level, packet delay
- Link cost is from a router to the network connecting it to another router.
- when a packet is in a LAN (which is typically a broadcast network), every node –including the router –can receive it
- No cost assigned when going .

1 

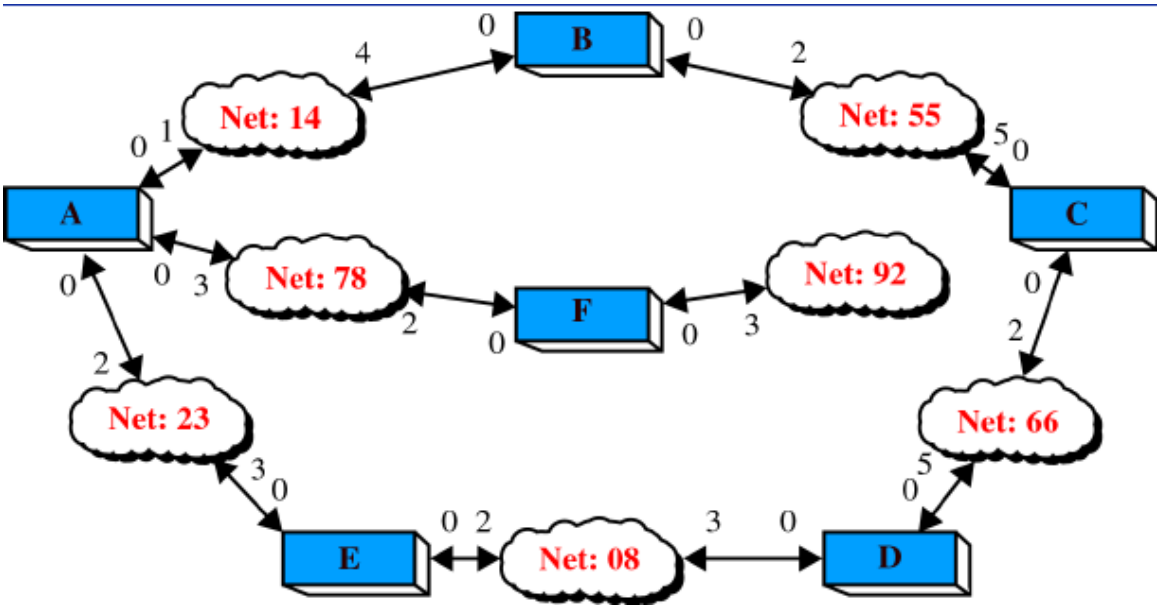
- Routers share information by advertising, which means sending link-state packets•.

- Every router builds a link-state packet and floods it through the network, so when all such packets have been received at a router, it can build its link-state database.
- Assuming that every router receives the same set of link-state packets (as if the routers were synchronized),every router builds the same link-state database. Using this database, each router can then calculate its routing table.

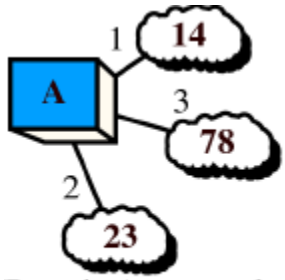
To calculate its routing table, a router uses Dijkstra's Shortest-Path algorithm

- First, identify all link costs in the network: either from the link-state database, or using the fact that the cost of any link from a network to a router is 0
- This algorithm builds a shortest-path spanning tree for the router such a tree has a route to all possible destinations, and no loops.
- The router running the algorithm is the root of its shortest-path spanning tree.
- Even if all routers' link-state databases are identical, the trees determined by the routers are different (since the root of each tree is different)
- A node is either a network or a router; nodes are connected by arcs.

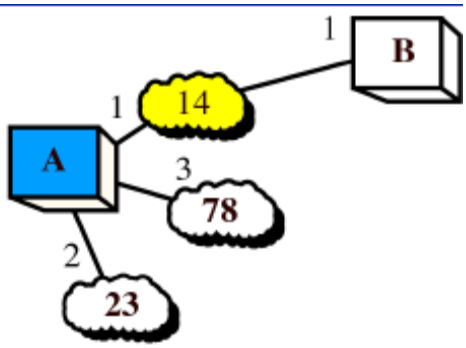
- The algorithm keeps track of 2 sets of nodes and arcs –Temporary and Permanent.
- Initially, the Temporary set contains all neighbor nodes of the router itself, and the arcs connecting them to the router; only the router is initially Permanent.
- When all nodes and arcs are in the Permanent set, the algorithm has terminated.
- Identify the Temporary node whose arc has the lowest cumulative cost from the root: this node and arc are moved into the Permanent set.
- Any nodes which are connected to the new Permanent node and are not already in the Temporary set, along with the connecting arcs, are made Temporary.
- Also, if any node already in the Temporary set has a lower cumulative cost from the root by using a route passing through the new Permanent node, then this new route replaces the existing one
- Repeat until all nodes and arcs are Permanent.



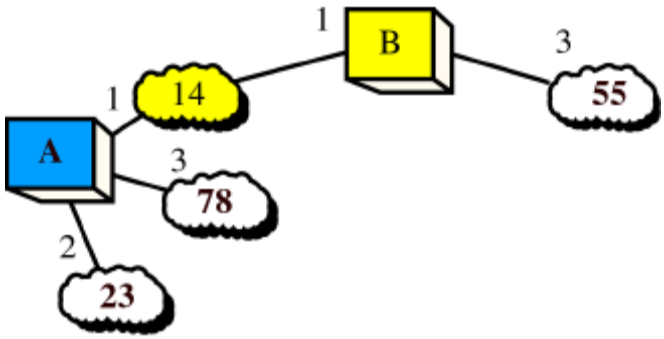
1
2
Let's follow the steps of the algorithm run by router A.



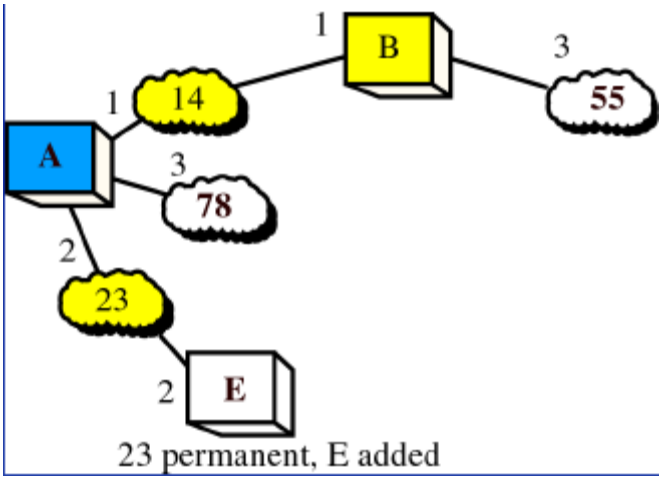
Root is A, networks
14, 78, 23 added



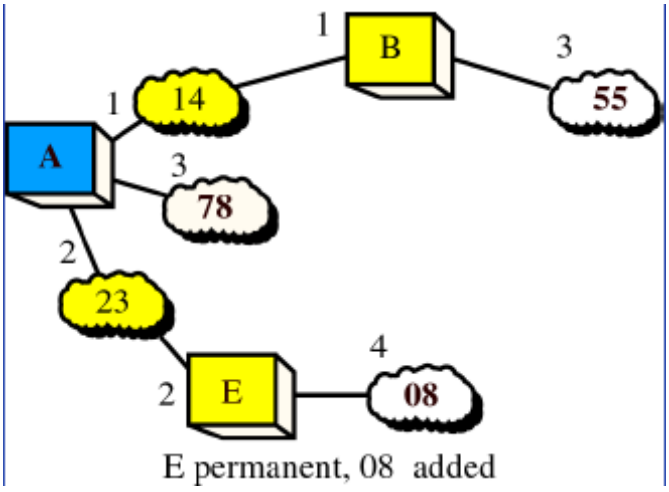
14 permanent, B added



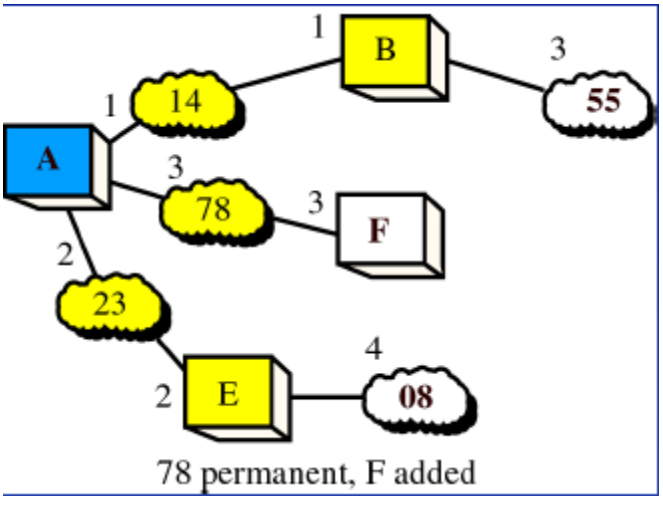
B Permanent, 55 added



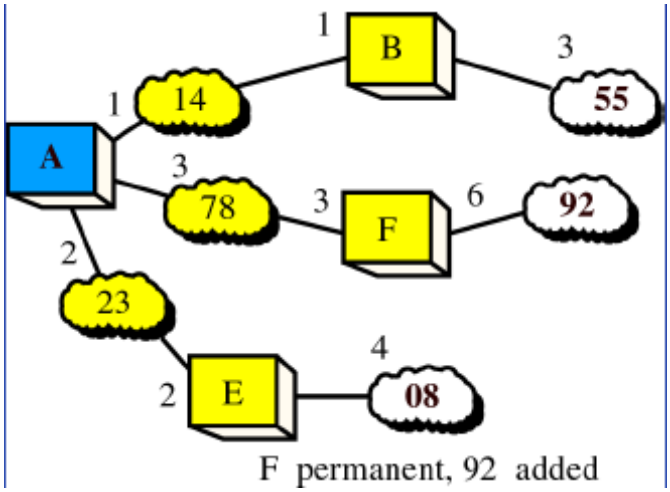
23 permanent, E added



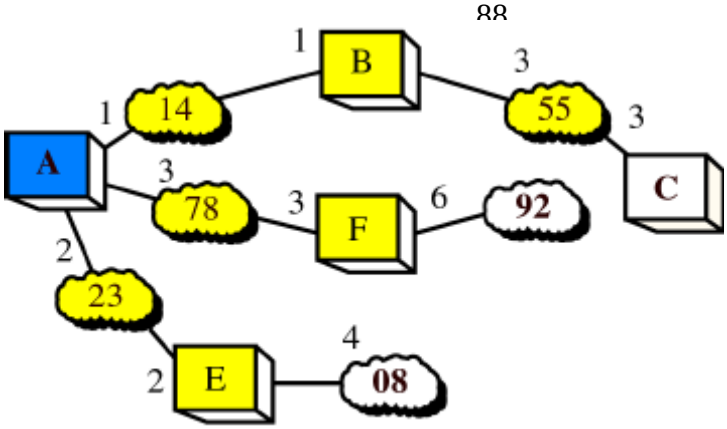
E permanent, 08 added



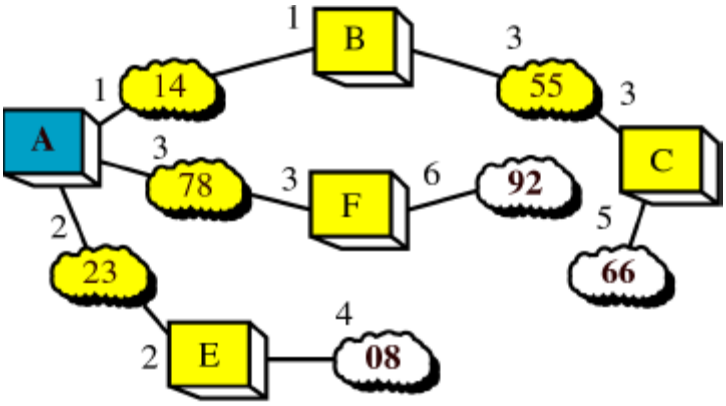
78 permanent, F added



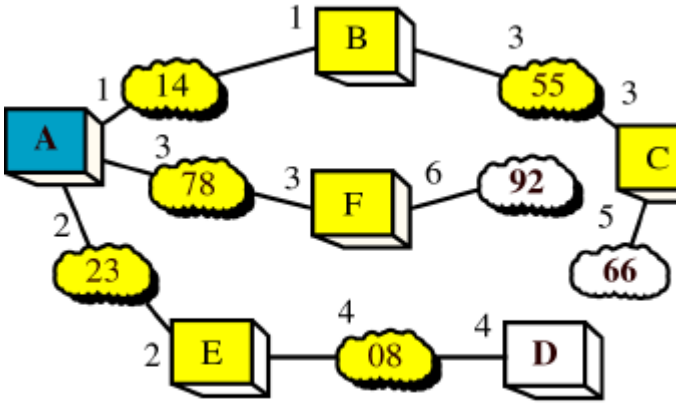
F permanent, 92 added



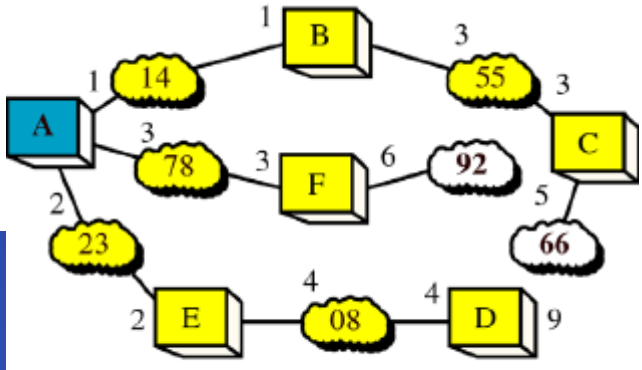
55 permanent, C added



C permanent, 66 added

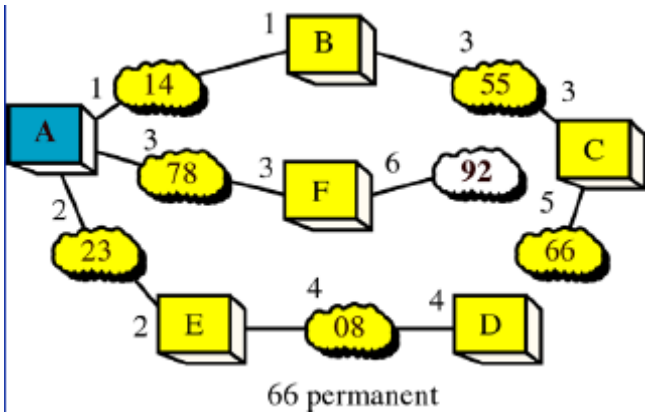


08 permanent, D added

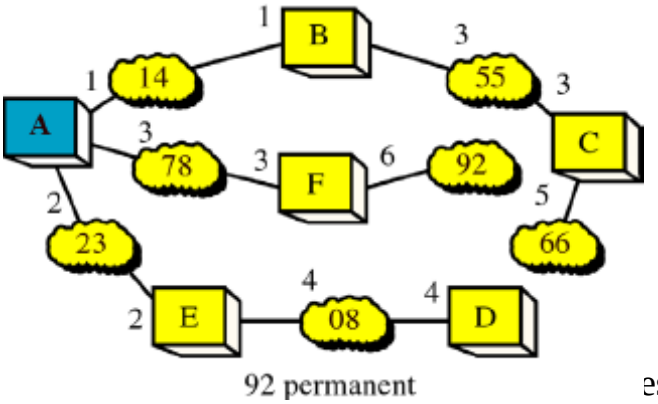


D permanent, 66 added.
But 9 > 5, so that link deleted

if the new arc to network 66 from router D had a lower cumulative cost than the one from router C, then the new link would replace the old one



Note: arcs are marked with their cumulative cost from the root (not individual costs)



all nodes and arcs are Permanent ⇒ STOP: this router's shortest-path spanning tree has been found

est-path spanning tree, it can build its routing table.

- To complete the Example, here is router A's link-state routing table

Net	Cost	Next router
08	4	E
14	1	--
23	2	--
55	3	B
66	5	B
78	3	--
92	6	F

Note: each router's routing table will (in general) be different

Networks 14, 23, and 78 don't have a "Next router" entry because they are directly connected to this router

- In large networks, the memory required to store the link-state database and the computation time to calculate the link-state routing table can be significant.
- in practice, since the link-state packet receptions are not synchronized, routers may be using different link-state databases to build their routing tables.

Note:

- Link-state routing algorithms have several desirable properties, e.g. rapid convergence; small amount of traffic generated; rapid response to topology changes.

UNIT - IV

The Transport Layer is responsible for end-to-end data transport
Primary functions include:

- Provision of connection oriented or connectionless service.
- Disassembling and reassembling data.
- Setup and release of connections across the network.

Services provided by Internet transport protocols

TCP service:
• connection- oriented: setup required between client, server
• reliable transport between sending and receiving process
• flow control: sender won't overwhelm receiver
• congestion control: throttle sender when network overloaded
• does not provide: timing, minimum bandwidth

UDP service:
• unreliable data transfer between sending and receiving process
• does not provide: connection setup, reliability, flow control, congestion control, timing, or bandwidth guarantee guarantees

UDP

•UDP is a connectionless transport protocol–extends IP's host-to-host delivery service into a process-to-process communication service

- can have multiple application processes on a single host, each with their own port number.
- A process is uniquely addressed by a < port, host > pair
- Common services are available at well-known (and reserved) ports on each host; user applications must choose their ports from the set of non-reserved ports.

•UDP doesn't support flow control or reliable/in-order delivery, but it does support error detection by computing an "optional" checksum over the UDP header, UDP data, and IP pseudo header(includes source and destination address fields from the IP header)

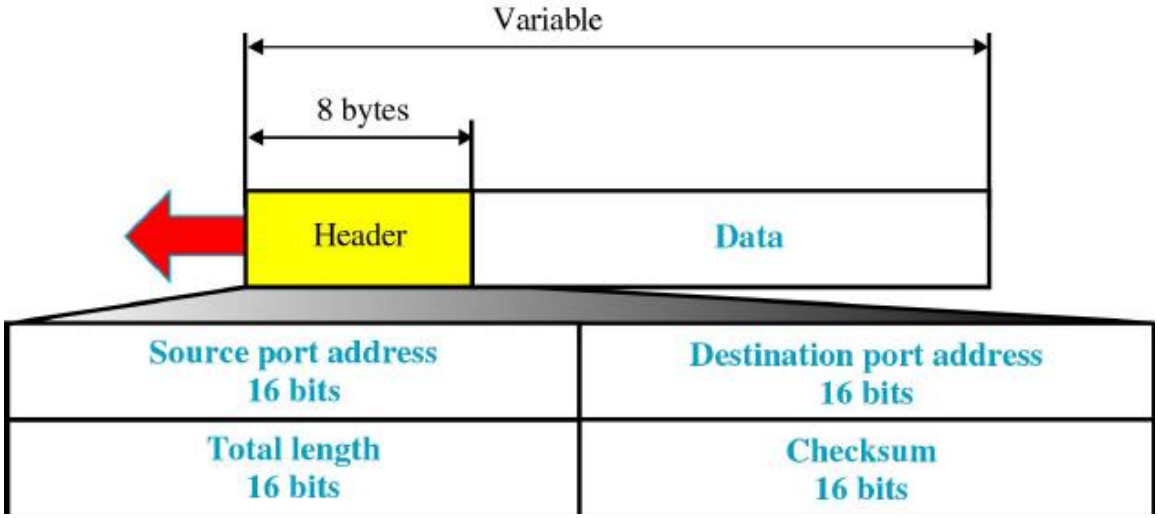
•New: Reliable UDP –provides reliable in-order delivery (up to a maximum number of retransmissions), with simple window flow control, for virtual connections.

Addressing

An address at the transport layer is typically a tuple (Station, Port) where

- Station is the network address of the host, and
- Port identifies the application

UDP Data Packet



- The source port, much like the source port in TCP, identifies the process on the originating system. TCP ports and UDP ports are not the same. There is no relationship between the two.
- The destination port identifies the receiving process on the receiving machine. Whereas the IP address identifies which machine should get the packet, the port identifies which machine should get the data.
- The length field contains the length of the UDP datagram. This includes the length of the UDP header and UDP data. It does not include anything added to the packet in-transit by other protocols -- but these are stripped away before UDP sees the datagram at the other side.

- The checksum field is used by UDP to verify the correctness of the UDP header and data. If the checksum indicates an error, the packet is dropped. UDP is unreliable, so it makes no attempt to mitigate the loss.

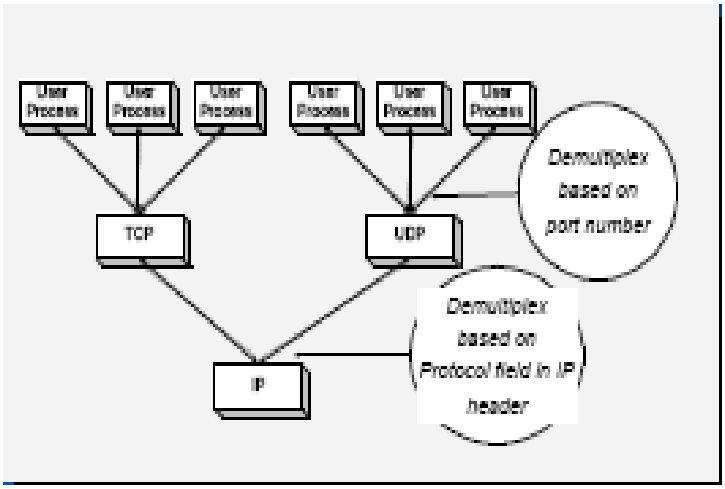
Application

- Datagram oriented
- unreliable, connectionless
- simple
- unicast and multicast
- Useful only for few applications, e.g., multimedia applications
- Used a lot for services

– network management(SNMP), routing (RIP),naming(DNS), etc.

Port Numbers

- UDP (and TCP) use port numbers to identify applications
- A globally unique address at the transport layer (for both UDP and TCP) is a tuple <IP address, port number>
- There are 65,535 UDP ports per host.



TCP: Transmission Control Protocol

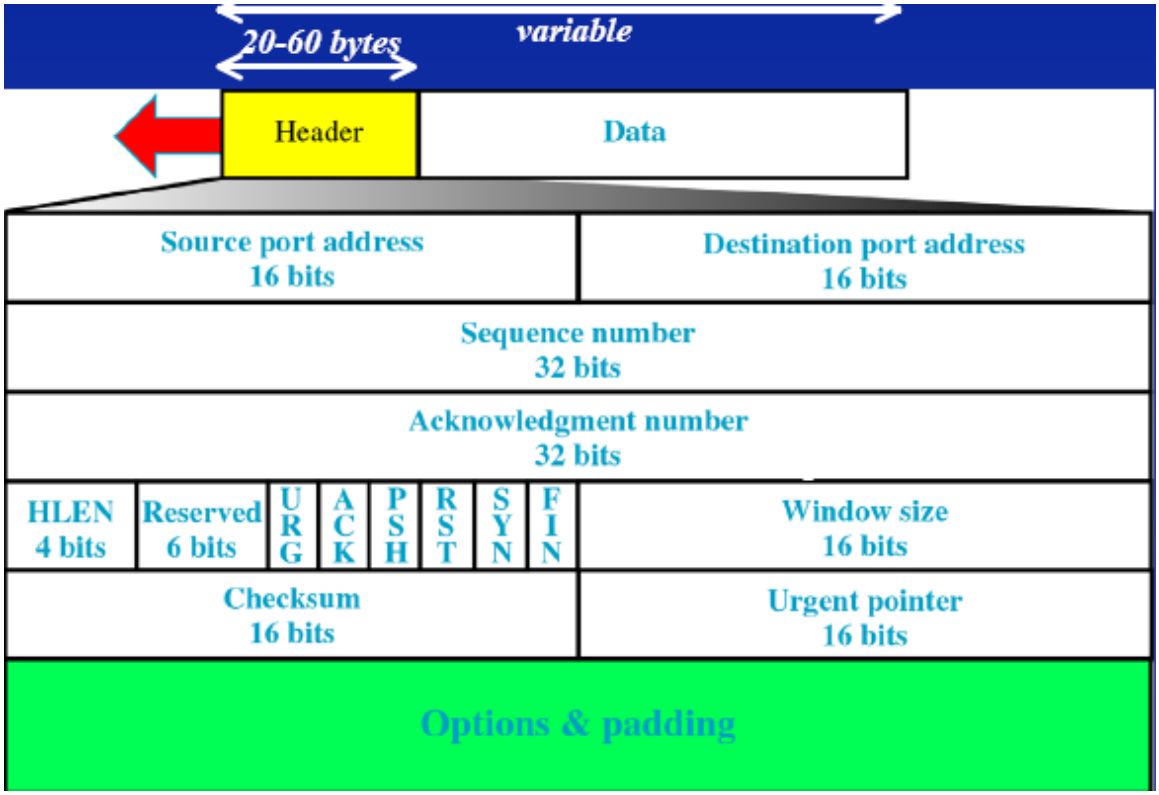
TCP is a reliable, point-to-point, connection-oriented, full-duplex protocol.

Reliable: A reliable protocol ensures that data sent from one machine to another will eventually be communicated correctly. It does not guarantee that this data will be transmitted correctly within any particular amount of

time -- just that given enough time, it will arrive. Life isn't perfect, and it is possible for corrupted data to be thought correct by a reliable protocol -- but the probability of this occurring is very, very, very low Point-to-point: Point-to-point protocols are those protocols that communicate information between two machines. By contrast, broadcast and multicast protocols communicate information from one host to many hosts.

- Connection-oriented :A connection oriented protocol involves a connection or session between the endpoints. In other words, each host is aware of the other and can maintain information about the state of communication between them. The connection needs to be initialized and destroyed. The shared state that is possible with a connection-oriented protocol is essential to a reliable protocol. In particular, the notion of a sequence number or serial number is a practical necessity, if not a theoretical necessity.
- Full-duplex:By full-duplex we mean a mode of communication such that both sides can send and receive concurrently

TCP Data Packet



TCP header fields

Flag bits:

– URG: Urgent pointer is valid
- If the bit is set, the following bytes contain an urgent message in the sequence number range "SeqNo <= urgent message <= SeqNo + urgent pointer"
- ACK: Segment carries a valid acknowledgement
- PSH: PUSH Flag
- Notification from sender to the receiver that the receiver should pass all data that it has to the application.
- Normally set by sender when the sender's buffer is empty
- RST: Reset the connection
- The flag causes the receiver to reset the connection.
- Receiver of a RST terminates the connection and indicates higher layer application about the reset
- SYN: Synchronize sequence numbers
- Sent in the first packet when initiating a connection
- FIN: Sender is finished with sending
- Used for closing a connection
- Both sides of a connection must send a FIN.

TCP Connection Establishment

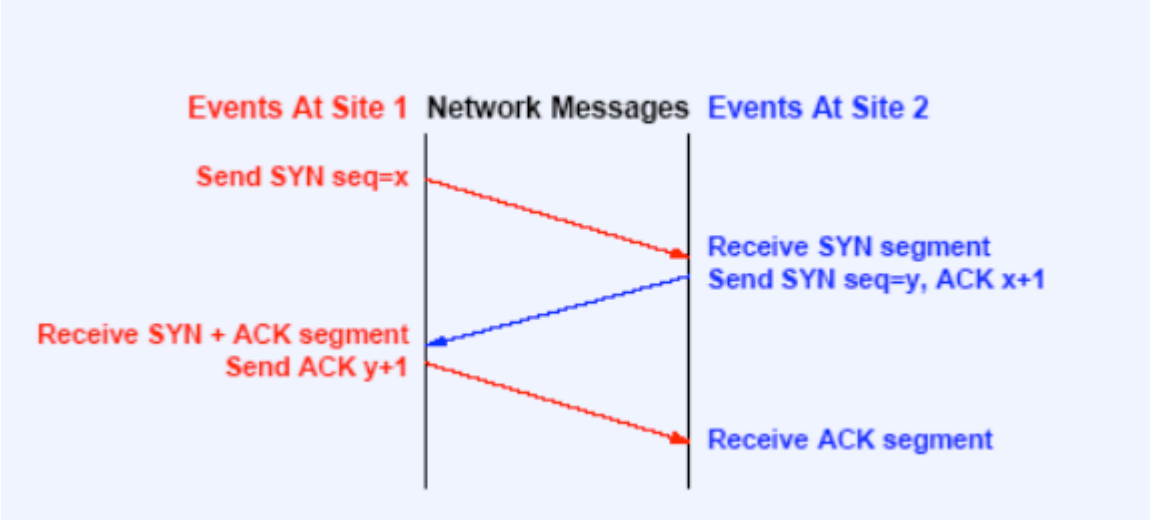
TCP uses a three-way handshake

(1) ACTIVE OPEN: Client sends a segment with
- SYN bit set *
- port number of client
- initial sequence number (ISN) of client

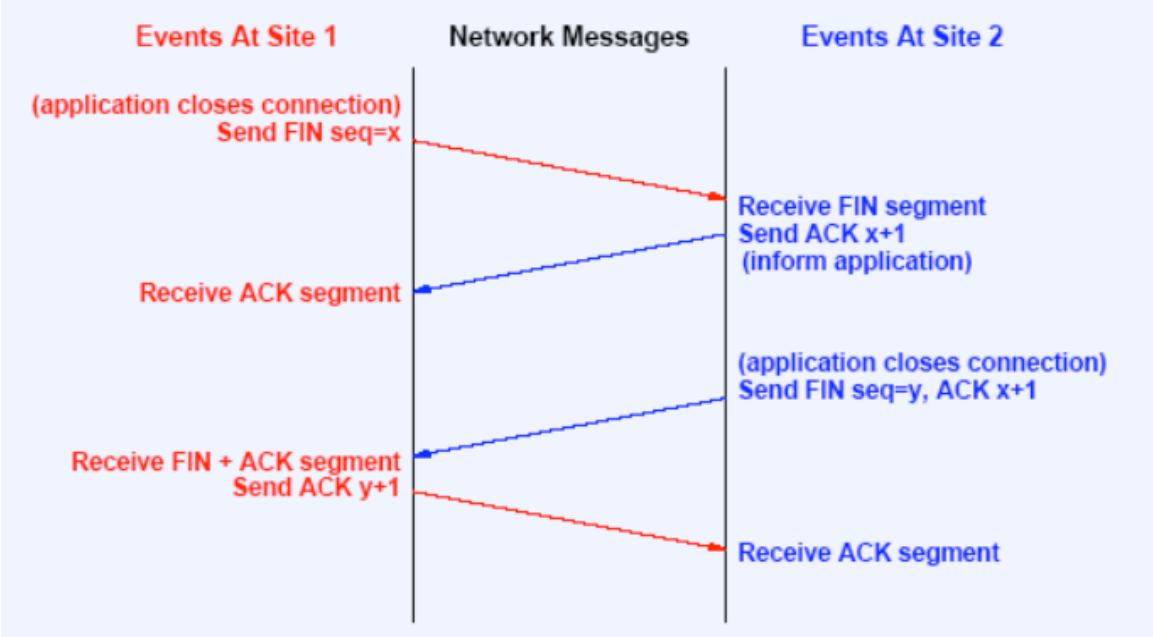
(2) PASSIVE OPEN: Server responds with a segment
- SYN bit set *
- initial sequence number of server
- ACK for ISN of client

(3) Client acknowledges by sending a segment with

- ACK ISN of server (* counts as one byte)



Connection Termination :



Congestion Control

Open Loop Congestion Control

-To prevent congestion before it happens

Retransmission policy :

Good Retransmission policy & Retransmission timer.

Window policy

-Selective Repeat Window.

Acknowledgement policy:

-Does not acknowledge every packet.

Discarding Policy:

-Good discarding Policy.

Admission Policy

-Switches check the resource requirement of flow.

Closed Loop Congestion control

- To alleviate congestion after it happens

Back Pressure(router con):

-Inform the previous upstream router.

Choke point:

-packet sent by router to Source.

Implicit Signaling:

-Source can detect

Explicit Signaling:

-Routers inform sender

Backward Signaling:

-Warn the Source (opp dir)

Forward Signaling:

-Warn the Destination

Slow Start :

Set cwnd size to max. seg size. Increases exponentially.

Additive Increase:

After it reaches threshold increase by 1 seg.If it reaches time-out then multiplicative decrease.

Multiplicative decrease:

Set the threshold to one half of last cwnd size. Each time it is reduced to one half of last cwnd size if a time –out occurs.

Traffic Shaping

- Traffic shaping controls the *rate* at which packets are sent (not just how many)
- At connection set-up time, the sender and carrier negotiate a traffic pattern (shape)
- Two traffic shaping algorithms are:
  - Leaky Bucket
  - Token Bucket

The Leaky Bucket Algorithm

- The **Leaky Bucket Algorithm** used to control rate in a network. It is implemented as a single-server queue with constant service time. If the bucket (buffer) overflows then packets are discarded.
-
- The leaky bucket enforces a constant output rate regardless of the burstiness of the input. Does nothing when input is idle.
- The host injects one packet per clock tick onto the network. This results in a uniform flow of packets, smoothing out bursts and reducing congestion.
- When packets are the same size (as in ATM cells), the one packet per tick is okay. For variable length packets though, it is better to allow a fixed number of bytes per tick.

Token Bucket Algorithm

- In contrast to the LB, the Token Bucket (TB) algorithm, allows the output rate to vary, depending on the size of the burst.
- In the TB algorithm, the bucket holds tokens. To transmit a packet, the host must capture and destroy one token.
- Tokens are generated by a clock at the rate of one token every Δt sec.
- Idle hosts can capture and save up tokens (up to the max. size of the bucket) in order to send larger bursts later.

Token bucket operation

- TB accumulates fixed size tokens in a token bucket
- Transmits a packet (from data buffer, if any are there) or arriving packet if the sum of the token sizes in the bucket add up to packet size
- More tokens are periodically added to the bucket (at rate Δt). If tokens are to be added when the bucket is full, they are discarded

Token bucket properties

- Does not bound the peak rate of small bursts, because bucket may contain enough token to cover a complete burst size

- Performance depends only on the sum of the data buffer size and the token bucket size

Domain Name System

Introduction

- A sys that can map a name to an address or an add to a name.
- Mapping was done using a host file

It has 2 columns

Name and address

- Every host could store the host file on its disk and should be updated from master file.
- If a program or a user wanted to map a name to an add. ,host consulted the host file and found mapping

- Divide the huge amt of info into smaller parts
- Store each part on a different computer
- The host that needs mapping can contact the closest computer holding the needed information
- i.e. DNS

Name Space

- Names assigned to machines must be selected from name space with control over the binding between names and IP addresses.
- A name space that maps each address to a unique name can be organised in two ways.

Flat Name Space

Hierarchical Name Space

Flat Name Space :

- ***Name is assigned to an address.***
- ***Name in this space is a sequence of characters without structure.***

Demerit:

- ***Can't be used in large system such as Internet***
- ***It must be centrally controlled to avoid ambiguity and duplication***

Hierarchical Name Space

- Each name is made of several parts
- First part – nature of the organization
- Second part – Name
- Third part – department
- So the authority to assign and control the name space can be decentralized

- Suffixes can be added to the name to define the host or resources
- ***To have a hierarchical name***

Space, DNS was designed

- ***Names are defined in an inverted tree structure with root at the top.***
- ***Tree can have only 128 levels***

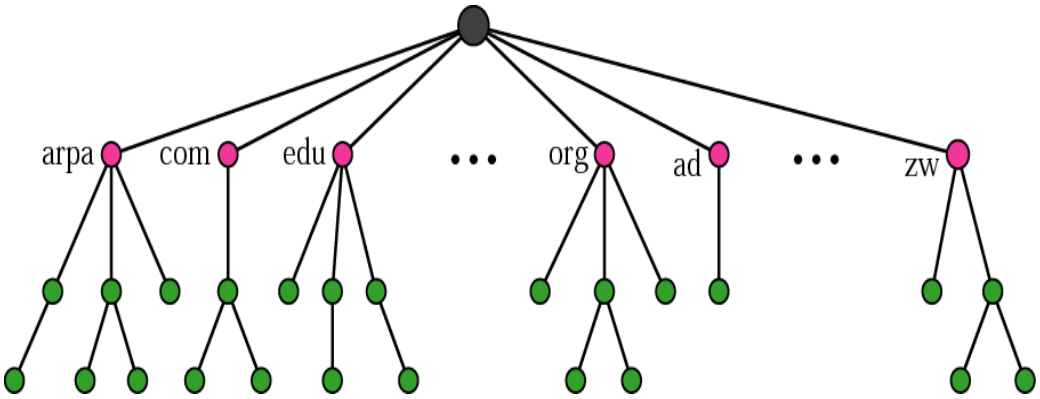
Label :

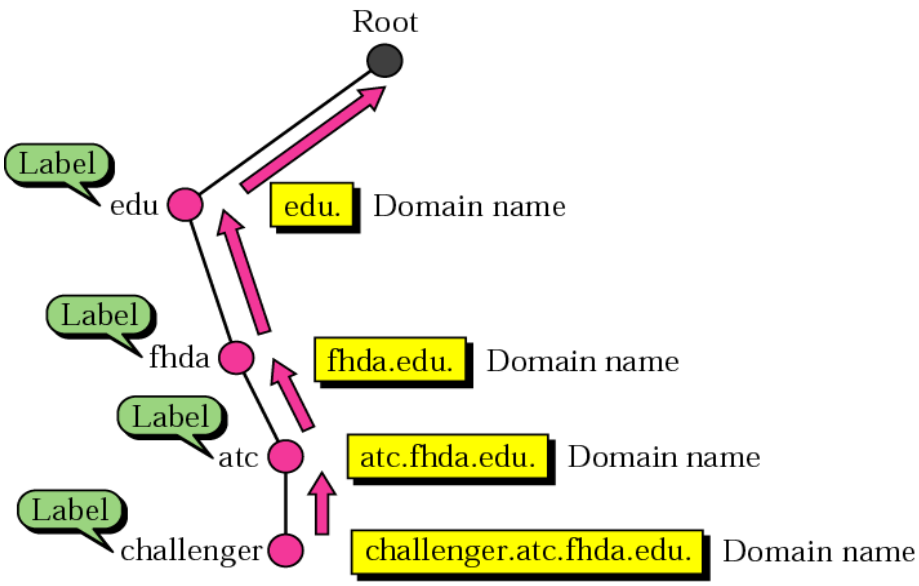
Each node in the tree has a label(a string with a maximum of 63 characters)

- Root label is a null string
- Children of a node have different labels which guarantees the uniqueness of domain names.

Domain Name:

- Each node in the tree has a domain name.
- DN is a sequence of labels separated by dots
- Always read from the node up to the root





FQDN

- If label is terminated by a null string it is called a FQDN
- A domain name that contains the full name of a host
- Contains all labels from specific to general
- Uniquely define the name of the host.

challenger.atc.fhda.edu.

PQDN

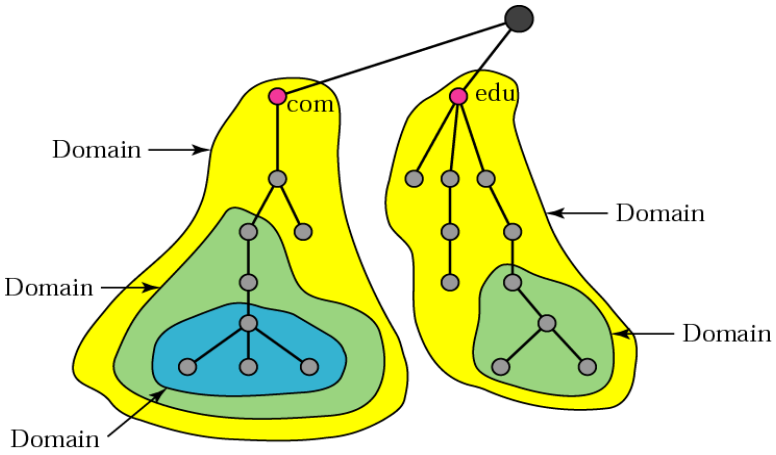
- If label is not terminated by a null string it is called PQDN.
- Starts from a node but does not reach the root.
- Used when the name to be resolved belongs to the same site as client.
- Resolver can supply the missing part called suffix ,to create an FQDN.

- DNS client adds suffix atc.fhda.deu before passing the address to the DNS server.

Domain:

- A domain is a sub-tree of the domain space.
- Domain may itself be divided into sub domains.

FQDN	PQDN
challenger.atc.fhda.edu. cs.hmme.com. www.funny.int.	challenger.atc.fhda.edu cs.hmme www

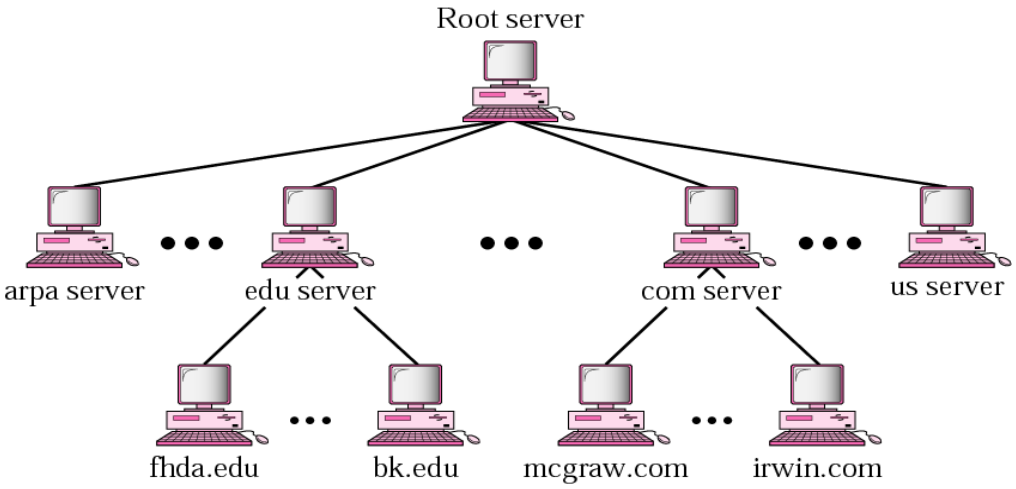


Distribution of Name Space

- The info contained in the domain name space must be stored.
- Not reliable to have info in 1 computer.
- So distribute the information among many computers called servers

Hierarchy of Name Servers

- *Distribute the info among many computers called DNS servers*
- *Divide the whole space into many domains based on the first level.*
- *Let the root stand alone and create as many domains as there are*
- Allows domains to be divided further into smaller domains
- Each server can be responsible for either a large or a small domain



Zone

- What a Server is responsible for or has authority over is called zone.
- If a server accepts responsibility for a domain and does not divide the domain into smaller domains.
- The domain and the zone refer to the same thing.
- Server makes a database called a zone file.
- It keeps all the information for every node under that domain

Root server

- Is a server whose zone consists of whole tree

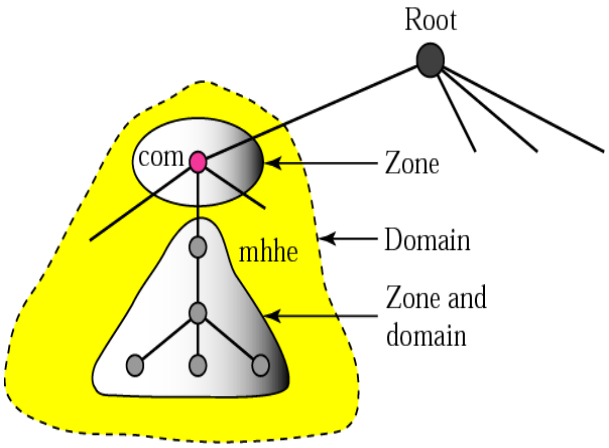
- It does not store any info about domain but delegates authority to other servers

Primary server:

- A server that stores a file about the zone for which it is an authority.
- Responsible for creating, maintaining the and updating the zone file
- It stores the zone file on a local disk

Secondary server

- A server that transfers the complete information about a zone from another server and stores the file on its local disk
- It neither creates nor updates the zone files.
- Updating is done by a primary server, which sends the updated version to secondary



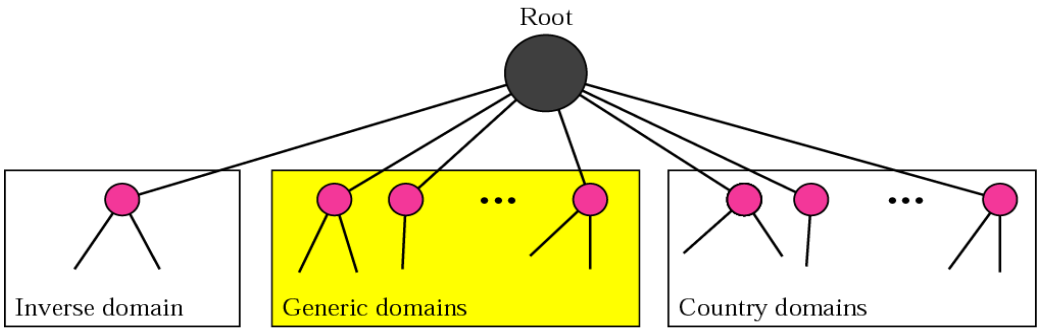
A primary server loads all information from the disk file; the secondary server loads all information from the primary server

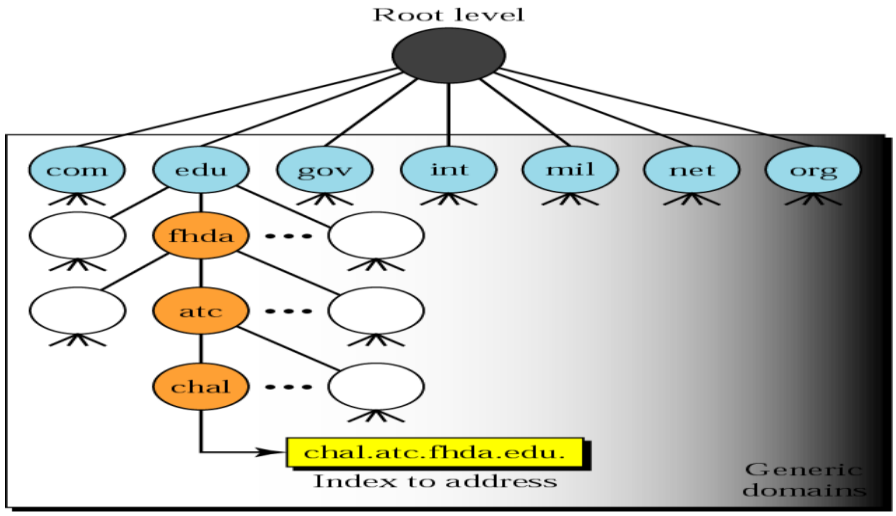
DNS In The Internet

- In the Internet, Domain name space (tree) is divided into 3 sections.

Generic Domain

- It defines registered hosts according to their generic behavior.
- Each node in the tree defines a domain, which is an index to the domain name space database.
- Seven labels describe three organization types.



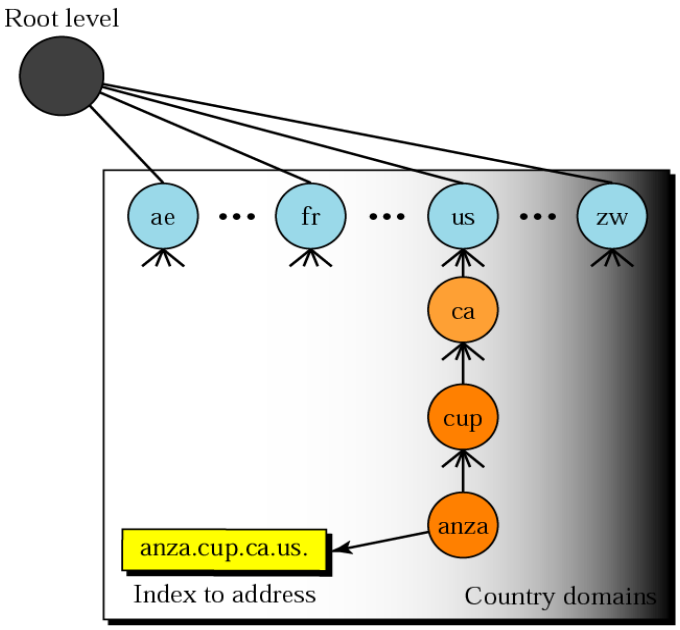


Generic Domain Labels

Label	Description
com	Commercial organizations

edu	**Educational institutions**
gov	**Government institutions**
int	**International organizations**
mil	**Military groups**
net	**Network support centers**
org	**Nonprofit organizations**

Country Domain
Follows the same format as the generic domain but uses two character country abbreviations.



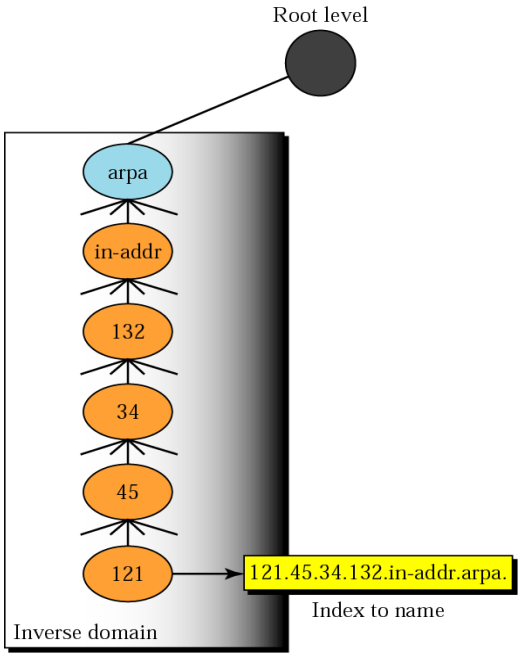
Inverse domain

It is used to map an address to a name

Ex:

- When a server has received a request from a client to do a task
- Whereas the server has a file that contains a list of authorized clients, the server lists only the IP address of the client

- To determine if the client is on the authorized list ,server send a query
- To inverse DNS server and ask for a mapping of address to a name
- This query is called inverse or pointer (PTR) query
- To handle this ,inverse domain is added to the domain space with the first level node called arpa
- Second level is also one single node named in-addr
- Rest of the domain defines the IP address.



Resolution

- Mapping a name to an address or an address to a name is called name-address resolution.

Resolver

- A host that needs to map an address to a name or a name to an address calls a DNS client named a resolver.
- It accesses the closest DNS server with a mapping request
- If the server has the information , it satisfies the resolver.
- Otherwise it refers the resolver to other servers or ask other servers to provide information.
- After the resolver receives the mapping ,it interprets to see if it is a real resolution or an error and finally delivers the result to the process that requested it.

Mapping Names to addresses

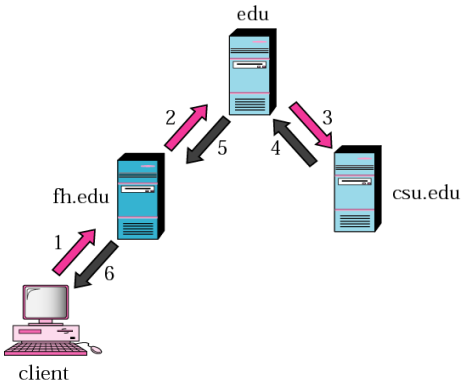
- The resolver gives a domain name to the server and asks for the corresponding address
- In this ,server checks the generic domain or the country domain to find the mapping
- If from the generic domain the resolver receives a domain name such as chal.atc.fhda.edu.

- Query is sent by the resolver to the local DNS server for resolution
- If cant refers the resolver to other servers or ask other servers directly
- If from the country domain, the resolver receives a domain name such as ch.fhda.cu.ca.us.
  Mapping addresses to names
- Client can send an IP address to a server to be mapped to a domain name – called PTR query
- To answer this uses inverse domain
- In the request IP address is reversed and 2 labels in-addr & arpa are appended to create a domain acceptable by the inverse domain section
- 132.34.45.121 ,121.45.34.132.in-addr.arpa.

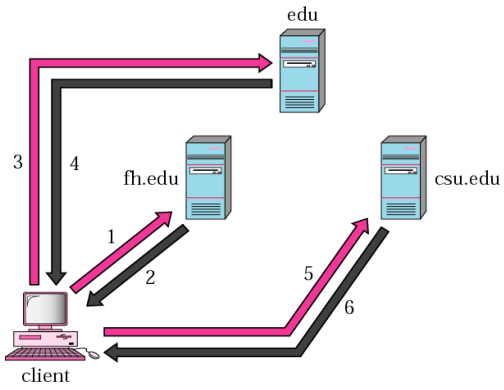
  Recursive Resolution

- The resolver expects the server to supply the final answer
- If server is the authority for the domain name ,it checks the database and responds
- If not the authority ,sends the request to another server (parent) and waits for response

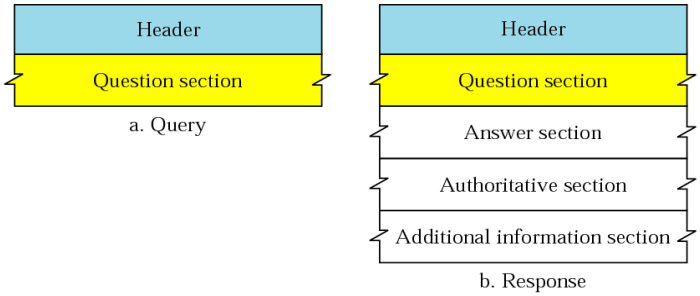
- If the parent is the authority respond otherwise sends the query to yet another server
- If resolved, response travels back until it reaches the requesting client
- This is recursive resolution



Iterative Resolution



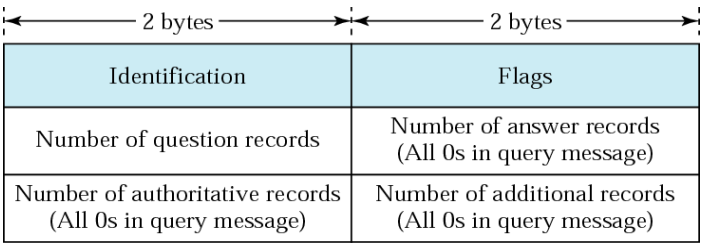
DNS Messages



Header format

- Identification
  Used by the client to match the response with the query.
  Uses a diff id no. each time it sends a query.
  Server duplicates this no. in the corresponding response.
- Flag

Collection of fields that define the
Type of msg
Type of answers requested
Type of desired resolution (recursive or iterative)



No. of question records
Contains the number of queries in the question section of the msg
No. of answer records
Contains the number of answer records in the answer section of the response msg. value is zero in the query msg
No. of authoritative records
- Contains the number of authoritative records in the authoritative section of a response msg
- value is zero in query msg

No. of additional records
- Contains the number of additional records in the additional section of a response msg
- value is zero in query msg

Question Section
- Consist of one or more question records
- Present on both query and response msg

Answer Section
- Consist of one or more resource records
- Present only on response msg
- It includes the answer from the server to the client (resolver)

Authoritative Section
- Consist of one or more resource records
- Present only on response msg
- It gives info (domain name) about one or more authoritative servers for the query

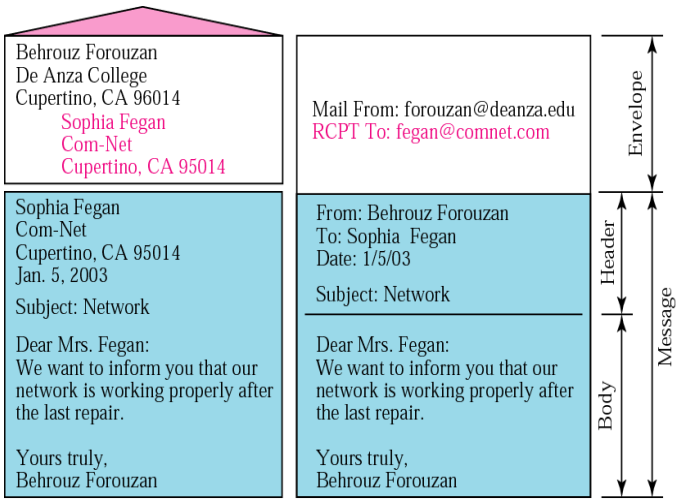
Additional Information Section
- Consist of one or more resource records
- Present only on response msg
- It gives additional info (domain name) that help the resolver

DNS can use the services of UDP or TCP, using the well-known port 53.

SMTP

Mail : Exchanges info between people

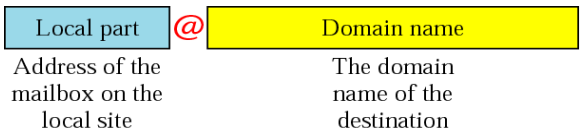
Format of an email



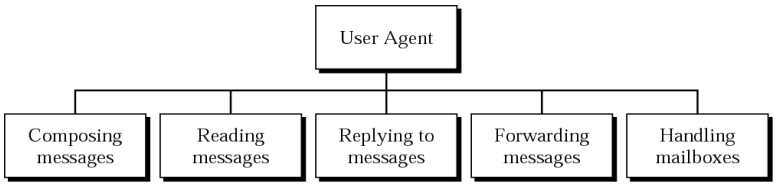
Addresses

- To deliver mail, an addressing system used by SMTP consists of 2 parts
- Local part : defines the name of a specific file called mail box
- All the mail received for a user is stored in the mail box for retrieval by the user agent
- Domain Name : comes from the DNS database or is a logical name (name of the organization)

Email address

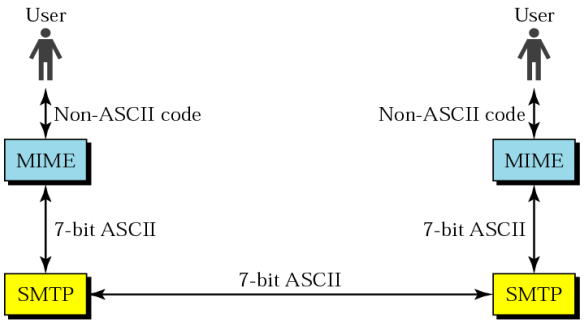


User agent
A s/w package that composes, reads, replies to, and forward messages.

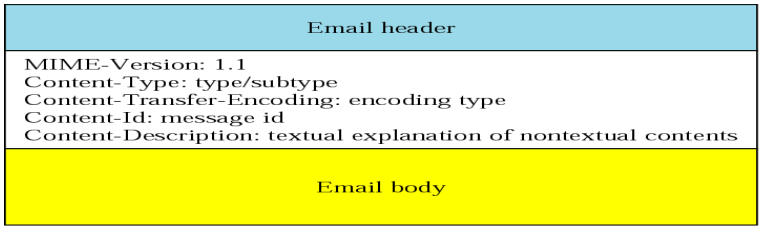


Some examples of command-driven user agents are mail, pine, and elm
Some examples of GUI-based user agents are Eudora, Outlook, and Netscape.

MIME:
It converts a Non-ASCII code to ASCII code.



MIME Header:



Data types and subtypes in MIME

Type	Subtype	Description
Text	**Plain**	**Unformatted text**
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**
	Ext. Body	**Body is a reference to another message**
Image	**JPEG**	**Image is in JPEG**
	GIF	**Video is in GIF format**

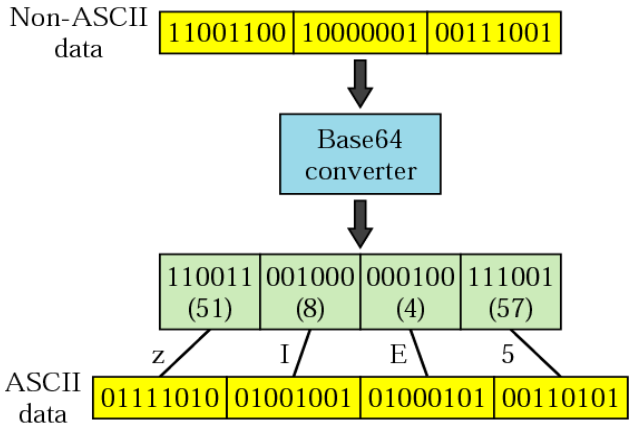
Video	**MPEG**	**Video is in MPEG format**
Audio	**Basic**	**Single-channel encoding of voice at 8 KHz**
Application	**PostScript**	**Adobe PostScript**
	Octet-Stream	**General binary data (8-bit bytes)**

Content-transfer encoding

Category	Description
Type	**ASCII characters and short lines**
7bit	**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 followed by an ASCII code.

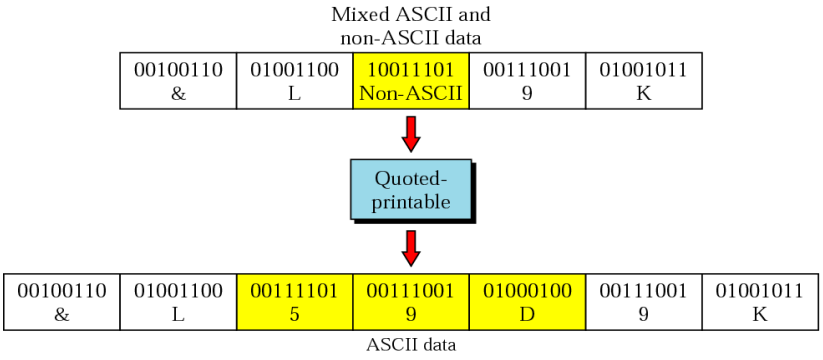
Base64



Base64 encoding table

Value	Code	Value	Code	Value	Code	Value	Code	Value	Code	Value	Code
0	A	11	L	22	W	33	h	44	s	55	3
1	B	12	M	23	X	34	i	45	t	56	4
2	C	13	N	24	Y	35	j	46	u	57	5
3	D	14	O	25	Z	36	k	47	v	58	6
4	E	15	P	26	a	37	l	48	w	59	7
5	F	16	Q	27	b	38	m	49	x	60	8
6	G	17	R	28	c	39	n	50	y	61	9
7	H	18	S	29	d	40	o	51	z	62	+
8	I	19	T	30	e	41	p	52	0	63	/
9	J	20	U	31	f	42	q	53	1	□カラットlog□좋□琰莒□ □Ü	□カラットlog□좋□琰
10	K	21	V	32	g	43	r	54	2	□カラットlog□좋□琰莒□ □Ü	□カラットlog□좋□琰

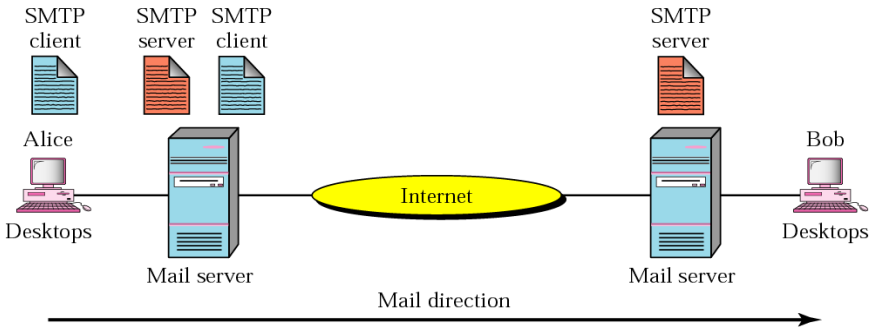
Quoted Printable:



MTA

- Actual mail transfer is done through MTA
- To send mail a system must have a client MTA and to receive a mail a server MTA
- Mail transfer occurs between two mail servers

MTA client and server



Commands and responses

- Uses commands and responses to transfer messages between an MTA client and an MTA server
- Command or reply is terminated by a two character end–of–line token

Commands

- Sent from client to server
- Consist of a keyword followed by zero or more arguments

Responses

- Sent from server to the client
- Response is a three digit code that may be followed by additional textual information

Mail transfer

- Transferring a mail message occurs in 3 phases

Connection establishment

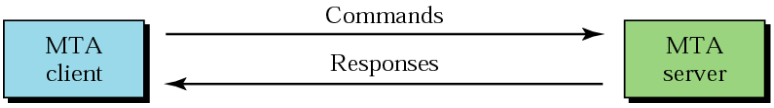
- After a client has made a TCP connection to the well known port 25 ,SMTP server starts the connection phase.

Message transfer

Message between a sender and one or more recipients can be exchanged.

- Connection Termination

After the message is transferred ,the client terminates the connection



Mail Delivery

Consists of 3 stages

Ist stage

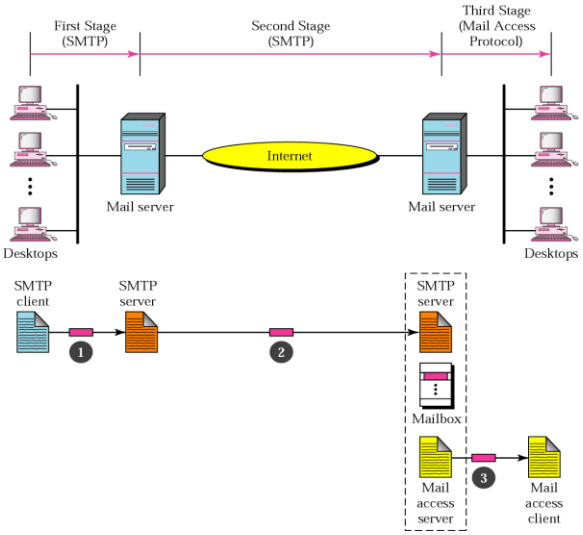
- Email goes from user agent to the local server.
- Mail does not go directly to the remote server.
- Mail is stored in the local server until it can be sent.
- User agent uses SMTP client s/w and the local server uses SMTP server s/w.

Second stage

- Email is relayed by local server, which now acts as SMTP client to the remote server, which is the SMTP server in this stage
- Email is delivered to the remote server ,not to the remote user agent

Third stage

- The remote user agent uses a mail access protocol such as POP3 or IMAP4 to access the mailbox and the mail



Mail Access Protocols

- SMTP is involved in the first and second stages but not in third stage, since it is a push protocol (pushes msgs from sender to receiver).
- The third stage needs a pull protocol
- Operation must start with the recipient
- Mail must stay in the mail server mailbox until the recipient receives it
- Third stage uses a mail access protocol(POP3,IMAP4)

POP3

- Simple but limited in functionality
- Mail access starts with the client when the user needs to download email from the mailbox on the mail server
- Client (user agent opens a connection with the server on TCP port 110.
- It sends its user name and password to access the mailbox
- User can then list and retrieve the mail messages one by one
- POP3 has two modes
  Delete mode& Keep mode

Delete mode

- Mail is deleted from the mail box after each retrieval

- Normally used when the user is working at permanent computer and save and organize the received mail after reading or replying

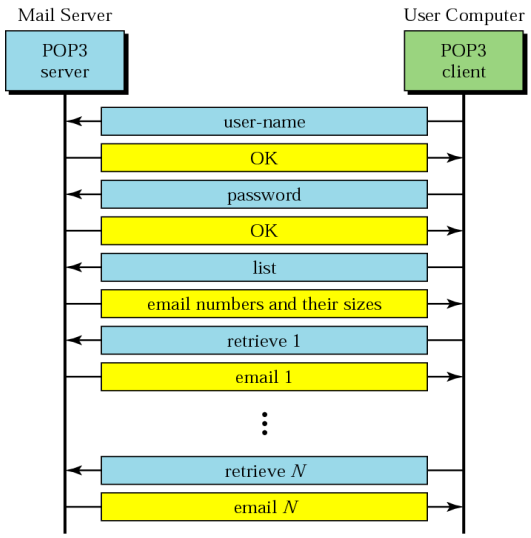
Keep mode

- Normally used when the user accesses mail away from primary computer. Mail is read but kept in the system for later retrieval and organizing.

- Assumes that each time a client accesses the server, the whole mailbox will be cleared out
- Not convenient when access their mailboxes from different clients (home or hotel)

IMAP4

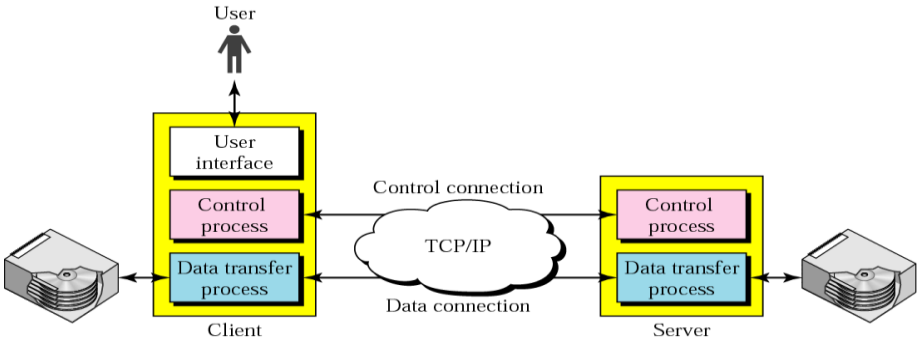
- Powerful and more complex.
- User can check the email header prior to downloading.
- User can check the contents of email for a specific string of characters prior to downloading.
- Can partially download email.
- User can create, delete or rename mailboxes on the mail server.
- Can create a hierarchy of mailboxes in a folder for email storage.



FTP

- For copying a file from one host to another

FTP uses the services of TCP. It needs two TCP connections. The well-known port 21 is used for the control connection, and the well-known port 20 is used for the data connection



- When a user starts an FTP session control connection opens
- While the control connection is open, the data connection can be opened and close multiple times if several files are transferred
- Conn remains open during the entire process
- Service type used is minimize delay
- User types commands and expects to receive responses without significant delay

Data connection

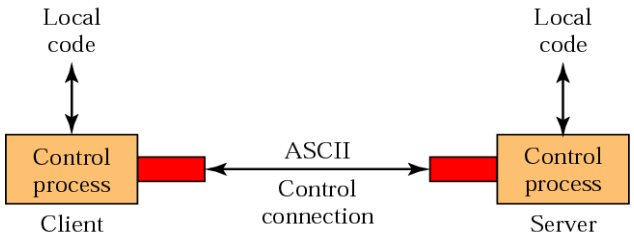
- Uses the well known port 20 at the server site
- Connection open when data ready to transfer
- Closed when it is not needed
- Service type used is maximize throughput

Communication

- FTP client and server run on different computers
- Must communicate with each other
- May use different operating system, diff character sets, diff file structures and diff file formats
- FTP make this compatible
- FTP has 2 diff approaches. one for ctrl conn & the other for data conn

Communication over ctrl conn

- Same approach as SMTP
- Uses the ASCII character set.
- Communication is achieved through commands and responses
- Each line is terminated with a two-character (carriage return and line feed) end-of-line token



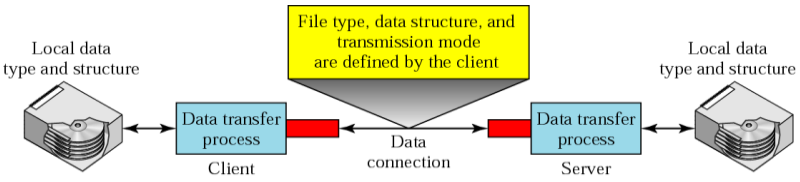
Communication over data conn

- To transfer files through data conn, client must

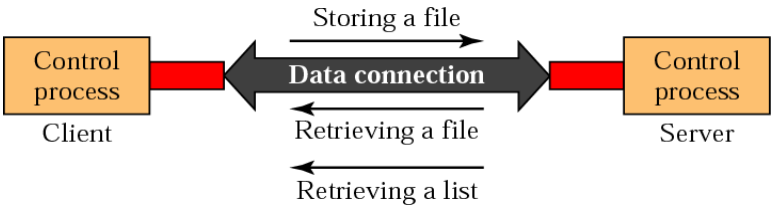
- Define the type of file
- Structure of the data
- Transmission mode

Heterogeneity solved by
File type ,data structure and transmission mode

Using the data connection



File transfer

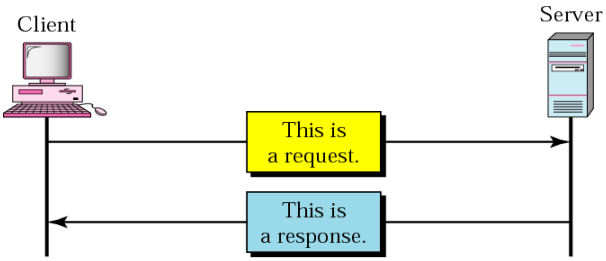


HTTP

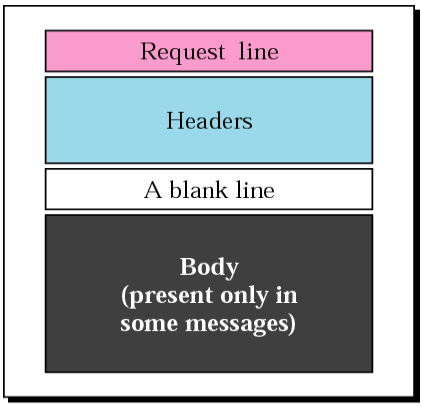
- Used mainly to access data on the www
- The protocol transfers data in the form of plain text, hyper text, audio and video and so on.
- A client sends a request ,which looks like mail to the server
- The server sends the response which looks like a mail reply to the client

The request and response messages carry data in the form of a letter with a MIME-like format

HTTP uses the services of TCP on well-known port 80.



Request Message



Request message

Request Line:

Request msg

Request type:

Several request types are defined

RT categorizes the request msgs into several methods

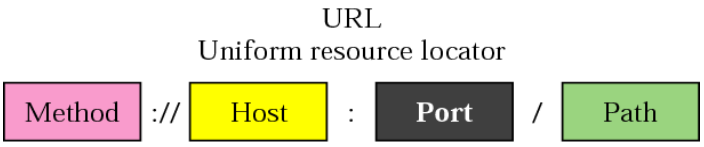
URL

A client that wants to access a web page needs an address.

To facilitate access of documents ,it uses URL.

It defines 4 things

- Method: a protocol used to retrieve the document (FTP and HTTP)
- Host : a computer where info is located
- Port number of server
- Path name of file where info is located
- Current version is HTTP 1.1



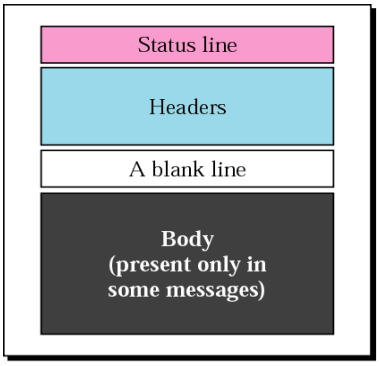
Methods

- Request type defines several kinds of messages referred as methods
- Request method is the actual command or request that a client issues to the server
- GET :if the client wants to retrieve the document from the server

- HEAD: if the client wants some info about the document
- POST: used by the client to provide some info to the server

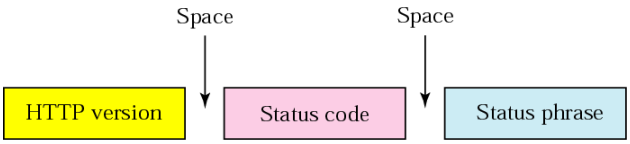
- PUT: used by the client wants to provide a new or replacement document to be stored on the server
- PATCH : similar to PUT with some differences to be implemented with the existing file.
- COPY : copies a file to another location
- MOVE: moves a file to another location
- DELETE : removes a document on the server
- LINK : creates a link or links from a doc to another location
- UNLINK: deletes the link created by the link

Response Message:

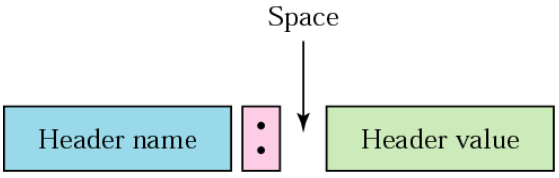


Response message

Status Line:



Header Format:



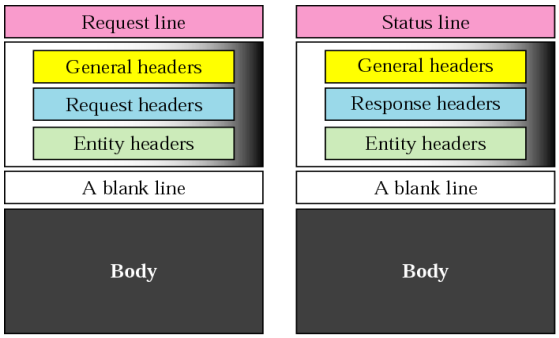
- Exchange additional info between the client and the server
- Example:

Client can request that the doc be sent in a special format
Server can send extra info about the document

- General header: gives info about the msg
- Request Header: can only be in response msg

Specifies the servers configuration and special info about the request

- Entity header :gives info about the body of the document



Other features

- Persistent Connection: the server leaves the connection open for more requests after sending a response.
- Non-Persistent Connection: one TCP connection is made for each request and response.

HTTP version 1.1 specifies a persistent connection by default

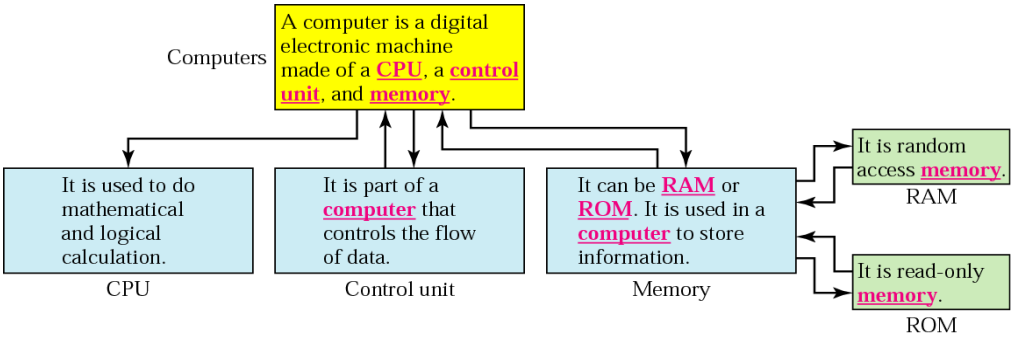
WWW

- Repository of info spread all over the world and linked together
- It has a unique combination of flexibility, portability and user-friendly features .
- It is a distributed client-server service.
- A client using a browser can access a service using a server.
- The service provided is distributed over many locations called websites.

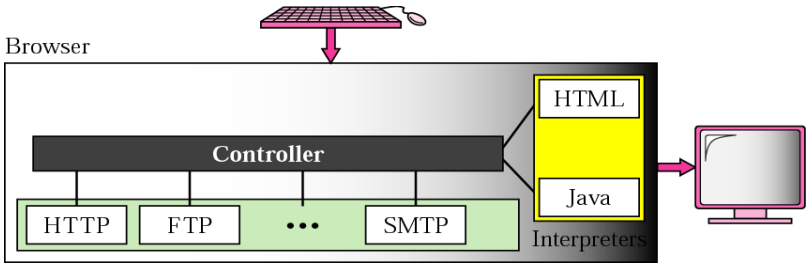
Hypertext and hypermedia

- Info is stored in a set of documents that are linked using the concept of pointers
- An item can be associated with another document by a pointer
- Hypermedia: It can contain pictures , graphics and sound
- A unit of Hypertext or hypermedia available on the web is called a page

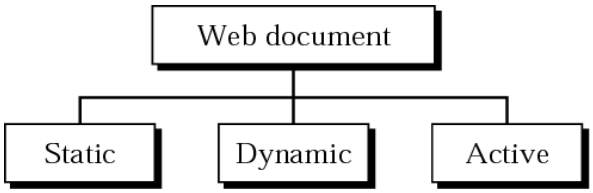
Hypertext:



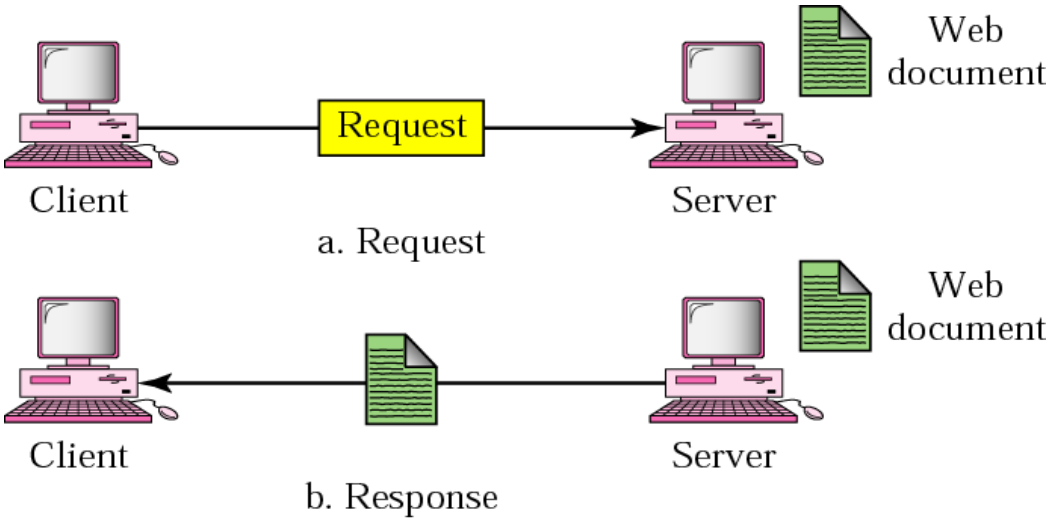
Browser Architecture:



Categories of Web documents



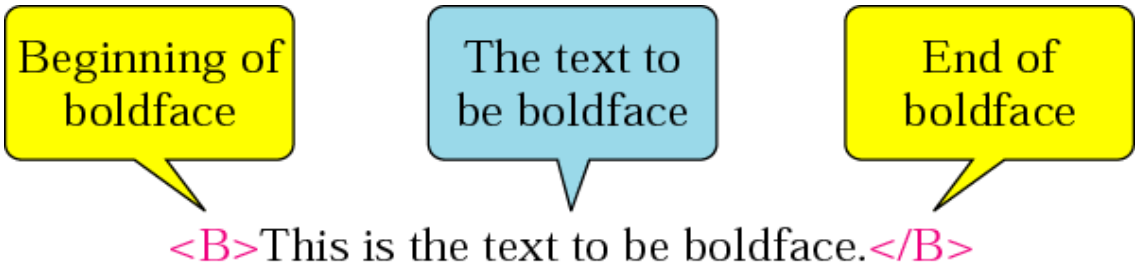
Static Document :



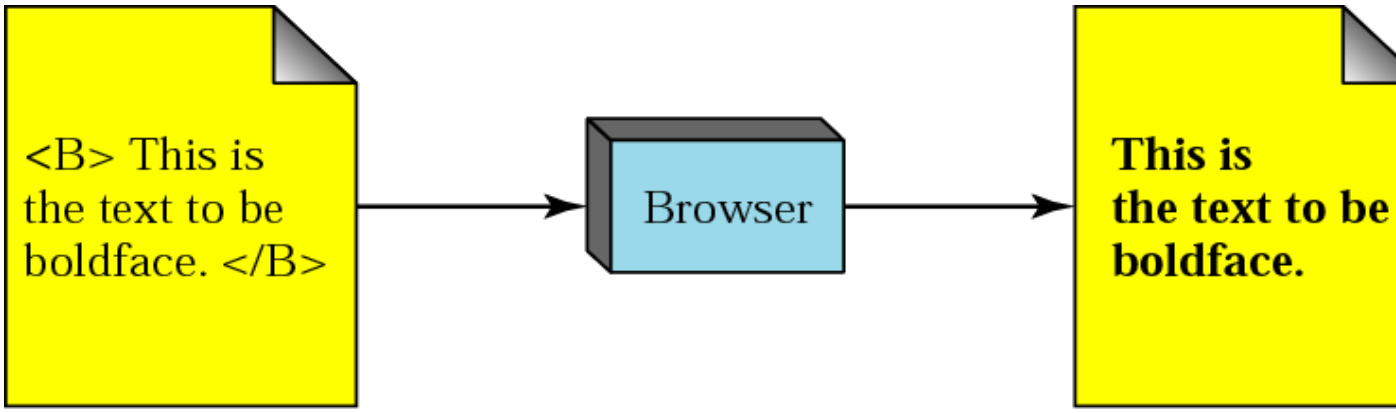
HTML

- A language for creating web pages.
- Allows to embed formatting instructions in the file itself.

Boldface tags



Effect of boldface tags



Structure of a Web page

- A web page is made of 2 parts
- Head and Body
- Head : contains the title of the page

Beginning and ending tags

a. Beginning tag

b. Ending tag

Common tags

Beginning Tag	Ending Tag	Meaning

Skeletal Tags		
<HTML>	</HTML>	Defines an HTML document
<HEAD>	</HEAD>	Defines the head of the document
<BODY>	</BODY>	Defines the body of the document
Title and Header Tags		
<TITLE>	</TITLE>	Defines the title of the document
<Hn>	</Hn>	Defines the title of the document

Beginning Tag	Ending Tag	Meaning
Text Formatting Tags		
<B>	</B>	Boldface
<I>	</I>	Italic
<U>	</U>	Underlined
_		Subscript
[	]	Superscript
Data Flow Tag		
<CENTER>	</CENTER>	Centered
 	</BR>	Line break

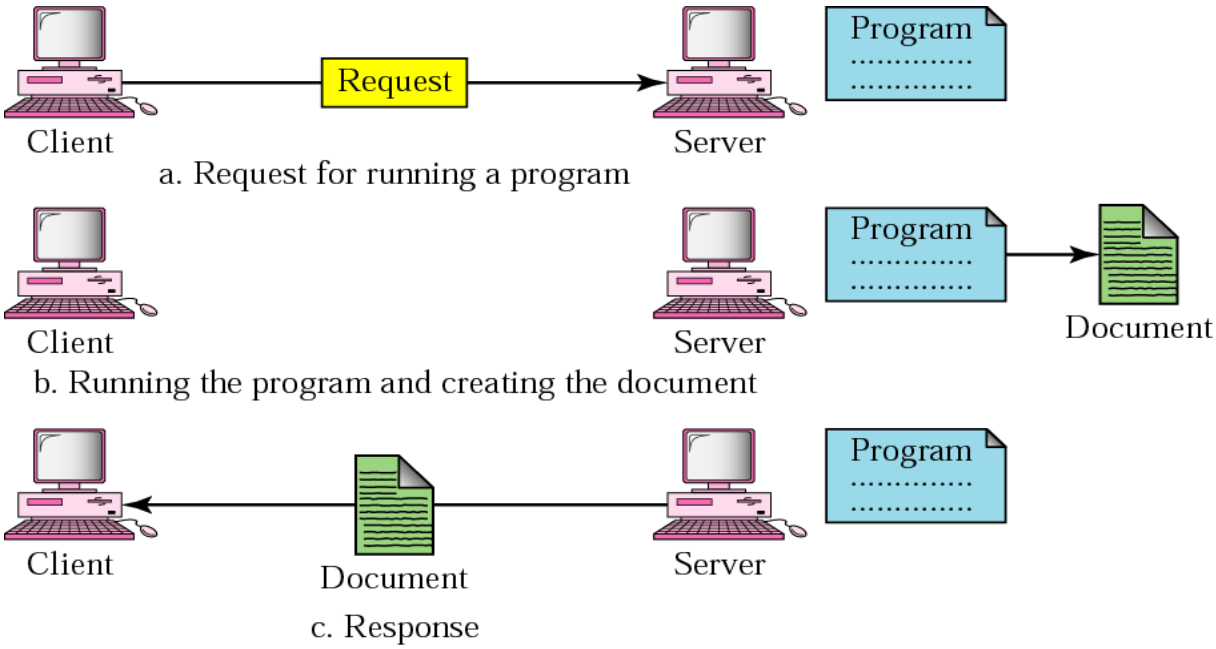
Beginning Tag	Ending Tag	Meaning
List Tags		
<OL>	</OL>	Ordered list
<UL>	</UL>	Unordered list

<LI>	**</LI>**	**An item in a list**
Image Tag		
<IMG>	□㌍log□좈□琰茞□ □ Ů	**Defines an image**
Hyperlink Tag		
<A>	**</A>**	**Defines an address (hyperlink)**
Executable Contents		
<APPLET>	**</APPLET>**	**The document is an applet**

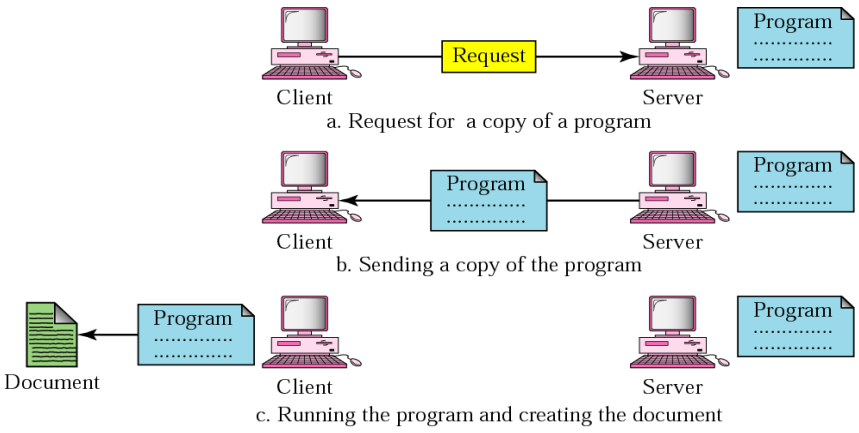
This example shows how tags are used to let the browser format the appearance of the text

Dynamic Document

- Do not exist in a predefined format
- It is created by a web server whenever a browser requests the document



Active document

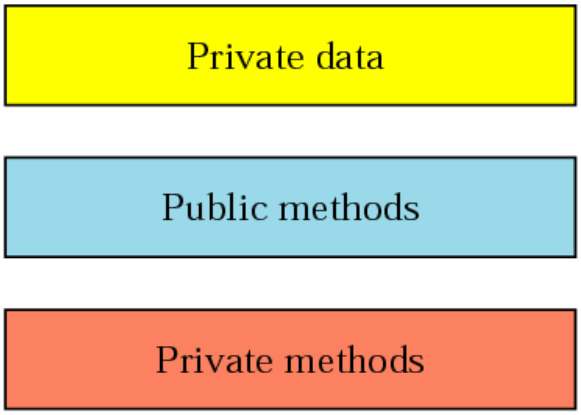


Skeleton of an applet

Import libraries

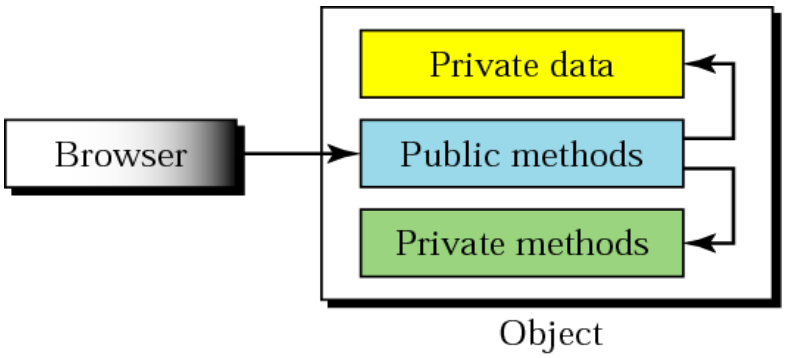
public class name extends applet

{

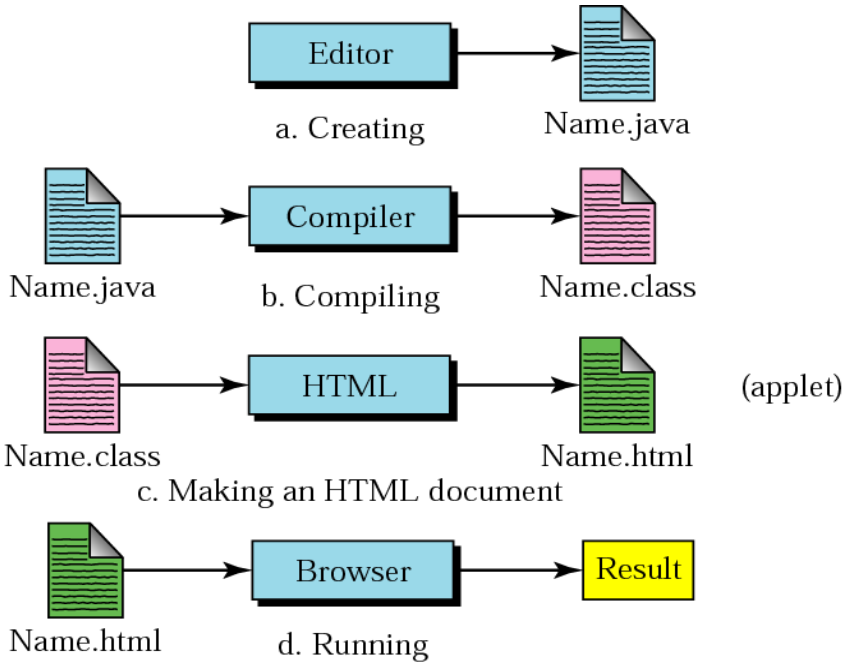


}

Instantiation of the object defined by an applet



Creation and compilation



HTML document carrying an applet

```
<HTML>
    <APPLET   CODE 5 "Name.class"
    WIDTH 5 mmm
    HEIGHT 5 nnn  ....  >
    </APPLET >
</HTM>
```

In this example, we first import two packages, java.awt and java.applet. They contain the declarations and definitions of classes and methods that we need. Our example uses only one publicly inherited class called First. We define only one public method, paint. The browser can access the instance of First through the

public method paint. The paint method, however, calls another method called drawString, which is defined in java.awt.*.

```
import java.applet.*;
import java.awt.*;

public class First extends Applet
{
      public void paint (Graphics  g)
       {
         g.drawString ("Hello World", 100, 100);
       }
}
```

UNIT II

DATA LINK LAYER

ERROR DETECTION AND CORRECTION

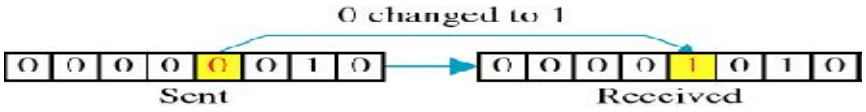
ERROR:

Data can be corrupted during transmission. For reliable communication, errors must be detected and corrected. Signals flows from one point to another. This is subjected to unpredictable interferences from heat, magnetism and other forms of electricity.

TYPES OF ERRORS:

- **Single bit Error:**

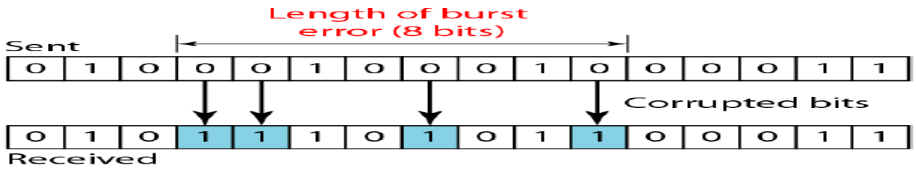
  The term single bit error means that only one bit of a given data unit is changed from 1 to 0 or 0 to 1. 010101 is changed to 110101 here only one bit is changed by single bit error.



- **Burst Error:**

  A burst error means that 2 or more bits in the data unit have changed.

  Example:



Three kinds of errors can occur:

- the bits in the frame can be inverted, anywhere within the frame including the data bits or the frame's control bits,
- additional bits can be inserted into the frame, before the frame or after the frame and
- Bits can be deleted from the frame.

DETECTION

Redundancy

Error detection use the concept of redundancy, which means adding extra bits for detecting errors at the destination .i.e., instead of repeating the entire data stream, a shorter group of bits may be appended to the end of each unit.

- To detect or correct errors, we need to send extra (redundant) bits with data.
- The receiver will be able to detect or correct the error using the extra information.
- Detection
  - Looking at the existence of any error, as YES or NO.
  - Retransmission if yes. (ARQ)
- Correction
  - Looking at both the number of errors and the location of the errors in a message.
  - Forward error correction. (FEC)

Coding

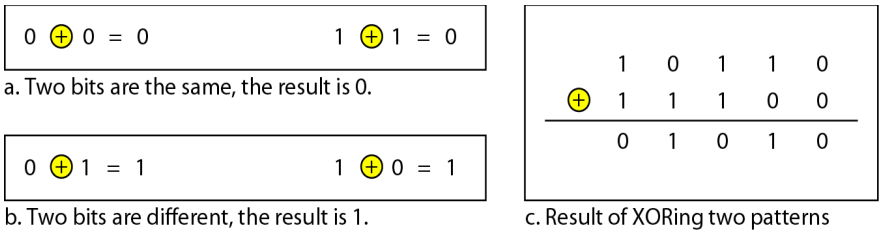
- Encoder vs. decoder
- Both encoder and decoder have agreed on a detection/correct method in priori.

Modulo Arithmetic

- In modulo-*N* arithmetic, we use only the integers in the range 0 to *N*−1, inclusive.
- Calculation
  - If a number is greater than *N*−1, it is divided by *N* and the remainder is the result.
  - If it is negative, as many N's as needed are added to make it positive.
- Example in Modulo-12
  - $15_{12} = 3_{12}$
  - $-3_{12} = 9_{12}$

Modulo-2 Arithmetic

- Possible numbers are {0, 1}
- Arithmetic
  - Addition
    - 0+0=0, 0+1=1, 1+0= 1, 1+1=2=0
  - Subtraction
    - 0-0=0, 0-1=-1=1, 1-0=1, 1-1=0
  - Surprisingly, the addition and subtraction give the same result.
  - XOR (exclusively OR) can replace both addition and subtraction.



a. Two bits are the same, the result is 0.

b. Two bits are different, the result is 1.

c. Result of XORing two patterns

Detection methods

- **Parity check**
- **Cyclic redundancy check**
- **checksum**

Parity check

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

SIMPLE PARITY CHECK

- A simple parity-check code is a single-bit error-detecting code in which $n = k + 1$ with $d_{min} = 2$.
- A simple parity-check code can detect an odd number of errors.

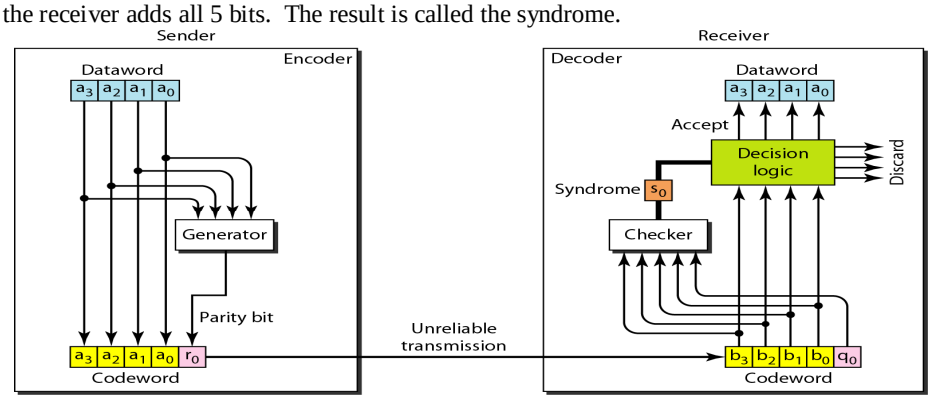
In a simple parity check a redundant bit is added to a string of data so that total number of 1's in the data become even or odd.

The total data bit is then passed through parity checking function. For even parity, it checks for even number of 1's and for odd parity it checks even number of 1's. If an error is detected the data is rejected.

Datawords	*Codewords*	*Datawords*	*Codewords*
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111
0100	01001	1100	11000
0101	01010	1101	11011
0110	01100	1110	11101
0111	01111	1111	11110

Encoder and decoder for simple parity-check code

- In modulo,
  - $r_0 = a_3+a_2+a_1+a_0$
  - $s_0 = b_3+b_2+b_1+b_0+q_0$
- Note that the receiver adds all 5 bits. The result is called the syndrome.



Example 1: data to be transmitted = 10110101

- 5 1's in the data
- Parity bit is 1
- Transmitted codeword = 101101011
- If receiver gets 101101011, parity check ok ---accept (OK)
- If receiver gets 101100011, parity check fails ---reject (OK), ask for frame to be re-transmitted
- If receiver gets 101110011, parity check ok ---accept (NOT OK: even number of errors undetected)
- If receiver gets 001100011, parity check ok ---accept (NOT OK: even number of errors undetected)

Let us look at some transmission scenarios. Assume the sender sends the dataword 1011. The codeword created from this dataword is 10111, which is sent to the receiver. We examine five cases:

1. No error occurs; the received codeword is 10111. The syndrome is 0. The dataword 1011 is created.
2. One single-bit error changes a_1. The received codeword is 10011. The syndrome is 1. No dataword is created.
3. One single-bit error changes r_0. The received codeword is 10110. The syndrome is 1. No dataword is created.
4. An error changes r_0 and a second error changes a_3 . The received codeword is 00110. The syndrome is 0. The dataword 0011 is created at the receiver. Note that here the dataword is wrongly created due to the syndrome value.
5. Three bits—a_3, a_2, and a_1—are changed by errors. The received codeword is 01011. The syndrome is 1. The dataword is not created. This shows that the simple parity check, guaranteed to detect one single error, can also find any odd number of errors.

CYCLIC REDUNDANCY CHECK

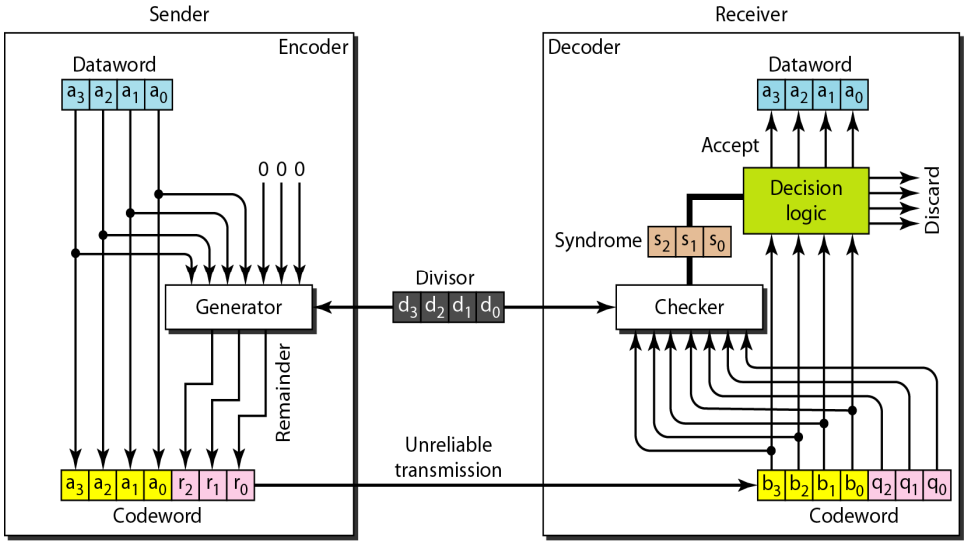
CRC is based on binary division. In CRC, instead of adding bits to achieve the desired parity, a sequence of redundant bits, called the CRC or the CRC remainder, is appended to the end of the data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number. At its destination, the incoming data unit is assumed to be intact and is therefore accepted. A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

STEP BY STEP PROCEDURE

- Dividing the data unit by a predetermined divisor derives the redundancy bits used by CRC; the remainder is CRC.
- First a starting of n 0's is appended to the data unit. The number n is one less than the number of bits in the predetermined divisor, which is n+1 bits.
- The newly elongated data unit is divided by the divisor, using a process called binary division. The remainder resulting from this division is the CRC.
- The CRC of n bits derived in step 2 replaces the appended 0s at the end of the data unit. Note that the CRC may consist of all 0s.
- The data unit arrives at the receiver data first, followed by the CRC. The receiver treats the whole string as unit and divides it by the same divisor that was used to find the CRC remainder.
- If the string arrives without error, the CRC checker yields a remainder of zero ad the data unit passes. If the string has been changed in transit, the division yields a non zero remainder and the data does not pass.

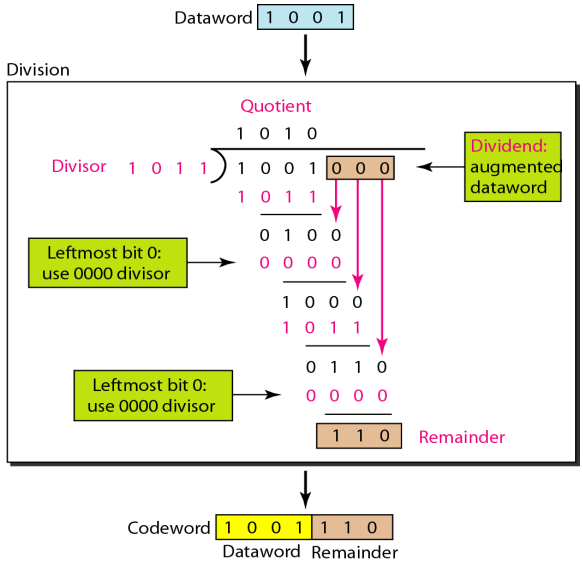
Dataword	*Codeword*	*Dataword*	*Codeword*
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

Architecture of CRC

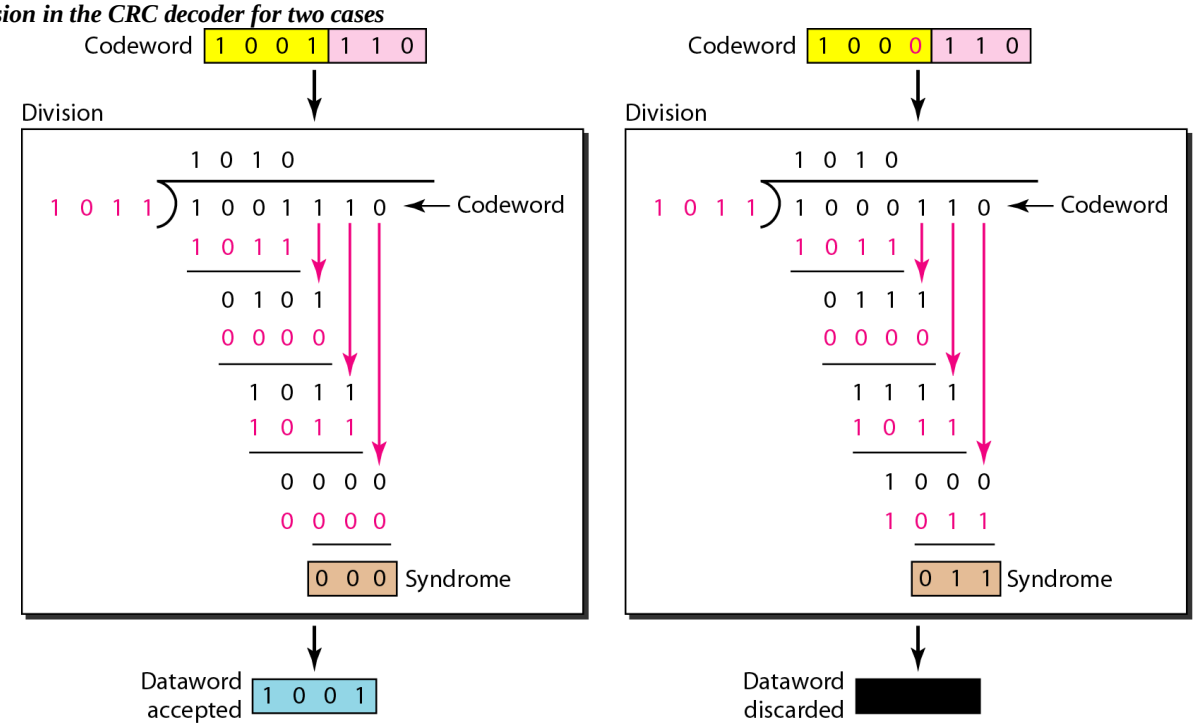


CRC GENERATOR AND CHECKER

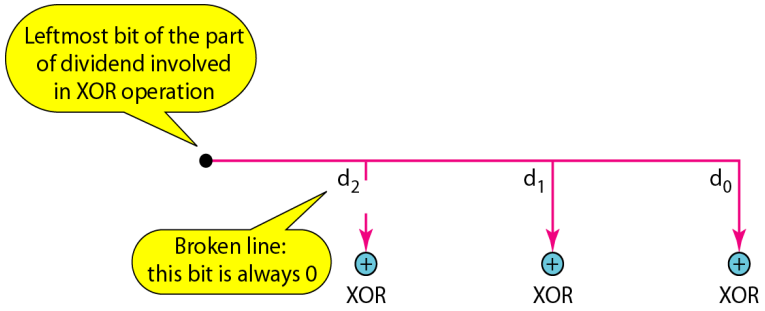
CRC GENERATOR



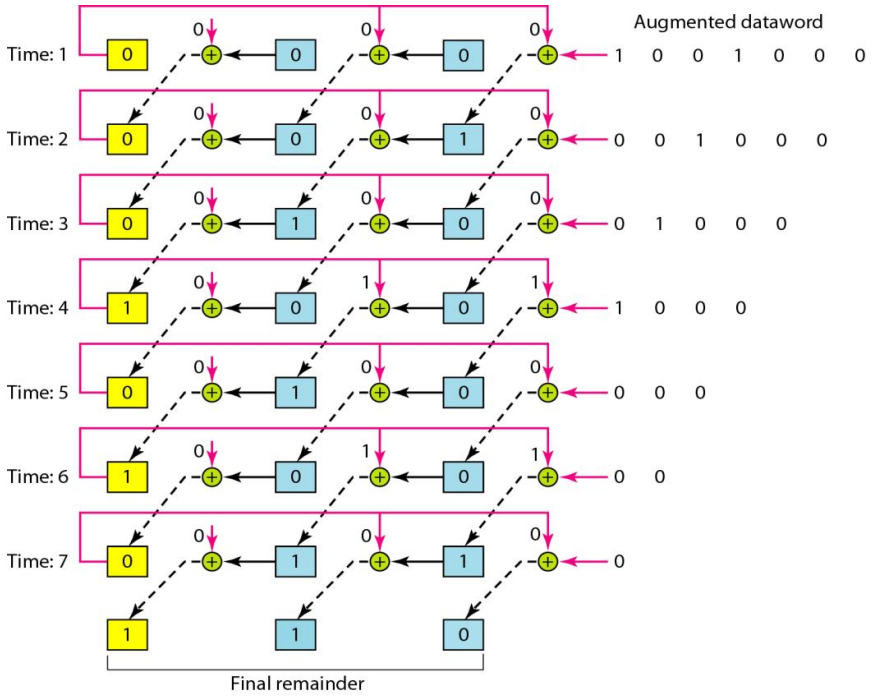
Division in the CRC decoder for two cases



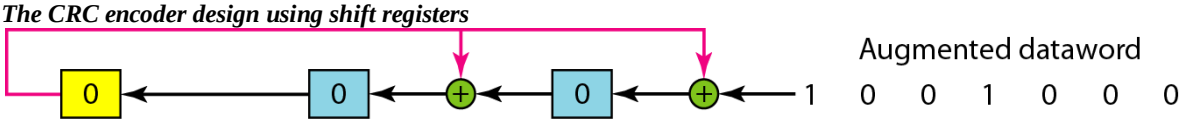
Hardwired design of the divisor in CRC



Simulation of division in CRC encoder



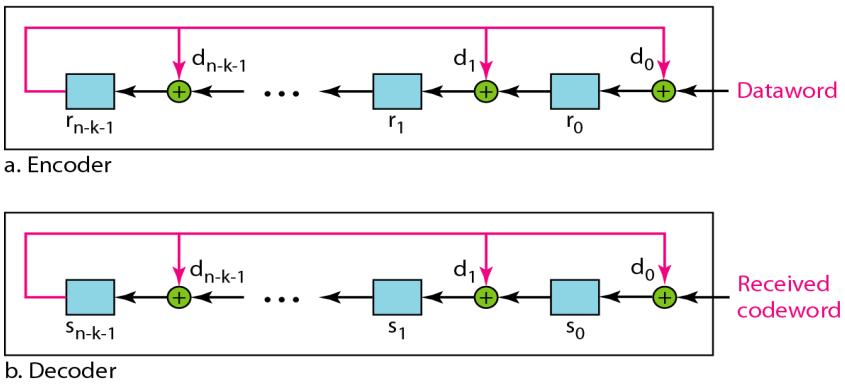
The CRC encoder design using shift registers



General design of encoder and decoder of a CRC code

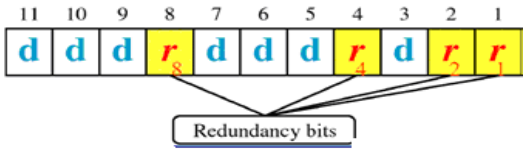
Note:

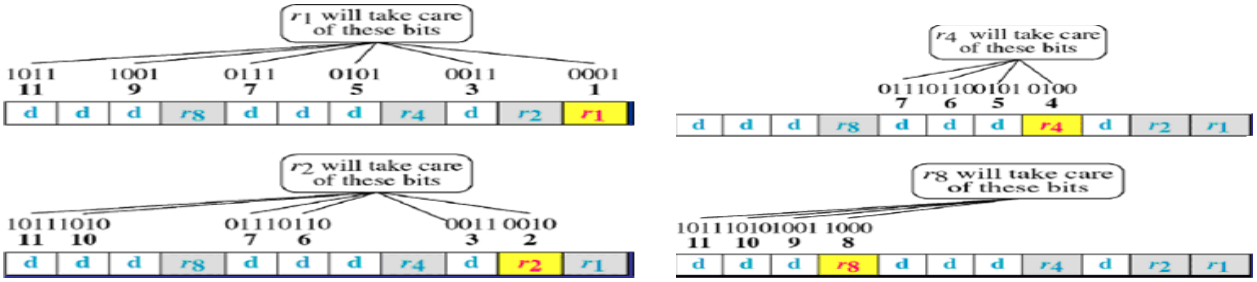
The divisor line and XOR are missing if the corresponding bit in the divisor is 0.

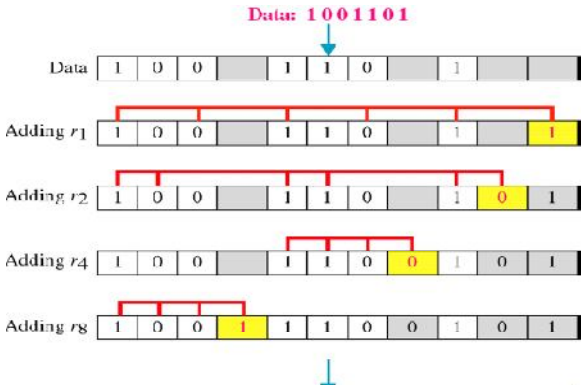


HAMMING CODE:

•A minimum ***number of redundancy bits*** needed to correct any single bit error in the data
•A minimum of 4 redundancy bits is needed if the number of data bits is 4.
•Redundancy bits in the Hamming code are placed in the codeword bit positions that are a power of 2
•Each redundancy bit is the parity bit for a different combination of data bits
•Each data bit may be included in more than one parity check.







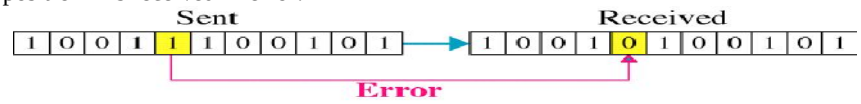
- Easy way to compute the redundancy bit values: write down binary representations for positions of data bits which contain a 1; compute parity bits for each "column"; put parity bits into codeword in correct order.
- Here: data is 1001101 so codeword will look like 100x110x1xx (where x denotes redundancy bits) □1's in positions 3, 6, 7, and 11

11: 1 0 1 1
7: 0 1 1 1
6: 0 1 1 0
3: 0 0 1 1
1 0 0 1

parity bits

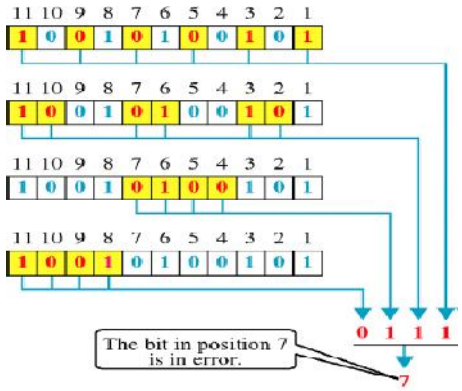
r1 r2 r4 r8

So codeword is 100*1*1100*1*01(as before)
suppose that the bit in position 7 is received in error:



• If the transmitted codeword is received error-free, the "new" parity bits the receiver computes will all be 0 ,the receiver knows no bit errors occurred.

• This simple form of Hamming code can be used to provide some protection against burst errors, by transmitting 1st bit from every codeword to be transmitted, then 2nd bit from every one of these codeword, and so on…In some cases, burst errors can be corrected



FLOW CONTROL AND ERROR CONTROL

The two main features of data kink layer are flow control and error control.

FLOW CONTROL

Flow control coordinates that amount of data that can be sent before receiving ACK It is one of the most important duties of the data link layer.

ERROR CONTROL

- Error control in the data link layer is based on ARQ (automatic repeat request), which is the retransmission of data.
- The term error control refers to methods of error detection and retransmission.
- Anytime an error is detected in an exchange, specified frames are retransmitted. This process is called ARQ.

FLOW AND ERROR CONTROL MECHANISMS

1. STOP-AND WAIT ARQ.
2. GO-BACK-N ARQ.
3. SELECTIVE-REPEAT ARQ.

STOP-AND- WAIT ARQ

This is the simplest flow and error control mechanism. It has the following features.

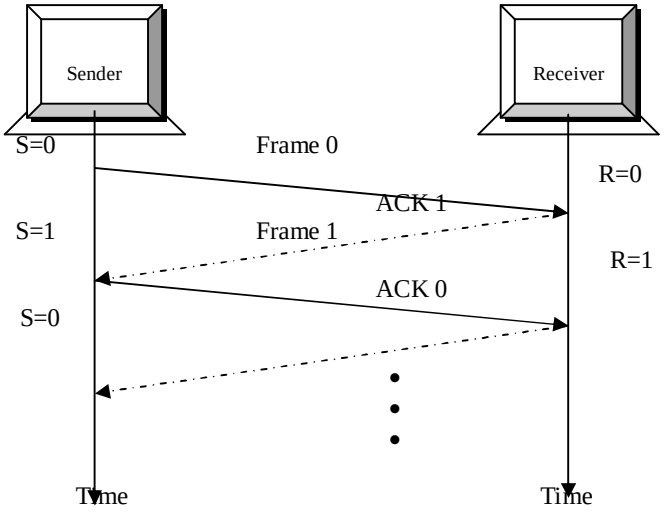
- The sending devise keeps the copy of the last frame transmitted until it receives an acknowledgement for that frame. Keeping a copy allows the sender to re- transmit lost or damaged frames until they are received correctly.
- Both data and acknowledgement frames are numbered alternately 0 and 1. A data frame 0 is acknowledged by an ACK
- A damaged or lost frame is treated in the same manner by the receiver. If the receiver detects an error in the received frame, it simply discards the frame and sends no acknowledgement.
- The sender has a control variable, which we call S, that holds the number of recently sent frame. The receiver has a control variable, which we call R that holds the number of the next frame expected.
- The sender starts a timer when it sends a frame. If an ACK is not received within an allotted time period the sender assumes that the frame was lost or damaged and resends it.

OPERATION:

The possible operations are
- Normal operation
- lost frame
- ACK lost
- delayed ACK.

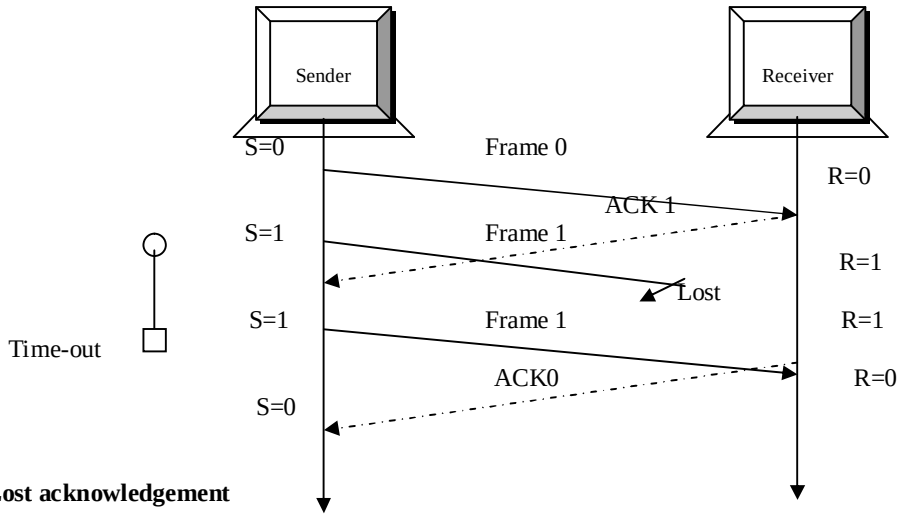
The sender sends frame 0 and wait to receive ACK 1. when ACK 1 is received it sends frame 1 and then waits to receive ACK 0, and so on.

The ACK must be received before the time out that is set expires. The following figure shows successful frame transmission.



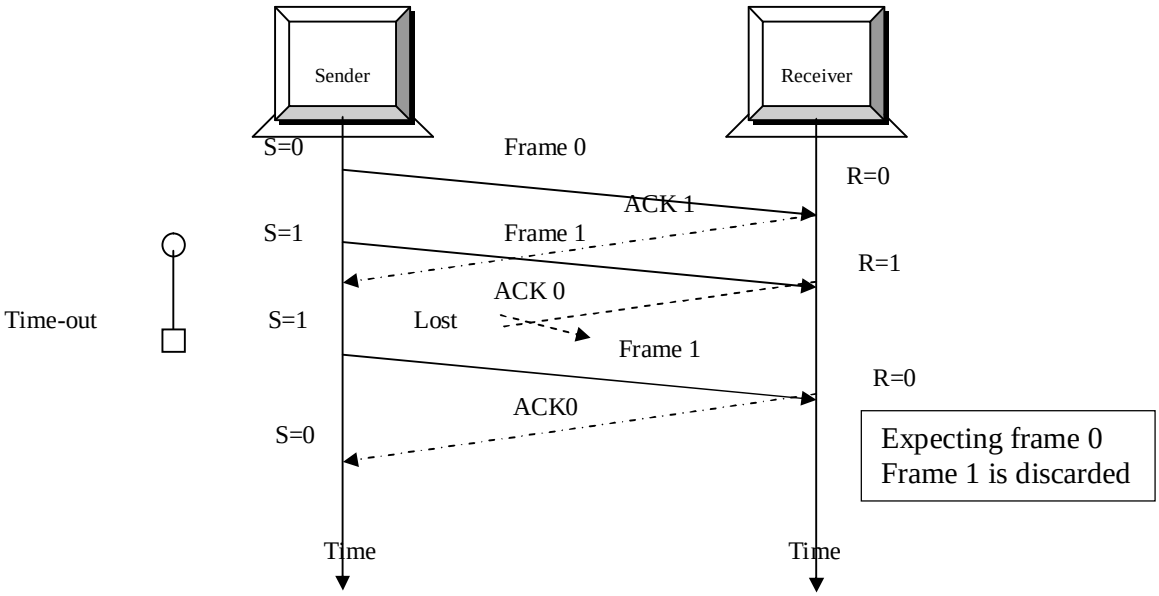
Lost or damaged acknowledgement

- When the receiver receives the damaged frame it discards it, which essentially means the frame is lost. The receiver remains silent about a lost frame and keeps its value of R.
- For example in the following figure the sender transmits frame 1, but it is lost. The receiver does nothing, retaining the value of R (1). After the timer at the sender site expires, another copy of frame 1 is sent.

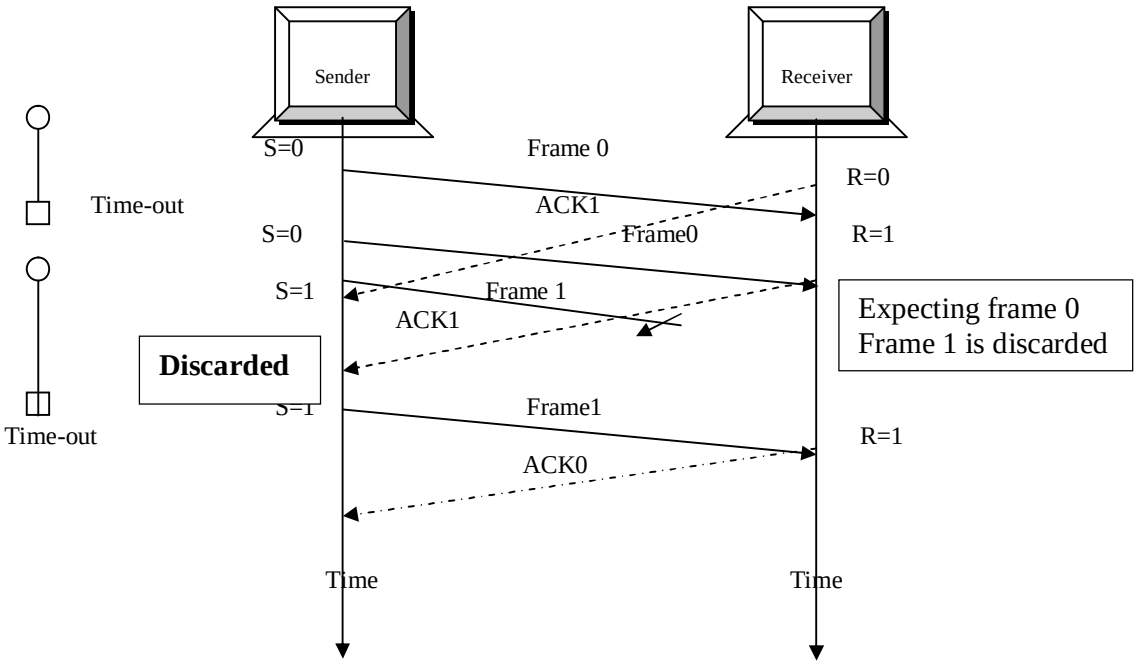


Lost acknowledgement

- A lost or damaged ACK is handle in the same by the sender; if the sender receives a damaged ACK, it discards it.
- The following figure shows a lost ACK 0.the waiting sender does not know if frame 1 has been received. When the timer for frame 1 expires the sender retransmits frame 1.
- Note that the receiver has already received frame 1 and is expecting to receive frame 0. Therefore, its silently discards the second copy of frame 1.



- **Delayed acknowledgement**
  - An ACK can be delayed at the receiver or by some problem with the link. The following figure shows the delay of ACK 1; it ids received after the timer for frame 0 as already expired.
  - The sender has already retransmitted a copy of frame 0. The receiver expects frame 1 so its simply discards the duplicate frame 0.
  - The sender has now received two ACK's, one that was delayed and one that was sent after the duplicate frame 0 arrived. The second ACK 1 is discarded.

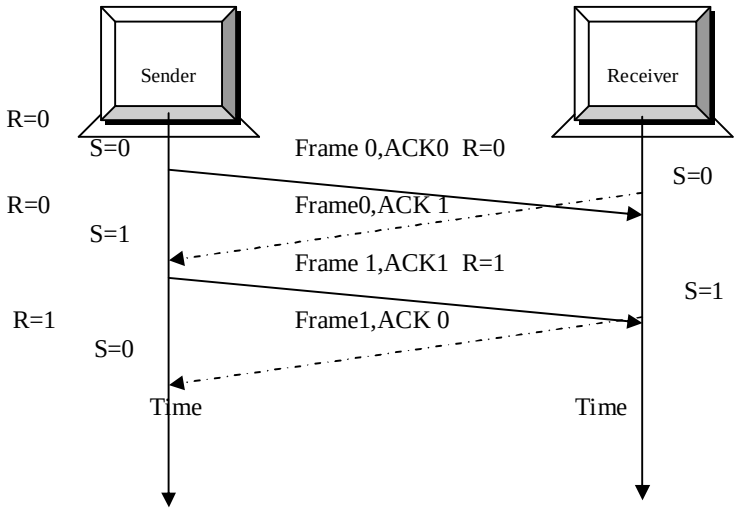


BIDIRECTIONAL TRANSMISSION

The stop – and – wait mechanism is unidirectional. We can have bi-directional transmission if the two parties have two separate channels for full duplex communication or share the same channel for off duplex transmission. In this case, each party needs both S and R variables to track frames sent and expected.

PIGGYBACKING

It's a method to combine a data frame with an ACK. In following figure both the sender and the receiver have data to send. Instead of sending separate data and ACK frames. It can save bandwidth because the overhead from a data frame and an ACK frame can be combined into just one frame



GO-BACK-N ARQ

- As in Stop-and-wait protocol senders has to wait for every ACK then next frame is transmitted. But in GO-BACK-N ARQ number of frames can be transmitted without waiting for ACK. A copy of each transmitted frame is maintained until the respective ACK is received.

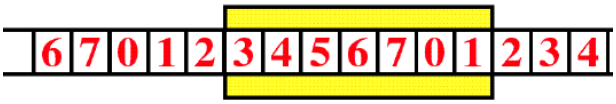
Features of GO-BACK-N ARQ

1. **Sequence numbers.**

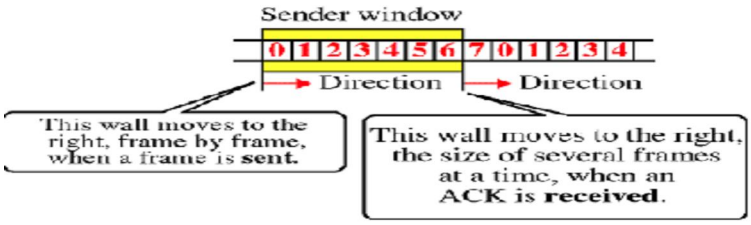
Sequence numbers of transmitted frames are maintained in the header of frame. If k is the number of bits for sequence number, then the numbering can range from 0 to 2k-1. Example: if k=3 means sequence numbers are 0 to 7.

2. Sender sliding window:

- Window is a set of frames in a buffer waiting for ACK. This window keeps on sliding in forward direction, the window size is fixed. As the ACK is received, the respective frame goes out of window and new frame to sent come into window. Figure illustrates the sliding window.
- If Sender receives. ACK 4, then it ***knows Frames up to*** and including Frame 3 were ***correctly received***

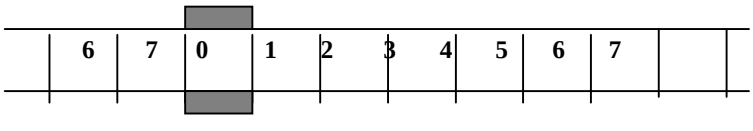


Window size=7



4. **Receiver sliding window:**

In the receiver side size of the window is always one. The receiver is expecting to arrive frame in specifies sequence. Any other frame is received which is out of order is discarded. The receiver slides over after receiving the expected frame. The following figure shows the receiver side-sliding window.



4. Control variables:

Sender variables and Receiver variables: Sender deals with three different variables S -> sequence number of recently sent frame S_F -> sequence number of first frame in the window.	The receiver deals with only one variable R -> sequence number of frame expected.

5. Timers

The sender has a timer for each transmitted frame. The receivers don't have any timer.

6. Acknowledgement:

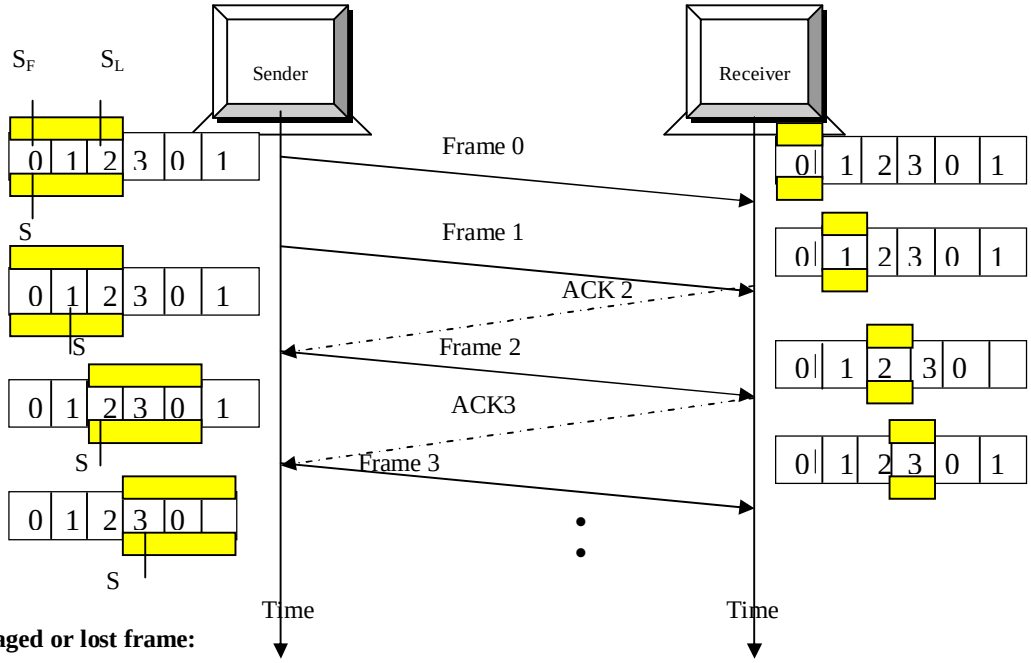
The receiver responds for frame arriving safely by positive ACK. For damaged or lost frames receiver doesn't reply, the sender has to retransmit it when timer of that frame elapsed. The receiver may ACK once for several frames.

7. Resending frames:

If the timer for any frame expires, the sender sender has to resend that frame and the subsequent frame also, hence the protocol is called GO-BACK-N ARQ.

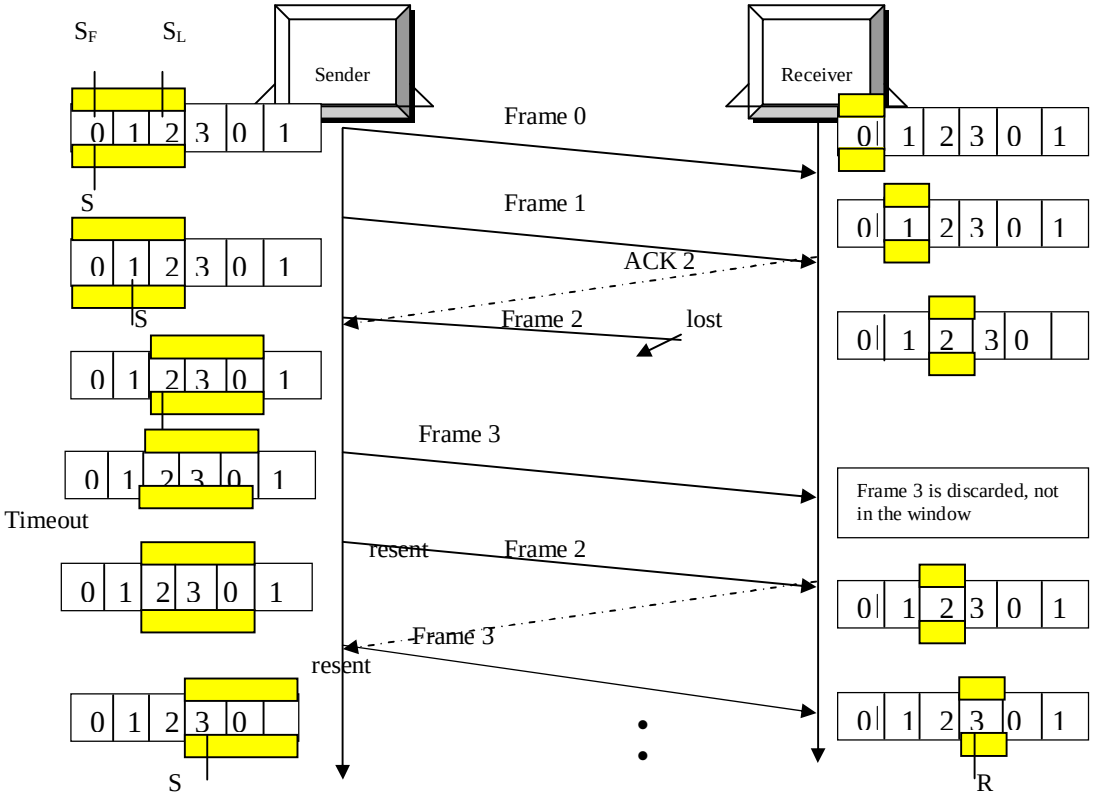
Operation

Normal operation: Following diagram shows this mechanism. The sender keeps track of the outstanding frames and updates the variables and windows as acknowledgements arrive.



Damaged or lost frame:

Figure shows that frame 2 is lost. Note that when the receiver receives frame 3, it is discarded because the receiver is expecting frame 2, not frame3. after the timer for frame 2 expires at the sender site, the sender sends frame 2 and 3.



Damaged or lost acknowledgement:

If an ACK is lost, we can have two situations. If the next ACK arrives before the expiration of timer, there is no need for retransmission of frames because ACK are cumulative in this protocol.. if the next ACK arrives after the timeout, the frame and all the frames after that are resent. The receiver never resends an ACK.

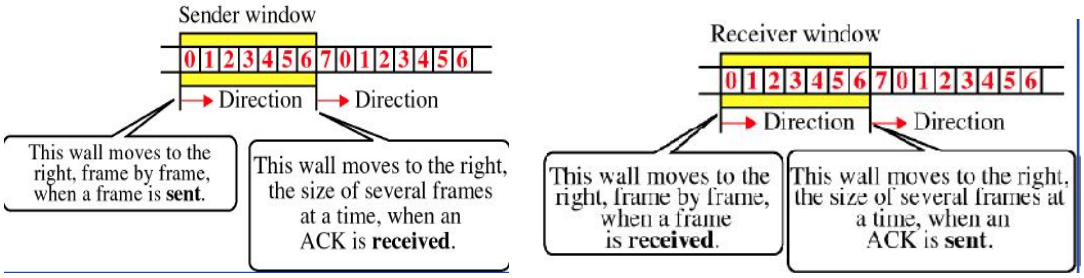
For diagrams refer your class work notes.

Delayed Acknowledgement:

A delayed ACK also triggers the resending of frames.

SELECTIVE REPEAT ARQ:

- The configuration and its control variables for this are same as those selective repeat ARQ.
- The size of the window should be one half of the value 2^m.
- The receiver window size must also be the size. In this the receiver is looking for a range of sequence numbers.
- The receiver has control variables R_F and R_L to denote the boundaries of the window.



Selective repeat also defines a negative ACK NAK that reports the sequence number of a damaged frame before the timer expires.

Operation

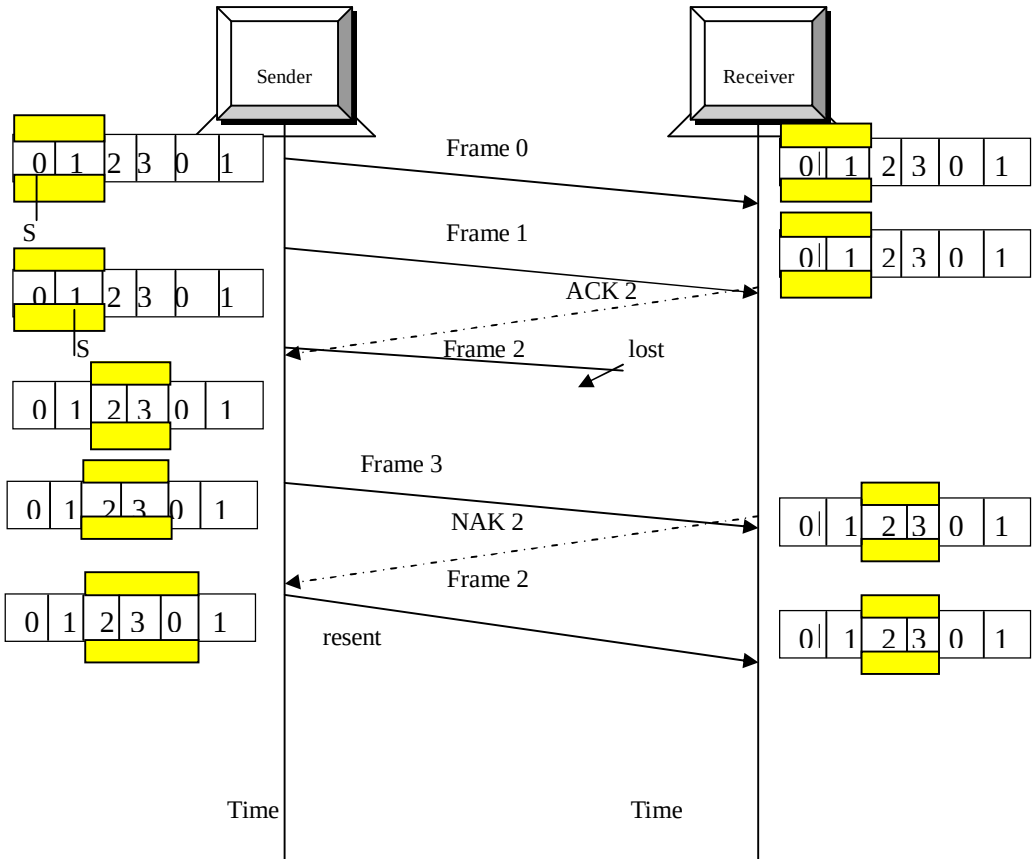
Normal operation

Normal operations of the selective repeat ARQ are same as GO-BACK-N ARQ mechanism.

Lost or damaged frame

The following figure shows operation of the mechanism with an example of a lost frame**.**

Frame 0 and 1 are accepted when received because they are in the range specified by the receiver window. When frame 3 is received, it is also accepted for the same reason. However the receiver sends a NAK 2 to show that frame 2 has not been received. When the sender receives the NAK 2, it resends only frame 2, which is then accepted because it is in the range of the window.



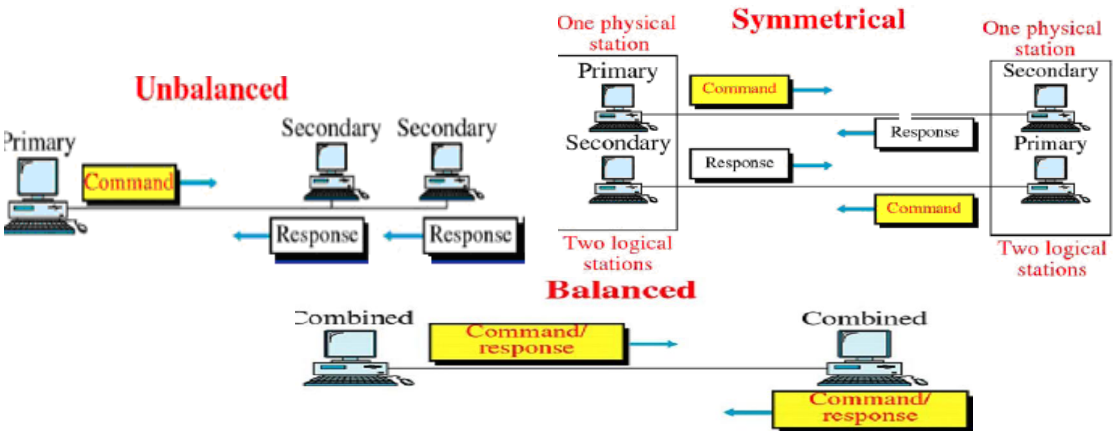
Lost and delayed ACKs and NAKs

In this sender also sets a timer for each frame sent. The remaining operations are same as GO-BACK-N ARQ.

High-level Data Link Control (HDLC) protocol

- HDLC standardized ISO in 1979 and accepted by most other standards bodies (ITU-T, ANSI)
- 3 types of end-stations:
  - *Primary*–sends commands
  - *Secondary*–can only respond to Primary's commands
  - *Combined*–can both command and respond
- 3 types of configuration

(Note: no balanced multipoint)



TRANSFER MODE

- Mode = relationship between 2 communicating devices;
- Describes who controls the link
  - NRM = Normal Response Mode
  - ABM = Asynchronous Balanced Mode

NRM:

Only difference is that secondary needs permission from the Primary in NRM, but doesn't need permission from the Primary in ARM.

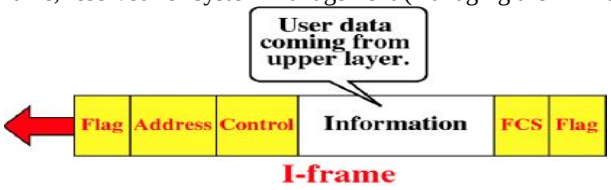
FRAMES:

Three types of Frames are

I-Frame – transports user data and control info about user data**.**

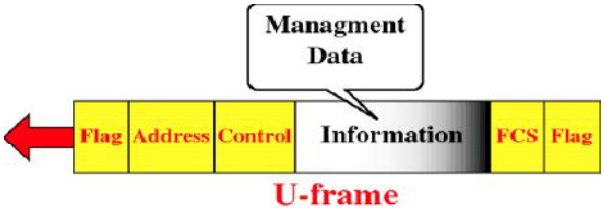
S-Frame – supervisory Frame, only used for transporting control information

U-Frame – unnumbered Frame, reserved for system management (managing the link itself)



FRAME FORMAT

U-Frames:



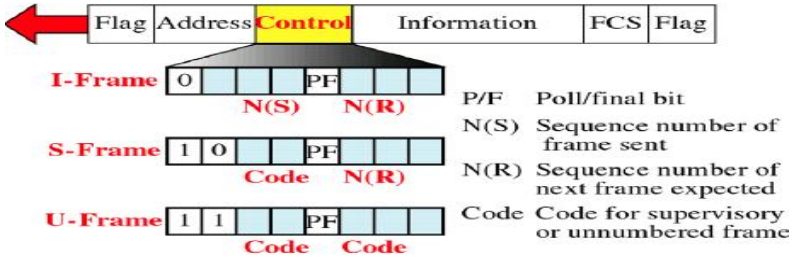
- U-frames are used for functions such as link setup. They do not contain any sequence numbers.
- Five code bits denote the frame type (but there are not 32 different possibilities):
- Set Asynchronous Balanced Mode (SABM).Used in the link set up to indicate ABM mode will be used.
- Set Normal Response Mode (SNRM).Used for asymmetric mode (master/slave).
- SABME and SNMRE—extended format.
- Disconnect (DISC).Used to disconnect the logical connection.
- Frame Reject (FRMR)—reject frame with incorrect semantics.
- Unnumbered Acknowledgement (UA).Used to acknowledge other frames in this class.
- Unnumbered Information (UI)–initialisation, poling and status information needed by the data link layer.
- U-frames may carry data when unreliable connectionless service is called for.

S-Frames:

S-frame

- S-frames are similar to unnumbered frames, the main difference being that they do carry sequence information.
- Some supervisory frames function as positive and negative acknowledgements, they therefore play a very important role in error and flow control.
- Two bits indicate the frame type, so that there are four possibilities.
  - Receiver Ready -RR(Positive Acknowledgement)
  - Receiver Not Ready -RNR
  - Reject -REJ(NAK go-back-N)
  - Selective Reject -SREJ(NAK selective retransmit)

Control Field:

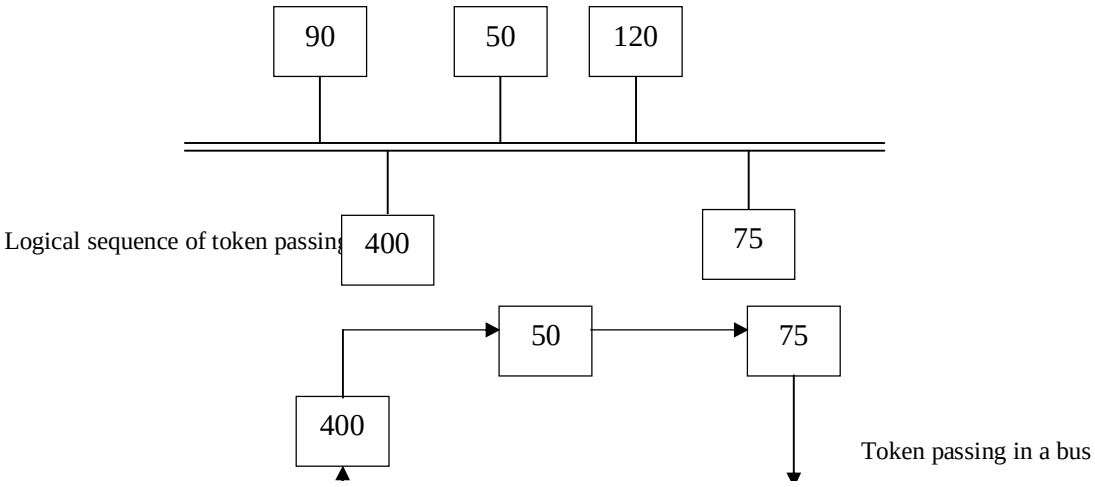


IEEE 802.4 TOKEN BUS

- IEEE 802.4 describes a token bus LAN standard.
- In token passing method stations, connected on a bus are arranged in a logical ring. When the logical ring is initiated, the highest number station may send the first frame. After this it passes permission to its immediate neighbor by sending a special frame called a token.
- The token propagates around the logical ring, with only the token holder being permitted to transmit frames. Since only one station at a time holds the token, collision do not occur.
- There is no relation between the physical location of the station on the bus and its logical sequence number..

The following figure shows the operation of the token bus.

Physical topology



Token passing in a bus

802.4 cable standards

- The token bus standard specifies three physical layer options in terms of transmission medium, signaling technique, data rate and maximum electrical cable segment length.

Medium options

1. Broadband: Transmission medium is co-axial cable and its uses AM/PSK as signaling techniques, data rate is 1,5,10 mbps.
2. Carrier band: Transmission medium is co-axial cable and its uses KSK as a signaling techniques, data rate is 1, 5,10Mbps.
3. Optical fiber: Transmission medium is optical fiber and its uses ASK with Manchester encoding as a signaling techniques, data rate is 5,10,20Mbps.

IEEE 802.4 Frame format

- Token bus frame format is shown in the following figure.

1	1	1	2-6	2-6	0-8182	4

Preamble	SD	FC	DA	SA	DATA	FCS	ED

- Preamble: the preamble is an at least one byte long pattern to establish bit synchronization
- SD: Start frame delimiter: Its also one byte unique bit pattern, which marks the start of the frame.
- FC: Frame control: The frame control field is used to distinguish data frames from control frames. For data frame, it carries the frames priority. The frame control field indicates the type of the frame data frame or control frame.
- DA: Destination address: The destination address field is 2 or 6 bytes long.
- SA: Source address: The destination address field is 2 or 6 bytes long.
- DATA: Data field
- FCS: Frame check sequence: frame check sequence is 4 bytes long and contains CRC code. It is used to detect transmission errors on DA, SA, FC and data fields.
- ED: End delimiter: It is a unique bit pattern, which marks the end of the frame. It is one byte long.
- The total length of the frame is 8191 bytes.

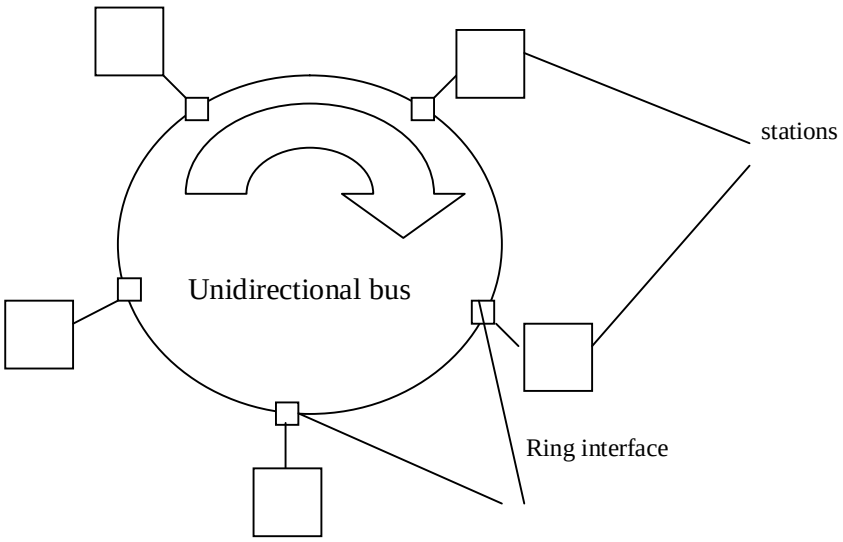
Performance:

For token ring, the slightly higher delay compared to CSMS/CD bus occurs. For higher transmission loads the token ring performs well.

IEEE 802.5 TOKEN RING

- IEEE 802.4 describes a token ring LAN standard.
- In a token ring a special bit pattern, called the token circulates around the ring when all stations are idle.
- When a station transmits, it breaks the ring and inserts its own frame with source and destination address.
- When the frame eventually returns to the originating station after completing the round, the station removes the frame and closes the ring. Because there is only one token, only one station can transmit at a given instant, thus solving the channel access problem.
- Each station is connected to the ring through a Ring Interface Unit (RIU). The sequence of token is determined by the physical locations of the stations on the ring.

The following figure shows the operation and arrangement of the Token Ring.



802.5 cable standards

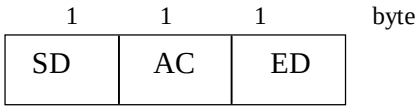
Its uses two types of transmission medium.

1. Shielded twisted pair cable: (STP)
   It uses differential Manchester encoding technique. Data rate is 4 or 16 Mbps. Maximum number of repeaters allowed is 250.
2. Unshielded twisted pair cable: (UTP)
   It uses differential Manchester encoding technique. Data rate is 4Mbps. Maximum number of repeaters allowed is 250.

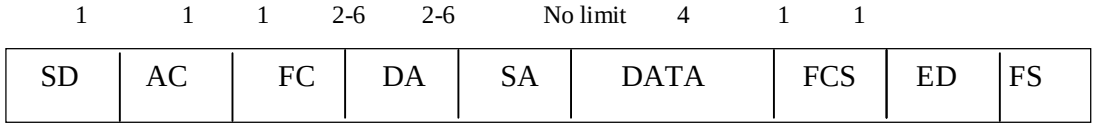
IEEE 802.5 Frame format

- Token ring frame format is shown in the following figure.

Token frame format



Data Frame



- SD: Start frame delimiter: Its also one byte unique bit pattern, which marks the start of the frame.
- AC: Access control: It is one byte long field containing priority bits(P), Token bit(T), monitoring bit(M), and reservation bir(R).
- FC: Frame control: The frame control field is used to distinguish data frames from control frames. For data frame, it carries the frames priority. The frame control field indicates the type of the frame data frame or control frame.
- DA: Destination address: The destination address field is 2 or 6 bytes long.
- SA: Source address: The destination address field is 2 or 6 bytes long.
- DATA: Data field
- FCS: Frame check sequence: frame check sequence is 4 bytes long and contains CRC code. It is used to detect transmission errors on DA, SA, FC and data fields.
- ED: End delimiter: It is a unique bit pattern, which marks the end of the frame. It is one byte long.
- FS: Frame status: This field is none byte long and contains a unique bit pattern marking the end of a token or a data frame.

Performance:

When traffic is light, the token will spend most of its time idly circulating around the ring. When traffic is heavy, there is a queue at each station. Network efficiency is more.

Disadvantages:

- A break in a link or repeater failures disturbs the entire network.
- Installation of new repeaters requires identification of two topologically adjacent repeaters.
- Since the ring is closed loop, a packet will circulate indefinitely unless it is removed.
- Each repeater adds an increment of delay.
- There is practical limit to the number of repeaters.

Fiber Distributed Data Interface

Introduction

The Fiber Distributed Data Interface (FDDI) specifies a 100-Mbps token-passing, dual-ring LAN using fiber-optic cable. FDDI is frequently used as high-speed backbone technology because of its support for high bandwidth and greater distances than copper. It should be noted that relatively recently, a related copper specification, called Copper Distributed Data Interface (CDDI), has emerged to provide 100-Mbps service over copper. CDDI is the implementation of FDDI protocols over twisted-pair copper wire. This chapter focuses mainly on FDDI specifications and operations, but it also provides a high-level overview of CDDI.

FDDI uses dual-ring architecture with traffic on each ring flowing in opposite directions (called counter-rotating). The dual rings consist of a primary and a secondary ring. During normal operation, the primary ring is used for data transmission, and the

secondary ring remains idle. As will be discussed in detail later in this chapter, the primary purpose of the dual rings is to provide superior reliability and robustness. Figure 8-1 shows the counter-rotating primary and secondary FDDI rings.

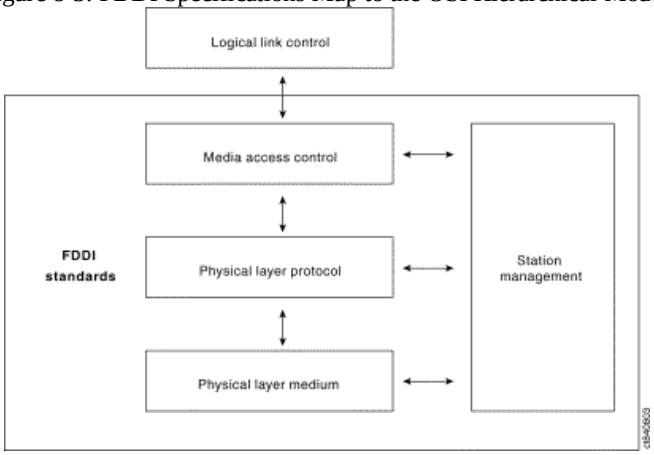
FDDI Specifications

FDDI specifies the physical and media-access portions of the OSI reference model. FDDI is not actually a single specification, but it is a collection of four separate specifications, each with a specific function. Combined, these specifications have the capability to provide high-speed connectivity between upper-layer protocols such as TCP/IP and IPX, and media such as fiber-optic cabling.

FDDI's four specifications are the Media Access Control (MAC), Physical Layer Protocol (PHY), Physical-Medium Dependent (PMD), and Station Management (SMT) specifications. The MAC specification defines how the medium is accessed, including frame format, token handling, addressing, algorithms for calculating cyclic redundancy check (CRC) value, and error-recovery mechanisms. The PHY specification defines data encoding/decoding procedures, clocking requirements, and framing, among other functions. The PMD specification defines the characteristics of the transmission medium, including fiber-optic links, power levels, bit-error rates, optical components, and connectors. The SMT specification defines FDDI station configuration, ring configuration, and ring control features, including station insertion and removal, initialization, fault isolation and recovery, scheduling, and statistics collection.

FDDI is similar to IEEE 802.3 Ethernet and IEEE 802.5 Token Ring in its relationship with the OSI model. Its primary purpose is to provide connectivity between upper OSI layers of common protocols and the media used to connect network devices. Figure 8-3 illustrates the four FDDI specifications and their relationship to each other and to the IEEE-defined Logical Link Control (LLC) sublayer. The LLC sublayer is a component of Layer 2, the MAC layer, of the OSI reference model.

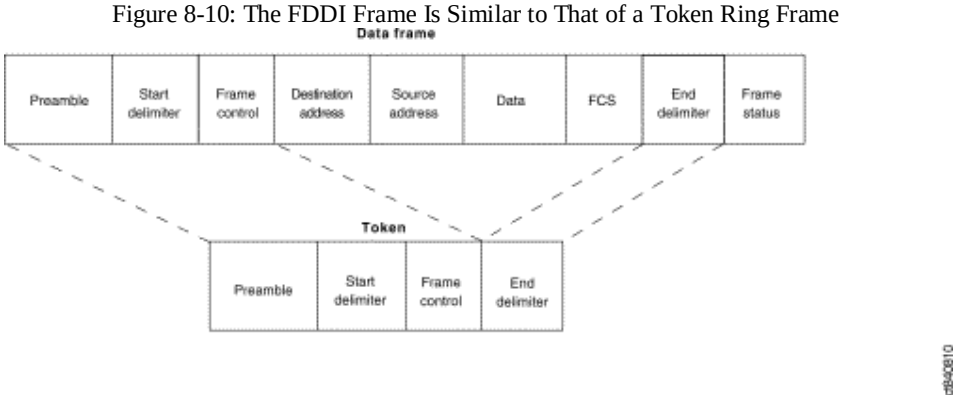
Figure 8-3: FDDI Specifications Map to the OSI Hierarchical Model



FDDI Frame Format

The FDDI frame format is similar to the format of a Token Ring frame. This is one of the areas in which FDDI borrows heavily from earlier LAN technologies, such as Token Ring. FDDI frames can be as large as 4,500 bytes. Figure 8-10 shows the frame format of an FDDI data frame and token.

Figure 8-10: The FDDI Frame Is Similar to That of a Token Ring Frame



FDDI Frame Fields

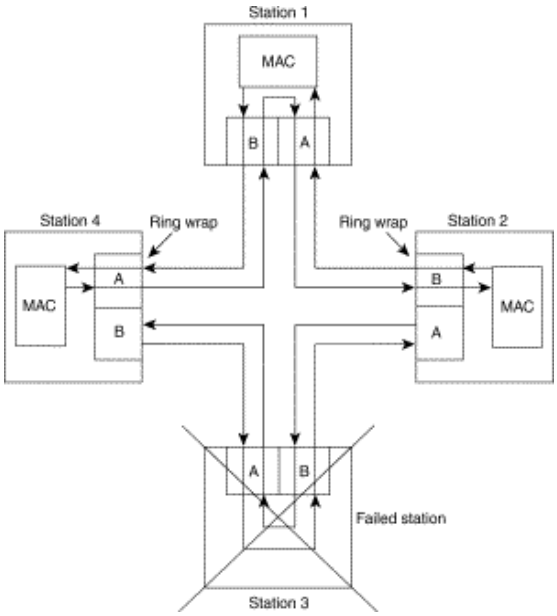
The following descriptions summarize the FDDI data frame and token fields illustrated in Figure 8-10.

- Preamble—Gives a unique sequence that prepares each station for an upcoming frame.
- Start delimiter—Indicates the beginning of a frame by employing a signaling pattern that differentiates it from the rest of the frame.
- Frame control—Indicates the size of the address fields and whether the frame contains asynchronous or synchronous data, among other control information.
- Destination address—Contains a unicast (singular), multicast (group), or broadcast (every station) address. As with Ethernet and Token Ring addresses, FDDI destination addresses are 6 bytes long.
- Source address—Identifies the single station that sent the frame. As with Ethernet and Token Ring addresses, FDDI source addresses are 6 bytes long.
- Data—Contains either information destined for an upper-layer protocol or control information.
- Frame check sequence (FCS)—Is filed by the source station with a calculated cyclic redundancy check value dependent on frame contents (as with Token Ring and Ethernet). The destination address recalculates the value to determine whether the frame was damaged in transit. If so, the frame is discarded.
- End delimiter—Contains unique symbols; cannot be data symbols that indicate the end of the frame.
- Frame status—Allows the source station to determine whether an error occurred; identifies whether the frame was recognized and copied by a receiving station.

Dual Ring

FDDI's primary fault-tolerant feature is the dual ring. If a station on the dual ring fails or is powered down, or if the cable is damaged, the dual ring is automatically wrapped (doubled back onto itself) into a single ring. When the ring is wrapped, the dual-ring topology becomes a single-ring topology. Data continues to be transmitted on the FDDI ring without performance impact during the wrap condition. Figure 8-6 and Figure 8-7 illustrate the effect of a ring wrapping in FDDI.

Figure 8-6: A Ring Recovers from a Station Failure by Wrapping



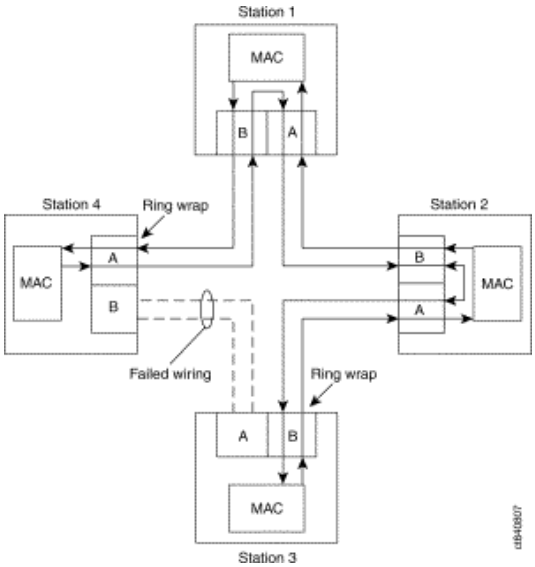


Figure 8-7: A Ring also wraps to withstand a Cable Failure

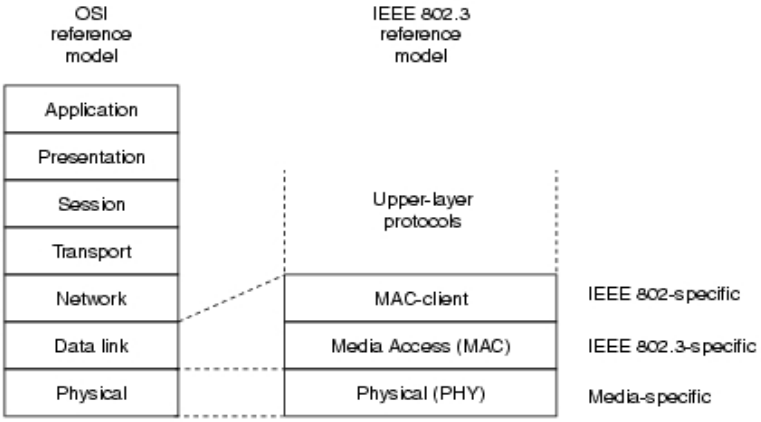
When a single station fails, as shown in Figure 8-6, devices on either side of the failed (or powered-down) station wrap, forming a single ring. Network operation continues for the remaining stations on the ring. When a cable failure occurs, as shown in Figure 8-7, devices on either side of the cable fault wrap. Network operation continues for all stations.

It should be noted that FDDI truly provides fault tolerance against a single failure only. When two or more failures occur, the FDDI ring segments into two or more independent rings that are incapable of communicating with each other.

The IEEE 802.3 Logical Relationship to the ISO Reference Model

Figure 7-4 shows the IEEE 802.3 logical layers and their relationship to the OSI reference model. As with all IEEE 802 protocols, the ISO data link layer is divided into two IEEE 802 sublayers, the Media Access Control (MAC) sublayer and the MAC-client sublayer. The IEEE 802.3 physical layer corresponds to the ISO physical layer.

Figure 7-4 Ethernet's Logical Relationship to the ISO Reference Model



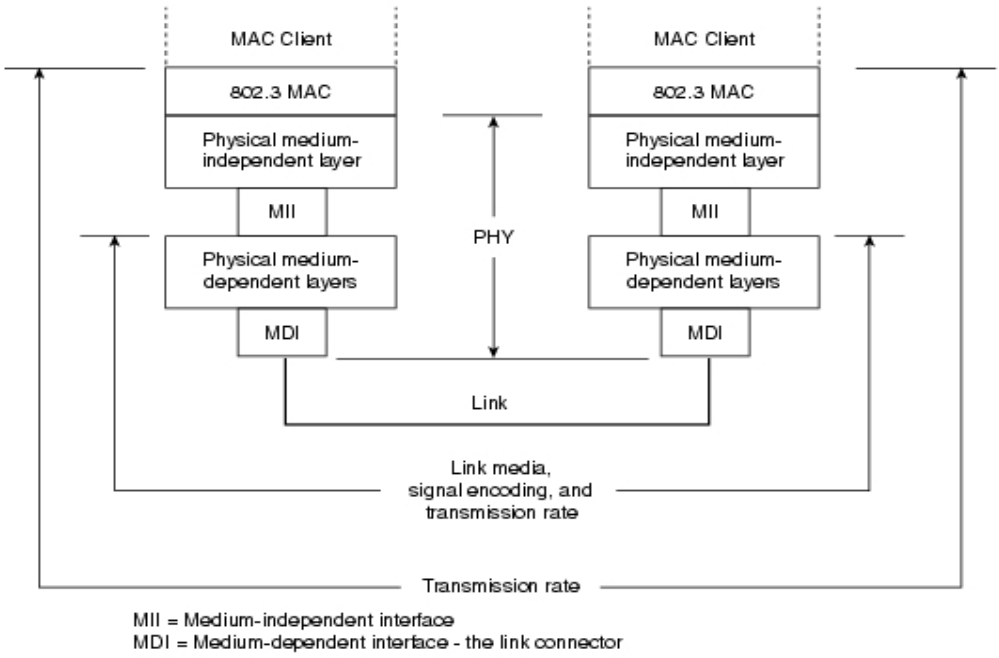
The MAC-client sublayer may be one of the following:

- Logical Link Control (LLC), if the unit is a DTE. This sublayer provides the interface between the Ethernet MAC and the upper layers in the protocol stack of the end station. The LLC sublayer is defined by IEEE 802.2 standards.

- Bridge entity, if the unit is a DCE. Bridge entities provide LAN-to-LAN interfaces between LANs that use the same protocol (for example, Ethernet to Ethernet) and also between different protocols (for example, Ethernet to Token Ring). Bridge entities are defined by IEEE 802.1 standards.

Because specifications for LLC and bridge entities are common for all IEEE 802 LAN protocols, network compatibility becomes the primary responsibility of the particular network protocol. Figure 7-5 shows different compatibility requirements imposed by the MAC and physical levels for basic data communication over an Ethernet link.

Figure 7-5 MAC and Physical Layer Compatibility Requirements for Basic Data Communication



The MAC layer controls the node's access to the network media and is specific to the individual protocol. All IEEE 802.3 MACs must meet the same basic set of logical requirements, regardless of whether they include one or more of the defined optional protocol extensions. The only requirement for basic communication (communication that does not require optional protocol extensions) between two network nodes is that both MACs must support the same transmission rate.

The 802.3 physical layer is specific to the transmission data rate, the signal encoding, and the type of media interconnecting the two nodes. Gigabit Ethernet, for example, is defined to operate over either twisted-pair or optical fiber cable, but each specific type of cable or signal-encoding procedure requires a different physical layer implementation.

The Ethernet MAC Sublayer

The MAC sub layer has two primary responsibilities:

- Data encapsulation, including frame assembly before transmission, and frame parsing/error detection during and after reception
- Media access control, including initiation of frame transmission and recovery from transmission failure

The Basic Ethernet Frame Format

The IEEE 802.3 standard defines a basic data frame format that is required for all MAC implementations, plus several additional optional formats that are used to extend the protocol's basic capability. The basic data frame format contains the seven fields shown in Figure 7-6.

• Preamble (PRE)—Consists of 7 bytes. The PRE is an alternating pattern of ones and zeros that tells receiving stations that a frame is coming, and that provides a means to synchronize the frame-reception portions of receiving physical layers with the incoming bit stream.

• Start-of-frame delimiter (SOF)—Consists of 1 byte. The SOF is an alternating pattern of ones and zeros, ending with two consecutive 1-bits indicating that the next bit is the left-most bit in the left-most byte of the destination address.

• Destination address (DA)—Consists of 6 bytes. The DA field identifies which station(s) should receive the frame. The left-most bit in the DA field indicates whether the address is an individual address (indicated by a 0) or a group address (indicated by a 1). The second bit from the left indicates whether the DA is globally administered (indicated by a 0) or locally administered (indicated by a 1). The remaining 46 bits are a uniquely assigned value that identifies a single station, a defined group of stations, or all stations on the network.

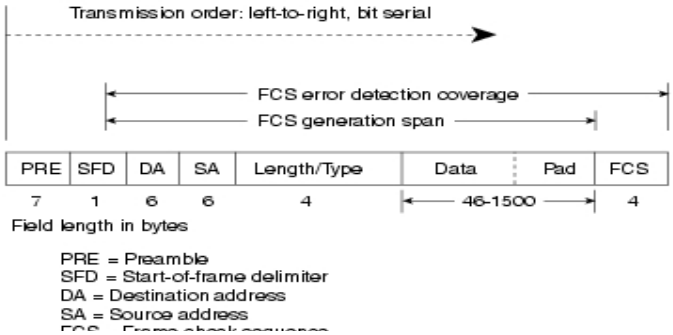
• Source addresses (SA)—Consists of 6 bytes. The SA field identifies the sending station. The SA is always an individual address and the left-most bit in the SA field is always 0.

• Length/Type—Consists of 2 bytes. This field indicates either the number of MAC-client data bytes that are contained in the data field of the frame, or the frame type ID if the frame is assembled using an optional format. If the Length/Type field value is less than or equal to 1500, the number of LLC bytes in the Data field is equal to the Length/Type field value. If the Length/Type field value is greater than 1536, the frame is an optional type frame, and the Length/Type field value identifies the particular type of frame being sent or received.

• Data—Is a sequence of n bytes of any value, where n is less than or equal to 1500. If the length of the Data field is less than 46, the Data field must be extended by adding a filler (a pad) sufficient to bring the Data field length to 46 bytes.

• Frame check sequence (FCS)—Consists of 4 bytes. This sequence contains a 32-bit cyclic redundancy check (CRC) value, which is created by the sending MAC and is recalculated by the receiving MAC to check for damaged frames. The FCS is generated over the DA, SA, Length/Type, and Data fields.

Figure 7-6 The Basic IEEE 802.3 MAC Data Frame Format



Note: Individual addresses are also known as unicast addresses because they refer to a single MAC and are assigned by the NIC manufacturer from a block of addresses allocated by the IEEE. Group addresses (a.k.a. multicast addresses) identify the end stations in a workgroup and are assigned by the network manager. A special group address (all 1s—the broadcast address) indicates all stations on the network.

Frame Transmission

Whenever an end station MAC receives a transmit-frame request with the accompanying address and data information from the LLC sublayer, the MAC begins the transmission sequence by transferring the LLC information into the MAC frame buffer.

- The preamble and start-of-frame delimiter are inserted in the PRE and SOF fields.
- The destination and source addresses are inserted into the address fields.
- The LLC data bytes are counted, and the number of bytes is inserted into the Length/Type field.
- The LLC data bytes are inserted into the Data field. If the number of LLC data bytes is less than 46, a pad is added to bring the Data field length up to 46.
- An FCS value is generated over the DA, SA, Length/Type, and Data fields and is appended to the end of the Data field.

After the frame is assembled, actual frame transmission will depend on whether the MAC is operating in half-duplex or full-duplex mode.

The IEEE 802.3 standard currently requires that all Ethernet MACs support half-duplex operation, in which the MAC can be either transmitting or receiving a frame, but it cannot be doing both simultaneously. Full-duplex operation is an optional MAC capability that allows the MAC to transmit and receive frames simultaneously.

What is a switch and its function?

A switch is a multi-input, multi-output device, which transfers packets from an input to one or more outputs.

Large networks can be built by interconnecting a number of switches. Hosts are connected to the switch using point-to-point link.

A switch receives packets on one of its links and transmits them on one or more other links. This is known as switching or forwarding.

List the different types of switched networks.

Circuit switched networks

Packet switched networks

- Datagram networks
- Virtual-circuit networks

Message switched networks

Bring out the differences between circuit and packet switching.

Circuit switching	*Packet switching*
Source and destination host are physically connected	No such physical connection exists
Switching takes place at the physical layer	Switching takes place at network (datagram) or data link layer (VCN)
Resources such as bandwidth, switch buffer & processing time, are allocated in advance.	Resources are allocated on demand
Resources remain allocated for the entire duration of data communication.	Resources can be reallocated when idle.
There is no delay during data transfer.	Delay exists at each switch during data transfer
Data transferred between the two stations is a continuous flow of signal	Data is transferred as discrete packets
Example: *Telephony*	Example: *Internet*

Explain packet switched networks in detail.

Datagram networks

Datagram network is referred to as *connectionless* network.

In connectionless, the switch does not keep information about the connection state. In a datagram network, the message is divided into *packets* of fixed or variable size.

The sender does not know whether the network will deliver or destination host is alive. There is *no resource allocation* for a packet. There is no reserved bandwidth on the links, and no scheduled processing time.

Resources are allocated on demand. The allocation is primarily done on a FCFS basis.

When a packet arrives at a switch, it must wait if there are other packets being processed. ○ The lack of reservation creates *delay*.

Each packet is treated independently of all others regardless of its source or destination.

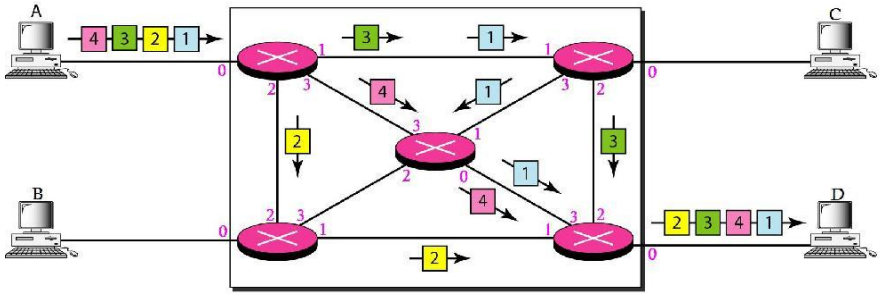
Packets belonging to the same message may travel different paths to reach their destination.

Packets can arrive out of order at the destination

Packets may also be dropped due to lack of resources.

A switch or link failure may not have any adverse effect, if an alternate path is available. 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.

Datagram switching is done at the *network* layer.



Datagram network

Destination	Port
A	3
B	0
C	3
D	3

Routing table

Routing table

Each switch has a *routing* table that contains destination address and the corresponding output port.

When the switch receives a packet:

- destination address is examined
- The routing table is consulted to find the corresponding port through which the packet should be forwarded.

The routing table is *dynamic* and is updated periodically.

Analysis

The efficiency of a datagram network is better than that of a circuit-switched network, since resources are allocated only when there are packets to be sent.

The resources can be reallocated if idle, for other packets.

Each packet experiences a wait at a switch if there are packets queued up, before it is forwarded.

Virtual Circuit Switching networks

A virtual-circuit network (VCN) is a *connection-oriented* model. It is implemented in the *data link* layer.

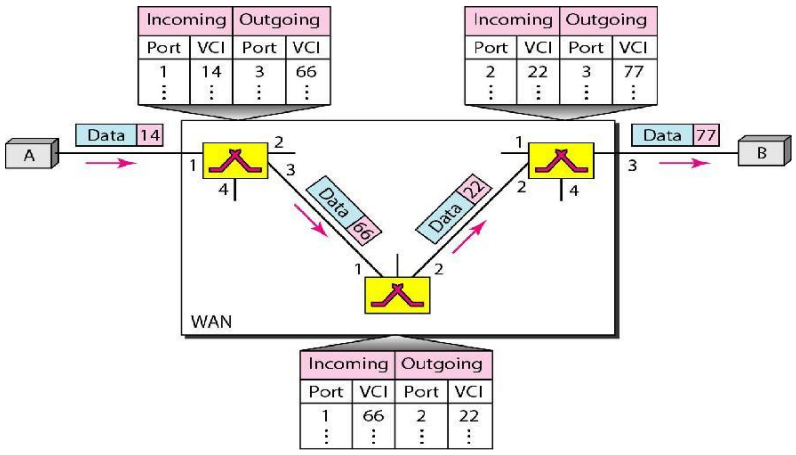
A *virtual connection* from the source to the destination is established before any data is sent. This is known as *connection setup* phase

- Each switch should contain an entry in VC table that has four columns.
- The entry contains incoming port, incoming VCI, outgoing port and outgoing VCI A *Virtual Circuit Identifier* (VCI) uniquely identifies a connection at a switch.
- A VCI is a small number that has only *link local scope*.
- A frame arrives at a switch with a VCI and when it leaves, it has a different VCI.
- VCI configured by the administrator is known as *permanent virtual circuit* (PVC)
- In large networks, VCI is set by a host through signaling, known as *switched virtual circuit* (SVC)

In the *teardown* phase, the source requests the switches to delete corresponding entries.

Permanent Virtual Circuit

The figure shows a frame from source A on its way to destination B and how its VCI changes during the trip as configured by the network administrator.



The frame arrives at port 1 with a VCI of 14.

The switch looks in its table to find port 1 and a VCI of 14. When it is found, the switch knows to change the VCI to 66 and send out the frame from port 3.

Each switch changes the VCI and routes the frame.

The data transfer phase is active until source sends all its frames to the destination. This process creates a *virtual circuit* between the source and destination.

Switched Virtual Circuit

Two steps are involved in the setup phase namely *setup request* and *acknowledgment*.

Setup Request—A setup request frame is sent from the source to the destination.

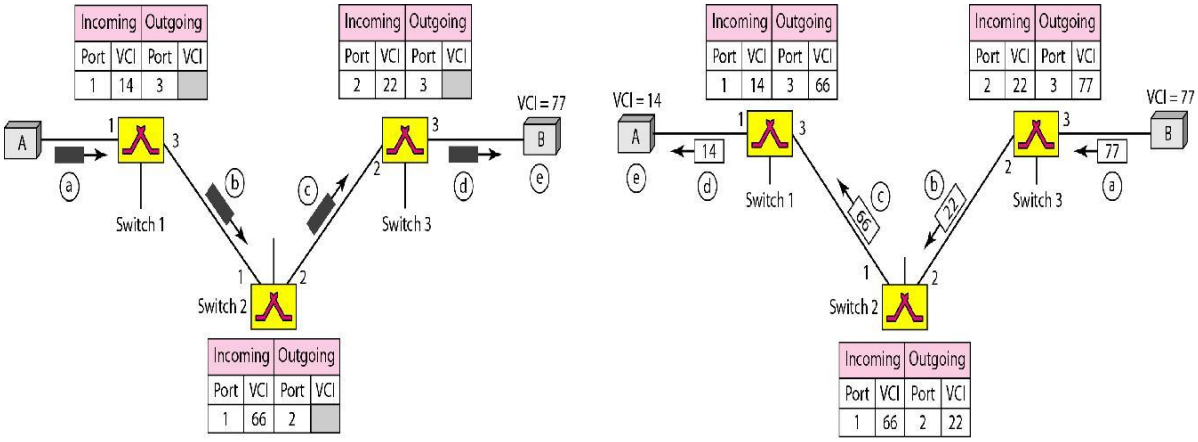
a) Source *A* sends a setup frame to *switch1*.
b) *Switch1* receives the setup request frame.
   - It knows that a frame going from *A* to *B* goes out through port *3*.
   - The switch creates an entry in its table for this virtual circuit.
   - The switch assigns the incoming port (1), chooses an available incoming VCI (14) and the outgoing port (3).
   - It *does not know* the outgoing VCI and is left blank.
   - Forwards frame to *switch2*.
c) *Switch2* receives the setup request frame.
   - The table entries made are incoming port (l), incoming VCI (66), and outgoing port (2). The frame is forwarded to *switch3*.
d) Similarly for *switch3* the entries are incoming port (2), incoming VCI (22), and outgoing port (3).
e) Destination *B* receives the setup frame, and if it is ready to receive frames from host *A*,
   - It assigns an unused VCI to the incoming frames that come from *A*, say 77.
   - When a frame comes with VCI 77, host *B* know that frames come from *A*.

Acknowledgment—A special ACK frame is used to complete entry in the switching table. a) The destination host *B* sends an acknowledgment to *switch3*.

- The ACK frame carries source and destination addresses so the switch knows which entry in the table is to be completed.
- The frame also carries VCI 77, chosen by the destination as the incoming VCI for frames from host *A*.
- Switch 3 uses this VCI to complete the outgoing VCI column for this entry. ○

77 is incoming VCI for destination *B* whereas for *switch3* it is outgoing VCI.

b) *Switch3* sends an acknowledgment to *switch2* that contains its incoming VCI in the table. Switch 2 uses this as the outgoing VCI in the table.
c) Similarly Switch 2 sends an acknowledgment to *switch1*. The process is the same and outgoing VCI is updated.
d) Finally *switch1* sends an acknowledgment to source *A*.
e) The source uses the incoming VCI from switch 1 as its outgoing VCI for the data frames to be sent to destination B.



Analysis

There is at least one RTT delay before data is sent due to setup request and acknowledgement

The per-packet overhead is reduced since VCI is a small number.

If a switch or link in a connection fails, the connection is teardown and a new one is setup

What is source routing?

All the information about network topology that is required to switch a packet across the network to the destination is provided by the source host.

The header contains an ordered list of intermediate hosts through which the packet must traverse.

For each packet that arrives on an input, the switch reads the port number in the header and transmits the packet on that output.

Source routing can be used in both datagram and virtual circuit networks

Explain Internetworking Protocol in detail.

Internetwork is interconnection of different *physical* networks to provide host-to-host packet delivery service.

- Internet is a *logical* network built on a collection of physical network.

The node that interconnects different networks is known as *router*.

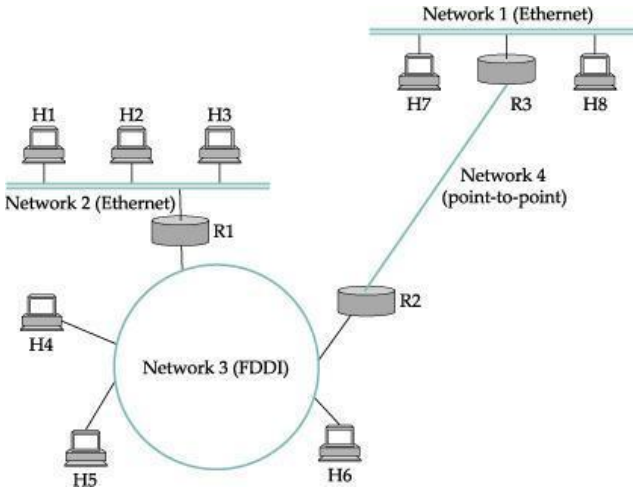
Internet Protocol (IP) is used to build scalable, heterogeneous internetworks.

The ability of IP to run over any networking technology is its strength.

IP provides neither error control nor flow control.

IP does nothing when a packet gets lost or corrupted. It is an unreliable service.

- If reliability is needed, IP must be paired with a reliable protocol such as TCP. The IP Service model has two parts
- *Datagram* (connectionless) model of data delivery
- *Addressing* scheme that identifies all hosts in the internetwork uniquely.



The above diagram shows an internetwork comprising different physical network such as Ethernet, FDDI ring and point-to-point link.

Each network has a set of hosts H_i and the networks are connected by set of routers R_j.

Datagram Forwarding

All IP datagram contain destination address to enable the network forward packets correctly in a connectionless manner.

Each router compares network id of the destination address with the network id of each of its interfaces.

- If a match occurs, then the destination lies on the same physical network as the interface, and the packet is *directly* delivered.
- Otherwise, the packet is forwarded to the *next hop* router after consulting its forwarding table.
- In case of no match, then the packet is forwarded to the *default router*.

Example

If *H1* sends a datagram to *H8*, then forwarding is as follows:

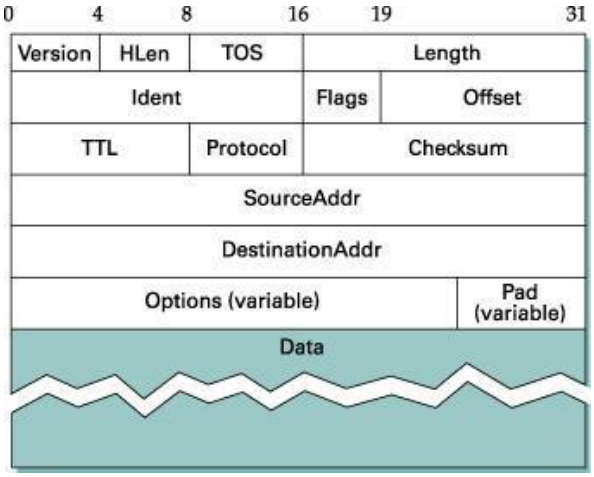
- *H1* sends datagram to its default router, say *R1*, since it cannot deliver directly.
- *R1* sends datagram to its default router, say *R2*, since H8 network id does not match any of its interface.
- *R2* forwards the datagram to *R3* based on its forwarding table shown below.

NetworkNum	NextHop
1	R3
2	R1

- R3 forwards the datagram directly to H8, since both are on the same network.

Packet Format

IPv4 datagram is a variable-length packet consisting of two parts, *header* and *data*. The header is 20–60 bytes and contains information essential to routing and delivery The minimum packet length is 20 bytes and maximum is 65,535 bytes.



Version specifies version of the IPv4 protocol, i.e. 4.

HLen defines length of the datagram header in 4-byte words. When there are no options, the value is 5 (5 × 4 = 20).

TOS allows packets to be treated differently based on application needs. The parameters used to distinguish are delay, throughput, reliability and cost

Length specifies the total packet length (header + data). The total length of the IPv4 datagram is restricted to 65,535 bytes (2^{16} - 1).

- If length is large for any lower layer protocols then fragmentation is done.

Ident a 16-bit identifier that uniquely identifies a datagram.

Flags It is a 3-bit field. The first bit is reserved. The second bit (D) is called the *do not fragment* bit. The third bit (M) is called the *more fragment* bit.

Offset shows relative position of this fragment with respect to the whole datagram. It is offset of the data in the original datagram measured in units of 8 bytes.

TTL defines lifetime of the datagram (default value 64) in hops.

- Each router decrements TTL by 1 before forwarding.
- If the value is zero, the datagram is discarded.

Protocol specifies the higher-level protocol such as (6–TCP, 17–UDP, 1–ICMP). Checksum contains 16-bit checksum for the packet header.

SourceAddr 32-bit address of the source host.

DestinationAddr 32-bit address of the destination host.

Options If *HLen* > 5 then options are specified (up to 40 bytes). Some options are:

- *Record Route* used to record the routers that handle the datagram.
- *Strict Source Route* used by the source to predetermine a route for the datagram.

Fragmentation

Each physical network has *Maximum Transmission Unit* (MTU), the largest IP datagram contained in a frame. MTU for FDDI is 4500, Ethernet is 1500, point-to-point is 532, etc.

The IP datagram is encapsulated in the physical network's frame through which it travels. If the datagram payload is greater than MTU, then it is *fragmented* to fit the link-layer frame. The fragmented packets are each of size MTU, except the last one.

If D flag bit is set, then datagram is not fragmented. If no alternate path is available, then it is discarded.

The router usually fragments the datagram, when it has to forward the packet over a network that has a smaller MTU. Each fragment is routed independently.

- A fragmented datagram may be further fragmented, if it encounters a network with a smaller MTU.

When a datagram is fragmented, the Ident field is copied to all fragments. The identification number helps the destination in reassembling the datagram.

On fragmentation the router changes three fields: Flags, Offset and Length.

The router sets the M bit in the flags field sets the Offset to 0 for the *first* fragment. For the *last* fragment M bit is not set.

IP does not attempt to recover from *missing* fragments and discards all other fragments. *Reassembly* is done at the receiving host and not at each router.

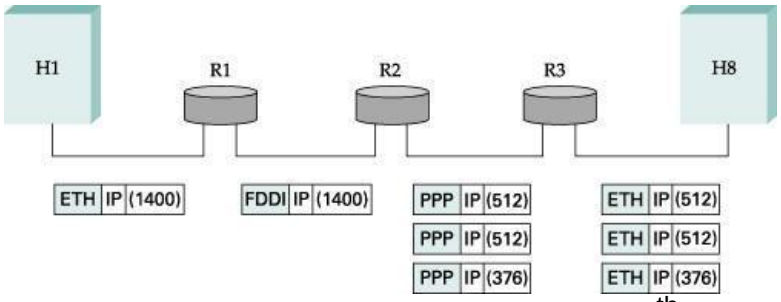
Example

Suppose host *H1* sends a datagram to host *H8* with a payload of 1400 bytes.

The datagram goes through the ETH and FDDI network without any fragmentation.

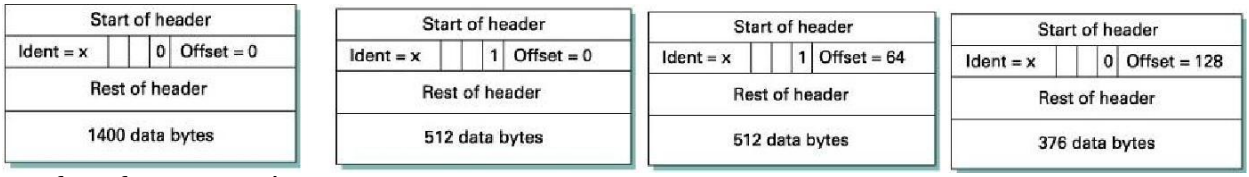
When the packet arrives at router *R2*, which has an MTU of 532 bytes, it is fragmented with a maximum payload of 512 (plus 20 bytes for IP header)

The three fragments are forwarded by router *R3* through Ethernet to the destination host.



The data carried in the second fragment starts with 513th byte, so the Offset field in this header is set to 64 (count of 8-byte chunks)

The third fragment contains the last 376 bytes of data, and Offset is set to 128.



Before fragmentation *After fragmentation at R2*

Global Addressing

IP addresses are hierarchical, i.e., it corresponds to hierarchy in the internetwork. IP addresses consist of two parts, *network id* and *host id*.

The network id identifies the network to which the host is attached.

- Hosts attached to the same network have the same network id in their IP address. The host id is used to uniquely identify a host on a network.

The routers have an address on each network, one for each interface.

IPv4 uses 32-bit addresses, i.e., address space is 2^{32} (more than 4 billion)

IPv4 address is expressed compactly as four octets (each in the range 0–255) in dotted decimal notation.

128.11.3.31

IPv4 Classful Addressing

In *classful* addressing, the address space is divided into five classes: A, B, C, D, and E.

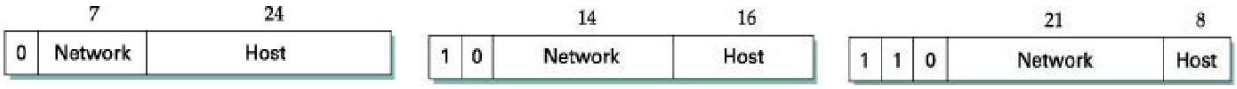
The class of an IP address is identified by seeing the MSBs in binary notation or first byte in decimal notation.

Class	Binary	Decimal	Application
A	0	0–127	Unicast
B	10	128–191	Unicast
C	110	192–223	Unicast
D	1110	224–239	Multicast
E	1111	240–255	Reserved

Classes A, B and C are used for *unicast* addressing.

Class D was designed for *multicasting* and class E is *reserved*.

Classes A, B, C have certain bits for the network part and rest for the host part i.e., networks belonging to a class and number of hosts attached to it are *fixed*.



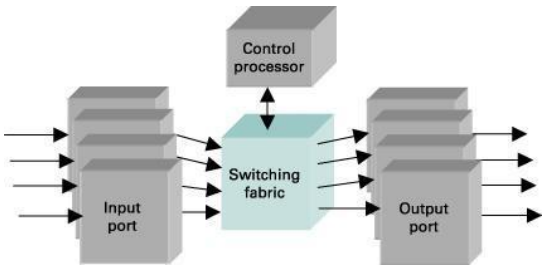
Class A *Class B* *Class C*

Class	No. of networks	No. of hosts per network	Designed for
A	126	$2^{24} - 2$	WAN
B	16,382	65,534	Campus networks
C	2^{21}	254	LAN

In c1assful addressing, a large part of the available addresses were wasted, since Class A and B were too large for most organizations.

Class C is suited only for small organization and reserved addresses were sparingly used.

State the components of a router



The control processor is responsible for implementing the routing protocols. The switching fabric transfers packets from one port to another

Routers are designed to handle variable-length packets

packetsize × *pps* = *linerate*, i.e, packet size at which the router can forward at line rate.

Detail the process of determining the physical address of a destination host (ARP).

A host or router to send an IP datagram, needs to know both the logical and physical address of the destination.

The destination IP address can be obtained from DNS host or forwarding table.

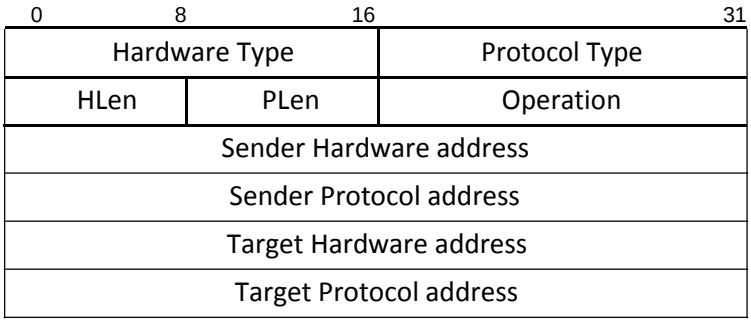
The physical address of the receiver is needed to pass through the physical network.

The Address Resolution Protocol (ARP) enables a source host to know the physical address of another node when the logical address is known.

ARP relies on broadcast support provided by physical networks such as Ethernet, Token ring, etc.

ARP enables each host on a network to build up a table of mapping between IP address and physical address.

Header Format



Hardware Type defines type of the physical network (1 for *ethernet*).

Protocol Type specifies the value of upper-layer protocol (8 for *IPv4*).

HLen specifies length of the physical address in bytes (6 for *Ethernet*).

PLen specifies length of the logical address in bytes (4 for *IPv4*).

Operation defines the type of ARP (1 for ARP request, 2 for ARP reply).

Sender Hardware address variable-length field contains physical address of the sender.

Sender Protocol address variable-length field contains logical address of the sender.

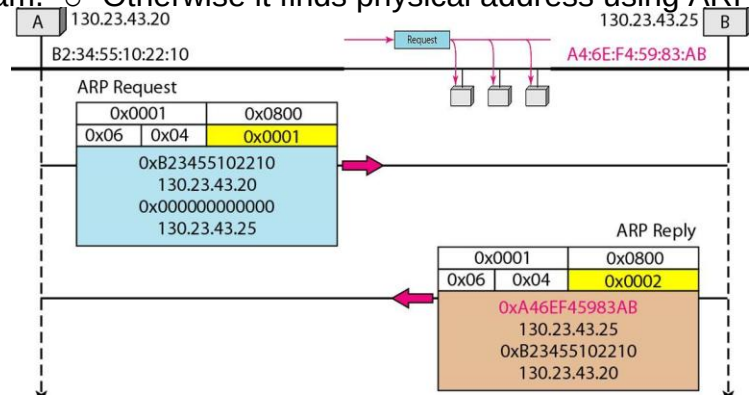
Target Hardware address variable-length field contains physical address of the target.

Target Protocol address variable-length field contains logical address of the target.

Address Translation

The host checks its ARP table with the logical address.

- If an entry exists, then corresponding physical address is used to send a datagram.
- Otherwise it finds physical address using ARP.



1. *ARP request* packet is created with Operation field set to 1.
2. The Target Physical address field is not known and *filled* with 0 (broadcast address).
3. The ARP request is encapsulated in IP packet and *broadcasted* on the physical network.
4. Each host takes note of sender's logical and physical address.
5. All nodes except the target node *discard* the packet.

6. The target node constructs an *ARP reply* packet with Operation set to 2.
7. ARP reply is *unicast*, sent back to the sender.
8. The sender receives the reply packet and *stores* target logical-physical address pair in its ARP table for sending future packets.
9. If target node *does not exist* on the same network, then ARP request is sent to the default router, which then forwards it to the next hop router and so on till destination.

ATMARP

ARP relies on broadcasting, whereas ATM network *does not* support broadcasting. *ATMARP* or Classical IP over ATM uses Logical IP Subnet (LIS).

The ATM network is divided into several *subnets*.

All nodes on the *same* subnet have the same network id.

Two nodes on the same subnet can *communicate* directly, whereas nodes on different subnets communicate via one or more routers.

Each node in the LIS is configured with ATM address of the ARP server to establish a *virtual circuit* to the ARP server when it boots.

The node sends a *registration* message that includes its IP and ATM address to the ARP server.

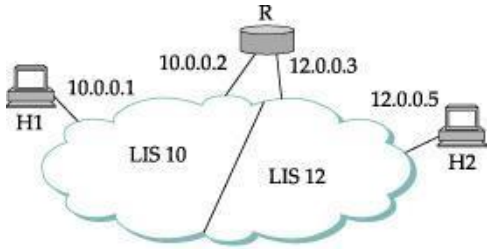
Thus ARP server builds the *database* of all node as <*IP address, ATM address*> pair.

Any node that wants to send a packet to some IP address *requests* the ARP server to provide the corresponding ATM address.

The ARP server performs a *lookup* operation and returns the ATM address.

The node can also maintain a *cache* of IP-to-ATM address mappings.

The source node establishes VC with the destination node and *sends* packets.



The above ATM network has two subnets.

- Host *H1* is connected to the router interface that connects to *LIS10*.
- Similarly *H2* is connected to interface *LIS12*.
- For hosts on different subnets (say from *H1* to *H2*), both hosts have to establish a virtual circuit to the router.

What is RARP? List the disadvantage.

A host knows its IP address from configuration file and physical address from its NIC.

A *diskless* workstation booted from its ROM or newly booted workstation does not know its IP address as it is assigned by the network administrator.

In such cases, Reverse Address Resolution protocol (RARP) allows the host to broadcast its physical address in a RARP request at the link-layer level.

The *RARP request* has a destination address of all 1s.

The RARP server receives the request, looks up the physical address in its configuration file and sends the IP address in a *RARP reply*.

RARP enables a host to know its logical address using its physical address.

If an administrator has several networks/subnets, then a RARP server is required for each network/subnet, since RARP broadcast is *not forwarded* by routers.

RARP is *replaced* by protocols such as BOOTP and DHCP.

Discuss the configuration of IP address to hosts automatically using DHCP.

Operating systems allow system administrator to *manually* configure IP address.
- Manual configuration is tedious and *error-prone* on any network.

Dynamic Host Configuration Protocol (DHCP) enables *auto* configuration of IP address to hosts using DHCP.

The drawback is it is difficult to identify a malfunctioning host.

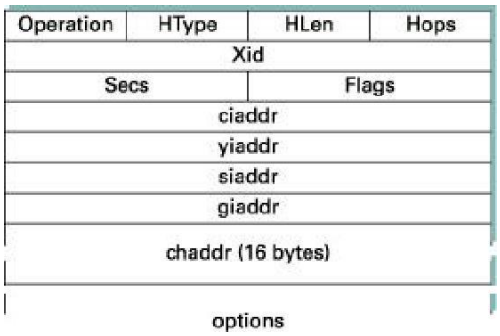
DHCP is derived from Bootstrap Protocol (BOOTP) and is *connectionless*.

The UDP *port* for sending data to server is 67 and port 68 for sending data to client.

DHCP provides both *static* (manual) and *dynamic* (automatic) address allocation.

For static allocation, a DHCP server has a manually created static database that binds physical address to IP address.

Header Format



Operation specifies type of DHCP packet.

HType value for type of the physical network (1 for ethernet). HLen length of the physical address in bytes (6 for ethernet). Xid specifies the transaction id.

ciaddr specifies client IP address in case of DHCPREQUEST

yiaddr this field is known as *your IP address*, to be filled by DHCP server. siaddr contains IP address of the DHCP server.

giaddr contains IP address of the Gateway or relay agent.

chaddr contains hardware (physical) address of the client.

Dynamic Address Allocation

Dynamic allocation is required when a host moves from one network to another or else is connected / disconnected from a network.

The administrator provides DHCP server a range of unassigned addresses to be assigned to hosts on demand.

To contact DHCP server, a booted/attached host broadcasts a DHCPDISCOVER message with IP address 255.255.255.255 encapsulated in a *UDP* packet.

The DHCP server *checks* its static database first.

- If the lookup is successful, the corresponding IP address is returned.

Otherwise, the server *selects* an unassigned IP address based on client's MAC address.

- Fills the selected address in yiaddr field and adds an entry to *dynamic* database.

- DHCP Server sends DHCPOFFER message containing Client IP and MAC address, server IP address and options (lease duration, default route, DNS server, etc.)

There can be multiple DHCP server on a network but the client accepts only one offer. The client broadcasts a DHCPREQUEST message requesting the offered address.

Based on transaction id, the corresponding DHCP server sends an acknowledgement as a DHCPACK containing the requested configuration.

When the lease expires, the client renews the lease.

- The server either *agrees* or *disagrees* with the renewal.

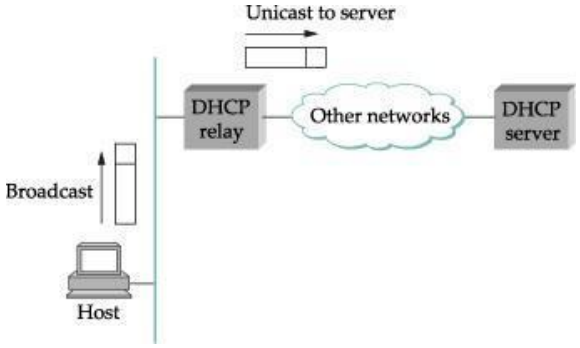
DHCP relay

DHCP is an *application* layer protocol.

- Both the server and client *need not* exist on the same network.

A DHCP *relay* agent receives broadcast message from the client.

- Stores it's address in giaddr and is sent as *unicast* to DHCP server.
- The DHCP server's response is sent to the relay agent, which is retransmitted back to the client.



***Write short notes on error reporting using ICMP*.**

The IP protocol is a best-effort delivery service.

- It has no error-reporting or error-correcting mechanism and also lacks mechanism for host and management queries.

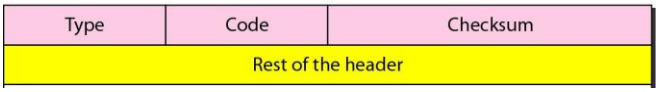
Internet Control Message Protocol (ICMP) is designed to handle these lacunae.

ICMP control messages are either used to *report errors* to the source host or used to *diagnose* network problems.

An ICMP message is *encapsulated* within an IP packet.

Debugging tools such as ping and traceroute use ICMP messages internally.

Header Format



Type 8 bit field that specifies type of the ICMP message.

Code 8-bit field that specifies the sub-type for the given type. Checksum contains 16-bit checksum sequence.

Rest of Header varies based on Type and Code field.

Control Messages

Destination Unreachable When a router *cannot route* a datagram, the datagram is discarded and sends a destination unreachable (Type = 3) message.

Source Quench When a router or host discards a datagram due to *congestion*, it sends a source-quench (Type = 4) message. This message acts as flow control.

Time Exceeded When TTL field becomes 0, the router discards the datagram and a time-exceeded (Type = 11) message is sent to the source host.

Parameter Problem If a router discovers ambiguous or *missing* value in any field of the datagram, it discards the datagram and sends parameter problem (Type = 12) message.

Redirection Redirect messages (Type = 5) are sent by the default router to inform the source host to *update* its forwarding table when the packet is routed on a wrong path.

Echo Request & Reply The combination of echo-request (Type = 8) and echo-reply (Type = 0) messages determines whether two systems can *communicate* at the IP level.

Timestamp Request & Reply Two machines can use the timestamp request (Type = 13) and timestamp reply messages (Type = 14) to determine the *round-trip time* (RTT).

Address Mask Request & Reply A host to obtain its subnet mask, sends an address mask request (Type = 17) message to the router, which responds with an address mask reply (Type = 18) message.

Router Solicitation & Advertisement For the host to know if the routers are functioning, it can broadcast a router solicitation (Type = 10) message. The router then broadcast its routing information using the router advertisement (Type = 9) message.

Discuss the various queuing methods in detail.

Routers have finite buffer space.

When a packet arrives, it is placed at *rear* end of queue at the router's buffer space. The packet at *front* of the queue is taken out of the queue for forwarding.

The common queuing algorithms are:

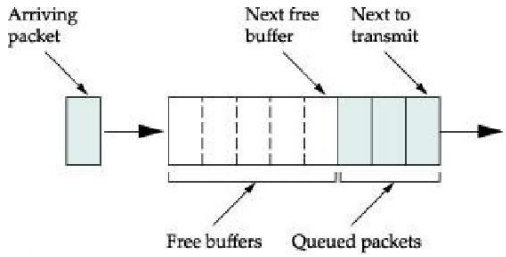
- First-In-First-Out (FIFO) Queuing
- Priority Queuing
- Fair Queuing
- Waited Fair Queuing

FIFO Queuing

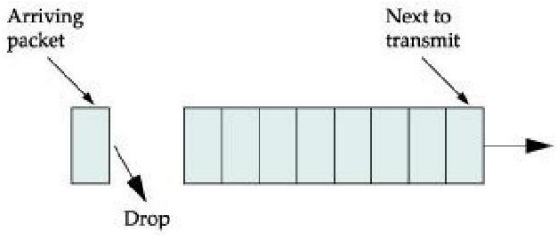
The first packet that arrives at a router is the first packet to be forwarded i.e., FIFO.

The router discards any packet that arrives when the queue is full. This is known as *Tail drop* since packets arriving at tail end of queue is dropped.

Thus FIFO queuing is a combination of FIFO scheduling discipline and Tail drop policy.



FIFO queuing



Tail Drop

Analysis

Simple to implement and is widely used.

Packets are *dropped* without regard to its flow type or importance.

Does not help in *congestion control* and it is left to TCP at the end hosts.

Priority Queuing

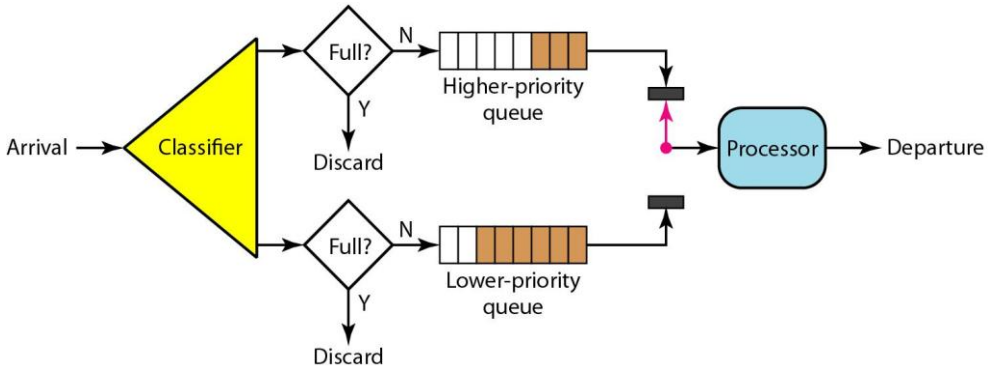
Priority queuing is a variation of FIFO queuing

Each packet is marked with a *priority*. The priority can be set in TOS field of IP header. Routers have a FIFO queue, one for each type of *priority*.

The router always forwards packets out of the *highest priority* queue. If that queue is empty, then packets in the next high priority queue is taken for processing.

Packets in the lowest priority queue are processed *last*.

The network can *charge* more to deliver high-priority packets than low-priority ones.



Analysis

A priority queue can provide better *QoS* than FIFO queue because high priority traffic such as multimedia, can reach the destination with less delay.

Routing updates after a topological change is marked in TOS field, helps in *stabilization* of routing tables.

The potential drawback is that packets in lower-priority queues may never be processed, if there is a continuous flow in high-priority queues. This condition is called *starvation*.

Fair Queuing (FQ)

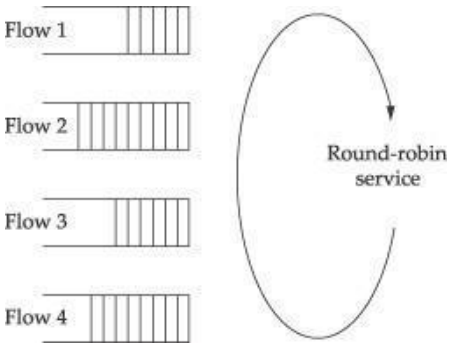
Fair Queuing addresses the problems of FIFO queuing such as non-discrimination of traffic sources and lack of congestion-control.

In fair queuing, a *separate queue* is maintained for each type of flow. Router services these queues in a *round-robin* manner.

When a flow's queue gets filled up, further packets are *discarded*. All flows have a *fair share* of the bandwidth.

FQ segregates traffic so that ill-behaved traffic sources do not *interfere* with the legitimate traffic sources.

FQ enforces fairness among a collection of flows managed by a well-behaved *congestion control* algorithm.



Round-robin servicing

Round-robin servicing needs to be done in terms of bit-by-bit, but interleaving bits from different packets is not feasible.

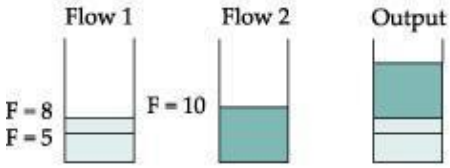
FQ simulates bit-by-bit RR by first determining when a given packet would finish being transmitted and then using it to sequence the packets for transmission as follows:

- Let P_i denote the length of packet *i*
- Let S_i denote the time when the router starts to transmit packet *i*
- Let F_i denote the time when the router finishes transmitting packet *i* ($F_i = S_i + P_i$)

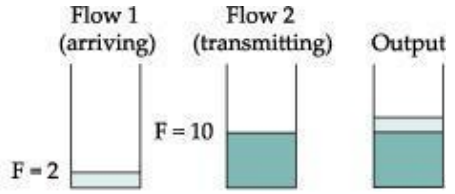
A packet can be transmitted after its arrival time A_i and not before its predecessor i-1 has been transmitted. Hence, $S_i = \max(F_{i-1}, A_i)$ and $F_i = \max(F_{i-1}, A_i) + P_i$

The packet with the lowest F_i timestamp is the next to be transmitted.

A newly arriving packet cannot preempt a packet that is currently being transmitted.



Shorter packets sent first



Non-preemptive of packet in progress

Analysis

The link is never idle as long as there is at least one packet in any of the flow. This characteristic is known as *work-conserving*.

If there are *n* flows, then a flow cannot use more than 1/*n* of the total bandwidth.

If some flows are empty, then their bandwidth is *shared* amongst the available flows.

Weighted Fair Queuing (WFQ)

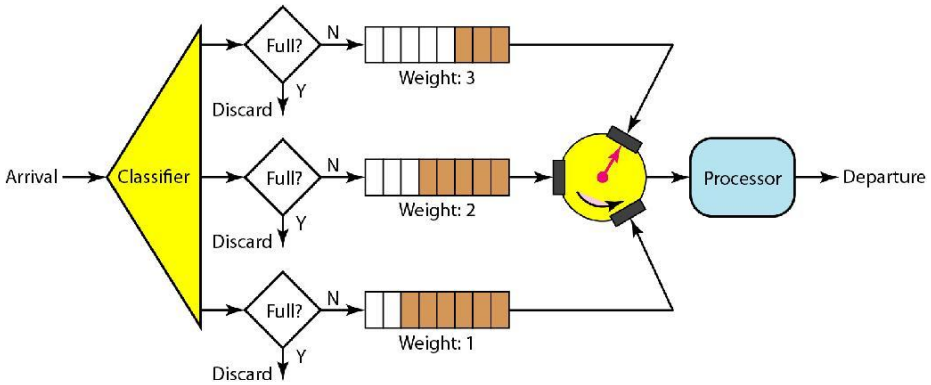
Weighted Fair Queuing is a variation of fair queuing.

In WFQ, each flow is assigned a *weight*, whereas FQ gives each queue a weight of 1.

The *weight* specifies how many bits to transmit each time the router services that queue The weight also implies the *percentage* of the link's bandwidth that flow will get.

Packets are *assigned* different classes and admitted to different queues based on their priority.

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.



In above example, three packets are processed from the first queue, two from the second queue, and one from the third queue.

Classify routing protocols

Intra-domain routing
- Distance vector routing (eg. RIP)
- Link state routing (eg. OSPF) Inter domain routing
- Path vector (eg. BGP)

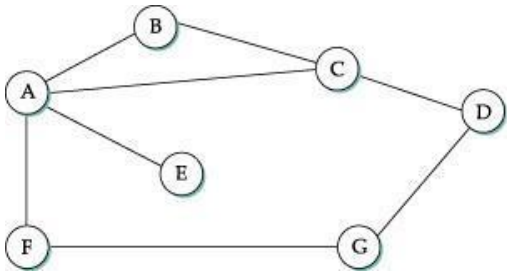
Explain distance vector routing in detail with an example.

Each node *knows* the distance (cost) to each of its directly connected neighbors.
Hosts that are not directly connected or if link is down, is assigned *infinite* cost.
Each node *constructs* a vector containing (*Destination*, *Cost*, *Next Hop*) to all other nodes and distributes to its neighbors.
Each node computes a vector (table) of *minimum* distance (cost) to every other node using the information from its neighbors.
Thus the table at each node guides a packet to the desired node by showing the *Next Hop*.

Initial State

For the given network, each node sets a distance of 1 (hops) to its immediate neighbors.
The distance for non-neighbors is marked as *unreachable* with value (infinity).

The initial routing table stored at *A* is



Destination	Cost	Next Hop
B	1	B
C	1	C
D		—
E	1	E
F	1	F
G		

Sharing & Updation

Each node *shares* its cost list (distance) to all of its directly connected neighbors. Node *A* receives distance vectors from *B*, *C*, *E* and *F*.
- For example the tables received by *A* from *C* and *F* are:

Destination	Cost	Next Hop
A	1	A
B	1	B
D	1	D
E		
F		
G		

Destination	Cost	Next Hop
A	1	A
B		
C		
D		
E		
G	1	G

Now node *A* can use information from its neighbors to *reach* other unreachable nodes. For example, node *F* tells node *A* that it can reach node *G* at a cost of *1*.

Each node updates its routing table by comparing with its neighbor tables as follows
- For each destination Total Cost is computed as:
  Total Cost = Cost(*Node*, *Neighbor*) + Cost(*Neighbor*, *Destination*).
- If Total Cost < NodeCost(*Destination*) then
  NodeCost(*Destination*) = Total Cost and Next Hop(*Destination*) = *Neighbor*

For example, *A* compares its table with *C*'s table

- Total Cost for *B* = Cost(*A*, *B*) + Cost(*B*, *C*)
  = 2 Since 2 > 1, there is no change
- Total Cost for *D* = Cost (*A*, *C*) + Cost(*C*, *D*) = 1 + 1 = 2.
  Since 2 < , entry for destination *D* in *A*'s table is changed to (*D*, 2, *C*) ○ Similarly other entries are checked and there is no change.

In a similar manner, *A* updates its routing table using information from *B*, *E* and *F*. The final routing table at *A* is

Destination	Cost	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Each node builds complete routing table after few exchanges with its neighbors.

The process of obtaining complete routing information to all nodes is called *convergence*. The sharing & updation process take place *periodically* and in case of *triggered* update. Periodic updation is normally done every 30 seconds.

Triggered Update

A node can test link status by using *hello* (control) packets.

Alternatively a link or node failure is presumed, if it does not receive periodic updates from its neighbor for a while.

This forces the node to *update* its neighbors, neighbors update their neighbors and so on. Assume that *F* detects that its link to *G* has failed.

- *F* sets its new distance to *G* as and shares its table with *A*. ○ Node *A* updates its distance to *G* as .
- Node *A* also receives periodic update from *C* with distance to *G* as 2. ○ Node *A* updates its distance to *G* as 3 through *C*.

Loop Instability

Suppose link from node *A* to *E* goes *down*.

A advertises a distance of *in nity* to *E*, meanwhile *B* and *C* advertise a distance of 2 to *E.* ○ *B* using information from *C*, concludes that *E* can be reached in 3 hops through *C*. ○ *B* advertises this to *A*, and *A* in turn updates *C* with a distance of 4 hops to *E*.

- Now node *C* advertises with a distance of 5 to *E* and so on.

Thus the nodes update each other until cost to *E* reaches a large number, say *infinity*.

Thus convergence does not occur. This problem is known as *loop instability*.

Solutions

Infinity is redefined to a small number. Most implementations define 16 as infinity. ○ Distance between any two nodes should not exceed 15 hops.

- Thus distance vector routing *cannot be used* in large networks.

When a node updates its neighbors, it does not send those routes it learned from each neighbor back to that neighbor. This is known as *split horizon*.

- For example, if *B* has the route (E, 2, A) in its table, then it does not include the route (E, 2) in its update to *A*.
- Continued *absence* of route update for a destination leads to deletion of its entry.

In *split horizon with poison reverse*, Node *B* can still advertise the value of (E, 2) to *A*, but with a warning message.

- This approach *delays* the convergence process and does not work well for large number of nodes.

Routing Information Protocol (RIP)

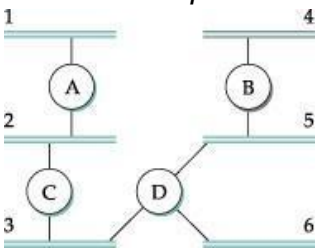
RIP is an intra-domain routing protocol used inside an autonomous system based on distance-vector algorithm.

It is extremely simple and widely used, since it was distributed with Unix BSD.

The routers advertise the cost of reaching networks, instead of reaching other routers. RIP takes the simplest approach, with all link costs being equal to 1.

The distance is defined as the number of links to reach the destination.

- The metric in RIP is called a *hop count*.



In example, Router *C* advertises to *A* that it can reach

- Networks *2* and *3* at a cost of 0 (*directly connected*),
- Networks *5* and *6* at cost 1 and network *4* at cost 2.

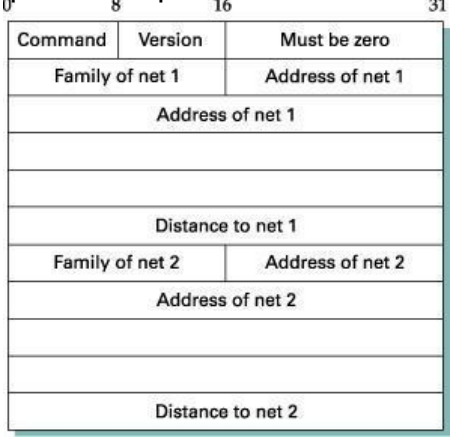
As in distance vector routing algorithm, a router updates cost and next hop information for each network number.

Infinity is defined as 16, i.e., any route in an AS using RIP cannot have more than 15 hops. It is limited to run on only smaller networks.

Routers running RIP send their advertisements every 30 seconds or when it is updated by another router

Packet Format

RIP packet format contains mostly (*network address*, *distance*) pair as shown. ○ RIP supports multiple address families that include IP.



Explain link state routing with an example.

Each node knows the *state* of link to its neighbors and the cost involved. Link-state routing protocols rely on two mechanisms:

- Reliable dissemination of link-state information
- Route calculation from the accumulated link-state knowledge

Reliable Flooding

Reliable flooding is the process of ensuring all nodes having a copy of the link-state information from all other nodes

Each node creates an update packet called the *link-state packet* (LSP) containing

- ID of the node
- List of directly connected neighbors of that node and cost to each one
- Sequence number
- Time to live

Each node *sends* its LSP out on each of its directly connected links.

Transmission of LSPs between adjacent routers is made reliable using acknowledgment. When a node receives LSP from a neighbor, it checks to see whether it has a copy.

- If not, store and forward the LSP on all other links except the incoming one.
- Otherwise, if the received LSP has a *bigger* sequence number, then it is stored and forwarded. The older one is *discarded*.

Since a node passes the recent LSP to its neighbors, which in turn forwards to their neighbors, the recent LSP eventually reaches all nodes.

LSP is generated either periodically or when there is a change in the topology.

Example

LSP arrives at node *X*, which sends it to neighbors *A* and *C*. *A* and *C* do not send it back to *X*, but send it to *B*.

Since *B* receives two identical copies of the LSP, it accepts one that comes first and discards the other.

B passes the LSP on to *D*. Since *D* has no neighbors to flood, the process is complete.

Reducing Overhead

Flooding creates traffic and is overhead on the network. Mechanisms to reduce are:

1. *Timer* using long timers, in terms of hours for periodic generation.
2. *Sequence number* 64-bit sequence numbers do not wrap around soon and is used to discard old LSPs.
3. *Time to live* A router decrements TTL before forwarding a LSP. When TTL reaches 0, the node refloods the LSP. All nodes delete their stored LSP for that ID.

Route Calculation

Once a node has copy of the LSP from every other node, it knows the entire network.

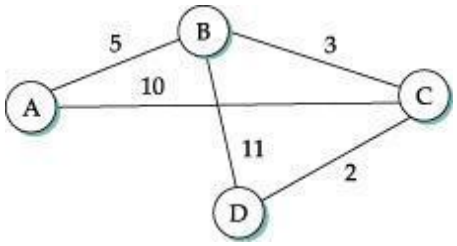
Each node computes its routing table directly from the LSPs using a variation of *Dijkstra* algorithm called *forward search* algorithm

Each node maintains two lists namely Tentative and Confirmed.

Each of these lists contains a set of entries of the form (*Destination*, *Cost*, *NextHop*)

Forward Search algorithm

1. Initialize the Confirmed list with an entry for the *Node* with a cost of 0.
2. For the node just added to Confirmed list, call it node *Next* and select its LSP.
3. For each neighbor of *Next*:
   a. Calculate cost to reach *Neighbor* as Cost(*Node*, *Next*) + Cost(*Next, Neighbor*).
   b. If *Neighbor* is currently on neither Confirmed nor Tentative list, then add (*Neighbor*, *Cost*, *NextHop*) to Tentative list.
   c. If *Neighbor* is currently on Tentative list, and the *Cost* is less than currently listed cost for *Neighbor*, then replace the current entry with (*Neighbor*, *Cost*, *NextHop*), where *NextHop* is the direction to reach *Next*.
4. If the Tentative list is empty then *Stop*.
   a. Otherwise, pick the entry from the Tentative list with the lowest cost, move it to the Confirmed list, and go to *Step 2*.



For the given network, the process of building routing table for node *D* is tabulated

Step	Confirmed	Tentative	Comment
1	(D, 0, –)		*D* is moved to Confirmed list initially
2	(D, 0, –)	(B, 11, B) (C, 2, C)	Based on D's LSP, its immediate neighbors *B* and *C* are added to Tentative list
3	(D, 0, –) (C, 2, C)	(B, 11, B)	The lowest-cost member *C* of Tentative list is moved onto Confirmed list. *C*'s LSP is to be examined next.
4	(D, 0, –) (C, 2, C)	(B, 5, C) (A, 12, C)	Cost to reach *B* through *C* is 5, so the entry (B,11,B) is replaced. *C*'s neighbor *A* is also added to Tentative list
5	(D, 0, –) (C, 2, C) (B, 5, C)	(A, 12, C)	The lowest-cost member *B* is moved to the Confirmed list. *B*'s LSP is to be examined next
6	(D, 0, –) (C, 2, C) (B, 5, C)	(A, 10, C)	Since *A* could be reached *B* at a lower cost than the existing one, the Tentative list entry (A,12,C) is replaced to (A,12,C).
7	(D, 0, –) (C, 2, C) (B, 5, C) (A, 10, C)		The lowest-cost and only member *A* is moved to Confirmed list. Processing is over.

Analysis

Link-state routing stabilizes quickly without generating much traffic and responds to changes in topology dynamically.

The amount of information stored (a LSP for each node) is large.

Open Shortest Path First Protocol (OSPF)

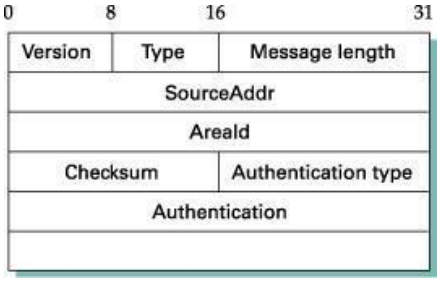
OSPF is one of the most widely used link-state routing protocols.

Authentication of routing messages Misconfigured hosts are capable of bringing down a network by advertising to reach every host with the lowest cost 0. Such disasters are averted by mandating routing updates to be authenticated.

Additional hierarchy In OSPF, a domain is partitioned into areas, i.e., a router need not know the complete network, instead only its area.

Load balancing OSPF allows multiple routes to the same place to be assigned the same cost and will cause traffic to be distributed evenly over those routes.

OSPF Header



Version represents the current version, i.e., 2.

Type represents the type value (1–5) of OSPF message.

- Type 1 also known as *hello* message to find out whether its neighbors are alive.
- Other types are used to *request*, *send* and *acknowledge* link-state messages. SourceAddr identifies the sender of the message

AreaId 32-bit identifier of the area in which the node is located Checksum 16-bit checksum

Authentication type has value 0 if no authentication is used, 1 for simple password and 2 for cryptographic authentication checksum.

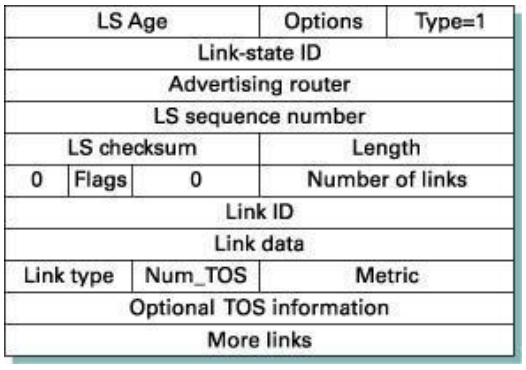
Authentication contains password or cryptographic checksum

Link State Advertisement (LSA)

The *basic* of OSPF message is the Link State Advertisement.

- A message can contain *multiple* LSAs.

LSA is used by routers to *advertise* the networks that are directly connected to it.
It is also used by routers to advertise the cost of reaching it over a link from other routers.



LS Age is incremented at each node until it reaches a maximum

Type defines type of LSA. *Type1* LSAs advertise the cost of links between routers.

Link-state ID 32-bit identifier that identifies the router.

Advertising router For type 1 LSA, it is same as Link-state ID

LS sequence number used to detect old or duplicate packets LS checksum covers all fields except LS Age

Length length of the LSA in bytes

Link ID and Link Data identify a link

Metric specifies cost of the link.

Link Type specifies type of link (for example, point-to-point)

TOS allows OSPF to choose different routes based on the value in TOS field

What are subnets and why are they required? Explain routing using subnets.

Classful Address Depletion

The network part of IPv4 is used to identify a *single* physical network. A *smaller* network requires a class C address.

For networks with *more than* 255 hosts, class B address is required.

Class B address are sought after in anticipation of more hosts to be *added* in the future.

IPv4 address space is *exhausted* in the process of assigning one per physical network.

At most *253* addresses can go *unused* in a class C network whereas over *64,000* addresses can go unused in a class B network.

This results in *inefficient* usage of the available address space.

Increase in network numbers, increases size of forwarding tables and degrade router performance.

Subnetting

Subnetting reduces the total number of network numbers by assigning a single network number to many adjacent physical networks.

Each physical network is referred to as *subnet*.

For subnetting, the subnets must be close to each other.

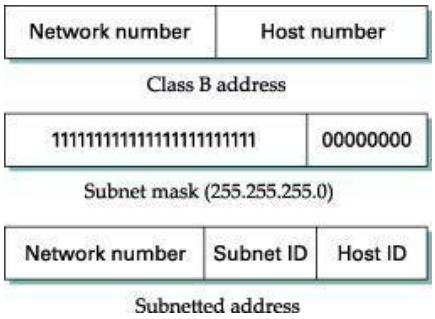
- For example, each department having a network within a college campus with a router connecting to the external world.

All nodes on a subnet is configured with a *subnet mask.*

Subnet mask introduces a level of hierarchy into IP address. It is written like a IP address (for example 255.255.255.0)

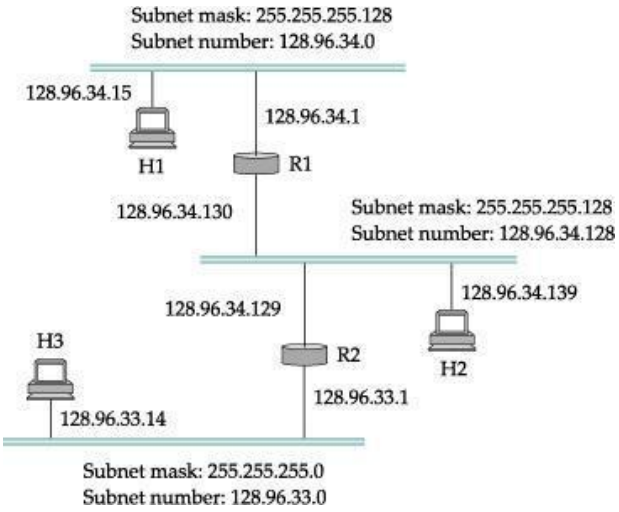
The bitwise AND of IP address and its subnet mask gives the *subnet number*.

Thus all nodes have the same *subnet number*, i.e., hosts on different physical network share a single network number.



Thus subnetting provides efficient usage of available address space by assigning a single network number amongst multiple adjacent physical networks.

Routing



When the host wants to send a packet to another host, it performs a bitwise AND between its own subnet mask and the destination IP address.

- If the result equals its own subnet number, then the packet is delivered directly over the subnet.
- Otherwise, the packet is sent to a router to be forwarded to another subnet.

For example, when H1 sends a packet to H2 in the above given network, then:

- H1 performs bitwise AND (255.255.255.128, 128.96.34.139) which is equal to 128.96.34.128
- This does not match the H1's subnet number 128.96.34.0
- Therefore H1 sends the packet to the default router R1

Routing Table

To support subnetting, entries in routing table are of the form (SubnetNumber, SubnetMask, NextHop)

To perform a lookup, the router performs a AND (destination address, SubnetMask) for each entry in the table.

If the result matches the SubnetNumber for an entry, then the packet is forward to the corresponding NextHop router

The outer world sees the collection of subnets as a single network and has only one entry in the forwarding table for all the subnets.

Routers within the campus must be able to route packets to the right subnet.

R1's forwarding table is as follows.

SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2

When H1 sends a datagram to H2, R1 receives the datagram from H1.

- It ANDs the H2 address 128.96.34.139 with SubnetMask and compares the result with SubnetNumber for each entry in the table.

- The result matches for the second entry. Thus the packet is delivered to H2 through Interface 1

Forwarding Algorithm

```
D = destination IP address
for each forwarding table entry (SubnetNumber, SubnetMask, NextHop)
   D1 = SubnetMask & D
   if D1 = SubnetNumber
        if NextHop is an interface
           deliver datagram directly to destination
        else
           deliver datagram to NextHop (a router)
```

Write short notes on CIDR.

Subnetting helps in address assignment, but does not prevents an organization go for a class B address, anticipating number of hosts could go beyond 255.

Exhaustion of address space centers on exhaustion of class B address.

If class C addresses were given, then number of entries in the routing table gets larger.

The address efficiency in class C can be as low as 0.78% (2/55) and in class B can be as low as 0.39% (256/65535).

Classless Interdomain Routing (CIDR) tries to balance between minimize the number of routing table entries and handling addresses space efficiently.

CIDR aggregates routes, by which an entry in the forwarding table is used to reach multiple networks.

Example1

Consider an autonomous system (AS) with 16 class C networks.

Instead of providing 16 class addresses at random, a block of contiguous class C address is given. For example, from 192.4.16 to 192.4.31

The bitwise analysis shows 20 MSBs (11000000 00000100 0001) are the same for that block, i.e., a 20-bit network id.

The 20-bit network number supports hosts that range between class B and C address.

Thus higher address efficiency is achieved by providing small chunks of address, smaller than class B network and a single network prefix to be used in forwarding table.

Restrictions

The addresses in a block must be contiguous.

The number of addresses in a block must be a power of 2.

The first address must be evenly divisible by the number of addresses.

A protocol such as BGP is required to support classless addressing.

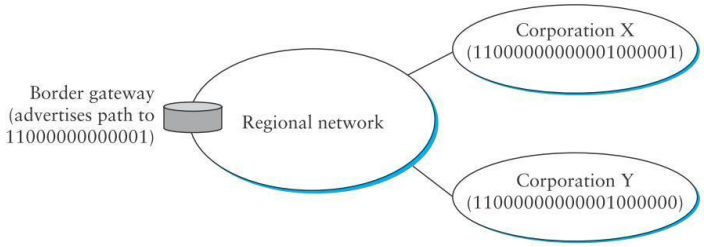
- The network number is represented as *<length, value>* pairs

Route Aggregation

Consider the case of an ISP to provide internet connectivity to a large number of corporation and campuses.

In example, two corporations served by the ISP is assigned adjacent 20-bit network prefixes.

Since both of them are reachable through ISP, the ISP advertises a 19-bit common prefix that both share.



What is an autonomous system?

Internet is so large that no one routing protocol can handle the task of updating the routing tables of all routers. Internet is divided into autonomous systems.

An autonomous system is a group of networks and routers under the authority of a single administration.

Routing inside an autonomous system is referred to as intra-domain routing. Routing between autonomous systems is referred to as inter-domain routing.

What is interdomain routing?

The interdomain routing involves AS sharing their reachability information with each other AS.

The goal of interdomain routing is *reachability* and not optimality.

The two major interdomain routing protocols are Exterior Gateway Protocol (*EGP*) and Border Gateway Protocol (*BGP*).

What are the problems in interdomain routing?

An internet backbone must be able to route packets to any destination, i.e., there should be a match in the routing/forwarding table.

Each AS has its own intradomain routing protocols and chooses the metric assigns to path. This varies from one AS to another.

Autonomous systems may not trust each other.

Write short notes on BGP.

Border Gateway Protocol (BGP) is an inter-domain routing protocol using path vector routing

Traffic on the internet can be classified into two types:

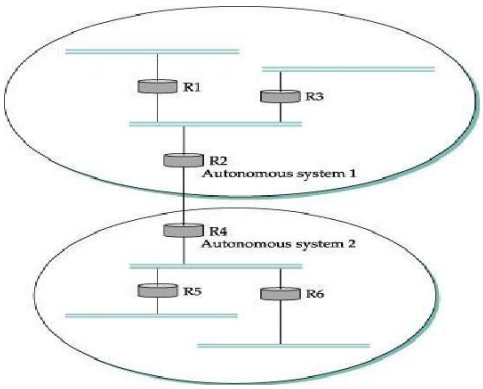
- *local* traffic that starts/ends on nodes within an AS
- *transit* traffic that passes through an

AS AS can be classified into three types

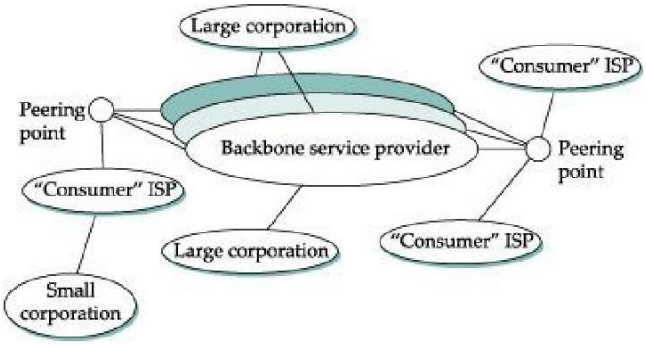
- *Stub AS* has only a single connection to one other AS. This AS can carry local traffic only, such as Small corporation.
- *Multihomed AS* has connections to more than one other AS but refuses to carry transit traffic, such as Large corporation.
- *Transit AS* has connections to more than one other AS and is designed to carry both transit and local traffic, such as the backbone providers

Each AS selects one of its nodes to be the *BGP speaker*.

Speaker node creates a routing table for that AS and advertises it to other BGP speakers in the neighboring ASs.



Network of autonomous systems



Multi backbone internet

Each AS also has a *border gateway* through which packets enter and leave the AS.

BGP advertises complete paths as an enumerated list of ASs to reach a particular network. BGP ensures that paths are *loop-free*.

The attributes in a path can be *well known* or *optional*. The well known attributes are recognized by all routers.

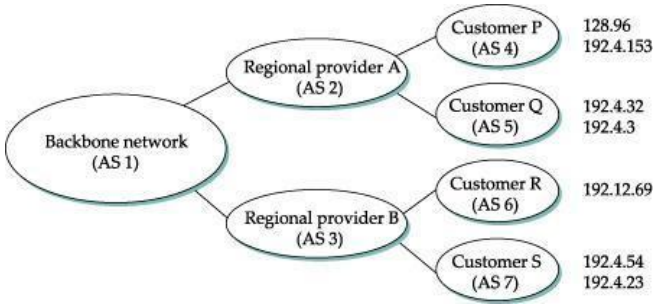
If there are different routes to a destination, the BGP speaker chooses the best one according to local policies, and then advertises.

A BGP speaker need not advertise any route to a destination, even if it has one.

Example

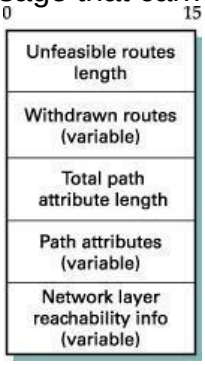
The BGP speaker for provider *A* (*AS2*) advertises that the networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached directly from *AS2*.

The backbone network, on receiving this advertisement, advertises that networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached along the path (*AS1*, *AS2*).



BGP speakers can cancel previously advertised paths if a critical link or node on a path goes down. This negative advertisement is known as *withdrawn* route.

The format of BGP-4 update message that carries advertisement is shown below



BGP Sessions

The exchange of routing information between two routers takes place in a BGP session. To create a reliable environment, BGP uses the services of TCP.
The routes need not be repeatedly sent, if there is no change. This is done by sending *keep alive* messages.

Two types of BGP session are *external* BGP (E-BGP) and *internal* BGP (I-BGP).

- E-BGP is used to exchange routing information between two speaker nodes belonging to two different ASs.
- I-BGP is used to exchange routing information between two routers inside an AS.

Discuss the notation, representation and address space of IPv6.

CIDR and subneting could not solve the address exhaustion faced by IPv4. IPv6 was evolved to solve this problem.

The striking features of IPv6 are:

- support for real-time services
- security support
- auto configuration
- enhanced routing functionality, including support for mobile hosts

Addresses Space

IPv6 provides a 128-bit address space to handle up to 3.4×10^{38} nodes.

IPv6 addresses do not have classes, but classification is based on the leading bits. The IPv4's classes A, B and C start with 001 prefix (unicast addresses).

Multicast address (start with a byte of 1s) serves the purpose of class D address.

Large chunks of address space are left *unassigned* to allow for new features in the future.

Common Prefixes

Prefix	Usage
0000 0000	Reserved
0000 001	Reserved for ISO protocol
0000 010	Reserved for Novell network layer
001	Aggregated Global Unicast Addresses (Class A, B and C)
010	Provider-based unicast addresses
100	Geographic-based unicast addresses
1111 1110 10	Link local use addresses
1111 1110 11	Site local use addresses
1111 1111	Multicast addresses

Address Notation

The standard representation is x:x:x:x:x:x:x:x where x is a hexadecimal representation of a 16-bit address separated by colon (:) as shown below

47CD:1234:4422:ACO2:0022:1234:A456:0124

An IPv6 address with a large number of contiguous 0s is written compactly by omitting the 0s (47CD:0000:0000:0000:0000:0000:A456:0124 is written as 47CD::A456:0124)

An IPv4 address can be mapped to IPv6 address by prefixing the 32-bit IPv4 address with 2 bytes of all 1s and then zero-extending the result to 128 bits.

128. 96.33.81 is written as ::FFFF:128.96.33.81

Address Aggregation

The goal of the IPv6 address allocation plan is to provide aggregation of routing information to reduce the burden on intradomain routers.

Aggregation is done by assigning prefixes at continental level.

Continental boundaries form natural divisions in the Internet topology

- For example, if all addresses in Europe have a common prefix, then routers in other continents would need one routing table entry for all networks in Europe.

The format for provider-based unicast address aggregation is shown below.

- RegistryID contains identifier assigned to the continent. It is either INTERNIC (North America), RIPNIC (Europe) or APNIC (Asia and Pacific)
- ProviderID variable-length field identifies the provider for Internet access such as an ISP.
- SubscriberID specifies the assigned subscriber identifier
- SubnetID defines a specific subnet under the territory of subscriber.
- InterfaceID contains the link level or physical address.

Addressing

Multicast address as in IPv4 is used to address a group of hosts.

IPv6 also defines *Anycast* addresses. A packet destined for an anycast address is delivered to only one member of the anycast group (the nearest one).

Reserved addresses start with prefix of eight 0s. It is classified into

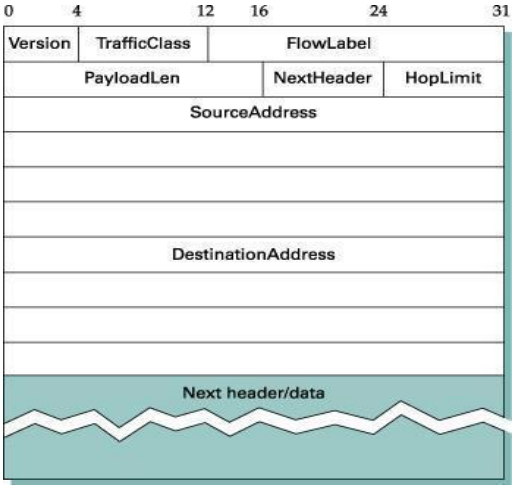
- *unspecified* address is used when a host does not know its address
- *loopback* address is used for testing purposes before connected to network
- *compatible* address is used when IPv6 hosts communicate through IPv4 network
- *mapped* address is used when a IPv6 host communicates with a IPv4 host.

IPv6 header defines *Local* addresses for private networks. It is classified into

- *Site* local address for use in a isolated site with several subnets.
- *Link* local address for use in a isolated subnet

Packet Format

The IPv6 base header is always 40 bytes long. The packet format is:



Version—specifies the IP version, i.e., 6.

TrafficClass—defines the priority of the packet with respect to traffic congestion. It is either congestion-controlled or non-congestion controlled

FlowLabel—is designed to provide special handling for a particular flow of data. The router handles flow with the help of a flow table.

PayloadLen—gives the length of the packet, excluding the IPv6 header

NextHeader—If options are required, then it is specified in one or more special headers following the IP header, its value is contained in NextHeader field. Otherwise, it identifies the higher-level protocol (TCP/UDP).

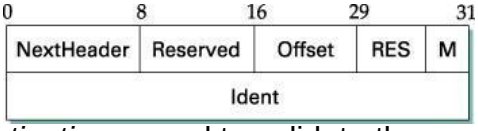
HopLimit—This field serves the same purpose as TTL field in IPv4.

SourceAddress and DestinationAddress—contains 16-byte address of the source and destination host respectively.

Extension Header

To provide greater functionality to IP datagram, the base header can be followed by up to six extension headers. They are:

1. *Hop-by-Hop*—used when the source needs to pass information to all routers visited by the datagram.
2. *Source Routing*—accounts for both strict and loose source routing.
3. *Fragmentation*—In IPv6, only the original source can fragment. A source must use a path MTU discovery technique to find the smallest MTU on the path.



4. *Authentication*—used to validate the sender and ensures the integrity of data
5. *Encrypted Security Payload*—provides confidentiality and guards against eavesdropping.
6. *Destination*—used when source needs to pass information to destination only. Intermediate routers cannot access this information.

Auto configuration

IPv6 provides a new form of *autoconfiguration* called *stateless* auto-configuration, which allows a host to be connected without the help of a DHCP server.

List the advantages of IPv6.

Large address space An IPv6 address is 128 bits long. Compared with the 32-bit address of IPv4, this is a huge (2^{96}) increase in the address space.

Better header format IPv6 uses a new header format in which options are separated from the base header and inserted, when needed.

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

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

Support for resource allocation In IPv6, flow label has been added to enable the source to request special handling of the packet such as real-time audio and video.

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

State the drawbacks of IPv4?

Despite all short-term solutions, such as subnetting, classless addressing, and NAT, address depletion is still a long-term problem in the Internet.

The Internet must accommodate real-time audio and video transmission that requires minimum delay strategies and reservation of resources, which are not provided in IPv4.

The Internet must provide encryption and authentication of data for some applications. No encryption or authentication is provided by IPv4.

How NAT helps to solve address space depletion?

The idea behind Network Address Translation (NAT) is that all hosts that uses Internet do not need to have globally unique addresses.

NAT enables a organization to have a large set of addresses internally and one address or a small set of addresses externally.

Three sets of addresses are reserved for internal use (10.0.0.0 – 10.255.255.255, 172.16.0.0 – 172.31.255.255 and 192.168.0.0 – 192.168.255.255).

The organization must have only one single connection to the Internet through a router that runs the NAT software.

Briefly explain IGMP?

Internet Group Message Protocol (IGMP) is a protocol that manages group membership.

Provides the multicast routers information about the membership status of hosts (routers) connected to the network.

Enables a multicast router to create and update list of loyal members related to each router interface.

The operations are:

- Joining a group
- Leaving a group
- Monitoring membership

Explain Multicast routing protocols in detail.

A host places a multicast address in the destination address field to send packets to a set of hosts belonging to a group.

Internet multicast can be implemented on physical networks that support broadcasting by extending forwarding functions. The extended ones are:

- Link-State Multicast
- Distance-Vector Multicast
- Protocol Independent Multicast (PIM)

Link-State Multicast

Multicasting is added to the existing link-state routing.

- Each router knows entire topology by way of update messages.
- Dijkstra's algorithm is used to compute shortest path spanning tree to reach all destinations.

Each router determines which groups have members on which LAN by monitoring the periodical announcements.

- If a host does not advertise periodically, then it has left the group.

Equipped with group and membership knowledge, each router computes shortest path multicast tree from any source to any group using Dijkstra's algorithm.

Link-state routing is expensive as each router must store a multicast tree from every source to every group.

Distance-Vector Multicast

Multicasting is added to existing distance-vector routing in two stages.

- Each router maintains a table of (Destination, Cost, NextHop) for all destination through exchange of distance vectors.
- Reverse Path Broadcast mechanism that floods packets to other networks
- Reverse Path Multicasting that prunes end networks that do not have hosts belonging to a multicast group.

Reverse-Path Broadcasting

A router when it receives a multicast packet from source *S* to a Destination from NextHop, then it forwards the packet on all out-going links.

The drawbacks are:

- It floods a network, even if it has no members for that group
- *Duplicate flooding*, i.e., packets are forwarded over the LAN by each router connected to that LAN.

Duplicate flooding is avoided by

- Designating a router on the shortest path as *parent* router.
- Only parent router is allowed to forward multicast packets from source *S* to that LAN.

Reverse-Path Multicasting

Multicasting is achieved by pruning networks that do not have members for a group *G*.

Pruning is achieved by identifying a *leaf* network, which has only one router (parent). The leaf network is *monitored* to determine if it has any members for group *G*.

The router then decides whether or not to forward packets addressed to G over that LAN. The information "*no members of G here*" is propagated up the shortest path tree.

Thus routers can come to know for which groups it should forward multicast packets. Including all this information in a routing update is expensive.

Protocol Independent Multicast (PIM)

The above two multicast routing did not scale well.

PIM divides multicast routing into *sparse* and *dense* mode.

In PIM sparse mode (PIM-SM), routers leave and join multicast group using PIM Join and Prune messages.

PIM designates a *rendezvous point* (RP) for each group in a domain to receive PIM messages.

All routers in the domain know the IP address of *RP* for each group.

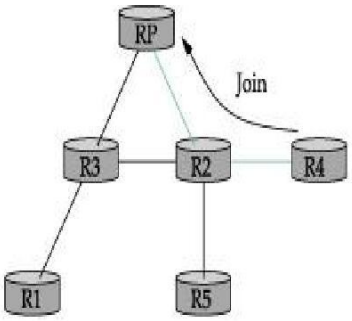
A multicast forwarding tree is built as a result of routers sending Join messages to the RP. The tree may be either *shared* by multiple senders or *source-specific* to a sender.

Shared Tree

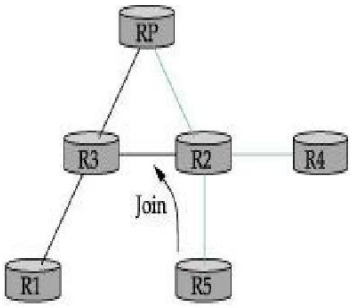
When a router sends Join message for group *G* to *RP*, it goes through a sequence of routers.

Each router along the path creates an entry (*, *G*) in its forwarding table for the shared tree before forwarding the Join message.

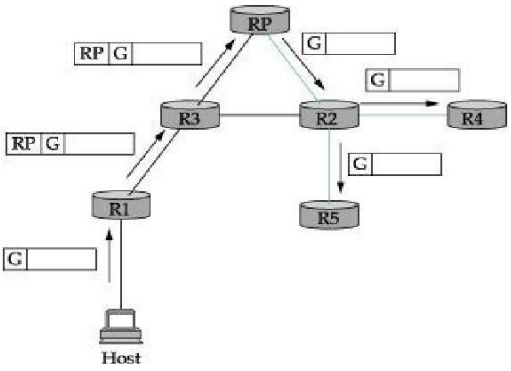
Eventually, the message arrives at RP. Thus a shared tree with RP as *root* is formed.



Join from R4



Join from R5



Multicast message from Host

The above figure shows router *R4* sending Join message for group *G* to *RP*.

It goes through *R2*. *R2* makes an entry (*, *G*) in its table and forwards the message to *RP*. Later when *R5* sends Join message for group *G*, it shares the tree. Therefore *R2* does not forwards the Join message.

When a host attached to router *R1*, sends a message to group *G*, which is received by *R1*. *R1* does not know about group *G*, therefore it encapsulates the multicast packet with unicast address and is tunneled along the way to *RP*.

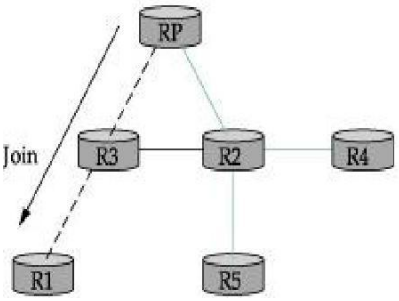
RP decapsulates the packet and sends the multicast packet to *R2*, which forwards it to routers *R4* and *R5* that have members for group *G*.

Source-specific tree.

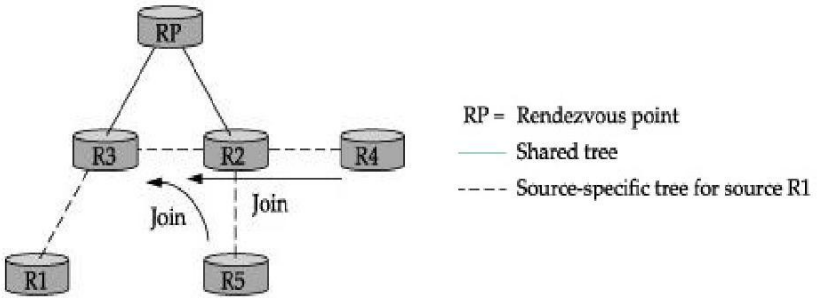
RP has the option of forcing about group *G*, onto other routers by sending a source-specific Join message to sending host, so that tunneling can be avoided.

The intermediary routers create an entry (*S*, *G*) for source-specific tree.

If more packets are sent from source *S* to group *G*, then other routers switch to source-specific tree with source host as root.



Source-specific Join from RP



Routers switch to Source tree

Analysis

PIM is protocol independent because, tree formation is based on path that Join messages follows based on *shortest* path.

Shared trees are more *scalable* than source-specific trees.

Source-specific trees enable *efficient* routing than shared trees.

Briefly explain the mechanisms to avoid and control congestion in the network

Congestion control refers to techniques that can either prevent congestion before it happens or remove congestion after it has happened thereby keeping the load below capacity.

Retransmission Policy

Retransmission increases congestion in the network. But, a good retransmission policy can prevent congestion.

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 Go-Back-N for congestion control, since it tries to send specific packets that have been lost or corrupted.

Acknowledgment Policy

Sending fewer ACK means imposing less load on the network.

A receiver may send an acknowledgment only if it has a packet to be sent. A receiver may decide to acknowledge only N packets at a time.

Discarding Policy

A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrity of the transmission.

For example, in audio transmission, if the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented or alleviated.

Admission Policy

An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual-circuit networks.

Switches in a flow first check the resource requirements of a flow before admitting it to the network.

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

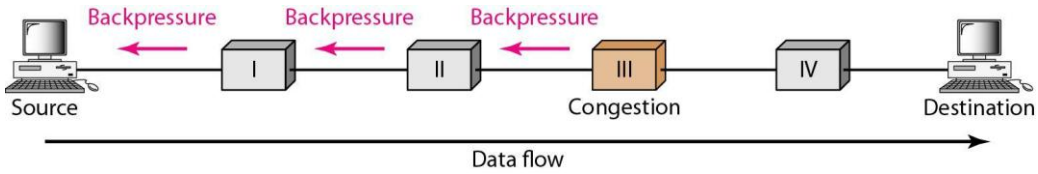
Backpressure

In backpressure mechanism, a congested node stops receiving data from the immediate upstream node or nodes.

This may cause the upstream node to become congested, and it in turn rejects data from upstream node, and so on.

Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow to the source.

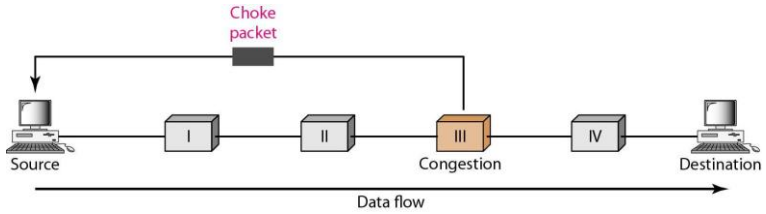
This technique is used in virtual circuit networks.



Choke Packet

A choke packet is a packet sent by a node to the source to inform it of congestion.

In choke packet method, warning is from the router which has encountered congestion, to the source station directly.



Implicit Signaling

In implicit signaling, the source guesses that there is congestion somewhere in the network from other symptoms.

For example, when a source sends several packets and there is no acknowledgment for a while, it assumes that network is congested. Therefore, the source slows down.

Explicit Signaling

The node that experiences congestion can explicitly send a signal to the source or destination by setting a bit that can be set in a packet

This bit can warn the source that there is congestion and that it needs to slow down to avoid discarding of packets.

The receiver use policies such as slowing down acknowledgments to alleviate congestion.

Illustrate and explain UDP and its packet format.

User Datagram Protocol (UDP) is a connectionless, unreliable transport protocol.

It does not add anything to the services of IP except process-to-process communication.

UDP is a simple multiplexer/demultiplexer that allow multiple processes on each host to share the network.

UDP does not implement flow control or reliable/ordered delivery.

UDP ensures delivering of message to the intended recipient by the use of checksum. If a process wants to send a small message and does not require reliability, UDP is used.

Port Number

Each process is assigned a unique 16-bit port number on that host. Processes are identified by (host, port) pair.

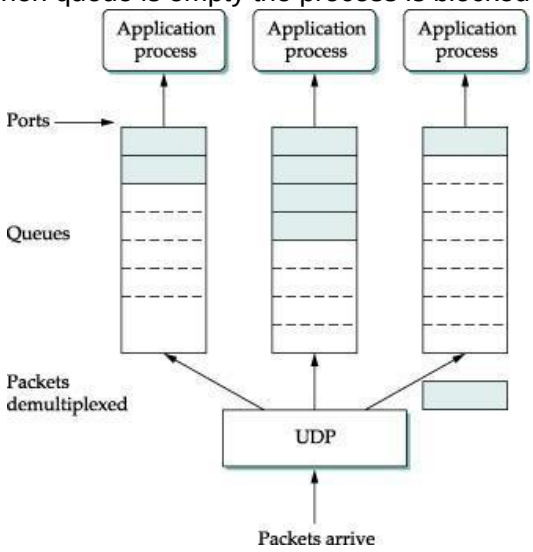
Processes can be classified as either as *client* / *server*.

- Client process usually initiates exchange of information with the server
- Server process is identified by a well-known port number (0 – 1023).
- Client process is assigned an ephemeral port number (49152 – 65,535) by operating system.
- Some well known UDP ports are:

Port	Protocol
7	Echo
13	Daytime
53	DNS
111	RPC
161	SNMP

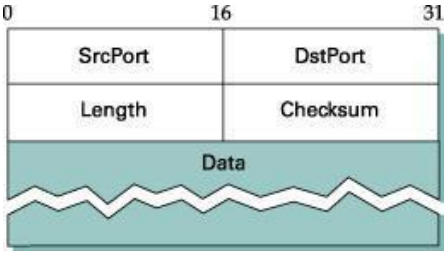
Ports are usually implemented as a message queue.

- When a message arrives, UDP appends the message to the end of the queue.
- When queue is full, the message is discarded.
- When a message is read, it is removed from the queue.
- When queue is empty the process is blocked



UDP Header

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



SrcPort and DstPort—Contains port number for both the sender (*source*) and receiver (*destination*) of the message.

Length—This 16-bit field defines total length of the user datagram, header plus data. The total length is less than 65,535 bytes as it is encapsulated in an IP datagram.
UDP length = IP length - IP header's length

Checksum—It is computed over pseudo header, UDP header and message content to ensure that message is correctly delivered to the exact recipient.

- The *pseudo header* consists of three fields from the IP header (protocol number (17), source and destination IP address), plus the UDP length field.

Applications

UDP is used for management processes such as SNMP.

UDP is used for some route updating protocols such as RIP. UDP is a suitable transport protocol for multicasting.

UDP is suitable for a process with internal flow and error control mechanisms such as Trivial File Transfer Protocol (TFTP).

Bring out the classification of port numbers.

Well-known ports range from 0 to 1023 are assigned and controlled by IANA.

Registered ports range from 1024 to 49,151 are not assigned or controlled by IANA. They can only be registered with IANA to prevent duplication.

Ephemeral (dynamic) ports range from 49,152 to 65,535 is neither controlled nor registered. It is usually assigned to a client process by the operating system.

Distinguish between network and transport layer

Network layer	*Transport layer*
The network layer is responsible for *host*-to-*host* delivery	The transport layer is responsible for *process*-to-*process* delivery of a packet
Host address is required for delivery	Host address and port number is required for delivery
Error detection is not offered	Error detection is done using checksum
Flow control is not done	Flow control is not done
Multicasting capability is not inbuilt	Multicasting is embedded into UDP

With a neat architecture, explain TCP in detail.

Transmission Control Protocol (TCP) offers a reliable, connection-oriented, byte-stream service

TCP guarantees the reliable, in-order delivery of a stream of bytes It is a full-duplex protocol

TCP supports demultiplexing mechanism for process-to-process communication.

TCP has built-in congestion-control mechanism, i.e., sender is prevented from overloading the network.

Process-to-Process Communication

Like UDP, TCP provides process-to-process communication. A TCP connection is identified a 4-tuple (SrcPort, SrcIPAddr, DstPort, DstIPAddr).

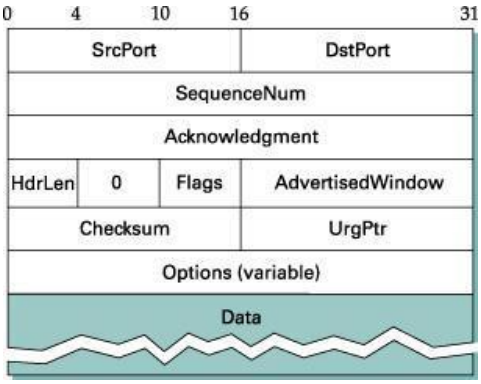
Some well-known port numbers used by TCP are

Port	Protocol
23	TELNET
25	SMTP
80	HTTP

Segment Format

TCP is a byte-oriented protocol, i.e. the sender writes bytes into a TCP connection and the receiver reads bytes out of the TCP connection.

TCP groups a number of bytes together into a packet called *segment* and adds a header onto each segment. Segment is encapsulated in a IP datagram and transmitted.



SrcPort and DstPort fields identify the source and destination ports.

SequenceNum field contains sequence number, i.e. first byte of data in that segment.

Acknowledgment defines byte number of the segment, the receiver expects next.

HdrLen field specifies the number of 4-byte words in the TCP header. Flags field contains six control bits or flags. They are set to indicate:

- *URG*—indicates that the segment contains urgent data.
- *ACK*—the value of acknowledgment field is valid.
- *PSH*—indicates sender has invoked the push operation.
- *RESET*—signifies that receiver wants to abort the connection.
- *SYN*—synchronize sequence numbers during connection establishment.
- *FIN*—terminates the connection

AdvertisedWindow field defines the receiver window and acts as flow control.

Checksum field is computed over the TCP header, the TCP data, and pseudoheader.

UrgPtr field indicates where the non-urgent data contained in the segment begins.

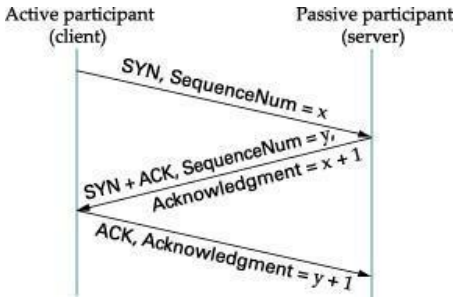
Optional information (max. 40 bytes) can be contained in the header.

Connection Establishment

The connection establishment in TCP is called *three-way handshaking*

1. The client (active participant) sends a segment to the server (passive participant) stating the initial sequence number it is to use (Flags = SYN, SequenceNum = *x*)
2. The server responds with a single segment that both acknowledges the client's sequence number (Flags = ACK, Ack = *x* + 1) and states its own beginning sequence number (Flags = SYN, SequenceNum = *y*).

3. Finally, the client responds with a segment that acknowledges the server's sequence number (Flags = ACK, Ack = *y* + 1).



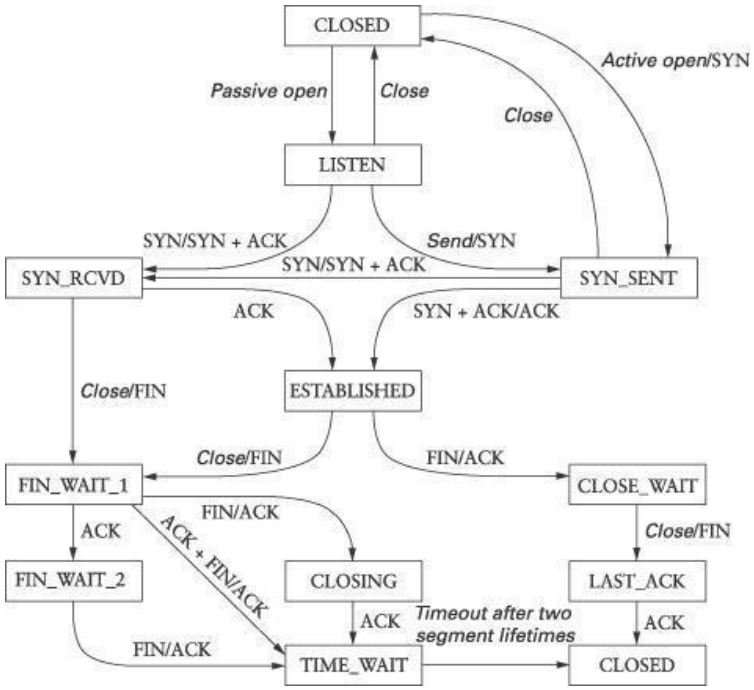
State Transition Diagram

The states involved in opening and closing a connection is shown above and below ESTABLISHED state respectively.

The operation of sliding window (i.e., retransmission) is not shown. The two events that trigger a state transition is:

- a segment arrives from its peer.
- the local application process invokes an operation on TCP

TCP's state transition diagram defines the semantics of both its *peer-to-peer* interface and its *service* interface.



Opening

1. The server first invokes a *passive* open on TCP, which causes TCP to move to LISTEN state
2. Later, the client does an *active* open, which causes its end of the connection to send a SYN segment to the server and to move to the SYN_SENT state.
3. When the SYN segment arrives at the server, it moves to SYN_RCVD state and responds with a SYN + ACK segment.
4. The arrival of this segment causes the client to move to the ESTABLISHED state and to send an ACK back to the server.

5. When this ACK arrives, the server finally moves to the ESTABLISHED state.
    a. Even if the client's ACK gets lost, sever will move to ESTABLISHED state when the first data segment from client arrives.

Closing

In TCP, the application process on both sides of the connection can independently close its half of the connection or simultaneously.

Three combinations of transitions from ESTABLISHED to CLOSED state are possible.

Connection Termination

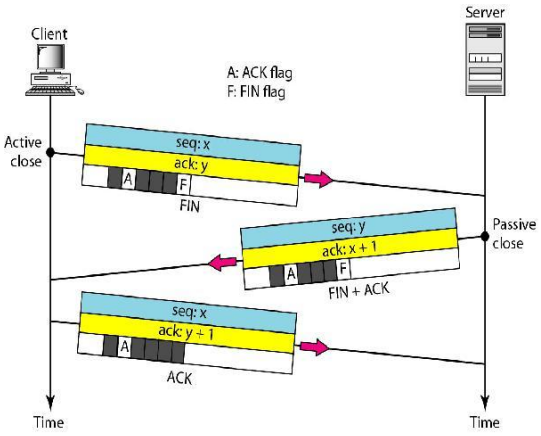
Three-way Handshaking

1. The client TCP after receiving a Close command from the client process sends a FIN segment. FIN segment can include the last chunk of data.
2. The server TCP responds with FIN + ACK segment to inform its closing.
3. The client TCP finally sends an ACK segment.

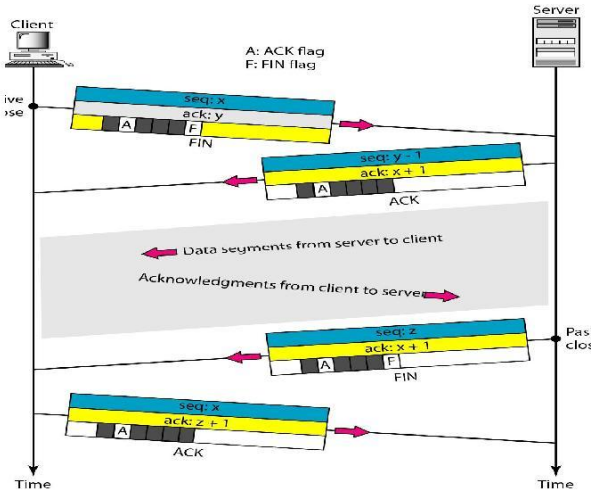
Four-way Half-Close

In TCP, one end can stop sending data while still receiving data, known as *half-close*. For instance, submit its data to the server initially for processing and close its connection. At a later time, the client receives the processed data from the server.

1. The client TCP half-closes the connection by sending a FIN segment.
2. The server TCP accepts the half-close by sending the ACK segment. The data transfer from the client to the server stops.
3. The server can send data to the client and acknowledgement can come from the client.
4. When the server has sent all the processed data, it sends a FIN segment to the client.
5. The FIN segment is acknowledged by the client.



Three-way close



Four-way close

What is urgent data in TCP?

At times an application may need to send urgent data, i.e., sending process wants a piece of data to be read out of order by the receiving process. For example, to abort a process by issuing Ctrl+C keystroke.

The sending TCP inserts the urgent data at beginning of the segment and sets URG flag.

The urgent pointer field in the header defines start of normal data.

When the receiving TCP receives a segment with the URG bit set, it delivers urgent data out of order to the receiving application.

What is push operation in TCP?

The receiving TCP buffers the data and delivers them to the application program when it is convenient for the receiving process.

In case of interactive applications, delayed delivery of data is not acceptable. The application program at the sending site can request a Push operation.

This instructs the sending TCP not to wait for the window to be filled. It must create a segment and send it immediately.

The sending TCP also sets the push bit (PSH) to let the receiving TCP know that the segment includes data that must be delivered to the receiving application program as soon as possible and not to wait for more data to come.

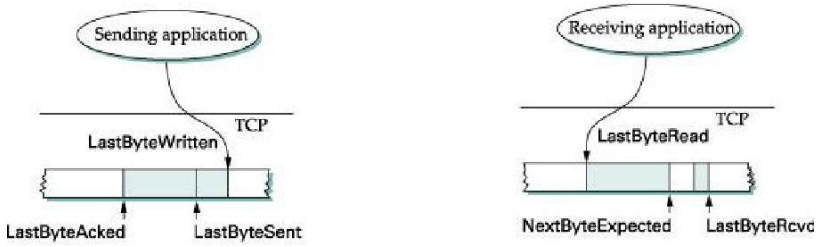
Explain TCPs adaptive control and its uses.

TCP uses a variant of sliding window known as adaptive flow control that:
- guarantees the reliable delivery of data in ordered manner
- enforces flow control at the sender

The receiver advertises a window size to the sender using AdvertisedWindow field in the TCP header

The sender cannot have unacknowledged data greater than value of AdvertisedWindow

Reliable and Ordered Delivery



Sender *Receiver*

TCP on the sending side maintains a *send* buffer that is divided into 3 segments namely acknowledged data, unacknowledged data and data to be transmitted

Similarly TCP on the receiving side maintains a *receive* buffer to hold data even if it arrives of order.

The send buffer maintains three variables namely LastByteAcked, LastByteSent, and LastByteWritten as shown above. The relation between them is obvious

LastByteAcked LastByteSent and LastByteSent LastByteWritten

The bytes to the left of LastByteAcked are not kept as it had been acknowledged.

The receive buffer maintains three variables namely LastByteRead, NextByteExpected, and LastByteRcvd. The relation between them is

LastByteRead < NextByteExpected and NextByteExpected LastByteRcvd + 1

If data are received in order, NextByteExpected is the next byte after LastByteRcvd Bytes to the left of LastByteRead is not buffered as it has been read by the application

Flow Control

The capacity of *send* and *receiver* buffer is MaxSendBuffer and MaxRcvBuffer respectively.

The sending TCP prevents overflowing of its buffer by maintaining

LastByteWritten LastByteAcked MaxSendBuffer

The receiving TCP avoids overflowing its receive buffer by maintaining

LastByteRcvd LastByteRead MaxRcvBuffer

The receiver throttles the sender by advertising a window that is no larger than the amount of *free* space that it can buffer as

AdvertisedWindow = MaxRcvBuffer ((NextByteExpected 1) LastByteRead)

When data arrives, the receiver acknowledges it as long as preceding bytes have arrived.

- LastByteRcvd moves to its right (incremented), and the advertised window shrinks

The advertised window expands when the data is read by the application

- It data is read as fast as it arrives then AdvertisedWindow = MaxRcvBuffer
- If it is read slow, it eventually leads to a AdvertisedWindow of size 0.

The sending TCP adheres to the advertised window by computing *effective* window, that limits how much data it should send as

EffectiveWindow = AdvertisedWindow (LastByteSent LastByteAcked)

When a acknowledgement arrives for *x* bytes, LastByteAcked is incremented by *x* and the buffer space is freed accordingly

Fast Sender vs. Slow Receiver

A slow receiver prevents being swamped with data from a fast receiver by using AdvertisedWindow field

Initially the fast sender transmits at a higher rate.

The receiver's buffer gets filled up. Hence, AdvertisedWindow shrinks, eventually to 0.

When the receiver advertises window of size 0, sender cannot transmit any further data. Therefore, the TCP at the sender blocks the sending process.

When the receiving process reads some data, those bytes are acknowledged. Thus the AdvertisedWindow expands.

The LastByteAcked is incremented and buffer space is freed to that extent,

The sending process becomes unblocked and is allowed to fill up the free space.

Checking AdvertisedWindow status

TCP always sends a segment in response that contains the latest values for the Acknowledge and AdvertisedWindow fields, even if these values have not changed.

Thus the sender can come to know the status of AdvertisedWindow even after the receiver advertises a window of size 0.

AdvertisedWindow

The TCP's AdvertisedWindow field is 16 bits long, half the size of SequenceNum The length of 16-bits ensures that it does not wrap around

The length of AdvertisedWindow is designed such that it allows the sender to keep the pipe full.

The 16-bit length also accounts for product of delay × bandwidth.

What is adaptive retransmission? Explain the algorithms used?

TCP guarantees reliability through retransmission.

- Retransmission due to timeout before ACK.
- Timeout is a function of RTT.
- RTT is highly variable between any two hosts on the internet. ○

Appropriate timeout is chosen using adaptive retransmission.

Original Algorithm

TCP estimates SampleRTT by computing the duration between sending of a packet and arrival of its ACK.

TCP then computes EstimatedRTT as a weighted average between the previous and current estimate as

EstimatedRTT = × EstimatedRTT + (1) × SampleRTT

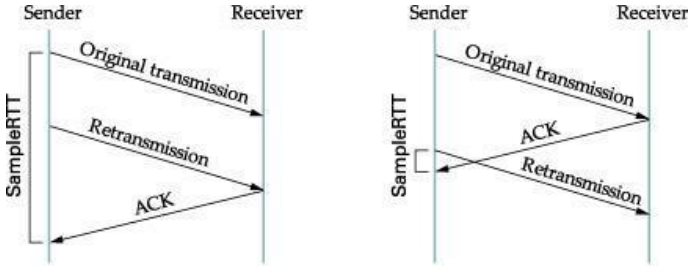
where is the smoothening factor and its value is in the range 0.8–0.9 Timeout is twice the EstimatedRTT

TimeOut = 2 × EstimatedRTT

Karn/Partridge Algorithm

The flaw discovered in original algorithm after years of use is

- whether ACK should be associated with the original or retransmission segment
- If ACK is associated with original one, then SampleRTT becomes too large
- If ACK is associated with retransmission, then SampleRTT becomes too small



Karn/Partridge proposed a solution to the above by making changes to the timeout mechanism.

Each time TCP retransmits, it sets the next timeout to be twice the last timeout.

- Loss of segments is mostly due to congestion and hence TCP source does not react aggressively to a timeout.

Jacobson/Karels Algorithm

The main problem with original algorithm is that variance of the sample RTTs is not taken into account.

- if variation among samples is small, then EstimatedRTT can be trusted
- otherwise timeout should not be tightly coupled with the EstimatedRTT

In this new approach, the sender measures a new SampleRTT as before. The Deviation amongst RTTs is computed as follows:

Difference = SampleRTT EstimatedRTT

EstimatedRTT = EstimatedRTT + (× Difference)

Deviation = Deviation + (|Difference| Deviation)

where is a fraction between 0 and 1

TCP now computes TimeOut as a function of both EstimatedRTT and Deviation as listed:

TimeOut = × EstimatedRTT + × Deviation

where = 1 and = 4 usually

When variance is small, difference between TimeOut and EstimatedRTT is negligible.

When variance is larger, Deviation plays a greater role in deciding TimeOut.

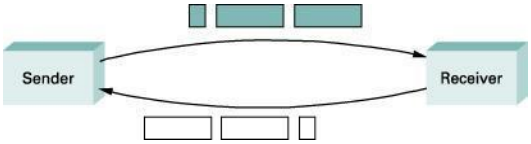
What is silly window syndrome? When should TCP transmit a segment?

When an ACK arrives, the window enlarges for transmission.

Even if window size is less than one MSS, TCP decides to go ahead and transmit a half-full segment.

The strategy of aggressively taking advantage of any available window leads to a situation now known as the *silly window syndrome*.

If the sender aggressively fills, then any small segments introduced into the system remains in the system indefinitely as it does not combine with adjacent segments to create larger ones as shown.



Nagle's Algorithm

Nagle's suggests a solution as to what the sending TCP should do when there is data to send and window size is less than one MSS. The algorithm is listed below:

When the application produces data to send

```
if both the available data and the window MSS
        send a full segment
else
if there is unACKed data in flight
        buffer the new data until an ACK arrives
else
        send all the new data now
```

It's always OK to send a full segment if the window allows.

It's also OK to immediately send a small amount of data if there are currently no segments in transit, but if there is anything in flight, the sender must wait for an ACK before transmitting the next segment.

Explain TCP congestion control techniques in detail.

In TCP congestion control, each source has to determine the available capacity in the network, so that it can send packets without loss.

By using ACKs to pace transmission of packets, TCP is said to be *self-clocking*. TCP maintains a state variable CongestionWindow for each connection. Therefore

```
MaxWindow = MIN(CongestionWindow, AdvertisedWindow)
EffectiveWindow = MaxWindow  (LastByteSent  LastByteAcked)
```

Thus, a TCP source is allowed to send no faster than *network* or *destination* host

The problem is that available bandwidth changes over time. The three congestion control mechanism are:

- Additive Increase/Multiplicative Decrease
- Slow Start
- Fast Retransmit and Fast Recovery

Additive Increase/Multiplicative Decrease (AIMD)

TCP source sets the CongestionWindow based on the level of congestion it perceives to exist in the network.

The additive increase/multiplicative decrease (AIMD) mechanism works as follows: ○ The source increases CongestionWindow when level of congestion goes down

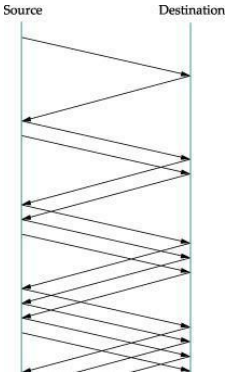
and decreases CongestionWindow when level of congestion goes up.

TCP interprets timeouts as a sign of congestion and reduces the rate at which it is transmitting.

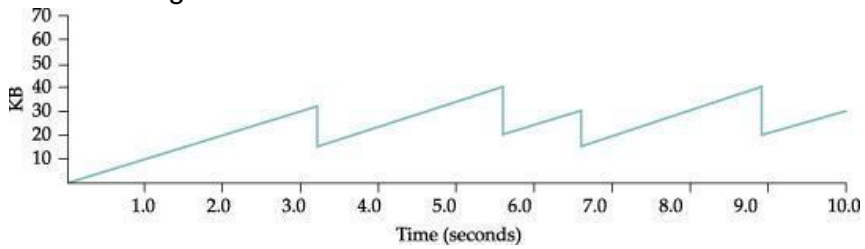
- Each time a timeout occurs, the source sets CongestionWindow to half of its previous value. This is known as *multiplicative decrease*.
- For example, if CongestionWindow is set to 16 packets, after a packet loss, it is set to 8.
- The CongestionWindow is not allowed to fall below one packet size or MSS, irrespective of the level of congestion.

Every time, the source successfully sends one packet, CongestionWindow is increased by a fraction (*additive increase*).

- An ACK acknowledges receipt of MSS bytes, the increment is computed as
  Increment = MSS × (MSS/CongestionWindow)
  CongestionWindow += Increment



This pattern of continually increasing and decreasing the congestion window continues throughout the lifetime of the connection



When the current value of CongestionWindow as a function of time, it results as a saw-tooth pattern.

AIMD decreases its CongestionWindow aggressively but increases conservatively.

- Having small CongestionWindow only results in less probability of packets being dropped.
- Thus congestion control mechanism becomes stable.

Since timeout is an indication of congestion that triggers multiplicative decrease, TCP needs the most accurate timeout mechanism.

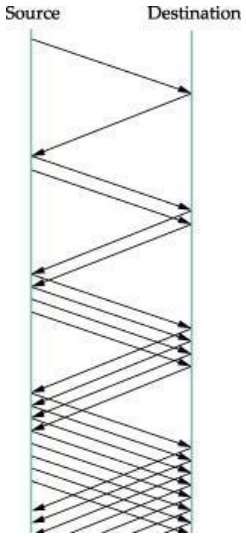
AIMD is appropriate only when source is operating close to network capacity.

Slow Start

Slow start increases the congestion window exponentially, rather than linearly. It is usually used from cold start.

The source starts by setting CongestionWindow to one packet.

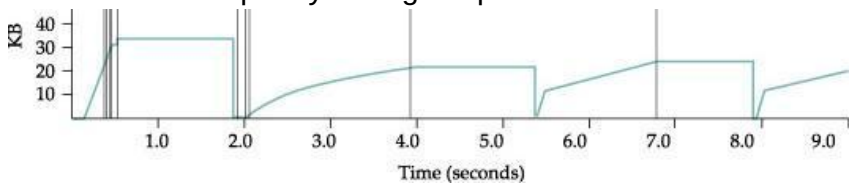
- When ACK arrives, TCP adds 1 to CongestionWindow and sends two packets.
- Upon receiving two ACKs, TCP increments CongestionWindow by 2 and sends four packets.
- Thus TCP doubles the number of packets every RTT as shown.



Slow start provides exponential growth and is designed to avoid bursty nature of TCP. Initially TCP has no idea about congestion, henceforth it increases CongestionWindow rapidly until there is a packet loss. When a packet is lost:

- TCP immediately decreases CongestionWindow by half (*multiplicative decrease*).
- It stores the current value of CongestionWindow as CongestionThreshold and resets to CongestionWindow one packet
- The CongestionWindow is incremented one packet for each ACK arrived until it reaches CongestionThreshold and thereafter one packet per RTT.

In initial stages, TCP loses more packets because it attempts to learn the available bandwidth quickly through exponential increase



In example, initial slow start causes increase in CongestionWindow up to 34KB. Congestion occurs at 2secs and loss of packets results.

- CongestionThreshold is set to 17KB and CongestionWindow to 1 packet.
- Thereafter additive increase is followed

Fast Retransmit and Fast Recovery

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

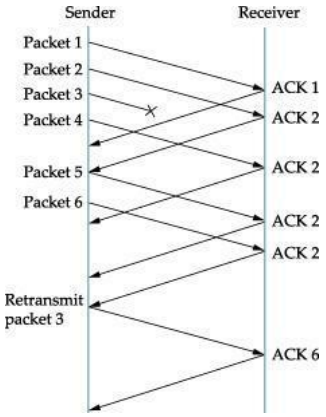
When a packet arrives out of order, the receiving TCP resends the same acknowledgment (*duplicate ACK*) it sent the last time.

The sending TCP waits for three duplicate ACK, to confirm that the packet is lost before retransmitting the lost packet.

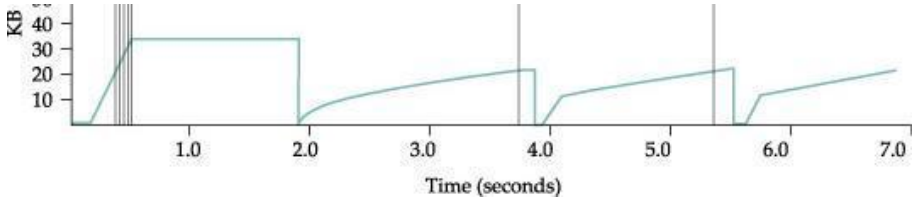
This is known as *fast retransmit* and it signals congestion.

Instead of setting CongestionWindow to one packet, this method uses the ACKs that are still in pipe to clock the sending of packets. This is called *fast recovery*.

The fast recovery mechanism removes slow start phase and follows additive increase. The fast retransmit/recovery results increase in throughput by 20%.



In example the third packet gets lost. The sender on receiving three duplicate ACKs (ACK 2) retransmits the third packet.



In graph shown, fast recovery avoids slow start from 3.8 to 4 sec. Therefore congestion window is reduced by half from 22 KB to 11 KB.

Slow start is only used at the beginning of a connection and after regular timeout.

At other times, the congestion window follows a pure additive increase/multiplicative decrease pattern

TCP's fast retransmit can detect up to three dropped packets per window.

Explain in detail about TCP congestion avoidance algorithms.

Congestion avoidance mechanisms prevent congestion before it actually occurs. When congestion is likely to occur, TCP decreases load on the network.

TCP creates loss of packets in order to determine bandwidth of the connection The three congestion-avoidance mechanisms are:

1. DECbit
2. Random Early Detection (RED)
3. Source-based congestion avoidance

DECbit

Was developed for use on Digital Network Architecture

In DEC bit, each router monitors the load it is experiencing and explicitly notifies the end node when congestion is about to occur by setting a binary congestion bit called **DECbit** in packets that flow through it.

The destination host copies the DECbit onto the ACK and sends back to the source.

Eventually the source reduces its transmission rate and congestion is avoided.

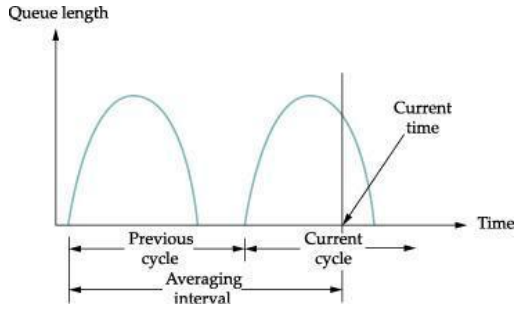
Algorithm

A single congestion bit is added to the packet header.

A router sets this bit in a packet if its average queue length is 1.

The average queue length is measured over a time interval that spans the last busy + last idle cycle + current busy cycle.

Router calculates average queue length by dividing the curve area by time interval



The source computes how many ACK has DEC bit set for the previous window packets it has sent.

1. If it is less than 50% then source increases its congestion window by 1 packet.
2. Otherwise, source decrease the congestion window by 87.5%.

Random Early Detection (RED)

Proposed by Floyd and Jackson

In RED, router implicitly notifies the source that congestion is likely to occur by dropping one of its packets.

The source is notified by timeout or duplicate ACK.

The router drops a few packets earlier before it runs out of space, so that it need not drop more packets later.

Each incoming packet is dropped with a probability known as *drop probability* when the queue length exceeds *drop level*.

Algorithm

RED computes average queue length using a weighted running average as follows:

AvgLen = (1 Weight) × AvgLen + Weight × SampleLen

- where 0 < Weight < 1 and SampleLen is length of the queue when a sample measurement is made.
- The weighted running average detects long-lived congestion.

RED has two queue length thresholds MinThreshold and MaxThreshold. When a packet arrives at the gateway, RED compares the current AvgLen with these thresholds and decides whether to queue or drop the packet as follows:

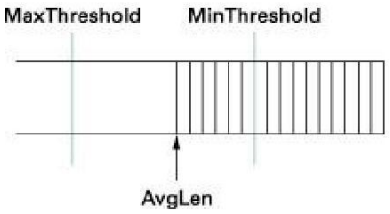
```
if AvgLen MinThreshold
        queue the packet
if MinThreshold < AvgLen < MaxThreshold
        calculate probability P
        drop the arriving packet with probability
P if MaxThreshold AvgLen
        drop the arriving packet
```

P is a function of both AvgLen and how long it has been since the last packet was dropped. It is computed as

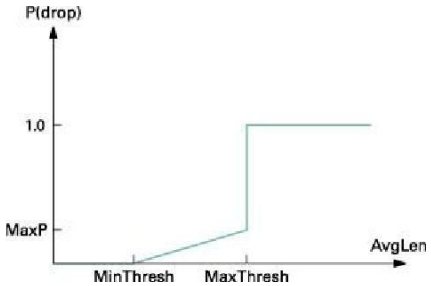
TempP = MaxP × (AvgLen MinThreshold)/(MaxThreshold MinThreshold) P
= TempP/(1 count × TempP)

The probability of drop increases slowly when AvgLen is between the two thresholds, reaching MaxP at the upper threshold, at which point it jumps to unity.

MaxThreshold is set to twice of MinThreshold as it works well for the Internet traffic. Because RED drops packets randomly, the probability that RED decides to drop a flow's packet(s) is roughly proportional to share of the bandwidth for that flow.



RED thresholds



Drop probability function

Source-Based Congestion Avoidance

The source looks for signs of congestion on the network, for example, a considerable increase in the RTT, indicate queuing at a router.

Some mechanisms

1. Every two round-trip delays, it checks to see if the current RTT is greater than the average of the minimum and maximum RTTs.
   a. If it is, then the algorithm decreases the congestion window by one-eighth.
   b. Otherwise the normal increase as in TCP.
2. Every RTT, it increases the window size by one packet and compares the throughput achieved to the throughput when the window was one packet smaller.
   a. If the difference is less than one-half the throughput achieved when only one packet was in transit, it decreases the window by one packet.

TCP Vegas

In standard TCP, it was observed that throughput increases as congestion window increases, but not beyond the available bandwidth.

Any further increase in the window size only results in packets taking up buffer space at the bottleneck router

TCP Vegas uses this idea to measure and control the right amount of extra data in transit.

If a source is sending too much extra data, it will cause long delays and possibly lead to congestion.

TCP Vegas's congestion-avoidance actions are based on changes in the estimated amount of extra data in the network.

A flow's BaseRTT is set to the minimum of all RTTs and is mostly the first packet sent.

The expected throughput is given by ExpectedRate = CongestionWindow/BaseRTT

The sending rate, ActualRate is computed by dividing number of bytes transmitted during a RTT by that RTT.

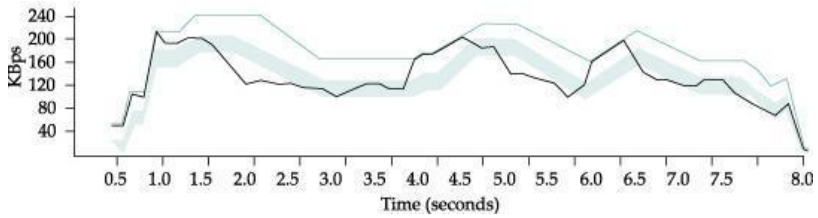
The difference between two rates is computed, say Diff = ExpectedRate – ActualRate Two thresholds and are defined such that <

- When Diff < , congestion window is linearly increased during the next RTT
- When Diff > , congestion window is linearly decreased during the next RTT
- When < Diff < , the congestion window is unchanged

When actual and expected output varies significantly, the congestion window is reduced as it indicates congestion in the network.

When actual and expected output is almost the same, the congestion window is increased to utilize the available bandwidth.

The overall goal is to keep between and extra bytes in the network.



The expected & actual throughput with thresholds and (shaded region) is shown.

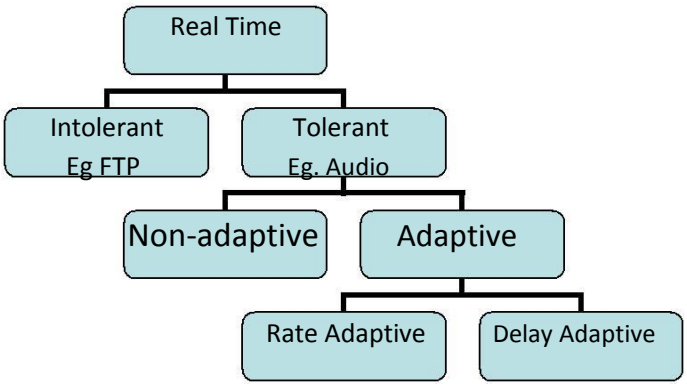
What is the need for QoS?

Certain applications are not satisfied with best-effort service offered by the network. ○ Multimedia applications require minimum bandwidth.

- Real-time applications require timeliness rather than ensuring correctness of the message.

Therefore the network needs to support different level of service for different set of application.

Classify the real-time applications based on QoS.



Tolerance is based on response to loss of packets

Adaptability is based on response to delay experienced.

Bit rate vs quality

Define QoS.

QoS is defined as a set of attributes pertaining to the performance of a connection. The attributes may be either user or network oriented.

QoS on the Internet can be broadly classified into

- *Integrated Services* (IntSrv)
- *Differentiated Services*

Explain how QoS is provided through integrated services.

Integrated Services (*IntSrv*) is a flow-based QoS model, i.e., user creates flow from source to destination and informs all routers of the resource requirement.

Service Classes

The two classes of service defined are *Guaranteed* and *Controlled load* service.

Guaranteed service in which the network assures that delay will not be beyond some maximum if flow stays within *TSpec*. It is designed for *intolerant* applications.

Controlled load service meets the need of tolerant, adaptive applications which requests low-loss or no-loss such as file transfer, e-mail, etc.

Flowspec

The set of information given to the network for a given flow is called *flowspec*. It has two parts namely

- *Tspec* defines the traffic characterization of the flow
- *Rspec* defines resources that the flow needs to reserve (buffer, bandwidth, etc.)

TSpec

The bandwidth of real-time application varies constantly for most application.

The average rate of flows cannot be taken into account as variable bit rate applications exceed the average rate. This leads to queuing and subsequent delay/loss of packets.

Token Bucket

The solution to manage varying bandwidth is to use *token bucket* filter that can describe bandwidth characteristics of a source/flow.

The two parameters used are token rate `r` and a bucket depth `B` A token is required to send a byte of data.

A source can accumulate tokens at rate `r`/second, but not more than `B` tokens.

Bursty data of more than `r` bytes per second is not permitted. Therefore bursty data should be spread over a long interval.

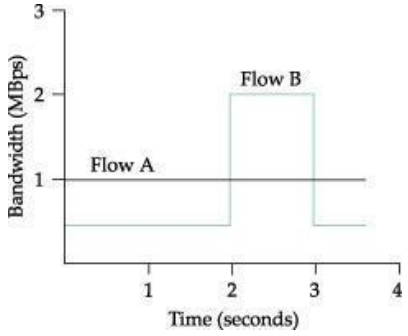
The token bucket provides information that is used by admission control algorithm to determine whether or not to consider the new request for service.

Example

Flow A generates data at a steady rate of 1 Mbps, which is described using a token bucket filter with rate `r` = 1 Mbps and a bucket depth `B` = 1 byte.

Flow B sends at rate of 0.5 Mbps for 2 seconds and then at 2 Mbps for 1 second, which is described using a token bucket filter with rate `r` = 1 Mbps and a bucket depth `B` = 1 MB.

The additional depth allows it to accumulate tokens when it sends 0.5 Mbps (2 × 0.5 = 1 MB) and uses the same to send for bursty data of 2 Mbps.



Admission Control

When a flow requests a level of service, admission control examines *TSpec* and *RSpec* of the flow.

It checks to see whether the desired service can be provided with currently available resources, without causing any worse service to previously admitted flows.

- If it can provide the service, the flow is admitted otherwise denied.

The decision to allow/deny a service can be *heuristic* such as "currently delays are within bounds, therefore another service can be admitted."

Admission control is closely related to *policy*. For example, a network admin will allow CEO to make reservations and forbid requests from other employees.

Reservation Protocol (RSVP)

The Resource Reservation Protocol (RSVP) is a signaling protocol to help IP create a flow and make a resource reservation.

RSVP provides resource reservations for all kinds of traffic including multimedia which uses multicasting. RSVP supports both unicast and multicast flows.

RSVP is a robust protocol that relies on *soft state* in the routers.

- Soft state unlike hard state (as in ATM, VC), times out after a short period if it is not refreshed. It does not require to be deleted.
- The default interval is 30 ms.

Since multicasting involves large number of receivers than senders, RSVP follows *receiver-oriented* approach that makes receivers to keep track of their requirements.

RSVP Messages

To make a reservation, the receiver needs to know:

- What traffic the sender is likely to send so as to make an appropriate reservation, i.e., *TSpec*.
- Secondly, what path the packets will travel.

The sender sends a PATH message to all receivers (*downstream*) containing *TSpec*.

A PATH message stores necessary information for the receivers on the way.

PATH messages are sent about every 30 seconds.

The receiver sends a reservation request as a RESV message back to the sender (*upstream*), containing sender's *TSpec* and receiver requirement *RSpec*.

Each router on the path looks at the RESV request and tries to allocate necessary resources to satisfy and passes the request onto the next router.

- If allocation is not feasible, the router sends an *error* message to the receiver

If there is any failure in the link a new path is discovered between sender and the receiver. The RESV message follows the new path thereafter.

A router reserves resources as long as it receives RESV message, otherwise released. If a router does not support RSVP, then best-effort delivery is followed.

Reservation Merging

In RSVP, the resources are not reserved for each receiver in a flow, but merged.

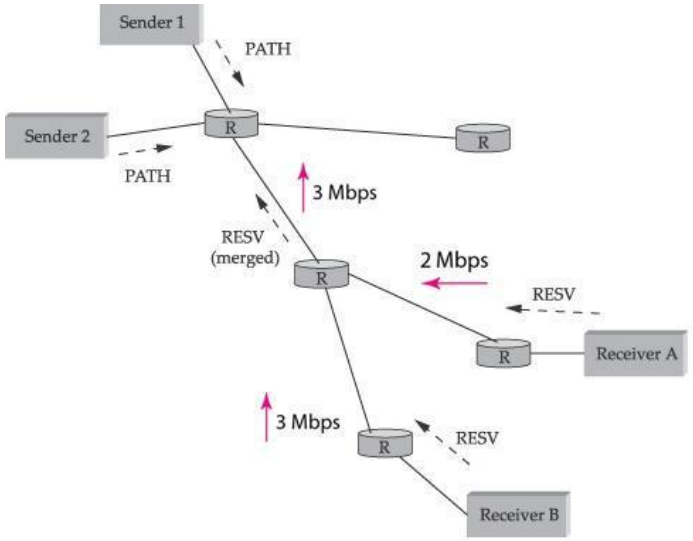
When a RESV message travels from receiver up the multicast tree, it is likely to come across a router where reservations have already been made for some other flow.

If new resource requirements can be met using existing allocations, then new allocation is not done.

- For example, receiver *B* has already made a request for 3 Mbps. If *A* comes with a new request for 2 Mbps, then no new reservations are made.

A router that handles multiple requests with one reservation is known as *merge point*. This is because, different receivers require different quality.

Reservation merging meets the needs of all receivers downstream of the *merge point*.



Packet classification is done by examining the fields *source address*, *destination address*, *protocol number*, *source port* and *destination port* in the packet header.

Weighted fair queuing or a combination of queuing disciplines is used.

List the disadvantages of integrated services

Scalability IntSrv requires router to maintain information for each flow, which is not feasible for today's internet growth

Service type limitation Only two types of services are provided. Certain applications may require more than the offered services.

Explain how QoS is provided through differentiated services

Differentiated Services (*DiffServ*) is a class-based QoS model designed for IP.

Premium class

The default *best-effort* model is enhanced as a new class called *premium*.

The premium packets have bits set (*marked*) in the header by the gateway router or by the ISP router.

IETF has defined a set of behaviors for routers known as per-hop behaviors (*PHB*).

IETF has replaced the existing TOS field in IPv4 or Class field in IPv6 with 6-bit DiffServ code points (*DSCP*) and remaining 2 bits unused.

6-bit DSCP can be used to define 64 PHB that could be applied to a packet.

The three PHBs defined are *default* PHB (DE PHB), *expedited forwarding* PHB (EF PHB) and *assured forwarding* PHB (AF PHB).

The DE PHB is the same as best-effort delivery and is compatible with TOS.

Expedited Forwarding (EF PHB)

Packets marked for EF treatment should be forwarded by the router with minimal delay (*latency*) and loss by ensuring required bandwidth.

A router guarantees EF, only if arrival rate of EF packets is less than forwarding rate

The rate limiting of EF packets is achieved by configuring routers at the edge of an administrative domain to ensure that it is less than bandwidth of the slowest link.

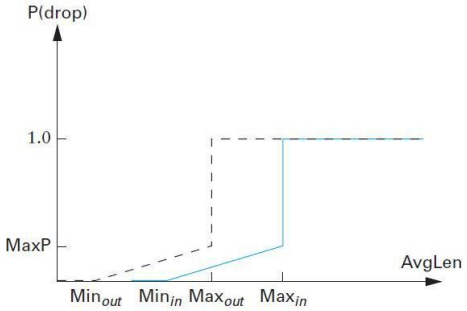
Queuing can be either using strict priority or weighted fair queuing.

- In strict priority, EF packets are preferred over others, leaving less chance for other packets to go through.
- In weighted fair queuing, other packets are given a chance, but there is a possibility of EF packets being dropped, if there is excessive EF traffic.

Assured Forwarding

The AF PHB is based on RED with In and Out (*RIO*) algorithm.

In RIO, the drop probability increases as the average queue length increases. The following example shows RIO with two classes named *in* and *out*.



The *out* curve has a lower MinThreshold than *in* curve, therefore under low levels of congestion, only packets marked *out* will be discarded.

If the average queue length exceeds Min_{in}, packets marked *in* are also dropped.

The terms *in* and *out* are explained with the example "Customer X is allowed to send up to *y* Mbps of assured traffic".

- If the customer sends packets less than *y* Mbps then packets are marked *in*.
- When the customer exceeds *y* Mbps, the excess packets are marked *out*.

Thus combination of profile meter at the edge router and RIO in all routers, assures (*but does not guarantee*) the customer that packets within the profile will be delivered

RIO does not change the delivery order of *in* and *out* packets.

If weighted fair queuing is used, then weight for the premium queue is chosen using the formula. It is based on the load of premium packets.

$$B_{premium} = W_{premium} / (W_{premium} + W_{best\text{-}effort})$$

- For example, if weight of premium queue is 1 and best-effort is 4, then only 20% of the link is reserved for premium packets.

How differentiated services overcome the limitations of integrated services?

1. The main processing was moved from the core of the network to edge of the network (*scalability*). Thus routers need not store information about flows. The applications define the type of service they need each time when a packet is sent.
2. The per-flow service is changed to per-class service. The router routes the packet based on class of service defined in the packet, not the flow. Different types of classes (*services*) based on the needs of applications.

Write short notes on ATM QoS.

The five ATM service classes are:

1. constant bit rate (*CBR*)
2. variable bit rate—real-time (*VBR-rt*)
3. variable bit rate—non-real-time (*VBR-nrt*)
4. available bit rate (*ABR*)
5. unspecified bit rate (*UBR*)

Constant Bit Rate

Sources of CBR traffic are expected to send at a constant rate.

The source's peak rate and average rate of transmission are equal.

CBR class is designed for customers who need real-time audio or video services. CBR is a relatively easy service for implementation

Variable Bit Rate

The VBR class is divided into two subclasses: real-time (*VBR-rt*) and non-real-time (*VBR-nrt*).

VBR-rt is designed for users who need real-time services (such as voice and video transmission) and use compression techniques to create a variable bit rate.

The traffic generated by the source is characterized by a token bucket, and the maximum total delay required through the network is specified.

VBR-nrt bears some similarity to IP's controlled load service. The source traffic is specified by a token bucket.

VBR-nrt is designed for users who do not need real-time services but use compression techniques to create a variable bit rate

Unspecified Bit Rate

UBR class is a best-effort delivery service that does not guarantee anything. UBR allows the source to specify a maximum rate at which it will send.

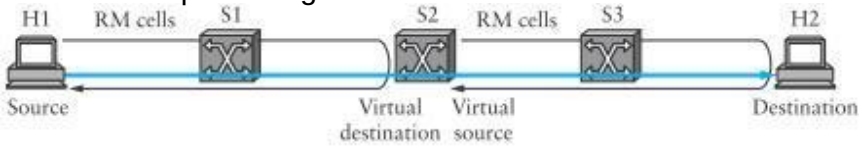
- Switches may make use of this information to decide whether to admit or reject or negotiate with the source for a less peak rate.

Available Bit Rate

ABR apart from being a service class also defines a set of congestion-control mechanism.

The ABR mechanisms operate over a virtual circuit by exchanging special ATM cells called resource management (RM) cells between the source and destination.

RM cells work as explicit congestion feedback mechanism as shown below.



ABR allows a source to increase or decrease its allotted rate as conditions dictate.

ABR class delivers cells at a minimum rate. If more network capacity is available, this minimum rate can be exceeded.

ABR is suitable for applications that are bursty in nature.

What is equation based congestion control?

TCP's congestion-control algorithm is not appropriate for real-time applications.

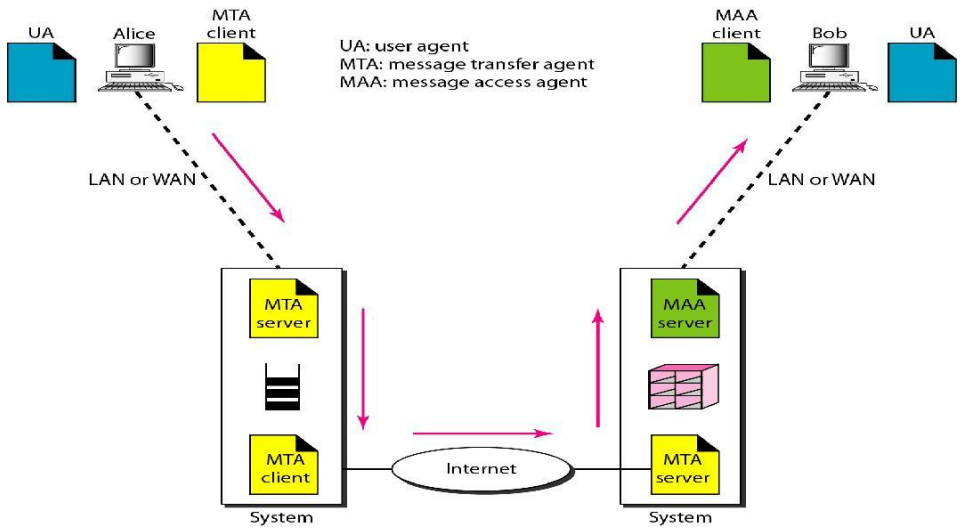
A smooth transmission rate is obtained by ensuring that flow's behavior adheres to an equation that models TCP's behavior.

$$\text{Rate} = \left(\frac{1}{\text{RTT} \times \sqrt{\rho}}\right)$$

To be TCP-friendly, the transmission rate must be inversely proportional to the round-trip time (RTT) and the square root of the loss rate ( ).

Discuss the components of an email system and the protocols used.

The e-mail system involved in Alice sending a message to Bob is shown.



User Agent

A user agent (UA) is software that is either *command* (eg. pine, elm) or *GUI* based (eg. Microsoft Outlook, Netscape). It facilitates:

- *Compose* helps to compose messages by providing template with built-in editor.
- *Read* checks mail in the incoming box and provides information such as sender, size, subject and flag (read, new).
- *Reply* allows user to reply (send message) back to sender
- *Forward* facilitates forwarding message to a third party.
- *Mailboxes* creates two mailboxes for each user, namely *inbox* (to store received emails) and *outbox* (to keep all sent mails).

Message Format

RFC822 defines message to have two parts namely *header* and a *body*.

The message header is a series of <CRLF> terminated lines. Each header line contains a *type* and *value* separated by a colon (:). It is filled by the user/system. Some of them are:

- From user who sent the message
- To identifies the message recipient(s).
- Subject says something about the purpose of the message
- Date when the message was transmitted
- E-mail address consists of *user_name@domain_name* where domain_name is hostname of the *mail server*.

The body of the message contains the actual information

- The header is separated from the message body by a *blank* line.

Initially email system was designed to send messages only in NVT 7-bit ASCII format.

- Languages such as French, German, Chinese, Japanese were not supported.
- Image, audio and video files cannot be sent.

***Multipurpose Internet Mail Extensions* (MIME)**

MIME is a supplementary protocol that allows *non-ASCII* data to be sent through e-mail.

MIME transforms non-ASCII data to NVT ASCII and delivers to client MTA. The NVT ASCII data is converted back to non-ASCII form at the recipient mail server.

MIME defines five headers. They are:

- MIME-Version specifies the current version 1.1
- Content-Type specifies message type such as *text* (plain, html), *image* (jpeg, gif), *audio*, *video* and *application* (postscript, msword). If more than one type exists, then it is termed as *multipart* (mixed).
- Content-Transfer-Encoding defines how data in the message body is encoded such as *binary*, *base64*, *7-bit*, etc.
- Content-Id unique identifier the whole message in a multiple message type.
- Content-Description describes type of the message body.

Example

MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="-------
417CA6E2DE4ABCAFBC5" From: Alice Smith <Alice@cisco.com>
To: Bob@cs.Princeton.edu
Subject: promised material
Date: Mon, 07 Sep 1998 19:45:19 -0400
---------417CA6E2DE4ABCAFBC5 Content-
Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
…
---------417CA6E2DE4ABCAFBC5
Content-Type: image/jpeg Content-
Transfer-Encoding: base64

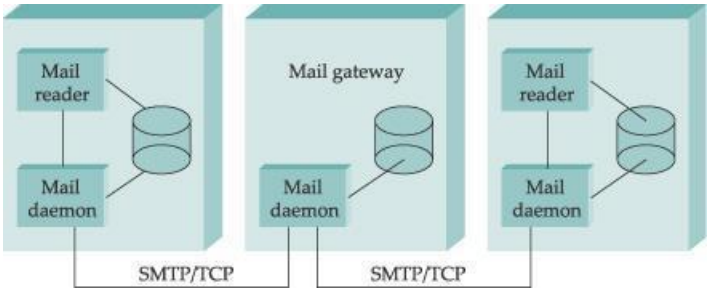
Message Transfer Agent (MTA): SMTP

Message Transfer Agent (MTA) is a mail daemon that helps to transmit/receive message over the network.

To send mail a system must have the client MTA, and to receive mail a system must have a server MTA.

Simple Mail Transfer Protocol (SMTP) defines communication between client/server MTA.

SMTP uses TCP connection on port 25 to forward the entire message and store at intermediate mail servers/mail gateways until it reaches the recipient mail server.



SMTP defines how commands and responses must be sent back and forth.

Command	Description
MAIL FROM	Sender of the message
RCPTTO	Recipient of the message
DATA	Body of the mail
QUIT	Terminate
VRFY	Name of recipient to be verified before forwarding
EXPN	Mailing list to be expanded

Common responses sent from server MTA are:

Code	Description
220	Service ready
250	Request completed
354	Start mail input
450	Mailbox not available
500	Syntax error; unrecognized command
551	User not local

Example

HELO cs.princeton.edu
250 Hello daemon@mail.cs.princeton.edu
[128.12.169.24] MAIL FROM:<Bob@cs.princeton.edu>
250 OK
RCPT TO:<Alice@cisco.com>
250 OK
DATA
354 Start mail input; end with <CRLF>.<CRLF>
Blah blah blah...
...etc. etc. etc.
<CRLF>.<CRLF>
250 OK
QUIT
221 Closing connection

In each exchange, the client posts a command and the server responds with a code. and a human-readable explanation for the code.
After the commands and responses, client sends the message which is ended by a period (.) and terminates the connection.

Message Access Agent (MAA)/Mail Reader: POP and IMAP

MAA or mail reader allows user to retrieve messages from the mailbox, so that user can perform actions such as reply, forwarding, etc.

The two message access protocols are:

- Post Office Protocol, version 3 (POP3)
- Internet Mail Access Protocol, version 4 (IMAP4)

SMTP is a push type protocol whereas POP3 and IMAP4 are pop type protocol.

POP3

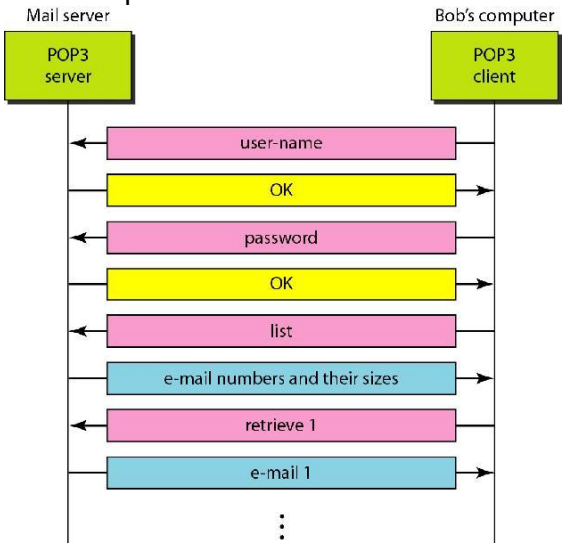
POP3 is simple and limited in functionality

POP3 works in two modes namely, *delete* and *keep* mode.

- In delete mode, mail is deleted from the mailbox after retrieval
- In keep mode, mail after reading is kept in mailbox for later retrieval.

POP3 client is installed on the recipient computer and POP3 server on the mail server. The client opens a connection to the server on TCP port 110.

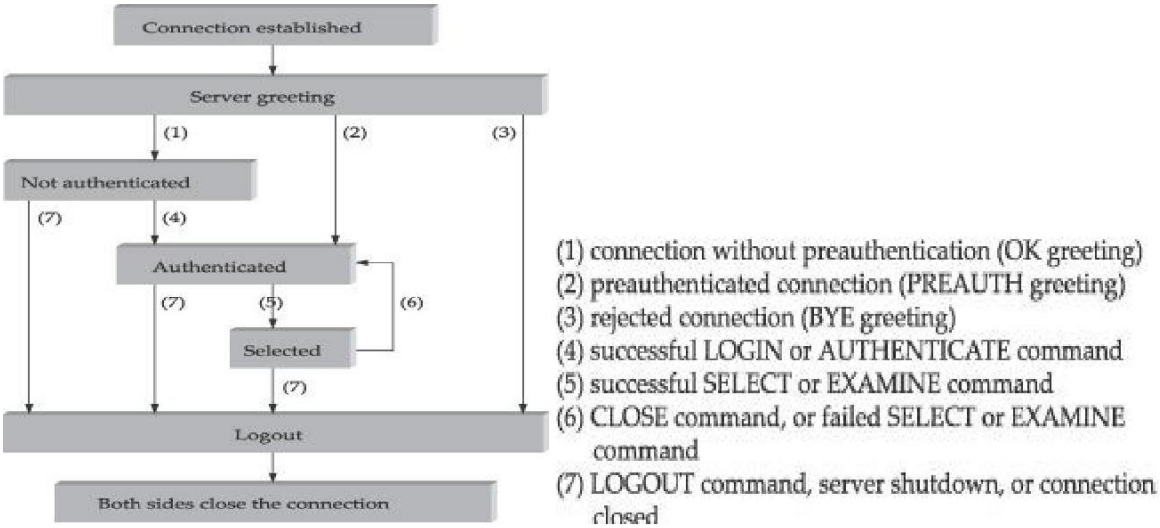
The client sends username and password to access the mailbox and retrieve the messages.



IMAP4

IMAP is a client/server protocol running over TCP. The client issues commands and the mail server responds.

The exchange begins with the client authenticating itself to access the mailbox. This is represented as a state transition diagram.



When the user asks to FETCH a message, server returns it in MIME format and the mail reader decodes it.

IMAP also defines message *attributes* such as size and *flags* such as Seen, Answered, Deleted and Recent.

Explain HTTP protocol in detail.

WWW is a distributed client/server service, in which a client (*web browser*) can access a service through a server, where the service is distributed over many locations called *sites*.

Web browsers allow users to access files using uniform resource locator (URL).

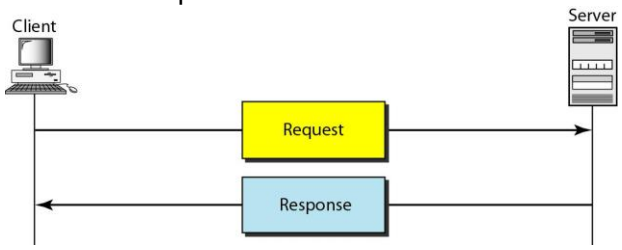
When user enters URL, the browser forms a *request* message and sends to the server.

The server retrieves the requested URL and sends it as a *response* message.

The browser displays the response in HTML / appropriate format.

HTTP is a stateless request/response protocol that governs client/server communication.

HTTP uses TCP connection on port 80 to transfer data between client and server.



Message Format

START_LINE <CRLF>
MESSAGE_HEADER
<CRLF> <CRLF>
MESSAGE_BODY <CRLF>

Request Message

Request Line
Request Header : *Value*
Blank Line
Body (optional)

Request Line

The Request line contains three fields as shown.

Request Type	*URL*	*HTTP version*

- *HTTP version* specifies current version of the protocol i.e., 1.1
- The *Request type* specifies methods that operate on the URL are:

Method	**Description**
GET	retrieve document specified as URL
HEAD	retrieve meta-information about the URL document
POST	send information from client to the server
PUT	store document under specified URL
TRACE	echoes the incoming request
OPTION	request information about available options
DELETE	delete specified URL

For example, the request line to retrieve file index.html on host cs.princeton.edu is

GET http://www.cs.princeton.edu/index.html HTTP/1.1

Request Header

Request Header specifies client's configuration and preferred document format:

Request Header	Description
Accept-charset	specifies the character set the client can handle
Authorization	specifies what permissions the client has
From	specifies e-mail address of the user
Host	specifies host name and port number of the server
If-modified-since	server sends the URL if it is newer than specified date
Referrer	specifies URL of the linked document
User-agent	specifies name of the browser

The above example using request header is specified as

```
GET index.html HTTP/1.1
Host: www.cs.princeton.edu
```

Response Messages

Status Line
Response Header : *Value*
Blank Line
Body

Status line

The Status line contains three fields as shown.

HTTP version	*Code*	*Status Phrase*

The status code field consists of three digits (1xx–Informational, 2xx–Success, 3xx– Redirection, 4xx–Client Error, 5xx–Server Error)

The status phrase explains the status code in text form. Some of them are:

Code	Phrase	Description
100	Continue	Initial request received, client to continue process
200	OK	Request is successful
201	Created	A new URL is created.
204	No content	There is no content in the body.
301	Moved permanently	The requested URL is no longer in use
304	Not modified	The document has not been modified
401	Unauthorized	The request lacks proper authorization
404	Not found	The document is not found
500	Internal server error	There is an error, such as a crash, at the server site

For example, the server reports as follows, if the requested file is not found

```
HTTP/1.1 404 Not Found
```

Response Header

Response Header	Description
Content-encoding	specifies the encoding scheme
Content-length	shows length of the document
Content-type	specifies the medium type
Expires	gives date and time up to which the document is valid
Last-modified	gives date and time when the document was last updated
Location	specifies location of the created or moved document

The response for a moved page is given below.

HTTP/1.1 301 Moved Permanently
Location: http://www.princeton.edu/cs/index.html.

Persistent vs non-persistent connection

In *non-persistent*, a TCP connection is required for each request/response

- Imposes high overhead on the server because the server needs N buffers for N URL pointers and TCP overhead for each connection

In *persistent*, Client and server can exchange multiple request/response messages over the same TCP connection

- Eliminates the connection setup overhead and load on the server
- TCP's congestion window mechanism is able to operate more efficiently.
- The server times out, if there is no request from the client for a specified period

Caching

Caching enables the client to retrieve document faster and reduces load on the server. Caching can be implemented at different places

- For example, the ISP router can cache pages. Further such request coming from its clients, the ISP responds.
- Proxy server is a host that keeps copies responses to recent requests. The client sends request to the proxy server. The proxy server either responds to client or forwards the request to the server.
- The browser also can cache pages.

Server assigns expiration date (using Expires header field) to each page, beyond which the page should not be cached.

Therefore prior to caching a page, its expiration date is checked. If a cached page reaches its expiration, then the page is deleted.

The proxy verifies whether it has the latest document by using If-Modified-Since header. A page must not be cached if no-cache directive is specified.

***Explain the role of DNS on a computer network*.**

Domain-names are easily remembered than IP address of a host, since it is user-friendly. Thus, need for a system to map domain name to an IP address that includes:

- A *namespace* to define domain names without conflict.
- *Binding* of domain names to IP address
- A *name server* that returns IP address for a given name

During early days of internet, there were only few hundred hosts

- A central authority called the Network Information Center (NIC) maintained name-to-address bindings in a flat-file called *hosts.txt*
- A new host that joins the internet would mail its name and IP address to NIC.
- NIC updates *hosts.txt* and mails to all hosts.
- Name server resolved domain names using a simple *lookup* on hosts.txt

As hosts grew to thousands and millions, the flat file approach failed, leading to evolution of DNS in mid 1980s.

Name Hierarchy

DNS uses *hierarchical* name space for domains in the Internet.

Hierarchical naming permits use of same sub-domain name in different domains. Domain names are case insensitive and can be up to 63 characters

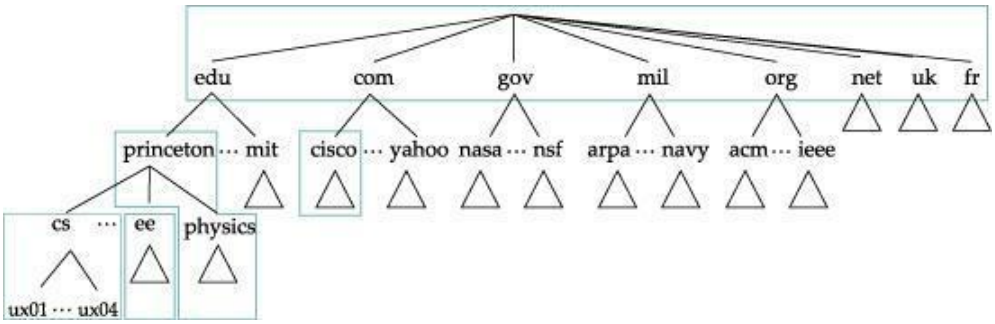
DNS names are processed from right to left and use *periods* (.) as separator.

DNS can be used to map names to values, not necessarily domain names to IP address.

DNS hierarchy can be visualized as a tree, where each *node* in the tree corresponds to a domain and the *leaves* relate to hosts.

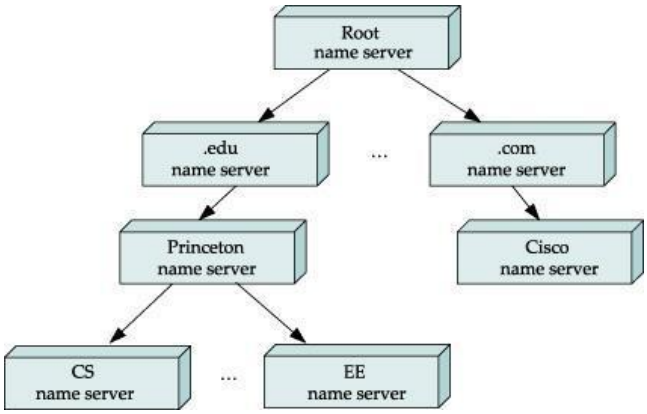
Six big domains are .edu (education) .com (commercial) .gov (US government) .mil (US military) .org (non-profitable organization) and .net (network providers).

Top level domain exist one for each country .fr (france) .in (india), etc.



Name Servers

The domain hierarchy is partitioned into *zones*. Topmost domains are managed by NIC. Each zone acts as *central* authority for that part of the sub-tree.



In the .edu hierarchy, *princeton* is a zone.

- Each zone can be further sub-divided that manage using their own name servers such as CS department under princeton university.

Each zone information is implemented on at least two name servers.

Clients send queries to name servers, and name servers respond to it.

The response contains either the host IP address or address of another name server Each name server contains a collection of *resource records*.

A resource record is a name-to-value binding and is a 5-tuple with the following fields

Name	Value	Type	Class	TTL

Name tells the domain to which this record applies. It is the primary search key, used to satisfy queries

The Type field tells what kind of record it is. Some commonly used types are:

- NS *Value* field contains a name server
- CNAME *Value* field contains canonical name for the host. Used to define aliases.
- MX *Value* field contains a mail server that accepts messages for the domain.
- A *Value* field contains an IP address

The Value field can be a number, a domain name, or an ASCII string. The semantics depend on the record type

For internet information, the Class field is always IN.

The TTL field gives an indication of how long the resource record is valid.

Root name server

The root name server contains an NS record for each second-level server.

It also has an A record that translates this name into corresponding IP address. The following shows part of .edu root name server

(princeton.edu, cit.princeton.edu, NS, IN)
(cit.princeton.edu, 128.196.128.233, A, IN)
…

Zone name server

The zone name server princeton.edu has a name server available on host cit.princeton.edu that contains the following records.

Some records contain A records, whereas others point to next level name servers.

(cs.princeton.edu, gnat.cs.princeton.edu, NS,
IN) (gnat.cs.princeton.edu, 192.12.69.5, A, IN)
…

Eventually, third-level name server, such as the domain cs.princeton.edu, contains A records for all of its hosts.

(cs.princeton.edu, gnat.cs.princeton.edu, MX, IN)
(cicada.cs.princeton.edu, 192.12.69.60, A, IN)
(cic.cs.princeton.edu, cicada.cs.princeton.edu, CNAME,
IN) (gnat.cs.princeton.edu, 192.12.69.5, A, IN)

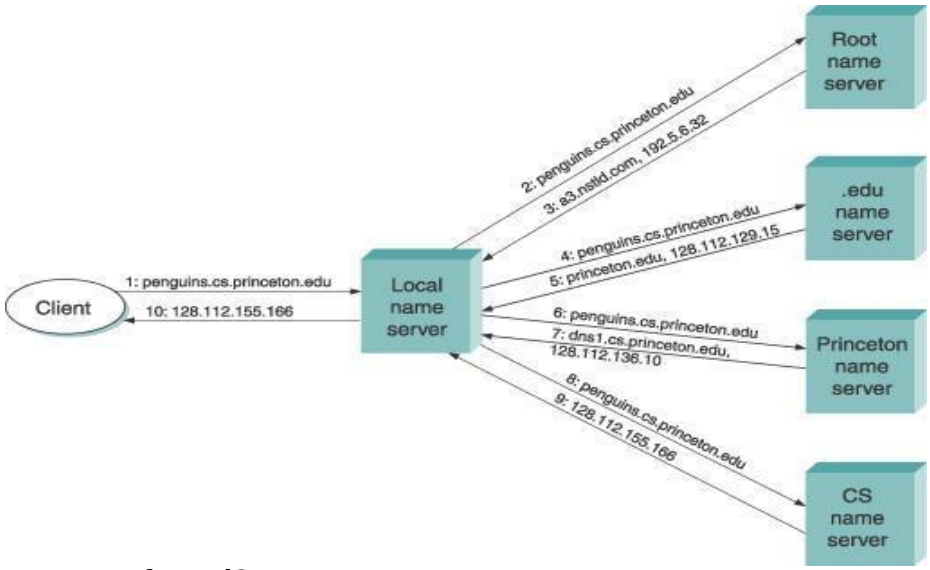
Name Resolution for cicada.cs.princeton.edu

1. The client first sends a query containing cicada.cs.princeton.edu to the *root* server.
2. The root server, does not finds an exact match, but locates the *NS* record for princeton.edu

3. The root returns the A record for princeton.edu back to the client.
4. The client sends the same query to 128.196.128.233 and receives the A record for cs.princeton.edu
5. Finally the client sends the query to 192.12.69.5 and gets the A record for cicada.cs.princeton.edu

Drawbacks

All hosts should know the root name server, which is not feasible.
Instead, the client can send query to the local name server that it knows
The local name server can query the root name server on behalf of the client.
Once the local NS gets the required response, it caches the A record based on TTL and sends the record to the client.



Why is POP not preferred?

It does not allow the user to organize their mail on the server The user cannot have different folders on the server
It does not allow the user to partially check the contents of the mail before downloading

List the advantages of IMAP over POP

IMAP4 is more powerful and more complex than POP3. The additional features are: A user can check the e-mail header prior to downloading.
A user can search 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.

Explain how SNMP is used to manage nodes on the network

Simple Network Management Protocol (SNMP) is a framework for managing devices in an internet using TCP/IP.
It provides a set of fundamental operations for monitoring and maintaining an internet SNMP uses the concept of *manager* and *agent*.

- A manager is a host that runs the SNMP client program.

- A managed station called an agent, is a router that runs the SNMP server program

SNMP is an application layer protocol, therefore it can monitor devices of different manufacturers installed on different physical networks. SNMP management includes:

- A manager that checks an agent by requests information on behavior of the agent.
- A manager forces an agent to perform a task by setting/resetting values in the agent database.
- An agent warns the manager of an unusual situation.

SNMP uses services of UDP on two well-known ports, 161 (agent) and 162 (manager). SNMP is supported by two other protocols in Internet Network management. They are:

- Structure of Management Information (SMI)
- Management Information Base

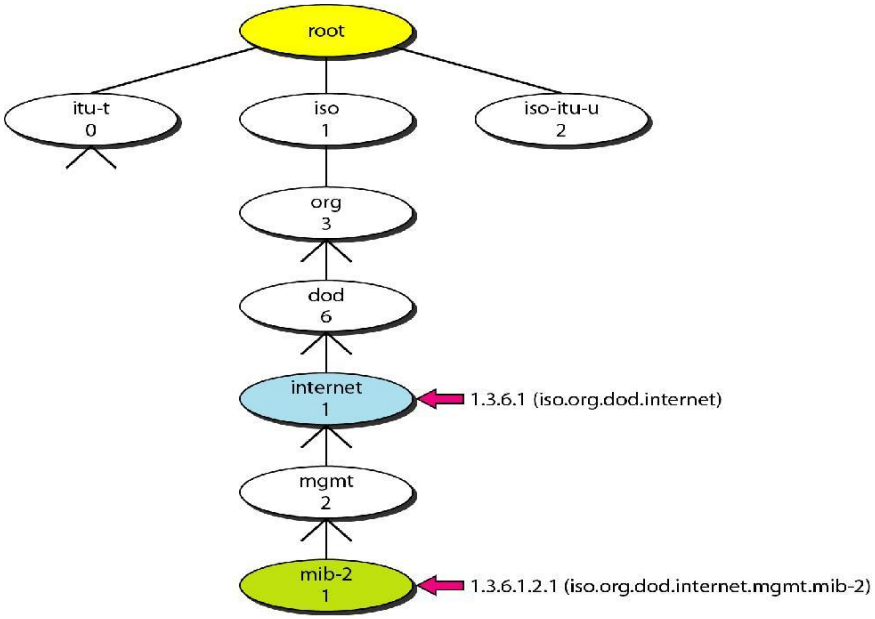
(MIB) The role of SNMP is to

- Define format of the packet to be sent from a manager to an agent and vice versa.
- Interprets the result and creates statistics
- Responsible for reading and setting object values The role of SMI is to
- Define rules for naming objects and object types.
- Uses *Basic Encoding Rules* to encode data to be transmitted over the network. The role of MIB is to
- creates a collection of named objects, their types, and their relationships to each other in an entity to be managed

Object Identifier

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 objects that are used in SNMP are located under the mib-2 object, so their identifiers always start with 1.3.6.1.2.1

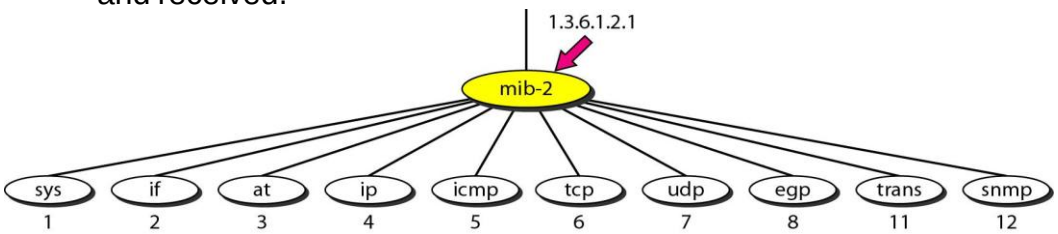
Object identifiers follow lexographic ordering.

MIB Groups

Each agent has its own MIB2 (version 2), which is a collection of all the objects that the manager can manage.

The objects in MIB2 are categorized under 10 different groups namely *system*, *interface*, *address translation*, *ip*, *icmp*, tcp, *udp*, *egp*, *transmission*, and *snmp*.

- sys (*system* defines general information about the node such as the name, location, and lifetime.
- if (*interface* defines information about all the interfaces of the node such as physical address and IP address, packets sent and received on each interface, etc.
- at (*address translation* defines information about the ARP table
- ip defines information related to IP such as the routing table, statistics on datagram forwarding, reassembling and drop, etc.
- tcp defines general information related to TCP, such as the connection table, time-out value, number of ports, and number of packets sent and received.
- udp information on UDP traffic such as total number of UDP packets sent and received.

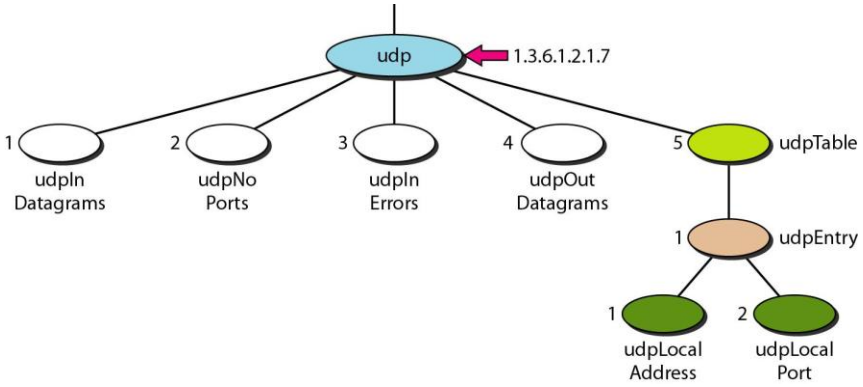


MIB variables

MIB variables are of two types namely *simple* and *table*.

To access any of the simple variable content, use *id* of the *group* (1.3.6.1.2.1.7) followed by the id of the *variable* and an instance suffix, which is 0.

- For example, variable *udpInDatagrams* is accessed as 1.3.6.1.2.1.7.1.0

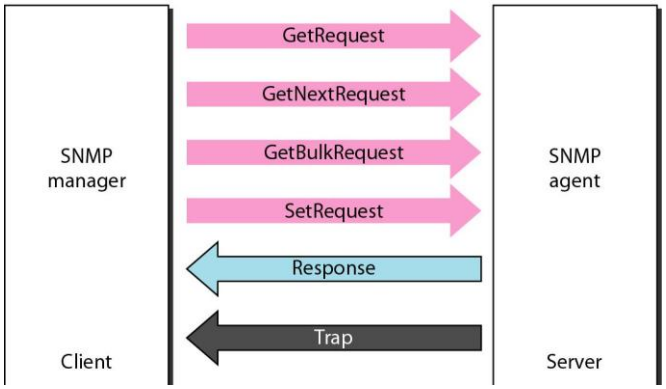


In case of table, only leaf elements are accessible.

- In this case, the group id is followed by table id and so on up to the leaf element.
- To access a specific instance (row) of the table, add the index to the above ids.
- The indexes are based on the value of one or more fields in the entries.
- Tables are ordered according to column-row rules, i.e one should go column by column from top to bottom.

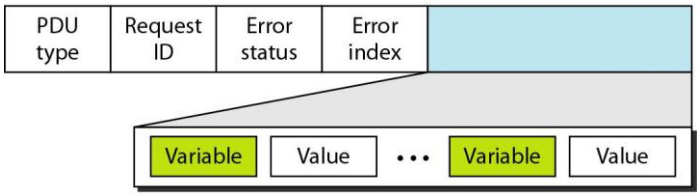
SNMPv3 PDU

SNMP is request/reply protocol that defines PDUs GetRequest, GetNextRequest, GetBulkRequest, SetRequest, Response and Trap.



- GetRequest used by manager to retrieve value of agent's variable(s)
- GetNextRequest used by manager to retrieve next entries in a agent's table
- SetRequest used by manager to set value of an agent's variable
- Response sent from an agent to manager in response to GetRequest/ GetNextRequest that contains value of variables
- Trap sent from an agent to the manager to report an event such as reboot.

PDU Format



The SNMP client puts the identifier for the MIB variable it wants to get into the request message, and sends this message to the server.

The server then maps this identifier into a local variable, retrieves the current value held in this variable, and uses BER to encode the value it sends back to the client.

Discuss Telnet in detail

TErminaL NETwork (TELNET) is a general-purpose client/server application program. TELNET is the standard TCP/IP protocol for virtual terminal.

TELNET enables connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system.

TELNET was designed during days of time-sharing environment in which a large computer supported multiple users.

Interaction between user and computer occurs through a terminal (keyboard + monitor + mouse).

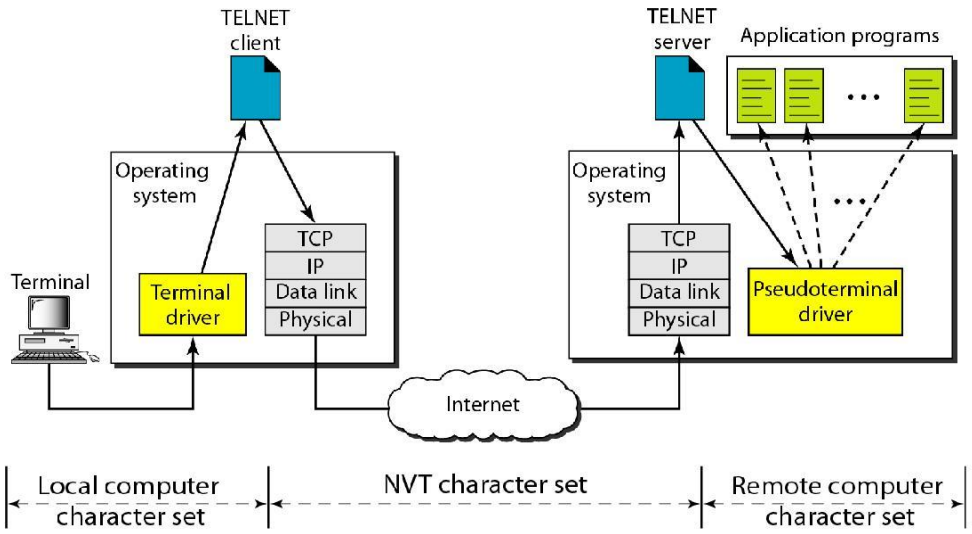
Each user has an identification name and a password.

To access, user logs into the system with a user id / log-in name.

The user is authenticated using password and hence unauthorized access is prevented.

Remote Logon

The process of remote login using TELNET client and server program is shown.



The user keystrokes are sent to the terminal driver, where the local operating system accepts the characters but does not interpret them.

The characters are sent to the TELNET *client*, which transforms the characters to a universal character set called *Network Virtual Terminal* (NVT) characters and puts it over the network.

The commands/text in NVT form reaches the remote host.

The TELNET *server* at well-known port 23, converts NVT characters onto remote character set.

Since the operating system is not designed to receive data from TELNET server, data is redirected via a pseudo terminal driver to the remote operating system.

The remote operating system passes the data to the corresponding applications.

NVT Character Set

Every operating system use a special combination of characters as tokens

- For example, the *end-of-file* token in DOS is Ctrl+z, whereas in UNIX it is Ctrl+d.

TELNET solves the problem of heterogeneity, by defining a universal interface called the network virtual terminal (NVT) character set.

Data transmitted over the network is NVT, whereas at the host level data is processed using its own character set.

NVT uses two sets of 8-bit characters, one for data and the other for control.

- For data, the MSB is 0 and for control it

is 1. Some NVT control characters are:

Character	Purpose
EOF	End of file
EOR	End of record
IP	Interrupt process
AYT	Are you there
EC	Erase character
EL	Erase line
IAC	Interrupt as control

TELNET uses the same connection to send both data and control characters.

To distinguish data from control characters, each sequence of control characters is preceded by a special control character called IAC.

For example, to display file1, the command is cat file1, by mistake the user types cat filea<*backspace*>1.

c	a	t		f	i	l	e	a	IAC	EC	1

Options

TELNET lets the client and server negotiate options before or during the session.

Options are extra features available with a more sophisticated terminal whereas simple terminals use default features. Some options are

Options	Purpose
Echo	Echo the received data to the sender
Status	Request the status of TELNET
Line mode	Change to line mode.

The control characters used for option negotiation are WILL, WONT, DO and DONT.

Modes

TELNET operate in three modes namely *default*, *character* and *line* mode.

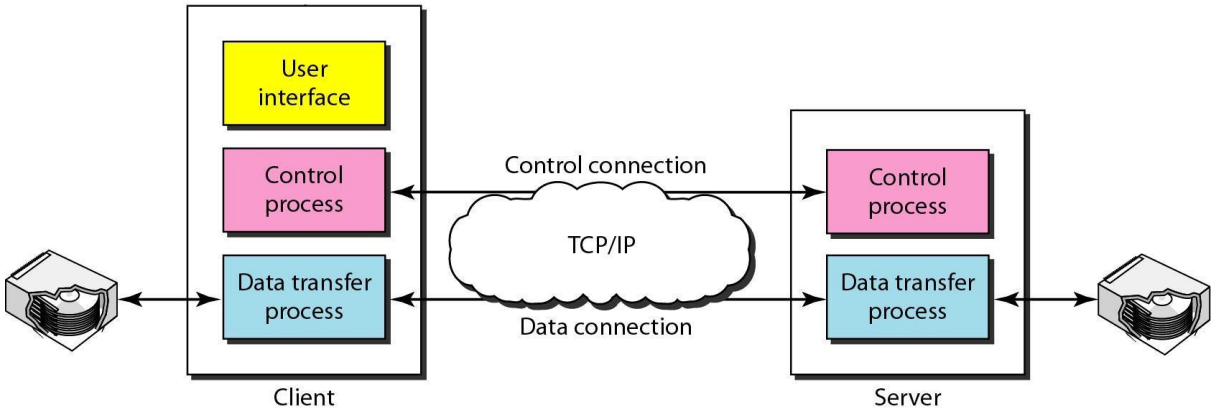
- In *default* mode, the client sends characters only after the line is typed.
- In *character* mode, each character typed is sent by the client to the server.
- In *line* mode, line editing is done by the client and sends after a line is typed

Briefly explain the transfer of file contents using FTP.

File Transfer Protocol (FTP) is the standard provided by TCP/IP for copying a file from one host to another.

FTP establishes two connections between hosts

- *Data* connection is used for data transfer
- *Control* connection is used for control information.
- FTP uses two well-known TCP ports, 21 for control and 20 for data connection.



Control Connection

FTP uses 7-bit NVT ASCII character set to communicate across the control connection. Communication is achieved through commands and responses.

Each command or response is only one short line terminated with <CRLF>

When a user starts an FTP session, the control connection opens.

While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.

Data Connection

File transfer occurs over the data connection under the control of the commands sent over the control connection.

A file transfer in FTP means one of the following:

- A file is to be copied from the server to the client. This is called *retrieving* a file. It is done under the supervision of the RETR command
- A file is to be copied from the client to the server. This is called *storing* a file. It is done under the supervision of the STOR command.
- A list of directory or file names is to be sent from the server to the client. This is done under the supervision of the LIST command.

The client defines the *type* of file to be transferred, the *structure* of the data, and the *transmission* mode.

Before sending the file through the data connection it is prepared for transmission through the control connection.

File Type

FTP can transfer either an *ASCII* file, *EBCDIC* file, or *image* file. ○ ASCII file is the default format for transferring text files. ○ IBM uses EBCDIC encoding.

- The image file is the default format for transferring binary files.

Data Structure

FTP interprets file's data structure as either *file*, *record* or *page* structure. ○ In file structure, the file is a continuous stream of bytes.

- In record structure, the file is divided into records (used only for text files)
- In page structure, the file is divided into pages. Each page has a page number and header. Page access can be random or sequential.

Transmission Mode

FTP uses *stream* (default), *block* or *compressed* mode of transmission.

- In stream mode, data is delivered to TCP as a continuous stream of bytes. If it's a file structure, end-of-file (EOF) is not needed. In case of record structure, each record is marked by a end-of-record (EOR) and the end of the file has a EOF character.
- In block mode, data is delivered to TCP in blocks, where each block is preceded by a 3-byte header. The first byte is the block descriptor and next 2 bytes define the size.
- In compressed mode, the compression used is run-length encoding. Consecutive appearance of character is replaced by an occurrence and count of repetitions.

Example

```
$ ftp voyager.deanza.tbda.edu
Connected to voyager.deanza.tbda.edu.
220 (vsFTPd 1.2.1)
530 Please login with USER and
PASS. Name: forouzan
331 Please specify the password.
```

Password:
230 Login successful.
ftp> Is reports
150 Here comes the directory listing.
drwxr-xr-x 23027 411 4096 Sep 24 2002 business
drwxr-xr-x 23027 411 4096 Sep 24 2002 school
226 Directory send OK.

What is anonymous FTP?

To use FTP, a user should know user name and password on the remote server.

Some sites have a set of files available for public access, to enable anonymous FTP. To access these files, a user does not need to have an account.

User access to the system is very limited. For example, most sites allow the user to download files.

Write short notes on PGP.

Pretty Good Privacy (PGP) is a popular approach in providing encryption and authentication capabilities for e-mail.

PGP takes note that each user has his own set of criteria by which he/she wants to trust the keys certified by someone else.

- For example, one may trust signed certificates of co-workers than a renowned politician and vice-versa.

PGP provides tools needed to manage the level of trust put in these certificates.

PGP allows certification relationships to form an *arbitrary mesh* and not a rigid hierarchy as in Privacy Enhanced Mail (PEM).

PGP allows each user to decide for themselves how much trust they wish to place in a given certificate

- As the number of trust-worthy signatures for a public key increase, validity for the same and the user's confidence level increases.

PGP key-signing parties are a regular feature of network community meetings such as IETF. The activities include:

- Collect public keys from known persons.
- Share their public key with others
- Get their public key signed by others
- Sign public key of others
- Collect certificate from trust-worthy persons.

PGP stores the set of collected certificates in a file called *key ring*.

PGP allows a wide variety of different cryptographic algorithms to be used

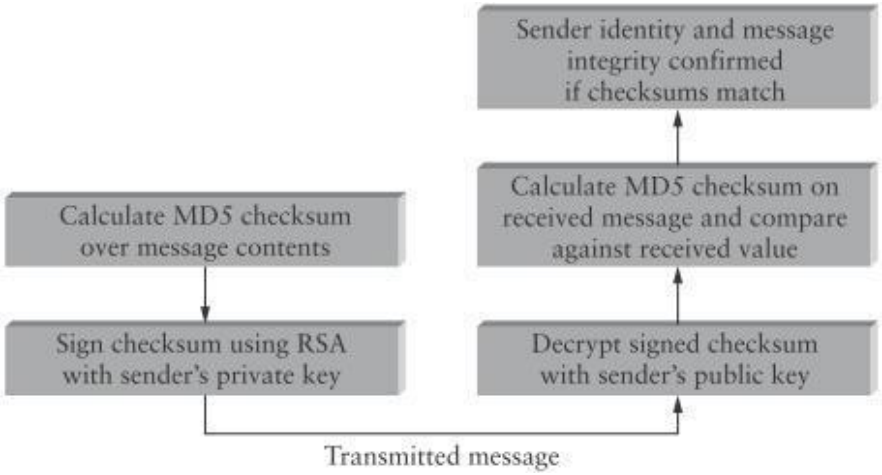
- The actual algorithms used in a message are specified in header fields

PGP allows a user to list his preferred algorithms in the file that contains his/her public key.

Integrity and Authentication

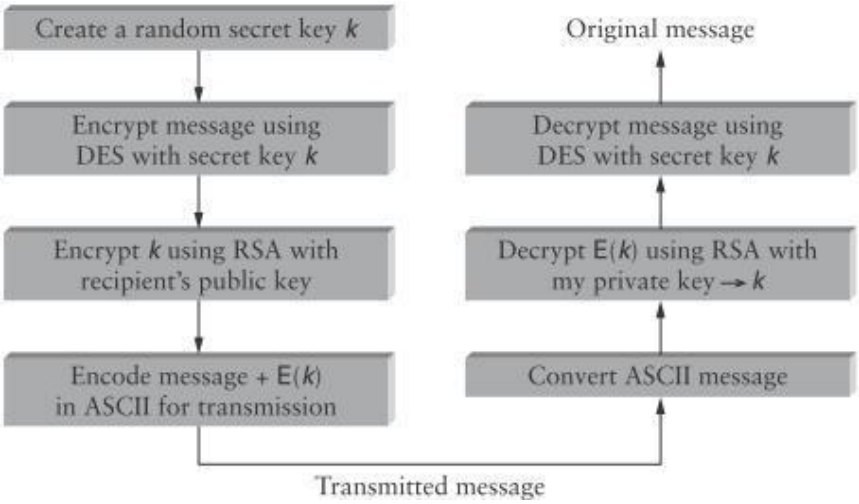
1. Integrity and authentication refers to A sending message to B and proves that it came from A.
2. A creates a cryptographic checksum over the message body, such as MD5 and then encrypts the checksum using A's private key.

3. On receipt of the message, B uses PGP's key management software to search his key ring for A's public key.
4. If key is found
    a. Checksum of the received message is calculated
    b. Encrypted checksum is decrypted using A's public key,
    c. The two checksums are compared. If both are same, then it confirms that A has sent the message and its integrity.
5. If key is not found, the sender and authenticity of the message cannot be verified.
6. Apart from signature verification, PGP also tells B the level of trust previously assigned to this public key.



Encryption

1. A randomly picks a per-message key k to encrypt the message using a symmetric algorithm such as DES
2. The per-message key k is encrypted using B's public key
3. PGP obtains B's public key from A's key ring and notifies A of the level of trust assigned to this key.
4. On receipt, B uses its private key to decrypt the per-message key k.
5. The same algorithm is applied to decrypt the message using per-message key k.



Write short notes on SSH.

Secure Shell (SSH) provides a remote login service in a secure manner SSH uses well-known port 22.

SSH is used to provide strong client/server authentication

- Passwords are not sent as clear text over the network. It is sent in encrypted form.
- Thus sending password through un-trusted network is not a problem Unlike Telnet and rlogin, SSH supports message integrity and confidentiality SSH version 2 consist of the following protocols
- Transport layer protocol SSH-TRANS
- Authentication protocol SSH-AUTH
- Connection protocol SSH-CONN

SSH-TRANS

SSH-TRANS provides an encrypted channel for communication. It runs on top of a TCP connection.

Client and server establish secure channel by having the client authenticate the server using RSA.

- Server informs the client of its public key at the time of connection
- Client warns the user when it tries to connect to the server for the first time, since it does not know the server

Once authenticated, the client and server establish a session key that they will use to encrypt any data sent over the channel.

- Client remembers the server's public key
- For future connection, the client compares server's response with the saved key

SSH-TRANS includes a negotiation of the encryption algorithm the two sides are going to use. For example, 3DES is commonly selected.

SSH-TRANS includes a message integrity check of all data exchanged over the channel.

SSH-AUTH

Server is authenticated during setup of SSH-TRANS channel by default User can authenticate using any of the three mechanism

1) *Login* with username and password. Password is sent in encrypted form
2) *Public key* encryption by asking the user to store user's public key on the server
3) *Host based* authentication requires the client to be authenticated when it connects to server for the first time. Further connection from a trusted host is believed to be from the same user.

In UNIX,

- /.ssh/known_hosts records the keys for all the hosts the user has logged into
- /.ssh/authorized_keys contains the public keys needed to authenticate the user when he or she logs into this machine
- /.ssh/identity contains the private keys for authenticating user on remote machine

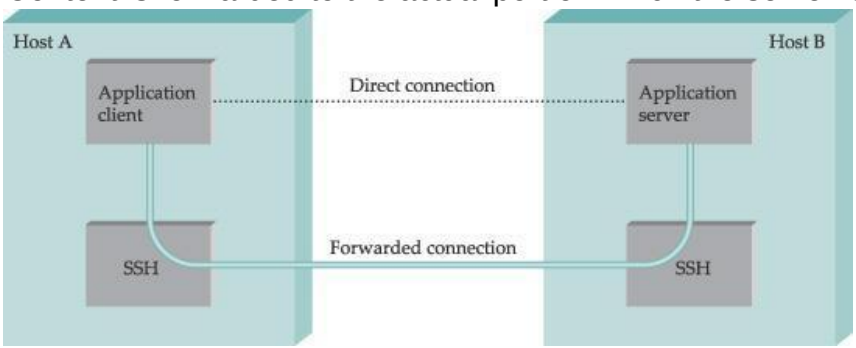
SSH-CONN

SSH can be extended to support insecure TCP applications such as X Windows, IMAP mail readers, etc using SSH-CONN.

Insecure applications are run by tunneling through SSH, known as *port forwarding*.

- Client on host *A* communicates with server on host *B* using SSH.
- Client data sent through SSH is encrypted at sender side

- The receiving SSH at well-known port decrypts the contents
- Content is forwarded to the actual port on which the server is listening



What is Web-based mail?

E-mail is such a common application that some websites today provide this service to anyone who accesses the site such as Hotmail, Yahoo, etc.

Mail transfer from Alice's browser to her mail server is done through HTTP

The message transfer from sending mail server to receiving mail server is through SMTP

Finally, the message from the receiving Web server to Bob's browser is done using HTTP

The website sends a form to be filled in by Bob, which includes log-in id and password.

If the credentials match, the e-mail is transferred from Web server to Bob's browser in HTML format.