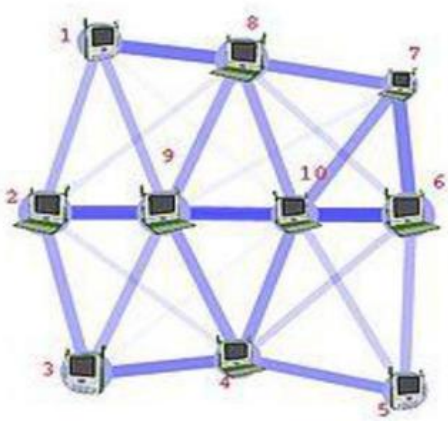
UNIT I

PHYSICAL LAYER: Introduction- Uses, Network Hardware, Software, Reference Models - Theoretical Basis for Communication - Electromagnetic Spectrum, Radio Transmission, Digital Modulation, Baseband Transmission - Transmission Media, Wireless Transmission.

PART – A

1. **Define Computer Networks?**

   A **computer network** or data **network** is a telecommunications **network** which allows **computers** to exchange data. In **computer networks**, networked computing devices passdata to each other along **network** links (data connections). Data is transferred in the form of packets.



2. **What is Network?**

   A network is a group of two or more computer systems linked together. There are many types of computer networks, including the following:

   - **Local-area networks (LANs):** The computers are geographically close together(that is, in the same building).
   - **Wide-area networks (WANs):** The computers are farther apart and areconnected by telephone lines or radio waves.
   - **Campus-area networks (CANs):** The computers are within a limited geographicarea, such as a campus or military base.
   - **Metropolitan-area networks MANs):** A data network designed for a town orcity.
   - **Home-area networks (HANs):** A network contained within a user's home thatconnects a person's digital devices

3. **What is Middleware?**

   Software that acts as a bridge between an operating system or database and applications, especially on a network

4. **What is VPN?**

   VPN is a network that is constructed by using public wires (usually the Internet) toconnect to a private network, such as a company's internal network. There are a number of systems that enable you to create networks using the Internet as the medium for transporting data.

5. **What is Server?**

A computer or device on a network that manages network resources. There are many different types of servers. For example:

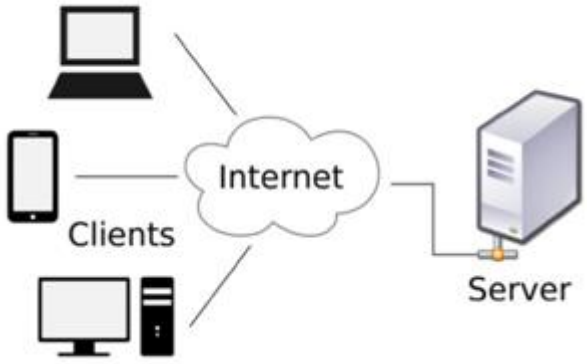
- File server: a computer and storage device dedicated to storing files. Any user on the network can store files on the server.
- Print server: a computer that manages one or more printers, and a network server is a computer that manages network traffic.
- Database server: a computer system that processes database queries.

6. What is Client?

The client part of a *client-server architecture*. Typically, a client is an application that runs on a personal computer or workstation and relies on a server to perform some operations. For example, an *e-mail client* is an application that enables you to send and receive e-mail.

7. What is Client Server Model?

The **client–server model** of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called **clients**.



8. What IP telephony?

IP telephony (Internet Protocol telephony) is a general term for the technologies that use the Internet Protocol's packet-switched connections to exchange voice, fax, and other forms of information that have traditionally been carried over the dedicated circuit-switched connections of the public switched telephone network (PSTN).

9. What are Cookies in Network?

A **cookie** is a mechanism that allows the server to store its own information about a user on the user's own computer. You can view the **cookies** that have been stored on your hard disk (although the content stored in each **cookie** may not make much sense to you). The location of the **cookies** depends on the browser.

10. Define Broadcast, Multicast, Unicast?

- **Unicast:** A term used in communication to describe a piece of information tosend from one point to another. There are only sender and receiver. All LANs support unicast transfer mode and most applications that employ TCP transport protocol uses unicast messaging.

- **Broadcast:** A term used for describing communication that is sent a piece ofinformation from one point to all other points. There is one sender and multiple receivers. All LANs support broadcast transmission.
- **Multicast:** A term described in communicating a piece of information sent fromone or more points to a set of other points. The senders and receivers are one or more

11. What is resource sharing?

In computing, a **shared resource**, or network **share**, is a computer **resource** made available from one host to other hosts on a computer network.

12. What is Phishing?

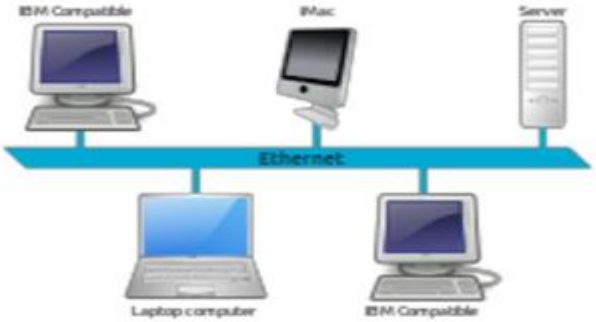
Phishing is an e-mail fraud method in which the perpetrator sends out legitimate-looking email in an attempt to gather personal and financial information from recipients.

13. What is Bluetooth?

Bluetooth is defined as being a *short-range radio technology* (or wireless technology) aimed at simplifying communications among Internet devices and between devices and the Internet. It also aims to simplify data synchronization between Internet devices and other computers.

14. Define LAN, WAN, PAN, and MAN.

LAN: A **local area network** (**LAN**) is a computer network that interconnects computers within a limited area such as a home, school, computer laboratory, or office building, using network media.



WAN: A wide area network (WAN) is a geographically dispersed telecommunications network. The term distinguishes a broader telecommunication structure from a local area network (LAN). A wide area network may be privately owned or rented, but the term usually connotes the inclusion of public (shared user) networks. An intermediate form of network in terms of geography is a metropolitan area network (MAN).

PAN: A **personal area network** (**PAN**) is a computer **network** used for data transmission among devices such as computers, telephones and **personal** digital assistants.

MAN: A **metropolitan area network** (**MAN**) is a **network** that interconnects users with computer resources in a geographic **area** or region larger than that covered by even a large

local **area network** (LAN) but smaller than the **area** covered by a wide **areanetwork** (WAN).

15. What is enterprise Networks

An **enterprise** private **network** is a computer **network** built by a business to interconnect its various company sites (such as production sites, offices and shops) in order to share computer resources

16. What is Access Point?

In computer networking, a wireless **access point** (AP) is a device that allows wireless devices to connect to a wired network using Wi-Fi, or related standards. The AP usually connects to a router (via a wired network) as a standalone device, but it can also be an integral component of the router itself.

17. Define Ethernet?

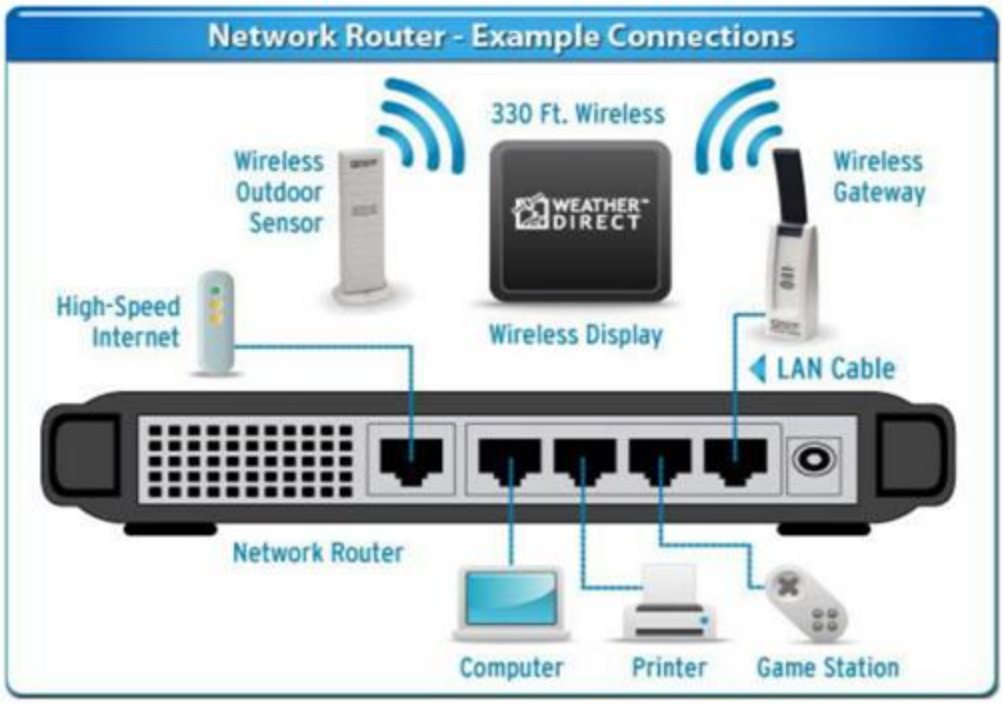
a system for connecting a number of computer systems to form a local area network, with protocols to control the passing of information and to avoid simultaneous transmission by two or more systems.

18. Define Switch, Router?

Switch: **Switch. In networks**, a device that filters and forwards packets between LAN segments. **Switches** operate at the data link layer (layer 2) and sometimes the **network** layer (layer 3) of the OSI Reference Model and therefore support any packet protocol.

Switch

Router: A **router** is a device that forwards data packets along **networks**. A **router** is connected to at least two **networks**, commonly two LANs or WANs or a LAN and its ISP's **network**. **Routers** are located at gateways, the places where two or more **networks** connect.



19. Define Internet or internetwork?

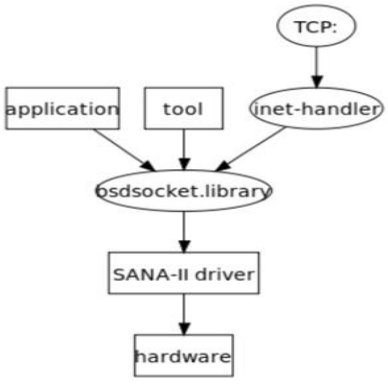
Internetworking is the practice of connecting a computer network with other networks through the use of gateways that provide a common method of routing information packets between the networks. The resulting system of interconnected networks is called an **internetwork**, or simply an internet.

20. Define Network Architecture?

Network architecture is the design of a communications **network**. It is a framework forthe specification of a **network's** physical components and their functional organization and configuration, its operational principles and procedures, as well as data formats used in its operation.

21. Define Protocol Stack?

The **protocol stack** is an implementation of a computer networking **protocol** suite. The terms are often used interchangeably. Strictly speaking, the suite is the **definition** of the **protocols**, and the **stack** is the software implementation of them.



22. Define Error Detection and Error Correction

Error detection: **Error detection** is the **detection** of **errors** caused by noise or otherimpairments during transmission from the transmitter to the receiver. **Error** correction is the **detection** of **errors** and reconstruction of the original, **error**-free data.

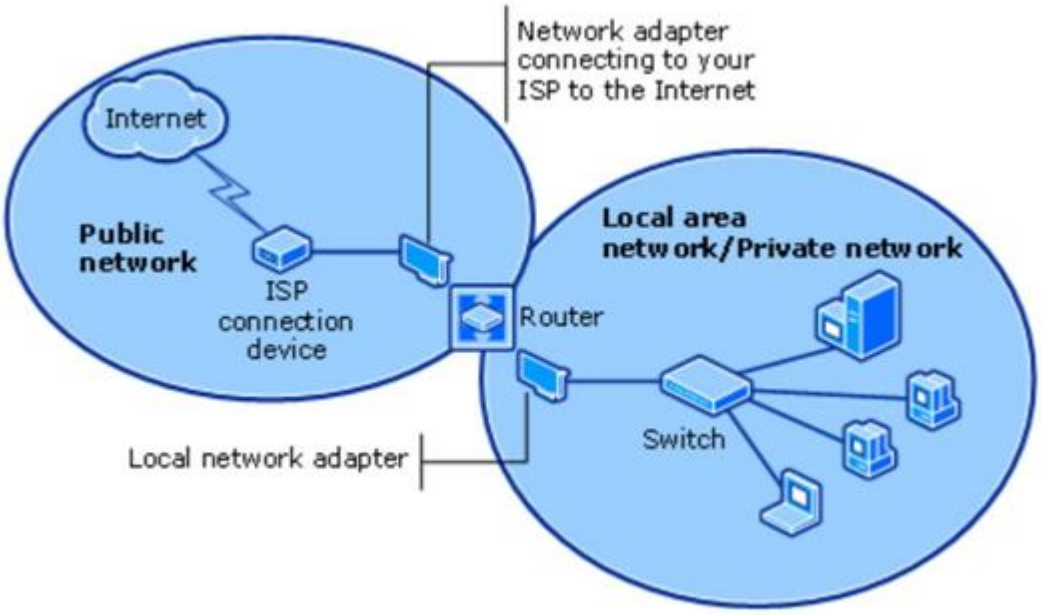
Error correction: Error correction is the process of detecting errors in transmitted messages and reconstructing the original error-free data. Error correction ensures that corrected and error-free messages are obtained at the receiver side.

23. Define Routing?

In internetworking, the process of moving a packet of data from source to destination. Routing is usually performed by a dedicated device called a router. Routing is a key feature of the Internet because it enables messages to pass from one computer to another and eventually reach the target machine. Each intermediary computer performs routing by passing along the message to the next computer. Part of this process involves analyzing a *routing table* to determine the best path.

24. Define Addressing or Naming?

An IP **address** is a unique numerical value that is used to identify a computer on a **network**. There are two kinds of IP **addresses**, public (also called globally unique IP **addresses**) and private. Public IP **addresses** are assigned by the Internet AssignedNumbers Authority (IANA)



25. What is statistical multiplexing?

Statistical multiplexing is a type of communication link sharing, very similar todynamic bandwidth allocation (DBA). In **statistical multiplexing**, a communication channel is divided into an arbitrary number of variable bit-rate digital channels or data streams.

26. What is Flow Control and Error Control?

- **Flow control** is the management of data **flow** between computers or devices orbetween nodes in a network so that the data can be handled at an efficient pace.

Too much data arriving before a device can handle it causes data overflow, meaning the data is either lost or must be retransmitted.

- In information theory and coding theory with applications in computer science and telecommunication, **error** detection and correction or **error control** are techniques that enable reliable delivery of digital data over unreliable communication channels.

27. Define Congestion

In data **networking** and queueing theory, **network congestion** occurs when a link or node is carrying so much data that its quality of service deteriorates. Typical effects include queueing delay, packet loss or the blocking of new connections.

28. Define Packet

A piece of a message transmitted over a **packet**-switching **network**. See under **packet** switching. One of the key features of a **packet** is that it contains the destination address in addition to the data. In IP **networks**, **packets** are often called datagrams

29. What is the difference between store-and-forward switching and cut-through switching?

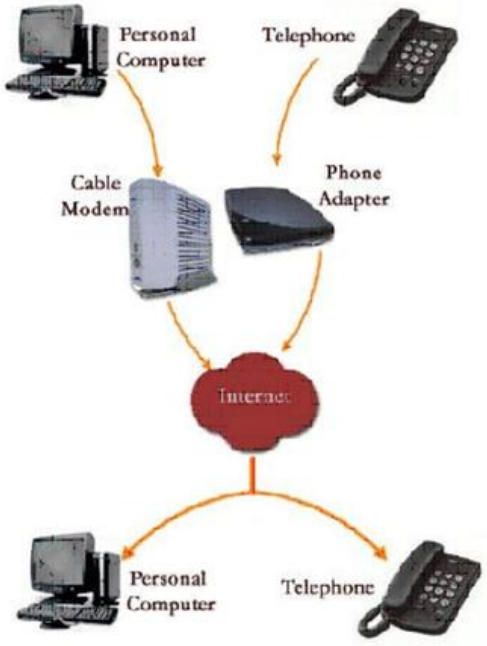
- **Store-and-Forward**: Store-and-Forward switchingwill wait until the entire framehas arrived prior to forwardingit. This method stores the entire frame in memory. Once the frameis in memory, the switch checks the destination address, sourceaddress, and the CRC. If no errors are present, the frame isforwarded to the appropriate port. This process ensures that thedestination network is not affected by corrupted or truncatedframes.
- **Cut-Through**: Cut-Through switching will beginforwarding the frame as soon asthe destination address isidentified. The difference between this and Store-and-Forward isthat Store-and-Forward receives the whole frame before forwarding.Since frame errors cannot be detected by reading only thedestination address, Cut-Through may impact network performance byforwarding corrupted or truncated frames. These bad frames cancreate broadcast storms wherein several devices on the networkrespond to the corrupted frames simultaneously.

30. Define Datagram.

A **datagram** is a basic transfer unit associated with a packet-switched network. The delivery, arrival time, and order of arrival need not be guaranteed by the network.

31. Define Internet Protocol (IP).

The **Internet Protocol** (**IP**) is the principal communications **protocol** in the **Internetprotocol** suite for relaying datagrams across network boundaries. Its routing functionenables internetworking, and essentially establishes the **Internet**.



32. What is ICMP (Internet Control Message Protocol)?

The Internet Control Message Protocol (**ICMP**) is one of the main protocols of the Internet Protocol Suite. It is used by network devices, like routers, to send error messages indicating, for example, that a requested service is not available or that a host or router could not be reached.

33. What is IMPs (Interface Message Processors?)

The **Interface Message Processor** (**IMP**) was the packet-switching node used to interconnect participant **networks** to the ARPANET from the late 1960s to 1989. It was the first generation of gateways, which are known today as routers.

34. Define Network Socket.

A **network socket** is an endpoint of an inter-process communication across a computer **network**. Today, most communication between computers is based on the InternetProtocol; therefore most **network sockets** are Internet **sockets**.

35. Define Sockets

A **socket** is one endpoint of a two-way communication link between two programs running on the network. A **socket** is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to. An endpoint is a combination of an IP address and a port number.

36. What is DSLAM (Digital Subscriber Line Access Multiplexer)?

A DSLAM (Digital Subscriber Line Access Multiplexer) is a network device, usually at a telephone company central office, that receives signals from multiple customer Digital Subscriber Line (DSL) connections and puts the signals on a high-speed backbone line using multiplexing techniques.

37. What is CMTS (Cable Modem Termination System)?

A **cable modem termination system** or **CMTS** is a piece of equipment, typically located in a **cable** company's headend or hubsite, which is used to provide high speed data services, such as **cable** Internet or voice over Internet Protocol, to **cable** subscribers.

38. Define Cable Modem.

A modem designed to operate over cable TV lines. Because the coaxial cable used by cable TV provides much greater bandwidth than telephone lines, a cable modem can be used to achieve extremely fast access to the World Wide Web. This, combined with the fact that millions of homes are already wired for cable TV, has made the cable modem something of a holy grail for Internet and cable TV companies.

39. Define multipath Fading.

Multipath signals are received in a terrestrial environment, i.e., where different forms ofpropagation are present and the signals arrive at the receiver from transmitter via a variety of paths. Therefore there would be **multipath** interference, causing **multi- pathfading**.

40. What is RFCs (Request For Comments)

A Request for Comments (**RFC**) is a publication of the Internet Engineering Task Force (IETF) and the Internet Society, the principal technical development and standards-setting bodies for the Internet.

41. Define Bandwidth.

In computing, **bandwidth** is the bit-rate of available or consumed information capacity expressed typically in metric multiples of bits per second. Variously, **bandwidth** may be characterized as network **bandwidth**, data **bandwidth**, or digital **bandwidth**.

Half-Duplex: Refers to the transmission of data in just one direction at a time. For example, a walkie-talkie is a half-duplex device because only one party can talk at a time. In contrast, a telephone is a *full-duplex* device because both parties can talk simultaneously. Duplex modes often are used in reference to network data transmissions.

43. Define attenuation.

Attenuation is a general term that refers to any reduction in the strength of a signal. **Attenuation** occurs with any type of signal, whether digital or analog. Sometimes calledloss, **attenuation** is a natural consequence of signal transmission over long distances.

44. Define Multiplexing and De-Multiplexing.

Multiplexing: In telecommunications and computer networks, multiplexing (sometimes contracted to muxing) is a method by which multiple analog message signals or digital data streams are combined into one signal over a shared medium. The aim is to share an expensive resource.

De-Multiplexing: Demultiplex (DEMUX) is the reverse of the multiplex (MUX) process – combining multiple unrelated analog or digital signal streams into one signal over a single shared medium, such as a single conductor of copper wire or fiber optic cable. Thus, demultiplex is reconverting a signal containing multiple analog or digital signal streams back into the original separate and unrelated signals.

45. Give any two design issues that occur in computer networking.

a. Connectivity
b. Cost-effective Resource Sharing
c. Support for common Services
d. Performance

46. What is meant by Network?

A network is two or more nodes connected by a direct link, or two or more networks connected by one or more nodes".

47. Give any two advantages of packet switching?

- They can provide variable data rates
- It is better for "bursty" traffic

48. What are the two categories of physical media? Give an example for each category?(April/May 2012)

a. a. Guided Media
   i. . Twisted –Pair cable
      1. Shielded TP
      2. Unshielded TP
   ii. Coaxial Cable
   iii. Fiber-optic cable

b. Unguided Media
   i. Terrestrial microwave
   ii. Satellite Communication

49. Write any four reasons for using layered protocols?

Design: In a layered model each layer is defined separately. Thus, the design problem isbroken up into smaller and manageable pieces. Another advantage is it makes protocol designers to specialize in one area (or layer).

Change: When changes are made to one layer, it reduces the impact on the other layers.For example, protocol in one layer can be changed easily without if affecting higher or lower layers. If the models was not layered and consisted of a single layer then any change affects the entire model.

Learning: The layered approach divided a big more complex tack into several smallertasks where each small task is performed by one layer. This makes it much easier to learn and understand the concept of each layer and the model.

Communication: The layered approach is useful for proper organizing and handling ofcommunication. It also provides a standard programming interface between two layers.

Standards: It is the most important reason for using a layered model. A layered modelprovides a guideline and framework not a rigid standard to be used by the various vendors when creating their products. This is important for interoperability between the various vendors products hat perform different data communication tasks.

50. What are the various types of network topology?

The most common nodes are computers and peripheral devices. Network topology is illustrated by showing these nodes and their connections using cables. There are a number of different types of network topologies, including **point-to-point, bus, star,ring, mesh, tree and hybrid.**

51. Classify the network access?

Access networks can be loosely divided into three categories:

- **Residential access networks**, connecting a home end system into the network;
- **Institutional access networks**, connecting an end system in a business or educationalinstitution into the network;

- **Mobile access networks**, connecting a mobile end system into the network

52. What is transmission delay?

Transmission Delay: Packet size / link bandwidth

Transmission delay is a function of the packet's length and has nothing to do with the distance between the two nodes. This delay is proportional to the packet's length in bits, It is given by the following formula:

$$D_T = N/R_{\text{Seconds}}$$

Where

D_T is the transmission delay in seconds

N is the number of bits, and

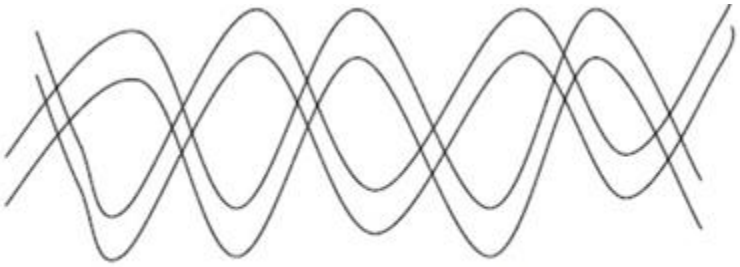
R is the rate of transmission (say in bits per second)

53. What are the various types of services that the internet provides to its applications?

The two primary types of services which are made available by a particular network layer and which actually are also useful classifications for many non-technical types of service industries are known as **connection-oriented** and **connectionless** communication.

54. What is twisted pair?

It has two cables one to carry signal and the other for ground reference the difference of the two levels is taken by the receiving node. Cross talk and noise will be added to the two cables by the noise source. Noise is equal on both cables if they are twisted as the distance is same to both from the source of noise and unequal if parallel. So cables are twisted to avoid interference from outside.



Twisted pair cable

PART – B

11 Marks

1. What are the uses of computer networks? Explain with respective of real world applications.

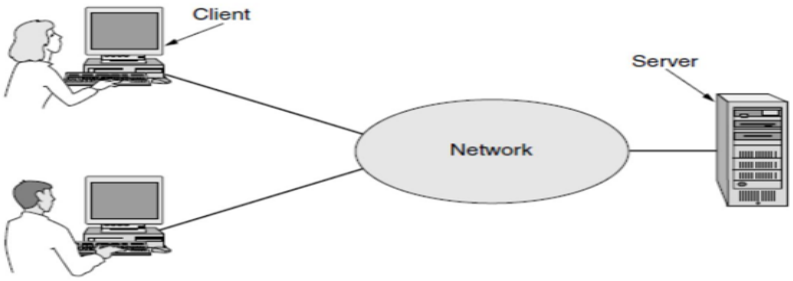
- **Business Applications**
- **Home Applications**
- **Mobile Users**
- **Social Issues**

Business Applications

Most companies have a substantial number of computers. For example, a company may have a computer for each worker and use them to design products, write brochures, and do the payroll. Initially, some of these computers may have worked in isolation from the others, but at some point, management may have decided to connect them to be able to distribute information

throughout the company. Put in slightly more general form, the issue here is **resource sharing**. The goal is to make all programs, equipment, and especially data available to anyone on the network without regard to the physical location of the resource or the user. An obvious and widespread example is having a group of office workers share a common printer. None of the individuals really needs a private printer, and a high-volume networked printer is often cheaper, faster, and easier to maintain than a large collection of individual printers. However, probably even more important than sharing physical resources such as printers, and tape backup systems, is sharing information. Companies small and large are vitally dependent on computerized information. Most companies have customer records, product information, inventories, financial statements, tax information, and much more online. If all of its computers suddenly went down, a bank could not last more than five minutes. A modern manufacturing plant, with a computer-controlled assembly line, would not last even 5 seconds. Even a small travel agency or three-person law firm is now highly dependent on computer networks for allowing employees to access relevant information and documents instantly.

- Networks called **VPNs** (**Virtual Private Networks**) may be used to join the individual networks at different sites into one extended network. In other words, the mere fact that a user happens to be 15,000 km away from his data should not prevent him from using the data as though they were local.
- The data are stored on powerful computers called **servers**. Often these are centrally housed and maintained by a system administrator. In contrast, the employees have simpler machines, called **clients**, on their desks, with which they access remote data, for example, to include in spreadsheets they are constructing.



A network with two clients and one server.

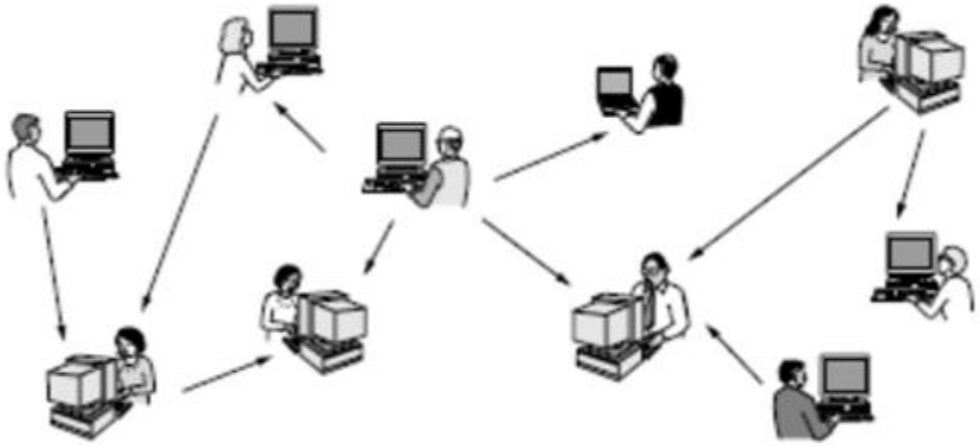
- This whole arrangement is called the **client-server model**. It is widely used and forms the basis of much network usage. The most popular realization is that of a **Webapplication**, in which the server generates Web pages based on its database in responseto client requests that may update the database.
- The client-server model is applicable when the client and server are both in the same building (and belong to the same company), but also when they are far apart.

The client-server model involves requests and replies.

- Telephone calls between employees may be carried by the computer network instead of by the phone company. This technology is called **IP telephony** or **Voice over IP** (**VoIP**) when Internet technology is used.
- A third goal for many companies is doing business electronically, especially with customers and suppliers. This new model is called **e-commerce** (**electronic commerce**) and it has grown rapidly in recent years. Airlines, bookstores, and other retailers have discovered that many customers like the convenience of shop-ping from home. Consequently, many companies provide catalogs of their goods and services online and take orders online. Manufacturers of automobiles, aircraft, and computers, among others, buy subsystems from a variety of suppliers and then assemble the parts. Using computer networks, manufacturers can place orders electronically as needed. This reduces the need for large inventories and enhances efficiency.

Home Applications

- The biggest reason to buy a home computer was probably for Internet access. Now, many consumer electronic devices, such as set-top boxes, game consoles, and clock radios, come with embedded computers and computer networks, especially wireless networks, and home networks are broadly used for entertainment, including listening to, looking at, and creating music, photos, and videos.
- Internet access provides home users with **connectivity** to remote computers. As with companies, home users can access information, communicate with other people, and buy products and services with e-commerce. The main benefit now comes from connecting outside of the home.
- Bob Metcalfe, the inventor of Ethernet, hypothesized that the value of a network is proportional to the square of the number of users because this is roughly the number of different connections that may be made (Gilder, 1993). This hypothesis is known as „„„Metcalfe‟s law.‟‟‟‟



In a peer-to-peer system there are no fixed clients and servers

- Peer-to-peer communication is often used to share music and videos.
- The Internet can be used by applications to carry audio (e.g., Internet radio stations) and video (e.g., YouTube).
- Between person-to-person communications and accessing information are **socialnetwork** applications. Here, the flow of information is driven by the relationships thatpeople declare between each other.
- One of the most popular social networking sites is **Facebook**. It lets people update their personal profiles and shares the updates with other people who they have declared to be their friends. Other social networking applications can make introductions via friends of friends, send news messages to friends such as Twitter above, and much more.
- Even more loosely, groups of people can work together to create content. A **wiki**, for example, is a collaborative Web site that the members of a community edit. The most famous wiki is the **Wikipedia**, an encyclopedia anyone can edit, but there are thousands of other wikis.

Tag	Full name	Example
B2C	Business-to-consumer	Ordering books online
B2B	Business-to-business	Car manufacturer ordering tires from supplier
G2C	Government-to-consumer	Government distributing tax forms electronically
C2C	Consumer-to-consumer	Auctioning second-hand products online
P2P	Peer-to-peer	Music sharing

Some forms of e-commerce.

- **Ubiquitous computing**, in which computing is embedded into everyday life, as in thevision of Mark Weiser (1991). Many homes are already wired with security systems that include door and window sensors, and there are many more sensors that can be folded in to a smart home monitor, such as energy consumption.
- A technology called **RFID** (**Radio Frequency IDentification**) will push this idea even further in the future. RFID tags are passive (i.e., have no battery) chips the size of stamps and they can already be affixed to books, passports, pets, credit cards, and other items in

the home and out. This lets RFID readers locate and Communicate with the items over a distance of up to several meters, depending on the kind of RFID. Originally, RFID was commercialized to replace barcodes. It has not succeeded yet because barcodes are free and RFID tags cost a few cents. Of course, RFID tags offer much more and their price is rapidly declining. They may turn the real world into the Internet of things (ITU, 2005).

Mobile Users

Mobile computers, such as laptop and handheld computers, are one of the fastest-growing segments of the computer industry. Their sales have already overtaken those of desktop computers. Why would anyone want one? People on the go often want to use their mobile devices to read and send email, tweet, watch movies, download music, play games, or simply to surf the Web for information. They want to do all of the things they do at home and in the office. Naturally, they want to do them from anywhere on land, sea or in the air.

Connectivity to the Internet enables many of these mobile uses. Since having a wiredconnection is impossible in cars, boats, and airplanes, there is a lot of interest in wireless networks. Cellular networks operated by the telephone companies are one familiar kind of wireless network that blankets us with coverage for mobile phones. Wireless **hotspots** based on the 802.11 standard are another kind of wireless network for mobile computers. They have sprung up everywhere that people go, resulting in a patchwork of coverage at cafes, hotels, airports, schools, trains and planes. Anyone with a laptop computer and a wireless modem can just turn on their computer on and be connected to the Internet through the hotspot, as though the computer were plugged into a wired network.

Wireless	Mobile	Typical applications
No	No	Desktop computers in offices
No	Yes	A notebook computer used in a hotel room
Yes	No	Networks in unwired buildings
Yes	Yes	Store inventory with a handheld computer

Combinations of wireless networks and mobile computing.

- **Text messaging** or **texting** is tremendously popular. It lets a mobile phone user type ashort message that is then delivered by the cellular network to another mobile subscriber.
- **Smart phones**, such as the popular iPhone, combine aspects of mobile phones andmobile computers. The (3G and 4G) cellular networks to which they connect can provide fast data services for using the Internet as well as handling phone calls.
- Since mobile phones know their locations, often because they are equipped with **GPS** (**Global Positioning System**) receivers, some services are intentionally location dependent.
- When equipped with **NFC** (**Near Field Communication**) technology the mobile can act as an RFID smartcard and interact with a nearby reader for payment.

Social Issues

- Network neutrality

- DMCA takedown notices
- Gmail.

2. Explain Network Hardware. (Or)Classify multiple processor systems by their rough physical size and Explain with neat Diagram. (Or) Explain the different types of Networks with neat diagram.

Broadly speaking, there are two types of transmission technology that are in widespread use. They are as follows:

1. Broadcast links.
2. Multicast links

2. Point-to-point links.

Broadcast networks have a single communication channel that is shared by all the machines on the network. Short messages, called **packets** in certain contexts, sent by any machine are received by all the others. An **address field** within the packet specifies the intended recipient. Upon receiving a packet, a machine checks the address field. If the packet is intended for the receiving machine, that machine processes the packet; if the packet is intended for some other machine, it is just ignored.

Broadcast systems generally also allow the possibility of addressing a packet to all destinations by using a **special code in the address field**. When a packet with this code is transmitted, it is received and processed by every machine on the network. This mode of operation is called **broadcasting.**

Some broadcast systems also support transmission to a subset of the machines, something known as **multicasting.** One possible scheme is to reserve one bit to indicate multicasting. The remaining n - 1 address bits can hold a group number. Each machine can "subscribe" to any or all of the groups. When a packet is sent to a certain group, it is delivered to all machines **subscribing to that group.**

In contrast,**point-to-point networks** consist of many connections between **individual pairs of machines.** To go from the source to the destination, a packet on this type of network may have to first visit one or more intermediate machines. Often multiple routes, of different lengths, are possible, so **finding good ones is important in point-to-point networks**. Point-to-point transmission with one sender and one receiver is sometimes called **unicasting.**

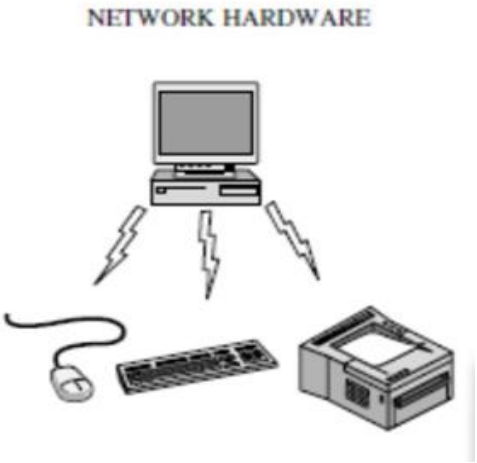
Figure 1-6. Classification of interconnected processors by scale.

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

Personal Area Networks

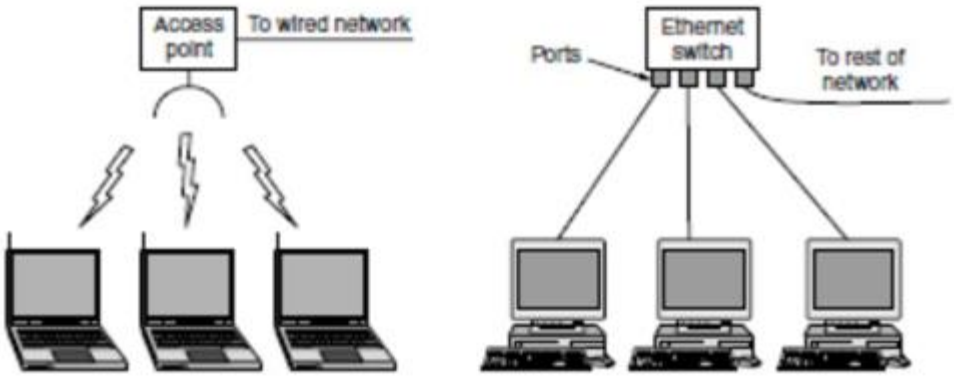
- **PANs** (**Personal Area Networks**) let devices communicate over the range of a person. Acommon example is a wireless network that connects a computer with its peripherals.
- Almost every computer has an attached monitor, keyboard, mouse, and printer. Without using wireless, this connection must be done with cables.

- A short-range wireless network called **Bluetooth** to connect these components without wires.



Local Area Networks

- **LAN** (**Local Area Network**).
- A LAN is a privately owned network that operates within and nearby a single building like a home, office or factory.
- LANs are widely used to connect personal computers and consumer electronics to let them share resources (e.g., printers) and exchange information. When LANs are used by companies, they are called **enterprise networks**.
- In most cases, each computer talks to a device in the ceiling as shown in Fig. (a). This device, called an **AP** (**Access Point**), **wireless router**, or **base station**, relays packets between the wireless computers and also between them and the Internet.
- There is a standard for wireless LANs called **IEEE 802.11**, popularly known as **WiFi**, which has become very widespread. It runs at speeds anywhere from 11 to hundreds of Mbps.



- Wireless and wired LANs (a) 802.11 (b) Switched Ethernet. The topology of many wired LANs is built from point-to-point links. IEEE 802.3, popularly called **Ethernet**, is, by far, the most common type of wired LAN. Fig. (b) shows a sample topology of **switched Ethernet**.

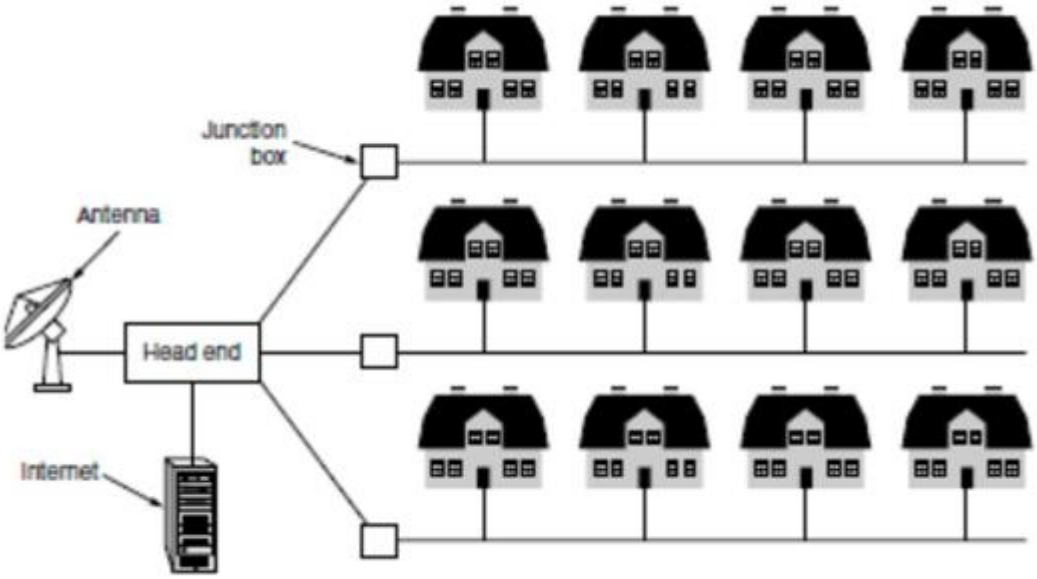
- Each computer speaks the Ethernet protocol and connects to a box called a **switch** with a point-to-point link. Hence the name. A switch has multiple **ports**, each of which can connect to one computer. The job of the switch is to relay packets between computers that are attached to it, using the address in each packet to determine which computer to send it to.
- The difficulty is how to carry both power and data signals at the same time. Part of the answer is that they use different frequency bands. In short, home LANs offer many opportunities and challenges. Most of the latter relate to the need for the networks to be easy to manage, dependable, and secure, especially in the hands of nontechnical users, as well as low cost.

Metropolitan Area Networks

- A **MAN** (**Metropolitan Area Network**) covers a city.
- The best-known examples of MANs are the cable television networks available in many cities.
- A MAN might look something like the system shown in Fig. In this figure we see both television signals and Internet being fed into the centralized **cable headend** for subsequent distribution to people‟s homes.

Wide Area Networks

- A **wide area network,** or **WAN,** spans a large geographical area, often a country or continent. It contains of machines intended for running user (i.e., application) programs. We will follow traditional usage and call these machines **hosts.** The term **end system** is sometimes also used in the literature. The hosts are connected by a **communicationsubnet,** or just **subnet** for short. The job of the subnet is to carry messages from host tohost, just as the telephone system carries words from speaker to listener. By separating the pure communication aspects of the network (the subnet) from the application aspects (the hosts), the complete network design is greatly simplified.
- In most wide area networks, the subnet consists of two distinct components: **transmission lines** & **switching elements.** Transmission lines (also called **circuits, channels,** or **trunks**) move bits between machines.
- The switching elements are specialized computers used to connect two or more transmission lines. When data arrive on an incoming line, the switching element must choose an outgoing line to forward them on.
- In most WANs, the network contains numerous cables or telephone lines, each one connecting a pair of routers. If two routers that do not share a cable nevertheless wish to communicate, they must do this indirectly, via other routers. When a packet is send from one router to another via one or more intermediate routers, the packet is received at each intermediate router in its entirety, stored there until the required output line is free, and the forwarded. A subnet using this principle is called a **Point to Point**, **store andforward** or **packet switched** subnet. Nearly all wide area networks (except those usingsatellite) have store and forward subnet. When the packet are small and all the same size, they often called **cells**.



A metropolitan area network based on cable TV

Internetworks

- Many networks exist in the world, often with different hardware and software.
- People connected to one network often want to communicate with people attached to a different one.
- The fulfillment of this desire requires that different, and frequently incompatible, networks be connected.
- A collection of interconnected net works is called an **internetwork** or **internet**. These terms will be used in a generic sense, in contrast to the worldwide Internet (which is one specific internet), which we will always capitalize. The Internet uses ISP networks to connect enterprise networks, home networks, and many other networksSubnets, networks, and internetworks are often confused. The term „„subnet"" makes the most sense in the context of a wide area network, where it refers to the collection of routers and communication lines owned by the network operator.
- The general name for a machine that makes a connection between two or more networks and provides the necessary translation, both in terms of hardware and software, is a **gateway**.
- Gateways are distinguished by the layer at which they operate in the protocol hierarchy. We will have much more to say about layers and protocol hierarchies starting in the next section, but for now imagine that higher layers are more tied to applications, such as the Web, and lower layers are more tied to transmission links, such as Ethernet.

3. Explain about Network Software.

a. Protocol Hierarchies.

b. Design Issues for the layers

c. Connection oriented and Connectionless Services.

d. Service Primitives

a. Protocol Hierarchies. (Explain about Protocol Hierarchies with Necessary Diagram.)

- To reduce their design complexity, most networks are organized as a stack of **layers** or **levels**, each one built upon the one below it. The number of layers, the name of eachlayer, the contents of each layer, and the function of each layer differ from network to network.
- The purpose of each layer is to offer certain services to the higher layers while shielding those layers from the details of how the Offered services are actually implemented. In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it.
- This concept is actually a familiar one and is used throughout computer science, where it is variously known as information hiding, abstract data types, data encapsulation, and object-oriented programming. The fundamental idea is that a particular piece of software (or hardware) provides a service to its users but keeps the details of its internal state and algorithms hidden from them.

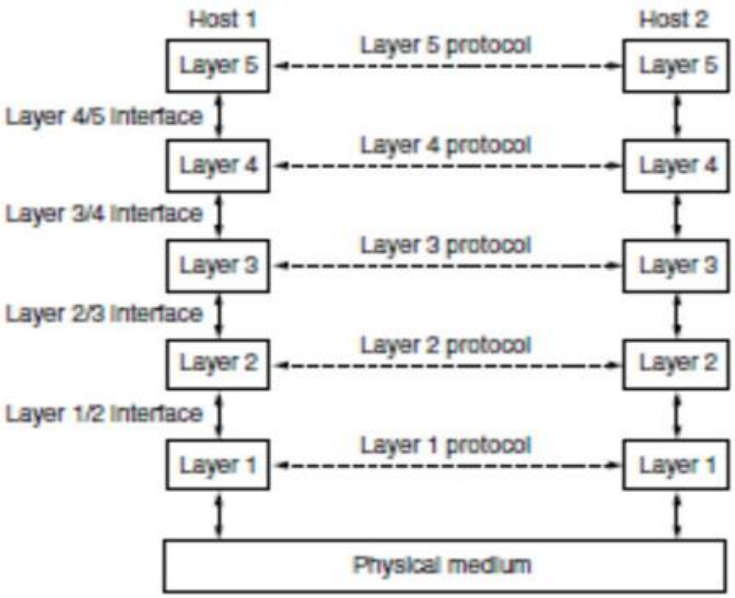


Figure. Layers, protocols, and interfaces

A five-layer network is illustrated in Fig. The entities comprising the corresponding layers on different machines are called **peers.** In reality, no data are directly transferred from layer *n* on one machine to layer *n* on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached.

- Above layer 1 is the **physical medium** through which actual communication occurs. In Fig, virtual communication is shown by dotted lines and physical communication by solid lines. Between each pair of adjacent layers is an **interface**.
- The interface defines which primitive operations and services the lower layer makes available to the upper one. When network designers decide how many layers to include in a network and what each one should do, one of the most important considerations is defining clean interfaces between the layers. Doing so, in turn, requires that each layer perform a specific collection of well-understood functions. In addition to minimizing the amount of information that must be passed between layers, clear-cut interfaces also make

it simpler to replace one layer with a completely different protocol or implementation (e.g., replacing all the telephone lines by satellite channels) because all that is required of the new protocol or implementation is that it offer exactly the same set of services to its upstairs neighbor as the old one did.

- It is common that different hosts use different implementations of the same protocol (often written by different companies). In fact, the protocol itself can change in some layer without the layers above and below it even noticing.
- A set of layers and protocols is called network **architecture**. The specification of architecture must contain enough information to allow an implementer to write the program or build the hardware for each layer so that it will correctly obey the appropriate protocol. Neither the details of the implementation nor the specification of the interfaces is part of the architecture because these are hidden away inside the machines and not visible from the outside. It is not even necessary that the interfaces on all machines in a network be the same, provided that each machine can correctly use all the protocols. A list of the protocols used by a certain system, one protocol per layer, is called a **protocolstack**.

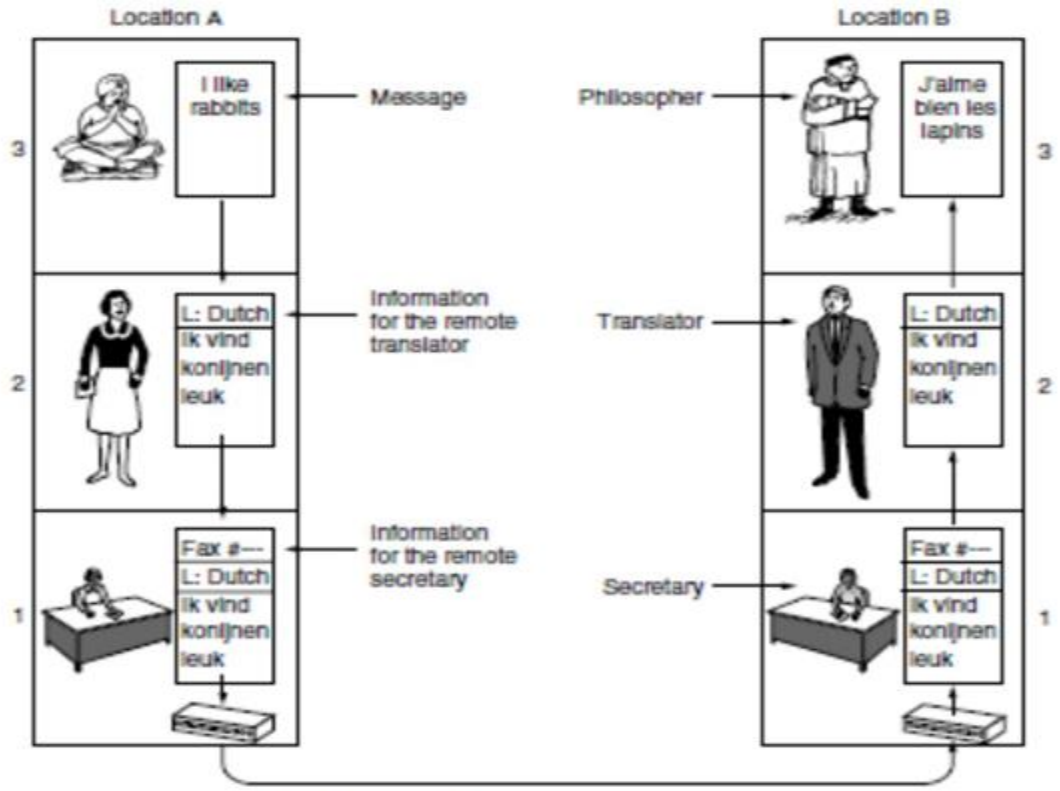


Figure. The philosopher-translator-secretary architecture

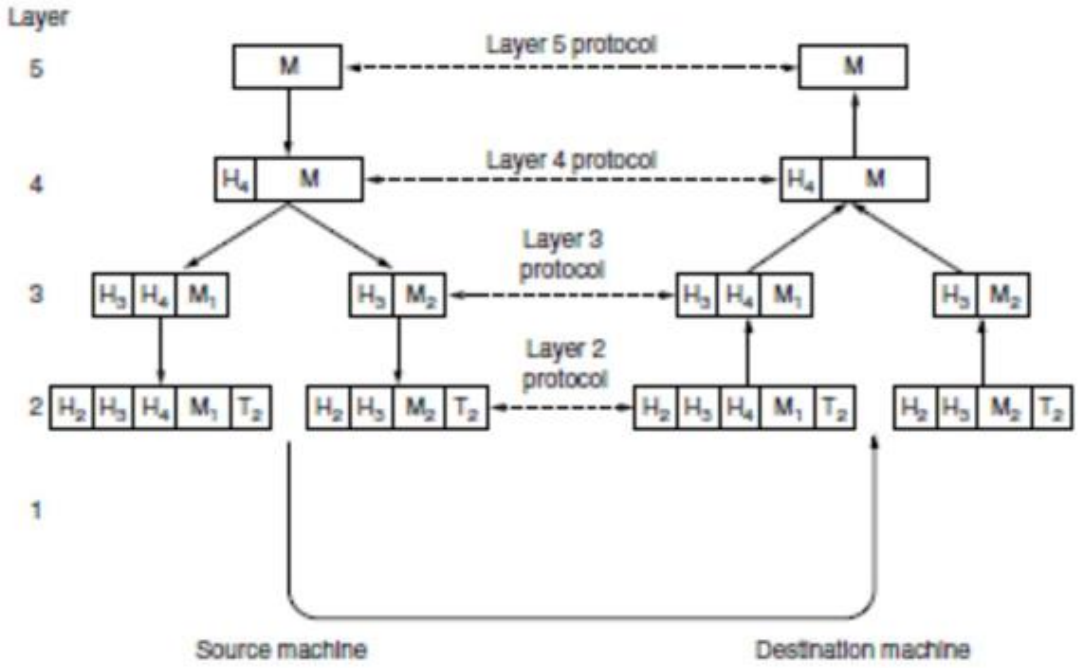


Figure. Example information flow supporting virtual communication in layer 5

The peer process abstraction is crucial to all network design. Using it, the unmanageable task of designing the complete network can be broken into several smaller, manageable design problems, namely, the design of the individual layers.

b.Design Issues for the layers. (Discuss about what are the Design Issues for the Layers.)

Some of the key design issues that occur in computer networks will come up in layer after layer. Below, we will briefly mention the more important ones.

- Reliability is the design issue of making a network that operates correctly even though it is made up of a collection of components that are themselves unreliable. Think about the bits of a packet traveling through the network.
- There is a chance that some of these bits will be received damaged (inverted) due to fluke electrical noise, random wireless signals, hardware flaws, software bugs and so on. How is it possible that we find and fix these errors? One mechanism for finding errors in received information uses codes for **error detection**. Information that is incorrectly received can then be retransmitted until it is received correctly. More powerful codes allow for **error correction**, where the correct message is recovered from the possibly incorrect bits that were originally received. Both of these mechanisms work by adding redundant information. They are used at low layers, to protect packets sent over individual links, and high layers, to check that the right contents were received.
- Another reliability issue is finding a working path through a network. Often there are multiple paths between a source and destination, and in a large network, here may be some links or routers that are broken.
- A second design issue concerns the evolution of the network. Over time, networks grow larger and new designs emerge that need to be connected to the existing network. We have recently seen the key structuring mechanism used to support change by dividing the overall problem and hiding implementation details: **protocol layering**. There are many other

strategies as well. Since there are many computers on the network, every layer needs a mechanism for identifying the senders and receivers that are involved in a particular message. This mechanism is called **addressing** or **naming**, in the low and high layers, respectively. An aspect of growth is that different network technologies often have different limitations.

- Designs that continue to work well when the network gets large are said to be **scalable**.
- A third design issue is resource allocation. Networks provide a service to hosts from their underlying resources, such as the capacity of transmission lines. To do this well, they need mechanisms that divide their resources so that one host does not interfere with another too much. Many designs share network bandwidth dynamically, according to the short-term needs of hosts, rather than by giving each host a fixed fraction of the bandwidth that it may or may not use. This design is called **statistical multiplexing**, meaning sharing based on the statistics of demand. It can be applied at low layers for a single link, or at high layers for a network or even applications that use the network.
    - An allocation problem that occurs at every level is how to keep a fast sender from swamping a slow receiver with data. Feedback from the receiver to the sender is often used. This subject is called **flow control**. Sometimes the problem is that the network is oversubscribed because too many computers want to send too much traffic, and the network cannot deliver it all. This overloading of the network is called **congestion**. One strategy is for each computer to reduce its demand when it experiences congestion. It, too, can be used in all layers. It is interesting to observe that the network has more resources to offer than simply bandwidth. For uses such as carrying live video, the timeliness of delivery matters a great deal. Most networks must

5. provide service to applications that want this **real-time** delivery at the same time that they provide service to applications that want high throughput. **Quality of service** is the name given to mechanisms that reconcile these competing demands. The last major design issue is to secure the network by defending it against different kinds of threats. One of the threats we have mentioned previously is that of eavesdropping on communications. Mechanisms that provide **confidentiality** defend against this threat, and they are used in multiple layers. Mechanisms for **authentication** prevent someone from impersonating someone else.

c. Connection oriented and Connectionless Services (Write short notes on Connection Oriented and Connection less Services. (or) Compare Connection Oriented versus Connection Less Services.

A Service Description

Connectionless and Connection-Oriented Services

The two primary types of services which are made available by a particular network layer and which actually are also useful classifications for many non-technical types of service industries are known as **connection-oriented** and **connectionless** communication.

Connection-oriented Services

One of the easiest ways to understand what a connection-oriented protocol is would be to think of a very familiar service upon which it's based: the telephone system. When I pick up the phone, I have an open circuit, and the dial tone carrier signal allows me to connect to a destination of my choosing.

Given valid input parameters, the service:

- Establishes the connection.
- Allows me to utilize the connection.
- Tears down the connection when I'm done using it

The primary difference between this method and that of a connectionless service is that in a connection-oriented system, all of my communications are taking place on the same transmission channel. On the other hand, with a connectionless service, all transmissions are independently routed, and perhaps re-assembled in some order at the other end -- the service in between has no inherent responsibility for ensuring ordinarily -- it need only assure that each transmission gets delivered from its source to its destination.

Connectionless Services

A good analogy for a connectionless service is the process of sending letters through the postal system. Each transmission (the "letter") contains the full destination address and is processed independent of related messages. As described above, the service has only to ensure that each reaches its host within certain time parameters. Unlike a connection-oriented service, the system has free reign on what happens enroute between the sender and receiver:

- A message can be delayed to ensure another arrives first.
- Widely different channels of communication can be used for transmitting messages.
- A message can be handed off to a trusted third party in the distribution network.
    - A message can be intercepted by a third party, copied or logged, and passed on to the intended receiver.

These operations are basically impossible for a connection-oriented service.

	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Movie download
	Unreliable connection	Voice over IP
Connectionless	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Text messaging
	Request-reply	Database query

Figure. Six different types of service

Types of Services Available in TCP/IP and OSI:

The OSI Reference Model provides for both connection-oriented and connectionless communication at the network level. However, it supports only connection-oriented communication at the Transport layer. It became obvious after the initial design of OSI that allowing both types of traffic at the transport layer was important even though it violated the idea of data abstraction which was central to the design of OSI.

TCP/IP, on the other hand, supports only connectionless traffic at the Network layer, but supports both modes in the Transport layer. This allows for simple request-response protocols to be easily implemented, though it complicates things somewhat for the user. At the Transport level, TCP, the **Transmission Control Protocol** (a connection-oriented service) as well as UDP, **Datagram Protocol** (a connectionless service) are provided.

Quality of Service

Reliability of connections achieved through connectionless and connection-oriented protocols is another major concern. All protocols are ***not*** created equal, and sacrifices in reliability can be made in exchange for greater speed, or vice versa. Often the trade-offs are worth it, assuming that we're attempting to fit our task intelligently into the capacities of the protocol.

Sometimes it's necessary for a "handshake" process to occur, especially if we need to authenticate each piece of traffic from a sender, but in many cases (such as streaming video), the performance hit involved is simply unacceptable. Not all applications require connections, and it's naive to think of either protocol as superior to the other. In certain cases, it's not even necessary to ensure that a message gets sent, so long as the chance that it was received is high enough (think of the high-volume email transmissions of spammers). Consider how difficult and time-consuming it would be if each spam message had to be acknowledged by the receiver and tracked by the spammer.

d. Service Primitives

A service is formally specified by a set of primitives (operations) available to a user process to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity.

If the protocol stack is located in the operating system, as it often is, the primitives are normally system calls. These calls cause a trap to kernel mode, which then turns control of the machine over to the operating system to send the necessary packets.

The set of primitives available depends on the nature of the service being provided. The primitives for connection-oriented service are different from those of connectionless service.

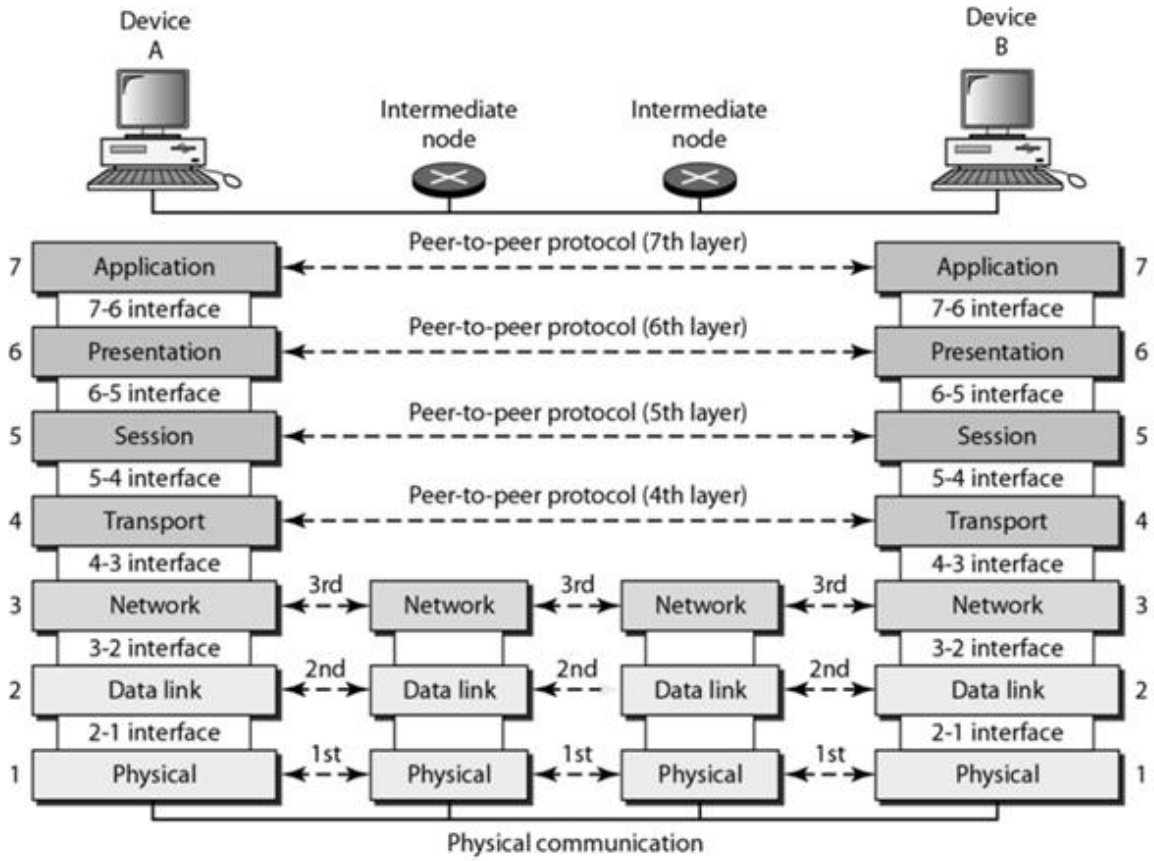
Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

4. Explain reference Models.

- OSI reference Model
- TCP/IP Reference Model
- Comparison of OSI reference model and TCP/IP reference model
- Critique of OSI and TCP/IP reference model

OSI REFERENCE MODEL

(Explain OSI Reference Model in Detail with neat diagram.)

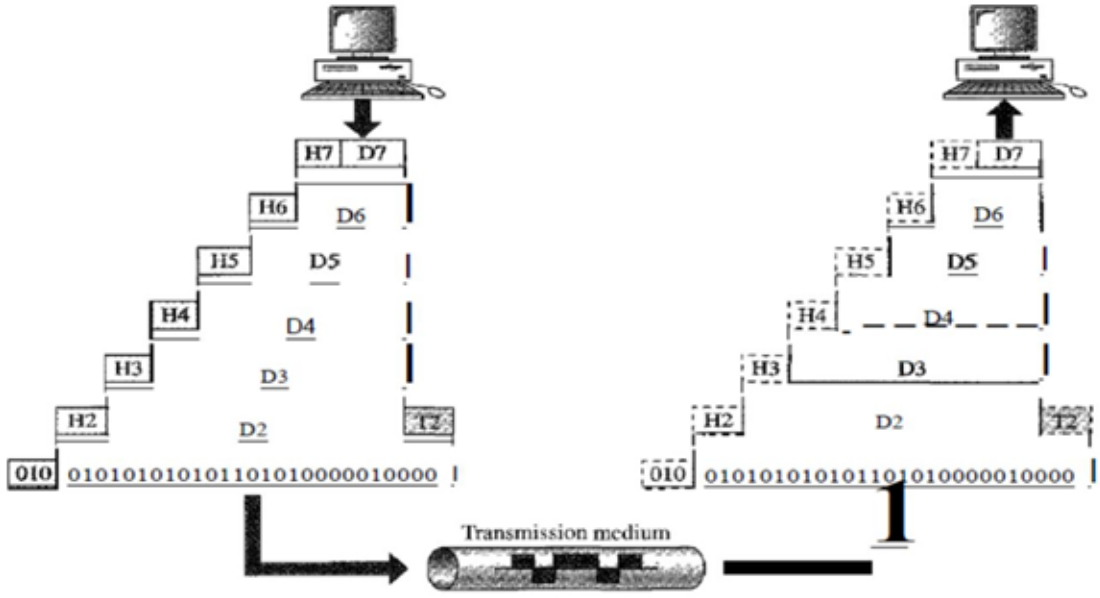


ISO defines a common way to connect computer by the architecture called Open System Interconnection (OSI) architecture. Network functionality is divided into seven layers.

Organization of the layers: The 7 layers can be grouped into 3 subgroups

1. **Network Support Layers**
   Layers 1,2,3 - Physical, Data link and Network are the network support layers. They deal with the physical aspects of moving data from one device to another such as electrical specifications, physical addressing, transport timing and reliability.
2. **Transport Layer**
   Layer4, transport layer, ensures end-to-end reliable data transmission on a single link.
3. **User Support Layers**
   Layers 5,6,7 – Session, presentation and application are the user support layers. They allow interoperability among unrelated software systems

An Data exchange using the OSI model



An exchange using the OSI Model

Functions of the Layers

1. Physical Layer

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

The physical layer is concerned with the following:

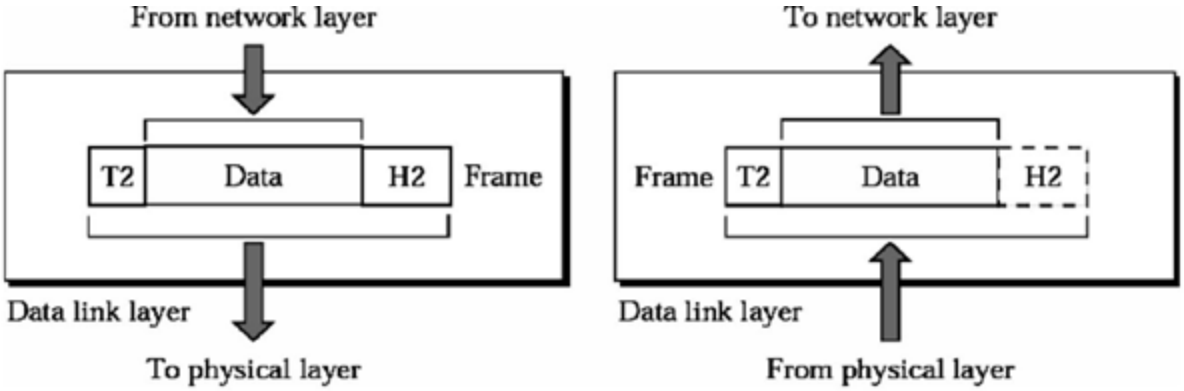
- **Physical characteristics of interfaces and media -** The physical layer defines thecharacteristics of the interface between the devices and the transmission medium.
- **Representation of bits -** To transmit the stream of bits, it must be encoded to signals.The physical layer defines the type of encoding.
- **Data Rate or Transmission rate -** The number of bits sent each second–is also definedby the physical layer.
- **Synchronization of bits -** The sender and receiver must be synchronized at the bit level.Their clocks must be synchronized.
- **Line Configuration -** In a point-to-point configuration, two devices are connectedtogether through a dedicated link. In a multipoint configuration, a link is shared between several devices.

  Physical Topology - The physical topology defines how devices are connected to make anetwork. Devices can be connected using a mesh, bus, star or ring topology.

  Transmission Mode - The physical layer also defines the direction of transmissionbetween two devices: simplex, half-duplex or full-duplex.

2.Data Link Layer

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

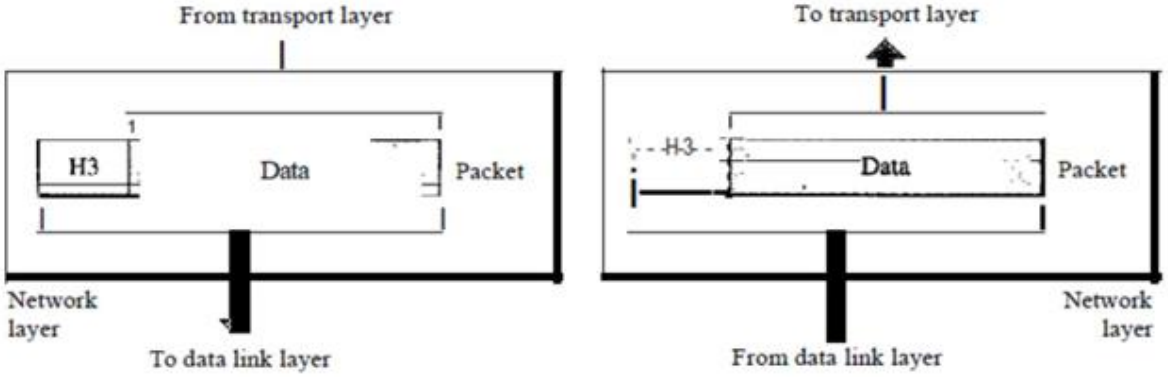


The other responsibilities of this layer are

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

3. NETWORK LAYER

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



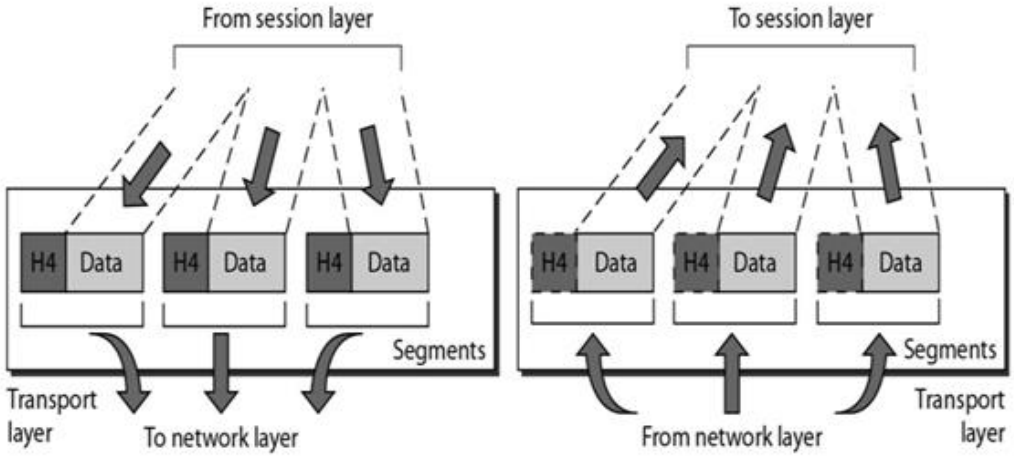
Network Layer

It is mainly required, when it is necessary to send information from one network to another. The other responsibilities of this layer are

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

4. TRANSPORT LAYER

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

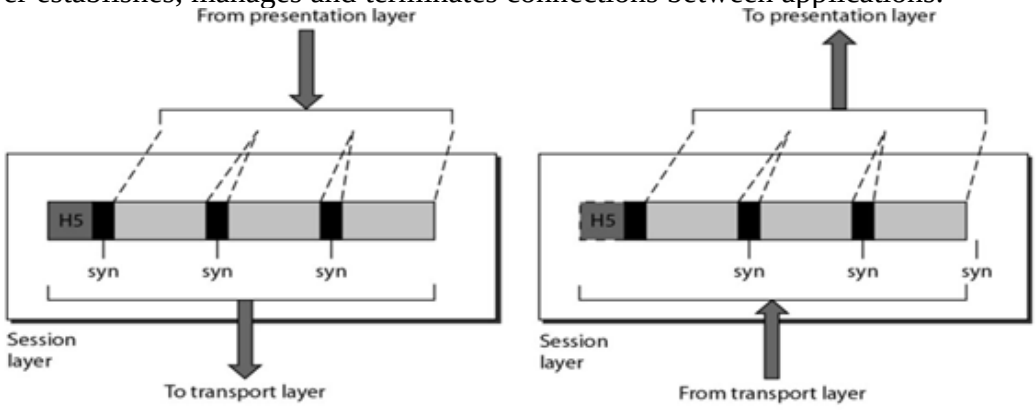


The other responsibilities of this layer are

- **Port addressing** - The header in this must therefore include a address called port address.This layer gets the entire message to the correct process on that computer.
- **Segmentation and reassembly** - The message is divided into segments and eachsegment is assigned a sequence number. These numbers are arranged correctly on the arrival side by this layer.
- **Connection control** - This can either be **connectionless or connection oriented.** Theconnectionless treats each segment as a individual packet and delivers to the destination. The connection-oriented makes connection on the destination side before the delivery. After the delivery the termination will be terminated.
- **Flow and error control** - Similar to data link layer, but process to process take place.

5. SESSION LAYER

This layer establishes, manages and terminates connections between applications.

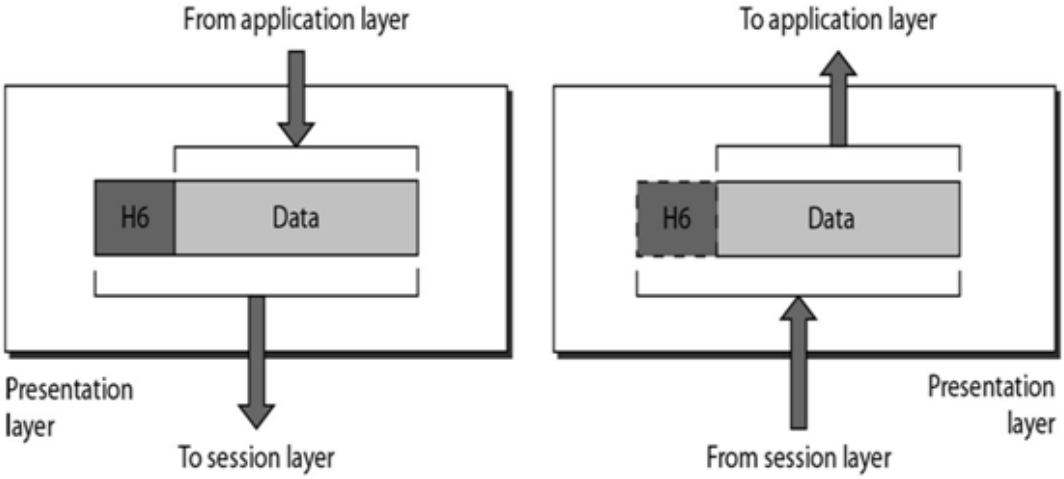


The other responsibilities of this layer are

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

6. PRESENTATION LAYER

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

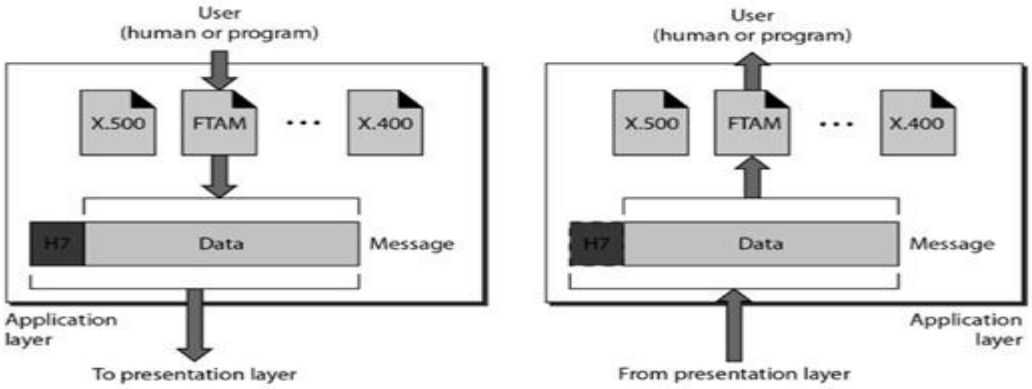


The other responsibilities of this layer are

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

7. APPLICATION LAYER

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



The other responsibilities of this layer are

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

TCP/IP REFERENCE MODEL

(Discuss about TCP/IP Reference Model with neat diagram)

Let us now turn from the OSI reference model to the reference model used in the grandparent of all wide area computer networks, the ARPANET, and its successor, the

worldwide Internet. Although we will give a brief history of the ARPANET later, it is useful to mention a few key aspects of it now.

The ARPANET was a research network sponsored by the DoD (U.S. Department of Defense). It eventually connected hundreds of universities and government installations, using leased telephone lines. When satellite and radio networks were added later, the existing protocols had trouble interworking with them, so a new reference architecture was needed. Thus, from nearly the beginning, the ability to connect multiple networks in a seamless way was one of the major design goals.

This architecture later became known as the **TCP/IP Reference Model**, after its two primary protocols. It was first described by Cerf and Kahn (1974), and later refined and defined as a standard in the Internet community (Braden, 1989). The design philosophy behind the model is discussed by Clark (1988). Given the DoD‟s worry that some of its precious hosts, routers, and internetwork gateways might get blown to pieces at a moment‟s notice by an attack from the Soviet Union, another major goal was that the network be able to survive loss of subnet hardware, without existing conversations being broken off. In other words, the DoD wanted connections to remain intact as long as the source and destination machines were functioning, even if some of the machines or transmission lines in between were suddenly put out of operation. Furthermore, since applications with divergent requirements were envisioned, ranging from transferring files to real-time speech transmission, a flexible architecture was needed.

The Link Layer

All these requirements led to the choice of a packet-switching network based on a connectionless layer that runs across different networks. The lowest layer in the model, the **link layer** describes what links such as serial lines and classic Ethernet must do to meet theneeds of this connectionless internet layer. It is not really a layer at all, in the normal sense of the term, but rather an interface between hosts and transmission links. Early material on the TCP/IP model has little to say about it.

The Internet Layer

The **internet layer** is the linchpin that holds the whole architecture together. It is shown in Fig. 1-21 as corresponding roughly to the OSI network layer. Its job is to permit hosts to inject packets into any network and have them travel independently to the destination (potentially on a different network). They may even arrive in a completely different order than they were sent, in which case it is the job of higher layers to rearrange them, if in-order delivery is desired. Note that „„internet““ is used here in a generic sense, even though this layer is present in the Internet

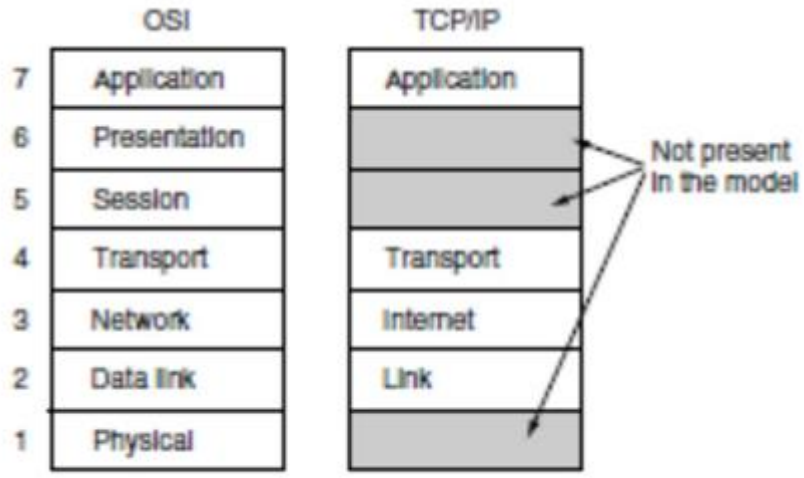


Figure. The TCP/IP reference model

The analogy here is with the (snail) mail system. A person can drop a sequence of international letters into a mailbox in one country, and with a little luck, most of them will be delivered to the correct address in the destination country. The letters will probably travel through one or more international mail gateways along the way, but this is transparent to the users. Furthermore, that each country (i.e., each network) has its own stamps, preferred envelope sizes, and delivery rules is hidden from the users. The internet layer defines an official packet format and protocol called **IP** (**Internet Protocol**), plus a companion protocol called **ICMP** (**Internet Control Message Protocol**) that helps it function. The job of the internet layer is to deliver IP packets where they are supposed to go. Packet routing is clearly a major issue here, as is congestion (though IP has not proven effective at avoiding congestion).

The Transport Layer

The layer above the internet layer in the TCP/IP model is now usually called the **transport layer**. It is designed to allow peer entities on the source and destination hosts to carryon a conversation, just as in the OSI transport layer. Two end-to-end transport protocols have been defined here.

- The first one, **TCP** (**Transmission Control Protocol**), is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet. It segments the incoming byte stream into discrete messages and passes each one on to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle.
- The second protocol in this layer, **UDP** (**User Datagram Protocol**), is unreliable, connectionless protocols for applications that do not want TCP‟s sequencing or flow control and wish to provide their own. It is also widely used for one-shot, client-server-type request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. The relation of IP, TCP, and UDP is shown in Fig. Since the model was developed, IP has been implemented on many other networks.

The Application Layer

The TCP/IP model does not have session or presentation layers. No need for them was perceived. Instead, applications simply include any session and presentation functions that they require. Experience with the OSI model has proven this view correct: these layers are of little use to most applications. On top of the transport layer is the **application layer**. It contains all the higher-level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP).

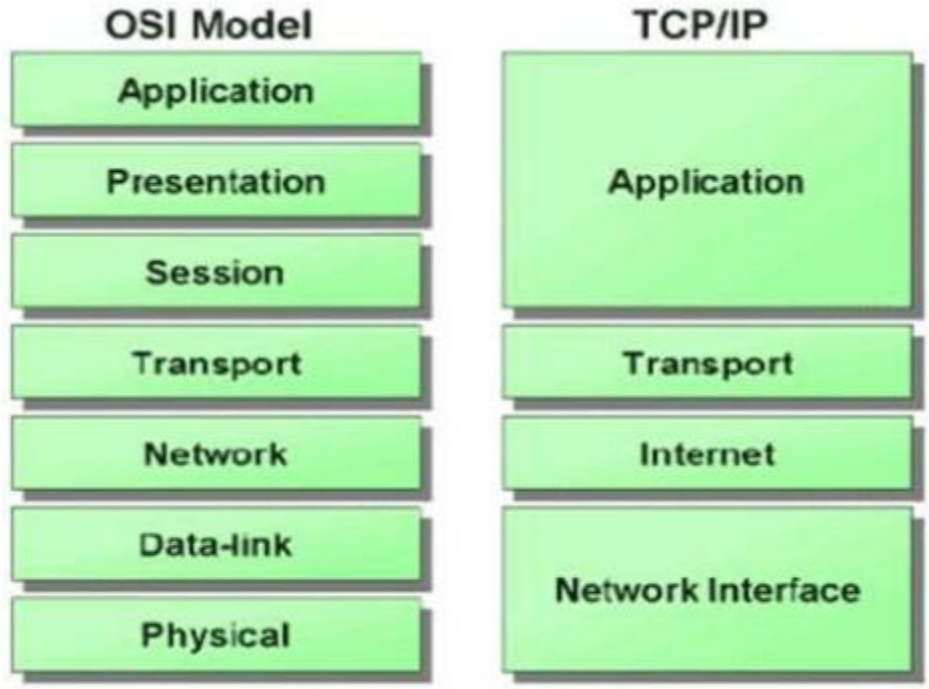
COMPARISON OF OSI REFERENCE MODEL AND TCP/IP REFERENCE MODEL (Compare the OSI Reference Model and TCP/IP Reference Model)

The OSI and TCP/IP models are having many similarities in the functionalities provided by the layers. The layers of TCP model behave similar to the layers of OSI model. But these two models do have differences.

Three concepts are central to the OSI model:

- Services.
- Interfaces.
- Protocols.

- Probably the biggest contribution of the OSI model is that it makes the distinction between these three concepts explicit. Each layer performs some *services* for the layer above it. The service definition tells what the layer does, not how entities above it access it or how the layer works. It defines the layer‟s semantics.
- A layer‟s *interface* tells the processes above it how to access it. It specifies what the parameters are and what results to expect. It, too, says nothing about how the layer works inside.



TCP/IP and the OSI model

Similarities:

The main similarities between the two models include the following:

- **They share similar architecture.** - Both of the models share a similararchitecture. This can be illustrated by the fact that both of them are constructed with layers.

- **They share a common application layer.-** Both of the models share a common"application layer". However in practice this layer includes different services depending upon each model.
- **Both models have comparable transport and network layers-** This can beillustrated by the fact that whatever functions are performed between the presentation and network layer of the OSI model similar functions are performed at the Transport layer of the TCP/IP model.
- **Both models assume that packets are switched-** Basically this means thatindividual packets may take differing paths in order to reach the same destination.

Differences:

- TCP/IP Protocols are considered to be standards around which the internet has developed. The OSI model however is a "generic, protocol- independent standard.“
- TCP/IP combines the presentation and session layer issues into its application layer.
- TCP/IP combines the OSI data link and physical layers into the network access layer.
- TCP/IP appears to be a more simpler model and this is mainly due to the fact that it has fewer layers.
- TCP/IP is considered to be a more credible model- This is mainly due to the fact because TCP/IP protocols are the standards around which the internet was developed therefore it mainly gains creditability due to this reason. Where as in contrast networks are not usually built around the OSI model as it is merely used as a guidance tool.
- The OSI model consists of 7 architectural layers whereas the TCP/IP only has 4 layers.

Comparison:

OSI Model	TCP/IP Model
OSI stands for Open System Interconnection because it allows any two different systems to communicate regardless of their architecture.	TP/IP stands for Transmission Control Protocol/Internet Protocol. It is named after these protocols, being part of this model.
OSI model has seven layers.	TCP/IP has four layers.
This model provides clear distinction between services, interfaces and protocols	It does not clearly distinguish between services, interfaces & protocols.
In this model, Protocols do not fit well into the model.	TCP and IP protocols fit well in the model.
Session & Presentation layers are present in this layer.	There is no session & presentation layer in this model
OSI model supports both connection oriented & connectionless in network layer but connection oriented comm. In transport layer.	TCP/IP supports only connectionless comm. In network layer but supports both in transport layer.

CRITIQUE OF OSI AND TCP/IP REFERENCE MODEL

(List and briefly explain about the Critique of OSI Reference Model and TCP/IP Reference Model)

Critique of OSI Reference Model

Neither the OSI model and its protocols nor the TCP/IP model and its protocols are perfect. Quite a bit of criticism can be, and has been, directed at both of them. In this section and the next one, we will look at some of these criticisms. We will begin with OSI and examine TCP/IP afterward. At the time the second edition of this book was published (1989), it appeared to many experts in the field that the OSI model and its protocols were going to take over the world and push everything else out of their way. This did not happen. Why? A look back at some of the reasons may be useful. They can be summarized as:

- Bad timing.
- Bad technology.
- Bad implementations.
- Bad politics.

Critique of TCP/IP Reference Model

- The TCP/IP model and protocols have their problems too. First, the model does not clearly distinguish the concepts of services, interfaces, and protocols. Good software engineering practice requires differentiating between the specification and the implementation, something that OSI does very carefully, but TCP/IP does not.
  Consequently, the TCP/IP model is not much of a guide for designing new networks using new technologies.
- Second, the TCP/IP model is not at all general and is poorly suited to describing any protocol stack other than TCP/IP. Trying to use the TCP/IP model to describe Bluetooth, for example, is completely impossible.
- Third, the link layer is not really a layer at all in the normal sense of the term as used in the context of layered protocols. It is an interface (between the network and data link layers). The distinction between an interface and a layer is crucial, and one should not be sloppy about it. Fourth, the TCP/IP model does not distinguish between the physical and data link layers.

These are completely different. The physical layer has to do with the transmission characteristics of copper wire, fiber optics, and wireless communication. The data link layer‟s job is to delimit the start and end of frames and get them from one side to the other with the desired degree of reliability. A proper model should include both as separate layers. The TCP/IP model does not do this. Finally, although the IP and TCP protocols were carefully thought out and well implemented, many of the other protocols were ad hoc, generally produced by a couple of graduate students hacking away until they got tired. The protocol implementations were then distributed free, which resulted in their becoming widely used, deeply entrenched, and thus hard to replace. Some of them are a bit of an embarrassment now. The virtual terminal protocol, TELNET, for example, was designed for a ten-character-per-second mechanical Teletype terminal.

5.Explain Theoretical basis for communication.

2.1 THE THEORETICAL BASIS FOR DATA COMMUNICATI

Information can be transmitted on wires by varying some physical property such as voltage or current. By representing the value of this voltage or current as a single-valued function of time, $f(t)$, we can model the behavior of the signal and analyze it mathematically. This analysis is the subject of the following sections.

2.1.1 Fourier Analysis

In the early 19th century, the French mathematician Jean-Baptiste Fourier proved that any reasonably behaved periodic function, $g(t)$ with period T, can be constructed as the sum of a (possibly infinite) number of sines and cosines:

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi nft) + \sum_{n=1}^{\infty} b_n \cos(2\pi nft) \qquad (2\text{-}1)$$

where $f = 1/T$ is the fundamental frequency, a_n and b_n are the sine and cosine amplitudes of the nth **harmonics** (terms), and c is a constant. Such a decomposition is called a **Fourier series**. From the Fourier series, the function can be reconstructed. That is, if the period, T, is known and the amplitudes are given, the original function of time can be found by performing the sums of Eq. (2-1).

A data signal that has a finite duration, which all of them do, can be handled by just imagining that it repeats the entire pattern over and over forever (i.e., the interval from T to $2T$ is the same as from 0 to T, etc.).

The a_n amplitudes can be computed for any given $g(t)$ by multiplying both sides of Eq. (2-1) by $\sin(2\pi kft)$ and then integrating from 0 to T. Since

$$\int_0^T \sin(2\pi kft)\,\sin(2\pi nft)\,dt = \begin{cases} 0 \text{ for } k \neq n \\ T/2 \text{ for } k = n \end{cases}$$

only one term of the summation survives: a_n. The b_n summation vanishes completely. Similarly, by multiplying Eq. (2-1) by $\cos(2\pi kft)$ and integrating between 0 and T, we can derive b_n. By just integrating both sides of the equation as it stands, we can find c. The results of performing these operations are as follows:

$$a_n = \frac{2}{T}\int_0^T g(t)\sin(2\pi nft)\,dt \qquad b_n = \frac{2}{T}\int_0^T g(t)\cos(2\pi nft)\,dt \qquad c = \frac{2}{T}\int_0^T g(t)\,dt$$

2.1.2 Bandwidth-Limited Signals

The relevance of all of this to data communication is that real channels affect different frequency signals differently. Let us consider a specific example: the transmission of the ASCII character "b" encoded in an 8-bit byte. The bit pattern that is to be transmitted is 01100010. The left-hand part of Fig. 2-1(a) shows the

output by the transmitting computer. The Fourier analysis of this sig
the coefficients:

$$a_n = \frac{1}{\pi n}[\cos(\pi n/4) - \cos(3\pi n/4) + \cos(6\pi n/4) - \cos(7\pi n/4)]$$

$$b_n = \frac{1}{\pi n}[\sin(3\pi n/4) - \sin(\pi n/4) + \sin(7\pi n/4) - \sin(6\pi n/4)]$$

$$c = 3/4$$

mean-square amplitudes, $\sqrt{a_n^2 + b_n^2}$, for the first few terms are shown c
right-hand side of Fig. 2-1(a). These values are of interest because the
are proportional to the energy transmitted at the corresponding frequency
transmission facility can transmit signals without losing some power in th
If all the Fourier components were equally diminished, the resulting sig
would be reduced in amplitude but not distorted [i.e., it would have the sam
squared-off shape as Fig. 2-1(a)]. Unfortunately, all transmission facilitie
different Fourier components by different amounts, thus introducing dis
Usually, for a wire, the amplitudes are transmitted mostly undiminished
up to some frequency f_c [measured in cycles/sec or Hertz (Hz)], with all
above this cutoff frequency attenuated. The width of the frequency
transmitted without being strongly attenuated is called the **bandwidth**. In
the cutoff is not really sharp, so often the quoted bandwidth is from 0 to
at which the received power has fallen by half.

The bandwidth is a physical property of the transmission medium that de-
for example, the construction, thickness, and length of a wire or fiber.
often used to further limit the bandwidth of a signal. 802.11 wireless
are allowed to use up to roughly 20 MHz, for example, so 802.11 radios
the signal bandwidth to this size. As another example, traditional (analog)
channels occupy 6 MHz each, on a wire or over the air. This filtering
signals share a given region of spectrum, which improves the overall ef-
of the system. It means that the frequency range for some signals will
at zero, but this does not matter. The bandwidth is still the width of the
frequencies that are passed, and the information that can be carried de-
only on this width and not on the starting and ending frequencies. Signals
from 0 up to a maximum frequency are called **baseband** signals. Signals
shifted to occupy a higher range of frequencies, as is the case for all wire-
transmissions, are called **passband** signals.

Now let us consider how the signal of Fig. 2-1(a) would look if the bandwidth
were low that only the lowest frequencies were transmitted [i.e., if the function
were being approximated by the first few terms of Eq. (2-1)]. Figure 2-1(b)
shows the signal that results from a channel that allows only the first harmonic

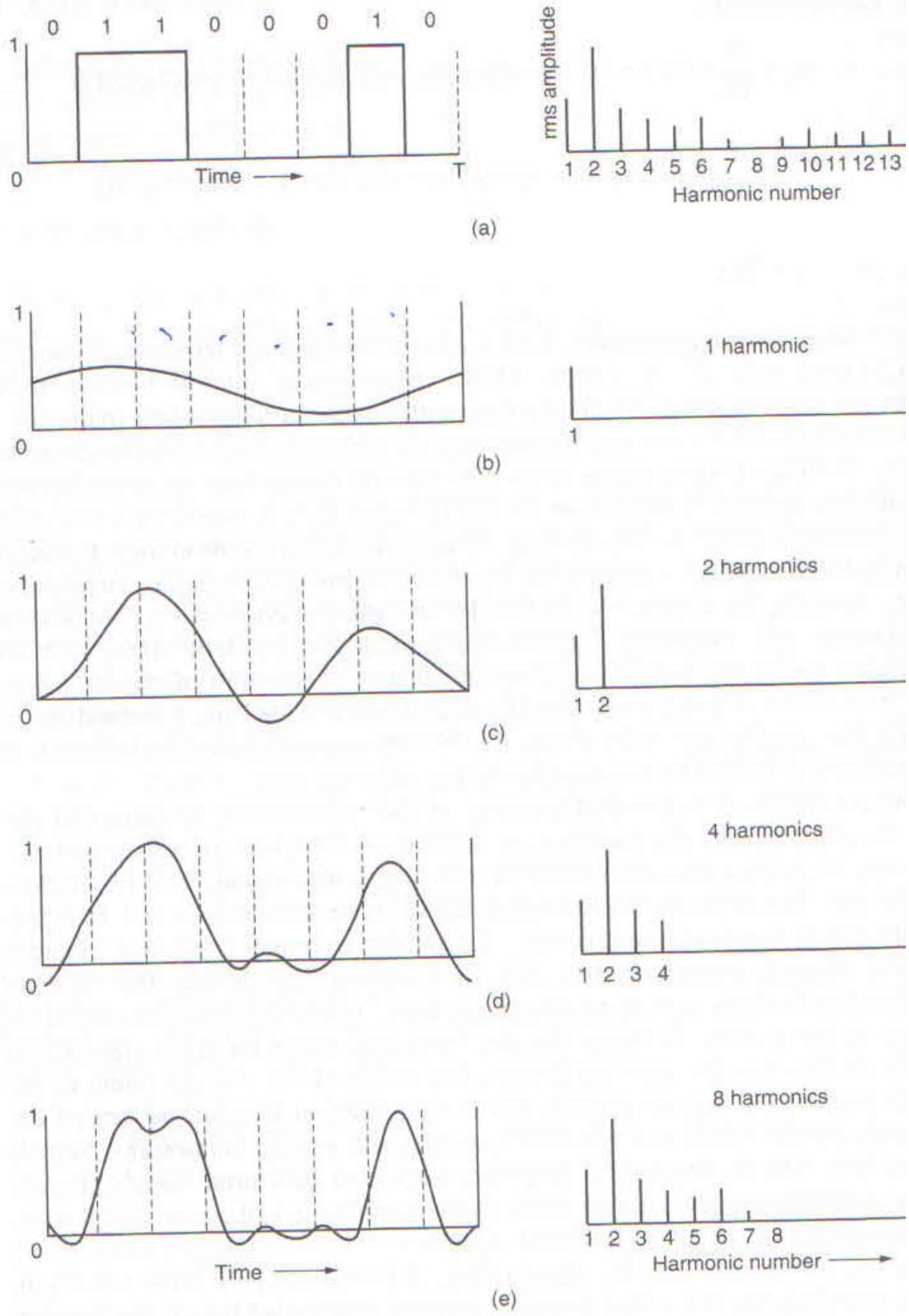


Figure 2-1. (a) A binary signal and its root-mean-square Fourier amplitudes. (b)–(e) Successive approximations to the original signal.

the fundamental, f) to pass through. Similarly, Fig. 2-1(c)–(e) show the spectra and reconstructed functions for higher-bandwidth channels. For digital transmission, the goal is to receive a signal with just enough fidelity to reconstruct the sequence of bits that was sent. We can already do this easily in Fig. 2-1(e), so it is wasteful to use more harmonics to receive a more accurate replica.

Given a bit rate of b bits/sec, the time required to send the 8 bits in our example 1 bit at a time is $8/b$ sec, so the frequency of the first harmonic of this signal is $b/8$ Hz. An ordinary telephone line, often called a **voice-grade line**, has an artificially introduced cutoff frequency just above 3000 Hz. The presence of this restriction means that the number of the highest harmonic passed through is roughly $3000/(b/8)$, or $24{,}000/b$ (the cutoff is not sharp).

For some data rates, the numbers work out as shown in Fig. 2-2. From these numbers, it is clear that trying to send at 9600 bps over a voice-grade telephone line will transform Fig. 2-1(a) into something looking like Fig. 2-1(c), making accurate reception of the original binary bit stream tricky. It should be obvious that at data rates much higher than 38.4 kbps, there is no hope at all for *binary* signals, even if the transmission facility is completely noiseless. In other words, limiting the bandwidth limits the data rate, even for perfect channels. However, coding schemes that make use of several voltage levels do exist and can achieve higher data rates. We will discuss these later in this chapter.

Bps	T (msec)	First harmonic (Hz)	# Harmonics sent
300	26.67	37.5	80
600	13.33	75	40
1200	6.67	150	20
2400	3.33	300	10
4800	1.67	600	5
9600	0.83	1200	2
19200	0.42	2400	1
38400	0.21	4800	0

Figure 2-2. Relation between data rate and harmonics for our example.

There is much confusion about bandwidth because it means different things to electrical engineers and to computer scientists. To electrical engineers, (analog) bandwidth is (as we have described above) a quantity measured in Hz. To computer scientists, (digital) bandwidth is the maximum data rate of a channel, a quantity measured in bits/sec. That data rate is the end result of using the analog bandwidth of a physical channel for digital transmission, and the two are related, as we discuss next. In this book, it will be clear from the context whether we mean analog bandwidth (Hz) or digital bandwidth (bits/sec).

2.1.3 The Maximum Data Rate of a Channel

As early as 1924, an AT&T engineer, Henry Nyquist, realized that even a fect channel has a finite transmission capacity. He derived an equation expres the maximum data rate for a finite-bandwidth noiseless channel. In 1948, Cl Shannon carried Nyquist's work further and extended it to the case of a cha subject to random (that is, thermodynamic) noise (Shannon, 1948). This pap the most important paper in all of information theory. We will just briefly s marize their now classical results here.

Nyquist proved that if an arbitrary signal has been run through a low-pass ter of bandwidth B, the filtered signal can be completely reconstructed by mak only $2B$ (exact) samples per second. Sampling the line faster than $2B$ times second is pointless because the higher-frequency components that such sampl could recover have already been filtered out. If the signal consists of V discr levels, Nyquist's theorem states:

$$\text{maximum data rate} = 2B \log_2 V \text{ bits/sec} \qquad (2$$

For example, a noiseless 3-kHz channel cannot transmit binary (i.e., two-lev signals at a rate exceeding 6000 bps.

So far we have considered only noiseless channels. If random noise is pr ent, the situation deteriorates rapidly. And there is always random (thermal) noi present due to the motion of the molecules in the system. The amount of therm noise present is measured by the ratio of the signal power to the noise power, ca ed the **SNR** (**Signal-to-Noise Ratio**). If we denote the signal power by S and t noise power by N, the signal-to-noise ratio is S/N. Usually, the ratio is expresse on a log scale as the quantity $10 \log_{10} S/N$ because it can vary over a tremendo range. The units of this log scale are called **decibels** (**dB**), with "deci" meani 10 and "bel" chosen to honor Alexander Graham Bell, who invented the tel phone. An S/N ratio of 10 is 10 dB, a ratio of 100 is 20 dB, a ratio of 1000 is 3 dB, and so on. The manufacturers of stereo amplifiers often characterize th bandwidth (frequency range) over which their products are linear by giving the 3 dB frequency on each end. These are the points at which the amplification fact has been approximately halved (because $10 \log_{10} 0.5 \approx -3$).

Shannon's major result is that the maximum data rate or **capacity** of a noisy channel whose bandwidth is B Hz and whose signal-to-noise ratio is S/N, is give by:

$$\text{maximum number of bits/sec} = B \log_2 (1 + S/N) \qquad (2-3$$

This tells us the best capacities that real channels can have. For example, ADS (Asymmetric Digital Subscriber Line), which provides Internet access over nor nal telephone lines, uses a bandwidth of around 1 MHz.

strongly on the distance of the home fr

6.Explain Transmission Media.

The purpose of the physical layer is to transport a raw bit stream from one machine to another. Various physical media can be used for the actual transmission.

Each one has its own niche in terms of bandwidth, delay, cost, and ease of installation and maintenance. Media are roughly grouped into guided media, such as copper wire and fiber optics, and unguided media, such as radio and lasers through the air. We will look at all of these in the following sections.

- Magnetic Media
- Twisted Pair
- Coaxial Cable
- Power Lines
- Fiber Optics

1. Magnetic Media

One of the most common ways to transport data from one computer to another is to write them onto magnetic tape or removable media (e.g., recordable DVDs), physically transport the tape or disks to the destination machine, and read them back in again. Although this method is not as sophisticated as using a geosynchronous communication satellite, it is often more cost effective, especially for applications in which high bandwidth or cost per bit transported is the key factor.

A simple calculation will make this point clear.

- An industry standard Ultrium tape can hold 200 gigabytes.
- A box 60 x 60 x 60 cm can hold about 1000 of these tapes, for a total capacity of 200 terabytes, or 1600 terabits (1.6 petabits).
- A box of tapes can be delivered anywhere in the United States in 24 hours by Federal Express and other companies.
- The effective bandwidth of this transmission is 1600 terabits/86,400 sec, or 19 Gbps. If the destination is only an hour away by road, the bandwidth is increased to over 400 Gbps. No computer network can even approach this.

For a bank with many gigabytes of data to be backed up daily on a second machine (so the bank can continue to function even in the face of a major flood or earthquake), it is likely that no other transmission technology can even begin to approach magnetic tape for performance. Of course, networks are getting faster, but tape densities are increasing, too.

If we now look at **cost**, we get a similar picture.

- The cost of an Ultrium tape is around $40 when bought in bulk.

- A tape can be reused at least ten times, so the tape cost is maybe $4000 per box per usage. Add to this another $1000 for shipping (probably much less), and we have a cost of roughly $5000 to ship 200 TB.
- This amounts to shipping a gigabyte for under 3 cents. No network can beat that. The moral of the story is:

Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway

2. Twisted Pair

Characteristics

- The bandwidth characteristics of magnetic tape are excellent, the delay characteristics are poor. Transmission time is measured in minutes or hours, not milliseconds.
- For many applications an on-line connection is needed. One of the oldest and still most common transmission media is twisted pair.
- A twisted pair consists of two insulated copper wires, typically about 1 mm thick.
- The wires are twisted together in a helical form, just like a DNA molecule. Twisting is done because two parallel wires constitute a fine antenna. When the wires are twisted, the waves from different twists cancel out, so the wire radiates less effectively.

The most common application of the twisted pair is the telephone system. Nearly all telephones are connected to the telephone company (telco) office by a twisted pair. Twisted pairs can run several kilometers without amplification, but for longer distances, repeaters are needed.

When many twisted pairs run in parallel for a substantial distance, such as all the wires coming from an apartment building to the telephone company office, they are bundled together and encased in a protective sheath. The pairs in these bundles would interfere with one another if it were not for the twisting. In parts of the world where telephone lines run on poles above ground, it is common to see bundles several centimeters in diameter.

Twisted pairs can be used for transmitting either **analog or digital signals.** The bandwidth depends on the thickness of the wire and the distance traveled, but several megabits/sec can be achieved for a few kilometers in many cases. Due to their adequate performance and low cost, twisted pairs are widely used and are likely to remain so for years to come.

Twisted pair cabling comes in several varieties, two of which are important for computer networks. Category 3 twisted pairs consist of two insulated wires gently twisted together. Four such pairs are typically grouped in a plastic sheath to protect the wires and keep them together. Prior to about 1988, most office buildings had one category 3 cable running from a central wiring closet on each floor into each office. This scheme allowed up to four regular telephones or two multiline telephones in each office to connect to the telephone company equipment in the wiring closet.

Starting around 1988, the more advanced category 5 twisted pairs were introduced. They are similar to category 3 pairs, but with more twists per centimeter, which results in less crosstalk and a better-quality signal over longer distances, making them more suitable for high-speed computer communication. Up-and-coming categories are 6 and 7, which are capable of handling signals with bandwidths of 250 MHz and 600 MHz, respectively (versus a mere 16 MHz and 100 MHz for categories 3 and 5, respectively).

All of these wiring types are often referred to as UTP (Unshielded Twisted Pair), to contrast them with the bulky, expensive, shielded twisted pair cables IBM introduced in the early 1980s, but which have not proven popular outside of IBM installations. Twisted pair cabling is illustrated in Fig. 2-3.

Figure 2-3. (a) Category 3 UTP. (b) Category 5 UTP.

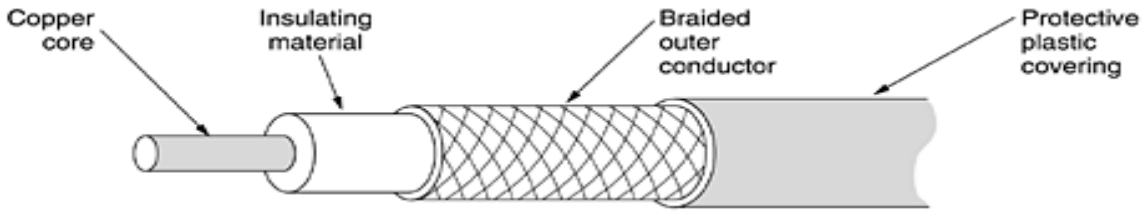
3. Coaxial Cable

Another common transmission medium is the coaxial cable (known to its many friends as just "coax" and pronounced "co-ax"). It has better shielding than twisted pairs, so it can span longer distances at higher speeds.

Two kinds of coaxial cable are widely used. One kind, 50-ohm cable, is commonly used when it is intended for digital transmission from the start. The other kind, 75-ohm cable, is commonly used for analog transmission and cable television but is becoming more important with the advent of Internet over cable. This distinction is based on historical, rather than technical, factors (e.g., early dipole antennas had an impedance of 300 ohms, and it was easy to use existing 4:1 impedance matching transformers).

A coaxial cable consists of a stiff copper wire as the core, surrounded by an insulating material. The insulator is encased by a cylindrical conductor, often as a closely-woven braided mesh. The outer conductor is covered in a protective plastic sheath. A cutaway view of a coaxial cable is shown in Fig. 2-4.

Figure 2-4. A coaxial cable.



The construction and shielding of the coaxial cable give it a good combination of high bandwidth and excellent noise immunity. The bandwidth possible depends on the cable quality, length, and signal-to-noise ratio of the data signal.

Modern cables have a bandwidth of close to 1 GHz. Coaxial cables used to be widely used within the telephone system for long-distance lines but have now largely been replaced by fiber optics on long-haul routes. Coax is still widely used for cable television and metropolitan area networks, however.

4. Power Lines:

The telephone and cable television networks are not only source of wiring that can be reused for data communication. Power lines deliver electric power to houses and electric wiring within houses distributes the power to electrical outlets.

The convienence of using power lines for networking should be clear. Simply plug a TV and a receiver into a wall, which you must do anyway because they need power, and they can send and receive movies over the electrical wiring.

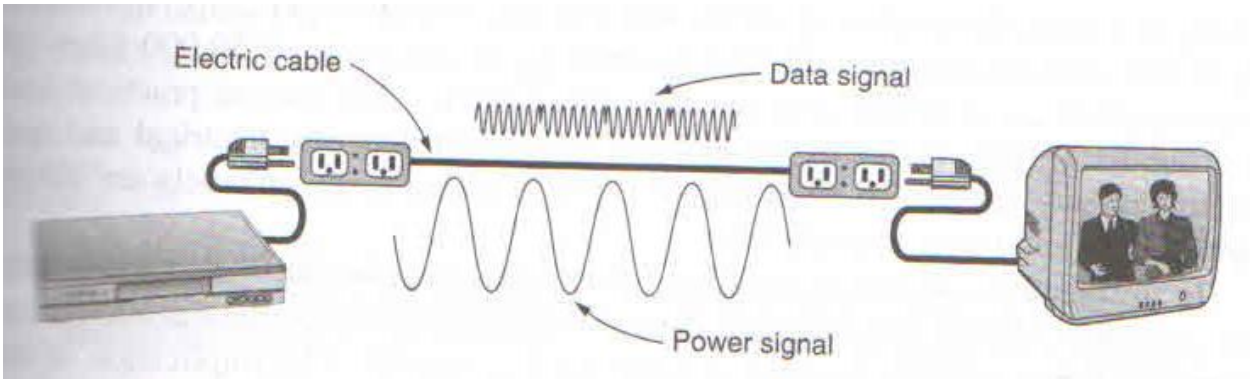


Figure 2-5. A network that uses household electrical wiring.

5. Fiber Optics

Many people in the computer industry take enormous pride in how fast computer technology is improving. The original (1981) IBM PC ran at a clock speed of 4.77 MHz. Twenty years later, PCs could run at 2 GHz, a gain of a factor of 20 per decade. Not too bad.

In the same period, wide area data communication went from 56 kbps (the ARPANET) to 1 Gbps (modern optical communication), a gain of more than a factor of 125 per decade, while at the same time the error rate went from 10^{-5} per bit to almost zero.

Furthermore, single CPUs are beginning to approach physical limits, such as speed of light and heat dissipation problems. In contrast, with current fiber technology, the achievable bandwidth is certainly in excess of 50,000 Gbps (50 Tbps) and many people are looking very hard for better technologies and materials. The current practical signaling limit of about 10 Gbps is due to our inability to convert between electrical and optical signals any faster, although in the laboratory, 100 Gbps has been achieved on a single fiber.

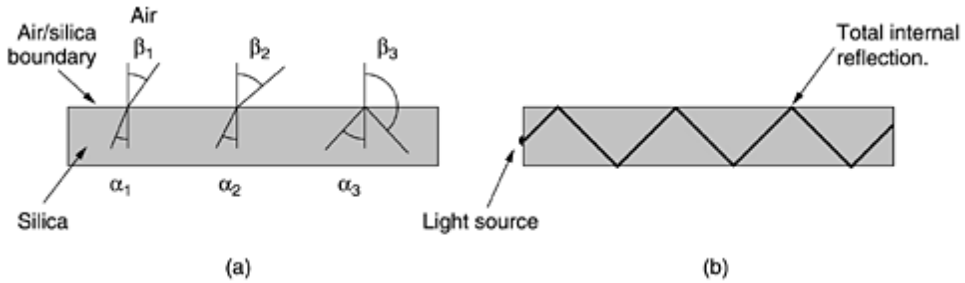
In the race between computing and communication, communication won. The full implications of essentially infinite bandwidth (although not at zero cost) have not yet sunk in to a generation of computer scientists and engineers taught to think in terms of the low Nyquist and Shannon limits imposed by copper wire. The new conventional wisdom should be that all computers are hopelessly slow and that networks should try to avoid computation at all costs, no matter how much bandwidth that wastes. In this section we will study fiber optics to see how that transmission technology works.

An optical transmission system has three key components: the light source, the transmission medium, and the detector. Conventionally, a pulse of light indicates a 1 bit and the absence of light indicates a 0 bit. The transmission medium is an ultra-thin fiber of glass. The detector generates an electrical pulse when light falls on it. By attaching a light source to one end of an optical fiber and a detector to the other, we have a unidirectional data transmission system that accepts an

electrical signal, converts and transmits it by light pulses, and then reconverts the output to an electrical signal at the receiving end.

This transmission system would leak light and be useless in practice except for an interesting principle of physics. When a light ray passes from one medium to another, for example, from fused silica to air, the ray is refracted (bent) at the silica/air boundary, as shown in Fig. 2-5(a). Here we see a light ray incident on the boundary at an angle a_1 emerging at an angle b_1. The amount of refraction depends on the properties of the two media (in particular, their indices of refraction). For angles of incidence above a certain critical value, the light is refracted back into the silica; none of it escapes into the air. Thus, a light ray incident at or above the critical angle is trapped inside the fiber, as shown in Fig. 2-5(b), and can propagate for many kilometers with virtually no loss.

Figure 2-5. (a) Three examples of a light ray from inside a silica fiber impinging on the air/silica boundary at different angles. (b) Light trapped by total internal reflection.



The sketch of Fig. 2-5(b) shows only one trapped ray, but since any light ray incident on the boundary above the critical angle will be reflected internally, many different rays will be bouncing around at different angles. Each ray is said to have a different mode, so a fiber having this property is called a multimode fiber.

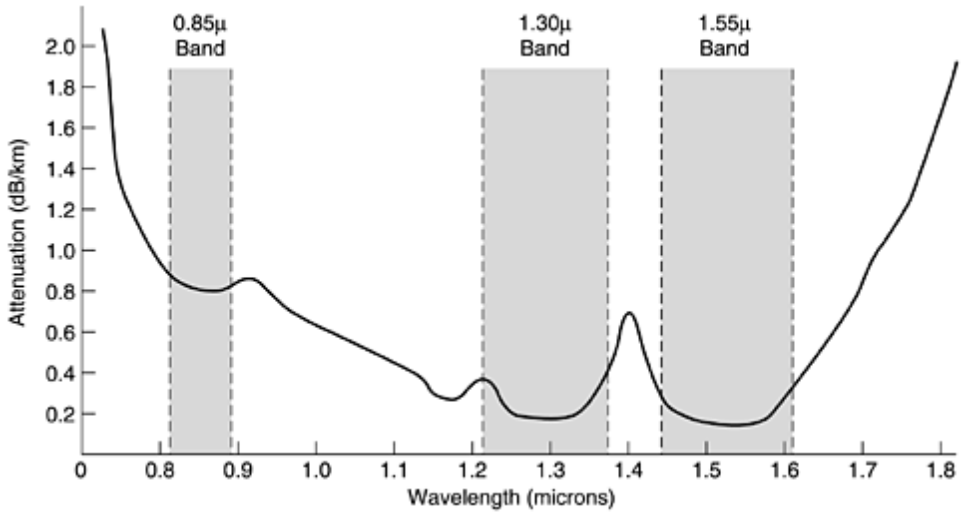
However, if the fiber's diameter is reduced to a few wavelengths of light, the fiber acts like a wave guide, and the light can propagate only in a straight line, without bouncing, yielding a single-mode fiber. Single-mode fibers are more expensive but are widely used for longer distances. Currently available single-mode fibers can transmit data at 50 Gbps for 100 km without amplification. Even higher data rates have been achieved in the laboratory for shorter distances.

Transmission of Light through Fiber

Optical fibers are made of glass, which, in turn, is made from sand, an inexpensive raw material available in unlimited amounts. Glassmaking was known to the ancient Egyptians, but their glass had to be no more than 1 mm thick or the light could not shine through. Glass transparent enough to be useful for windows was developed during the Renaissance. The glass used for modern optical fibers is so transparent that if the oceans were full of it instead of water, the seabed would be as visible from the surface as the ground is from an airplane on a clear day.

The attenuation of light through glass depends on the wavelength of the light (as well as on some physical properties of the glass). For the kind of glass used in fibers, the attenuation is shown in Fig. 2-6 in decibels per linear kilometer of fiber. The attenuation in decibels is given by the formula

Figure 2-6. Attenuation of light through fiber in the infrared region.



$$\text{Attenuation in decibels} = 10 \log_{10} \frac{\text{transmitted power}}{\text{received power}}$$

For example, a factor of two loss gives an attenuation of $10 \log_{10} 2 = 3$ dB. The figure shows the near infrared part of the spectrum, which is what is used in practice. Visible light has slightly shorter wavelengths, from 0.4 to 0.7 microns (1 micron is 10^{-6} meters). The true metric purist would refer to these wavelengths as 400 nm to 700 nm, but we will stick with traditional usage.

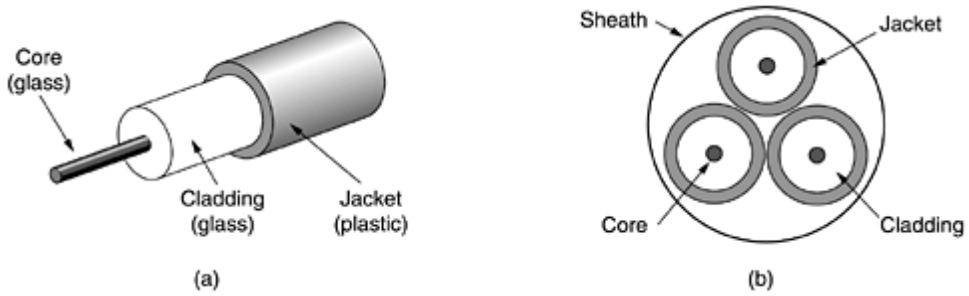
Three wavelength bands are used for optical communication. They are centered at 0.85, 1.30, and 1.55 microns, respectively. The last two have good attenuation properties (less than 5 percent loss per kilometer). The 0.85 micron band has higher attenuation, but at that wavelength the lasers and electronics can be made from the same material (gallium arsenide). All three bands are 25,000 to 30,000 GHz wide.

Light pulses sent down a fiber spread out in length as they propagate. This spreading is called chromatic dispersion. The amount of it is wavelength dependent. One way to keep these spread-out pulses from overlapping is to increase the distance between them, but this can be done only by reducing the signaling rate. Fortunately, it has been discovered that by making the pulses in a special shape related to the reciprocal of the hyperbolic cosine, nearly all the dispersion effects cancel out, and it is possible to send pulses for thousands of kilometers without appreciable shape distortion. These pulses are called solitons. A considerable amount of research is going on to take solitons out of the lab and into the field.

Fiber Cables

Fiber optic cables are similar to coax, except without the braid. Figure 2-7(a) shows a single fiber viewed from the side. At the center is the glass core through which the light propagates. In multimode fibers, the core is typically 50 microns in diameter, about the thickness of a human hair. In single-mode fibers, the core is 8 to 10 microns.

Figure 2-7. (a) Side view of a single fiber. (b) End view of a sheath with three fibers.



The core is surrounded by a glass cladding with a lower index of refraction than the core, to keep all the light in the core. Next comes a thin plastic jacket to protect the cladding. Fibers are typically grouped in bundles, protected by an outer sheath. Figure 2-7(b) shows a sheath with three fibers.

Terrestrial fiber sheaths are normally laid in the ground within a meter of the surface, where they are occasionally subject to attacks by backhoes or gophers. Near the shore, transoceanic fiber sheaths are buried in trenches by a kind of seaplow. In deep water, they just lie on the bottom, where they can be snagged by fishing trawlers or attacked by giant squid.

Fibers can be connected in three different ways. First, they can terminate in connectors and be plugged into fiber sockets. Connectors lose about 10 to 20 percent of the light, but they make it easy to reconfigure systems.

Second, they can be spliced mechanically. Mechanical splices just lay the two carefully-cut ends next to each other in a special sleeve and clamp them in place. Alignment can be improved by passing light through the junction and then making small adjustments to maximize the signal. Mechanical splices take trained personnel about 5 minutes and result in a 10 percent light loss.

Third, two pieces of fiber can be fused (melted) to form a solid connection. A fusion splice is almost as good as a single drawn fiber, but even here, a small amount of attenuation occurs.

For all three kinds of splices, reflections can occur at the point of the splice, and the reflected energy can interfere with the signal.

Two kinds of light sources are typically used to do the signaling, LEDs (Light Emitting Diodes) and semiconductor lasers. They have different properties, as shown in Fig. 2-8. They can be tuned in wavelength by inserting Fabry-Perot or Mach-Zehnder interferometers between the source and the fiber. Fabry-Perot interferometers are simple resonant cavities consisting of two parallel mirrors. The light is incident perpendicular to the mirrors. The length of the cavity selects out those wavelengths that fit inside an integral number of times. Mach-Zehnder interferometers separate the light into two beams. The two beams travel slightly different distances. They are recombined at the end and are in phase for only certain wavelengths.

Figure 2-8. A comparison of semiconductor diodes and LEDs as light sources.

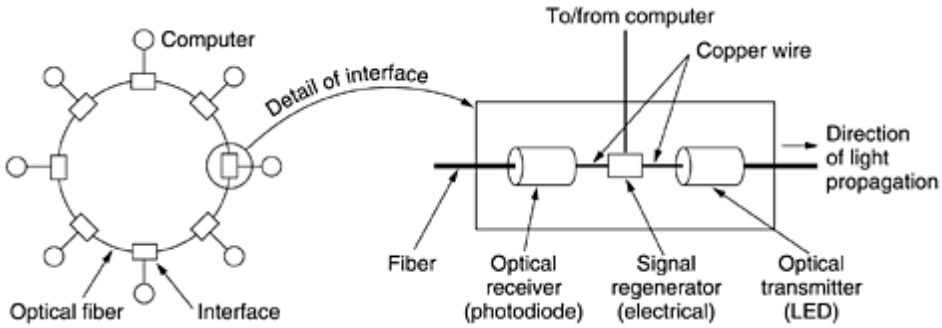
Item	LED	Semiconductor laser
Data rate	Low	High
Fiber type	Multimode	Multimode or single mode
Distance	Short	Long
Lifetime	Long life	Short life
Temperature sensitivity	Minor	Substantial
Cost	Low cost	Expensive

The receiving end of an optical fiber consists of a photodiode, which gives off an electrical pulse when struck by light. The typical response time of a photodiode is 1 nsec, which limits data rates to about 1 Gbps. Thermal noise is also an issue, so a pulse of light must carry enough energy to be detected. By making the pulses powerful enough, the error rate can be made arbitrarily small.

Fiber Optic Networks

Fiber optics can be used for LANs as well as for long-haul transmission, although tapping into it is more complex than connecting to an Ethernet. One way around the problem is to realize that a ring network is really just a collection of point-to-point links, as shown in Fig. 2-9. The interface at each computer passes the light pulse stream through to the next link and also serves as a T junction to allow the computer to send and accept messages.

Figure 2-9. A fiber optic ring with active repeaters.



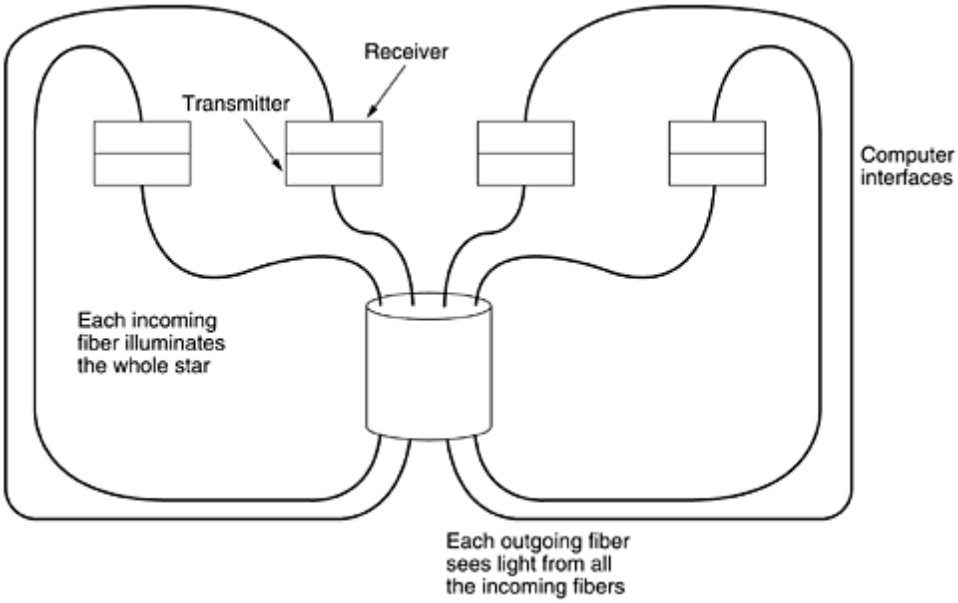
Two types of interfaces are used. A passive interface consists of two taps fused onto the main fiber. One tap has an LED or laser diode at the end of it (for transmitting), and the other has a photodiode (for receiving). The tap itself is completely passive and is thus extremely reliable because a broken LED or photodiode does not break the ring. It just takes one computer off-line.

The other interface type, shown in Fig. 2-9, is the active repeater. The incoming light is converted to an electrical signal, regenerated to full strength if it has been weakened, and retransmitted as light. The interface with the computer is an ordinary copper wire that comes into the signal regenerator. Purely optical repeaters are now being used, too. These devices do not require the optical to electrical to optical conversions, which means they can operate at extremely high bandwidths.

If an active repeater fails, the ring is broken and the network goes down. On the other hand, since the signal is regenerated at each interface, the individual computer-to-computer links can be kilometers long, with virtually no limit on the total size of the ring. The passive interfaces lose light at each junction, so the number of computers and total ring length are greatly restricted.

A ring topology is not the only way to build a LAN using fiber optics. It is also possible to have hardware broadcasting by using the passive star construction of Fig. 2-10. In this design, each interface has a fiber running from its transmitter to a silica cylinder, with the incoming fibers fused to one end of the cylinder. Similarly, fibers fused to the other end of the cylinder are run to each of the receivers. Whenever an interface emits a light pulse, it is diffused inside the passive star to illuminate all the receivers, thus achieving broadcast. In effect, the passive star combines all the incoming signals and transmits the merged result on all lines. Since the incoming energy is divided among all the outgoing lines, the number of nodes in the network is limited by the sensitivity of the photodiodes.

Figure 2-10. A passive star connection in a fiber optics network.



Comparison of Fiber Optics and Copper Wire

It is instructive to compare fiber to copper. Fiber has many advantages. To start with, it can handle much higher bandwidths than copper. This alone would require its use in high-end networks. Due to the low attenuation, repeaters are needed only about every 50 km on long lines, versus about every 5 km for copper, a substantial cost saving. Fiber also has the advantage of not being affected by power surges, electromagnetic interference, or power failures. Nor is it affected by corrosive chemicals in the air, making it ideal for harsh factory environments.

Oddly enough, telephone companies like fiber for a different reason: it is thin and lightweight. Many existing cable ducts are completely full, so there is no room to add new capacity. Removing all the copper and replacing it by fiber empties the ducts, and the copper has excellent resale value to copper refiners who see it as very high grade ore. Also, fiber is much lighter than copper. One thousand twisted pairs 1 km long weigh 8000 kg. Two fibers have more capacity and weigh only 100 kg, which greatly reduces the need for expensive mechanical support systems that must be maintained. For new routes, fiber wins hands down due to its much lower installation cost.

Finally, fibers do not leak light and are quite difficult to tap. These properties gives fiber excellent security against potential wiretappers.

On the downside, fiber is a less familiar technology requiring skills not all engineers have, and fibers can be damaged easily by being bent too much. Since optical transmission is inherently unidirectional, two-way communication requires either two fibers or two frequency bands on one fiber. Finally, fiber interfaces cost more than electrical interfaces. Nevertheless, the future of all fixed data communication for distances of more than a few meters is clearly with fiber. For a discussion of all aspects of fiber optics and their networks, see (Hecht, 2001).

7.Explain Wireless Transmission

Our age has given rise to information junkies: people who need to be on-line all the time. For these mobile users, twisted pair, coax, and fiber optics are of no use. They need to get their hits of data for their laptop, notebook, shirt pocket, palmtop, or wristwatch computers without

being tethered to the terrestrial communication infrastructure. For these users, wireless communication is the answer. In the following sections, we will look at wireless communication in general, as it has many other important applications besides providing connectivity to users who want to surf the Web from the beach.

Some people believe that the future holds only two kinds of communication: fiber and wireless. All fixed (i.e., non mobile) computers, telephones, faxes, and so on will use fiber, and all mobile ones will use wireless.

Wireless has advantages for even fixed devices in some circumstances. For example, if running a fiber to a building is difficult due to the terrain (mountains, jungles, swamps, etc.), wireless may be better. It is noteworthy that modern wireless digital communication began in the Hawaiian Islands, where large chunks of Pacific Ocean separated the users and the telephone system was inadequate.

- The Electromagnetic Spectrum
- Radio Transmission
- Microwave Transmission
- Infrared and millimeter waves
- Lightwave Transmission

1. The Electromagnetic Spectrum

When electrons move, they create electromagnetic waves that can propagate through space (even in a vacuum). These waves were predicted by the British physicist James Clerk Maxwell in 1865 and first observed by the German physicist Heinrich Hertz in 1887. The number of oscillations per second of a wave is called its frequency, f, and is measured in Hz (in honor of Heinrich Hertz). The distance between two consecutive maxima (or minima) is called the wavelength, which is universally designated by the Greek letter l (lambda).

When an antenna of the appropriate size is attached to an electrical circuit, the electromagnetic waves can be broadcast efficiently and received by a receiver some distance away. All wireless communication is based on this principle.

In vacuum, all electromagnetic waves travel at the same speed, no matter what their frequency. This speed, usually called the speed of light, c, is approximately 3×10^8 m/sec, or about 1 foot (30 cm) per nanosecond. (A case could be made for redefining the foot as the distance light travels in a vacuum in 1 nsec rather than basing it on the shoe size of some long-dead king.) In copper or fiber the speed slows to about 2/3 of this value and becomes slightly frequency dependent. The speed of light is the ultimate speed limit. No object or signal can ever move faster than it.

The fundamental relation between f, l, and c (in vacuum) is

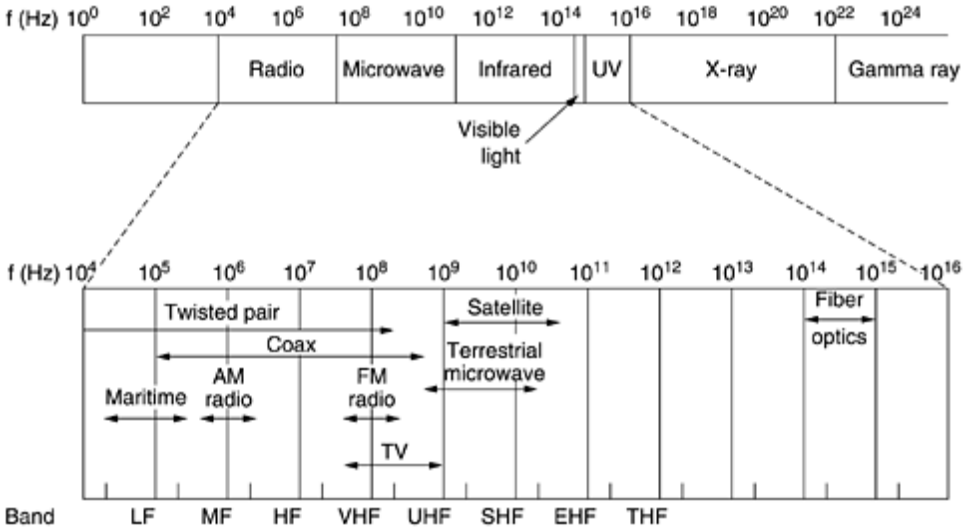
Equation 2

$$\lambda f = c$$

Since c is a constant, if we know f, we can find l, and vice versa. As a rule of thumb, when l is in meters and f is in MHz, lf 300. For example, 100-MHz waves are about 3 meters long, 1000-MHz waves are 0.3-meters long, and 0.1-meter waves have a frequency of 3000 MHz.

The electromagnetic spectrum is shown in Fig. 2-11. The radio, microwave, infrared, and visible light portions of the spectrum can all be used for transmitting information by modulating the amplitude, frequency, or phase of the waves. Ultraviolet light, X-rays, and gamma rays would be even better, due to their higher frequencies, but they are hard to produce and modulate, do not propagate well through buildings, and are dangerous to living things. The bands listed at the bottom of Fig. 2-11 are the official ITU names and are based on the wavelengths, so the LF band goes from 1 km to 10 km (approximately 30 kHz to 300 kHz). The terms LF, MF, and HF refer to low, medium, and high frequency, respectively. Clearly, when the names were assigned, nobody expected to go above 10 MHz, so the higher bands were later named the Very, Ultra, Super, Extremely, and Tremendously High Frequency bands. Beyond that there are no names, but Incredibly, Astonishingly, and Prodigiously high frequency (IHF, AHF, and PHF) would sound nice.

Figure 2-11. The electromagnetic spectrum and its uses for communication.



The amount of information that an electromagnetic wave can carry is related to its bandwidth. With current technology, it is possible to encode a few bits per Hertz at low frequencies, but often as many as 8 at high frequencies, so a coaxial cable with a 750 MHz bandwidth can carry several gigabits/sec. From Fig. 2-11 it should now be obvious why networking people like fiber optics so much.

If we solve Eq. (2-2) for f and differentiate with respect to l, we get

$$\frac{df}{d\lambda} = -\frac{c}{\lambda^2}$$

If we now go to finite differences instead of differentials and only look at absolute values, we get

Equation 2

$$\Delta f = \frac{c\Delta\lambda}{\lambda^2}$$

Thus, given the width of a wavelength band, Dl, we can compute the corresponding frequency band, Df, and from that the data rate the band can produce. The wider the band, the higher the data rate. As an example, consider the 1.30-micron band of Fig. 2-6. Here we have $l=1.3 \times 10^{-6}$ and $Dl = 0.17 \times 10^{-6}$,soDf is about 30 THz. At, say, 8 bits/Hz, we get 240 Tbps.

Most transmissions use a narrow frequency band (i.e., Df/f 1) to get the best reception (many watts/Hz). However, in some cases, a wide band is used, with two variations. In frequency hopping spread spectrum, the transmitter hops from frequency to frequency hundreds of times per second. It is popular for military communication because it makes transmissions hard to detect and next to impossible to jam. It also offers good resistance to multipath fading because the direct signal always arrives at the receiver first. Reflected signals follow a longer path and arrive later. By then the receiver may have changed frequency and no longer accepts signals on the previous frequency, thus eliminating interference between the direct and reflected signals. In recent years, this technique has also been applied commercially—both 802.11 and Bluetooth use it, for example.

As a curious footnote, the technique was co-invented by the Austrian-born sex goddess Hedy Lamarr, the first woman to appear nude in a motion picture (the 1933 Czech film Extase). Her first husband was an armaments manufacturer who told her how easy it was to block the radio signals then used to control torpedos. When she discovered that he was selling weapons to Hitler, she was horrified, disguised herself as a maid to escape him, and fled to Hollywood to continue her career as a movie actress. In her spare time, she invented frequency hopping to help the Allied war effort. Her scheme used 88 frequencies, the number of keys (and frequencies) on the piano. For their invention, she and her friend, the musical composer George Antheil, received U.S. patent 2,292,387. However, they were unable to convince the U.S. Navy that their invention had any practical use and never received any royalties. Only years after the patent expired did it become popular.

The other form of spread spectrum, direct sequence spread spectrum, which spreads the signal over a wide frequency band, is also gaining popularity in the commercial world. In particular, some second-generation mobile phones use it, and it will become dominant with the third generation, thanks to its good spectral efficiency, noise immunity, and other properties. Some wireless LANs also use it. We will come back to spread spectrum later in this chapter. For a fascinating and detailed history of spread spectrum communication, see (Scholtz, 1982).

For the moment, we will assume that all transmissions use a narrow frequency band. We will now discuss how the various parts of the electromagnetic spectrum of Fig. 2-11 are used, starting with radio.

2. Radio Transmission

Radio waves are easy to generate, can travel long distances, and can penetrate buildings easily, so they are widely used for communication, both indoors and outdoors. Radio waves also are omnidirectional, meaning that they travel in all directions from the source, so the transmitter and receiver do not have to be carefully aligned physically.

Sometimes omnidirectional radio is good, but sometimes it is bad. In the 1970s, General Motors decided to equip all its new Cadillacs with computer-controlled antilock brakes. When the driver stepped on the brake pedal, the computer pulsed the brakes on and off instead of locking them on hard. One fine day an Ohio Highway Patrolman began using his new mobile radio to call headquarters, and suddenly the Cadillac next to him began behaving like a bucking bronco. When

the officer pulled the car over, the driver claimed that he had done nothing and that the car had gone crazy.

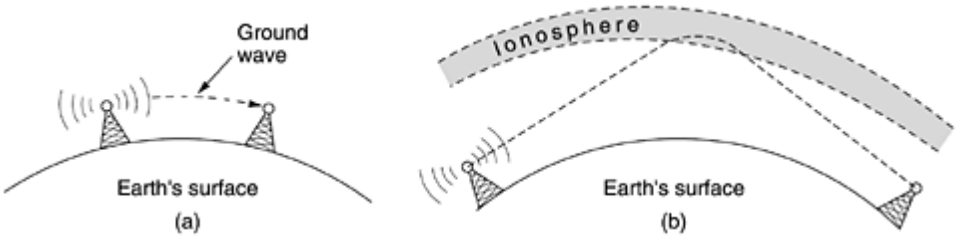
Eventually, a pattern began to emerge: Cadillacs would sometimes go berserk, but only on major highways in Ohio and then only when the Highway Patrol was watching. For a long, long time General Motors could not understand why Cadillacs worked fine in all the other states and also on minor roads in Ohio. Only after much searching did they discover that the Cadillac's wiring made a fine antenna for the frequency used by the Ohio Highway Patrol's new radio system.

The properties of radio waves are frequency dependent. At low frequencies, radio waves pass through obstacles well, but the power falls off sharply with distance from the source, roughly as $1/r^2$ in air. At high frequencies, radio waves tend to travel in straight lines and bounce off obstacles. They are also absorbed by rain. At all frequencies, radio waves are subject to interference from motors and other electrical equipment.

Due to radio's ability to travel long distances, interference between users is a problem. For this reason, all governments tightly license the use of radio transmitters, with one exception, discussed below.

In the VLF, LF, and MF bands, radio waves follow the ground, as illustrated in Fig. 2-12(a). These waves can be detected for perhaps 1000 km at the lower frequencies, less at the higher ones. AM radio broadcasting uses the MF band, which is why the ground waves from Boston AM radio stations cannot be heard easily in New York. Radio waves in these bands pass through buildings easily, which is why portable radios work indoors. The main problem with using these bands for data communication is their low bandwidth [see Eq. (2-3)].

Figure 2-12. (a) In the VLF, LF, and MF bands, radio waves follow the curvature of the earth. (b) In the HF band, they bounce off the ionosphere.



In the HF and VHF bands, the ground waves tend to be absorbed by the earth. However, the waves that reach the ionosphere, a layer of charged particles circling the earth at a height of 100 to 500 km, are refracted by it and sent back to earth, as shown in Fig. 2-12(b). Under certain atmospheric conditions, the signals can bounce several times. Amateur radio operators (hams) use these bands to talk long distance. The military also communicate in the HF and VHF bands.

3. Microwave Transmission

Above 100 MHz, the waves travel in nearly straight lines and can therefore be narrowly focused. Concentrating all the energy into a small beam by means of a parabolic antenna (like the familiar satellite TV dish) gives a much higher signal-to-noise ratio, but the transmitting and receiving antennas must be accurately aligned with each other. In addition, this directionality allows multiple transmitters lined up in a row to communicate with multiple receivers in a row without interference, provided some minimum spacing rules are observed. Before fiber optics, for decades these microwaves formed the heart of the long-distance telephone transmission system. In fact,

MCI, one of AT&T's first competitors after it was deregulated, built its entire system with microwave communications going from tower to tower tens of kilometers apart. Even the company's name reflected this (MCI stood for Microwave Communications, Inc.). MCI has since gone over to fiber and merged with WorldCom.

Since the microwaves travel in a straight line, if the towers are too far apart, the earth will get in the way (think about a San Francisco to Amsterdam link). Consequently, repeaters are needed periodically. The higher the towers are, the farther apart they can be. The distance between repeaters goes up very roughly with the square root of the tower height. For 100-meter-high towers, repeaters can be spaced 80 km apart.

Unlike radio waves at lower frequencies, microwaves do not pass through buildings well. In addition, even though the beam may be well focused at the transmitter, there is still some divergence in space. Some waves may be refracted off low-lying atmospheric layers and may take slightly longer to arrive than the direct waves. The delayed waves may arrive out of phase with the direct wave and thus cancel the signal. This effect is called multipath fading and is often a serious problem. It is weather and frequency dependent. Some operators keep 10 percent of their channels idle as spares to switch on when multipath fading wipes out some frequency band temporarily.

The demand for more and more spectrum drives operators to yet higher frequencies. Bands up to 10 GHz are now in routine use, but at about 4 GHz a new problem sets in: absorption by water. These waves are only a few centimeters long and are absorbed by rain. This effect would be fine if one were planning to build a huge outdoor microwave oven for roasting passing birds, but for communication, it is a severe problem. As with multipath fading, the only solution is to shut off links that are being rained on and route around them.

In summary, microwave communication is so widely used for long-distance telephone communication, mobile phones, television distribution, and other uses that a severe shortage of spectrum has developed. It has several significant advantages over fiber. The main one is that no right of way is needed, and by buying a small plot of ground every 50 km and putting a microwave tower on it, one can bypass the telephone system and communicate directly. This is how MCI managed to get started as a new long-distance telephone company so quickly. (Sprint went a completely different route: it was formed by the Southern Pacific Railroad, which already owned a large amount of right of way and just buried fiber next to the tracks.)

Microwave is also relatively inexpensive. Putting up two simple towers (may be just big poles with four guy wires) and putting antennas on each one may be cheaper than burying 50 km of fiber through a congested urban area or up over a mountain, and it may also be cheaper than leasing the telephone company's fiber, especially if the telephone company has not yet even fully paid for the copper it ripped out when it put in the fiber.

The Politics of the Electromagnetic Spectrum

To prevent total chaos, there are national and international agreements about who gets to use which frequencies. Since everyone wants a higher data rate, everyone wants more spectrum. National governments allocate spectrum for AM and FM radio, television, and mobile phones, as well as for telephone companies, police, maritime, navigation, military, government, and many other competing users. Worldwide, an agency of ITU-R (WARC) tries to coordinate this allocation so devices that work in multiple countries can be manufactured. However, countries are not bound by ITU-R's recommendations, and the FCC (Federal Communication Commission), which does

the allocation for the United States, has occasionally rejected ITU-R's recommendations (usually because they required some politically-powerful group giving up some piece of the spectrum).

Even when a piece of spectrum has been allocated to some use, such as mobile phones, there is the additional issue of which carrier is allowed to use which frequencies. Three algorithms were widely used in the past. The oldest algorithm, often called the beauty contest, requires each carrier to explain why its proposal serves the public interest best. Government officials then decide which of the nice stories they enjoy most. Having some government official award property worth billions of dollars to his favorite company often leads to bribery, corruption, nepotism, and worse. Furthermore, even a scrupulously honest government official who thought that a foreign company could do a better job than any of the national companies would have a lot of explaining to do.

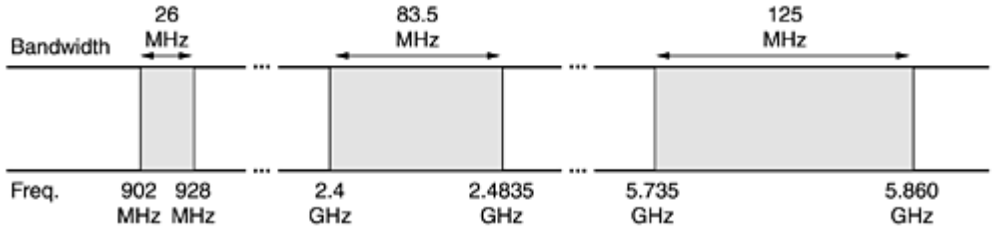
This observation led to algorithm 2, holding a lottery among the interested companies. The problem with that idea is that companies with no interest in using the spectrum can enter the lottery. If, say, a fast food restaurant or shoe store chain wins, it can resell the spectrum to a carrier at a huge profit and with no risk.

Bestowing huge windfalls on alert, but otherwise random, companies has been severely criticized by many, which led to algorithm 3: auctioning off the bandwidth to the highest bidder. When England auctioned off the frequencies needed for third-generation mobile systems in 2000, they expected to get about $4 billion. They actually received about $40 billion because the carriers got into a feeding frenzy, scared to death of missing the mobile boat. This event switched on nearby governments' greedy bits and inspired them to hold their own auctions. It worked, but it also left some of the carriers with so much debt that they are close to bankruptcy. Even in the best cases, it will take many years to recoup the licensing fee.

A completely different approach to allocating frequencies is to not allocate them at all. Just let everyone transmit at will but regulate the power used so that stations have such a short range they do not interfere with each other. Accordingly, most governments have set aside some frequency bands, called the ISM (Industrial, Scientific, Medical) bands for unlicensed usage. Garage door openers, cordless phones, radio-controlled toys, wireless mice, and numerous other wireless household devices use the ISM bands. To minimize interference between these uncoordinated devices, the FCC mandates that all devices in the ISM bands use spread spectrum techniques. Similar rules apply in other countries

The location of the ISM bands varies somewhat from country to country. In the United States, for example, devices whose power is under 1 watt can use the bands shown in Fig. 2-13 without requiring a FCC license. The 900-MHz band works best, but it is crowded and not available worldwide. The 2.4-GHz band is available in most countries, but it is subject to interference from microwave ovens and radar installations. Bluetooth and some of the 802.11 wireless LANs operate in this band. The 5.7-GHz band is new and relatively undeveloped, so equipment for it is expensive, but since 802.11a uses it, it will quickly become more popular.

Figure 2-13. The ISM bands in the United States.



4. Infrared and Millimeter Waves

Unguided infrared and millimeter waves are widely used for short-range communication. The remote controls used on televisions, VCRs, and stereos all use infrared communication. They are relatively directional, cheap, and easy to build but have a major drawback: they do not pass through solid objects (try standing between your remote control and your television and see if it still works). In general, as we go from long-wave radio toward visible light, the waves behave more and more like light and less and less like radio.

On the other hand, the fact that infrared waves do not pass through solid walls well is also a plus. It means that an infrared system in one room of a building will not interfere with a similar system in adjacent rooms or buildings: you cannot control your neighbor's television with your remote control. Furthermore, security of infrared systems against eavesdropping is better than that of radio systems precisely for this reason. Therefore, no government license is needed to operate an infrared system, in contrast to radio systems, which must be licensed outside the ISM bands. Infrared communication has a limited use on the desktop, for example, connecting notebook computers and printers, but it is not a major player in the communication game.

5. Lightwave Transmission

Unguided optical signaling has been in use for centuries. Paul Revere used binary optical signaling from the Old North Church just prior to his famous ride. A more modern application is to connect the LANs in two buildings via lasers mounted on their rooftops. Coherent optical signaling using lasers is inherently unidirectional, so each building needs its own laser and its own photodetector. This scheme offers very high bandwidth and very low cost. It is also relatively easy to install and, unlike microwave, does not require an FCC license.

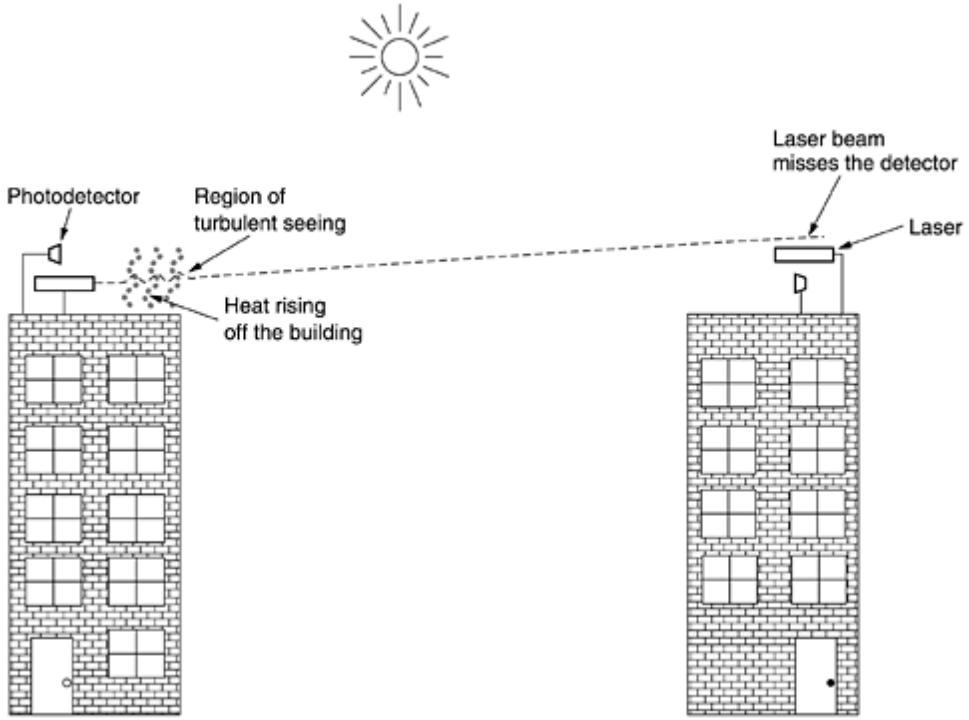
The laser's strength, a very narrow beam, is also its weakness here. Aiming a laser beam 1-mm wide at a target the size of a pin head 500 meters away requires the marksmanship of a latter-day Annie Oakley. Usually, lenses are put into the system to defocus the beam slightly.

A disadvantage is that laser beams cannot penetrate rain or thick fog, but they normally work well on sunny days. However, the author once attended a conference at a modern hotel in Europe at which the conference organizers thoughtfully provided a room full of terminals for the attendees to read their e-mail during boring presentations. Since the local PTT was unwilling to install a large number of telephone lines for just 3 days, the organizers put a laser on the roof and aimed it at their university's computer science building a few kilometers away. They tested it the night before the conference and it worked perfectly. At 9 a.m. the next morning, on a bright sunny day, the link failed completely and stayed down all day. That evening, the organizers tested it again very carefully, and once again it worked absolutely perfectly. The pattern repeated itself for two more days consistently.

After the conference, the organizers discovered the problem. Heat from the sun during the daytime caused convection currents to rise up from the roof of the building, as shown in Fig. 2-14. This

turbulent air diverted the beam and made it dance around the detector. Atmospheric "seeing" like this makes the stars twinkle (which is why astronomers put their telescopes on the tops of mountains—to get above as much of the atmosphere as possible). It is also responsible for shimmering roads on a hot day and the wavy images seen when one looks out above a hot radiator.

Figure 2-14. Convection currents can interfere with laser communication systems. A bidirectional system with two lasers is pictured here.



8.Explain Baseband Transmission in detail.

Baseband transmission is transmission of the encoded signal using its own baseband frequencies i.e. without any shift to higher frequency ranges. It is used for short distances.

Steps in Baseband Transmission

Let us understand the baseband transmission step by step.

Step 1 − The most straightforward form of digital modulation is to use a positive voltage which represents ‘1’ and negative voltage represents ‘0’.

Step 2 − For an optical signal the presence of light may represent ‘1’ and absence of light represent ‘0’. This scheme is called NRZ (Non-Return-to-Zero).

Step 3 − Once sent, the NRZ signal propagates down the wire.

Step 4 − At the other end, the receiver converts it into bits by sampling the signal at regular intervals of time.

Step 5 − This signal will not look exactly like the signal that was sent.

Step 6 − It will be attenuated and distorted by the channel and noise at the receiver.

Step 7 − To decode the bits, the receiver maps the signal samples to the closest symbols.

Step 8 − For NRZ, a positive voltage will be taken to indicate that ‘1’ was sent and negative voltage will be taken to indicate that ‘0’ was sent.

Step 9 − NRZ is a good starting point for studies because it is simple.

Step 10 − In practice, it is rarely used by itself.

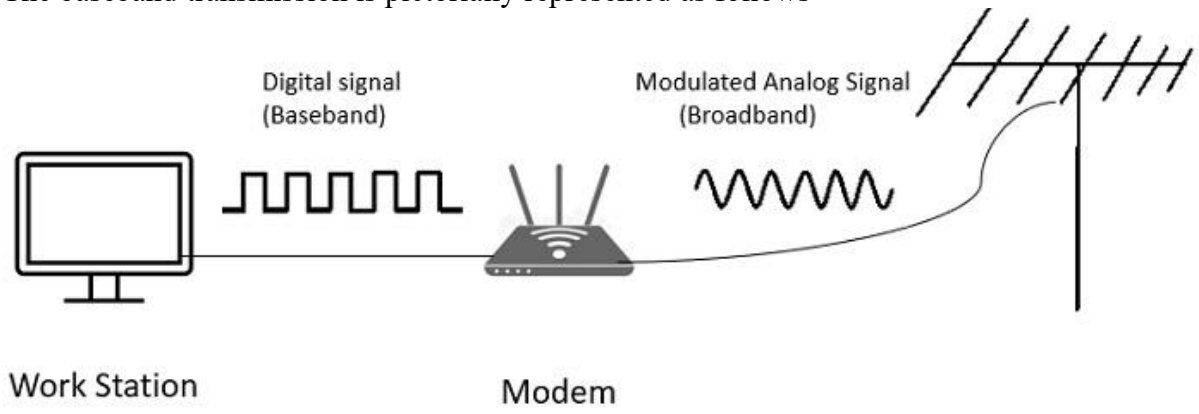
Step 11 − More complex schemes can convert bits to signals that better meet engineering considerations.

Step 12 − These schemes are called line codes.

Step 13 − Line codes help with −

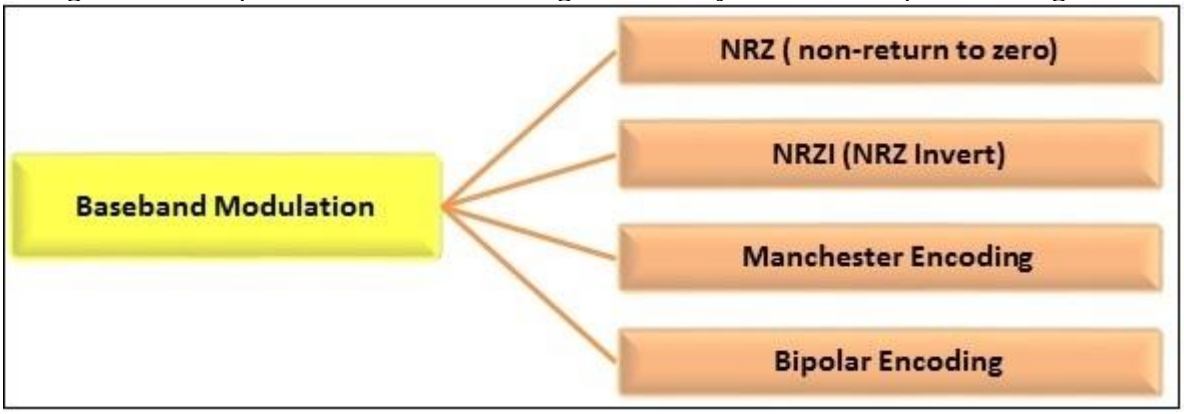
- Bandwidth efficiency.
- Clock recovery.
- DC balance.

The baseband transmission is pictorially represented as follows −



In baseband transmission, the data bits are directly converted into signals. Generally a higher voltage level represents the bit 1, while a lower voltage level represents bit 0.

The different encoding schemes are shown in the diagram. Among these, the first three are come in the category of polar encoding. In polar signaling, one logical state is represented by only one voltage state. In bipolar schemes, two voltage levels may be used to represent a logical state.



NRZ (Non – Return to Zero)

NRZ is an unipolar coding scheme. Here, a high voltage represents 1, while a low voltage represents 0. Non-return to zero implies that the signal does not return to zero at the middle of the bit.

NRZ-I (NRZ Invert)

NRZ-I is an polar coding scheme. In NRZI, bit 1 is represented by a transition in voltage, while bit 0 is represented by no such transitions. It has an average signal rate of N/2 baud.

Manchester Encoding

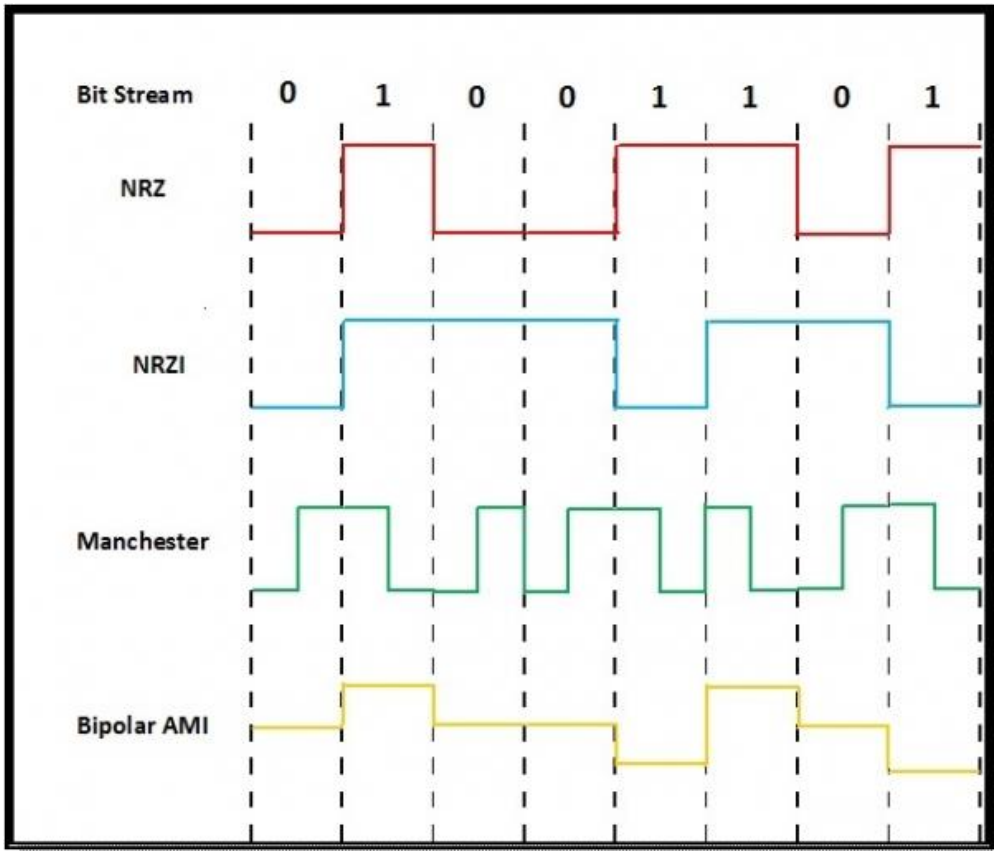
Manchester encoding is a biphase coding scheme. Bit 1 is represented by a voltage transition from high to low, while bit 0 is represented by a voltage transition from low to high.

Bipolar Encoding

It is also called Alternate Mark Inversion or AMI. Three voltage levels are used here. Here, bit 0 is represented by no line signal, while bit 1 is represented by a positive or negative voltage level, alternating for successive ones.

Example

Let there be a bit stream 01001101. The following diagram plots the different encoding schemes –



9.Explain Digital Modulation in detail.

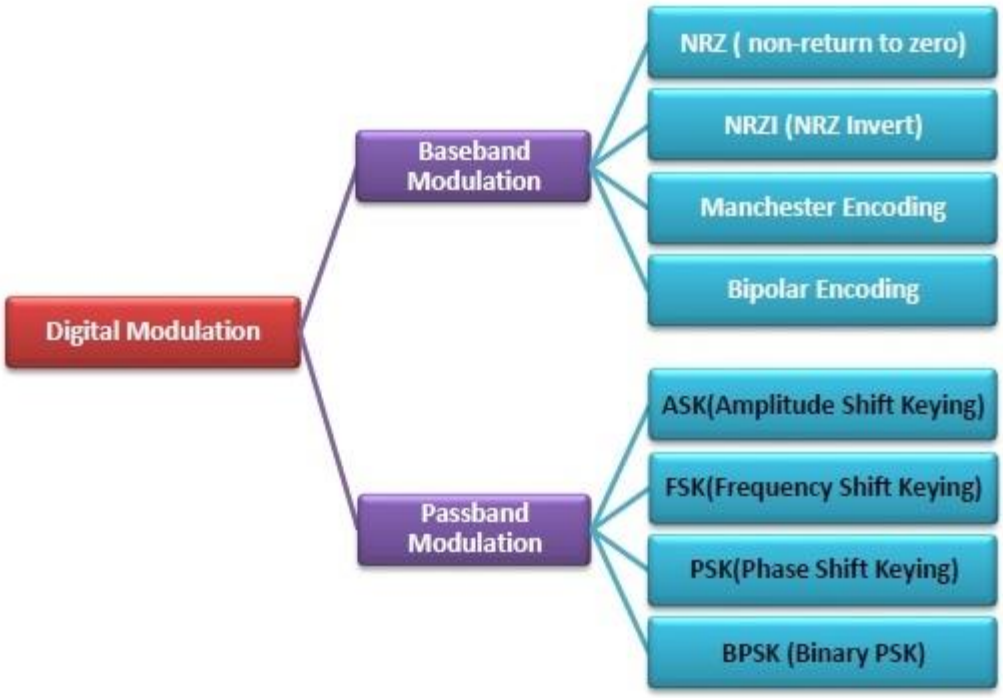
Digital Modulation

Digital modulation is the process of converting a digital bit stream into an analog carrier wave for transmission via a communication channel.

Digital modulation is broadly divided into two categories −

- **Bandpass Modulation as in baseband transmission:**
  Here, the bits are converted directly into signals.
- **Passband Modulation as in passband transmission:**
  Here, the amplitude, phase or frequency of the carrier signal is regulated to transmit the bits.

The following diagram illustrates the different digital modulation schemes −



Write baseband transmission also

UNIVERSITY QUESTIONS:

2 marks:

1. What are the functions of transport layer in OSI reference model?
2. What are the standards used in wireless communication?

11 marks:

1. Discuss different types of transmission media used in networks.
2. Write brief note on transmission medium
3. Compare internet and OSI reference model and list the advantages of each.
4. Briefly discuss the theoretical basis behind communication system.
5. Explain Baseband Transmission

UNIT II

Data link layer – design issues – Services - Framing - Error Control - Flow Control - Error detection and correction codes - data link layer protocols -Simplex Protocol – Sliding window Protocols - Medium Access control sublayer – Channel allocation problem – Multiple Access protocols – ALOHA – CSMA Protocols - Collision-Free Protocols - Limited-Contention Protocols - Wireless LANs - 802.11 Architecture - 802.16 Architecture – Data link layer Switching - Uses of Bridges - Learning Bridges - Spanning Tree Bridges - Repeaters, Hubs, Bridges, Switches, Routers, and Gateways - Virtual LANs

2 MARKS

1. What are the responsibilities of data link layer?

Specific responsibilities of data link layer include the following.

- Framing
- Physical addressing
- Flow control
- Error control
- Access control

2. Mention the types of errors.

There are 2 types of errors

- Single-bit error.
- Burst-bit error.

3. Define the following terms.

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

b) Burst error: Means that 2 or more bits in the data unit have changed from 1 to 0 from 0 to 1.

4. List out the available detection methods.

There are 4 types of redundancy checks are used in data communication.

a) Vertical redundancy checks (VRC).

b) Longitudinal redundancy checks (LRC).

c) Cyclic redundancy checks (CRC).

d) Checksum.

5. What is redundancy?

It is the error detecting mechanism, which means a shorter group of bits or extra bits may be app ended at the destination of each unit.

6. Write short notes on VRC.

The most common and least expensive mechanism for error detection is the vertical redundancy check (VRC) often called a parity check. In this technique a redundant bit called a parity bit, is appended to every data unit so, that the total number of 0’s in the unit (including the parity bit) becomes even.

7. Write short notes on LRC.

In longitudinal redundancy check (LRC), a block of bits is divided into rows and a redundant row of bits is added to the whole block.

8. State the purpose of CRC code?(NOV 2012)

A CRC-enabled device calculates a short, fixed-length binary sequence, known as the *CRC code*, for each block of data and sends or stores them both together. When a block is read or received the device repeats the calculation; if the new CRC does not match the one calculated earlier, then the block contains a data error and the device may take corrective action such as rereading or requesting the block be sent again, otherwise the data is assumed to be error free**Write short notes on CRC generator.**

A CRC generator uses a modulo-2 division.

a) In the first step, the 4 bit divisor is subtracted from the first 4 bit of the dividend.

b) Each bit of the divisor is subtracted from the corresponding

c) bit of the dividend without disturbing the next higher bit.

10. Write short notes on CRC checker.

A CRC checker functions exactly like a generator. After receiving the data appended with the CRC it does the same modulo-2 division. If the remainder is all 0's the CRC is dropped and the data accepted. Otherwise, the received stream of bits are discarded and the dates are resent.

11. Define checksum.

The error detection method used by the higher layer protocol is called checksum. Checksum is based on the concept of redundancy.

12. What are the steps followed in checksum generator?

The sender follows these steps

- The units are divided into k sections each of n bits.
- All sections are added together using 2's complement to get the sum.
- The sum is complemented and become the checksum.
- The checksum is sent with the data.

13. List out the steps followed is checksum checker side.

The receiver must follow these steps

a) The unit is divided into k section each of n bits.
b) All sections are added together using 1's complement to get the sum.
c) The sum is complemented.
d) If the result is zero.

14. Write short notes on error correction.

It is the mechanism to correct the errors and it can be handled in 2 ways.

a) When an error is discovered, the receiver can have the sender retransmit the entire data unit.
b) A receiver can use an error correcting coder, which automatically corrects certain errors.

15. Mention the types of error correcting methods.

There are 2 error-correcting methods.

a) Single bit error correction
b) Burst error correction.

16. What is the purpose of hamming code?

A hamming code can be designed to correct burst errors of certain lengths. So the simple strategy used by the hamming code to correct single bit errors must be redesigned to be applicable for multiple bit correction

17. Compare Error Detection and Error Correction:

The correction of errors is more difficult than the detection. In error detection, checks only any error has occurred. In error correction, the exact number of bits that are corrupted and location in the message are known. The number of the errors and the size of the message are important factors.

18. What is Forward Error Correction:

Forward error correction is the process in which the receiver tries to guess the message by using redundant bits.

19. Define Retransmission:

Retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. Resending is repeated until a message arrives that the receiver believes is error-freed.

20. What are Data Words?

In block coding, we divide our message into blocks, each of k bits, called datawords. The block coding process is one-to-one. The same dataword is always encoded as the same codeword.

21. What are Code Words?

"r" redundant bits are added to each block to make the length n = k + r. The resulting n-bit blocks are called codewords. 2n – 2k codewords that are not used. These codewords are invalid or illegal.

22. What is a Linear Block Code?

A linear block code is a code in which the exclusive OR (addition modulo-2) of two valid codeword's creates another valid codeword.

23. What are Cyclic Codes?

Cyclic codes are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.

24. Define Encoder:

A device or program that uses predefined algorithms to encode, or compress audio or video data for storage or transmission use. A circuit that is used to convert between digital video and analog video.

25. Define Decoder

A device or program that translates encoded data into its original format (e.g., it decodes the data). The term is often used in reference to MPEG-2 video and sound data, which must be decoded before it is output.

26.What is framing?

Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address. The destination address defines where the packet has to go and the sender address helps the recipient acknowledge the receipt.

27. What is Fixed –Size Framing?

In fixed-size framing, there is no need for defining the boundaries of the frames. The size itself can be used as a delimiter.

28. What is Bit Stuffing?

Bit stuffing is the process of adding one extra 0 whenever five consecutive Is follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

29. Define Character Stuffing.

In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.

30. What is Flow Control?

Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

31. What is Error Control?

Error control is both error detection and error correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender. In the data link layer, the term error control refers primarily to methods of error detection and retransmission.

32. What Automatic Repeat Request (ARQ)?

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

33. What is Stop-and-Wait Protocol?

In Stop and wait protocol, sender sends one frame, waits until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame

34. What is Stop-and-Wait Automatic Repeat Request?

Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires.

35. What is usage of Sequence Number in Reliable Transmission?

The protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame. Since we want to minimize the frame size, the smallest range that provides unambiguous communication. The sequence numbers can wrap around.

36.What is Pipelining?

In networking and in other areas, a task is often begun before the previous task has ended. This is known as pipelining.

37. What is Sliding Window?

The sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver. In other words, he sender and receiver need to deal with only part of the possible sequence numbers.

38. What is Piggy Backing?

A technique called piggybacking is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

39. What is frame bursting?(NOV 2011)

Frame-bursting is a communication protocol feature used at the link layer in communication networks to alter the transmission characteristics in order to benefit from higher data transfer throughput.

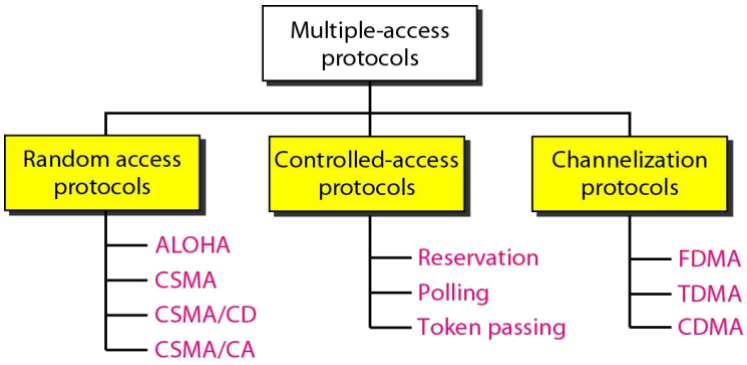
40. Define FDDI? (NOV 2011)

FDDI (Fiber Distributed Data Interface) is a set of ANSI and ISO standards for data transmission on fiber optic lines in a local area network (LAN) that can extend in range up to 200 km (124 miles). The FDDI protocol is based on the token ring protocol. In addition to being large geographically, an FDDI local area network can support thousands of users. FDDI is frequently used on the backbone for a wide area network (WAN).

41. What are adaptive algorithms?(APR 2011)

An **adaptive algorithm** is an algorithm that changes its behavior based on information available at the time it is run. This might be information about computational resources available, or the history of data recently received

42. Name the Multiple Access Protocols



43. Define ARP

ARP stands for Address resolution protocol, maps an IP address to a MAC address

44. What do you mean by RARP?

RARP stands for Reverse Address resolution protocol, maps an MAC address to a IP address

45.Define DHCP

The Dynamic Host Configuration Protocol has been derived to provide dynamic configuration. DHCP is also needed when a host moves from network to network or is connected and disconnected from a network.

46. What is Ethernet

A system for connecting a number of computer systems to form a local area network, with protocols to control the passing of information and to avoid simultaneous transmission by two or more systems

47. List out various internetworking devices

- Gateway
- Routers
- Switches
- Access points
- Repeaters
- Hubs
- Bridges

48. Define Hubs

A common connection point for devices in a network. Hubs are commonly used to connect segments of a LAN. A hub contains multiple ports. When a packet arrives at one port, it is copied to the other ports so that all segments of the LAN can see all packets

49. Define Bridges

A **bridge** device filters data traffic at a network boundary. Bridges reduce the amount of traffic on a local area network (LAN) by dividing it into two segments. Bridges operate at the data link layer (Layer 2) of the OSI model. Bridges inspect incoming traffic and decide whether to forward or discard it.

50. Define Switch and its uses

A **network switch** is a small hardware device that joins multiple computers together within one local area network (LAN).Ethernet switch devices were commonly used on home networks before home routers became popular; broadband routers integrate Ethernet switches directly into the unit as one of their many functions. High-performance network switches are still widely used in corporate networks and data centers.

51. What is the use of repeater?

A repeater is used to amplify signals carried by a network. The function of a repeater is to receive incoming signals or a packet of data, regenerate the signals to their original strength and retransmit them. When a repeater amplifies the electric signals in a network, they allow transmissions to travel a greater distance. For a repeater to work, both network segments must be identical.

52. What are the problems overcome by bridge when compared with hub?(NOV 2012)

The biggest problem with hubs is their simplicity. Since every packet is sent out to every computer on the network, there is a lot of wasted transmission. This means that the network can easily become bogged down. A bridge goes one step up on a hub in that it looks at the destination of the packet before sending. If the destination address is not on the other side of the bridge it will not transmit the data

53. 53. Define PPP

PPP (Point-to-Point Protocol) is a protocol for communication between two computers using a serial interface, typically a personal computer connected by phone line to a server

54. What is ATM Network

The **Asynchronous Transfer Mode (ATM)** protocol architecture is designed to support the transfer of data with a range of guarantees for quality of service. The user data is divided into small, fixed-length packets, called cells, and transported over virtual connections. ATM operates over high data rate physical circuits, and the simple structure of ATM cells allows switching to be performed in hardware, which improves the speed and efficiency of ATM switches.

55. Define ATM adaptation layer

The basic function of the ATM adaptation layer is to convert the user data supplied by higher layers into 48-byte blocks of data. The ATM adaptation layer is divided into two sub-layers –

- The convergence sub-layer, and
- The segmentation and re-assembly sub-layer

56. What is convergence layer?

The convergence sub-layer provides services to higher layers through a set of protocols

57. What is segmentation and re-assembly sub-layer?

The segmentation and re-assembly sub-layer separates the messages from the convergence sub-layer into ATM cells.

58. Define MPLS

Multiprotocol Label Switching (**MPLS**) is a mechanism in high-performance telecommunications networks that directs data from one network node to the next based on short path labels rather than long network addresses, avoiding complex lookups in a routing table. The labels identify virtual links (*paths*) between distant nodes rather than endpoints. MPLS can encapsulate packets of various network protocols

59. What is Ring Topology

A **ring network** is a network topology in which each node connects to exactly two other nodes, forming a single continuous pathway for signals through each node - a ring. Data travel from node to node, with each node along the way handling every packet.

60. Define Physical Ring

Devices are attached via a series of point-to-point links that form a closed loop. In most physical ring topologies, the links were typically simplex, resulting in transmissions that always moved in one direction around the ring. Each device took the signal it received on its input link and repeated the signal to its output link.

61. Define Logical Ring

Logical topology, or **signal topology**, is the arrangement of devices on a computernetwork and how they communicate with one another. How devices are connected to the network through the actual cables that transmit data, or the physical structure of the network, is called the physical topology. Physical topology defines how the systems are physically connected. It represents the physical layout of the devices on the network. The logical topology defines how the systems communicate across the physical topologies.

11 MARKS

1.DescribeDatalink layer Design Issues.

The Data Link Layer is the second layer in the OSI model, above the Physical Layer, which ensures that the error free data is transferred between the adjacent nodes in the network. It breaks the datagram passed down by above layers and converts them into frames ready for transfer. This is called **Framing.**

It provides two main functionalities

- Reliable data transfer service between two peer network layers

- Flow Control mechanism which regulates the flow of frames such that data congestion is not there at slow receivers due to fast senders.

FUNCTIONS

- Providing a well-defined service interface to the network layer.
- Dealing with transmission errors.
- Regulating the flow of data so that slow receivers are not swamped by fast senders– flow control.

The two main functions of the data link layer are:

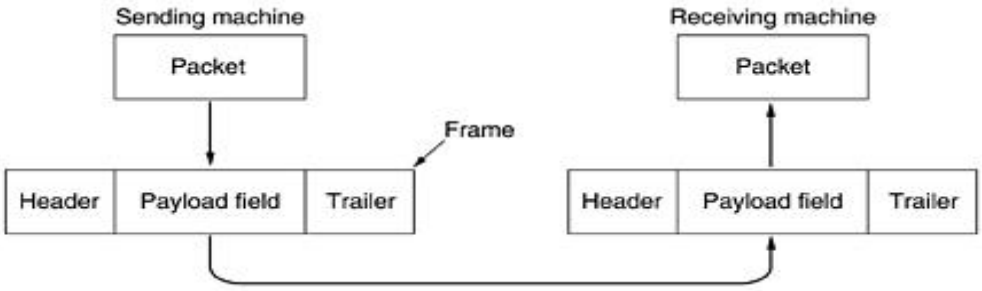
1. **Data Link Control (DLC)**: It deals with the design and procedures for communication b/w nodes: node-to-node communication.
2. **Media Access Control (MAC)**:It explains how to share the link.

1. DATA LINK CONTROL (DLC):

The frame contains

1. Frame header
2. Payload field for holding packet
3. Frame trailer

Figure 1.1 Relationships between Packets and Frames



Data link control functions includes

(1) Services

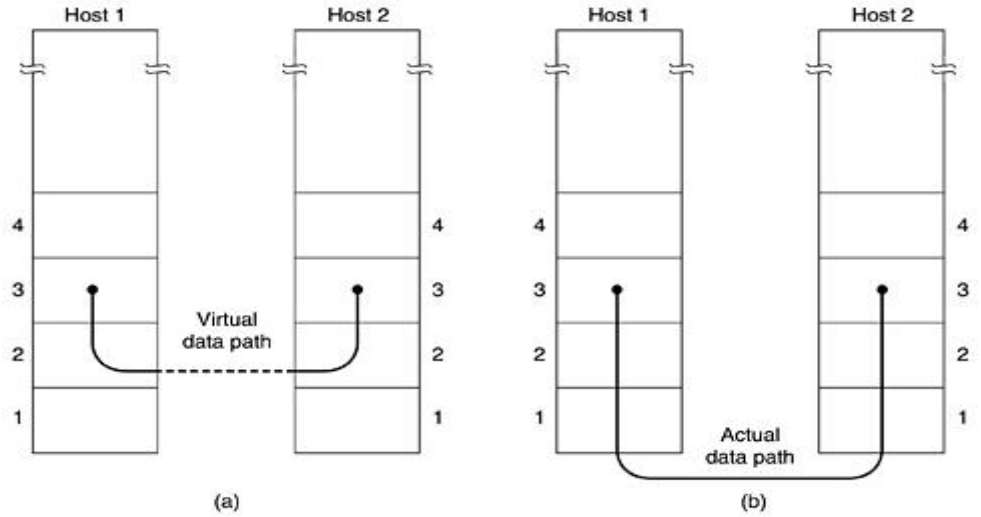
(2) Framing.

(3) Error Control.

(4) FlowControl.

SERVICES PROVIDED TO NETWORK LAYER:

Figure 1.2 (a) Virtual communication. (b) Actual communication.



Transferring data from the network layer on the source machine to the network layer on the destination machine. The data link layer can be designed to offer various services. The actual services offered can vary from system to system. Three reasonable possibilities that are commonly provided are

1. **Unacknowledged connectionless service**

   - Source machine sends independent frames to destination machine having destination machine acknowledge them
   - No logical connection
   - Used when error rate is very low
   - Good for real-time traffic (voice)

2. **Acknowledged connectionless service**

   - No logical connection
   - Each frame sent is individually acknowledged
   - Useful over unreliable channels (i.e. wireless systems)

3. **Acknowledged connection-oriented service**

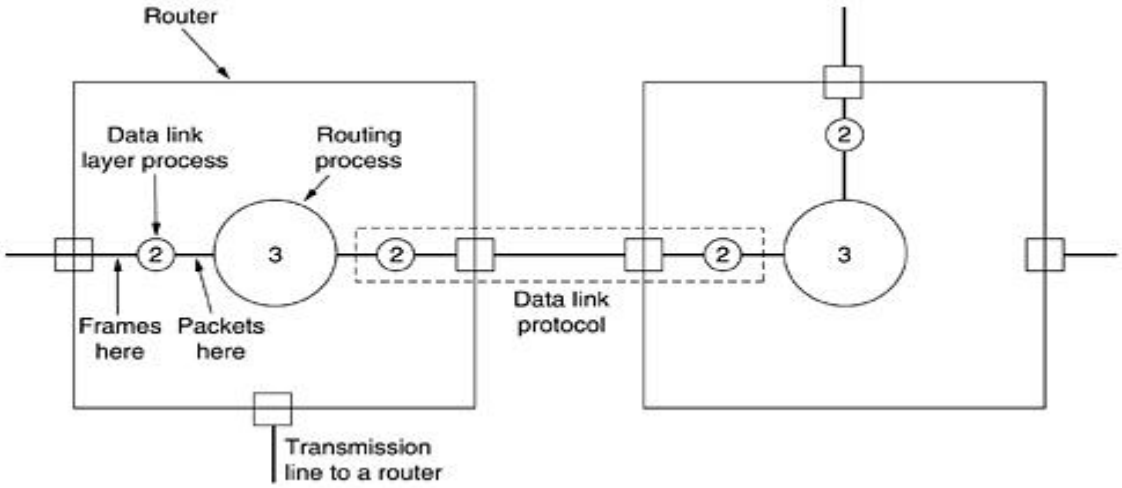
   - Source and destination machines establish a connection before any data are transferred
   - Each frame is numbered
   - DLL guarantees that...
     - Each frame is received
     - Each frame is received exactly once
     - Each frame is received in the right order

3 PHASES

When connection-oriented service is used, transfers go through three distinct phases

1. Connection established
2. Frames are transmitted
3. Connection released

Figure 1.3 Placement of the data link Protocol



- Consider a typical example: a WAN subnet consisting of routers connected by point-to-point leased telephone lines.
- When a frame arrives at a router, the hardware checks it for errors, and then passes the frame to the data link layer software.
- The data link layer software checks to see if this is the frame expected, and if so, gives the packet contained in the payload field to the routing software.
- The routing software then chooses the appropriate outgoing line and passes the packet back down to the data link layer software, which then transmits it. The flow over two routers is shown in**Fig. 1-3.**

FRAMING

Breaking the bit stream up into frames is more difficult than it at first appears. One way to achieve this framing is to insert time gaps between frames, much like the spaces between words in ordinary text. However, networks rarely make any guarantees about timing, so it is possible these gaps might be squeezed out or other gaps might be inserted during transmission.

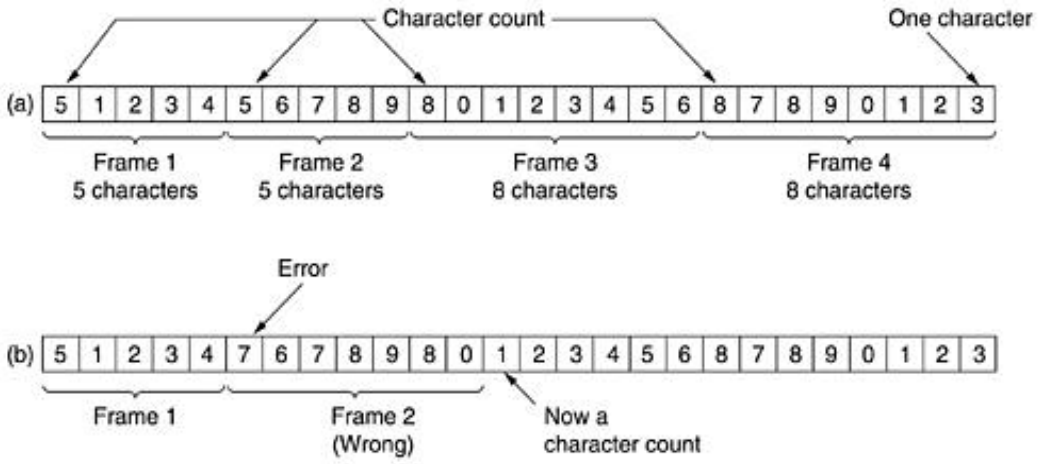
There are four methods:

1. Character count.
2. Flag bytes with byte stuffing.
3. Starting and ending flags, with bit stuffing.
4. Physical layer coding violations.

Character count:

The first framing method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is. This technique is shown in Fig. 3-4(a) for four frames of sizes 5, 5, 8, and 8 characters, respectively.

Figure 3-4. A character stream. (a) Without errors. (b) With one error.



Explanation (Figure 3-4.(a) A character stream Without errors.)

- The first framing method uses a field in the header to specify the number of characters in the frame.
- When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is.
- This technique is shown in Fig. 3-4(a) for four frames of sizes 5, 5, 8, and 8 characters, respectively.
- The trouble with this algorithm is that the count can be garbled by a transmission error.

Explanation (Figure 3-4.(b) A character stream with errors.)

- For example, if the character count of 5 in the second frame of Fig. 3-4(b) becomes a 7, the destination will get out of synchronization and will be unable to locate the start of the next frame.
- Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts.
- Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many characters to skip over to get to the start of the retransmission. For this reason, the character count method is rarely used anymore.

<u>Flag bytes with byte stuffing:</u>

Character-oriented framing approach

- In a character-oriented approach, data to be carried are 8-bit characters.
- The header, which normally carries the source and destination addresses and other control

information.

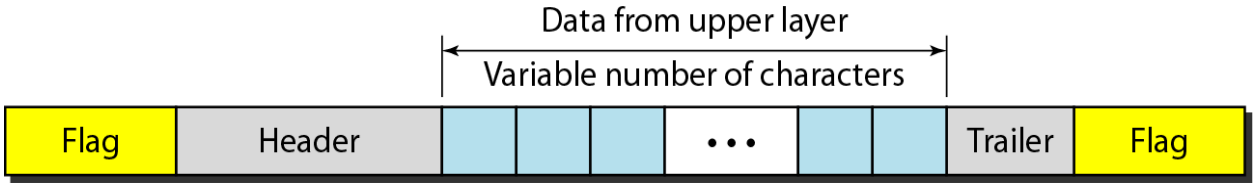
- Trailer carries error detection or error correction redundant bits, are also multiples of 8 bits.
- To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and

the end of a frame.

- The flag, composed of protocol-dependent special characters, signals the start or end of a

frame.

Figure: shows the format of a frame in a character-oriented protocol



Advantage:

1. Simple framing method.
2. Character-oriented framing was popular when only text was exchanged by the dataLink layers.
3. The flag could be selected to be any character not used for text communication.

Disadvantage:

1. Even if with checksum, the receiver knows that the frame is bad there is no way to tell where the next frame starts.
2. Asking for retransmission doesn't help either because the start of the retransmitted frame is not known.
3. Hence No longer used.

Starting and ending character with byte stuffing

Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.

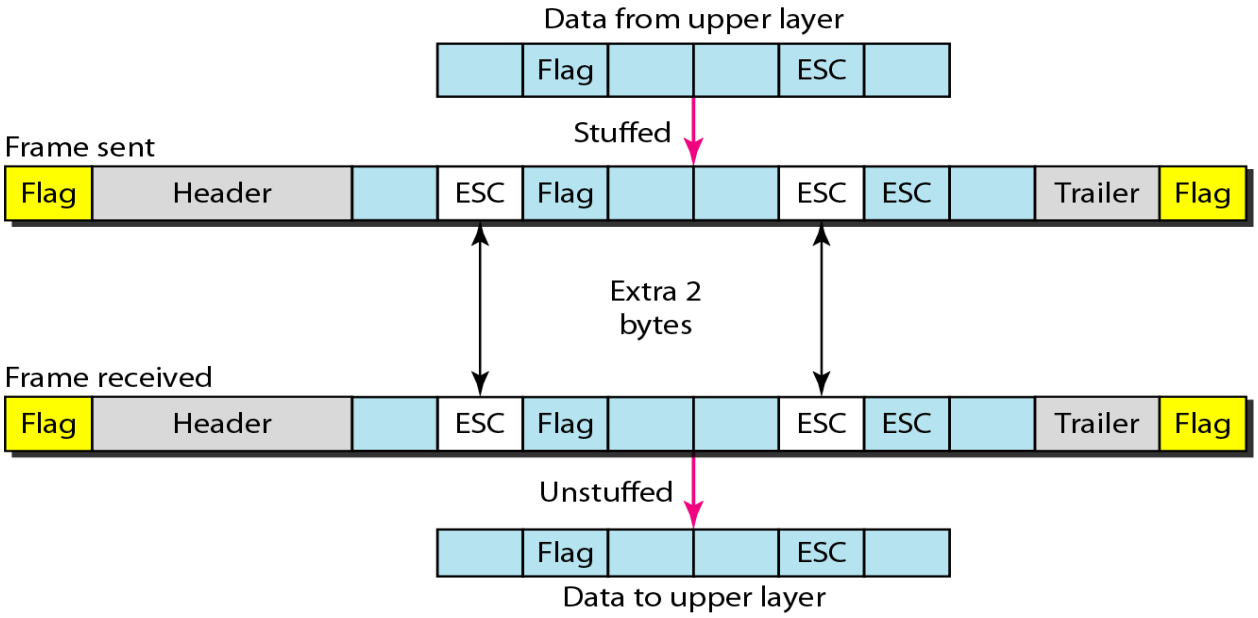


Figure : Byte stuffing and unstuffing

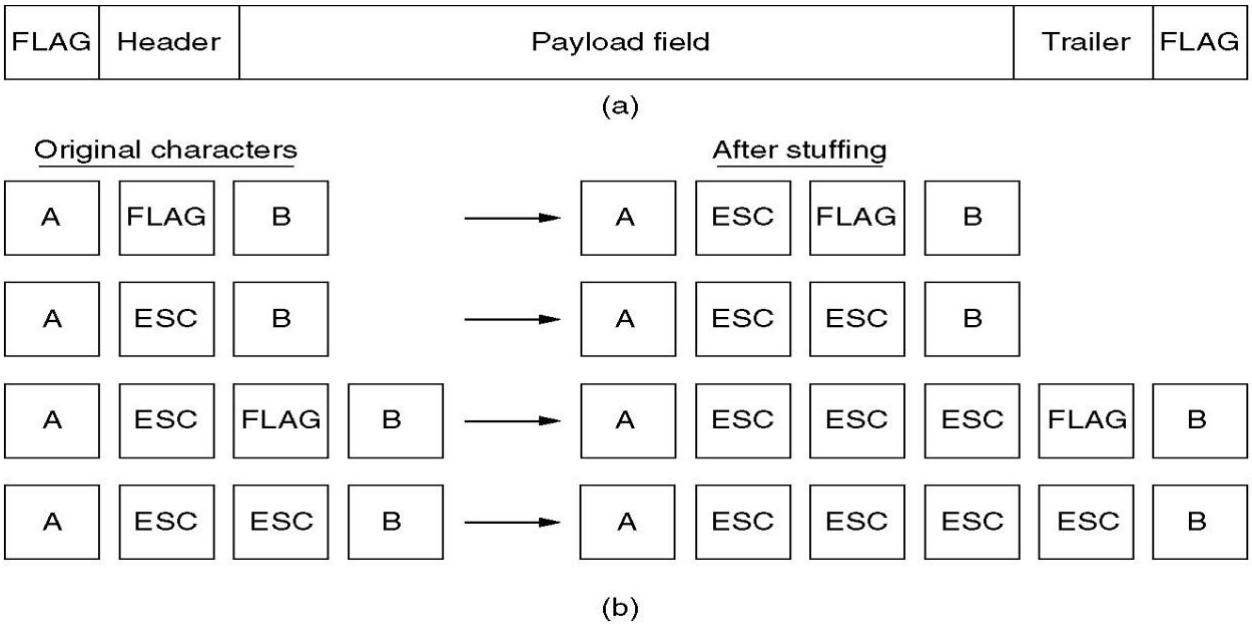


Fig: Framing with byte stuffing

Problem: fixed character size: assumes character size to be 8 bits: can't handle heterogeneousenvironment.

Bit-Oriented framing approach

- Bit stuffing is the process of adding one extra 0 whenever five consecutive 1's follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.
- Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame, as shown in Figure below
- This flag can create the same type of problem. That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame.
- We do this by stuffing 1 single bit (instead of I byte) to prevent the pattern from looking like aflag. The strategy is called bit stuffing.

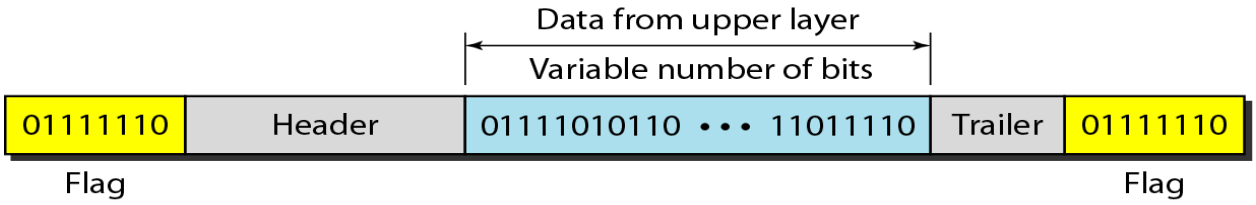


Figure (a)

Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

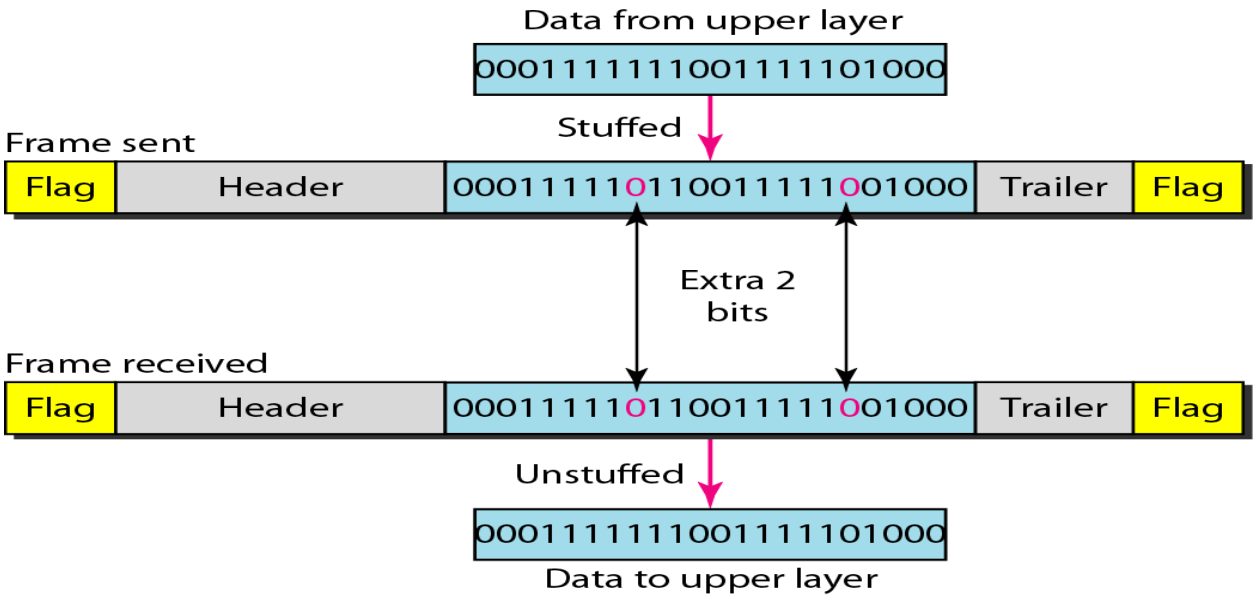


Figure (b)

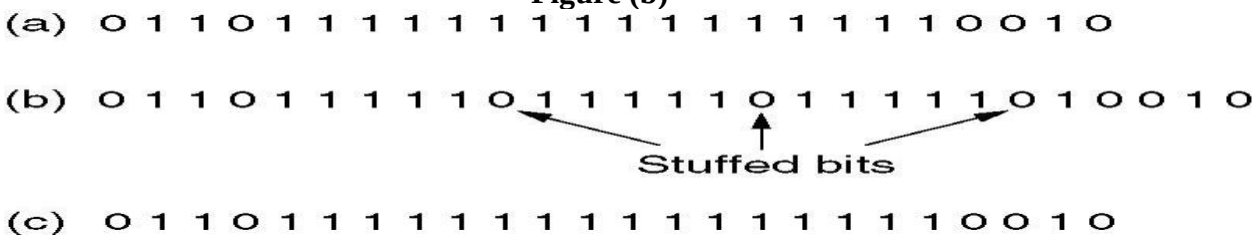


Figure (c)

(a) The original data.

(b) The data as they appear on the line.

(c) The data as they are stored in receiver's memory after destuffing.

Physical layer coding violation:

The last method of framing is only applicable to networks in which the encoding on the physical medium contains some redundancy

. For example, some LANs encode 1 bit of data by using 2 physical bits. Normally, a 1 bit is a high-low pair and a 0 bit is a low-high pair. The scheme means that every data bit has a transition in the middle, making it easy for the receiver to locate the bit boundaries. The combinations high-high and low-low are not used for data but are used for delimiting frames in some protocols.

As a final note on framing, many data link protocols use a combination of a character count with one of the other methods for extra safety. When a frame arrives, the count field is used to locate the end of the frame. Only if the appropriate delimiter is present at that position and the checksum is correct is the frame accepted as valid. Otherwise, the input stream is scanned for the next delimiter.

ERROR CONTROL

- How do we make sure that all frames are eventually delivered to the network layer at the destination and in the proper order?
- Provide sender with some acknowledgement about what is happening with the receiver
- Sender could wait for acknowledgement

Disadvantages

- If a frame vanishes, the receiver will not send an acknowledgement thus, sender will wait forever
- Dealt with by timers and sequence numbers – important part of DLL
- Sender transmits a frame, starts a timer.
- Timer set to expire after interval long enough for frame to reach destination, be processed, and have acknowledgement sent to sender
- Is a danger of frame being transmitted several times, however dealt with by assigning sequence numbers to outgoing frames, so that receiver can distinguish retransmissions from originals.

FLOW CONTROL

What do we do when a sender transmits frames faster than the receiver can accept them?

- **Feedback-based flow control** – receiver sends back information to the sender, giving it permission to send more data or at least telling the sender how the receiver is doing
- **Rate-based flow control** – the protocol has a built-in mechanism that limits the rate at which the sender may transmit data, using feedback from the receiver.

2.Discuss briefly about Error Detection.
ERROR

- When data is being transmitted from one machine to another, it may possible that data become corrupted on its way. Some of the bits may be altered, damaged or lost during transmission. Such a condition is known as **error**.

TYPES OF ERRORS

- **Single bit error**: Only one bit gets corrupted. Common in Parallel transmission.
- **Burst error:** More than one bit gets corrupted very common in serial transmission of data occurs when the duration of noise is longer than the duration of one bit.

Single bit error:

- The term single-bit error means that only one bit of given data unit (such as a byte, character, or data unit) is changed from 1 to 0 or from 0 to 1 as shown in Fig. 3.2.1.
- Single bit errors are least likely type of errors in serial data transmission.
- For example, if 16 wires are used to send all 16 bits of a word at the same time and one of the wires is noisy, one bit is corrupted in each word.

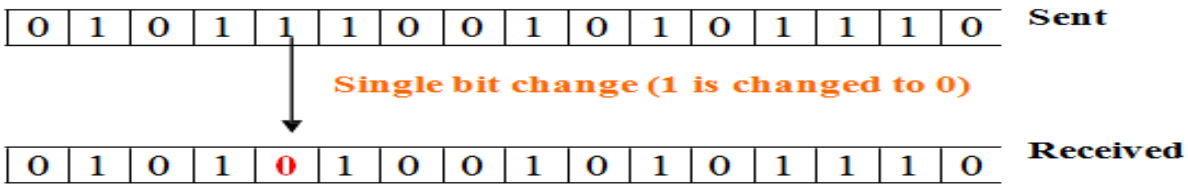


Figure 3.2.1 Single bit error

Burst error:

- More than one bit gets corrupted very common in serial transmission of data occurs when the duration of noise is longer than the duration of one bit.
- The noise affects data; it affects a set of bits.
- The number of bits affected depends on the data rate and duration of noise.

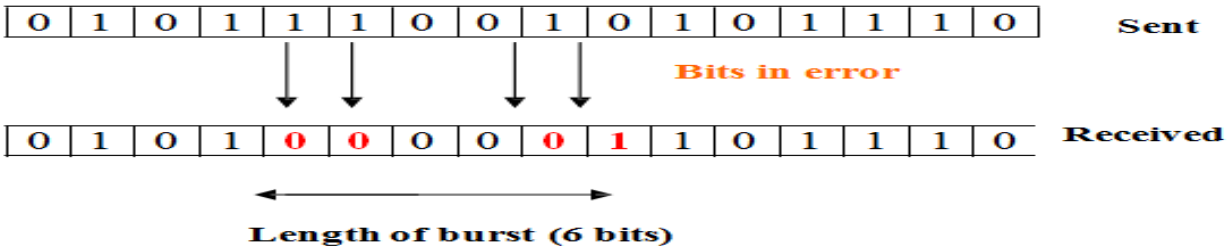


Figure 3.2.2 Burst Error

ERROR DETECTION TECHNIQUES

Basic approach used for error detection is the use of redundancy, where additional bits are added to facilitate detection and correction of errors. Popular techniques are

- Simple Parity check
- Two-dimensional Parity check
- Checksum
- Cyclic redundancy check

Redundancy is the method in which some extra bits are added to the data so as to check whether the data contain error or not.

m - data bits (i.e., message bits)
r - redundant bits (or check bits).
n - total number of bits
n= (m + r).

An n-bit unit containing data and check-bits is often referred to as an n-bit codeword.

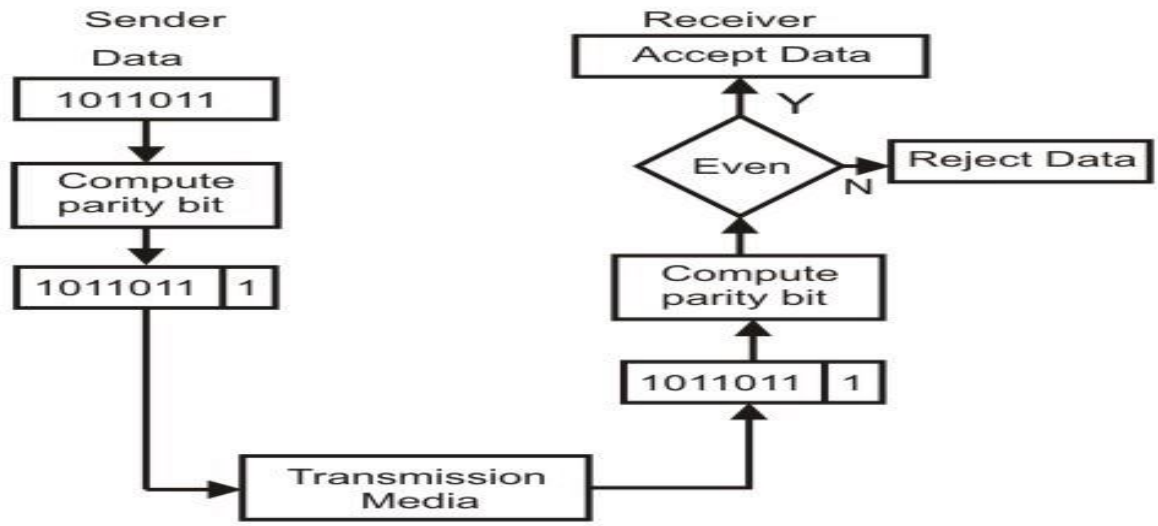
SIMPLE PARITY CHECK

The simplest and most popular error detection scheme. Appends a Parity bit to the end of the data.

Even Parity: **01000001 –** Number of ones in the group of bits is even
Odd Parity: **11000001 -** Number of ones in the group of bits is odd

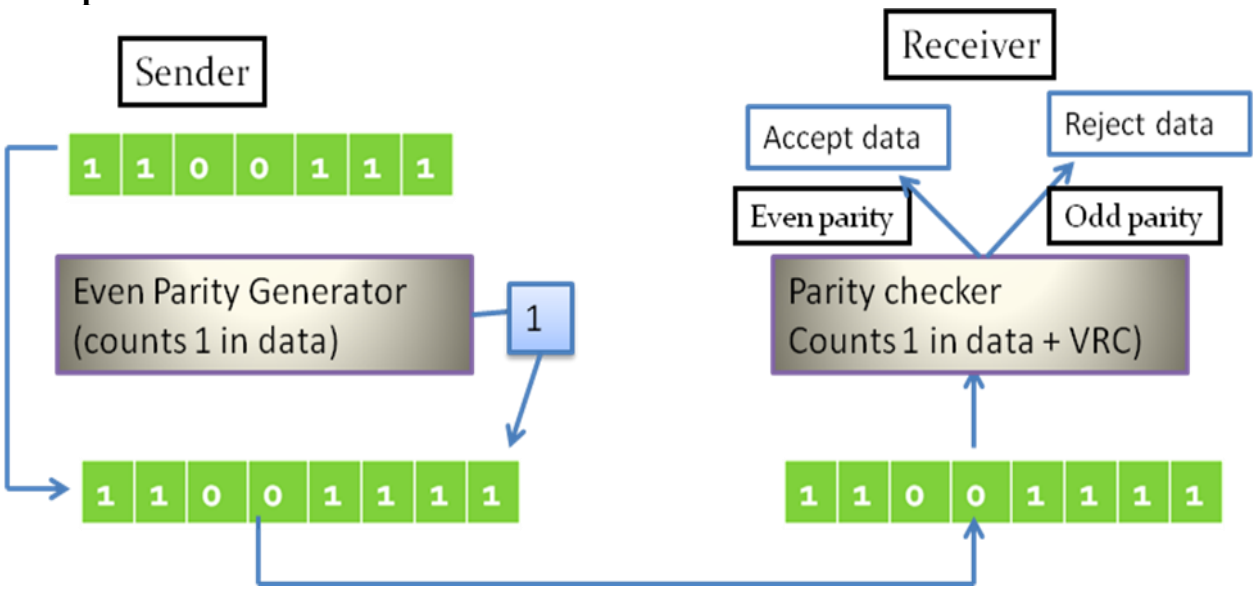
A parity of 1 is added to the block if it contains an odd number of 1's (ON bits) and 0 is added if it contains an even number of 1's.Atthereceivingendtheparitybitis computedfromthereceiveddatabitsandcomparedwiththereceivedparitybit.

Thisschememakesthetotalnumberof1'seven,thatiswhyitiscalled *evenparitychecking*.Consideringa4-bitword,differentcombinationsofthedatawords andthecorrespondingcodewords aregiveninTable3.2.1.



Decimal value	Data Block	Parity bit	Code word
0	0000	0	00000
1	0001	1	00011
2	0010	1	00101
3	0011	0	00110
4	0100	1	01001
5	0101	0	01010
6	0110	0	01100
7	0111	1	01111
8	1000	1	10001
9	1001	0	10010
10	1010	0	10100
11	1011	1	10111
12	1100	0	11000
13	1101	1	11011
14	1110	1	11101
15	1111	0	11110

Example:



PERFORMANCE OF SIMPLE PARITY CHECK

- Simple parity check can **detect all single-bit error**

- It can also detect burst error, if the number of bits in **even or odd.**
- The technique is not foolproof against burst errors that **invert more than one bit**. If an even number of bits is inverted due to error, the **error is not detected.**

TWO-DIMENSION PARITY CHECKING

- Performance can be improved by using two dimensional parity check, which **organizes the block of bits in the form of table.**
- Parity check bits are **calculated from each row**, which is equivalent to a simple parity check.
- Parity check bits are also **calculated for all columns.**
- Both are sent along with the data.
- At the receiving end these are compared with the parity bits calculated on the received data.

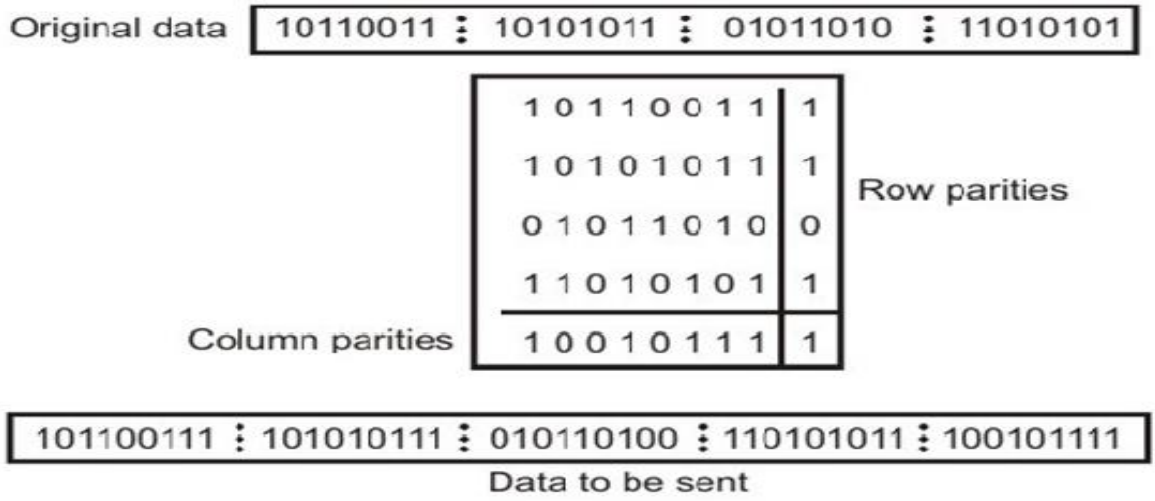


Figure 3.2.4 Two-dimension Parity Checking

Performance:

- If two bits in one data unit are damaged and two bits in exactly same position in another data unit are also damaged, The 2-D Parity check **checker will not detect an error**.
- For example, if two data units: **11001100 and 10101100.**
- If first and second from last bits in each of them is changed, making the data units as **01001110 and 00101110**, the error cannot be detected by 2-D Parity check.

CHECKSUM

- In checksum error detection scheme, the **data is divided into k segments each of m bits.**
- In the sender's end the segments are added using **1's complement arithmetic to get the sum.**
- The sum is complemented to get the checksum. The **checksum** segment is sent **along with the data segments**
-

Example 1:

Sender **Reciever:**

10101001	subunit1
00111001	subunit 2
11100010	sum
00011101	Complement of sum

10101001 subunit1
00111001 subunit2
00011101 Checksum
11111111 sum
00000000
complement
Conclusion = Accept data.

10101001	00111001	00011101
Data		checksum

Example 2: K= 10110011, 10101011, 01011111, 11010101

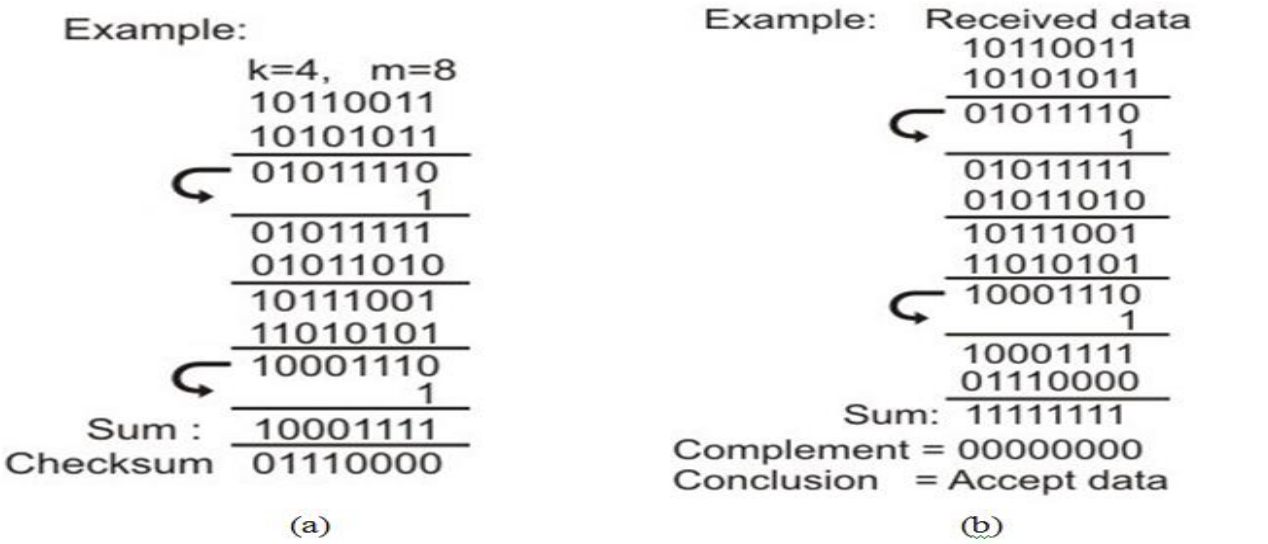


Figure 3.2.5 (a) Sender's end for the calculation of the checksum, (b) Receiving end for checking the checksum

CYCLIC REDUNDANCY CHECK

- One of the most powerful and commonly used error detecting codes.

Basic approach:

- Given a m-bit block of bit sequence, the sender generates an n-bit sequence known as **frame sequence check(FCS),** so that the resulting frame, consisting of m+n bits exactly divisible by **same predetermined number.**
- The receiver divides the incoming frame by that number and, if there is **no reminder, assumes there was no error.**

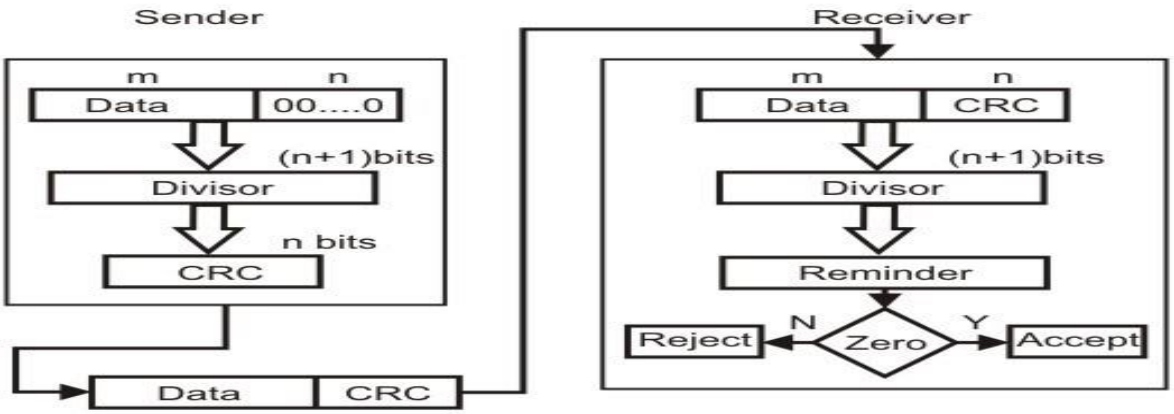


Fig.3.2.7 by dividingasample4-bitnumberbythecoefficientofthegeneratorpolynomialx^3+x+1,whichis1011,using themodulo-2arithmetic.

Modulo-2arithmeticisabinaryadditionprocesswithoutany carryover,whichisjusttheExclusive-OR operation.

Considerthecasewherek=1101. Hencewehavetodivide1101000(i.e.*k*appended by3zeros)by1011,whichproduces theremainder r=001,sothatthebitframe*(k+r)*=1101001isactuallybeingtransmitted throughthecommunicationchannel.

Atthereceivingend,ifthereceivednumber,i.e.,1101001isdividedbythesamegeneratorpolynomial1011togettheremainderas000,it canbeassumedthatthedataisfreeof errors.

Sender: Sender transmit the data along with remainder (**CRC**)

1111 Quotient Data: 1101

1011 | 1101000 ← K Divisor: 1011

1011

1100

1011

1110

1011

1010 Data to be sent: 1 1 0 1 0 0 1

1011 Data CRC

001 Reminder (r)

Receiver:

1111 Quotient Data: 1101001

1011 | 1101001 Divisor: 1011

1011

1100

1011

1110

1011

1011

1011

000 Reminder

Note: Remainder is zero, no error. Receiver can accept the data.

Performance of CRC

- CRC can detect all single-bit errors.
- CRC can detect all double-bit errors(three1's)
- CRC can detect any odd number of errors of less than the degree of the polynomial.
- CRC detects most of the larger burst errors with a high probability.

3.Discuss about Error Correction.

Conceptoferror-correction canbeeasilyunderstoodbyexaminingthesimplestcaseof single-bit errors.Aswehavealreadyseenthatasingle-biterrorcanbedetectedby additionofaparitybitwith thedata,whichneededtobesend.

Asingleadditional bitcandetecterror,butit'snotsufficient enoughtocorrectthaterrortoo.Forcorrecting anerroronehastoknowtheexactposition oferror,i.e.exactlywhichbitisinerror(to locatetheinvalidbits).

In the digital world, error correction can be done in two ways:

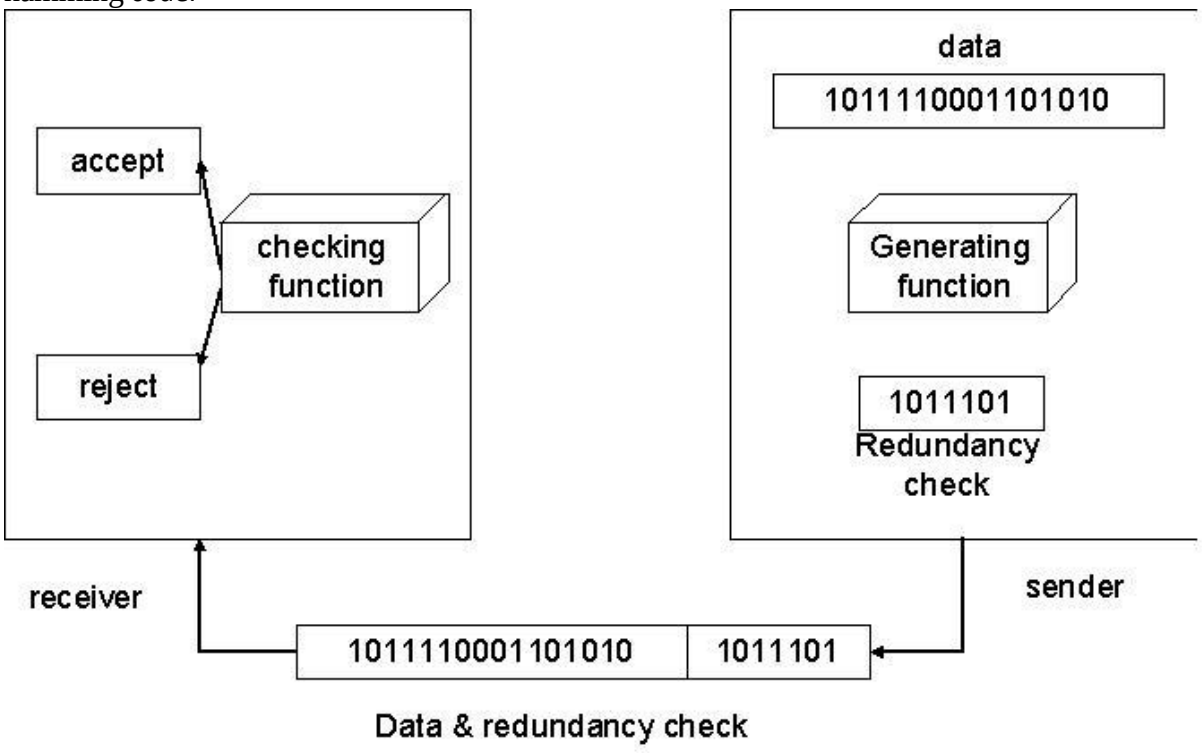
 Backward Error Correction When the receiver detects an error in the data received, it requests back the sender to retransmit the data unit.

 Forward Error Correction When the receiver detects some error in the data received, it executes error-correcting code, which helps it to auto-recover and to correct some kinds of errors.

1. Hamming code
2. Binary convolutional codes.
3. Reed Solomon codes.
4. Low density Parity check codes (LDPC Codes)

HAMMING CODE

Hamming code is a set of error-correction code s that can be used to detect and correct bit errors that can occur when computer data is moved or stored. Hamming code is named for R. W. Hamming of Bell Lab The hamming code can be applied to data units of any length and uses the relationship between data and redundancy bits Positions of redundancy bits in hamming code.



The combinations used to calculate each of the four r values for a seven bit data sequence are as follows:
r1 :1,3,5,7,9,11
r2 : 2,3,6,7,10,11
r3 : 4,5,6,7
r4 : 8,9,10,11
Here, r1 bit is calculated using all bit positions whose binary representation includes a 1 in the rightmost position (0001, 0011, 0101, 0111, 1001, and 1011). The r2 bit is calculated using all bit positions with a 1 in the second position (0010, 0011, 0110, 0111, 1010 and 1011), and for r3 1 at third bit position (0100, 0101, 0110 and 0111) for r4 1 at fourth bit position (1000, 1001, 1010 and 1011).

Calculating the r Values:

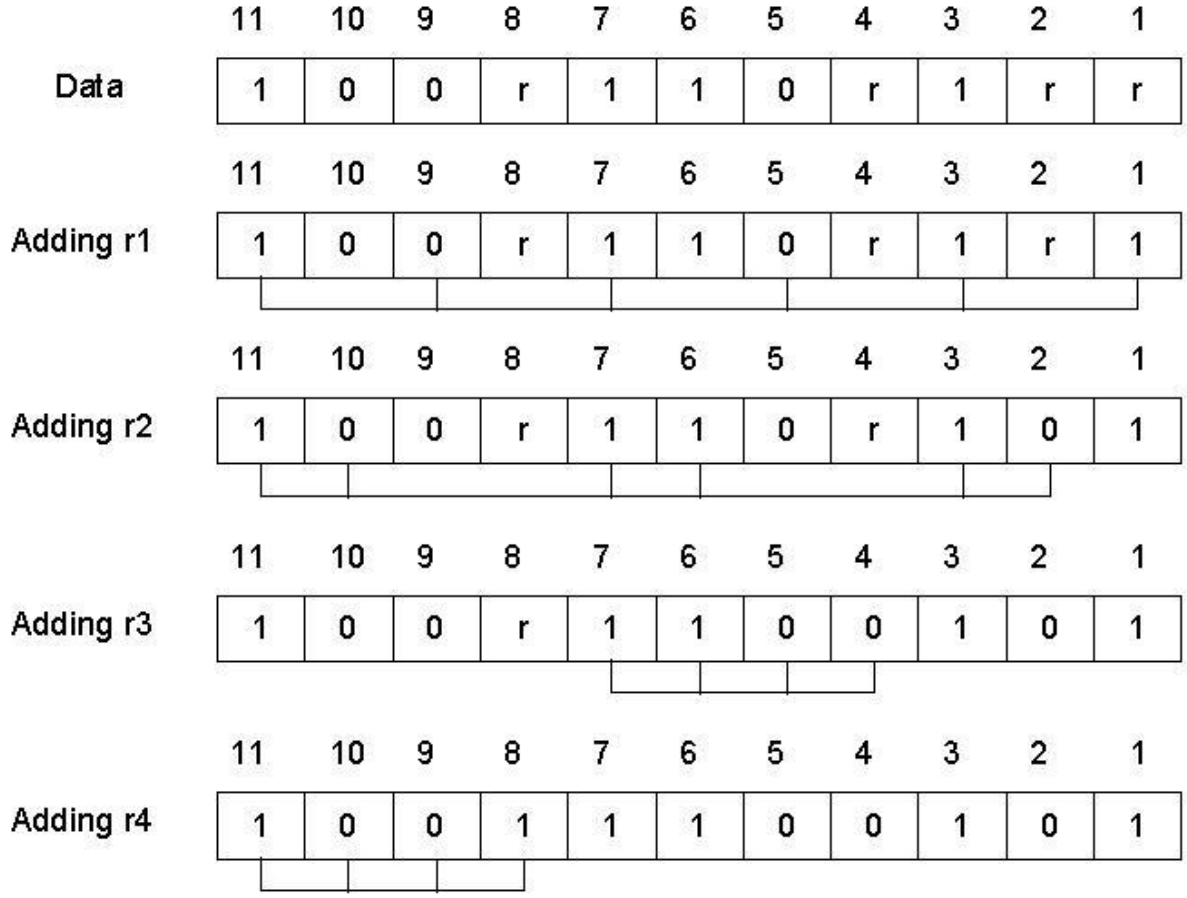
In the first step, we place each bit of the original character in its appropriate positions in the 11 bit unit. Then, we calculate the even parities for the various bit combinations. The parity value of each combination is the value of the corresponding r bit. For example r1 is calculated to provide even parity for a combination of bits 3, 5, 7, 9, 11.

Error Detection and Correction:

Example:

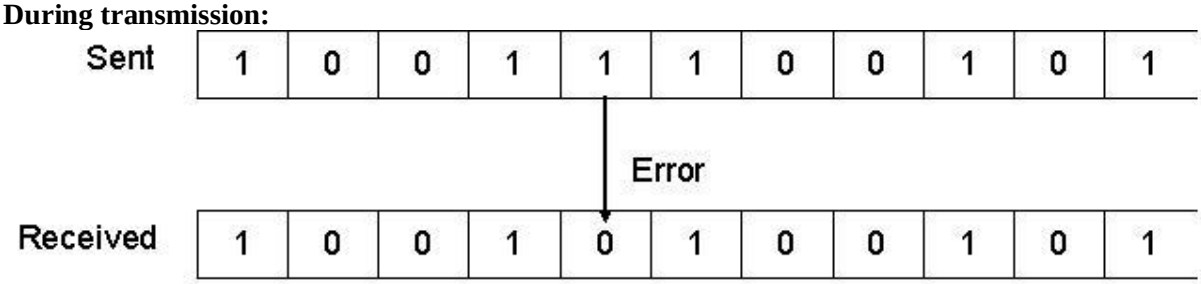
At the sender:

Data to be sent: 1001101



Data sent with redundancy bits: 10011100101

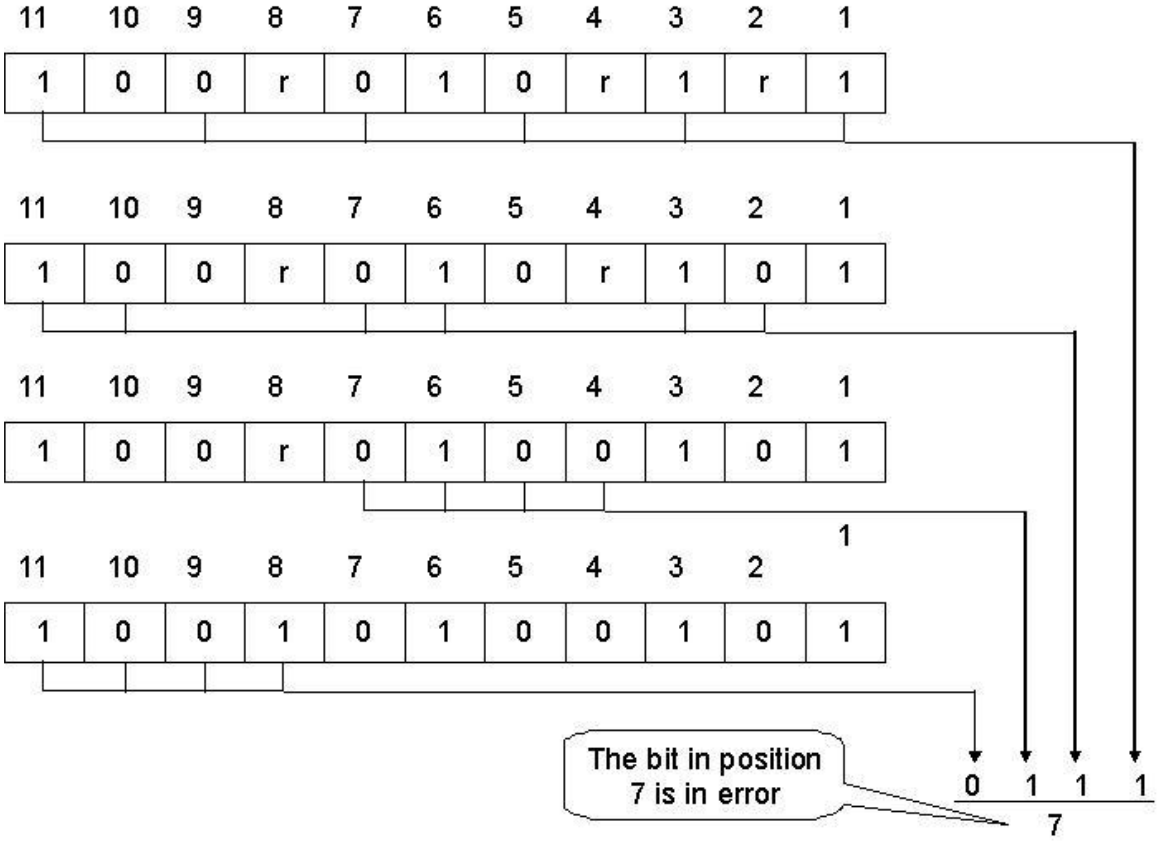
During transmission:



At the receiver:

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

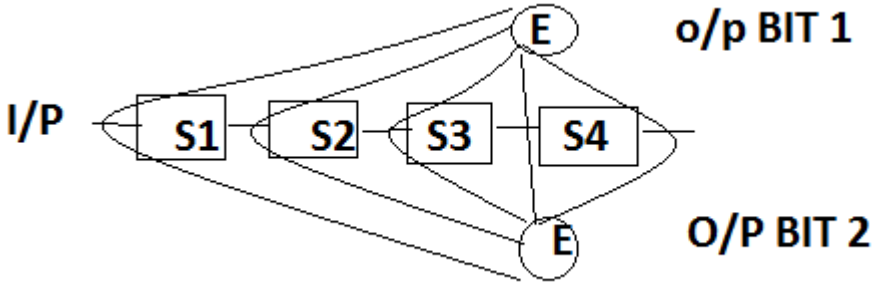
a binary number in order of r position (r8, r4, r2, r1).



Once the bit is identified, the receiver can reverse its value and correct the error.

BINARY CONVOLUTIONAL CODES

An encoder processes a sequence of input bits and generates a sequence of output bits. There is no natural message size or encoding encoding boundary as in a block code. The output depends on the current and previous input bits. That is the encoder has memory. The number of previous bit on which output depends is called constraint length.



REED SOLOMON CODES

Reed Solomon codes are linear block codes and systematic too. It operates on individual bits. , Reed Solomon codes operate on m bits. Every n degree polynomial is uniquely determined by n+1 points. For eg.a line having the form ax+b is determined by two points. Extra points on same line are redundant which is helpful for error correction.

LOW DENSITY PARITY CHECK CODES(LDPC CODES)

LPDC codes are linear block. each output is formed from fraction of input bits. This can be used in matrix representation.There are large blocks and can be excellence for error correction.

4.Write a note on Datalink protocols. (or) Write a note on Simplex Protocols.

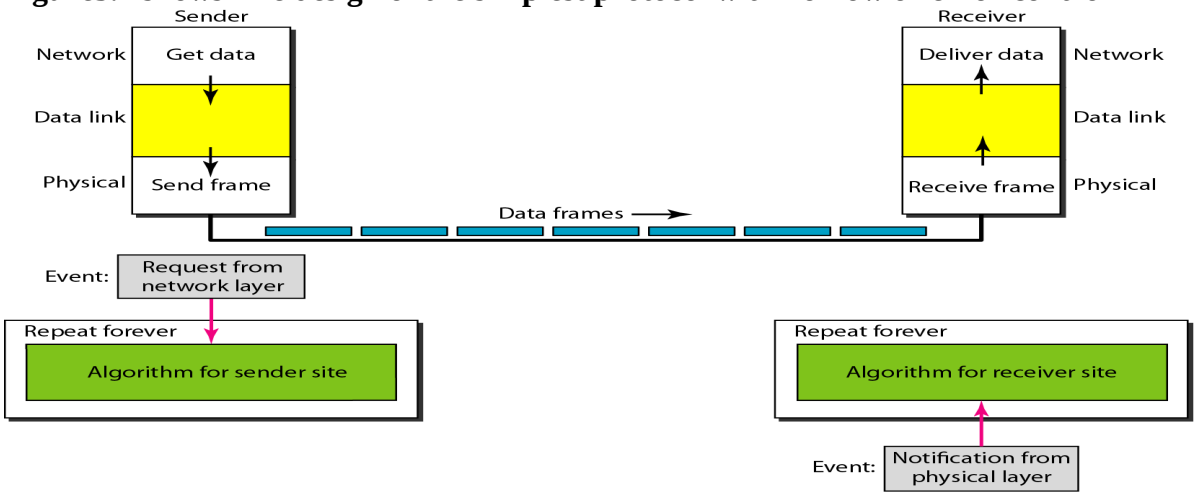
1. A Utopian Simplex protocols
2. A Simplex Stop-and-wait protocol for an Error-Free channel
3. A Simplex Stop-and-wait protocols for a Noisy Channel

i) A UTOPIAN SIMPLEX PROTOCOL

The following assumption has been made for developing the (algorithm) simplex protocol.

- The channel is a perfect noiseless channel.
- Hence an ideal channel in which no frames are lost, duplicated, or corrupted.
- No flow control and error control used.
- It is a unidirectional protocol in which data frames are traveling in only one direction- from the sender to receiver.
- Both transmitting and receiving network layer are always ready.
- Processing time that is small enough to be negligible.
- Infinite buffer space is available.

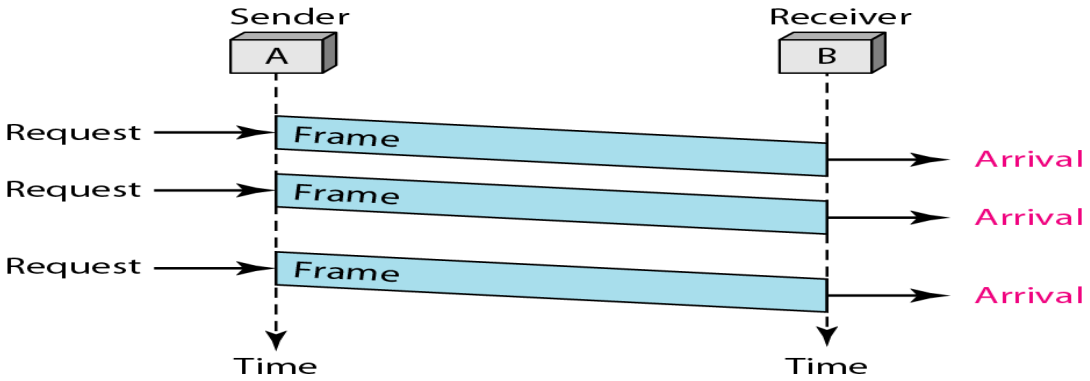
Figure3.1 shows The design of the simplest protocol with no flow or error control



Example for simplex protocol

- **Figure 3.2** below shows an example of communication using this protocol. It is very simple.
- The sender sends a sequence of frames without even thinking about the receiver.
- To send three frames, three events occur at the sender site and three events at the receiver site.

Figure3.2 Example for simplex protocol



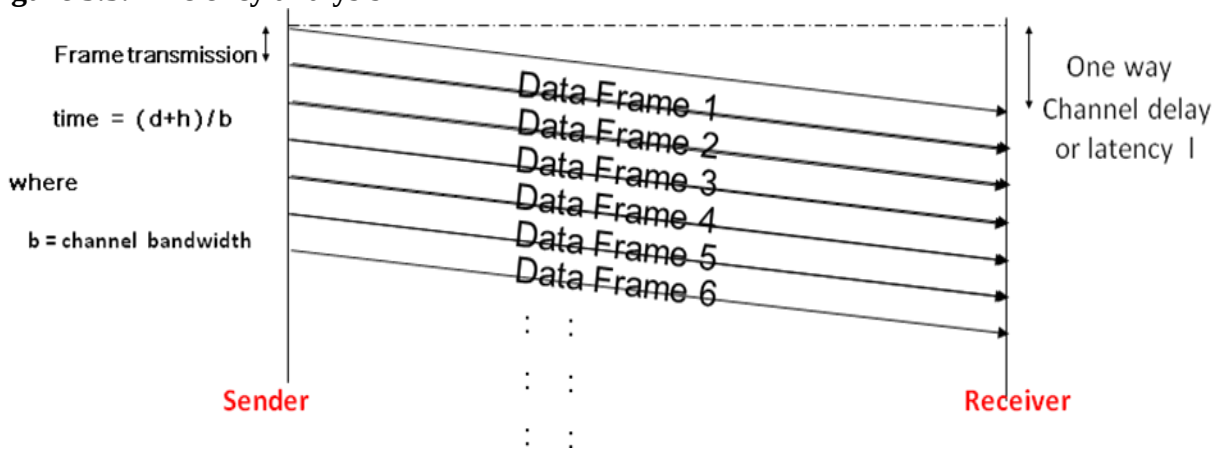
Efficiency analysis

- Transmission in one direction
- The receiver is always ready to receive the next frame (has infinite buffer storage).
- Error-free communication channel.
- No acknowledgments or retransmissions used.
- If frame has d data bits and h overhead bits, channel bandwidth b bits/second:

Maximum Channel Utilization = data size / frame size = $d / (d + h)$

Maximum Data Throughput = $d/(d+h)$ * channel bandwidth = $d/(d+h)$ * b

Figure 3.3.Efficiency analysis



/* Protocol 1 (Utopia) provides for data transmission in one direction only, from sender to receiver. The communication channel is assumed to be error free and the receiver is assumed to be able to process all the input infinitely quickly.Consequently, the sender just sits in a loop pumping data out onto the line as fast as it can. */

```
typedefenum {frame arrival} event type;
#include "protocol.h"
void sender1(void)
{
        frame s; /* buffer for an outbound frame */
        packet buffer; /* buffer for an outbound packet */
        while (true)
                {
                from network layer(&buffer); /* go get something to send */
                s.info = buffer; /* copy it into s for transmission */
                to physical layer(&s); /* send it on its way */
                }
}

void receiver1(void)
   {
        frame r;
        event type event; /* filled in by wait, but not used here */
        while (true)
                {
                wait for event(&event); /* only possibility is frame arrival */
                from physical layer(&r); /* go get the inbound frame */
                to network layer(&r.info); /* pass the data to the network layer */
                }
 }
```

ii) A SIMPLEX STOP-AND-WAIT PROTOCOL FOR AN ERROR-FREE CHANNEL

The following assumption has been made for developing the Stop-and-Wait Protocol

- The channel is a perfect noiseless channel.
- Flow control used
- It is a bidirectional protocol in which frames are traveling in both direction
- Both transmitting and receiving network layer are always not ready.

- Processing time considerable
- Finite buffer space is available
- The receiver may not be always ready to receive the next frame (finite buffer storage).
- Receiver sends a positive acknowledgment frame to sender to transmit the next data frame which showed in the below figure(3.4).
- Error-free communication channel assumed. No retransmissions used.
- Maximum channel utilization » (time to transmit frame /round trip time)*d / (d +h)

» $d / (b * R)$

- Maximum data throughput » Channel Utilization * Channel Bandwidth

» $d/ (b * R) * b = d / R$

Figure 3.4. Stop-and-Wait protocol flow diagram

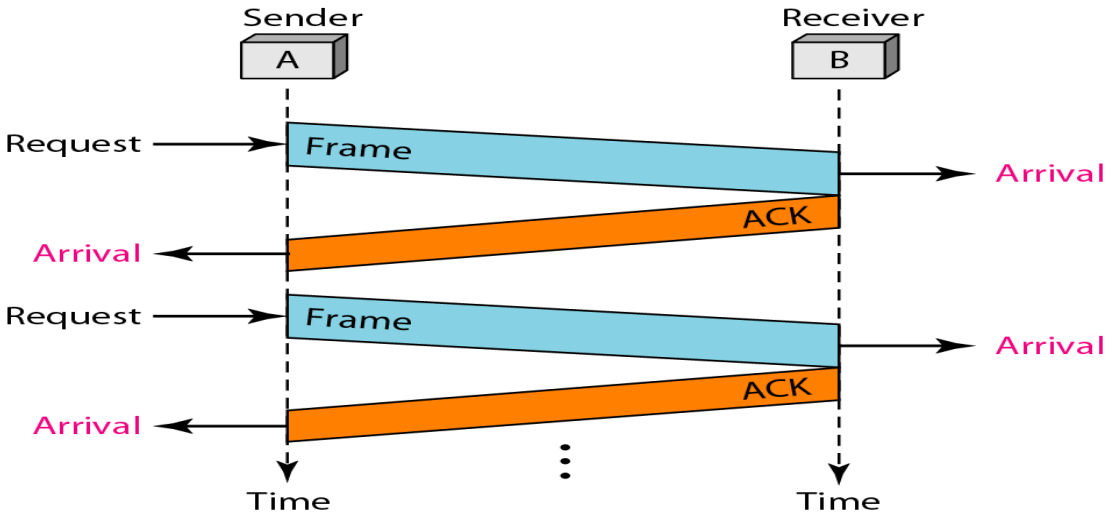
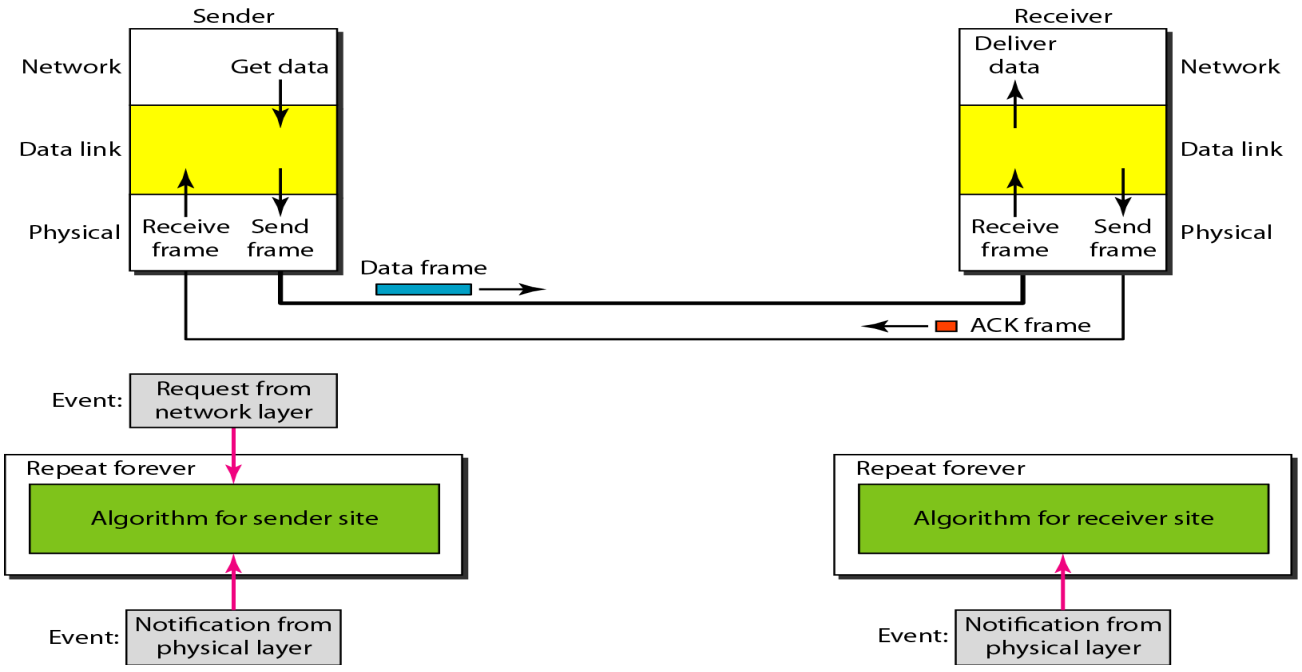


Figure 3.5 Design of Stop-and-Wait Protocol



/* Protocol 2 (Stop-and-wait) also provides for a one-directional flow of data from sender to receiver. The communication channel is once again assumed to be error free, as in protocol 1. However, this time the receiver has only a finite buffer capacity and a finite processing speed, so the protocol must explicitly prevent the sender from flooding the receiver with data faster than it can be handled. */

```
typedefenum {frame arrival} event type;
#include "protocol.h"

void sender2(void)
{
        frame s; /* buffer for an outbound frame */
        packet buffer; /* buffer for an outbound packet */
        event type event; /* frame arrival is the only possibility */
        while (true)
                {
                from network layer(&buffer); /* go get something to send */
                s.info = buffer; /* copy it into s for transmission */
                to physical layer(&s); /* bye-bye little frame */
                wait for event(&event); /* do not proceed until given the go ahead */
                }
}



void receiver2(void)
{
        frame r, s; /* buffers for frames */
        event type event; /* frame arrival is the only possibility */
        while (true)
                {
                wait for event(&event); /* only possibility is frame arrival */
                from physical layer(&r); /* go get the inbound frame */
                to network layer(&r.info); /* pass the data to the network layer */
                to physical layer(&s); /* send a dummy frame to awaken sender */
                }
}
```

iii) **A SIMPLEX STOP-AND-WAIT PROTOCOL FOR A NOISY CHANNEL**

Automatic Repeat Request (ARQ)

Purpose: To ensure a sequence of information packets is delivered in order and without errors or duplications despite transmission errors & losses.

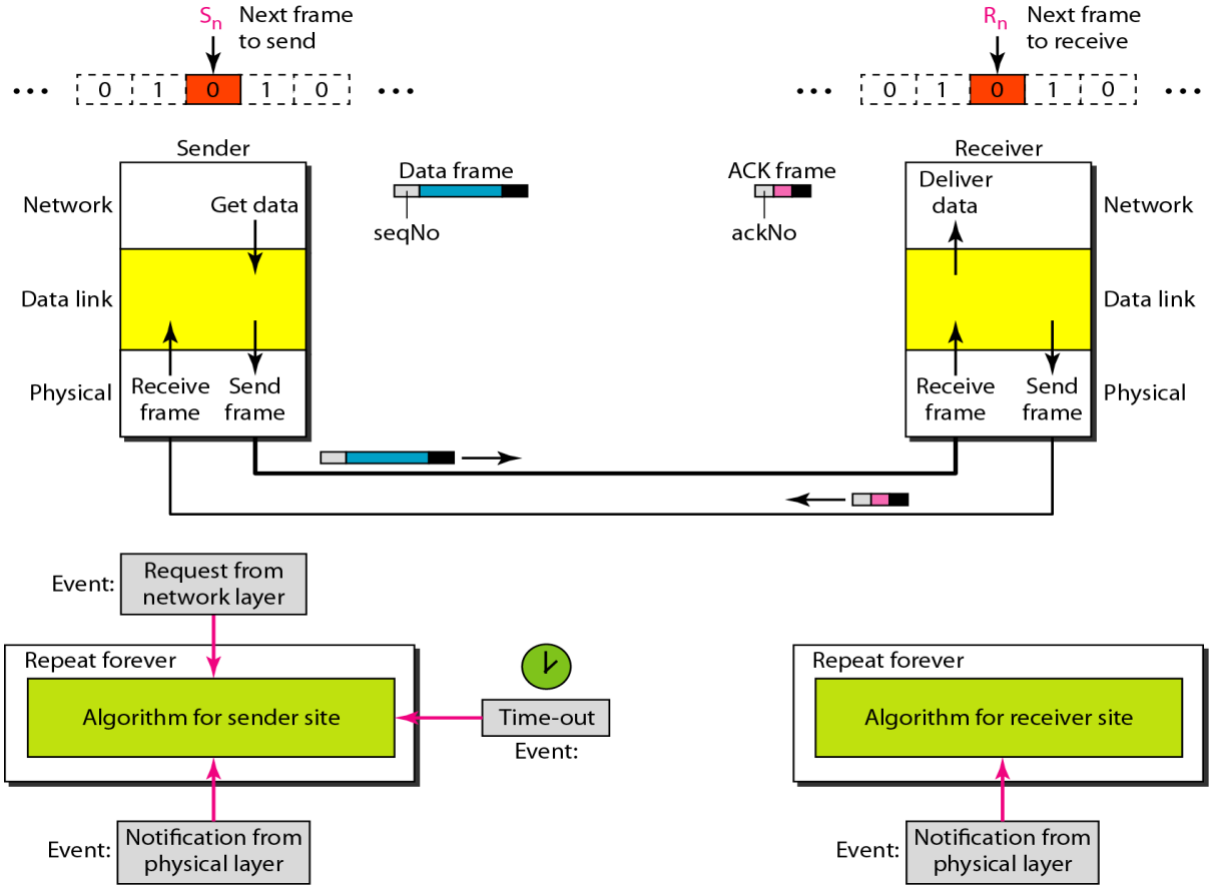
1. <u>STOP AND WAIT WITH ARQ</u>

Automatic Repeat Request (ARQ), an error control method, is incorporated with stop and wait flow control protocol

- If error is detected by receiver, it discards the frame and send a negative ACK (NAK), causing sender to re-send the frame.
- In case a frame never got to receiver, sender has a timer: each time a frame is sent, timer is set ! If no ACK or NAK is received during timeout period, it re-sends the frame

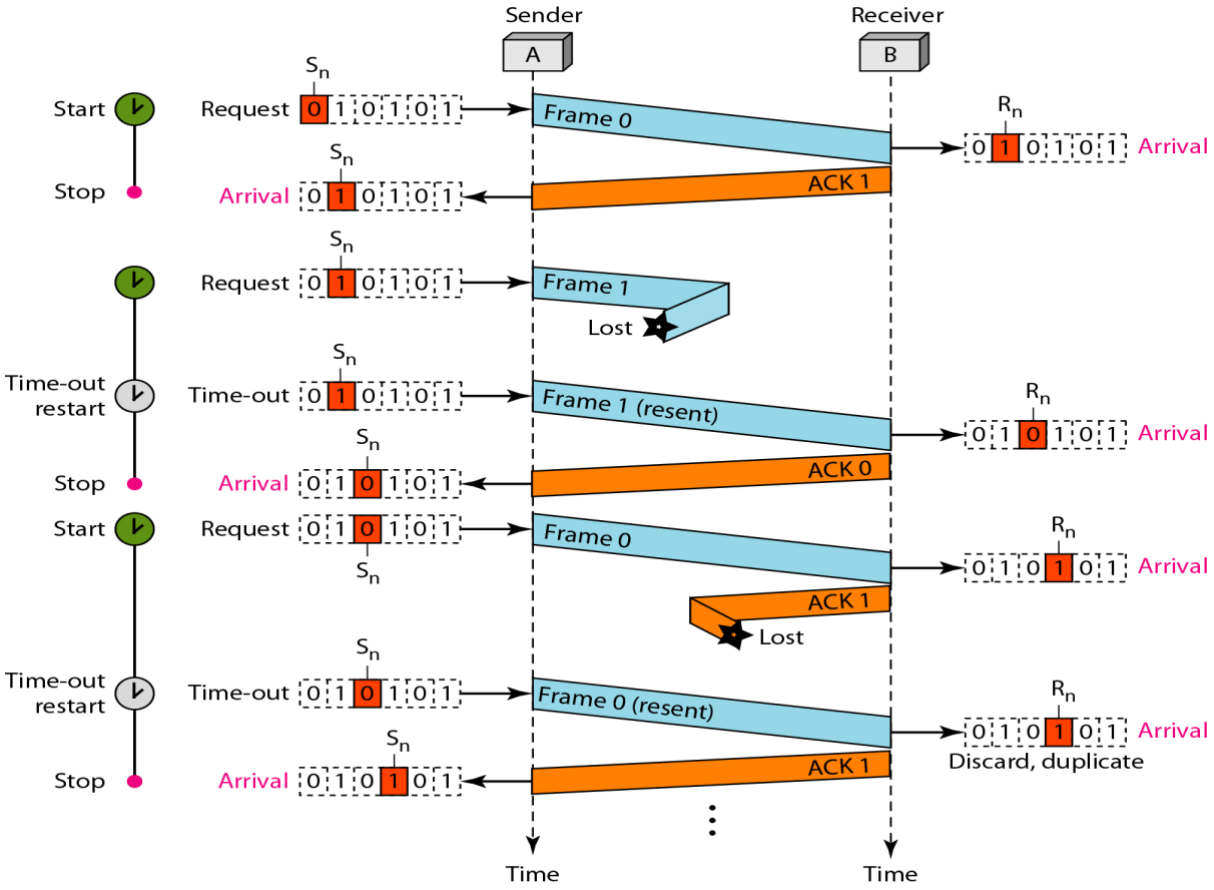
- Timer introduces a problem: Suppose timeout and sender retransmits a frame but receiver actually received the previous transmission ! receiver has duplicated copies.
- To avoid receiving and accepting two copies of same frame, frames and ACKs are alternatively labeled 0 or 1: ACK0 for frame 1, ACK1 for frame 0.

Figure 3.6 showsDesign of the Stop-and-Wait ARQ Protocol



Example

Figure 3.7 Shows an example of Stop-and-Wait ARQ.



Event :

- Frame 0 is sent and acknowledged.
- Frame 1 is lost and resent after the time-out.
- The resent frame 1 is acknowledged and the timer stops.
- Frame 0 is sent and acknowledged, but the acknowledgment is lost.
- The sender has no idea if the frame or the acknowledgment is lost.
- So after the time-out, it resends frame0, which is acknowledged.

ADVANTAGES OF STOP AND WAIT ARQ

- It can be used for noisy channels
- It has both error and flow control mechanism
- It has timer implementation

DISADVANTAGES OF STOP AND WAIT ARQ

- Efficiency is very less.
- Only 1 frame is sent at a time
- Timer should be set for each individual frame
- No pipelining
- sender window size is 1(disadvantage over Go back N ARQ)
- receiver window size is 1(disadvantage over selective repeat ARQ)

```
/* Protocol 3 (PAR) allows unidirectional data flow over an unreliable channel. */

#define MAX SEQ 1 /* must be 1 for protocol 3 */
typedefenum {frame arrival, cksum err, timeout} event type;
#include "protocol.h"
void sender3(void)
{
        seq nr next frame to send; /* seq number of next outgoing frame */
```

```
        frame s; /* scratch variable */
        packet buffer; /* buffer for an outbound packet */
        event type event;
        next frame to send = 0; /* initialize outbound sequence numbers */
        from network layer(&buffer); /* fetch first packet */
        while (true)
          {
        s.info = buffer; /* construct a frame for transmission */
        s.seq = next frame to send; /* insert sequence number in frame */
        to physical layer(&s); /* send it on its way */
        start timer(s.seq); /* if answer takes too long, time out */
        wait for event(&event); /* frame arrival, cksum err, timeout */
        if (event == frame arrival)
            {
        from physical layer(&s); /* get the acknowledgement */
        if (s.ack == next frame to send)
                  {
                stop timer(s.ack); /* turn the timer off */
                from network layer(&buffer); /* get the next one to send */
                inc(next frame to send); /* invert next frame to send */
                }
            }
        }
}

void receiver3(void)
{
        seq nr frame expected;
        frame r, s;
        event type event;
        frame expected = 0;
        while (true)
         {
        wait for event(&event); /* possibilities: frame arrival, cksum err */
        if (event == frame arrival)          /* a valid frame has arrived */
                {
                from physical layer(&r); /* go get the newly arrived frame */
                if (r.seq == frame expected)   /* this is what we have been waiting for */
                        {
                        to network layer(&r.info); /* pass the data to the network layer */
                        inc(frame expected); /* next time expect the other sequence nr */
                        }
                s.ack = 1 − frame expected; /* tell which frame is being acked */
                to physical layer(&s); /* send acknowledgement */
                }
          }
}
```

5.Discuss about Sliding Window protocol.

Data-link layer is responsible for implementation of point-to-point flow and error control mechanism.

Flow Control

When a data frame (Layer-2 data) is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which

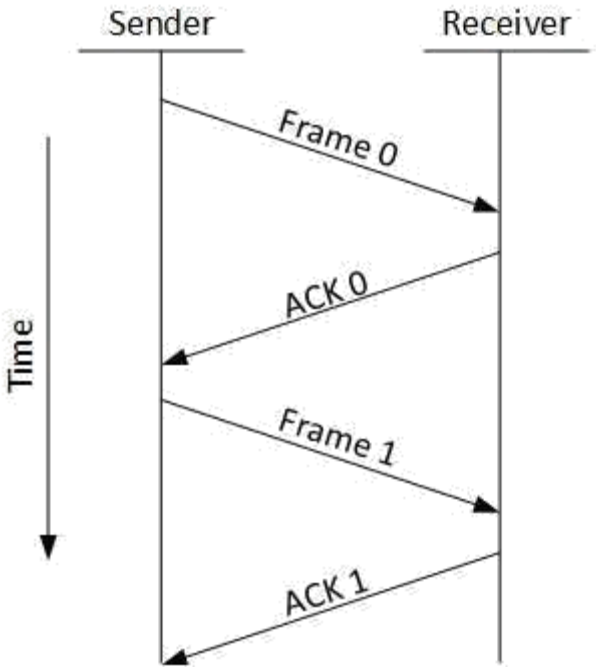
the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.

wo types of mechanisms can be deployed to control the flow:

1. **A One-Bit Sliding Window Protocol (Stop and wait)**
2. **A Protocol Using Go-Back-N**
3. **A Protocol Using Selective Repeat**

- **Stop and Wait**

  This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.



- **Sliding Window**

  In this flow control mechanism, both sender and receiver agree on the number of data-frames after which the acknowledgement should be sent. As we learnt, stop and wait flow control mechanism wastes resources, this protocol tries to make use of underlying resources as much as possible.

Error Control

When data-frame is transmitted, there is a probability that data-frame may be lost in the transit or it is received corrupted. In both cases, the receiver does not receive the correct data-frame and sender does not know anything about any loss. In such case, both sender and receiver are equipped with some protocols which helps them to detect transit errors such as loss of data-frame. Hence, either the sender retransmits the data-frame or the receiver may request to resend the previous data-frame.

Requirements for error control mechanism:

- **Error detection** - The sender and receiver, either both or any, must ascertain that there issome error in the transit.

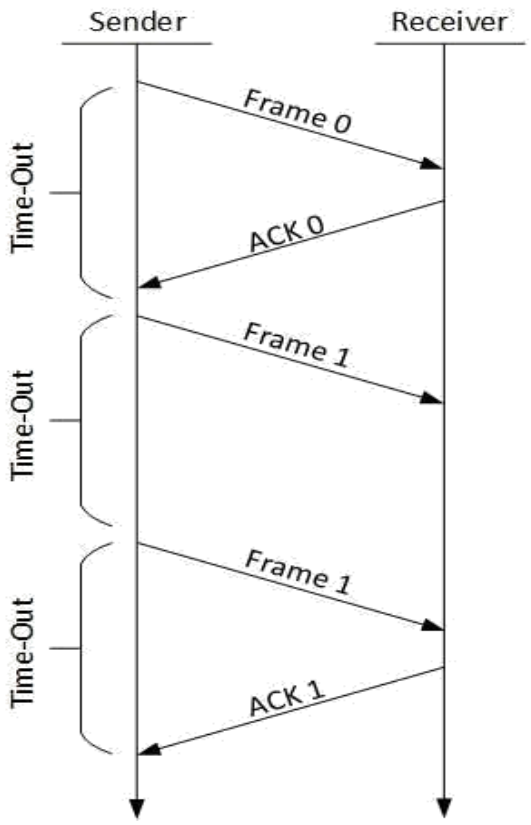
- **Positive ACK** - When the receiver receives a correct frame, it should acknowledge it.

- **Negative ACK** - When the receiver receives a damaged frame or a duplicate frame, it sends aNACK back to the sender and the sender must retransmit the correct frame.

- **Retransmission:** The sender maintains a clock and sets a timeout period. If anacknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it's acknowledgement is lost in transit

There are three types of techniques available which Data-link layer may deploy to control the errors by Automatic Repeat Requests (ARQ):

1. **A One-Bit Sliding Window Protocol (Stop and wait)**



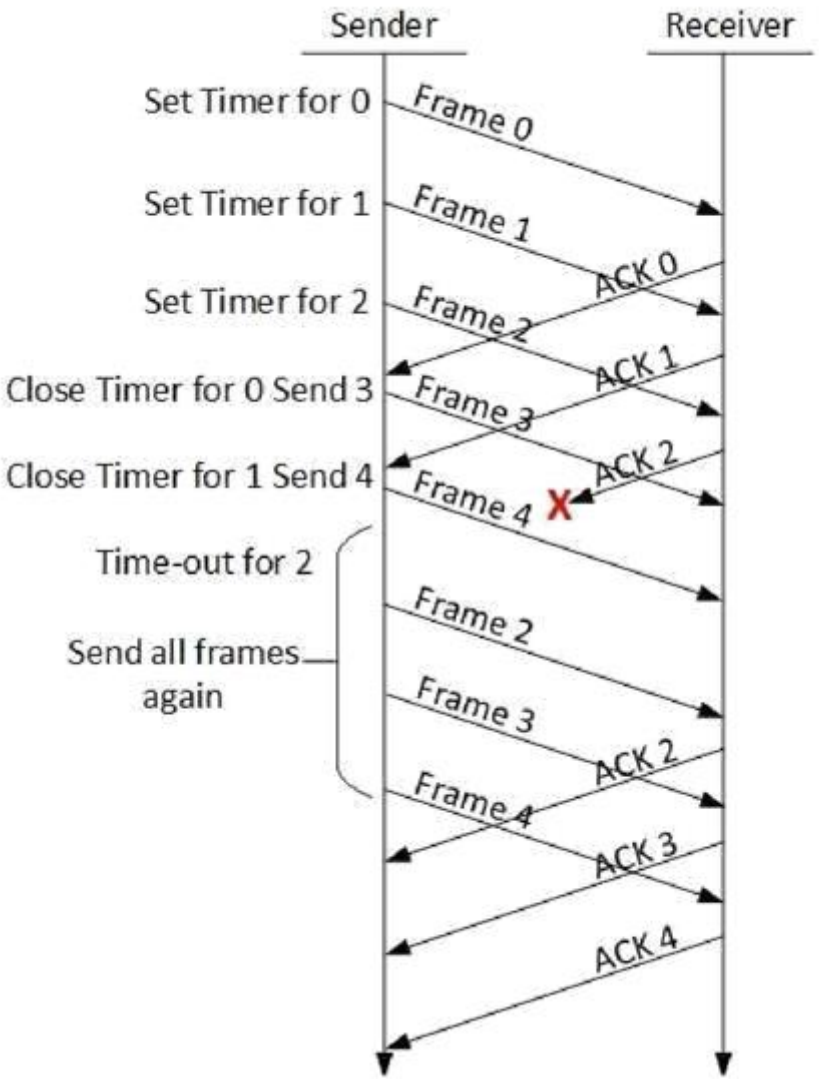
The following transition may occur in Stop-and-Wait ARQ:

- The sender maintains a timeout counter.
- When a frame is sent, the sender starts the timeout counter.
- If acknowledgement of frame comes in time, the sender transmits the next frame in queue.
- If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
- If a negative acknowledgement is received, the sender retransmits the frame.

2.A Protocol Using Go-Back-N

Stop and wait ARQ mechanism does not utilize the resources at their best.When the

acknowledgement is received, the sender sits idle and does nothing. In Go-Back-N ARQ method, both sender and receiver maintain a window.

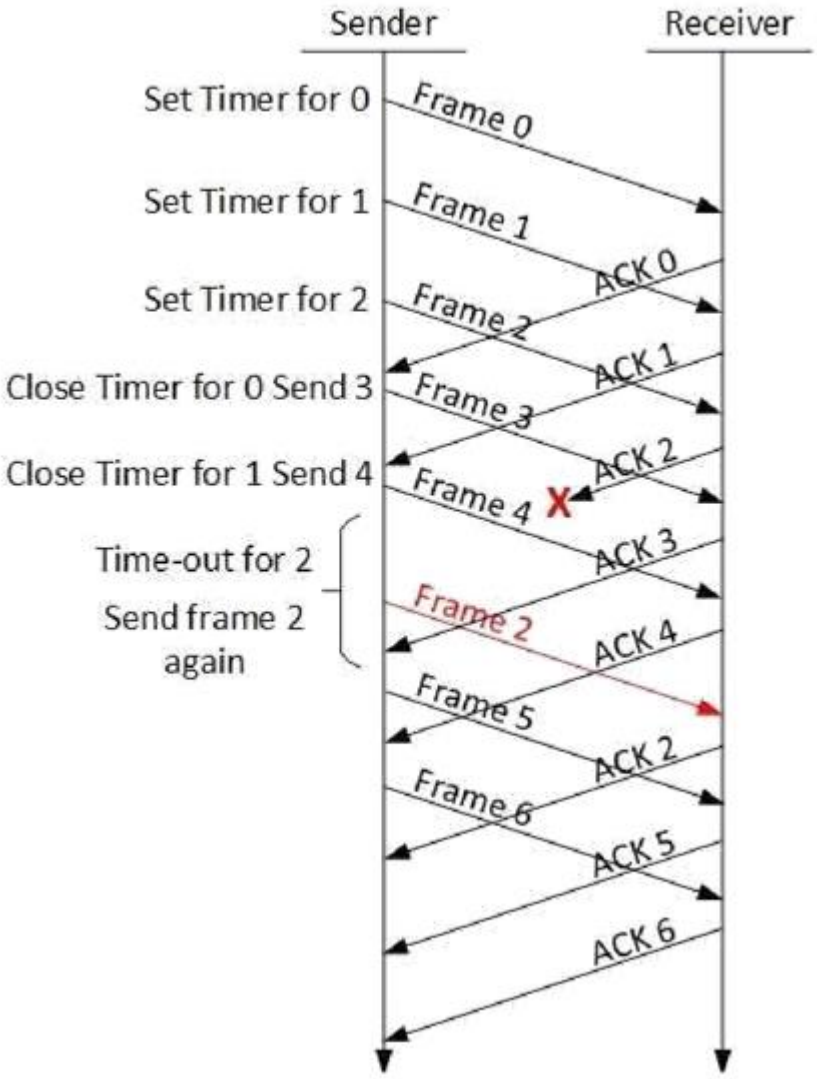


The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones. The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of incoming frame's sequence number.

When the sender sends all the frames in window, it checks up to what sequence number it has received positive acknowledgement. If all frames are positively acknowledged, the sender sends next set of frames. If sender finds that it has received NACK or has not receive any ACK for a particular frame, it retransmits all the frames after which it does not receive any positive ACK.

3.A Protocol Using Selective Repeat

In Go-back-N ARQ, it is assumed that the receiver does not have any buffer space for its window size and has to process each frame as it comes. This enforces the sender to retransmit all the frames which are not acknowledged.



In Selective-Repeat ARQ, the receiver while keeping track of sequence numbers, buffers the frames in memory and sends NACK for only frame which is missing or damaged.

The sender in this case, sends only packet for which NACK is received.

6. Discuss datalink layer

1. Datalink layer design issues.

2.Error detection

3. Error correction

4. Datalink layer protocol or Simplex Protocol

5.Sliding Window Protocol.

7.Discuss Medium Access Control Sublayer.

1.Channel Allocation Problem.

2.**M**ultiple Access protocol.

3. Wireless LAN (802.11)

4. Broadband wireless (802.16)

5.Datalink layer Switching

8.Discuss Channel Allocation Problem.

The channel allocation problem is how to allocate a single broadcast channel among competing users.

1.Static Channel Allocation

2.Dynamic Channel Allocation

STATIC CHANNEL ALLOCATION IN LANs AND MANs

- The traditional way of allocating a single channel, such as a telephone trunk, among multiple competing users is **Frequency Division Multiplexing** (FDM).
- If there are *N* users, the bandwidth is divided into *N* equal-sized portions, each user being assigned one portion. Since each user has a private frequency band, there is no interference between users.
- When there is only a small and constant number of a user, each of which has a heavy (buffered) load of traffic (e.g., carriers' switching offices), FDM is a simple and efficient allocation mechanism.
- However, when the number of senders is large and continuously varying or the traffic is bursty, FDM presents some problems.
- If the spectrum is cut up into *N* regions and fewer than *N* users are currently interested in communicating, a large piece of valuable spectrum will be wasted.
- If more than *N* users want to communicate, some of them will be denied permission for lack of bandwidth, even if some of the users who have been assigned a frequency band hardly ever transmit or receive anything.
- However, even assuming that the number of users could somehow be held constant at *N*, dividing the single available channel into static sub channels is inherently inefficient.
- The basic problem is that when some users are quiescent, their bandwidth is simply lost. They are not using it, and no one else is allowed to use it either.
- Furthermore, in most computer systems, data traffic is extremely bursty (peak traffic to mean traffic ratios of 1000:1 are common). Consequently, most of the channels will be idle most of the time.
- The poor performance of static FDM can easily be seen from a simple queuing theory calculation. Let us start with the mean time delay, *T*, for a channel of capacity *C* bps, with an arrival rate of λ frames/sec, each frame having a length drawn from an exponential probability density function with mean 1/μ bits/frame. With these parameters the arrival rate is λ frames/sec and the service rate is μ *C* frames/sec. From queuing theory it can be shown that for Poisson arrival and service times,

$$T = \frac{1}{\mu C - \lambda}$$

- For example, if *C* is 100 Mbps, the mean frame length, 1/μ, is 10,000 bits, and the frame arrival rate, λ, is 5000 frames/sec, then *T* = 200 μ sec. Note that if we ignored the queuing delay and just asked how long it takes to send a 10,000 bit frame on a 100-Mbps network, we would get the (incorrect) answer of 100 μ sec. That result only holds when there is no contention for the channel.
- Now let us divide the single channel into *N*independent sub channels, each with capacity *C/N* bps. The mean input rate on each of the sub channels will now be λ/*N*. Recomputing*T* we get

Equation 4

$$T_{\text{FDM}} = \frac{1}{\mu(C/N) - (\lambda/N)} = \frac{N}{\mu C - \lambda} = NT$$

- The mean delay using FDM is *N* times worse than if all the frames were somehow magically arranged orderly in a big central queue.
- Precisely the same arguments that apply to FDM also apply to time division multiplexing (TDM). Each user is statically allocated every *N*th time slot. If a user does not use the allocated slot, it just lies fallow. The same holds if we split up the networks physically. Using our previous example

again, if we were to replace the 100-Mbps network with 10 networks of 10 Mbps each and statically allocate each user to one of them, the mean delay would jump from 200 μ sec to 2 msec.

- Since none of the traditional static channel allocation methods work well with bursty traffic, we will now explore dynamic methods.

DYNAMIC CHANNEL ALLOCATION IN LANs AND MANSs

FIVE KEY ASSUMPTIONS

Station Model.

- The model consists of *N*independent **stations** (e.g., computers, telephones, or personal communicators), each with a program or user that generates frames for transmission. Stations are sometimes called **terminals**.
- The probability of a frame being generated in an interval of length $\Delta t$is $\lambda\Delta t$, where λ is a constant (the arrival rate of new frames). Once a frame has been generated, the station is blocked and does nothing until the frame has been successfully transmitted.

Single Channel Assumption.

- A single channel is available for all communication. All stations can transmit on it and all can receive from it.
- As far as the hardware is concerned, all stations are equivalent, although protocol software may assign priorities to them.

Collision Assumption.

- If two frames are transmitted simultaneously, they overlap in time and the resulting signal is garbled. This event is called a **collision**.
- All stations can detect collisions. A collided frame must be transmitted again later. There are no errors other than those generated by collisions.

4a. Continuous Time.

Frame transmission can begin at any instant. There is no master clock dividing time into discrete intervals.

4b. Slotted Time.

- Time is divided into discrete intervals (slots). Frame transmissions always begin at the start of a slot.
- A slot may contain 0, 1, or more frames, corresponding to an idle slot, a successful transmission, or a collision, respectively.

5a. Carrier Sense.

Stations can tell if the channel is in use before trying to use it. If the channel is sensed as busy, no station will attempt to use it until it goes idle.

5b. No Carrier Sense.

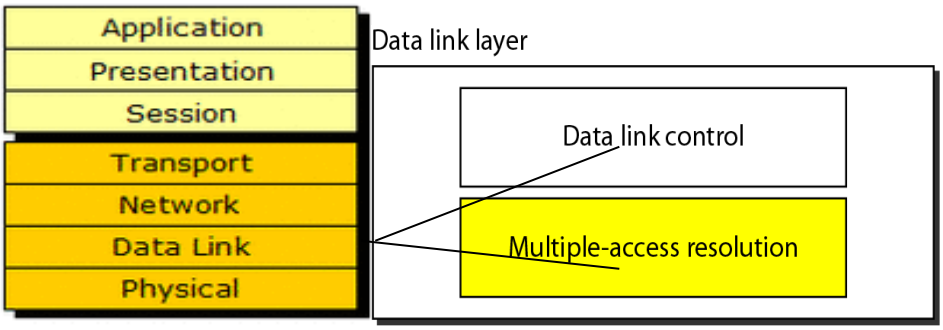
- Stations cannot sense the channel before trying to use it. They just go ahead and transmit. Only later can they determine whether the transmission was successful.

9. Discuss about Multiple Access protocols.

- The media access control (MAC) data communication protocol sub-layer, also known as the **medium access control**, is a **sublayer of the data link** layer specified in the seven-layer OSI model.

- It provides addressing and channel access control mechanisms that make it possible for several terminals or network nodes to communicate within a multiple access network that

incorporates a shared medium, e.g. Ethernet. The hardware that implements the MAC is referred to as a medium access controller.



- The MAC sub-layer acts as an interface between the logical link control (LLC) sublayer and the network's physical layer. The MAC layer emulates a full-duplex logical communication channel in a multi-point network. This channel may provide unicast, multicast or broadcast communication service.

- The channel access control mechanisms provided by the MAC layer is known as a **multiple access protocol**. This makes it possible for several stations connected to the same physical medium to share it.

- Examples of shared physical media are bus networks, ring networks, hub networks, wireless networks and half-duplex point-to-point links.

- The multiple access protocol may detect or avoid data packet collisions if a packet mode contention based channel access method is used, or reserve resources to establish a logical channel if a circuit switched or channelization based channel access method is used. The channel access control mechanism relies on a physical layer multiplex scheme.

The most commonly used random access protocols

1. The ALOHA protocol ,
2. CSMA (carrier sense multiple access) protocol ,
3. CSMA/CD (carrier sense multiple access /collision detection) protocol and
4. Collision-Free Protocols
5. Limited-Contention Protocols

The ALOHA protocol ,

In a wireless broadcast system or a half-duplex two-way link, Aloha works perfectly. But as networks become more complex, for example in an Ethernet system involving multiple sources and destinations in which data travels many paths at once, trouble occurs because data frames collide (conflict). The
heavier the communications volume, the worse the collision problems become.
To minimize the number of collisions, thereby optimizing network efficiency and increasing the number of subscribers that can use a given network, a scheme called slotted Aloha was developed.
Further improvement can be realized by a more sophisticated protocol called Carrier Sense Multiple Access with Collision Detection (CSMA).

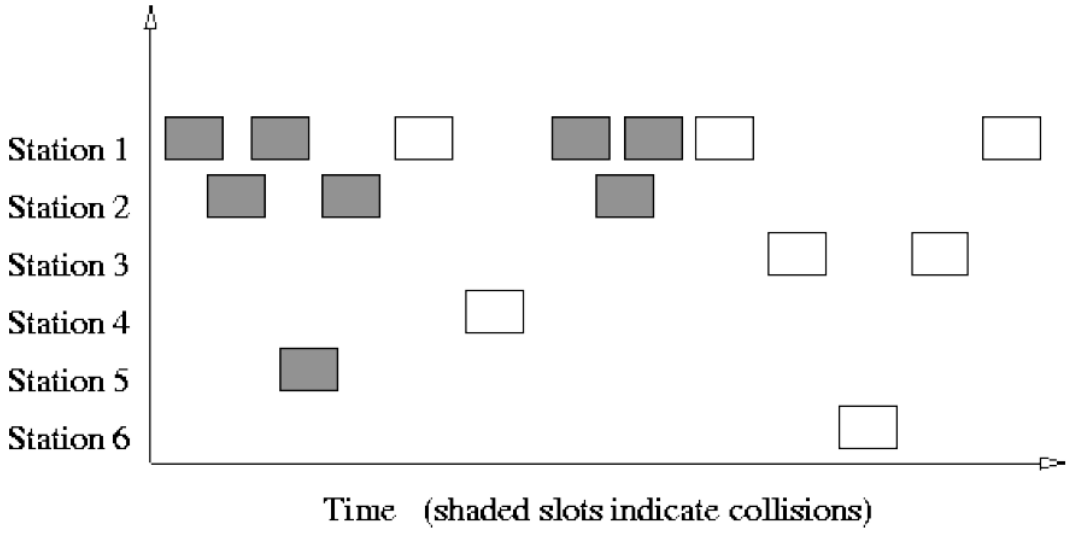
The two several of ALOHA are:

 PURE ALOHA

 SLOTTED ALOHA

Pure Aloha Protocol

With Pure Aloha, stations are allowed access to the channel whenever they have data to transmit. Because the threat of data collision exists, each station must either monitor its transmission on the rebroadcast or await an acknowledgment from the destination station. By comparing the transmitted packet with the received packet or by the lack of an acknowledgement, the transmitting station can determine the success of the transmitted packet. If the transmission was unsuccessful it is resent after a random amount of time to reduce the probability of re-collision.



Advantages:

- Superior to fixed assignment when there is a large number of bursty stations.
- Adapts to varying number of stations.

Disadvantages:

 Theoretically proven throughput maximum of 18.4%.

 Requires queueing buffers for retransmission of packets.

Slotted Aloha Protocol

By making a small restriction in the transmission freedom of the individual stations, the throughput o Aloha protocol can be doubled. Assuming constant length packets, transmission time is broken into s equivalent to the transmission time of a single packet. Stations are only allowed to transmit at slot boundaries. When packets collide they will overlap completely instead of partially. This has the effec doubling the efficiency of the Aloha protocol and has come to be known as Slotted Aloha

Advantages:

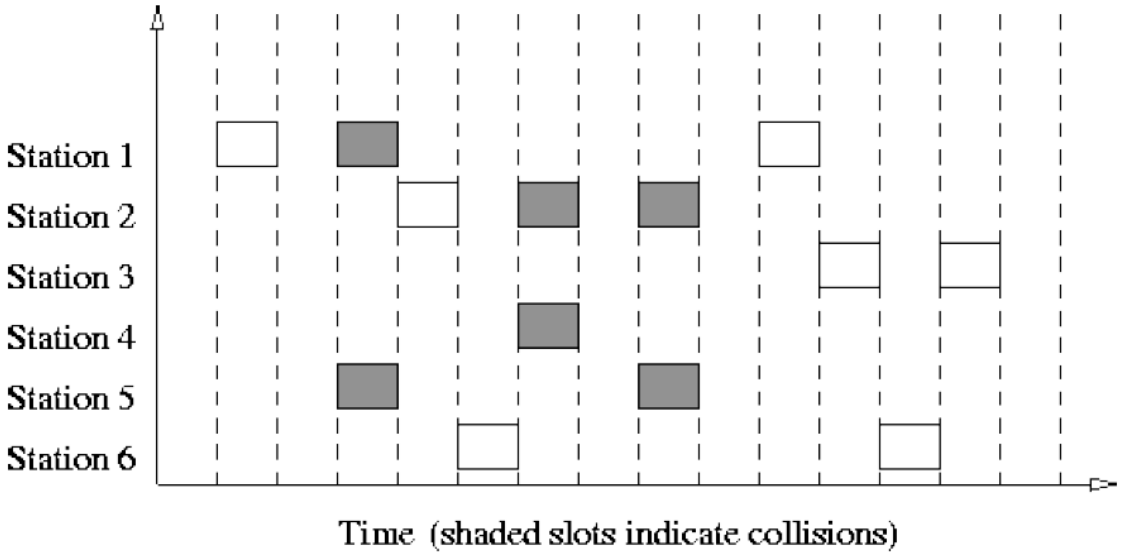
 Doubles the efficiency of Aloha.

 Adaptable to a changing station population.

Disadvantages:

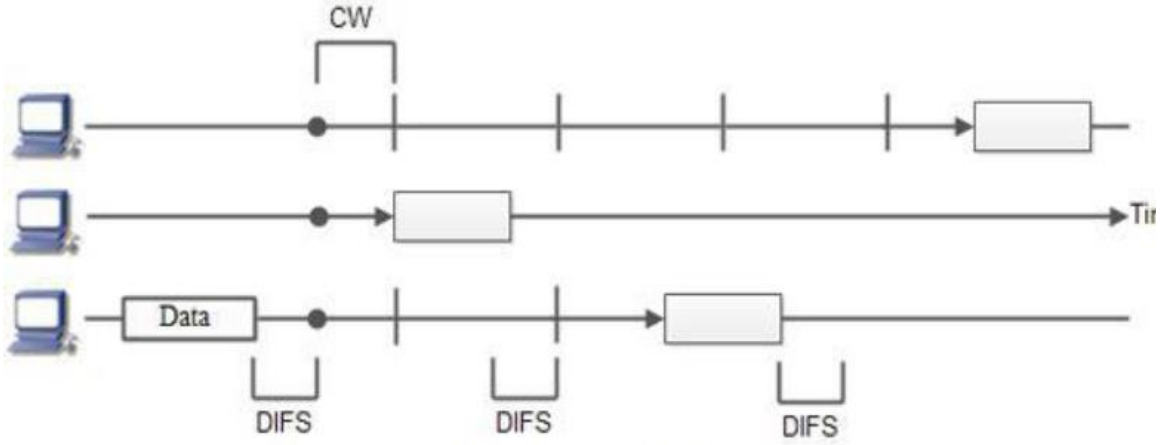
 Theoretically proven throughput maximum of 36.8%.

 Requires queueing buffers for retransmission of packets.



CSMA (carrier sense multiple access) protocol

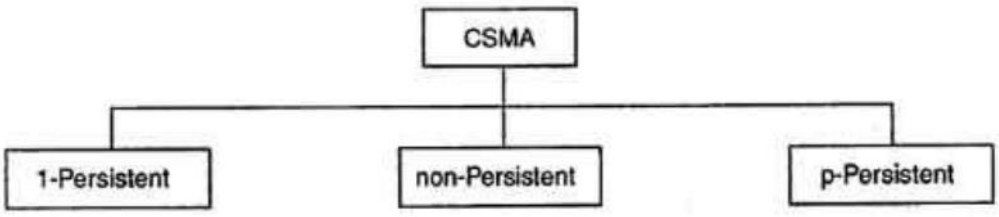
Carrier Sensed Multiple Accesses (CSMA): CSMA is a network access method used on shared network topologies such as Ethernet to control access to the network. Devices attached to the network cable listen (carrier sense) before transmitting. If the channel is in use, devices wait before transmitting. MA (Multiple Access) indicates that many devices can connect to and share the same network. All devices have equal access to use the network when it is clear.



Sharing time with CSMA method.

There Are Three Different Type of CSMA Protocols

(I) I-persistent CSMA



(i) I-persistent CSMA

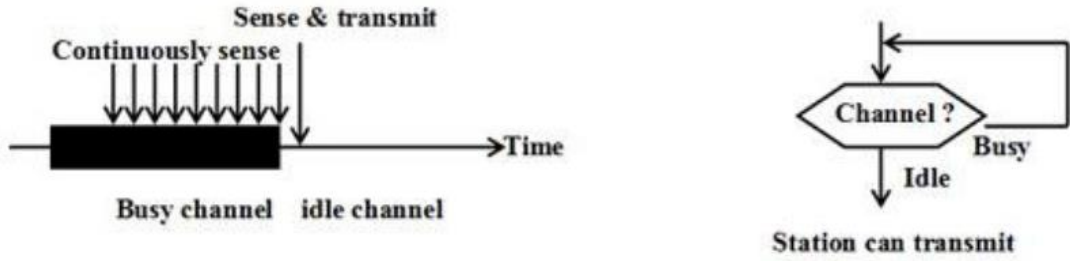
 In this method, station that wants to transmit data continuously senses the channel to check whether the channel is idle or busy.

 If the channel is busy, the station waits until it becomes idle.

 When the station detects an idle-channel, it immediately transmits the frame with probability 1. Hence it is called I-persistent CSMA.

 This method has the highest chance of collision because two or more stations may find channel to be idle at the same time and transmit their frames.

 When the collision occurs, the stations wait a random amount of time and start allover again.



1-persistent CSMA

(ii) Non-persistent CSMA

 In this scheme, if a station wants to transmit a frame and it finds that the channel is busy (some other station is transmitting) then it will wait for fixed interval of time.

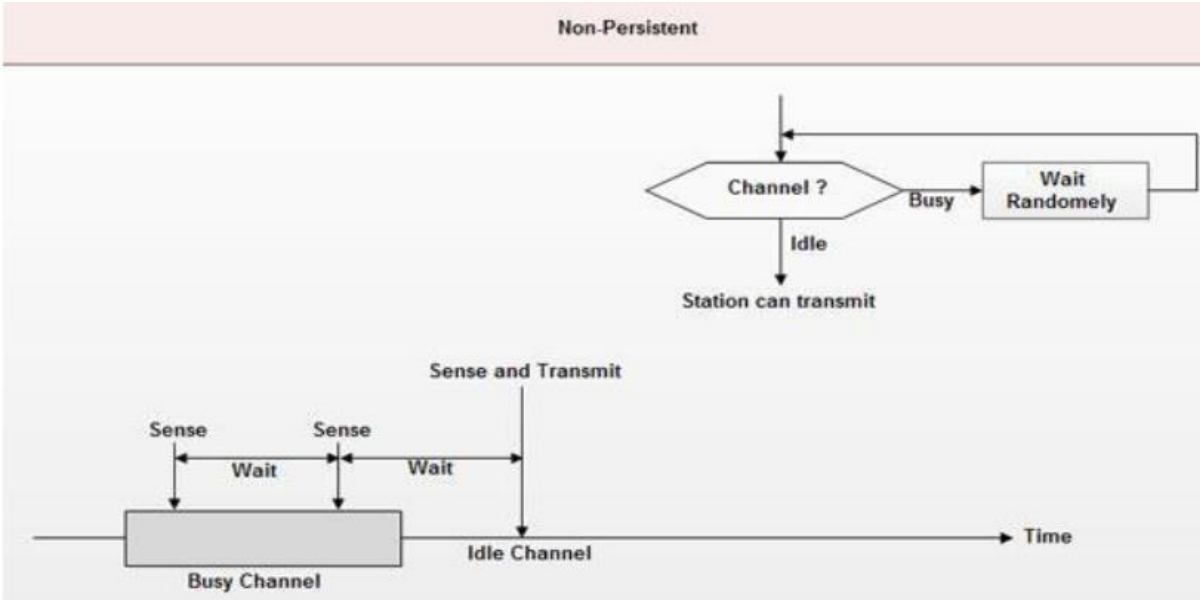
 After this time, it again checks the status of the channel and if the channel is. free it will transmit.

 A station that has a frame to send senses the channel.

 If the channel is idle, it sends immediately.

 If the channel is busy, it waits a random amount of time and then senses the channel again.

 In non-persistent CSMA the station does not continuously sense the channel for the purpose of capturing it when it detects the end of previous transmission.



(iii) p-persistent CSMA

 This method is used when channel has time slots such that the time slot duration is equal to or greater than the maximum propagation delay time.

 Whenever a station becomes ready to send, it senses the channel.

 If channel is busy, station waits until next slot.

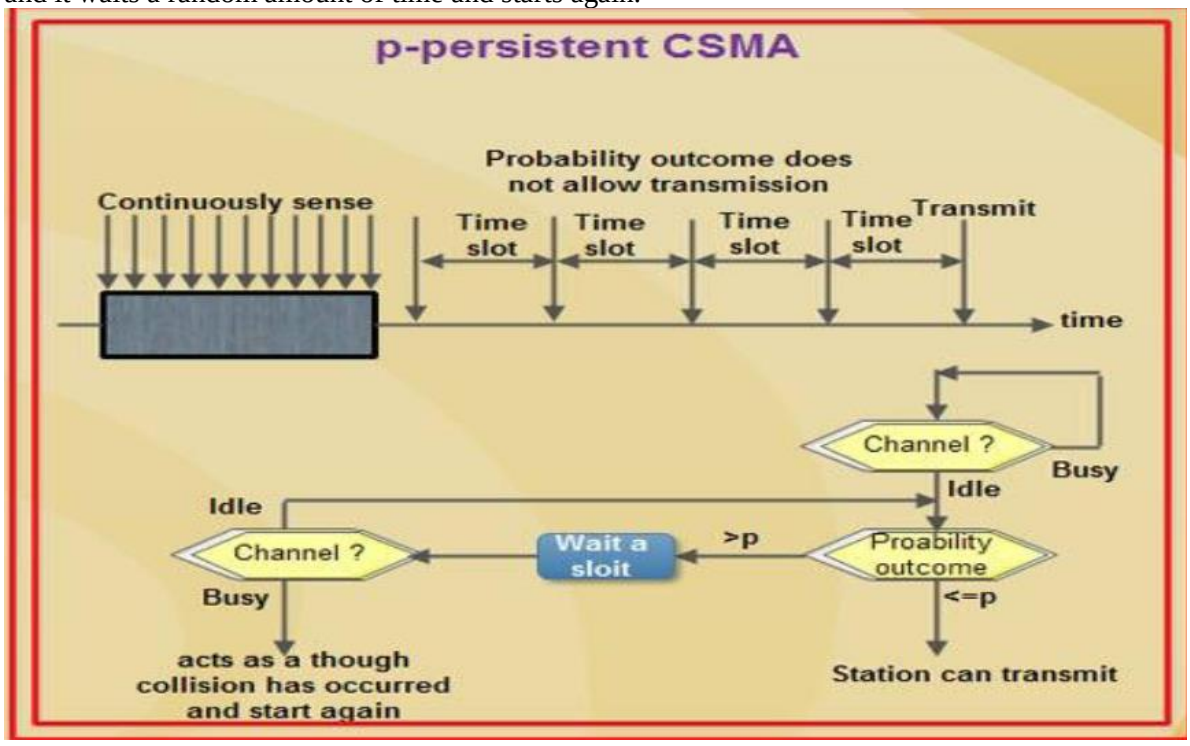
 If channel is idle, it transmits with a probability p.

 With the probability q=l-p, the station then waits for the beginning of the next time slot.

 If the next slot is also idle, it either transmits or waits again with probabilities p and q.

 This process is repeated till either frame has been transmitted or another station has begun transmitting.

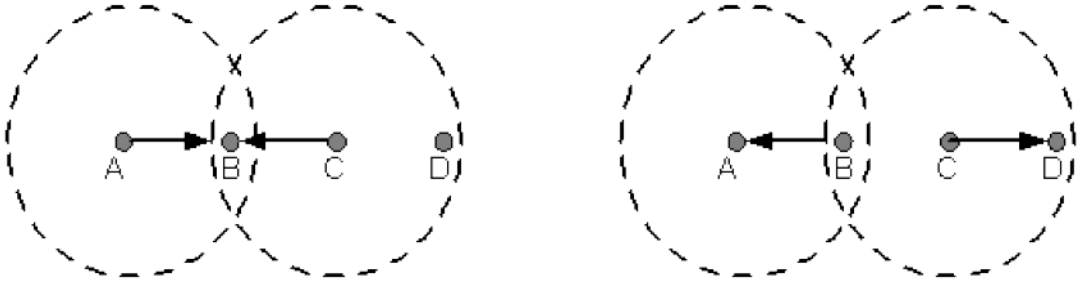
 In case of the transmission by another station, the station acts as though a collision has occurred and it waits a random amount of time and starts again.



CSMA/CD (carrier sense multiple access /collision detection) protocol and

CSMA with Collision Avoidance
CSMA/CD would break down in wireless networks because of hidden node and exposed nodes problems.

Hidden Node Problem
In the case of wireless network it is possible that A is sending a message to B, but C is out of its range and hence while "listening" on the network it will find the network to be free and might try to send packets to B at the same time as A. So, there will be a collision at B. The problem can be looked upon as if A and C are hidden from each other. Hence it is called the "hidden node problem".
Exposed Node Problem
If C is transmitting a message to D and B wants to transmit a message to A, B will find the network to be busy as B hears C trnasmitting. Even if B would have transmitted to A, it would not have been a problem at A or D. CSMA/CD would not allow it to transmit message to A, while the two transmissions could have gone in parallel.



Addressing hidden node problem (CSMA/CA)
Suppose A wants to send a packet to B. Then it will first send a small packet to B called **"Request to Send" (RTS)**. In response, B sends a small packet to A called **"Clear to Send" (CTS)**. Only after A receives a CTS, it transmits the actual data. Now, any of the nodes which can hear either CTS or RTS assume the network to be busy. Hence even if some other node which is out of range of both A and B sends an RTS to C (which can hear at least one of the RTS or CTS between A and B), C would not send a CTS to it and hence the communication would not be established between C and D.
One issue that needs to be addressed is how long the rest of the nodes should wait before they can transmit data over the network. The answer is that the RTS and CTS would carry some information about the size of the data that B intends to transfer. So, they can calculate time that would be required for the transmission to be over and assume the network to be free after that. Another interesting issue is what a node should do if it hears RTS but not corresponding CTS.
One possibility is that it assumes the recipient node has not responded and hence no transmission is going on, but there is a catch in this. It is possible that the node hearing RTS is just on the boundary of the node sending CTS. Hence, it does hear CTS but the signal is so deteriorated that it fails to recognize it as a CTS. Hence to be on the safer side, a node will not start transmission if it hears either of an RTS or CTS.

Collision-Free Protocols

Bit-Map Method

In this method, there N slots. If node 0 has a frame to send, it transmits a 1 bit during the first slot. No other node is allowed to transmit during this period. Next node 1 gets a chance to transmit 1 bit f it has something to send, regardless of what node 0 had transmitted. This is done for all the nodes. In general node j may declare the fact that it has a frame to send by inserting a 1 into slot j. Hence after all nodes have passed, each node has complete knowledge of who wants to send a frame. Now they begin transmitting in numerical order. Since everyone knows who is transmitting and when, there could never be any collision.

Binary Countdown

In this protocol, a node which wants to signal that it has a frame to send does so by writing its address into the header as a binary number. The arbitration is such that as soon as a node sees that a higher bit position that is 0 in its address has been overwritten with a 1, it gives up. The final result is the address of the node which is allowed to send. After the node has transmitted the whole process is repeated all over again.

Nodes	Addresses
A	0010
B	0101
C	1010
D	1001

1010

Node C having higher priority gets to transmit. The problem with this protocol is that the nodes with address always win.

Limited-Contention Protocols

One could combine the best properties of the contention and contention - free protocols, that is, protocol which used contention at low loads to provide low delay, but used a cotention-free technique at high load to provide good channel efficiency. Such protocols do exist and are called Limited contention protocols.

Adaptive Tree Walk Protocol

The following is the method of adaptive tree protocol. Initially all the nodes are allowed to try to aquire the channel. If it is able to aquire the channel, it sends its frame. If there is collision then the nodes are divided into two equal groups and only one of these groups compete for slot 1. If one of its member aquires the channel then the next slot is reserved for the other group. On the other hand, if there is a collision then that group is again subdivided and the same process is followed. This can be better understood if the nodes are thought of as being organised in a binary tree as shown in the following figure.

For example, consider the case of nodes G and H being the only ones wanting to transmit. At slot 1 a collision will be detected and so 2 will be tried and it will be found to be idle. Hence it is pointless to probe 3 and one should directly go to 6,7.

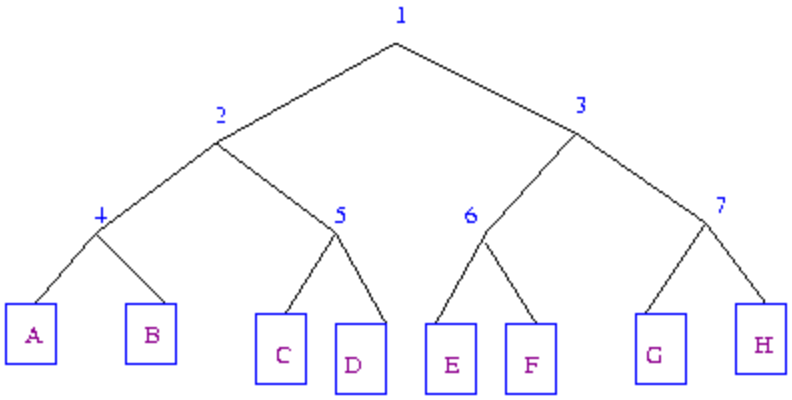


Fig. Adaptive Tree Walk abstraction of nodes in binary tree.

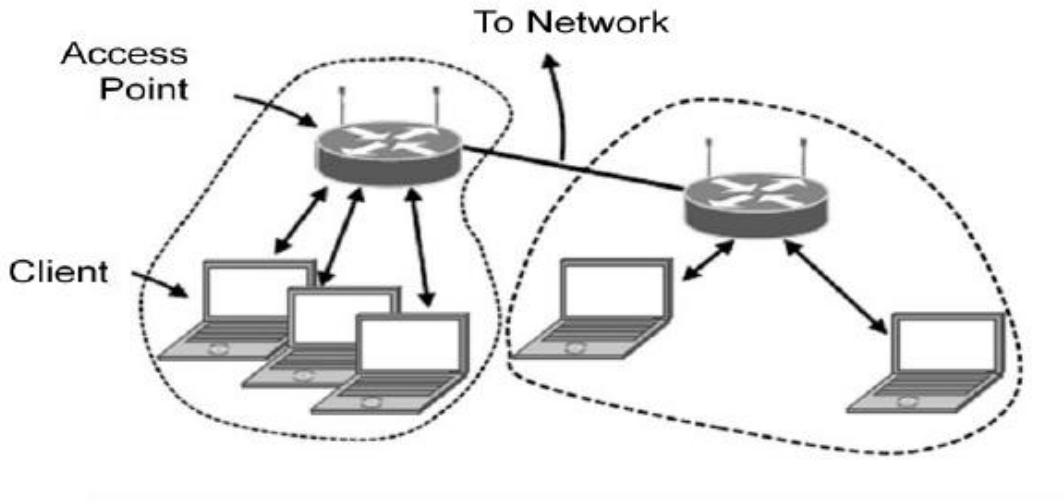
10. Discuss about wireless LANs. (802.11)

Wireless LANs are increasingly popular, and homes, offices, cafes, libraries, airports, zoos, and other public places are being outfitted with them to connect computers, PDAs, and smart phones to the Internet. Wireless LANs can also be used to let two or more nearby computers communicate without using the Internet.

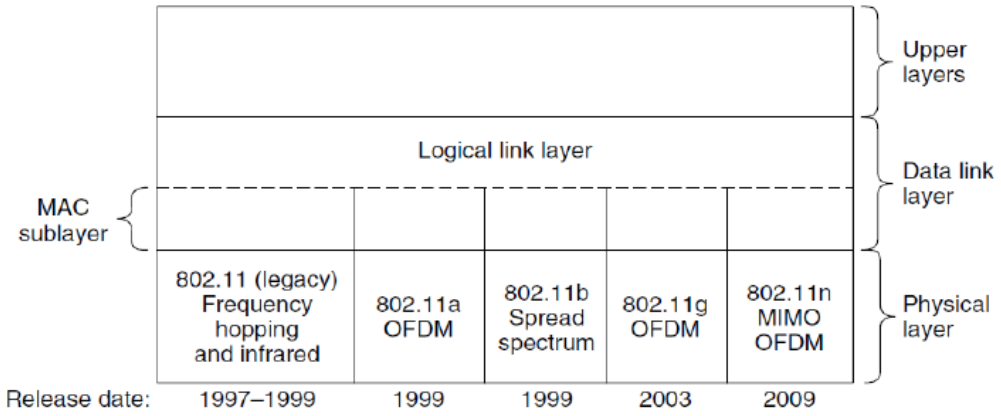
The 802.11 Architecture and Protocol Stack

802.11 networks can be used in two modes. The most popular mode is to connect clients, such as laptops and smart phones, to another network, such as a company intranet or the Internet. In infrastructure mode, each client is associated with an **AP** (**Access Point**) that is in turn connected to the other network. The client sends and receives its packets via the AP. Several access points may be connected together, typically by a wired network called a **distribution system**, to form an extended 802.11 network. In this case, clients can send frames to other clients via their APs. The other mode, shown in Fig is an **ad hoc network**. This mode is a collection of computers that are associated so that they can directly send frames to each other. There is no access point. Since Internet access is the killer application for wireless, ad hoc networks are not very popular.

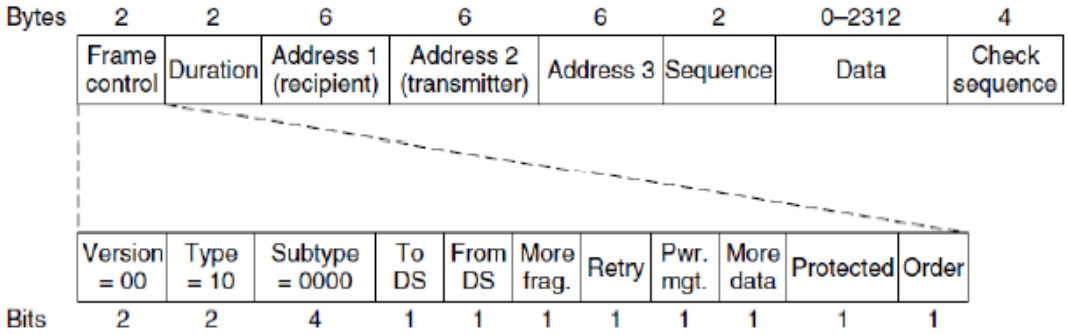
802.11 architecture



802.11 architecture – infrastructure mode



802.11 protocol stack



Format of the 802.11 data frame

The physical layer corresponds fairly well to the OSI physical layer, but the data link layer in all the 802 protocols is split into two or more sublayers. In 802.11, the MAC (Medium Access Control) sublayer determines how the channel is allocated, that is, who gets to transmit next. Above it is the LLC (Logical Link Control) sublayer, whose job it is to hide the differences between the different 802 variants and make them indistinguishable as far as the network layer is concerned.

Several transmission techniques have been added to the physical layer as 802.11 has evolved since it first appeared in 1997. Two of the initial techniques, infrared in the manner of television remote controls and frequency hopping in the 2.4-GHz band, are now defunct. The third initial technique, direct sequence spread spectrum at 1 or 2 Mbps in the 2.4-GHz band, was extended to run at rates up to 11 Mbps and quickly became a hit. It is now known as 802.11b.

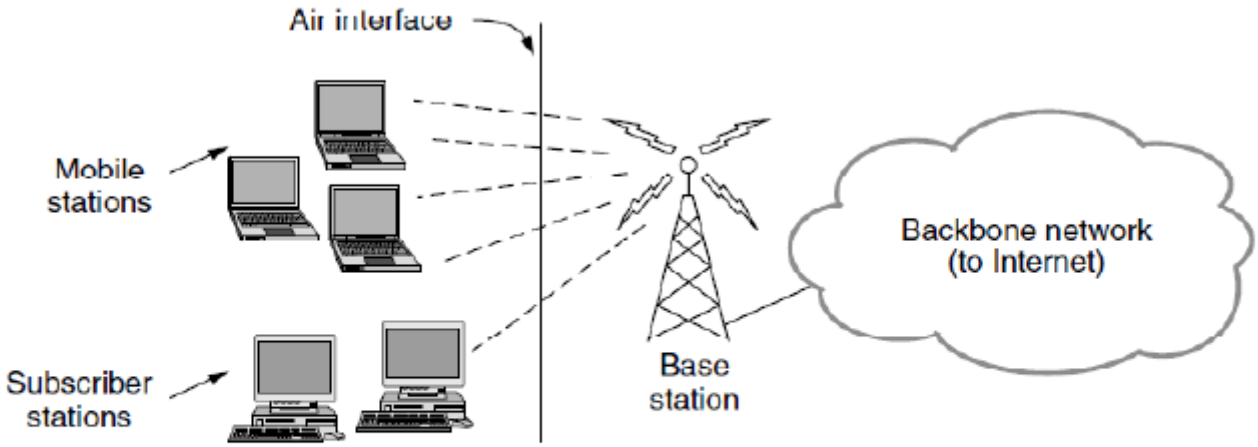
To give wireless junkies a much-wanted speed boost, new transmission techniques based on the OFDM (Orthogonal Frequency Division Multiplexing) scheme we described in Sec. 2.5.3 were introduced in 1999 and 2003. The first is called 802.11a and uses a different frequency band, 5 GHz. The second stuck with 2.4 GHz and compatibility. It is called 802.11g. Both give rates up to 54 Mbps. Most recently, transmission techniques that simultaneously use multiple antennas at the transmitter and receiver for a speed boost were finalized as 802.11n in Oct. 2009. With four antennas and wider channels, the 802.11 standard now defines rates up to a startling 600 Mbps.

11. Discuss about broadband Wireless. (802.16)

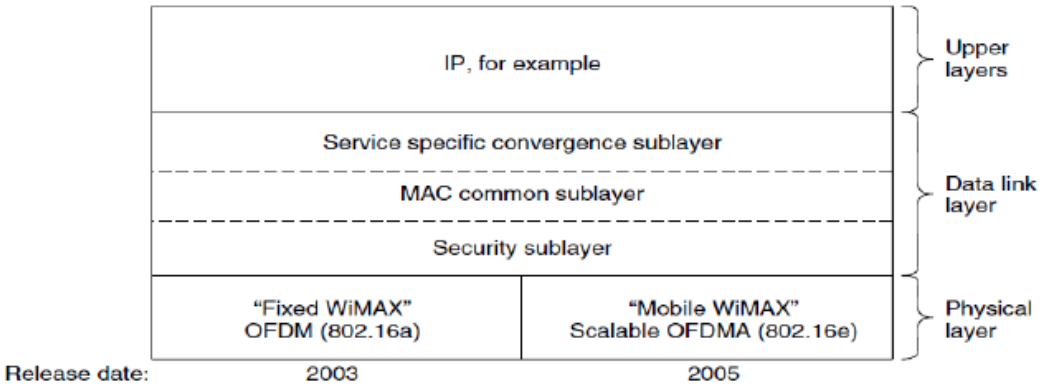
The 802.16 Architecture and Protocol Stack

The 802.16 architecture is shown in Fig. Base stations connect directly to the provider's backbone network, which is in turn connected to the Internet. The base stations communicate with stations over the wireless air interface. Two kinds of stations exist. Subscriber stations remain in a fixed location, for example, broadband Internet access for homes. Mobile stations can receive service while they are moving, for example, a car equipped with WiMAX.The 802.16 protocol stack that is used across the air interface is shown in Fig. The general structure is similar to that of the other 802 networks, but with more sublayers. The bottom layer deals with transmission, and here we have shown only the popular offerings of 802.16, fixed and mobile WiMAX. There is a different physical layer for each offering. Both layers operate in licensed spectrum below 11 GHz and use OFDM, but in different ways. Above the physical layer, the data link layer consists of three sublayers.

The bottom one deals with privacy and security, which is far more crucial for public outdoor networks than for private indoor networks. It manages encryption, decryption, and key management. Next comes the MAC common sublayer part. This part is where the main protocols, such as channel management, are located. The model here is that the base station completely controls the system. It can schedule the downlink (i.e.,base to subscriber) channels very efficiently and plays a major role in managing the uplink (i.e., subscriber to base) channels as well.



The 802.16 architecture



The 802.16 protocol stack

An unusual feature of this MAC sublayer is that, unlike those of the other 802 protocols, it is comple connection oriented, in order to provide quality of service guarantees for telephony and multim communication.The service-specific convergence sublayer takes the place of the logical link sublayer in the o 802 protocols. Its function is to provide an interface to the network layer. Different convergence layers are def to integrate seamlessly with different upper layers. The important choice is IP, though the standard def mappings for protocols such as Ethernet and ATM too. Since IP is connectionless and the 802.16 MAC subl is connection-oriented, this layer must map between addresses and connections.

12. Discuss about Data link layer Switching.

- Uses of Bridges
- Learning Bridges
- Spanning Tree Bridge
- Repeaters , hubs, Bridges, Switsche, Routers and gateways.
- Virtual LAN (V LAN)

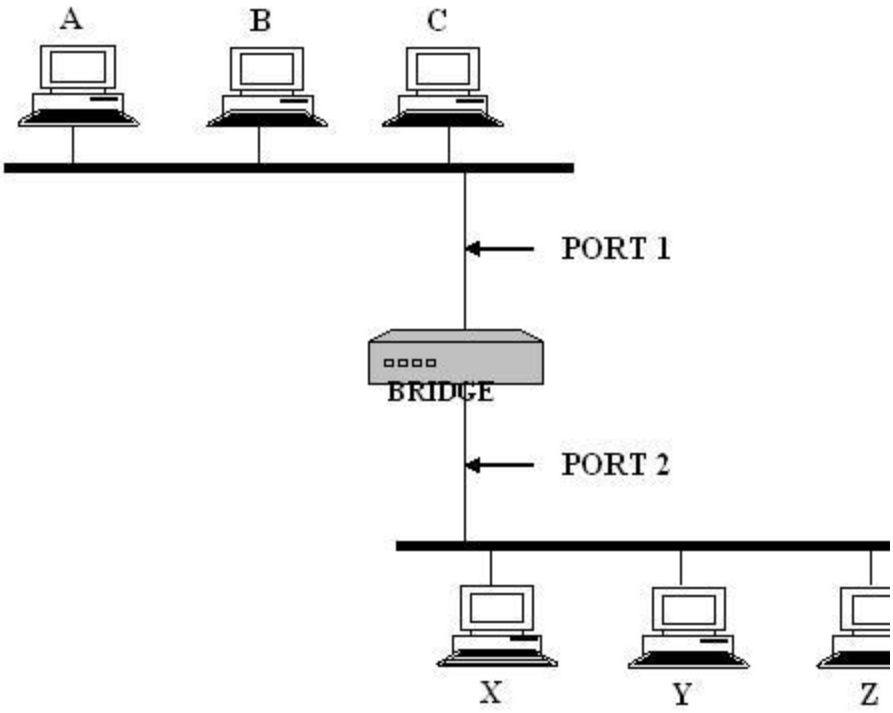
Uses of Bridges

In networks, a bridge is a product that connects a local area network (LAN) to another local area network that uses the same protocol (for example, Ethernetor token ring). You can envision a bridge as being a device that decides whether a message from you to someone else is going to the local area network in your building or to someone on the local area network in the building across the street. A bridge examines each message on a LAN, "passing" those known to be within the same LAN, and forwarding those known to be on the other interconnected LAN (or LANs).

Learning Bridges

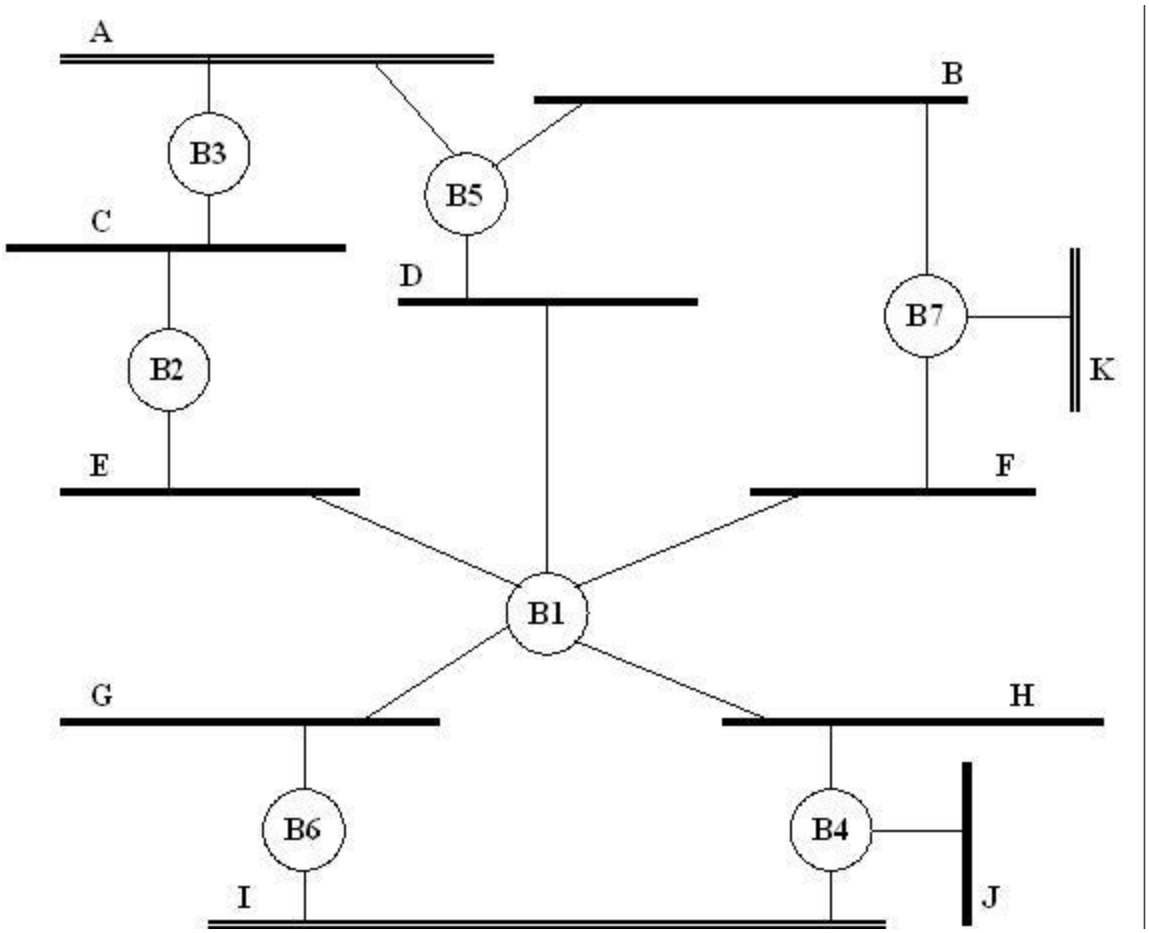
Bridges maintains a forwarding table which contains each host with their port number. Having a human maintain this table is quite a burden, so a bridge can learn this information for itself. The idea is for each bridge to inspect the source address in all the time. Also a timeout is associated with each entry and the bridge is cards the entry after a specified period of time.

HOST	PORT
A	1
B	1
C	1
X	2
Y	2
Z	2



Spanning Tree Bridge

If the extended LAN is having loops then the frames potentially loop through the extended LAN forever. There are two reasons to an extended LAN to have a loop in it. One possibility is that the network is managed by more than one administrator; no single person knows the entire configuration of the network. Second, loops are built in to network on purpose to provide redundancy in case of failure. Bridges must be able to correctly handle loops. This problem is addressed by having the bridges run a distributed spanning tree algorithm.

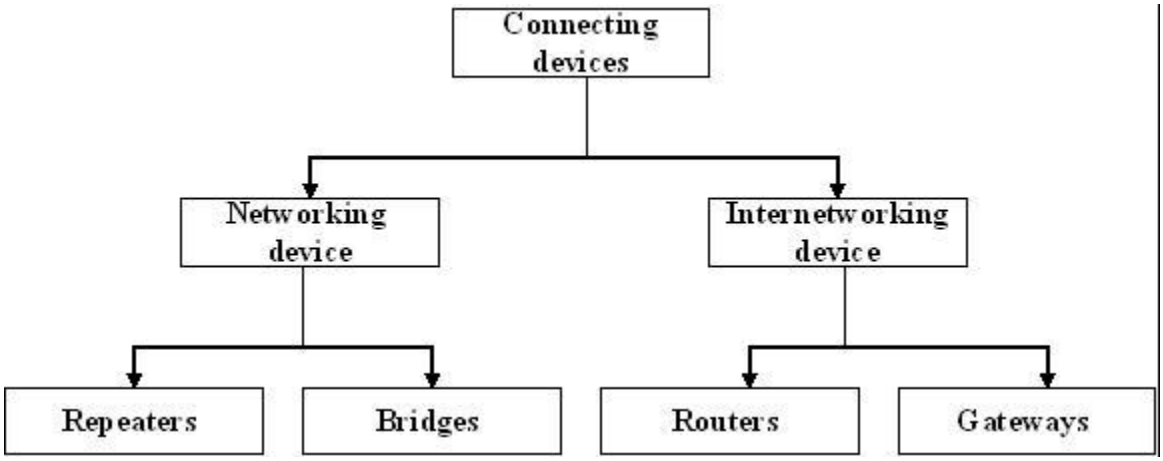


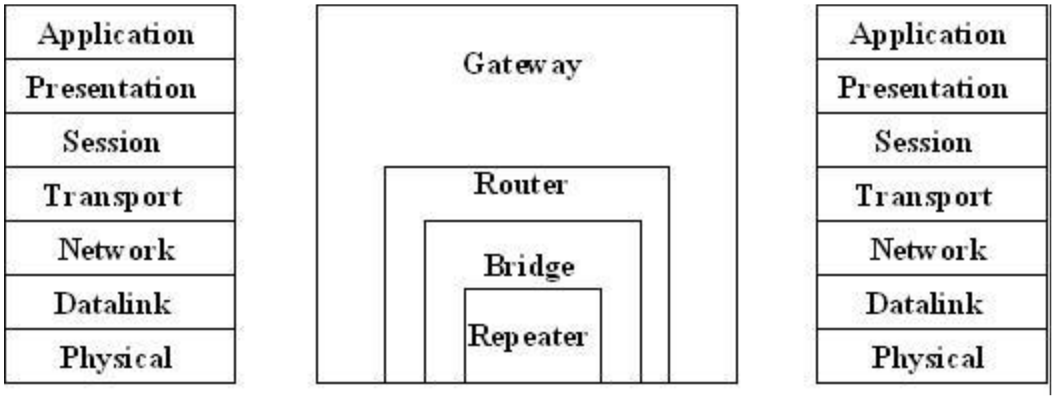
The spanning tree algorithm wad developed by Digital Equipment Corporation. The main idea is for the bridges to select the ports over which they will forward frames. The algorithm selects as follows. Each bridge has a unique identifier. In the above example they are labeled as B1, B2, B3 … the algorithm first elects the bridge with smallest ID as the root of the spanning tree. The root bridge always forwards frames out over all of its ports. Then each bridge computes the shortest path to root and notes which of its ports is on this path. This port is also elected as the bridge's preferred path to the root.

Finally, all the bridges connected to a given LAN elect a single designated bridge that will be responsible for forwarding frames toward the root bridge. Each LANs designated bridge is the one that is closest to the root, and if two or more bridges are equally close to the root, then the bridge which having smallest ID wins. In the above example, B1 is the root bridge since it having the smallest ID. Both B3 and B5 are connected to LAN A, but B5 is the designated bridge since it is closer to the root. Similarly B5 and B7 are connected to LAN B, but B5 is the designated bridge even they are equally closer to the root since B5 having smallest ID.

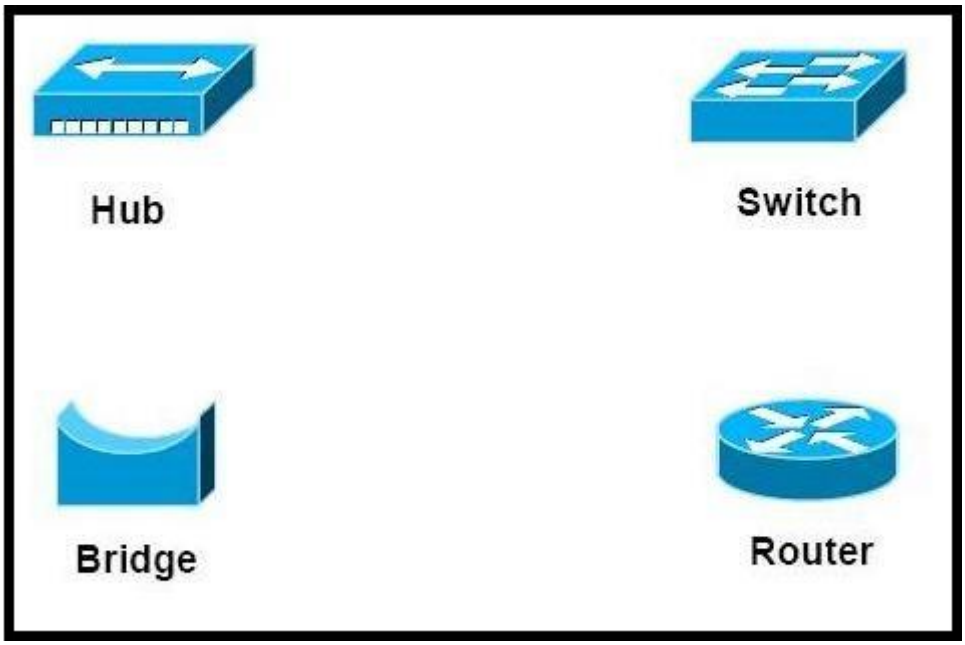
Repeaters , hubs, Bridges, Switsche, Routers and gateways.

Networking and internetworking devices are classified into four categories: repeaters, bridges, routers, and gateways.





Bridges and lan switches:
It is a node that forward frames from one Ethernet to the other. This node would be in promiscuous mode, accepting all frames transmitted on either of the Ethernets, so it could forward them to the other. A bridge is connected between two LANs with port. By using the port number the LANs are addressed. Connected LANs are known as extended LAN



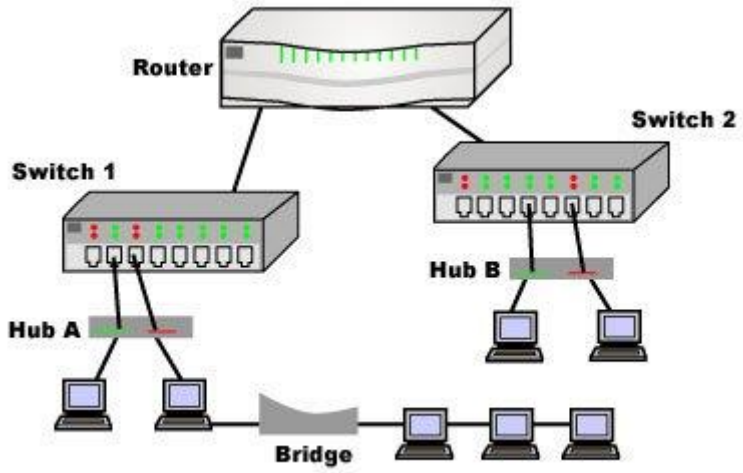
1. Repeater – A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network. An important point to be noted about repeaters is that they do no amplify the signal. When the signal becomes weak, they copy the signal bit by bit and regenerate it at the original strength. It is a 2 port device.

2. Hub – A hub is basically a multiport repeater. A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices. In other words, collision domain of all hosts connected through Hub remains one. Also, they do not have intelligence to find out best path for data packets which leads to inefficiencies and wastage.

3. Bridge – A bridge operates at data link layer. A bridge is a repeater, with add on functionality of filtering content by reading the MAC addresses of source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port, thus making it a 2 port device.

4. Switch – A switch is a multi port bridge with a buffer and a design that can boost its efficiency(large number of ports imply less traffic) and performance. Switch is data link layer device. Switch can perform error checking before forwarding data, that makes it very efficient as it does not forward packets that have errors and forward good packets selectively to correct port only. In other words, switch divides collision domain of hosts, butbroadcast domain remains same.

5. Routers – A router is a device like a switch that routes data packets based on their IP addresses. Router is mainly a Network Layer device. Routers normally connect LANs and WANs together and have a dynamically updating routing table based on which they make decisions on routing the data packets. Router divide broadcast domains of hosts connected through it.



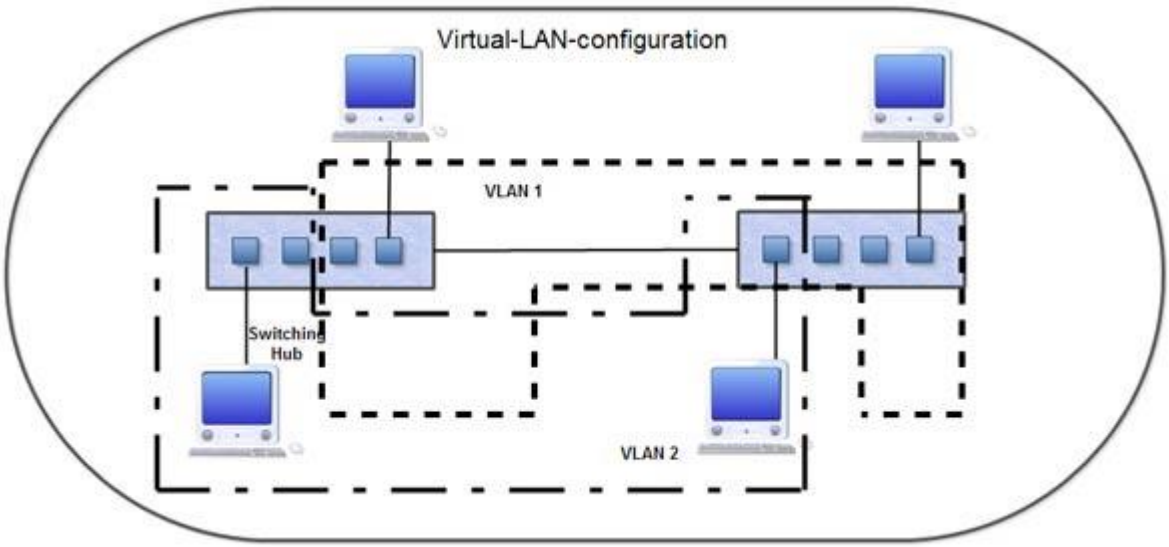
6. Gateway – A gateway, as the name suggests, is a passage to connect two networks together that may work upon different networking models. They basically works as the messenger agents that take data from one system, interpret it, and transfer it to another system. Gateways are also called protocol converters and can operate at any network layer. Gateways are generally more complex than switch or router.

Virtual LAN (V LAN)

VLANs have the primary role to enable easier configuration and management of large corporate networks built around many bridges. There are several implementation strategies for these virtual networks.**Virtual LAN** is software that is employed to provide multiple networks in single hub by grouping terminals connected to switching hubs. It is a LANs that is grouped together by logical addresses into a virtual LAN instead of a physical LAN through a switch. The switch can support many virtual LANs that operate with having different network addresses or as subnets. Users within a virtual LAN are grouped either by IP address or by port address, with each node attached to the switch via a dedicated circuit. Users also can be assigned to more than one virtual LAN.

The VLAN can be defined as a broadcast domain in which the broadcast address reaches all stations belonging to the VLAN. Communications within the VLAN can be secured, and between those two controlled separate

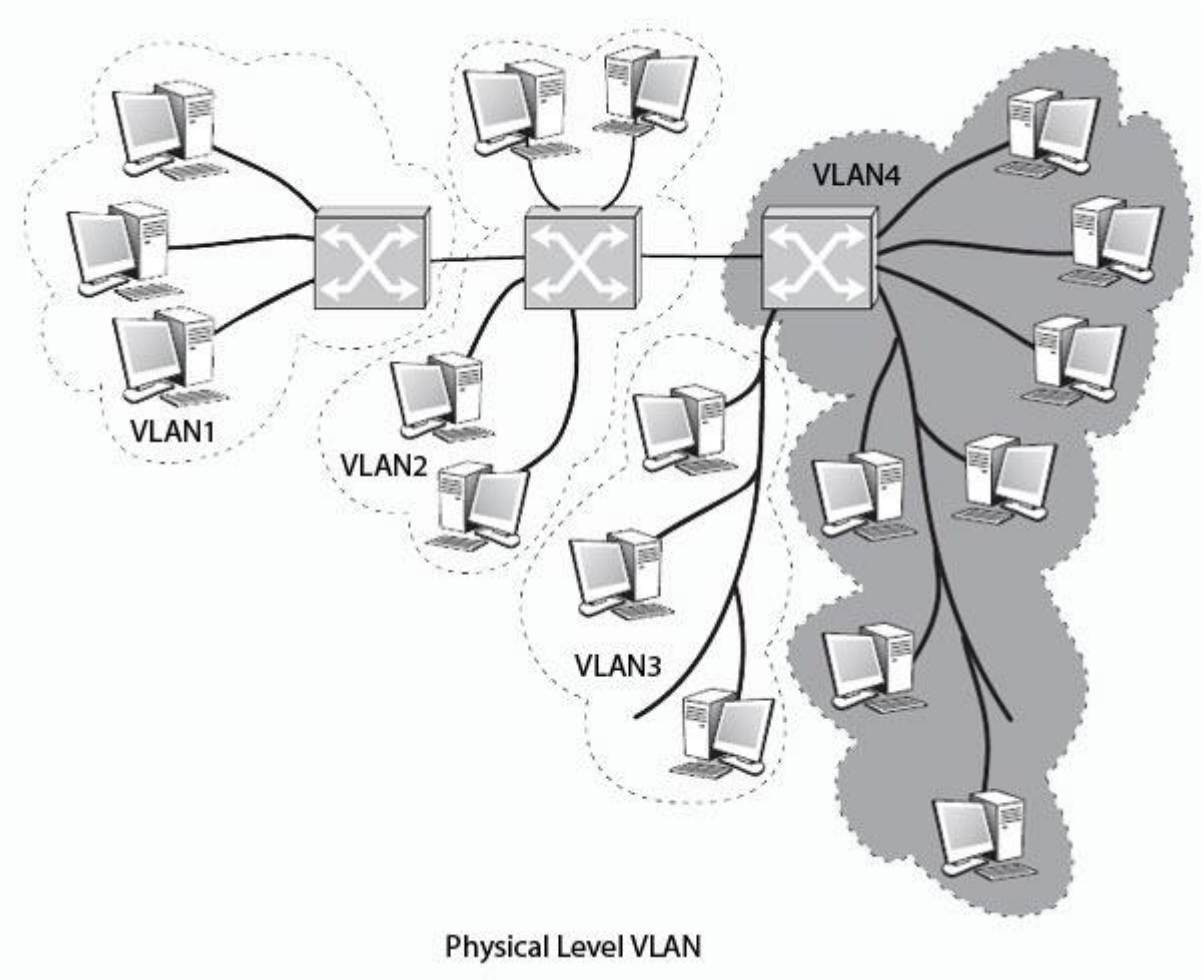
VLANs.



A router is generally required to establish communication between VLANs.

Several types of VLAN are defined according to the groupings of system stations:

• The level of physical or VLAN level 1, which includes stations belonging to the same physical network or multiple physical networks but connected by a common address management. Figure shows a level of VLAN 1.



Characteristics of VLAN

1. Individual VLAN acts as a separate LAN, thus sharing the traffic among VLANs and reducing the congestion
2. Workstations can be provided with full bandwidth at each port
3. Relocation of terminals becomes easy

University Questions:

2 Marks

1.What are the difference between collision free and contention based network protocols?

2.What are Virtual LAN?

11 Marks

1.Briefly discuss in repeaters, Hubs, Bridges, switches and Router.

2.Briefly discuss how CSMA protocol work.

3.Discuss error correction in datalink layer.

4.Explain how spanning tree works**.**

UNIT III

Network layer – design issues – Routing algorithms - The Optimality Principle - Shortest Path Algorithm – Flooding - Distance Vector Routing - Link State Routing - Hierarchical Routing - Broadcast Routing - Multicast Routing Congestion Control – Approaches - Traffic-Aware Routing - Admission Control - Traffic Throttling - Load Shedding – Internetworking - Tunneling - Internetwork Routing - Packet Fragmentation - IP v4 - IP Addresses – IPv6 - Internet Control Protocols – OSPF - BGP

2-MARKS

1. **What are the services offered by network layer?**
   - Logical addressing
   - Routing
2. **What are the methods of packet switching?**
   - Virtual Circuit approach.
   - Datagram approach
3. **Differentiate virtual circuit and datagram's.**
   VC is connection oriented and datagram is connectionless.
4. **What is meant by virtual path?**
   Virtual path is a set of connections between two switches.
5. **What is meant by routing algorithm?**
   The algorithm that manages routing tables and makes the routing decisions is called routing algorithm.
6. **What are the desirable properties of a routing algorithms?**
   - Correctness
   - Simplicity
   - Robustness
   - Stability
   - Fairness
   - Optimality
7. **What are the types of routing algorithms?**
   - Non adaptive routing algorithm
   - Adaptive routing algorithm
8. **Distinguish between adaptive and non adaptive routing algorithms.**
   Non adaptive Routing:
   Once a pathway to a destination has been selected the router sends all packets for that destination along that one route. The routing decisions are not based on the condition or topology of the networks.
   Adaptive Routing:
   Router may select a new route for each packet.(even packets belonging to the same transmission).The routing decisions are based on the condition or topology of the networks.
9. **What are the metrics used by routing protocols?**
   Path length, reliability, delay, bandwidth, load and communication cost.
10. **What is static routing?**
    Static routing is a form of **routing** that occurs when a **router** uses a manually-configured **routing** entry, rather than information from a dynamic **routing** traffic.

11. What is dynamic routing?

Dynamic routing, also called adaptive **routing**, describes the capability of a system, through which routes are characterized by their destination, to alter the path that the route takes through the system in response to a change in conditions.

12. What is adaptive routing algorithm?

Adaptive routing algorithms change their routing decisions to reflect changes in the topology and usually the traffic as well. Distance vector and link state are examples of this.

13. Why is adaptive routing superior to non adaptive routing?

Adaptive routing is superior to non adaptive routing because adaptive routing may select a new route for each packet in response to change in condition and topology of the networks.

14. What is the router's role in controlling the packet lifetime?

As packet is generated, each packet is market with a lifetime; usually the number of hops that are allowed before a packet is considered lost and, accordingly destroyed. As each router encounters the packet subtracts 1 from the total before passing it on. When the lifetime total reaches 0, the packet is destroyed.

15. In routing what does the term SHORTEST mean?

The term Shortest mean the combination of many factors including shortest, cheapest fastest most reliable and so on.

16. Define flooding?

Flooding means that a router sends its information to all of its neighbors and all of its output ports

17. What are the advantages of flooding?

- Simple
- Needs no network information or routing tables
- Robust for failure prone networks.
- Shortest path is always found.

18. What are the most popular routing algorithms?

a. Distance Vector routing
b. Link State routing
c. Hierarchical routing

19. What is Distance vector routing?

Distance vector routing is a simple **routing** protocol used in packet-switched networks that utilizes **distance** to decide the best packet forwarding path. **Distance** is typically represented by the hop count

20. What are the three main elements of distance vector algorithms?

- Knowledge about the entire autonomous system.
- Routing only to neighbours
- Information sharing at regular intervals

21. What are the main disadvantages of distance vector routing?

- Split horizon
- Count to infinity problem

22. What is Link state routing?

Link-state routing protocols, such as OSPF and IS-IS, create a topology of the network and place themselves at the root of the tree. **Link-state** protocols implement an algorithm called the shortest path first (SPF, also known as Dijkstra's Algorithm) to determine the path to a remote destination.

23. What are the three main elements of Link state routing?

- Knowledge about the neighborhood.
- Sharing with every other network.
- Information sharing when there is a change.
- What algorithm does link state routing use to calculate the routing tables.
- Dijkstra algorithm is used to calculate the routing table.

24. What is Hierarchical routing?

Hierarchical routing is a method of **routing** in networks that is based on **hierarchical** addressing.

25. **Explain Multicasting.**

A form of addressing in which a set of computer is assigned one address, a copy of any datagram sent to the address is delivered to each of the computers in the set.

26. Define the term broadcasting.

A form of delivery in which one copy of a packet is delivered to each computer on a network.

27. Why is it that in a broadcast network, the network layer is often thin or even nonexistent?

Network layer is responsible for host to host delivery and for routing the packets through the routers or switches. In broadcast there is no need of addressing the packets, routing and address verification.

28. What is congestion control?

Congestion control is a global issue – involves every router and host within the subnet. Flow **control** – scope is point-to-point; involves just sender and receiver.

29. **What is congestion? Why congestion occurs?**

- In a packet switching network, packets are introduced in the nodes (i.e. *offered load*), and the nodes in-turn forward the packets (i.e. *throughput*) into the network. When the "offered load" crosses certain limit, then there is a sharp fall in the throughput. This phenomenon is known as **congestion**.

  In every node of a packet switching network, queues (or buffers) are maintained to receive and transmit packets (store/forward network) . Due to busty nature of the network traffic there may be situations where there is overflow of the queues. As a result there will be re-transmission of several packets, which further increases the network traffic. This finally leads to **congestion**

30. **What are the two basic mechanisms of congestion control?**

The two basic mechanisms of congestion control are:

a. One is preventive, where precautions are taken so that congestion can not occur.
b. Another is recovery from congestion, when congestion has already taken place

31. **How congestion control is performed by leaky bucket algorithm?**

- In **leaky bucket algorithm,** a buffering mechanism is introduced between the host computer and the network in order to regulate the flow of traffic. Busty traffic are generated by the host computer and introduced in the network by leaky bucket mechanism in the following manner

a. Packets are introduced in the network in one per tick
b. In case of buffer overflow packets are discarded

32. **In what way token bucket algorithm is superior to leaky bucket algorithm?**

The leaky bucket algorithm controls the rate at which the packets are introduced in the network, but it is very conservative in nature. Some flexibility is introduced in token bucket algorithm. In token bucket algorithm tokens are generated at each tick (up to certain limit). For an incoming packet to be transmitted it must capture a token and the transmission takes place at the same rate. Hence some of the busty packets are transmitted at the same rate if tokens are available and thus introduces some amount of flexibility in the system. This also improves the performance.

33. **What is choke packet? How is it used for congestion control?**

Choke packet scheme is a close loop mechanism where each link is monitored to examine how much utilization is taking place. If the utilization goes beyond a certain threshold limit, the link goes to a warning and a special packet, called **choke packet** is sent to the source. On receiving the choke packet, the source reduced the traffic in order to avoid congestion.

The congestion control in the choke packet scheme can be monitored in the following manner.

- Each link is monitored to estimate the level of utilization.
- If the utilization crosses a certain threshold limit, the link goes to a warning state and a choke packet is send to the source.
- On receiving the choke packet, the source reduces the transmitting limit to a certain level (say, by 50%).
- If still warning state persists, more choke packets are sent further reducing the traffic. This continues until the link recovers from the warning state.
- If no further choke packet is received by the source within a time interval, the traffic is increased gradually so that the system doesn't go to congestion state again.

34. What is Admission control?

Admission Control is a validation process in communication systems where a check is performed before a connection is established to see if current resources are sufficient for the proposed connection.

35. What is Traffic Throttling or traffic shapping?

Bandwidth throttling is the intentional slowing of Internet service by an Internet service provider. It is a reactive measure employed in communication *networks* in an apparent attempt to regulate network traffic and *minimize* bandwidth congestion.

36. What is Load Shedding?

Load shedding techniques reduce the load of a system when under severe stress, in the case of *network* monitoring in order to avoid uncontrolled packet loss.

37. What is mean by internetworks?

When two or more networks are connected, they become internetwork or internet. **Internetworking** is the practice of connecting a computer network with other networks through the use of gateways that provide a common method of routing information packets between the networks. The resulting system of interconnected networks is called an **internetwork**, or simply an internet.

38. Define Tunneling?

Tunneling, also known as "port forwarding," is the transmission of data intended for use only within a private, usually corporate network through a public network in such a way that the routing nodes in the public network are unaware that the transmission is part of a private network.

39. What is internetwork routing?

The number assigned to a single network in an *internetwork*. Network addresses are used by hosts and *routers* when *routing* a packet from a source to a destination in an *internetwork*. Host address. Also known as a host ID or a node ID.

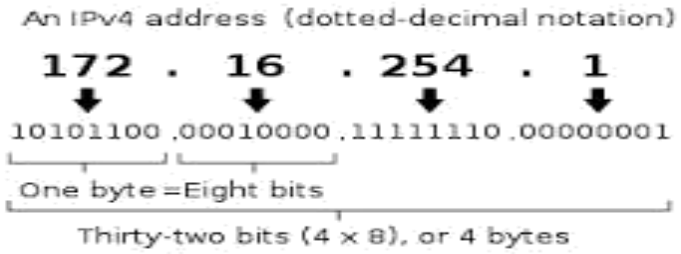
40. What is packet fragmentation?

The Internet Protocol (IP) implements datagram **fragmentation**, breaking it into smaller pieces, so that **packets** may be formed that can pass through a link with a smaller maximum transmission unit (**MTU**) than the original datagram size.

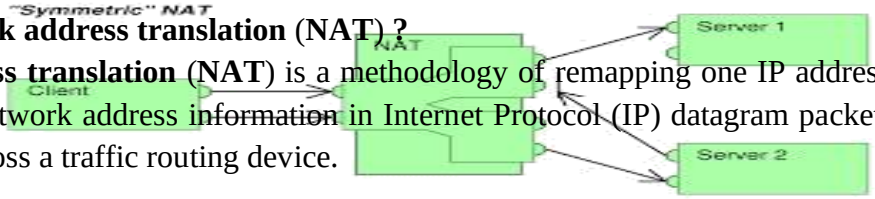
41. What is IPV4?

Internet Protocol version 4 (**IPv4**) is the fourth version in the development of the Internet Protocol (IP).

It is one of the core protocols of standards-based internetworking methods in the Internet, and was the first version deployed for production in the ARPANET in 1983.



42. What is Network address translation (NAT) ?



Network address translation (**NAT**) is a methodology of remapping one IP address space into another by modifying network address information in Internet Protocol (IP) datagram packet headers while they are in transit across a traffic routing device.

43. What is IPV6?

IPv6 is the Internet's next-generation protocol, designed to replace the current Internet Protocol, IP Version 4. In order to communicate over the Internet, computers and other devices must have sender and receiver addresses. These numeric addresses are known as Internet Protocol addresses.

44. What is an Internet Protocol (IP)?

The protocol that defines both the format of packets used on a TCP/IP internet and the mechanisms for routing a packet to its destination.

45. What is an IP address?

An IP address is a 32 - bit address that uniquely and universally define the connection of a host or a router to the Internet. The sender must know the IP address of the destination computer before sending a packet.

46. What are the categories of IP addresses?

IP addresses were divided into five categories as follows.

- Class A
- Class B
- Class C
- Class D
- Class E

47. Discuss the class field in IP address.

If the address is given in binary notation, the first few bits can tell us the class of the address.

- Class A - 0
- Class B - 10
- Class C - 110
- Class D - 1110
- Class E - 1111

When the address is given in dotted decimal notation, then look at the first byte to determine the class of the address..

- Class A - 0 to 127
- Class B – 128 to 191
- Class C – 192 to 223
- Class D – 224 to 239
- Class E – 240 to 255

48. What is a host-id and net-id?

Net-id – The portion of the IP address that identifies the network called the netid. Host-id – The portion of the IP address that identifies the host or router on the network is called the hostid.

49. How does a netid differ from a network address?

A network address has both netid and hostid with 0's for the hostid.

50. What is the purpose of subnetting?

- When we divide a network into several subnets, we have three levels of hierarchy.
- The netid is the first level, defines the site.
- The subnetid is the 2nd level, defines the physical subnetwork.
- The hostid is the 3rd level defines the connection of the host to the subnetwork.

51. What are the benefits of subnetting a network?

- Reduced network traffic
- Optimized network performance
- Simplified network management
- Facilities spanning large geographical distance.

52. Define Masking.

Masking is a process that extracts the address of the physical network from an IP address.

53. What is the difference between boundary level masking and non-boundary level masking.

Boundary level Masking:

- If the masking is at the boundary level, the mask numbers are either 255 or 0, finding the subnetwork address is very easy.

Non Boundary level Masking:

- If the masking is not at the boundary level, the mask numbers are not just 255 or 0, finding the subnetwork address involves using the bitwise AND operators.

54. What is the class of each of the following addresses?

- 10011101 10001111 11111100 11001111 – Class B
- 11011101 10001111 11111100 11001111 – Class C
- 01111011 10001111 11111100 11001111 – Class A
- 11101011 10001111 11111100 11001111 – Class D
- 11110101 10001111 11111100 11001111 – Class E

55. **Find the class of each address**.

- 4.23.145.90 – Class A
- 227.34.78.7 – Class D
- 246.7.3.8 – Class E
- 29.6.8.4 – Class A
- 198.76.9.23 – Class C

56. What is Supernetting?

Supernetting combines several networks into one lager one.

57. Identify the class and default subnet mask of the IP address 217.65.10.7.

It belongs to class C.

Default subnet mask – 255.255.255.192

58. What are the fields present in IP address?

Netid and Hostid.

Netid – portion of the ip address that identifies the network.

Hostid – portion of the ip address that identifies the host or router on the networks.

59. What is the time to live field in IP header?

Time to live field is counter used to limit packet lifetimes counts in second and default value is 255 sec.

60. Identify the class and default subnet mask of the IP address 217.65.10.7

IP address 217.65.10.7 belongs to class C address and default subnet mask is 255.255.255.0.

61. What is internet control Message protocol?

The **Internet Control** Message **Protocol** (ICMP) is one of the main **protocols** of the **Internet Protocol** Suite. It is used by network devices, like routers, to send error messages indicating, for example, that a requested service is not available or that a host or router could not be reached.

62. What is address resolution?

Address resolution is a process of obtaining the physical address of a computer based on its IP address, in order to be able to finally actually transmit the frame or datagrams over the network to which the node belongs.

63. What is OSPF?

Open Shortest Path First (**OSPF**) is a routing protocol for Internet Protocol (IP) networks. It uses a link state routing algorithm and falls into the group of interior routing protocols, operating within a single autonomous system (AS)

64. What is BGP?

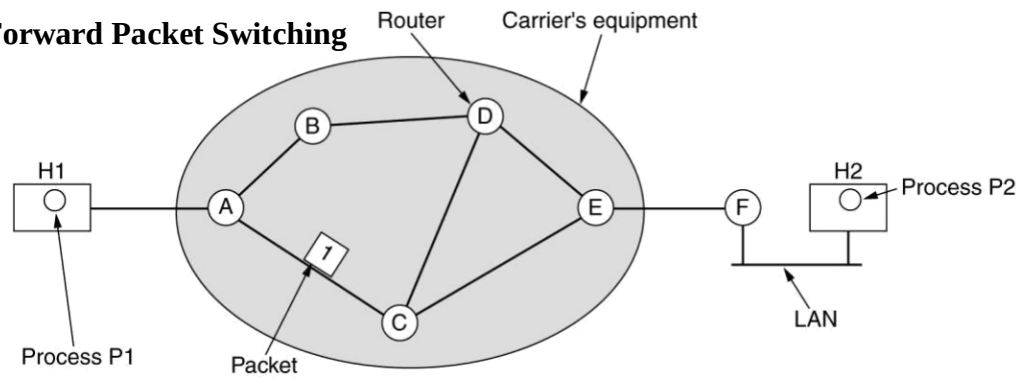
Border Gateway Protocol (**BGP**) is a standardized exterior gateway protocol designed to exchange routing and reachability information between autonomous systems (AS) on the Internet. The protocol is often classified as a path vector protocol but is sometimes also classed as a distance-vector routing protocol.

11-MARKS

1. Discuss about Design issues of Network layer

- Store-and-Forward Packet Switching
- Services Provided to the Transport Layer
- Implementation of Connectionless Service
- Implementation of Connection-Oriented Service
- Comparison of Virtual-Circuit and Datagram Subnets

Store-and-Forward Packet Switching



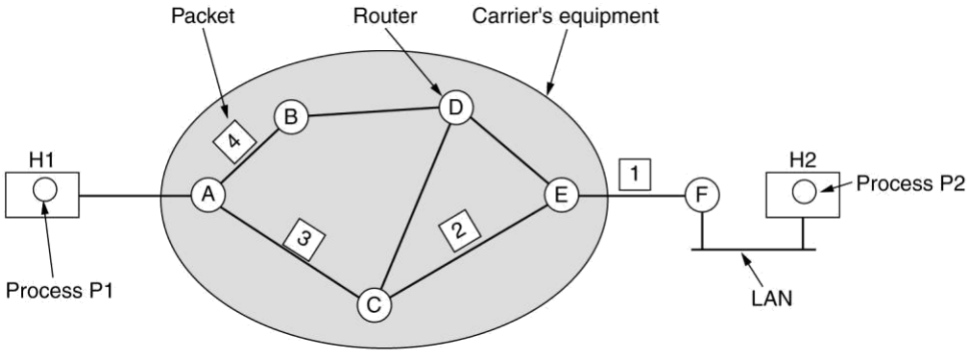
- A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the carrier.
- The packet is stored there until it has fully arrived so the checksum can be verified. Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered. This mechanism is store-and-forward packet switching.

Services Provided to the Transport Layer

- The services should be independent of the router technology.
- The transport layer should be shielded from the number, type, and topology of the routers present.
- The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

Implementation of Connectionless Service

- Two different organizations are possible, depending on the type of service offered.
- If connectionless service is offered, packets are injected into the subnet individually and routed independently of each other. No advance setup is needed. In this context, the packets are frequently called datagrams and the subnet is called a datagram subnet.
- If connection-oriented service is used, a path from the source router to the destination router must be established before any data packets can be sent. This connection is called a VC (virtual circuit) and the subnet is called a virtual-circuit subnet.



A's table

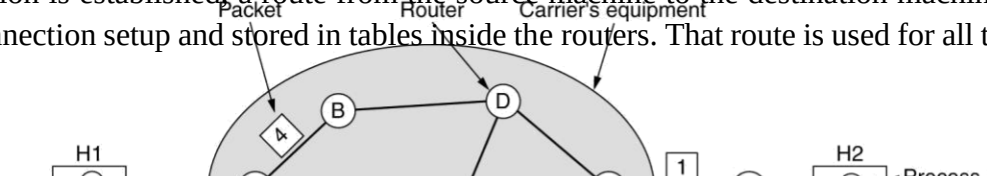
Dest. Line	initially	later
A	–	–
B	B	B
C	C	C
D	B	B
E	C	B
F	C	B

C's table	
A	A
B	A
C	–
D	D
E	E
F	E

E's table	
A	C
B	D
C	C
D	D
E	–
F	F

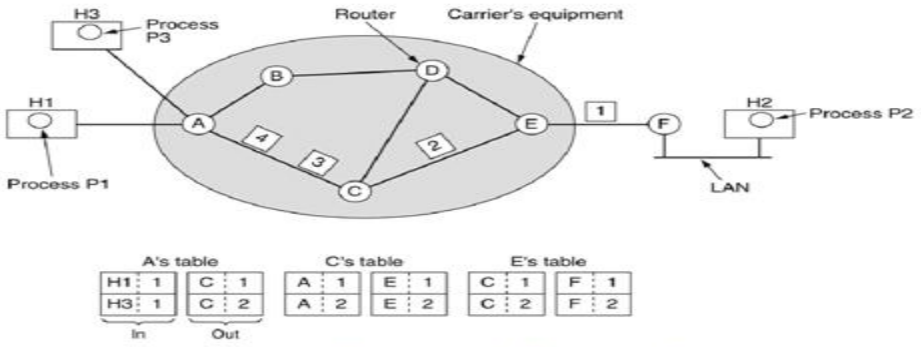
Implementation of Connection-Oriented Service

- For connection-oriented service, we need a virtual-circuit subnet.
- The idea behind virtual circuits is to avoid having to choose a new route for every packet sent. Instead,when a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers. That route is used for all traffic flowing over



the connection, exactly the same way that the teleph-one system works. When the connection is released, the virtual circuit is also terminated.

- With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to .



Routing within a virtual-circuit subnet

Comparison of Virtual-Circuit and Datagram Subnets

Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

- Inside the subnet, several trade-offs exist between virtual circuits and datagram's.
- One trade-off is between router memory space and bandwidth.
- Virtual circuits allow packets to contain circuit numbers instead of full destination addresses. If the packets tend to be fairly short, a full destination address in every packet may represent a significant amount of overhead and hence, wasted bandwidth.
- The price paid for using virtual circuits internally is the table space within the routers. Depending upon the

relative cost of communication circuits versus router memory, one or the other may be cheaper.

- Another trade-off is setup time versus address parsing time.
- Using virtual circuits requires a setup phase, which takes time and consumes resources. However, figuring out what to do with a data packet in a virtual-circuit subnet is easy: the router just uses the circuit number to index into a table to find out where the packet goes.
- In a datagram subnet, a more complicated lookup procedure is required to locate the entry for the destination.
- Virtual circuits have some advantages in guaranteeing quality of service and avoiding congestion within the subnet.
- The loss of a communication line is fatal to virtual circuits using it but can be easily compensated for if datagrams are used. Datagrams also allow the routers to balance the traffic throughout the subnet, since routes can be changed partway through a long sequence of packet transmissions.

2. Discuss about Various Routing Algorithms

Definition: The routing algorithm is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on.

Properties of routing algorithm:

- correctness,
- simplicity,
- robustness,
- stability,
- fairness,
- Optimality.

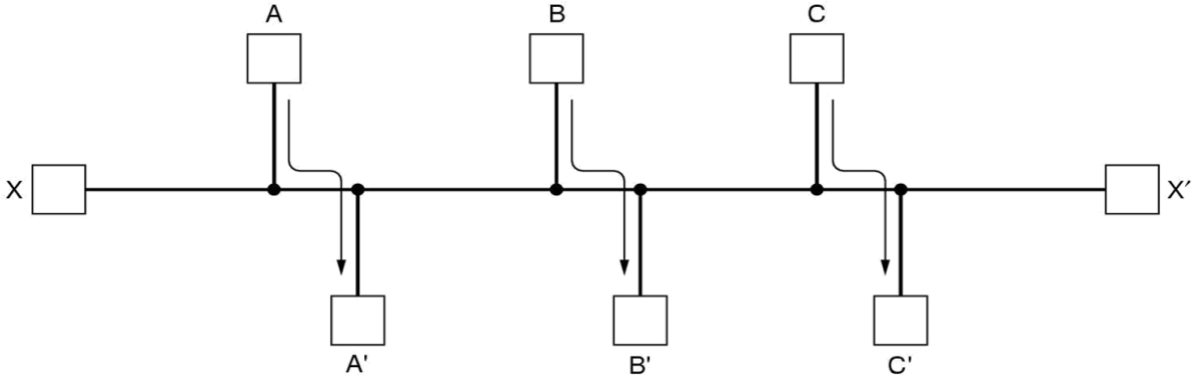
Robustness:

- Once a major network comes on the air, it may be expected to run continuously for years without system-wide failures. During that period there will be hardware and software failures of all kinds. Hosts, routers, and lines will fail repeatedly, and the topology will change many times.
- The routing algorithm should be able to cope with changes in the topology and traffic without requiring all jobs in all hosts to be aborted and the network to be rebooted every time some router crashes**.**

Stability:

- It is also an important goal for the routing algorithm. There exist routing algorithms that never converge to equilibrium, no matter how long they run. A stable algorithm reaches equilibrium and stays there.

Fairness and optimality may sound obvious, but as it turns out, they are often contradictory goals.

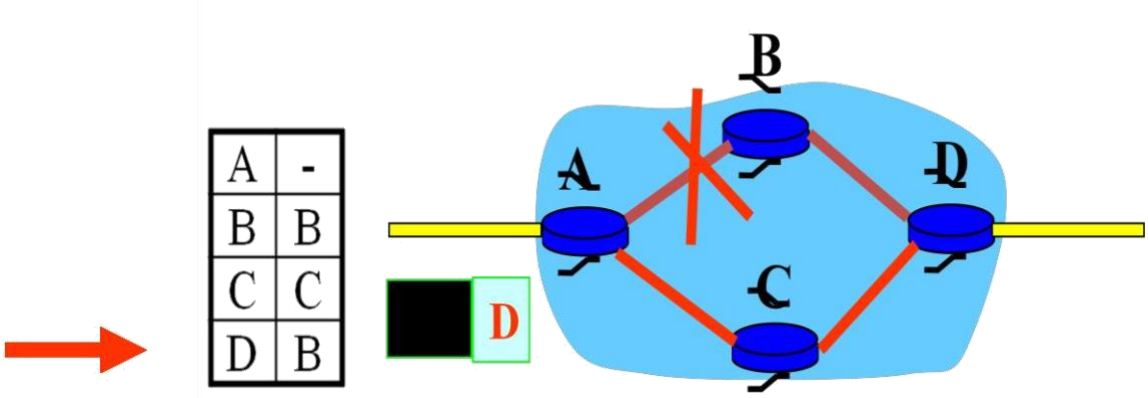


- There is enough traffic between A and A', between B and B', and between C and C' to saturate the horizontal links. To maximize the total flow, the X to X' traffic should be shut off altogether. Unfortunately, X and X' may not see it that way. Evidently, some compromise between global efficiency and fairness to individual connections is needed.

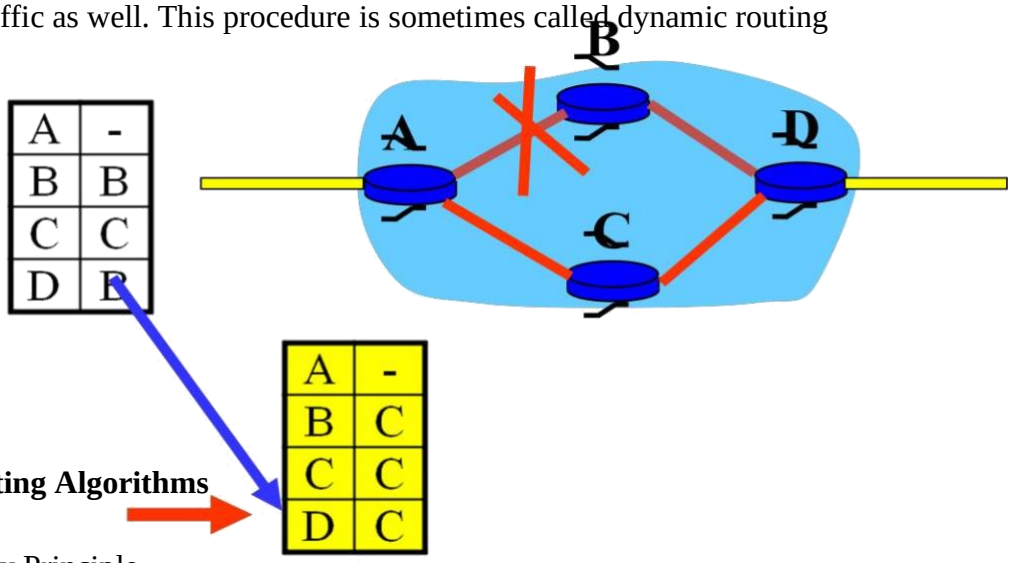
Category of algorithm

- Nonadaptive
- Adaptive

Nonadaptive algorithms do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J is computed in advance, off-line, and downloaded to the routers when the network is booted**.** This procedure is sometimes called static routing**.**



Adaptive algorithms, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well. This procedure is sometimes called dynamic routing



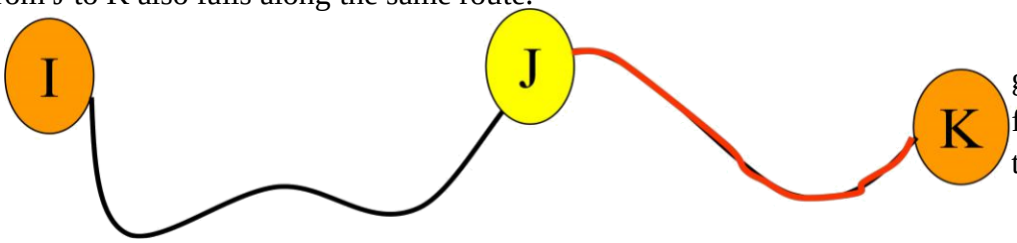
Various Routing Algorithms

- The Optimality Principle
- Shortest Path Routing
- Flooding

- Distance Vector Routing
- Link State Routing
- Hierarchical Routing
- Broadcast Routing
- Multicast Routing

OPTIMALITY PRINCIPLE:

The Optimality Principle: if router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route.



given destination form a tree rooted at the **sink tree**.

The set of optimal routes from destination. Such a tree is called a all sources to a

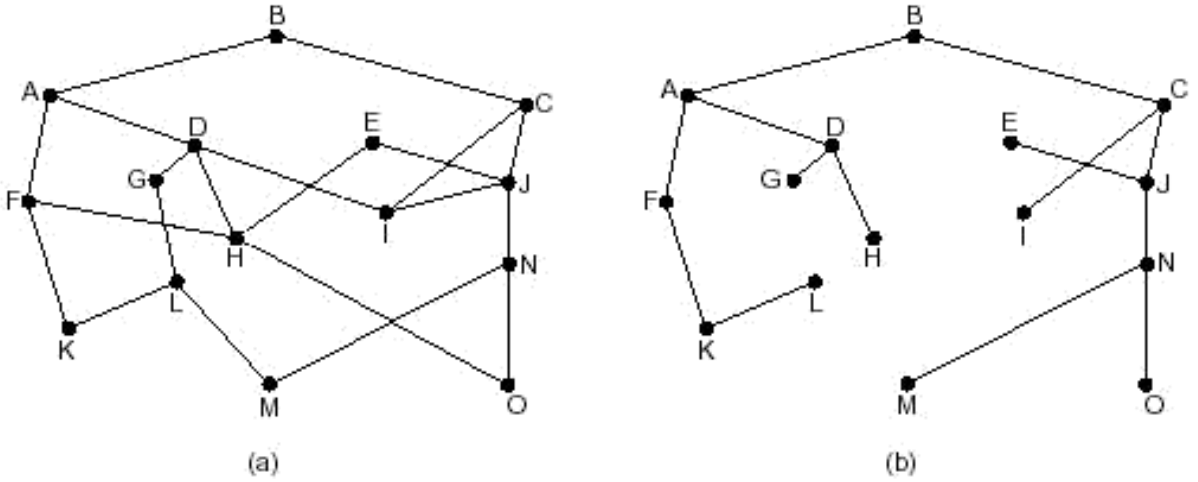


Figure (a) A subnet. (b) A sink tree for router B

- Note: A sink tree is not necessarily unique; other trees with the same path lengths may exist.
- The goal of all routing algorithms is to discover and use the sink trees for all routers.
- A technique to study routing algorithms: The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line (often

called a link).

- To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.
- One way of measuring path length is the number of hops. Another metric is the geographic distance in kilometers . Many other metrics are also possible. For example, each arc could be labeled with the mean queuing and transmission delay for some standard test packet as determined by hourly test runs.
- In the general case, the labels on the arcs could be computed as a function of the distance, bandwidth, average traffic, communication cost, mean queue length, measured delay, and other factors. By changing the weighting function, the algorithm would then compute the "shortest" path measured according to any one of a number of criteria or to a combination of criteria.

SHORTEST PATH ALGORITHM (Dijkstra algorithm)

- Each node is labeled (in parentheses) with its distance from the source node along the best known path. Initially, no paths are known, so all nodes are labeled with infinity.
- As the algorithm proceeds and paths are found, the labels may change, reflecting better paths.
- A label may be either **tentative** or **permanent**. Initially, all labels are tentative. When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter.

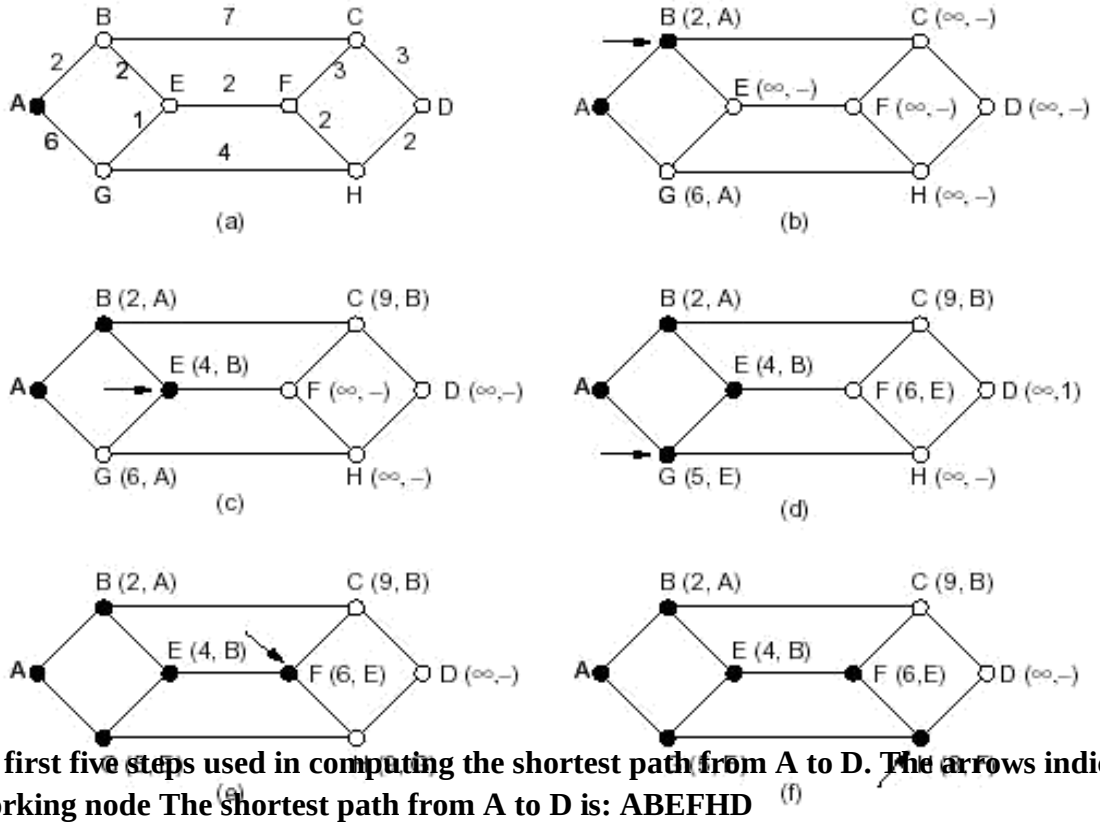


Figure. The first five steps used in computing the shortest path from A to D. The arrows indicate the working node The shortest path from A to D is: ABEFHD

```
#define MAX_NODES 1024                  /* maximum number of nodes */
#define INFINITY 1000000000             /* a number larger than every maximum path */
int n, dist[MAX_NODES][MAX_NODES];/* dist[i][j] is the distance from i to j */

void shortest_path(int s, int t, int path[])
{ struct state {                        /* the path being worked on */
      int predecessor;                  /* previous node */
      int length;                       /* length from source to this node */
      enum {permanent, tentative} label; /* label state */
  } state[MAX_NODES];

  int i, k, min;
  struct state *p;

  for (p = &state[0]; p < &state[n]; p++) { /* initialize state */
      p->predecessor = -1;
      p->length = INFINITY;
      p->label = tentative;
  }
  state[t].length = 0;  state[t].label = permanent;
  k = t;                                /* k is the initial working node */
  do {                                  /* Is there a better path from k? */
      for (i = 0; i < n; i++)           /* this graph has n nodes */
          if (dist[k][i] != 0 && state[i].label == tentative) {
              if (state[k].length + dist[k][i] < state[i].length) {
                  state[i].predecessor = k;
                  state[i].length = state[k].length + dist[k][i];
              }
          }

      /* Find the tentatively labeled node with the smallest label. */
      k = 0; min = INFINITY;
      for (i = 0; i < n; i++)
          if (state[i].label == tentative && state[i].length < min) {
              min = state[i].length;
              k = i;
          }
      state[k].label = permanent;
  } while (k != s);

  /* Copy the path into the output array. */
  i = 0;  k = s;
  do {path[i++] = k; k = state[k].predecessor; } while (k >= 0);
}
```

FLOODING ALGORITHM

- Every incoming packet is sent out on every outgoing line except the one it arrived on.

- Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process.

o **One such measure** is to have a hop counter contained in the header of each packet, which is decremented at each hop, with the packet being discarded when the counter reaches zero.

o **An alternative technique** for damming the flood is to keep track of which packets have been flooded, to avoid sending them out a second time.

- A variation of flooding that is slightly more practical is **selective flooding**.In this algorithm the routers do not send every incoming packet out on every line, only on those lines that are going approximately in the right direction.

- Applications of flooding algorithm:

1. military applications
2. distributed database applications
3. wireless networks
4. as a metric against which other routing algorithms can be compared

DISTANCE VECTOR ROUTING

- A dynamic routing algorithm

- Distance vector routing algorithms operate by having each router maintain a table (i.e, a vector) giving the best known distance to each destination and which line to use to get there. These tables are updated by exchanging information with the neighbors. (also named the distributed Bellman-Ford routing algorithm and the Ford-Fulkerson algorithm)

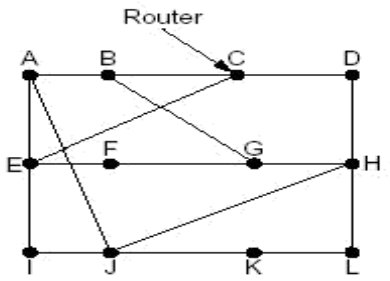
- Table content: In distance vector routing, each router maintains a routing table indexed by, and containing one entry for, each router in the subnet. This entry contains two parts: the preferred outgoing line to use for that destination and an estimate of the time or distance to that destination.

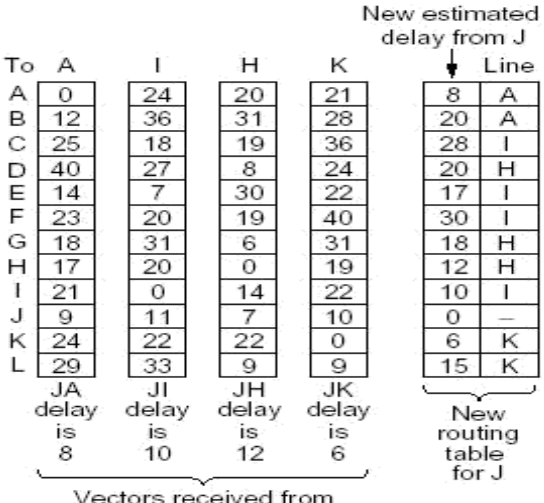
- Table updating method:

  - Assume that the router knows the delay to each of its neighbors. Once every T msec each router sends to each neighbor a list of its estimated delays to each destination. It also receives a similar list from each neighbor.

  - Imagine that one of these tables has just come in from neighbor X, with Xi being X's estimate of how long it takes to get to router i. If the router knows that the delay to X is m msec, it also knows that it can reach router i via X in Xi + m msec.

  - By performing this calculation for each neighbor, a router can find out which estimate seems the best and use that estimate and the corresponding line in its new routing table. Note that the old routing table is not used in the calculation.





Part (a) shows a subnet. The first four columns of part (b) show the delay vectors received from the neighbors of router J. Suppose that J has measured or estimated its delay to its neighbors, A, I, H, and K as 8, 10, 12, and 6 msec, respectively.

LINK STATE ROUTING

- A dynamic routing algorithm

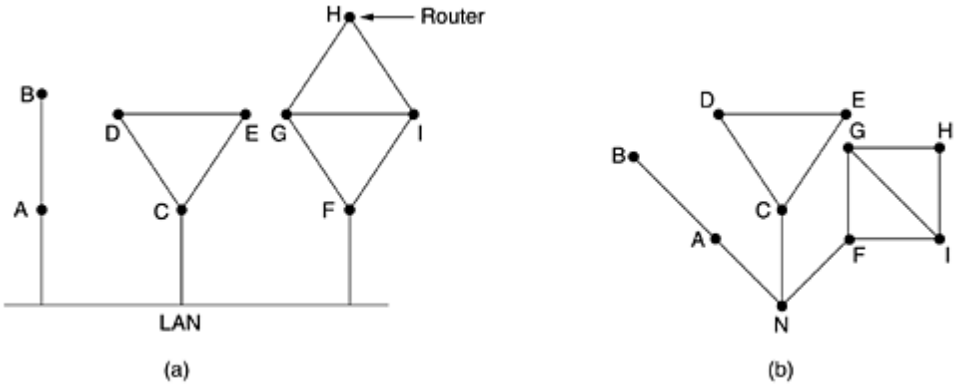
- The idea behind link state routing can be stated as five parts. Each router must do the following:
  a. Discover its neighbors and learn their network addresses.
  b. Measure the delay or cost to each of its neighbors.
  c. Construct a packet telling all it has just learned.
  d. Send this packet to all other routers.
  e. Compute the shortest path to every other router.
- In effect, the complete topology and all delays are experimentally measured and distributed to every router. Then Dijkstra's algorithm can be run to find the shortest path to every other router.

Learning about the Neighbors

- It accomplishes this goal by sending a special HELLO packet on each point-to-point line. The router on the other end is expected to send back a reply telling who it is.

- These names must be globally unique because when a distant router later hears that three routers are all connected to F, it is essential that it can determine whether all

three mean the same F.



Measuring Line Cost

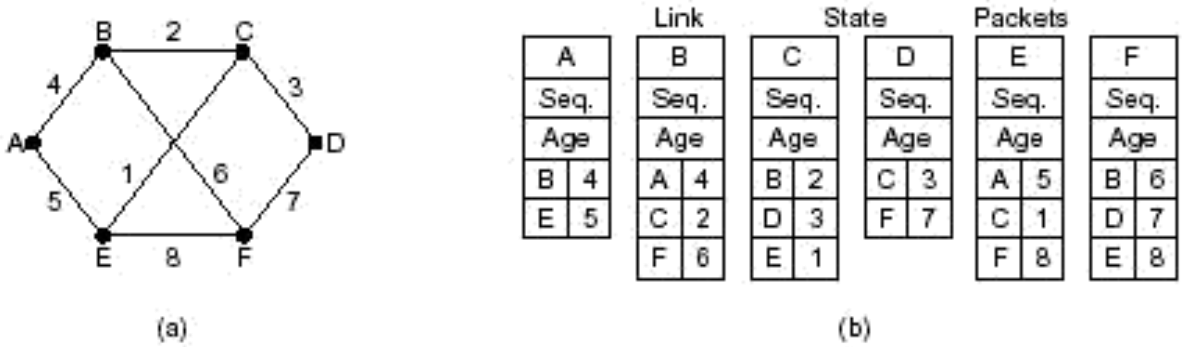
- The most direct way to determine this delay is to send over the line a special ECHO packet that the other side is required to send back immediately.

- By measuring the round-trip time and dividing it by two, the sending router can get a reasonable estimate of the delay.

- For even better results, the test can be conducted several times, and the average used. Of course, this method implicitly assumes the delays are symmetric, which may not always be the case.

Building Link State Packets

- The packet starts with the identity of the sender, followed by a sequence number and age (to be described later), and a list of neighbors. For each neighbor, the delay to that neighbor is given**.**



(a) subnet. (b) The link state packets for this subnet

- Building the link state packets is easy. The hard part is determining when to build them. o One possibility is to build them periodically, that is, at regular intervals.

o Another possibility is to build them when some significant event occurs, such as a line or neighbor going down or coming back up again or changing its properties appreciably.

Distributing the Link State Packets

- The basic distribution algorithm:The fundamental idea is to use flooding to distribute the link state packets.

- To keep the flood in check, each packet contains a sequence number that is incremented for each new packet sent. Routers keep track of all the (source router, sequence) pairs they see.

- When a new link state packet comes in, it is checked against the list of packets already seen.

  o If it is new, it is forwarded on all lines except the one it arrived on. o If it is a duplicate, it is discarded.

o If a packet with a sequence number lower than the highest one seen so far ever arrives, it is rejected as being obsolete since the router has more recent data.

- First problem with this algorithm: if the sequence numbers wrap around, confusion will reign. The solution here is to use a 32-bit sequence number. With one link state packet per second, it would take 137 years to wrap around, so this possibility can be ignored.

- Second problem: if a router ever crashes, it will lose track of its sequence number. If it starts

again at 0, the next packet will be rejected as a duplicate.

- Third problem :if a sequence number is ever corrupted and 65,540 is received instead of 4 (a 1-bit error), packets 5 through 65,540 will be rejected as obsolete, since the current sequence number is thought to be 65,540.

- The solution to all these problems is to include the age of each packet after the sequence number and decrement it once per second. When the age hits zero, the information from that router is discarded.

			Send flags			ACK flags			
Source	Seq.	Age	A	C	F	A	C	F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

Computing the New Routes

- Once a router has accumulated a full set of link state packets, it can construct the entire subnet graph because every link is represented.

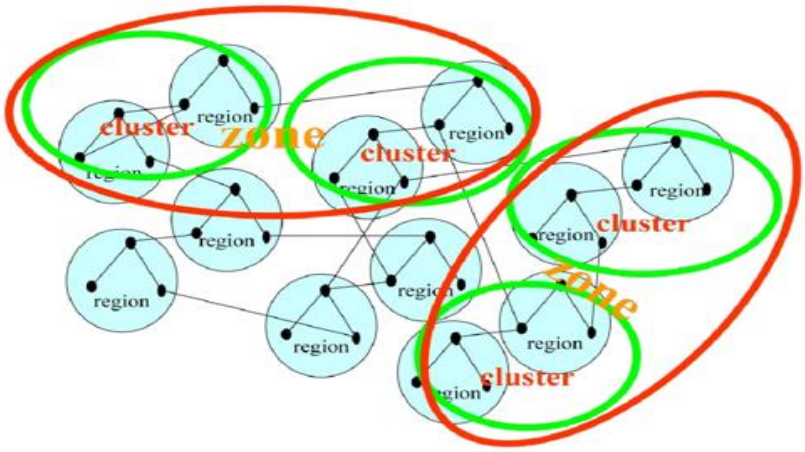
- Now Dijkstra's algorithm can be run locally to construct the shortest path to all possible destinations.

HIERARCHIAL ROUTING

- The routers are divided into what we will call regions, with each router knowing all the details about how to route packets to destinations within its own region, but knowing nothing about the internal structure of other regions.

- For huge networks, a two-level hierarchy may be insufficient; it may be necessary to group the regions into clusters, the clusters into zones, the zones into groups, and so on, until we run out of names for aggregations.

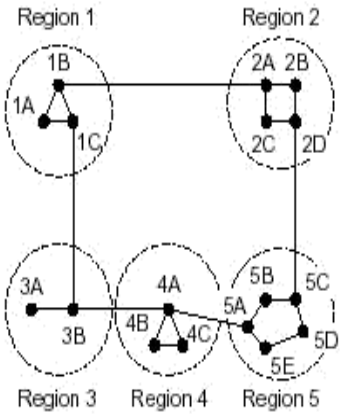
-



- The full routing table for router 1A has 17 entries, as shown in (b).

- When routing is done hierarchically, as in (c), there are entries for all the local routers as before, but all other regions have been condensed into a single router, so all traffic for region 2 goes via the 1B -2A line, but the rest of the remote traffic goes via the 1C - 3B line.
- Hierarchical routing has reduced the table from 17 to 7 entries.

\



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

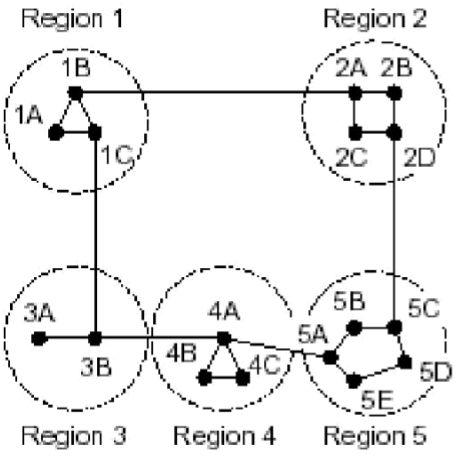
Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

- Unfortunately, these gains in space are not free. There is a penalty to be paid, and this penalty is in the form of increased path length

- For example, the best route from 1A to 5C is via region 2, but with hierarchical routing all traffic to region 5 goes via region 3, because that is better for most destinations in region 5



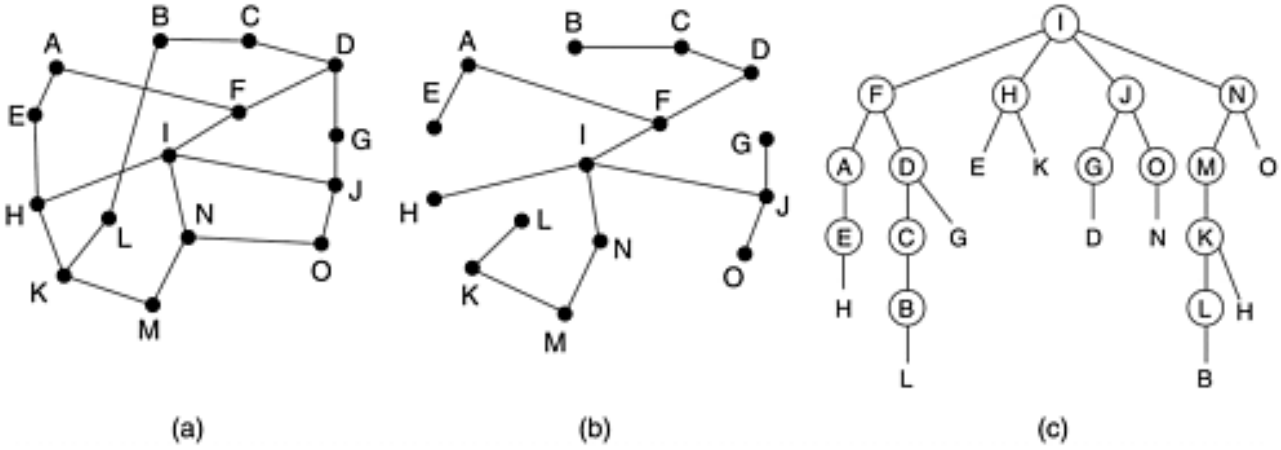
Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

BROADCAST ROUTING

Sending a packet to all destinations simultaneously is called broadcasting.
The source simply sends a distinct packet to each destination. Not only is the method wasteful of bandwidth, but it also requires the source to have a complete list of all destinations.
The problem with flooding as a broadcast technique is that it generates too many packets and consumes too much bandwidth.

Reverse path forwarding. (a) A subnet. (b) A sink tree. (c) The tree built by reverse path forwarding.

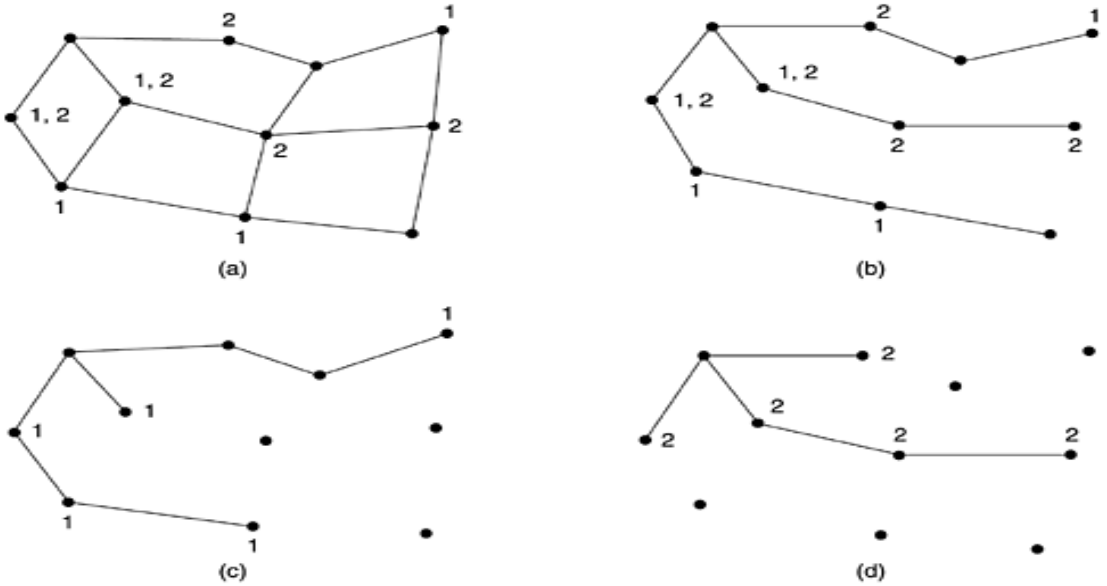


Part (a) shows a subnet, part (b) shows a sink tree for router *I* of that subnet, and part (c) shows how the reverse path algorithm works.

- When a broadcast packet arrives at a router, the router checks to see if the packet arrived on the line that is normally used for sending packets to the source of the broadcast. If so, there is an excellent chance that the broadcast packet itself followed the best route from the router and is therefore the first copy to arrive at the router.

- This being the case, the router forwards copies of it onto all lines except the one it arrived on. If, however, the broadcast packet arrived on a line other than the preferred one for reaching the source, the packet is discarded as a likely duplicate.

MULTICAST ROUTING



- To do multicast routing, each router computes a spanning tree covering all other routers. For example, in Fig. 5-17(a) we have two groups, 1 and 2.
- Some routers are attached to hosts that belong to one or both of these groups, as indicated in the figure.
- A spanning tree for the leftmost router is shown in Fig. 5-17(b). When a process sends a multicast packet to a group, the first router examines its spanning tree and prunes it, removing all lines that do not lead to hosts that are members of the group.
- In our example, Fig. 5-17(c) shows the pruned spanning tree for group 1. Similarly, Fig. 5-17(d) shows the pruned spanning tree for group 2. Multicast packets are forwarded only along the appropriate spanning tree.

3.Explain in detail about Congestion Control Algorithms and its approaches

Too many packets present in (a part of) the network causes packet delay and loss that degrades performance. This situation is called **congestion**.

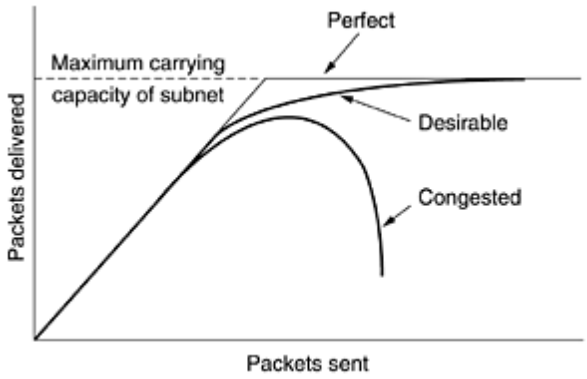
- **Approaches to Congestion Control**
    - **Congestion Prevention Policies**
- **Traffic-Aware Routing**
- **Admission Control**
- **Traffic Throttling**
    - **Choke Packets**
    - **Explicit Congestion Notification**
    - **Hop by Hop back Pressure**
- **Load Shedding**
    - **Random Early Detection**

- When too many packets are present in (a part of) the subnet, performance degrades. This situation is called **congestion**.

- Figure 5-25 depicts the symptom. When the number of packets dumped into the subnet by the hosts is within its carrying capacity, they are all delivered (except for a few that are afflicted with transmission errors) and the number delivered is proportional to the number sent.

- However, as traffic increases too far, the routers are no longer able to cope and they begin losing packets. This tends to make matters worse. At very high traffic, performance collapses completely and almost no packets are delivered.

Figure 5-25. When too much traffic is offered, congestion sets in and performance degrades sharply.



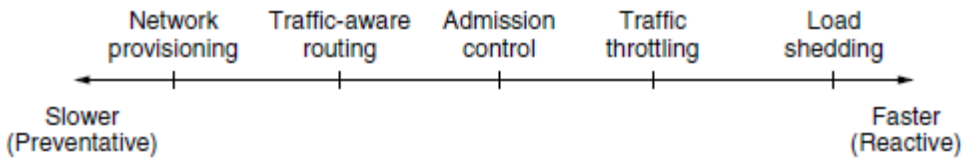
- Congestion can be brought on by several factors. If all of a sudden, streams of packets begin arriving on three or four input lines and all need the same output line, a queue will build up.

- If there is insufficient memory to hold all of them, packets will be lost.

- Slow processors can also cause congestion. If the routers' CPUs are slow at performing the bookkeeping tasks required of them (queuing buffers, updating tables, etc.), queues can build up, even though there is excess line capacity. Similarly, low-bandwidth lines can also cause congestion.

APPROACHES TO CONGESTION CONTROL

- Many problems in complex systems, such as computer networks, can be viewed from a control theory point of view. This approach leads to dividing all solutions into two groups: open loop and closed loop.



Timescales Of Approaches To Congestion Control

- Open loop solutions attempt to solve the problem by good design.
- Tools for doing open-loop control include deciding when to accept new traffic, deciding when to discard packets and which ones, and making scheduling decisions at various points in the network.
- Closed loop solutions are based on the concept of a feedback loop.
- This approach has three parts when applied to congestion control:

1. Monitor the system to detect when and where congestion occurs.
2. Pass this information to places where action can be taken.
3. Adjust system operation to correct the problem.

- A variety of metrics can be used to monitor the subnet for congestion. Chief among these are the percentage of all packets discarded for lack of buffer space, the average queue lengths, the number of packets that time out and are retransmitted, the average packet delay, and the standard deviation of packet delay. In all cases, rising numbers indicate growing congestion.
- The second step in the feedback loop is to transfer the information about the congestion from the point where it is detected to the point where something can be done about it.
- In all feedback schemes, the hope is that knowledge of congestion will cause the hosts to take appropriate action to reduce the congestion.
- The presence of congestion means that the load is (temporarily) greater than the resources (in part of the system) can handle. Two solutions come to mind: increase the resources or decrease the load.

CONGESTION PREVENTION POLICIES

The methods to control congestion by looking at open loop systems. These systems are designed to minimize congestion in the first place, rather than letting it happen and reacting after the fact. They try to achieve their goal by using appropriate policies at various levels. In Fig. 5-26 we see different data link, network, and transport policies that can affect congestion (Jain, 1990).

Figure 5-26. Policies that affect congestion.

Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

The **data link layer Policies.**

- The **retransmission policy** is concerned with how fast a sender times out and what it transmits upon timeout. A jumpy sender that times out quickly and retransmits all outstanding packets using go back n will put a heavier load on the system than will a leisurely sender that uses selective repeat.
- Closely related to this is the **buffering policy**. If receivers routinely discard all out-of-order packets, these packets will have to be transmitted again later, creating extra load. With respect to congestion control, selective repeat is clearly better than go back n.
- **Acknowledgement policy** also affects congestion. If each packet is acknowledged immediately, the acknowledgement packets generate extra traffic. However, if acknowledgements are saved up to piggyback onto reverse traffic, extra timeouts and retransmissions may result. A tight flow control scheme (e.g., a small window) reduces the data rate and thus helps fight congestion.

The **network layer Policies.**

- The choice between using **virtual circuits and using datagrams** affects congestion since many congestion control algorithms work only with virtual-circuit subnets.
- **Packet queueing and service policy** relates to whether routers have one queue per input line, one queue per output line, or both. It also relates to the order in which packets are processed (e.g., round robin or priority based).
- **Discard policy** is the rule telling which packet is dropped when there is no space.
- A good **routing algorithm** can help avoid congestion by spreading the traffic over all the lines, whereas a bad one can send too much traffic over already congested lines.

- **Packet lifetime management** deals with how long a packet may live before being discarded. If it is too long, lost packets may clog up the works for a long time, but if it is too short, packets may sometimes time out before reaching their destination, thus inducing retransmissions.

  The **transport layer Policies**,

- The **same issues occur as in the data link layer**, but in addition, determining the **timeout interval** is harder because the transit time across the network is less predictable than the transit time over a wire between two routers. If the timeout interval is too short, extra packets will be sent unnecessarily. If it is too long, congestion will be reduced but the response time will suffer whenever a packet is lost.

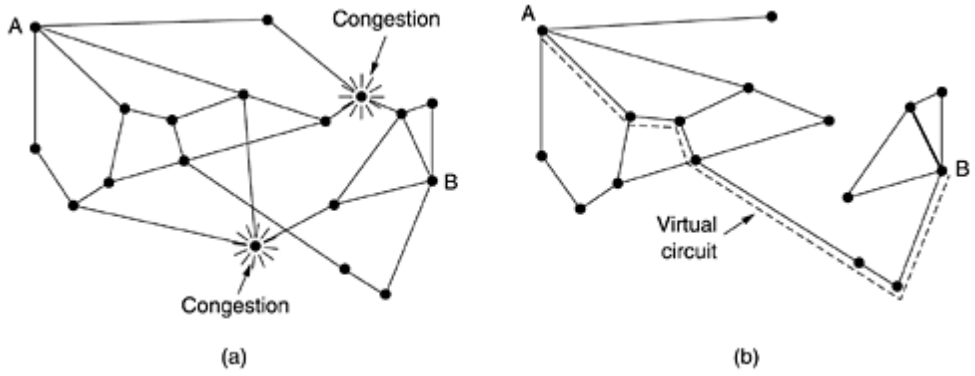
ADMISSION CONTROL

- One technique that is widely used to keep congestion that has already started from getting worse is **admission control**.

- Once congestion has been signaled, no more virtual circuits are set up until the problem has gone away.

- An alternative approach is to allow new virtual circuits but carefully route all new virtual circuits around problem areas. For example, consider the subnet of Fig. 5-27(a), in which two routers are congested, as indicated.

Figure 5-27. (a) A congested subnet. (b) A redrawn subnet that eliminates the congestion. A virtual circuit from* A *to* B *is also shown.



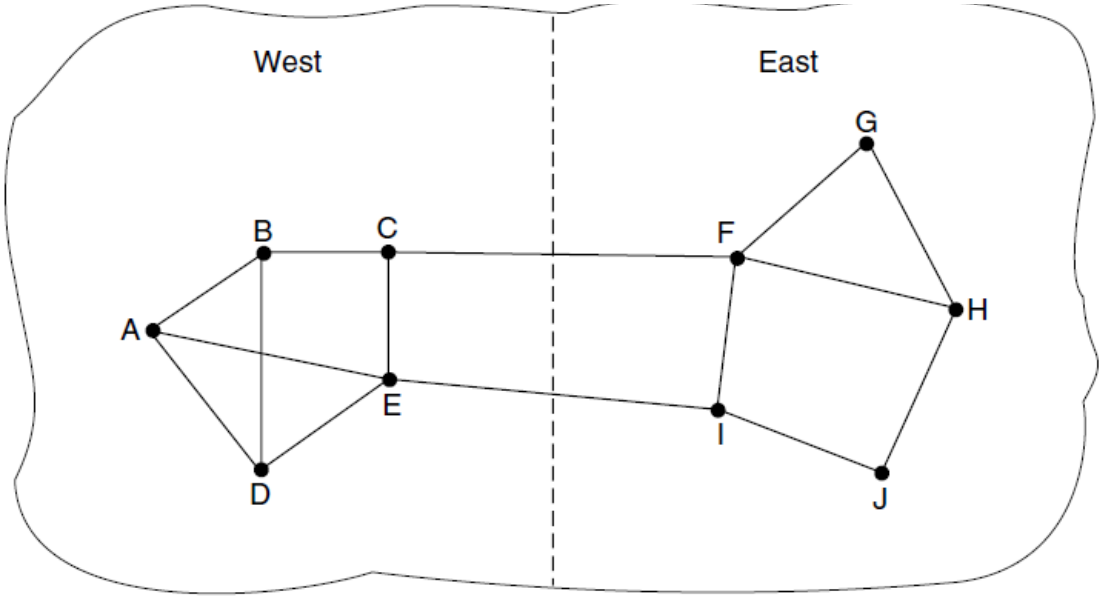
Suppose that a host attached to router *A* wants to set up a connection to a host attached to router *B*. Normally, this connection would pass through one of the congested routers. To avoid this situation, we can redraw the subnet as shown in Fig. 5-27(b), omitting the congested routers and all of their lines. The dashed line shows a possible route for the virtual circuit that avoids the congested routers.

TRAFFIC AWARE ROUTING

These schemes adapted to changes in topology, but not to changes in load. The goal in taking load into account when computing routes is to shift traffic away from hotspots that will be the first places in the network to experience congestion.

The most direct way to do this is to set the link weight to be a function of the (fixed) link bandwidth and propagation delay plus the (variable) measured load or average queuing delay. Least-weight paths will then favor paths that are more lightly loaded, all else being equal.

Consider the network of Fig. 5-23, which is divided into two parts, East and West, connected by two links, *CF* and *EI*. Suppose that most of the traffic between East and West is using link *CF*, and, as a result, this link is heavily loaded with long delays. Including queueing delay in the weight used for the shortest path calculation will make *EI* more attractive. After the new routing tables have been installed, most of the East-West traffic will now go over *EI*, loading this link. Consequently, in the next update, *CF* will appear to be the shortest path. As a result, the routing tables may oscillate wildly, leading to erratic routing and many potential problems.



If load is ignored and only bandwidth and propagation delay are considered, this problem does not occur. Attempts to include load but change weights within a narrow range only slow down routing oscillations. Two techniques can contribute to a successful solution. The first is multipath routing, in which there can be multiple paths from a source to a destination. In our example this means that the traffic can be spread across both of the East to West links. The second one is for the routing scheme to shift traffic across routes slowly enough that it is able to converge.

TRAFFIC THROTTLING

- Each router can easily monitor the utilization of its output lines and other resources. For example, it can associate with each line a real variable, *u*, whose value, between 0.0 and 1.0, reflects the

recent utilization of that line. To maintain a good estimate of *u*, a sample of the instantaneous line utilization, *f* (either 0 or 1), can be made periodically and *u* updated according to

$$u_{\text{new}} = au_{\text{old}} + (1 - a)f$$

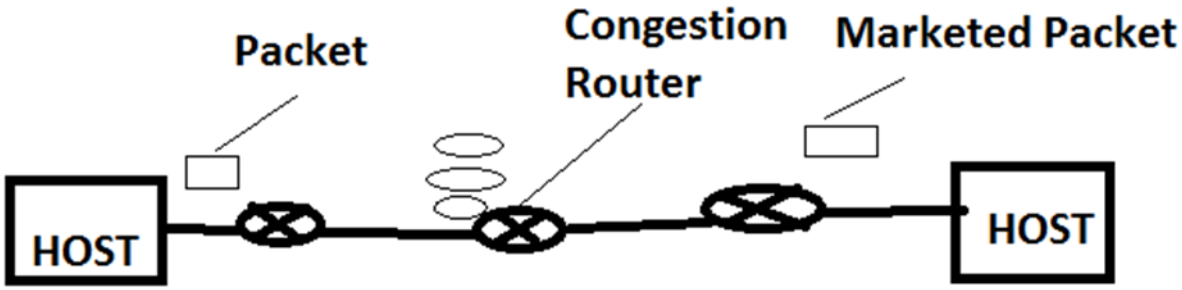
where the constant *a* determines how fast the router forgets recent history.

Whenever *u* moves above the threshold, the output line enters a "warning" state. Each newly-arriving packet is checked to see if its output line is in warning state. If it is, some action is taken. The action taken can be one of several alternatives, which we will now discuss.

CHOKE PACKETS

- In this approach, the router sends a **choke packet** back to the source host, giving it the destination found in the packet.
- The original packet is tagged (a header bit is turned on) so that it will not generate any more choke packets farther along the path and is then forwarded in the usual way.
- When the source host gets the choke packet, it is required to reduce the traffic sent to the specified destination by *X* percent. Since other packets aimed at the same destination are probably already under way and will generate yet more choke packets, the host should ignore choke packets referring to that destination for a fixed time interval. After that period has expired, the host listens for more choke packets for another interval. If one arrives, the line is still congested, so the host reduces the flow still more and begins ignoring choke packets again. If no choke packets arrive during the listening period, the host may increase the flow again.
- The feedback implicit in this protocol can help prevent congestion yet not throttle any flow unless trouble occurs.
- Hosts can reduce traffic by adjusting their policy parameters.
- Increases are done in smaller increments to prevent congestion from reoccurring quickly.
- Routers can maintain several thresholds. Depending on which threshold has been crossed, the choke packet can contain a mild warning, a stern warning, or an ultimatum.

Explicit Congestion Notification

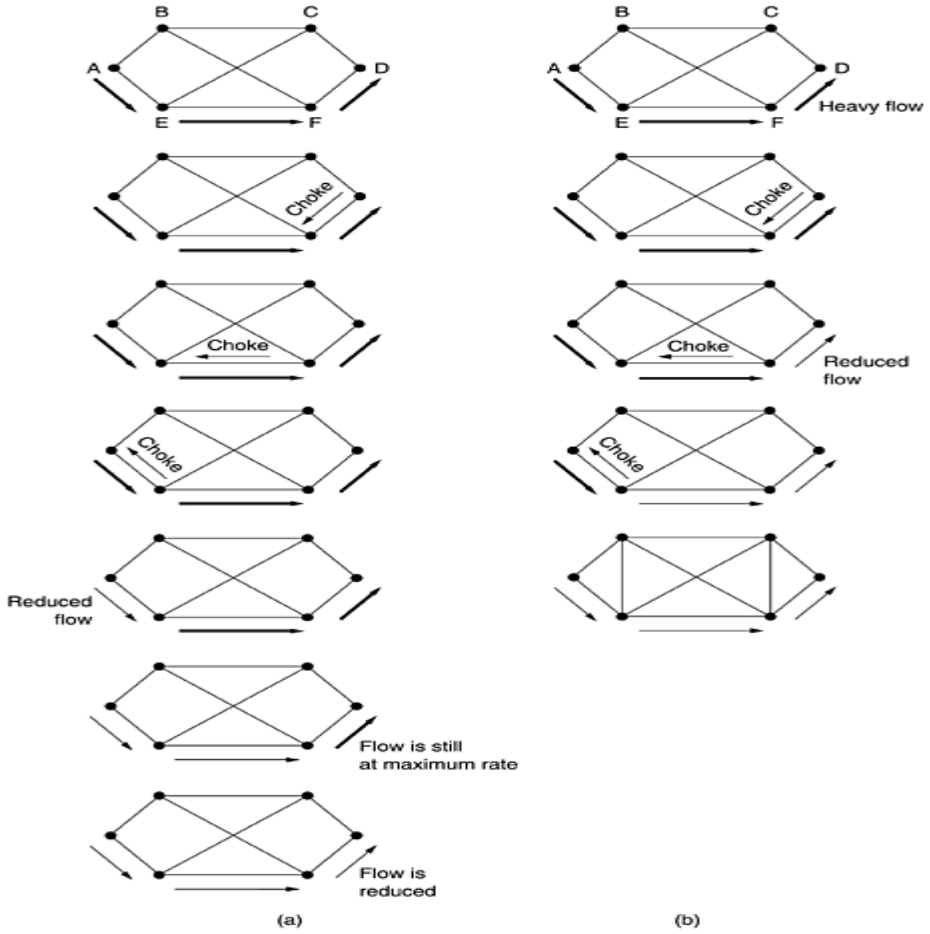


HOP-BY-HOP BACK PRESSURE

- At high speeds or over long distances, sending a choke packet to the source hosts does not work well because the reaction is so slow.

Consider, for example, a host in San Francisco (router *A* in Fig. 5-28) that is sending traffic to a host in New York (router *D* in Fig. 5-28) at 155 Mbps. If the New York host begins to run out of buffers, it will take about 30 msec for a choke packet to get back to San Francisco to tell it to slow down. The choke packet propagation is shown as the second, third, and fourth steps in Fig. 5-28(a). In those 30 msec, another 4.6 megabits will have been sent. Even if the host in San Francisco completely shuts down immediately, the 4.6 megabits in the pipe will continue to pour in and have to be dealt with. Only in the seventh diagram in Fig. 5-28(a) will the New York router notice a slower flow.

Figure 5-28. (a) A choke packet that affects only the source. (b) A choke packet that affects each hop it passes through.



An alternative approach is to have the choke packet take effect at every hop it passes through, as shown in the sequence of Fig. 5-28(b). Here, as soon as the choke packet reaches *F*, *F* is required to reduce the flow to *D*. Doing so will require *F* to devote more buffers to the flow, since the source is still sending away at full blast, but it gives *D* immediate relief, like a headache remedy in a television commercial. In the next step, the choke packet reaches *E*, which tells *E* to reduce the flow to *F*. This action puts a greater demand on *E*'s buffers but gives *F* immediate relief. Finally, the choke packet reaches *A* and the flow genuinely slows down.

The net effect of this hop-by-hop scheme is to provide quick relief at the point of congestion at the price of using up more buffers upstream. In this way, congestion can be nipped in the bud without losing any packets.

LOAD SHEDDING

- When none of the above methods make the congestion disappear, routers can bring out the heavy artillery: load shedding.
- **Load shedding** is a fancy way of saying that when routers are being in undated by packets that they cannot handle, they just throw them away.

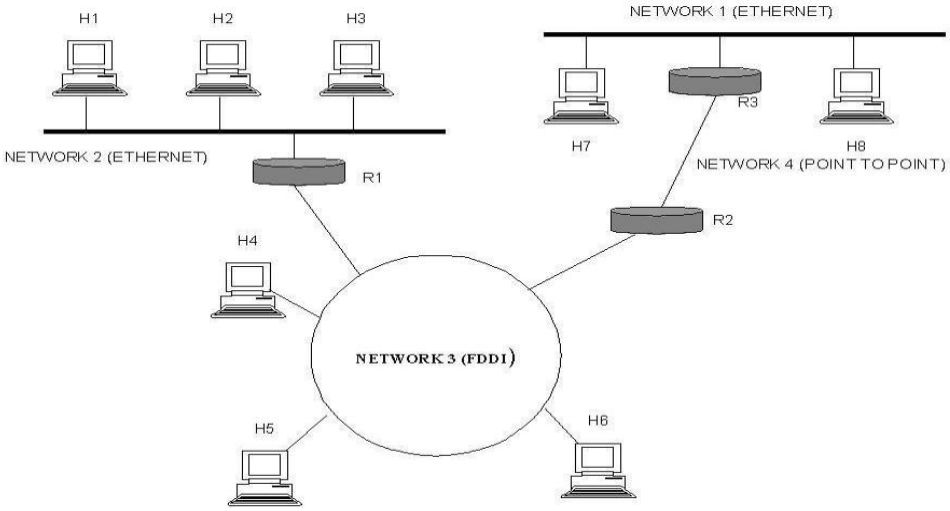
- A router drowning in packets can just pick packets at random to drop, but usually it can do better than that.
- Which packet to discard may depend on the applications running.
- To implement an intelligent discard policy, applications must mark their packets in priority classes to indicate how important they are. If they do this, then when packets have to be discarded, routers can first drop packets from the lowest class, then the next lowest class, and so on.

RANDOM EARLY DETECTION

- It is well known that dealing with congestion after it is first detected is more effective than letting it gum up the works and then trying to deal with it. This observation leads to the idea of discarding packets before all the buffer space is really exhausted. A popular algorithm for doing this is called **RED** (**Random Early Detection)**.
- In some transport protocols (including TCP), the response to lost packets is for the source to slow down. The reasoning behind this logic is that TCP was designed for wired networks and wired networks are very reliable, so lost packets are mostly due to buffer overruns rather than transmission errors. This fact can be exploited to help **reduce congestion**.
- By having routers drop packets before the situation has become hopeless (hence the "early" in the name), the idea is that there is time for action to be taken before it is too late. To determine when to start discarding, routers maintain a running average of their queue lengths. When the average queue length on some line exceeds a threshold, the line is said to be congested and action is taken.

4.Explain about internetworking

An internetwork is often referred to as a network of networks because it is made up of lots of smaller networks. The nodes that interconnect the networks are called routers. They are also sometimes called gateways, but since this term has several other connotations, we restrict our usage to router. The internet protocol is the key tool used today to build scalable, heterogeneous internetwork.



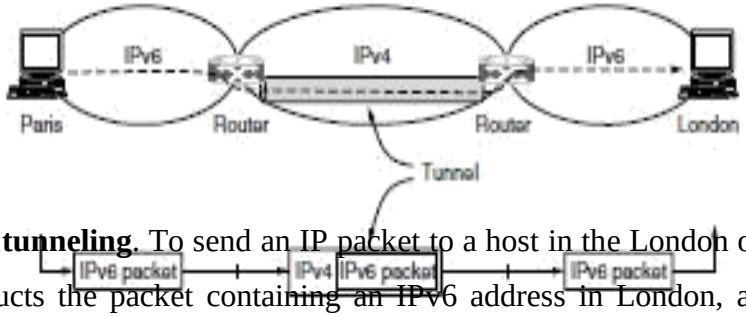
- Tunneling

- Internetwork Routing
- Packet Fragmentation

TUNNELING

- If they are two geographically separate networks, which want to communicate with each other, they may deploy a dedicated line between or they have to pass their data through intermediate networks.

- Tunneling is a mechanism by which two or more same networks communicate with each other, by passing intermediate networking complexities. Tunneling is configured at both ends

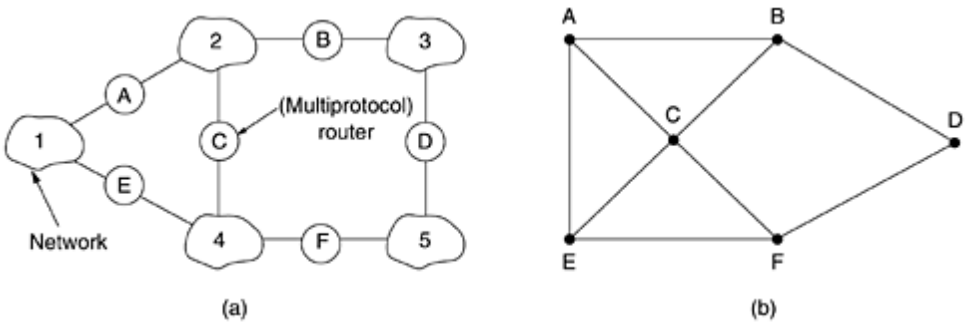


A technique called **tunneling**. To send an IP packet to a host in the London office, a host in the Paris office constructs the packet containing an IPv6 address in London, and sends it to the multiprotocol router that connects the Paris IPv6 network to the IPv4 Internet. When this router gets the IPv6 packet, it encapsulates the packet with an IPv4 header addressed to the IPv4 side of the multiprotocol router that connects to the London IPv6 network. That is, the router puts a (IPv6) packet inside a (IPv4) packet. When this wrapped packet arrives, the London router removes the original IPv6 packet and sends it onward to the destination host.

INTERNETWORK ROUTING

- Routing through an internetwork is similar to routing within a single subnet, but with some added complications.
- Consider, for example, the internetwork of Fig. 5-49(a) in which five networks are connected by six (possibly multiprotocol) routers. Making a graph model of this situation is complicated by the fact that every router can directly access (i.e., send packets to) every other router connected to any network to which it is connected. For example, *B* in Fig. 5-49(a) can directly access *A* and *C* via network 2 and also *D* via network 3. This leads to the graph of Fig. 5-49(b).

Figure 5-49. (a) An internetwork. (b) A graph of the internetwork.



- Once the graph has been constructed, known routing algorithms, such as the distance vector and link state algorithms, can be applied to the set of multiprotocol routers.

- This gives a two-level routing algorithm: within each network an **interior gateway protocol** is used, but between the networks, an **exterior gateway protocol** is used ("gateway" is an older term for "router").

- Network in an internetwork is independent of all the others, it is often referred to as an **Autonomous System** (**AS**).

- A typical internet packet starts out on its LAN addressed to the local multiprotocol router (in the MAC layer header). After it gets there, the network layer code decides which multiprotocol router to forward the packet to, using its own routing tables. If that router can be reached using the packet's native network protocol, the packet is forwarded there directly. Otherwise it is tunneled there, encapsulated in the protocol required by the intervening network. This process is repeated until the packet reaches the destination network.

- One of the differences between internetwork routing and intranet work routing is that internetwork routing may require crossing international boundaries. Various laws suddenly come into play, such as Sweden's strict privacy laws about exporting personal data about Swedish citizens from Sweden. Another example is the Canadian law saying that data traffic originating in Canada and ending in Canada may not leave the country. This law means that traffic from Windsor, Ontario to Vancouver may not be routed via nearby Detroit, even if that route is the fastest and cheapest.

  Another difference between interior and exterior routing is the cost. Within a single network, a single charging algorithm normally applies. However, different networks may be under different managements, and one route may be less expensive than another. Similarly, the quality of service offered by different networks may be different, and this may be a reason to choose one route over another.

FRAGMENTATION

Each network imposes some maximum size on its packets. These limits have various causes, among them:

1. Hardware (e.g., the size of an Ethernet frame).

2. Operating system (e.g., all buffers are 512 bytes).

3. Protocols (e.g., the number of bits in the packet length field).

4. Compliance with some (inter)national standard.

5. Desire to reduce error-induced retransmissions to some level.

6. Desire to prevent one packet from occupying the channel too long.

- From the above factors maximum payloads range from 48 bytes (ATM cells) to 65,515 bytes (IP packets), although the payload size in higher layers is often larger.
- If the original source packet is too large to be handled by the destination network? The routing algorithm can hardly bypass the destination.
- The only solution to the problem is to allow gateways to break up packets into **fragments**, sending each fragment as a separate internet packet. Packet-switching networks, too, have trouble putting the fragments back together again.

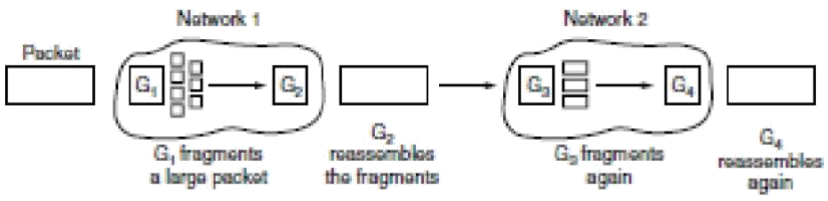
- Most Ethernet segments have their maximum transmission unit (MTU) fixed to 1500 bytes. A data packet can have more or less packet length depending upon the application. Devices in the transit path also have their hardware and software capabilities which tell what amount of data that device can handle and what size of packet it can process.

- If the data packet size is less than or equal to the size of packet the transit network can handle, it is processed neutrally. If the packet is larger, it is broken into smaller pieces and then forwarded. This is called packet fragmentation. Each fragment contains the same destination and source address and routed through transit path easily. At the receiving end it is assembled again.

- If a packet with DF (don't fragment) bit set to 1 comes to a router which can not handle the packet because of its length, the packet is dropped.

- When a packet is received by a router has its MF (more fragments) bit set to 1, the router then knows that it is a fragmented packet and parts of the original packet is on the way.

- If packet is fragmented too small, the overhead is increases. If the packet is fragmented too large, intermediate router may not be able to process it and it might get dropped. The alternative solution to the problem is to allow routers to break up packets into fragments, sending each fragment as a separate network layer packet



(a)

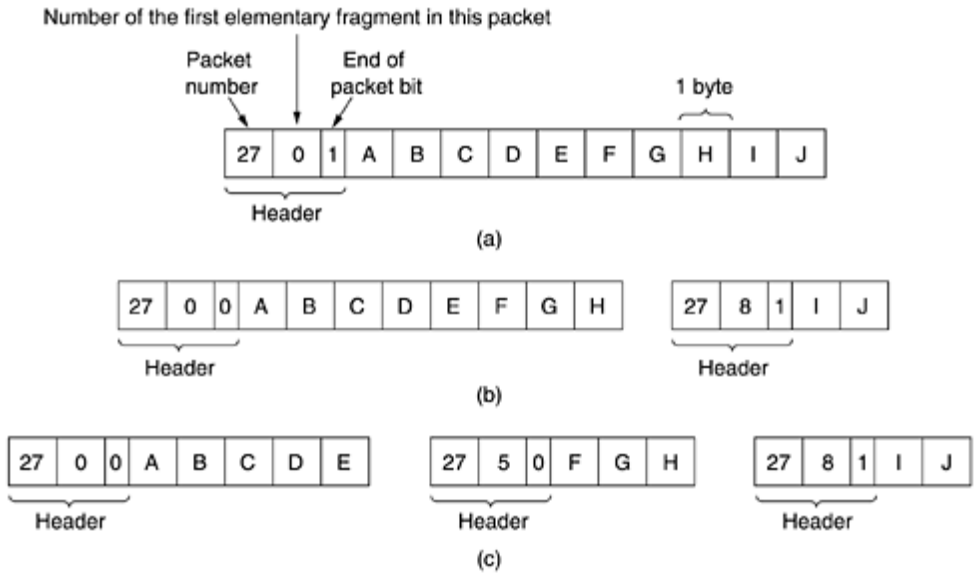
Transparent fragmentation is straightforward but has some problems. For one thing, the exit router must know when it has received all the pieces, so either a count field or an ‘‘end of packet’’ bit must be provided. Also, because all packets must exit via the same router so that they can be reassembled, the routes are constrained. By not allowing some fragments to follow one route to the ultimate destination and other fragments a disjoint route, some performance may be lost. More significant is the amount of work that the router may have to do. It may need to buffer the fragments as they arrive, and decide when to throw them away if not all of the fragments arrive. Some of this work may be wasteful, too, as the packet may pass through a series of small packet networks and need to be repeatedly fragmented and reassembled.

The **Non fragmentation** strategy is to refrain from recombining fragments at any intermediate routers. Once a packet has been fragmented, each fragment is treated as though it were an original packet. The routers pass the fragments, and reassembly is performed only at the destination host.The main advantage of nontransparent fragmentation is that it requires routers to do less work. IP works this way.

This approach requires two sequence fields in the internet header:

- The original packet number and the fragment number. There is clearly a trade-off between the size of the elementary fragment and the number of bits in the fragment number. Because the elementary fragment size is presumed to be acceptable to every network, subsequent fragmentation of an internet packet containing several fragments causes no problem. The ultimate limit here is to have the elementary fragment be a single bit or byte, with the fragment number then being the bit or byte offset within the original packet, as shown in Fig. 5-51.

Figure 5-51. Fragmentation when the elementary data size is 1 byte. (a) Original packet, containing 10 data bytes. (b) Fragments after passing through a network with maximum packet size of 8 payload bytes plus header. (c) Fragments after passing through a size 5 gateway.



Some internet protocols take this method even further and consider the entire transmission on a virtual circuit to be one giant packet, so that each fragment contains the absolute byte number of the first byte within the fragment.

5.Explain in detail about Internet Protocol Version 4 (IPv4)

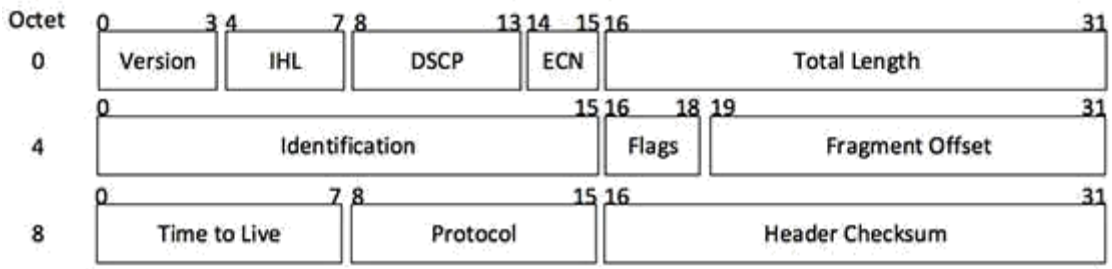
- Internet Protocol is one of the major protocols in the TCP/IP protocols suite. This protocol works at the network layer of the OSI model and at the Internet layer of the TCP/IP model. Thus this protocol has the responsibility of identifying hosts based upon their logical addresses and to route data among them over the underlying network

- IP provides a mechanism to uniquely identify hosts by an IP addressing scheme. IP uses best effort delivery, i.e. it does not guarantee that packets would be delivered to the destined host, but it will do its best to reach the destination. Internet Protocol version 4 uses 32-bit logical address.

- Internet Protocol being a layer-3 protocol (OSI) takes data Segments from layer-4 (Transport) and divides it into packets. IP packet encapsulates data unit received from above layer and add to its own header information.

(IP Encapsulation)

The encapsulated data is referred to as IP Payload. IP header contains all the necessary information to deliver the packet at the other end.



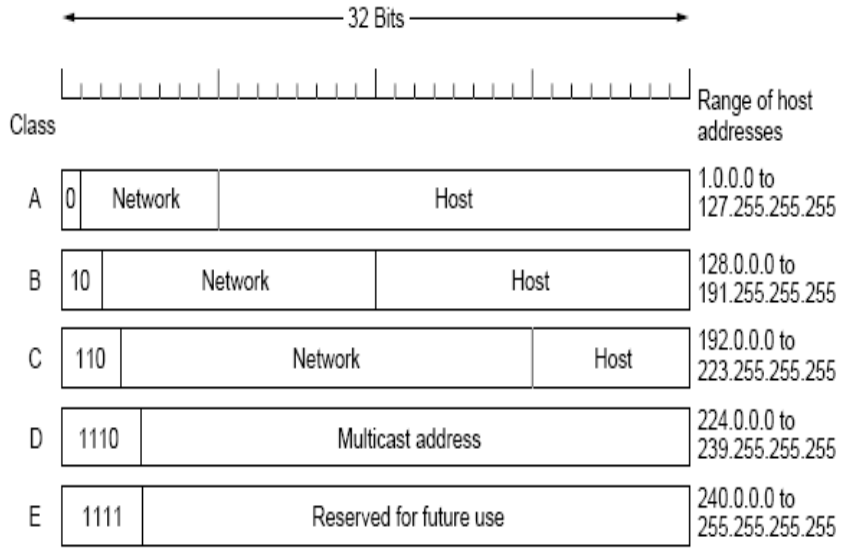
IP header includes many relevant information including Version Number, which, in this context, is 4. Other details are as follows:

- **Version:** Version no. of Internet Protocol used (e.g. IPv4).
- **IHL:** Internet Header Length; Length of entire IP header.
- **DSCP:** Differentiated Services Code Point; this is Type of Service.
- **ECN:** Explicit Congestion Notification; It carries information about the congestion seen in the route.
- **Total Length:** Length of entire IP Packet (including IP header and IP Payload).
- **Identification:** If IP packet is fragmented during the transmission, all the fragments contain same identification number. to identify original IP packet they belong to.
- **Flags:** As required by the network resources, if IP Packet is too large to handle, these 'flags' tells if they can be fragmented or not. In this 3-bit flag, the MSB is always set to '0'.
- **Fragment Offset:** This offset tells the exact position of the fragment in the original IP Packet.
- **Time to Live:** To avoid looping in the network, every packet is sent with some TTL value set, which tells the network how many routers (hops) this packet can cross. At each hop, its value is decremented by one and when the value reaches zero, the packet is discarded.
- **Protocol:** Tells the Network layer at the destination host, to which Protocol this packet belongs to, i.e. the next level Protocol. For example protocol number of ICMP is 1, TCP is 6 and UDP is 17.
- **Header Checksum:** This field is used to keep checksum value of entire header which is then used to check if the packet is received error-free.
- **Source Address:** 32-bit address of the Sender (or source) of the packet.
- **Destination Address:** 32-bit address of the Receiver (or destination) of the packet.
- **Options:** This is optional field, which is used if the value of IHL is greater than 5. These options may contain values for options such as Security, Record Route, Time Stamp, etc.

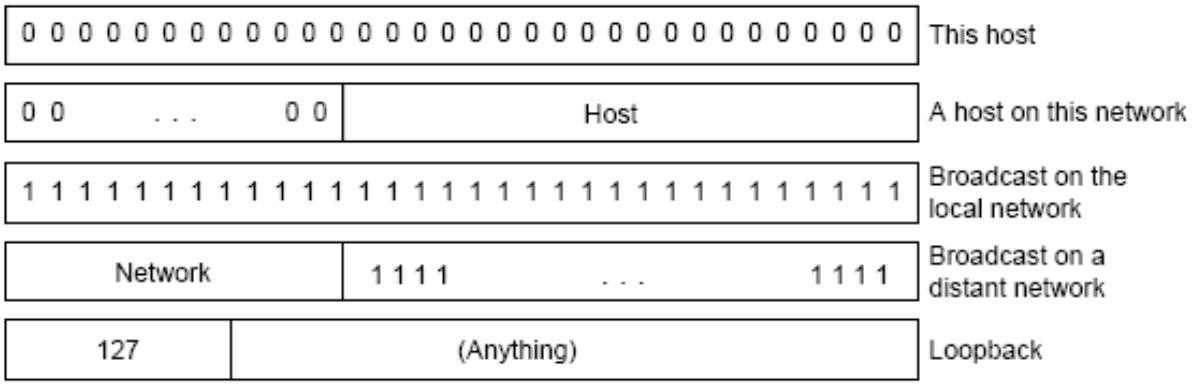
5. Explain IP Address

IP ADDRESS

- Every host and router on the Internet has an IP address, which encodes its network number and host number.
- All IP addresses are 32 bits long. It is important to note that an IP address does not actually refer to a host. It really refers to a network interface, so if a host is on two networks, it must have two IP addresses.
- IP addresses were divided into the five categories



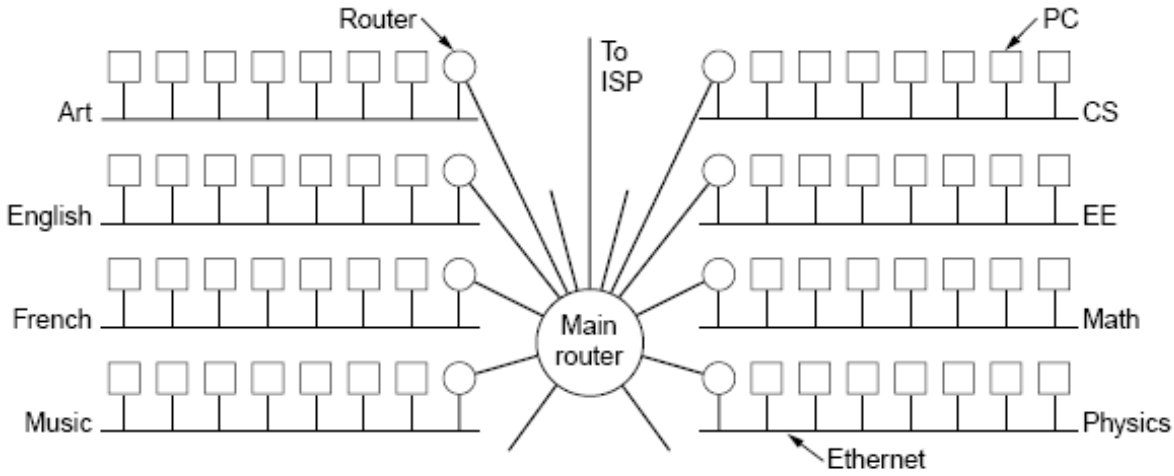
- The values 0 and -1 (all 1s) have special meanings. The value 0 means this network or this host. The value of -1 is used as a broadcast address to mean all hosts on the indicated network.



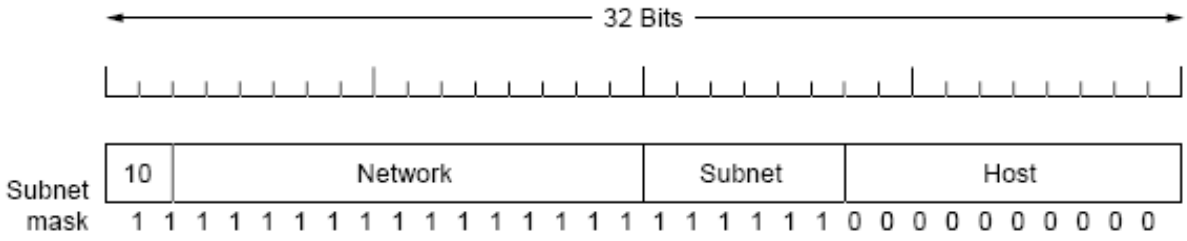
SUBNET

- All the hosts in a network must have the same network number. This property of IP addressing can cause problems as networks grow. For example……

- The problem is the rule that a single class A, B, or C address refers to one network, not to a collection of LANs.
- The solution is to allow a network to be split into several parts for internal use but still act like a single network to the outside world.



- To implement subnetting, the main router needs a subnet mask that indicates the split between network + subnet number and host.
- For example, if the university has a B address(130.50.0.0) and 35 departments, it could use a 6-bit subnet number and a 10-bit host number, allowing for up to 64 Ethernets, each with a maximum of 1022 hosts.
- The subnet mask can be written as 255.255.252.0. An alternative notation is /22 to indicate that the subnet mask is 22 bits long.



7. Explain in detail about Internet Protocol v6 (IPv6)

IETF (Internet Engineering Task Force) has redesigned IP addresses to mitigate the drawbacks of IPv4. The new IP address is version 6 which is 128-bit address, by which every single inch of the earth can be given millions of IP addresses.

Today majority of devices running on Internet are using IPv4 and it is not possible to shift them to IPv6 in the coming days. There are mechanisms provided by IPv6, by which IPv4 and IPv6 can co-exist unless the Internet entirely shifts to IPv6:

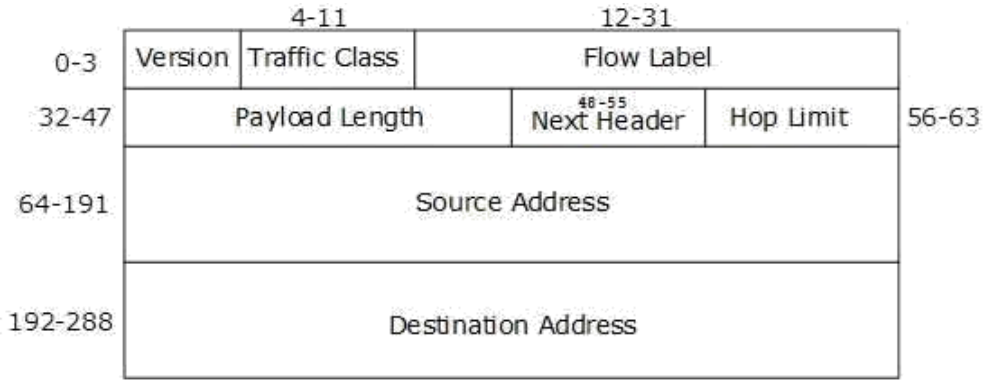
- Dual IP Stack

Tunneling (6to4 and 4to6)

- NAT Protocol Translation

- An IPv6 address is 4 times larger than IPv4, but surprisingly, the header of an IPv6 address is only 2 times larger than that of IPv4. IPv6 headers have one Fixed Header and zero or more Optional (Extension) Headers. All the necessary information that is essential for a router is kept in the Fixed Header. The Extension Header contains optional information that helps routers to understand how to handle a packet/flow.

- **Fixed Header**



- IPv6 fixed header is 40 bytes long and contains the following information.

Field & Description

- **Version** (4-bits): It represents the version of Internet Protocol, i.e. 0110

- **Traffic Class** (8-bits): These 8 bits are divided into two parts. The most significant 6 bits are used for Type of Service to let the Router Known what services should be provided to this packet. The least significant 2 bits are used for Explicit Congestion Notification (ECN).

- **Flow Label** (20-bits): This label is used to maintain the sequential flow of the packets belonging to a communication. The source labels the sequence to help the router identify that a particular packet belongs to a specific flow of information. This field helps avoid re-ordering of data packets.

It is designed for streaming/real-time media.

- **Payload Length** (16-bits): This field is used to tell the routers how much information a particular packet contains in its payload. Payload is composed of Extension Headers and Upper Layer data. With 16 bits, up to 65535 bytes can be indicated; but if the Extension Headers contain Hop-by-Hop Extension Header, then the payload may exceed 65535 bytes and this field is set to 0.

- **Next Header** (8-bits): This field is used to indicate either the type of Extension Header, or if the Extension Header is not present then it indicates the Upper

  Layer PDU. The values for the type of Upper Layer PDU are same as IPv4's.

  Hop Limit (8-bits): This field is used to stop packet to loop in the network infinitely. This is same as TTL in IPv4. The value of Hop Limit field is decremented by 1 as it passes a link (router/hop). When the field reaches 0 the packet is discarded.

- **Source Address** (128-bits): This field indicates the address of originator of the packet.

- **Destination Address** (128-bits): This field provides the address of intended recipient of the packet.

8. Explain Internet Control Protocol

i. Internet Control Message Protocol (ICMP)
ii. Address Resolution Protocol(ARP)
iii. Dynamic Host Configuration protocol(DHCP)

INTERNET CONTROL MESSAGE PROTOCOL(ICMP)

The operation of the Internet is monitored closely by the routers. When something unexpected occurs during packet processing at a router, the event is reported to the sender by the **ICMP** (**Internet Control Message Protocol**). ICMP is also used to test the Internet. About a dozen types of ICMP messages are defined. Each ICMP message type is carried encapsulated in an IP packet.

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo and echo reply	Check if a machine is alive
Timestamp request/reply	Same as Echo, but with timestamp
Router advertisement/solicitation	Find a nearby router

The Principle ICMP Message Type

- The DESTINATION UNREACHABLE message is used when the router cannot locate the destination or when a packet with the *DF* bit cannot be delivered because a ‘‘small-packet’’ network stands in the way.

- The TIME EXCEEDED message is sent when a packet is dropped because its *TtL (Time to live)* counter has reached zero. This event is a symptom that packets are looping, or that the counter values are being set too low.

  The PARAMETER PROBLEM message indicates that an illegal value has been detected in a header field. This problem indicates a bug in the sending host’s IP software or possibly in the software of a router transited.

  The SOURCE QUENCH message was long ago used to throttle hosts that were sending too many packets. When a host received this message, it was expected to slow down. It is rarely used anymore because when congestion occurs, these packets tend to add more fuel to the fire and it is unclear how to respond to them.

  The REDIRECT message is used when a router notices that a packet seems to be routed incorrectly. It is used by the router to tell the sending host to update to a better route.

  The ECHO and ECHO REPLY messages are sent by hosts to see if a given destination is reachable and currently alive. Upon receiving the ECHO message, the destination is expected to send back an ECHO REPLY message. These messages are used in the **ping** utility that checks if a host is up and on the Internet.

  The TIMESTAMP REQUEST and TIMESTAMP REPLY messages are similar, except that the arrival time of the message and the departure time of the reply are recorded in the reply. This facility can be used to measure network performance.

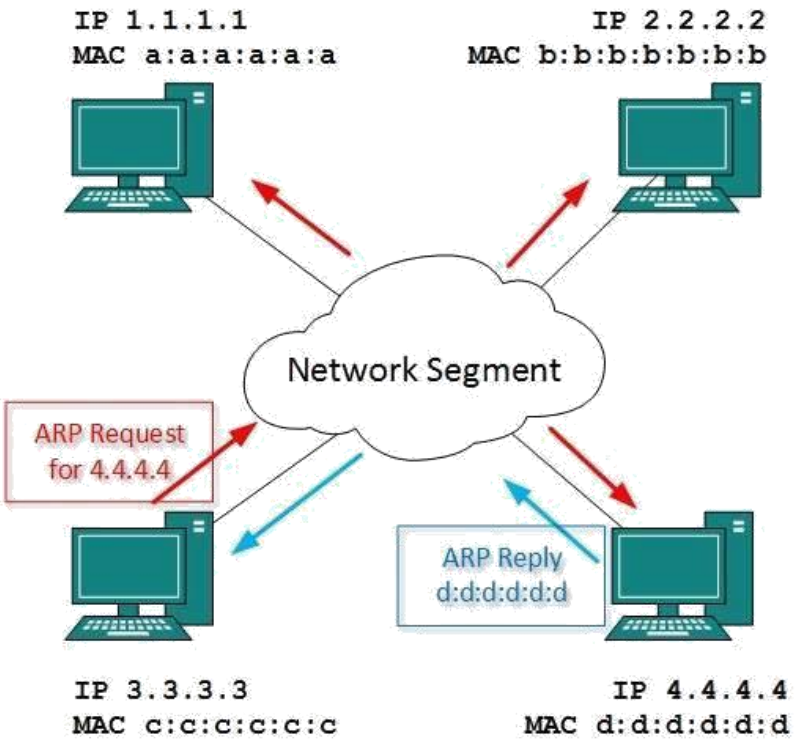
  The ROUTER ADVERTISEMENT and ROUTER SOLICITATION messages are used to let hosts find nearby routers. A host needs to learn the IP address of at least one router to be able to send packets off the local network.

ADDRESS RESOLUTION PROTOCOL(ARP)

- While communicating, a host needs Layer-2 (MAC) address of the destination machine which belongs to the same broadcast domain or network. A MAC address is physically burnt into the Network Interface Card (NIC) of a machine and it never changes.

- On the other hand, IP address on the public domain is rarely changed. If the NIC is changed in case of some fault, the MAC address also changes. This way, for Layer-2 communication to take place, a mapping between the two is required.

- To know the MAC address of remote host on a broadcast domain, a computer wishing to initiate communication sends out an ARP broadcast message asking,



"Who has this IP address?" Because it is a broadcast, all hosts on the network segment (broadcast domain) receive this packet and process it. ARP packet contains the IP address of destination host, the sending host wishes to talk to. When a host receives an ARP packet destined to it, it replies back with its own MAC address.

- Once the host gets destination MAC address, it can communicate with remote host using Layer-2 link protocol. This MAC to IP mapping is saved into ARP cache of both sending and receiving hosts. Next time, if they require to communicate, they can directly refer to their respective ARP cache.

- Reverse ARP is a mechanism where host knows the MAC address of remote host but requires to know IP address to communicate.

DYNAMIC HOST CONFIGURATION PROTOCOL(DHCP)

ARP make the assumption that the hosts are configured with some basic information, such as their own Ip address. It is possible to manually configure each computer but that is tedious and error – prone . This is called DHCP.

9. Explain about OSPF (Open Shortest Path First)- THE INTERIOR GATEWAY ROUTING PROTOCOL

OSPF Open Shortest Path First is a link-state routing protocol that was developed in 1991.OSPF was developed as a replacement for the distance vector routing protocol RIP.

It operates as a classless routing protocol that uses the concept of **areas** for network scalability. OSPF's major advantages over RIP are its fast convergence and its scalability to much larger network implementations. OSPF has a default administrative distance of 110. As a classless routing protocol, it does not use a Transport layer protocol, as OSPF packets are sent directly over IP.

- **Hello**: are used to establish and maintain adjacency with other OSPF routers. They are also used to elect the Designated Router (DR) and Backup Designated Router (BDR) on multi access networks (like Ethernet or Frame Relay).
    - **Database Description** (DBD or DD): contains an abbreviated list of the sending router's link-state database and is used by receiving routers to check against the local link-state database
- **Link-State Request** (LSR): used by receiving routers to request more information about any entry in the DBD
- **Link-State Update** (LSU): used to reply to LSRs as well as to announce new information. LSUs contain seven different types of Link-State Advertisements (LSAs)
- **Link-State Acknowledgement** (LSAck): sent to confirm receipt of an LSU message

OSPF Hello Packets

The OSPF Hello packet is used to establish neighbor adjacencies. By default, OSPF Hello packets are sent :

- Every 10 seconds on multi-access and point-to-point segments
- Every 30 seconds on non-broadcast multi-access (NBMA) segments (Frame Relay, X.25, ATM).

OSPF Dead Intervals

OSPF dead interval is measured as the period of time an OSPF router will wait before terminating adjacency with a neighbor. The Dead interval is four times the Hello interval, by default.

- For multi-access and point-to-point segments, this period is 40 seconds.
- For NBMA networks, the Dead interval is 120 seconds

For routers to become adjacent, their Hello interval, Dead interval, Network types and subnet masks must match.

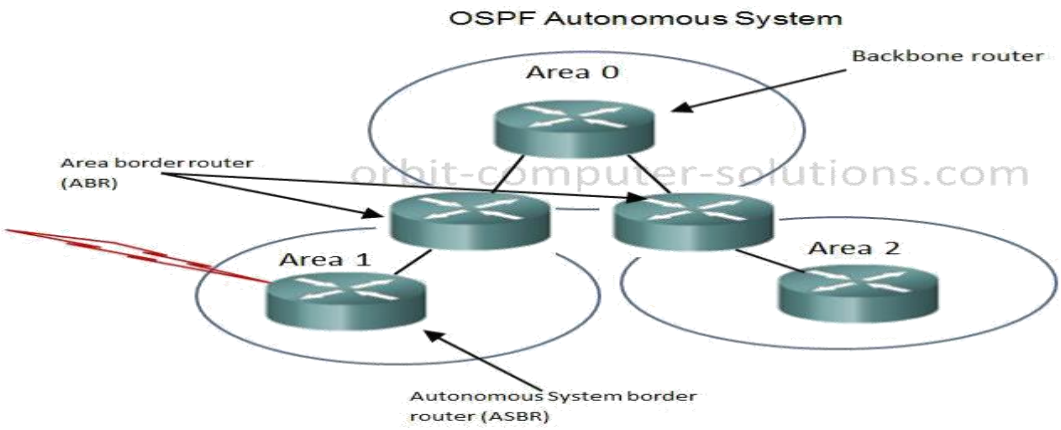
OSPF Router IDs

The OSPF router ID is used to exclusively identify each router in the OSPF routing domain. Cisco routers derive the router ID based on three criteria and with the following preference:

- the Use of the IP address configured with the OSPF **router ID** command.
- If the **router ID** is not configured, the router chooses highest IP address of any of its loopback interfaces.
- If no loopback interfaces are configured, the router chooses highest active IP address of any of its physical interfaces.

OSPF Area IDs

The **area** ID refers to the OSPF network domain area. An OSPF area is a group of routers that share link-state information. All OSPF routers in the same domain area must have the same link-state information - received from neighbours - in their link-state databases.



Open Shortest Path First (OSPF) is a popular routing protocol for IP networks for several key reasons:-

It is classless,

- Offers full CIDR and VLSM support,
- It scales well, converges quickly, and guarantees loop free routing.
- It also supports address summarization and the tagging of external routes, similar to EIGRP (Enhanced Interior Gateway Routing Protocol).

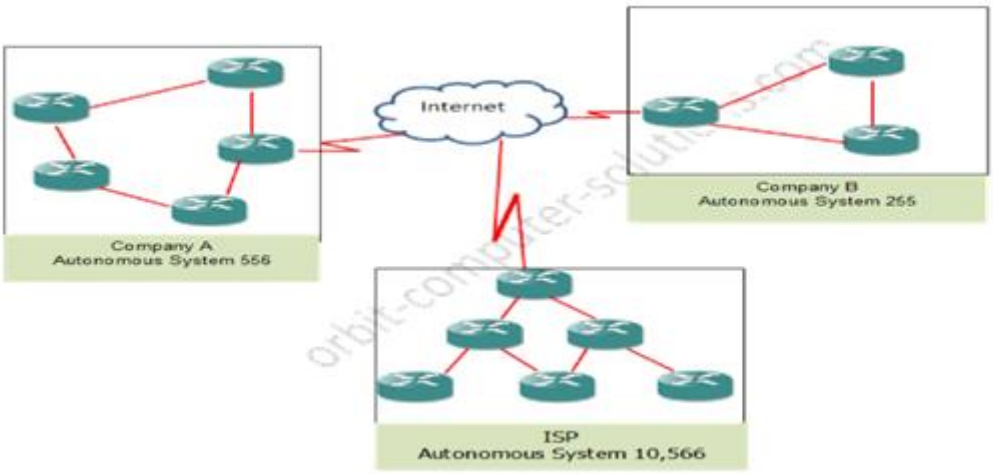
OSPF uses a large, dimensionless metric on every link - which can also be referred to as "cost - , with a maximum value of 65,535. In those protocols, each router updates the total metric as it passes the route on to the next router. However, in OSPF, the routers distribute the individual link costs to one another. The maximum cost for an individual link, then, is 65,535. Any given path through an OSPF network can include many high-

cost links, but still be usable. This is quite different from RIP, for example, where a few high-cost links along a path can make the entire path unusable.

10. Discuss about BGP (Border Gateway Protocol)- The EXTERIOR GATEWAY ROUTING PROTOCOL

- BGP is a complex, advanced distance Exterior Gateway Protocol (EGP), BGP exchange routing information between Autonomous Systems (ASs).
- Unlike Interior routing protocols such as RIP, EIGRP, and OSPF that run inside a company's network, BGP uses a different basic algorithm for building a loop-free topology than any of the above mentioned protocols.
- BGP is especially used for exchanging routing information between all of the major Internet Service Providers (ISPs), as well between larger client sites and their respective ISPs. And, in some large enterprise networks, BGP is used to interconnect different geographical or administrative regions.
- BGP is Primarily used to support the complexity of the public Internet,CISCO has added several clever and useful features to its BGP implementation (BGP 4). Some of the primary attributes of BGP is the use of pieces of information about a known route, where it came from, and how to reach it, A BGP router will also generate an error message if it receives a route that is missing these are mandatory attributes.

Clients/ Corporate Networks being connected by BGP



Types of BGP

There are different terms used when describing BGP. these including:

- 1.Internal BGP (iBGP) operates inside an autonomous System (AS)
- 2.External BGP (eBGP), which is also known as an interdomain routing protocol, operates outside an AS and connects one AS to another. These terms are just used to describe the same protocol just the area of operation is what differs.

Autonomous Systems (AS)

An autonomous system can be a company, ISP or an entire corporate network comprised of multiple locations connecting to the network. Each autonomous System (AS) uses BGP to advertise routes in its network that need to be visible outside of the network; it also uses BGP to learn about the reachability and routes by listening to advertisement announcements from other autonomous systems. Each of these enterprise network, commercial enterprise or ISP must be identified by an autonomous system number (ASN). This number allows a hierarchy to be maintained when sharing route information.

There are 65,535 (from 0to 65,535) available autonomous system numbers that can be assigned. BGP assigns 64,512 - 65,534 ASNs to be private. Being private means this ASN connect to only one other ASN (sometimes multiple ASN) and these ASNs can't cause loop by themselves.

UNIVERSITY QUESTIONS:

2 marks:

1.Define Routing Algorithm.(Dec 2014)
2.What is Virtual Circuit? (Dec 2014)
3.Define JCMP.(April 2015)
4.What is router?(April 2015)
5.Difference between IPv4 and IPv6 (Nov 2015)
6.Why admission control required in network? (Nov 2015)

11 marks:

1.Explain broadcast and multicast routing.(Dec2014)
2. Explain Internet protocol.(Dec 2014)
3.Describe IPv4 (Dec 2014)
4. Explain Optimality principal.(Nov2015)
5. Describe Distance Vector routing.(Nov2015)

UNIT-IV

Transport layer - Services - Berkeley Sockets -Example – Elements of Transport protocols – Addressing - Connection Establishment - Connection Release - Flow Control and Buffering – Multiplexing – Congestion Control - Bandwidth Allocation - Regulating the Sending Rate – UDP-RPC – TCP - TCP Segment Header - Connection Establishment - Connection Release - Transmission Policy - TCP Timer Management - TCP Congestion Control

2 MARKS

1. **What is transport layer?**

   The Transport layer is the fourth layer of the OSI reference model. In computer networking, a **transport layer** provides end-to-end or host-to-host communication services for applications within a layered architecture of network components and protocols. The **transport layer** provides services such as connection-oriented data stream support, reliability, flow control, and multiplexing.

2. **What is function of transport layer?**

   The protocol in the transport layer takes care in the delivery of data from one application program on one device to an application program on another device. They act as a link between the upper layer protocols and the services provided by the lower layer.

3. **What are the duties of the transport layer?**
   The services provided by the transport layer End-to- end delivery Addressing Reliable delivery Flow control Multiplexing
4. **What is the difference between network layer delivery and the transport layer delivery?**
   Network layer delivery: The network layer is responsible for the source-to-destination delivery of packet across multiple network links.
   Transport layer delivery: The transport layer is responsible form source-to-destination delivery of the entire message.
5. **What are the responsibilities of Transport Layer?**
   The Transport Layer is responsible for source-to-destination delivery of the entire message.
   - Service-point Addressing
   - Segmentation and reassembly
   - Connection Control
   - Flow Control
   - Error Control
6. **What are the four aspects related to the reliable delivery of data?**

   The four aspects are,
   - Error control
   - Sequence control
   - Loss control
   - Duplication contro
7. **What are the flow characteristics related to QOS?**
   The flow characteristics related to QOS are
   - Reliability
   - Delay

- Jitter
- Bandwidth

8. **What are the techniques to improve QOS?**
   The techniques to improve QOS are
   - Scheduling
   - Traffic shaping
   - Resource reservation
   - Admission control
9. **Write the relationship between transport and network layer?**
   Transport layer:
   - Logical communication between processes.
   - Responsible for checking that data available in session layer are error free.
   - Protocols used at this layer are :
     - TCP(Transmission Control Protocol)
     - UDP(User Datagram Protocol)
     - SCTP(Stream Control Transmission Protocol)

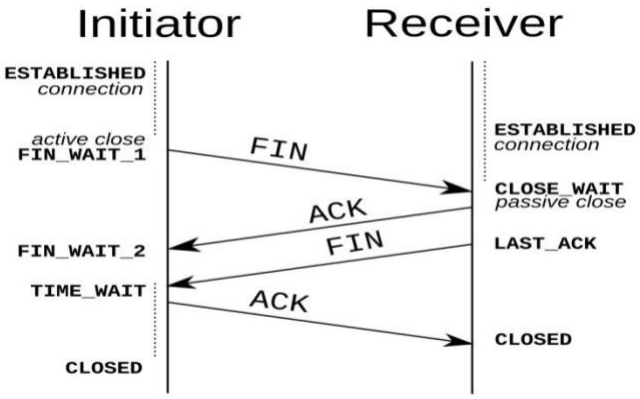
   Network layer:
   - Logical communication between hosts.
   - Responsible for logical addressing and translating logical addresses (ex. amazon.com) into physical addresses (ex. 180.215.206.136)
   - Protocols used at this layer are :
     - IP(Internet Protocol)
     - ICMP(Internet Control Message Protocol)
     - IGMP(Internet Group Message Protocol)
     - RARP(Reverse Address Resolution Protocol)
     - ARP(Address Resolution Protocol)

10. **Define Berkeley sockets?**
    Berkeley sockets (or BSD **sockets**) is a computing library with an application programming interface (API) for internet **sockets** and Unix domain **sockets**, used for inter-process communication (IPC).The API has evolved with little modification from a de facto standard into part of the POSIX specification. **POSIX sockets** are basically Berkeley sockets.

11. **What are the Elements Of Transport protocols?**
    - Addressing
    - Connection Establishment
    - Connection Release
    - Flow Control and Buffering
    - Multiplexing

12. **What is Connection Establishment?**
    To **establish** a **connection**, the three-way (or 3-step) handshake occurs: SYN: The active open is performed by the client sending a SYN to the server. The client sets the segment's

sequence number to a random value A. SYN-ACK: In response, the server replies with a SYN-ACK.



13. **What is Connection Release?**

Dropping connections isn't trivial either, as it turns out. An asymmetric release means that either host can destroy the connection (like hanging up the telephone). But data can be lost since there is no coordination between parties, so data that is "in the pipe" is lost once the connection is destroyed. Symmetric release requires both parties to agree to a release. If both parties know they are done sending data, and agree, then no data is lost.

14. **What is two army problems?**

- White army in valley. Blue arm in hills on either side of valley. White army can defeat either blue army in isolation, but blue armies together can defeat white army. How do the blue armies coordinate an attack on the white army? Their only communication is via messaging through the valley where messengers may be lost (i.e. an unreliable channel).
- Blue army #1 sends message: attack at time X. Blue army #2 receives this message and sends an acknowledgment to it. Does the attack happen at time X? No, since blue army #2 can't know that it's ack was received. Adding an ack to the ack (three-way handshake) doesn't help, since now blue army #1 doesn't know if his ack to the ack got through, and if it didn't blue army #2 won't attack, so blue army #1 shouldn't attack either.

15. What is Transmission delay?

- In a network based on packet switching, **transmission delay** (or **store-and-forward delay**, also known as **packetization delay**) is the amount of time required to push all of the packet's bits into the wire. In other words, this is the delay caused by the data-rate of the link.
- Transmission delay is a function of the packet's length and has nothing to do with the distance between the two nodes. This delay is proportional to the packet's length in bits. It is given by the following formula:

$$D_T = N/R \text{seconds}$$

where

D_T is the transmission delay in seconds

N is the number of bits, and
R is the rate of transmission (say in bits per second)

16. **What is round-trip time?**

In telecommunications, the **round-trip** delay **time** (RTD) or **round-trip time** (**RTT**) is the length of **time** it takes for a signal to be sent plus the length of **time** it takes for an acknowledgment of that signal to be received.

17. **Define Flow control?**

- **Flow control** is the management of data **flow** between computers or devices or between nodes in a network so that the data can be handled at an efficient pace. Too much data arriving before a device can handle it causes data overflow, meaning the data is either lost or must be retransmitted
- Finally, some **transport layer** protocols, for example TCP, but not UDP, provide end-to-end reliable communication, i.e. error recovery by means of error detecting code and automatic repeat request (ARQ) protocol. The ARQ protocol also provides **flow control**, which may be combined with congestion avoidance.

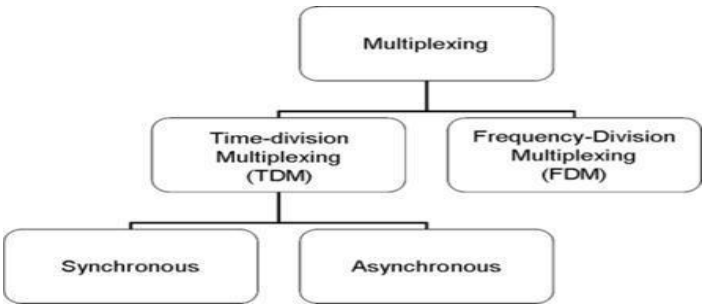
18. **What is buffering?**

- The sender's transport layer must worry about overwhelming both the network and the receiver. The network may exceed the carrying capacity, and the receiver may run out of buffers. Buffers are statically allocated kernel memory so that storing received TPDUs can be done quickly.
- Buffering isn't the only thing that limits the flow control in the transport layer. Suppose the receiver had an infinite supply of memory to dedicate to buffers. You still have the limit of the subnet's carrying capacity. This is the issue of congestion control.

19. **What is Multiplexing?**

In telecommunications and **computer networks**, **multiplexing** (sometimes contracted to muxing) is a method by which multiple analog message signals or digital data streams are combined into one signal over a shared medium. The aim is to share an expensive resource.

20. What are the different types of Multiplexing?



There are two basic forms of multiplexing used:

- Time division multiplexing (TDM)
- Frequency division multiplexing (FDM)

21. **What is meant by de-multiplexing?**

To separate two or more channels previously multiplexed. Demultiplexing is the reverse of multiplexing. Demultiplex (DEMUX) is the reverse of the multiplex (MUX) process – combining multiple unrelated analog or digital signal streams into one signal over a single shared medium, such as a single conductor of copper wire or fiber optic cable. Thus, demultiplex is reconverting a signal containing multiple analog or digital signal streams back into the original separate and unrelated signals.

22. **Define congestion Control?**

Congestion Control. When one part of the subnet (e.g. one or more routers in an area) becomes overloaded, **congestion** results. Because routers are receiving packets faster than they can forward them, one of two things must happen.

23. **What is Bandwidth allocation?**

Bandwidth allocation is the process of assigning radio frequencies to different applications. The radio spectrum is a finite resource creating the need for an effective **allocation** process.

24. **What is Dynamic bandwidth allocation?**

- **Dynamic bandwidth allocation** is a technique by which traffic bandwidth in a shared telecommunications medium can be allocated on demand and fairly between different users of that bandwidth. Where the sharing of a link adapts in some way to the instantaneous traffic demands of the nodes connected to the link.
- **D**ynamic bandwidth allocation takes advantage of several attributes of shared networks: (1) all users are typically not connected to the network at one time (2) even when connected, users are not transmitting data (or voice or video) at all times (3) most traffic occurs in bursts -- there are gaps between packets of information that can be filled with other user traffic.

25. Define UDP?

UDP (**User Datagram Protocol**) is a communications protocol that offers a limited amount of service when messages are exchanged between computers in a network that uses the Internet Protocol (IP). **UDP** is an alternative to the Transmission Control Protocol (TCP) and, together with IP, is sometimes referred to as **UDP/IP.**

26. What is the drawback of UDP?

- There are no guarantees with udp. a packet may not be delivered, or delivered twice, or delivered out of order; you get no indication of this unless the listening program at the other end decides to say something. tcp is really working in the same environment; you get roughly the same services from ip and udp. however, tcp makes up for it fairly well, and in a standardized manner.
- UDP has no flow control. implementation is the duty of user programs.

- Routers are quite careless with udp. they never retransmit it if it collides, and it seems to be the first thing dropped when a router is short on memory. udp suffers from worse packet loss than tcp.

27. **What is Datagram?**

A **datagram** is a basic transfer unit associated with a packet-switched network. The delivery, arrival time, and order of arrival need not be guaranteed by the network.

28. **Define Remote Procedure call?**

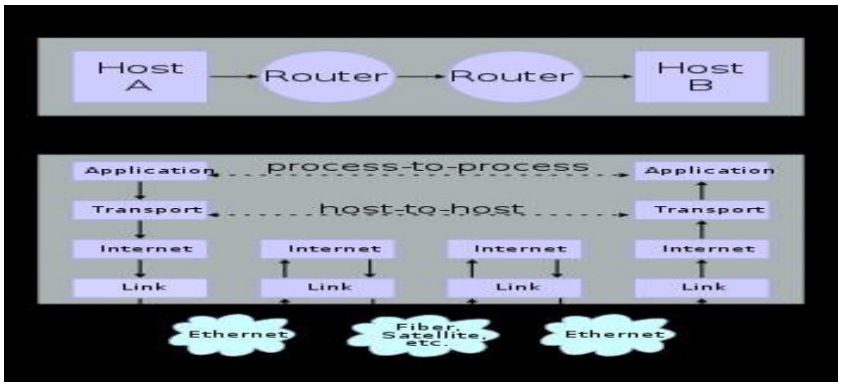
In computer science, a **remote procedure call** (**RPC**) is an inter-process communication that allows a computer program to cause a subroutine or procedure to execute in another address space (commonly on another computer on a shared **network**) without the programmer explicitly coding the details for this remote interaction.

29. **Define TCP?**

- The Transmission Control Protocol (TCP) is a core protocol of the Internet Protocol Suite. It originated in the initial network implementation in which it complemented the Internet Protocol (IP). Therefore, the entire suite is commonly referred to as TCP/IP. TCP provides reliable, ordered, and error-checked delivery of a stream of octets between applications running on hosts communicating over an IP network. TCP is the protocol that major Internet applications such as the World Wide Web, email, remote administration and file transfer rely on.
- If an acknowledgement for a segment is not received within the timeout, it is re-transmitted. TCP uses a congestion window in the sender side to do congestion avoidance. The congestion window indicates the maximum amount of data that can be sent out on a connection without being acknowledged.

30. **Define TCP/IP?**

It is commonly known as **TCP/IP**, because its most important protocols, the Transmission Control Protocol (**TCP**) and the Internet Protocol (**IP**), were the first networking protocols defined in this standard.



31. List out the services provided by TCP?

TCP/IP services are divided into two groups: services provided to other protocols and services provided to end users directly.

Services Provided to Other Protocols

These services are designed to actually accomplish the internetworking functions of the protocol suite. For example, at the network layer, IP provides functions such as addressing, delivery, and datagram packaging, fragmentation and reassembly. At the transport layer, TCP and UDP are concerned with encapsulating user data and managing connections between devices.

End-User Services

WWW services are provided through the Hypertext Transfer Protocol (HTTP), a TCP/IP application layer protocol. HTTP in turn uses services provided by lower-level protocols. All of these details are of course hidden from the end users, which is entirely on purpose.

32. **Discuss the TCP connections needed in FTP?**

- TCP connections needed in FTP establish two connections between the hosts.
- One connection is used for data transfer, the other for control information. The control connection uses very simple rules of communication.
- The data connection needs more complex rules due to the variety of data types transferred.

33. What is the use of option field in TCP?

Options: Provides a way to add extra facilities not covered by the regular header. eg,

- Maximum TCP payload that sender is willing to handle. The maximum size of segment is called MSS (Maximum Segment Size). At the time of handshake, both parties inform each other about their capacity. Minimum of the two is honoured.

  This information is sent in the options of the SYN packets of the three way handshake.
- Window scale option can be used to increase the window size. It can be specified by telling the receiver that the window size should be interpreted by shifting it left by specified number of bits. This header option allows window size up to 230.

34. What do you mean by receive window?

- In computer networking, **RWIN** (TCP Receive Window) is the amount of data that a computer can accept without acknowledging the sender. If the sender has not received acknowledgement for the first packet it sent, it will stop and wait and if this wait exceeds a certain limit, it may even retransmit.
- Even if there is no packet loss in the network, windowing can limit throughput. Because TCP transmits data up to the window size before waiting for the acknowledgements, the full bandwidth of the network may not always get used. The limitation caused by window size can be calculated as follows:
- Where RWIN is the TCP Receive Window and RTT is the round-trip time for the path.

35. Why an Application developer would ever choose to build an application over UDP rather than over TCP?

UDP is a transport protocol that does not check for errors. One would use it when speed of the transport is desired, not quality or reliability. If you were streaming video or using voice, then getting the packets to the destination quickly is more important than making sure each sound gets to the receiver.

36. The transport layer creates the connection between source and destination. What are the three events involved in the connection?

For security, the transport layer may create a connection between the two end ports. A connection is a single logical path between the source and destination that is associated with all packets in a message. Creating a connection involves three steps:

- Connection establishment
- Data transfer & Connection release.

37. What is the drawback of UDP?

- There are no guarantees with UDP. It is possible that a packet may not be delivered, or delivered twice, or delivered not in time.
- you have to manually break the data into packets

38. What is the use of option field in TCP?

Options may occupy space at the end of the TCP header and are a multiple of 8 bits in length. All options are included in the checksum. An option may begin on any octet boundary The option-length counts the two octets of option-kind and option-length as well as the option-data octets.

39. What is round-trip time?

Round-trip time (RTT), also called round-trip delay, is the time required for a signal pulse or packet to travel from a specific source to a specific destination and back again. In this context, the source is the computer initiating the signal and the destination is a remote computer or system that receives the signal and retransmits it.

11 MARKS

The transport layer is not just another layer. It is the heart of the whole protocol hierarchy. Its task is to provide reliable, cost-effective data transport from the source machine to the destination machine, independently of the physical network or networks currently in use. Without the transport layer, the whole concept of layered protocols would make little sense. We will study the transport layer in detail, including its services, design, protocols, and performance.

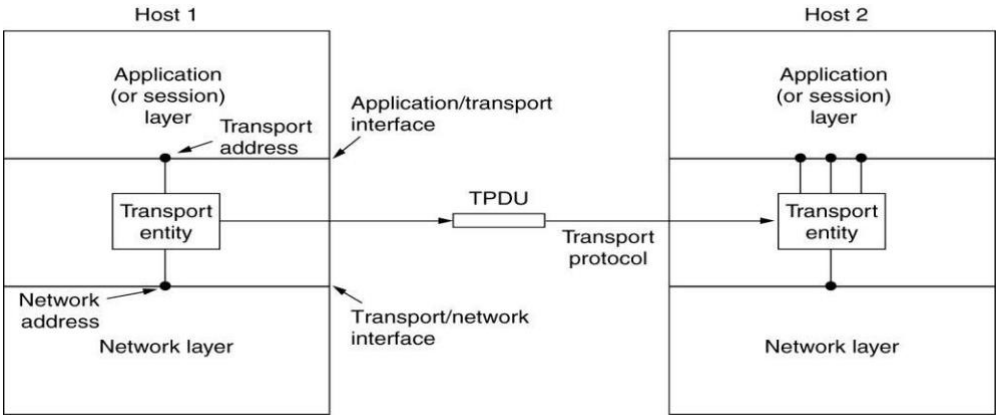
1. **Briefly explain transport layer services?**

- Services Provided to the Upper Layers
- Transport Service Primitives
  - TPDU
- Berkeley Sockets
- An Example of Socket Programming:
  - An Internet File Server

Services Provided to the Upper Layers

The ultimate goal of the transport layer is to provide efficient, reliable, and cost-effective service to its users, normally processes in the application layer. To achieve this goal, the transport layer makes use of the services provided by the network layer. The hardware and/or software within

the transport layer that does the work is called the transport entity. The (logical) relationship of the network, transport, and application layers is illustrated in Fig.



The network, transport, and application layers.

There are two types of network service, connection-oriented and connectionless; there are also two types of transport service. The connection-oriented transport service is similar to the connection-oriented network service in many ways. In both cases, connections have three phases: establishment, data transfer, and release. Addressing and flow control are also similar in both layers. Furthermore, the connectionless transport service is also very similar to the connectionless network service. It can be difficult to provide a connectionless transport service on top of a connection-oriented network service.

The transport code runs entirely on the users' machines, but the network layer mostly runs on the routers, which are operated by the carrier (at least for a wide area network). What happens if the network layer offers inadequate service? Suppose that it **frequently loses packets**? What happens if routers **crash from time to time**?

Problems occur, that's what. **The users have no real control over the network layer,** so they cannot solve the problem of poor service by using better routers or putting more error handling in the data link layer. The only possibility is to put on top of the network layer another layer that improves the quality of the service. In essence, **the existence of the transport layer makes it possible for the transport service to be more reliable than the underlying network service**. Lost packets and mangled data can be detected and compensated for by the transport layer. Furthermore, the transport service primitives can be implemented as calls to library procedures in order to make them independent of the network service primitives.

In transport layer, application programmers can write code according to a standard set of **primitives** and have these programs work on a wide variety of networks, without having to worry about dealing with different subnet interfaces and unreliable transmission. The bottom four layers can be seen as the **transport service provider**, whereas the upper layer(s) are the **transport service user**. This distinction of provider versus user has a considerable impact on the design of the layers and it forms the major boundary between the provider and user of the reliable data transmission service.

Transport Service Primitives

The transport service is similar to the network service, but there are also some important differences. The main difference is that the network service is intended to model the service offered by real networks, warts and all. Real networks can lose packets, so the network service is generally unreliable. The connection-oriented transport service, in contrast, is reliable. Of course, real networks are not error-free, but that is precisely the purpose of the transport layer—to provide a reliable service on top of an unreliable network.

A second difference between the network service and transport service is whom the services are intended for. The network service is used only by the transport entities. Many programs (and thus programmers) see the transport primitives. Consequently, the transport service must be convenient and easy to use.

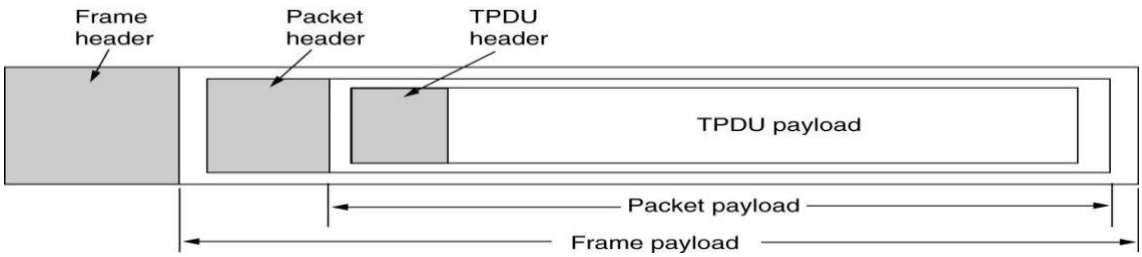
The server executes a LISTEN primitive, typically by calling a library procedure that makes a system call that blocks the server until a client turns up. When a client wants to talk to the server, it executes a CONNECT primitive. The transport entity carries out this primitive by blocking the caller and sending a packet to the server. Encapsulated in the payload of this packet is a transport layer message for the server's transport entity.

Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

The primitives for a simple transport service

TPDU (Transport Protocol Data Unit)

TPDU (Transport Protocol Data Unit) is a term used for messages sent from transport entity to transport entity. Thus, TPDUs (exchanged by the transport layer) are contained in packets (exchanged by the network layer). In turn, packets are contained in frames (exchanged by the data link layer). When a frame arrives, the data link layer processes the frame header and passes the contents of the frame payload field up to the network entity. The network entity processes the packet header and passes the contents of the packet payload up to the transport entity. This nesting is illustrated in Fig.

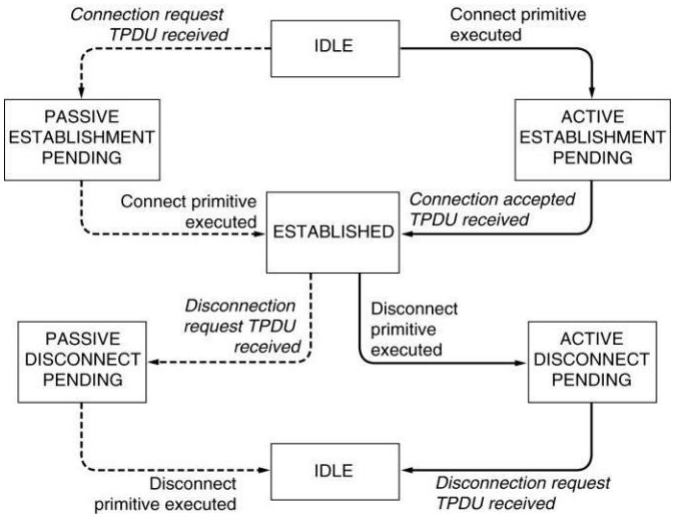


The nesting of TPDUs, packets, and frames.

Getting back to our client-server example, the client's CONNECT call causes a CONNECTION REQUEST segment to be sent to the server. When it arrives, the transport entity checks to see that the server is blocked on a LISTEN (i.e., is interested in handling requests). If so, it then unblocks the server and sends a CONNECTION ACCEPTED segment back to the client. When this segment arrives, the client is unblocked and the connection is established. Data can now be exchanged using the SEND and RECEIVE primitives. In the simplest form, either party can do a (blocking) RECEIVE to wait for the other party to do a SEND. When the segment arrives, the receiver is unblocked. It can then process the segment and send a reply. As long as both sides can keep track of whose turn it is to send, this scheme works fine.

Disconnection has two variants: asymmetric and symmetric. In the asymmetric variant, either transport user can issue a DISCONNECT primitive, which results in a DISCONNECT segment being sent to the remote transport entity. Upon its arrival, the connection is released. In the symmetric variant, each direction is closed separately, independently of the other one. When one side does a DISCONNECT, that means it has no more data to send but it is still willing to accept data from its partner. In this model, a connection is released when both sides have done a DISCONNECT.

State Diagrams



A state diagram for a simple connection management scheme. Transitions labeled in italics are caused by packet arrivals. The solid lines show the client's state sequence. The dashed lines show the server's state sequence.

Berkeley Sockets.

This methods provided by the Berkeley sockets API library:

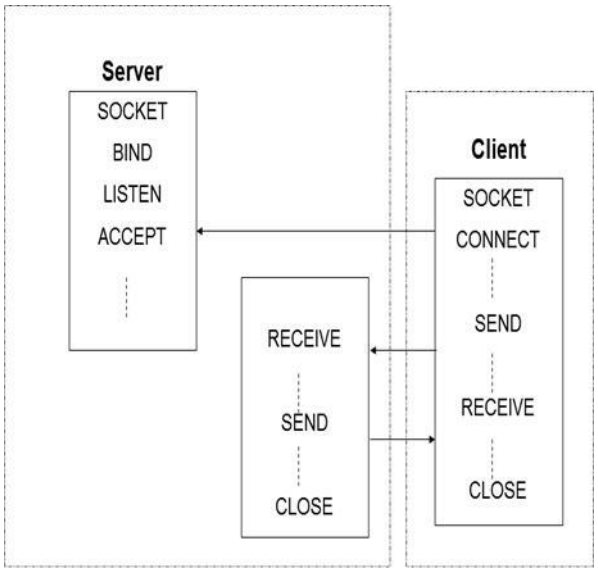
- socket() creates a new socket of a certain socket type, identified by an integer number, and allocates system resources to it.
- bind() is typically used on the server side, and associates a socket with a socket address structure, i.e. a specified local port number and IP address.
- listen() is used on the server side, and causes a bound TCP socket to enter listening state.

- connect() is used on the client side, and assigns a free local port number to a socket. In case of a TCP socket, it causes an attempt to establish a new TCP connection.
- accept() is used on the server side. It accepts a received incoming attempt to create a new TCP connection from the remote client, and creates a new socket associated with the socket address pair of this connection.
- send() and recv(), or write() and read(), or sendto() and recvfrom(), are used for sending and receiving data to/from a remote socket.
- close() causes the system to release resources allocated to a socket. In case of TCP, the connection is terminated.

Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

The socket primitives for TCP

The socket API is often used with the TCP protocol to provide a connection-oriented service called a **reliable byte stream**. Two examples are **SCTP** (**Stream Control Transmission Protocol**) defined in RFC 4960 and **SST** (**Structured Stream Transport**) (Ford, 2007).



The working principal of Berkeley socket

Example of socket programming (or) socket programming with TCP and UDP Socket Programming Example:
Internet File Server

The code has many limitations (discussed below), but in principle the server code can be compiled and run on any UNIX system connected to the Internet. The client code can be compiled and run on any other UNIX machine on the Internet, anywhere in the world.

```
/* This page contains a client program that can request a file from the server program
 * on the next page. The server responds by sending the whole file.
 */

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

#define SERVER_PORT 12345               /* arbitrary, but client & server must agree */
#define BUF_SIZE 4096                   /* block transfer size */

int main(int argc, char **argv)
{
  int c, s, bytes;
  char buf[BUF_SIZE];                   /* buffer for incoming file */
  struct hostent *h;                    /* info about server */
  struct sockaddr_in channel;           /* holds IP address */

  if (argc != 3) fatal("Usage: client server-name file-name");
  h = gethostbyname(argv[1]);           /* look up host's IP address */
  if (!h) fatal("gethostbyname failed");

  s = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
  if (s <0) fatal("socket");
  memset(&channel, 0, sizeof(channel));
  channel.sin_family= AF_INET;
  memcpy(&channel.sin_addr.s_addr, h->h_addr, h->h_length);
  channel.sin_port= htons(SERVER_PORT);

  c = connect(s, (struct sockaddr *) &channel, sizeof(channel));
  if (c < 0) fatal("connect failed");

  /* Connection is now established. Send file name including 0 byte at end. */
  write(s, argv[2], strlen(argv[2])+1);

  /* Go get the file and write it to standard output. */
  while (1) {
      bytes = read(s, buf, BUF_SIZE);   /* read from socket */
      if (bytes <= 0) exit(0);          /* check for end of file */
      write(1, buf, bytes);             /* write to standard output */
  }
}

fatal(char *string)
{
  printf("%s\n", string);
  exit(1);
}
```

Client code using sockets

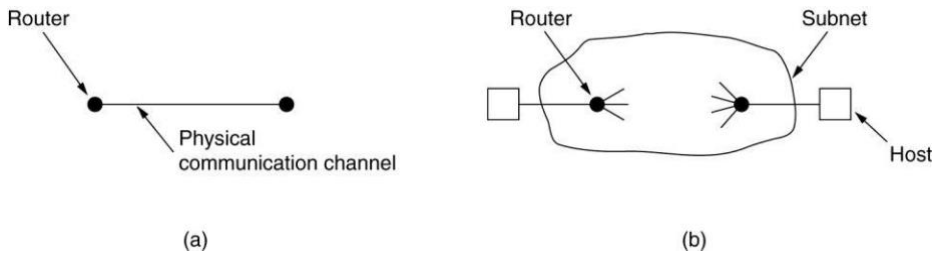
The client code starts with some includes and declarations. Execution begins by checking to see if it has been called with the right number of arguments (*argc* = 3 means the program name plus two arguments). Note that *argv* [1] contains the name of the server (e.g., *flits.cs.vu.nl*) and is converted to an IP address by *gethostbyname*. This function uses DNS to look up the name. Next, a socket is created and initialized. After that, the client attempts to establish a TCP connection to the server, using *connect*. If the server is up and running on the named machine and attached to *SERVER PORT* and is either idle or has room in its *listen* queue, the connection will (eventually) be established.

Using the connection, the client sends the name of the file by writing on the socket. The number of bytes sent is one larger than the name proper, since the 0 byte terminating the name must also be sent to tell the server where the name ends. Now the client enters a loop, reading the file block by block from the socket and copying it to standard output. When it is done, it just exits. The procedure *fatal* prints an error message and exits. The server needs the same procedure, but it was omitted due to lack of space on the page. Since the client and server are compiled separately and normally run on different computers, they cannot share the code of *fatal*.

4. Explain in detail about Elements of Transport Protocols.

- Addressing
- Connection Establishment
- Connection Release
  -Asymmetric release
  - Symmetric release
- Flow Control and Buffering
  -Chained Fixed-size Buffers
  -Chained Variable-size Buffers
  - One large Circular Buffer per Connection
- Multiplexing
  -Upward Multiplexing
  -Downward multiplexing

The transport service is implemented by a **transport protocol** used between the two transport entities. To deal with error control, sequencing, and flow control, among other issues. At the data link layer, two routers communicate directly via a physical channel, whether wired or wireless, whereas at the transport layer, this physical channel is replaced by the entire network as shown in fig.



(a) Environment of the data link layer. (b) Environment of the transport layer.

1. ADDRESSING

When an application (e.g., a user) process wishes to set up a connection to a remote application process, it must specify which one to connect to. The method normally used is to define transport addresses to which processes can listen for connection requests. In the Internet, these endpoints are called **ports**.

There are two types of access points.

TSAP (Transport Service Access Point) to mean a specific endpoint in the transport layer.

The analogous endpoints in the network layer (i.e., network layer addresses) are not surprisingly called **NSAPs (Network Service Access Points).** IP addresses are examples of NSAPs.

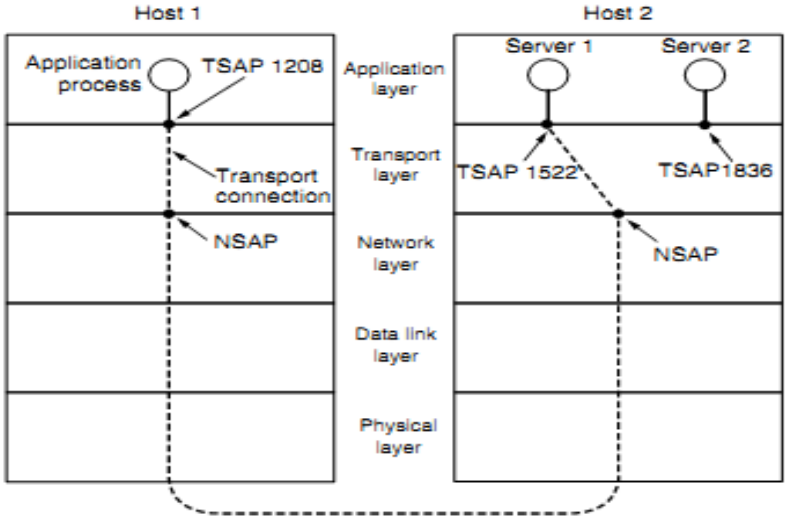


Fig 4.5: TSAP and NSAP network connections

Application processes, both clients and servers, can attach themselves to a local TSAP to establish a connection to a remote TSAP. These connections run through NSAPs on each host. The purpose of having TSAPs is that in some networks, each computer has a single NSAP, so some way is needed to distinguish multiple transport endpoints that share that NSAP.

A possible scenario for a transport connection is as follows:

1. A mail server process attaches itself to TSAP 1522 on host 2 to wait for an incoming call. How a process attaches itself to a TSAP is outside the networking model and depends entirely on the local operating system. A call such as our LISTEN might be used, for example.
2. An application process on host 1 wants to send an email message, so it attaches itself to TSAP 1208 and issues a CONNECT request. The request specifies TSAP 1208 on host 1 as the source and TSAP 1522 on host 2 as the destination. This action ultimately results in a transport connection being established between the application process and the server.
3. The application process sends over the mail message.
4. The mail server responds to say that it will deliver the message.
5. The transport connection is released.

2. CONNECTION ESTABLISHMENT:

With packet lifetimes bounded, it is possible to devise a fool proof way to establish connections safely. Packet lifetime can be bounded to a known maximum using one of the following techniques:

- Restricted subnet design
- Putting a hop counter in each packet
- Time stamping in each packet

Using a 3-way hand shake, a connection can be established. This establishment protocol doesn't require both sides to begin sending with the same sequence number.

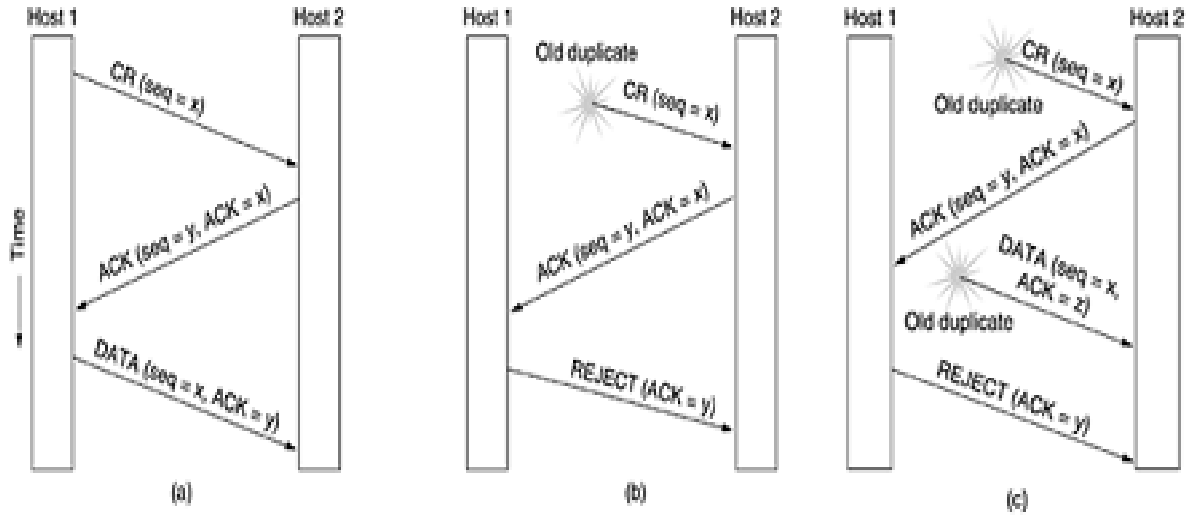


Fig 4.6: Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNEC TION REQUEST (a) Normal operation. (b) Old duplicate CONNECTION REQUEST appearing out of nowhere. (c) Duplicate CONNECTION REQUEST and duplicate ACK .

- The **first technique** includes any method that prevents packets from looping, combined with some way of bounding delay including congestion over the longest possible path. It is difficult, given that internets may range from a single city to international in scope.

- The **second method** consists of having the hop count initialized to some appropriate value and decremented each time the packet is forwarded. The network protocol simply discards any packet whose hop counter becomes zero.
- The **third method** requires each packet to bear the time it was created, with the routers agreeing to discard any packet older than some agreed-upon time.

In **fig (A)** Tomlinson (1975) introduced the **three-way handshake**.

- This establishment protocol involves one peer checking with the other that the connection request is indeed current. Host 1 chooses a sequence number, x , and sends a CONNECTION REQUEST segment containing it to host 2. Host 2replies with an ACK segment acknowledging x and announcing its own initial sequence number, y.

- Finally, host 1 acknowledges host 2's choice of an initial sequence number in the first data segment that it sends

In **fig (B)** the first segment is a delayed duplicate CONNECTION REQUEST from an old connection.

- This segment arrives at host 2 without host 1's knowledge. Host 2 reacts to this segment by sending host1an ACK segment, in effect asking for verification that host 1 was indeed trying to set up a new connection.
- When host 1 rejects host 2's attempt to establish a connection, host 2 realizes that it was tricked by a delayed duplicate and abandons the connection. In this way, a delayed duplicate does no damage.

- The worst case is when both a delayed CONNECTION REQUEST and an ACK are floating around in the subnet.

In **fig (C)** previous example, host 2 gets a delayed CONNECTION REQUEST and replies to it.

- At this point, it is crucial to realize that host 2 has proposed using y as the initial sequence number for host 2 to host 1 traffic, knowing full well that no segments containing sequence number y or acknowledgements to y are still in existence.
- When the second delayed segment arrives at host 2, the fact that z has been acknowledged rather than y tells host 2 that this, too, is an old duplicate.

- The important thing to realize here is that there is no combination of old segments that can cause the protocol to fail and have a connection set up by accident when no one wants it.

3.CONNECTION RELEASE:

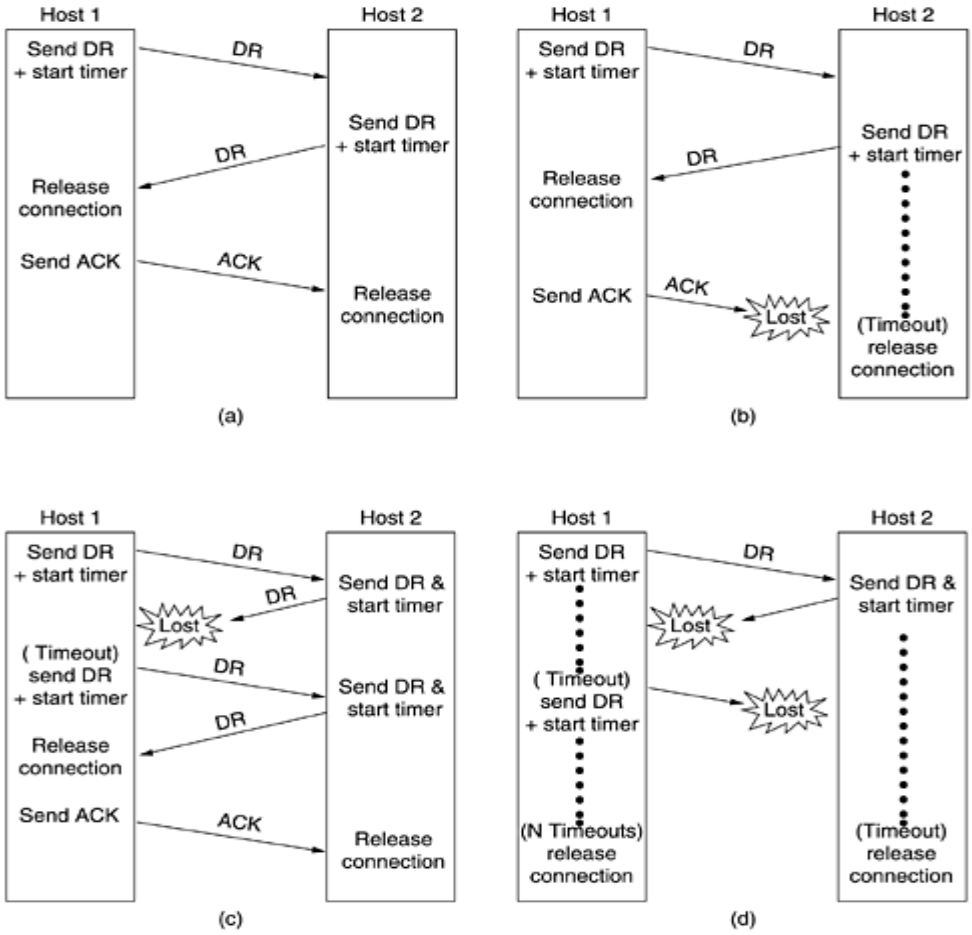
A connection is released using either asymmetric or symmetric variant. But, the improved protocol for releasing a connection is a 3-way handshake protocol.

There are two styles of terminating a connection:

1) Asymmetric release and
2) Symmetric release.

Asymmetric release is the way the telephone system works: when one party hangs up, the connection is broken. **Symmetric release** treats the connection as two separate unidirectional connections and requires each one to be released separately.

Fig-(a)	Fig-(b)	Fig-(c)	Fig-(d)
One of the user sends a DISCONNECTION REQUEST TPDU in order to initiate connection release. When it arrives, the recipient sends back a DR-TPDU, too, and starts a timer. When this DR arrives, the original sender sends back an ACK-TPDU and releases the connection. Finally, when the ACK-TPDU arrives, the receiver also releases the connection.	Initial process is done in the same way as in fig-(a). If the final ACK-TPDU is lost, the situation is saved by the timer. When the timer is expired, the connection is released.	If the second DR is lost, the user initiating the disconnection will not receive the expected response, and will timeout and starts all over again.	Same as in fig-(c) except that all repeated attempts to retransmit the DR is assumed to be failed due to lost TPDUs. After 'N' entries, the sender just gives up and releases the connection.



4.FLOW CONTROL AND BUFFERING:

Flow control is done by having a sliding window on each connection to keep a fast transmitter from over running a slow receiver. Buffering must be done by the sender, if the network service is unreliable. The sender buffers all the TPDUs sent to the receiver. The buffer size varies for different TPDUs. They are:

a) Chained Fixed-size Buffers
b) Chained Variable-size Buffers
c) One large Circular Buffer per Connection

(a). Chained Fixed-size Buffers:

If most TPDUs are nearly the same size, the buffers are organized as a pool of identical size buffers, with one TPDU per buffer.

(b). Chained Variable-size Buffers:

This is an approach to the buffer-size problem. i.e., if there is wide variation in TPDU size, from a few characters typed at a terminal to thousands of characters from file transfers, some problems may occur:

- If the buffer size is chosen equal to the largest possible TPDU, space will be wasted whenever a short TPDU arrives.

- If the buffer size is chosen less than the maximum TPDU size, multiple buffers will be needed for long TPDUs.

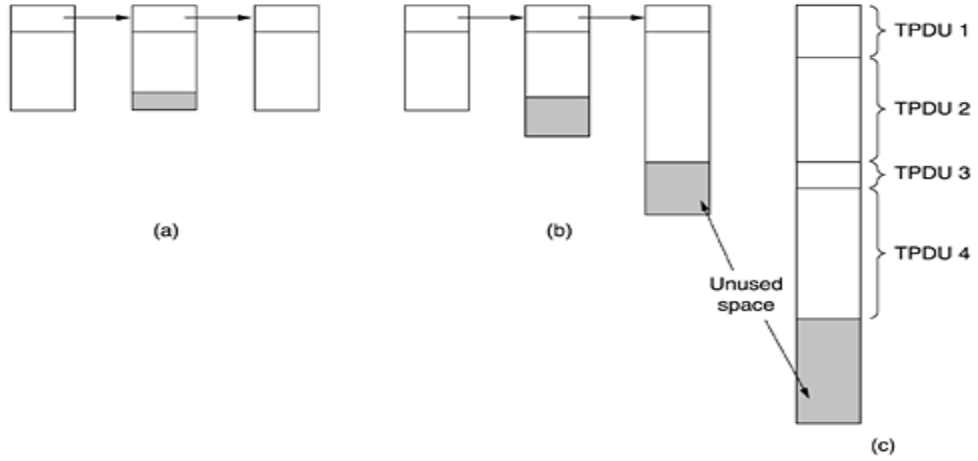
To overcome these problems, we employ variable-size buffers.

(c). One large Circular Buffer per Connection:

A single large circular buffer per connection is dedicated when all connections are heavily loaded.

1. Source Buffering is used for low band width bursty traffic
2. Destination Buffering is used for high band width smooth traffic.
3. Dynamic Buffering is used if the traffic pattern changes randomly.

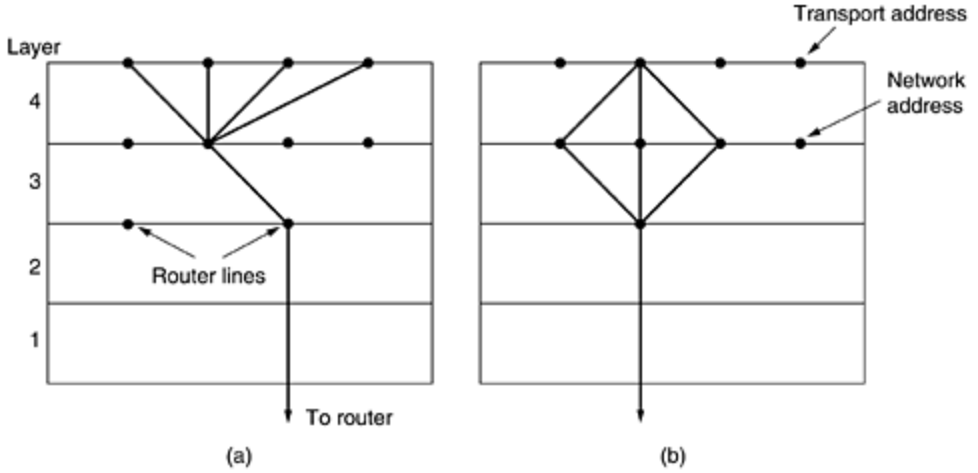
Figure 6-15. (a) Chained fixed-size buffers. (b) Chained variable-sized buffers. (c) One large circular buffer per connection.



5.MULTIPLEXING:

In networks that use virtual circuits within the subnet, each open connection consumes some table space in the routers for the entire duration of the connection. If buffers are dedicated to the virtual circuit in each router as well, a user who left a terminal logged into a remote machine, there is need for multiplexing. There are 2 kinds of multiplexing:

Figure 6-17. (a) Upward multiplexing. (b) Downward multiplexing



(a). UP-WARD MULTIPLEXING:

In the below figure, all the 4 distinct transport connections use the same network connection to the remote host. When connect time forms the major component of the carrier's bill, it is up to the

transport layer to group port connections according to their destination and map each group onto the minimum number of port connections.

(b). <u>DOWN-WARD MULTIPLEXING:</u>

- If too many transport connections are mapped onto the one network connection, the performance will be poor.
- If too few transport connections are mapped onto one network connection, the service will be expensive.

The possible solution is to have the transport layer open multiple connections and distribute the traffic among them on round-robin basis, as indicated in the below figure:
With 'k' network connections open, the effective band width is increased by a factor of 'k'.

3.Explain in detail about the principle of congestion control?

Congestion Control

Congestion occurs at routers, so it is detected at the network layer. However, congestion is ultimately caused by traffic sent into the network by the transport layer. The only effective way to control congestion is for the transport protocols to send packets into the network more slowly.

- Desirable bandwidth allocation

- Efficience & Power
-Max-Min Fairness
-Convergence

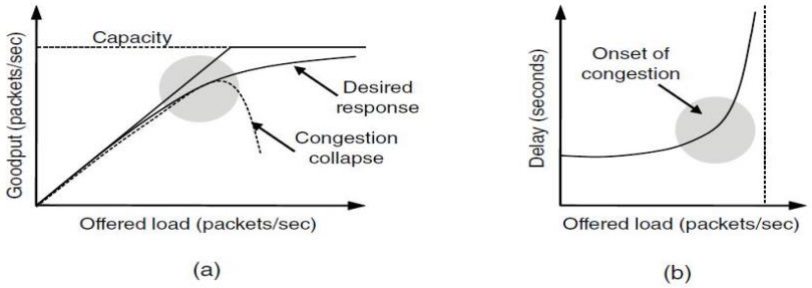
- Regulating the sending rate
  - AIMD (Additive Increase Multiplicative Decrease)

Desirable Bandwidth Allocation (1)

It is to find a good allocation of bandwidth to the transport entities that are using the network. A good allocation will deliver good performance because it uses all the available bandwidth but avoids congestion, it will be fair across competing transport entities, and it will quickly track changes in traffic demands.

Efficiency and power

Consider there is a 100-Mbps link, five transport entities should get 20 Mbps each. They should usually get less than 20 Mbps for good performance. The reason is that the traffic is often bursty. The goodput (or rate of useful packets arriving at the receiver) as a function of the offered load. This curve and a matching curve for the delay as a function of the offered load are given in Fig. As the load increases in Fig(a) goodput initially increases at the same rate, but as the load approaches the capacity, goodput rises more gradually. This falloff is because bursts of traffic can occasionally mount up and cause some losses at buffers inside the network. If the transport protocol is poorly designed and retransmits packets that have been delayed but not lost, the network can enter congestion collapse. The corresponding delay is given in Fig(b) Initially the delay is fixed, representing the propagation delay across the network. As the load approaches the capacity, the delay rises, slowly at first and then much more rapidly. This is again because of bursts of traffic that tend to mound up at high load.

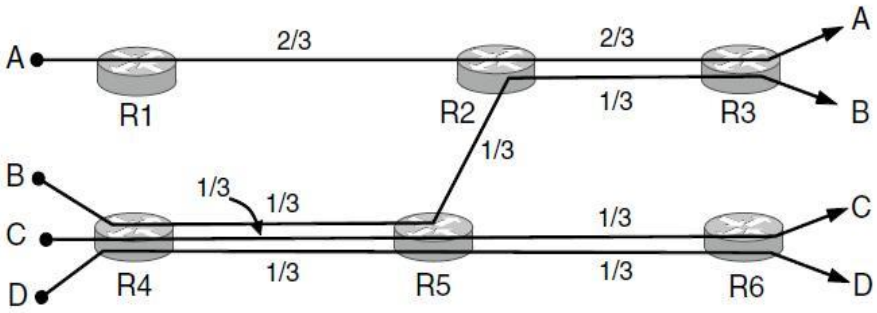


(a) Goodput and (b) delay as a function of offered load

Max-Min Fairness

An allocation is max-min fair if the bandwidth given to one flow cannot be increased without decreasing the bandwidth given to another flow with an allocation that is no larger. That is, increasing the bandwidth of a flow will only make the situation worse for flows that are less well off. A max-min fair allocation is shown for a network with four flows, *A*, *B*, *C*, and *D*, in Fig. Each of the links between routers has the same capacity, taken to be 1 unit, though in the general case the links will have different capacities. Three flows compete for the bottom-left link between routers *R4* and *R5*. Each of these flows therefore gets 1/3 of the link.

The remaining flow, *A*, competes with *B* on the link from *R2* to *R3*. Since *B* has an allocation of 1/3, *A* gets the remaining 2/3 of the link. Notice that all of the other links have spare capacity. However, this capacity cannot be given to any of the flows without decreasing the capacity of another, lower flow. For example, if more of the bandwidth on the link between *R2* and *R3* is given to flow *B*, there will be less for flow *A*. This is reasonable as flow *A* already has more bandwidth. However, the capacity of flow *C* or *D* (or both) must be decreased to give more bandwidth to *B*, and these flows will have less bandwidth than *B*. Thus, the allocation is max-min fair.

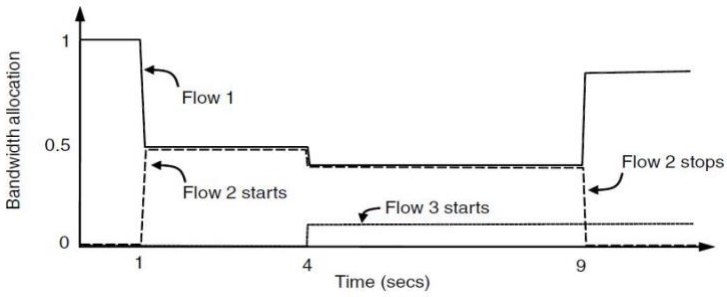


Max-min bandwidth allocation for four flows

Convergence

Initially, flow 1 has all of the bandwidth. One second later, flow 2 starts. It needs bandwidth as well. The allocation quickly changes to give each of these flows half the bandwidth. At 4 seconds, a third flow joins. However, this flow uses only 20% of the bandwidth, which is less than its fair share (which is a third). Flows 1 and 2 quickly adjust, dividing the available bandwidth to each have 40% of

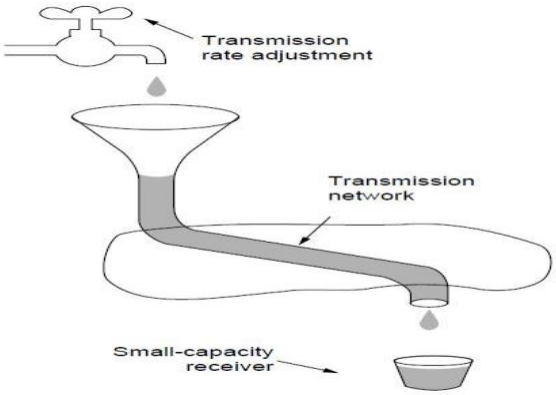
the bandwidth. At 9 seconds, the second flow leaves, and the third flow remains unchanged. The first flow quickly captures 80% of the bandwidth. At all times, the total allocated bandwidth is approximately 100%, so that the network is fully used, and competing flows get equal treatment.(but do not have to use more bandwidth than they need).



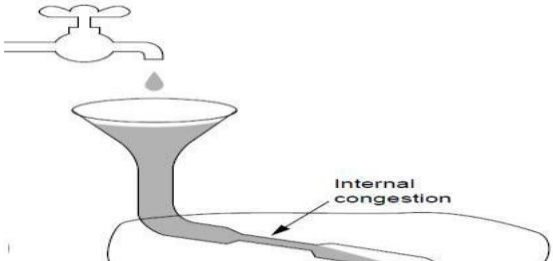
Changing bandwidth allocation over time

Regulating the Sending Rate

How do we regulate the sending rates to obtain a desirable bandwidth allocation? The sending rate may be limited by two factors. The first is flow control, in the case that there is insufficient buffering at the receiver. The second is congestion, in the case that there is insufficient capacity in the network. In Fig(a), we see a thick pipe leading to a small-capacity receiver. This is a flow-control limited situation. As long as the sender does not send more water than the bucket can contain, no water will be lost. In Fig(b), the limiting factor is not the bucket capacity, but the internal carrying capacity of the network. If too much water comes in too fast, it will back up and some will be lost (in this case, by overflowing the funnel). These cases may appear similar to the sender, as transmitting too fast causes packets to be lost.



A fast network feeding a low-capacity receiver



A slow network feeding a high-capacity receiver

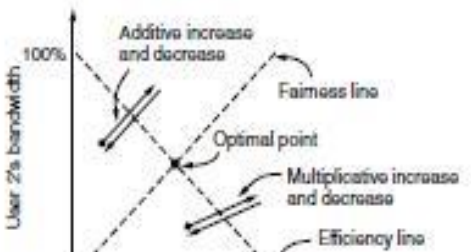
FAST TCP measures the round trip delay and uses that metric as a signal to avoid congestion. Finally, in the form of congestion control most prevalent in the Internet today, TCP with drop-tail or RED routers, packet loss is inferred and used to signal that the network has become congested. There are many variants of this form of TCP, including CUBIC TCP, which is used in Linux. Combinations are also possible. For example, Windows includes Compound TCP that uses both packet loss and delay as feedback signals. For example, if XCP tells senders the rate to use, the senders may simply use that rate. When a congestion signal is given, the senders should decrease their rates. The way in which the rates are increased or decreased is
given by a **control law**.

Protocol	Signal	Explicit?	Precise?
XCP	Rate to use	Yes	Yes
TCP with ECN	Congestion warning	Yes	No
FAST TCP	End-to-end delay	No	Yes
Compound TCP	Packet loss & end-to-end delay	No	Yes
CUBIC TCP	Packet loss	No	No
TCP	Packet loss	No	No

Signals of some congestion control protocols

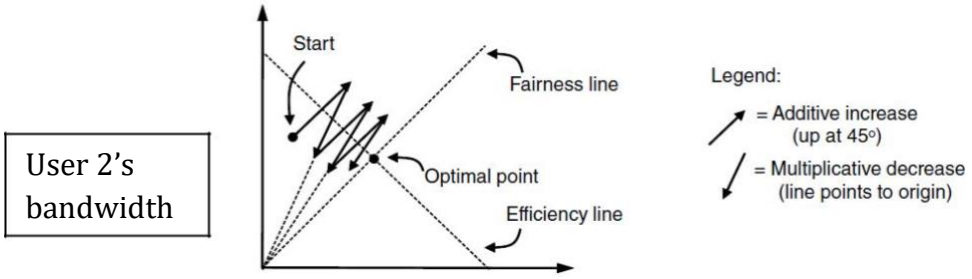
AIMD (**Additive Increase Multiplicative Decrease**) is the appropriate control law to arrive at the efficient and fair operating point. The graph in Fig shows the bandwidth allocated to user 1 on the x-axis and to user 2 on the y-axis. When the allocation is fair, both users will receive the same amount of bandwidth. This is shown by the dotted fairness line. When the allocations sum to 100%, the capacity of the link, the allocation is efficient. This is shown by the dotted efficiency line. A congestion signal is given by the network to both users when the sum of their allocations crosses this line.

The intersection of these lines is the desired operating point, when both users have the same bandwidth and all of the network bandwidth is used. Consider what happens from some starting allocation if both user 1 and user 2 additively increase their respective bandwidths over time. For example, the users may each increase their sending rate by 1 Mbps every second. Eventually, the operating point crosses the efficiency line and both users receive a congestion signal from the network. At this stage, they must reduce their allocations. However, an additive decrease would simply cause them to oscillate along an additive line.



Additive and multiplicative bandwidth adjustments

Now consider the case that the users additively increase their bandwidth allocations and then multiplicatively decrease them when congestion is signaled. This behavior is the AIMD control law, and it is shown in Fig. It can be seen that the path traced by this behavior does converge to the optimal point that is both fair and efficient. By the same argument, the only other combination, multiplicative increase and additive decrease, would diverge from the optimal point. AIMD is the control law that is used by TCP, based on this argument and another stability argument (that it is easy to drive the network into congestion and difficult to recover, so the increase policy should be gentle and the decrease policy aggressive). Instead of adjusting the rate directly, a strategy that is often used in practice is to adjust the size of a sliding window. TCP uses this strategy. If the window size is *W* and the round-trip time is *RTT*, the equivalent rate is *W/RTT*. This strategy is easy to combine with flow control, which already uses a window, and has the advantage that the sender paces packets using acknowledgements and hence slows down in one *RTT* if it stops receiving reports that packets are leaving the network.



User 1's allocation

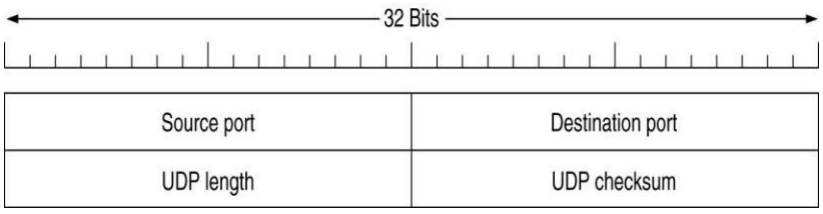
Additive Increase Multiplicative Decrease (AIMD) control law

4.Explain Internet Transport Protocol – UDP(USER DATAGRAM PROTOCOL)

- UDP
- Remote Procedure Call (RPC)

UDP (or) UDP SEGMENT STRUCTURE

UDP (User Datagram Protocol) provides a way for applications to send encapsulated IP datagrams and send them without having to establish a connection. UDP transmits segments consisting of an 8-byte header followed by the payload.



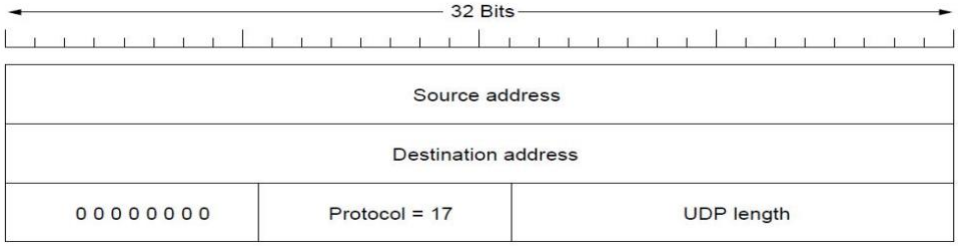
The UDP header.

UDP transmits **segments** consisting of an 8-byte header followed by the payload. The header is shown in Fig. The two **ports** serve to identify the endpoints within the source and destination machines. When a UDP packet arrives, its payload is handed to the process attached to the destination port. This attachment occurs when the BIND primitive or something similar is used in TCP (the binding process is the same for UDP).

- The source port is primarily needed when a reply must be sent back to the source. By copying the *Source port* field from the incoming segment into the *Destination port* field of the outgoing segment, the process sending the reply can specify which process on the sending machine is to get it.
- The *UDP length* field includes the 8-byte header and the data. The minimum length is 8 bytes, to cover the header. The maximum length is 65,515 bytes, which is lower than the largest number that will fit in 16 bits because of the size limit on IP packets.

- An optional *Checksum* is also provided for extra reliability. It checksums the header, the data, and a conceptual IP pseudoheader. When performing this computation, the *Checksum* field is set to zero and the data field is padded out with an additional zero byte if its length is an odd number.

The pseudoheader for the case of IPv4 is shown in Fig. It contains the 32-bit IPv4 addresses of the source and destination machines, the protocol number for UDP (17), and the byte count for the UDP segment (including the header). It is different but analogous for IPv6. Including the pseudoheader in the UDP checksum computation helps detect misdelivered packets, but including it also violates the protocol hierarchy since the IP addresses in it belong to the IP layer, not to the UDP layer. TCP uses the same pseudoheader for its checksum.



The IPv4 pseudoheader included in the UDP checksum.

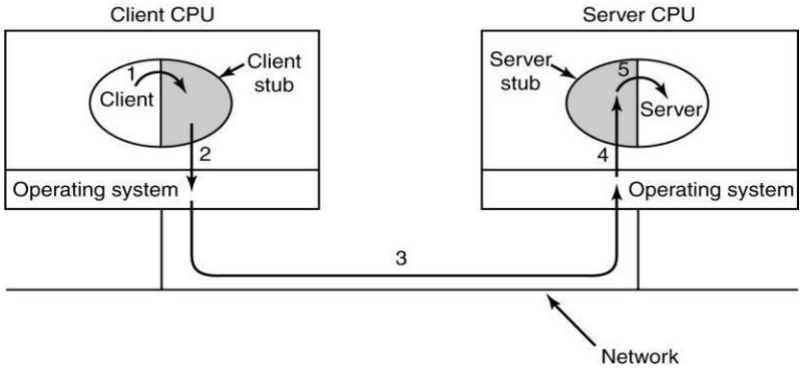
REMOTE PROCEDURE CALL(RPC)

When a process on machine 1 calls a procedure on machine 2, the calling process on 1 is suspended and execution of the called procedure takes place on 2. Information can be transported from the caller to the callee in the parameters and can come back in the procedure result. No message passing is visible to the application programmer. This technique is known as **RPC** (**Remote Procedure Call**) and has become the basis for many networking applications.

The idea behind RPC is to make a remote procedure call look as much as possible like a local one. To call a remote procedure, the client program must be bound with a small library procedure, called the

client stub, that represents the server procedure in the client's address space. Similarly, the server is bound with a procedure called the server stub. These procedures hide the fact that the procedure call from the client to the server is not local.

- Step 1 is the client calling the client stub. This call is a local procedure call, with the parameters pushed onto the stack in the normal way.
- Step 2 is the client stub packing the parameters into a message and making a system call to send the message. Packing the parameters is called **marshaling**.
- Step 3 is the operating system sending the message from the client machine to the server machine.
- Step 4 is the operating system passing the incoming packet to the server stub.
- Finally, step 5 is the server stub calling the server procedure with the unmarshaled parameters. The reply traces the same path in the other direction.



Steps in making a remote procedure call. The stubs are shaded.

5.Explain Internet Transport Protocol – TCP (TRANSMISSION CONTROL PROTOCOL)

TCP (Transmission Control Protocol) was specifically designed to provide a reliable end-to-end byte stream over an unreliable internetwork. An internetwork differs from a single network because different parts may have wildly different topologies, bandwidths, delays, packet sizes, and other parameters. TCP was designed to dynamically adapt to properties of the internetwork and to be robust in the face of many kinds of failures.

1. TCP Service Model
2. The TCP Segment Header
3. Connection Establishment
4. Connection Release
5. TCP Transmission Policy
6. TCP Sliding Window
7. TCP Timer Management.
8. TCP Congestion Control

The TCP Service Model

TCP service is obtained by both the sender and receiver creating end points, called sockets. Each socket has a socket number (address) consisting of the IP address of the host and a 16-bit number local to that host, called a port. A port is the TCP name for a TSAP. For TCP service to be obtained, a connection must be explicitly established between a socket on the sending machine and a socket on the receiving machine.

- full duplex and point-to-point
- byte stream
- immediate data
- urgent data

Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	E-mail
69	TFTP	Trivial File Transfer Protocol
79	Finger	Lookup info about a user
80	HTTP	World Wide Web
110	POP-3	Remote e-mail access
119	NNTP	USENET news

Some assigned ports.

TCP Header (or) TCP segment structure

- **Source and destination port: These** fields identify the local endpoint of the connection. Each host may decide for itself how to allocate its own ports starting at 1024. The source and destination socket numbers together identify the connection.

- **Sequence and ACK number:** This field is used to give a sequence number to each and every byte transferred. This has an advantage over giving the sequence numbers to every packet because data of many small packets can be combined into one at the time of retransmission, if needed. The ACK signifies the next byte expected from the source and not the last byte received. The ACKs are cumulative instead of selective.Sequence number space is as large as 32-bit although 17 bits would have been enough if the packets were delivered in order. If packets reach in order, then according to the following formula:

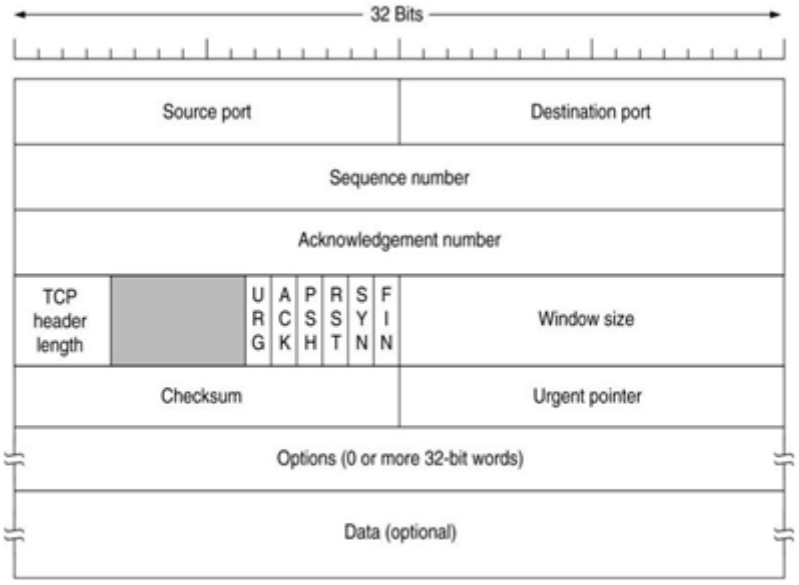
  (sender's window size) + (receiver's window size) < (sequence number space)

  The sequence number space should be 17-bits. But packets may take different routes and reach out of order. So, we need a larger sequence number space. And for optimisation, this is 32-bits.

- **Header length: This** field tells how many 32-bit words are contained in the TCP header. This is needed because the options field is of variable length.
- **Flags:** There are six one-bit flags.

1. **URG:** This bit indicates whether the urgent pointer field in this packet is being used.
2. **ACK: This** bit is set to indicate the ACK number field in this packet is valid.
3. **PSH:** This bit indicates PUSHed data. The receiver is requested to deliver the data to the application upon arrival and not buffer it until a full buffer has been received.
4. **RST:** This flag is used to reset a connection that has become confused due to a host crash or some other reason.It is also used to reject an invalid segment or refuse an attempt to open a connection. This causes an abrupt end to the connection, if it existed.
5. **SYN:** This bit is used to establish connections. The connection request(1st packet in 3-way handshake) has SYN=1 and ACK=0. The connection reply (2nd packet in 3-way handshake) has SYN=1 and ACK=1.
6. **FIN:** This bit is used to release a connection. It specifies that the sender has no more fresh data to transmit. However, it will retransmit any lost or delayed packet. Also, it will continue to receive data from other side. Since SYN and FIN packets have to be acknowledged, they must have a sequence number even if they do not contain any data.

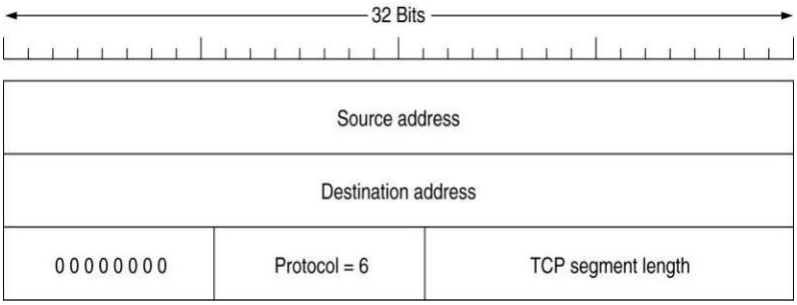
- **Window Size:** Flow control in TCP is handled using a variable-size sliding window. The Window Size field tells how many bytes may be sent starting at the byte acknowledged. Sender can send the bytes with sequence number between (ACK#) to (ACK# + window size - 1) A window size of zero is legal and says that the bytes up to and including ACK# -1 have been received, but the receiver would like no more data for the moment. Permission to send can be granted later by sending a segment with the same ACK number and a nonzero Window Size field.
- **Checksum:** This is provided for extreme reliability. It checksums the header, the data, and the conceptual pseudoheader. The pseudoheader contains the 32-bit IP address of the source and destination machines, the protocol number for TCP(6), and the byte count for the TCP segment (including the header).Including the pseudoheader in TCP checksum computation helps detect misdelivered packets, but doing so violates the protocol hierarchy since the IP addresses in it belong to the IP layer, not the TCP layer.
- **Urgent Pointer:** Indicates a byte offset from the current sequence number at which urgent data are to be found. Urgent data continues till the end of the segment. This is not used in practice. The same effect can be had by using two TCP connections, one for transferring urgent data.
- **Options:** Provides a way to add extra facilities not covered by the regular header. eg,
    - Maximum TCP payload that sender is willing to handle. The maximum size of segment is called MSS (Maximum Segment Size). At the time of handshake, both parties inform each other about their capacity. Minimum of the two is honoured.
      This information is sent in the options of the SYN packets of the three way handshake.
    - Window scale option can be used to increase the window size. It can be specified by telling the receiver that the window size should be interpreted by shifting it left by specified number of bits. This header option allows window size up to 230.
- **Data:** This can be of variable size. TCP knows its size by looking at the IP size header.



TCP Header

Checksum

A Checksum is also provided for extra reliability. It checksums the header, the data, and the conceptual pseudo header shown in Fig.



The pseudoheader included in the TCP checksum.

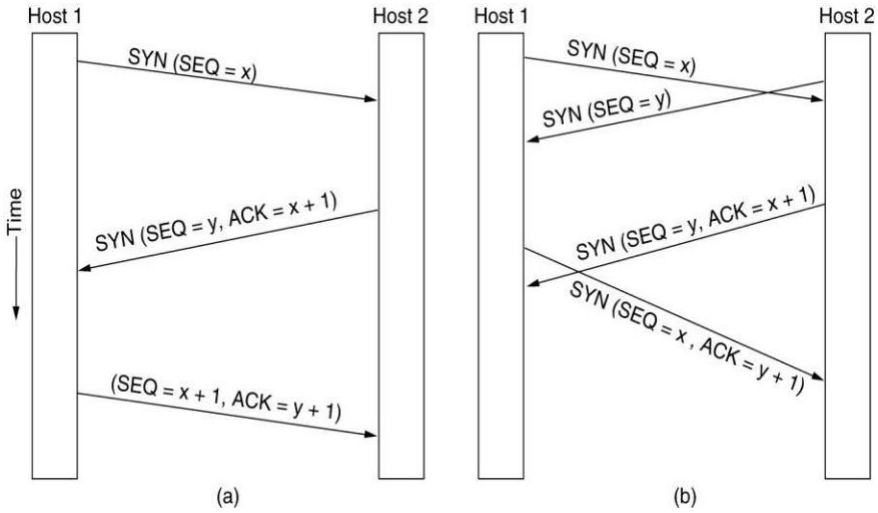
TCP Connection Establishment

Connections are established in TCP by means of the three-way handshake are discussed To establish a connection, one side, say, the server, passively waits for an incoming connection by executing the LISTEN and ACCEPT primitives in that order, either specifying a specific source or nobody in particular. The other side, say, the client, executes a CONNECT primitive, specifying the IP address and port to which it wants to connect, the maximum TCP segment size it is willing to accept, and optionally some user data (e.g., a password).

The CONNECT primitive sends a TCP segment with the *SYN* bit on and *ACK* bit off and waits for a response. When this segment arrives at the destination, the TCP entity there checks to see if there is a process that has done a LISTEN on the port given in the *Destination port* field. If not, it

sends a reply with the *RST* bit on to reject the connection. If some process is listening to the port, that process is given the incoming TCP segment. It can either accept or reject the connection. If it accepts, an acknowledgement segment is sent back. The sequence of TCP segments sent in the normal case is shown in Fig(a).

Note that a *SYN* segment consumes 1 byte of sequence space so that it can be acknowledged unambiguously. In the event that two hosts simultaneously attempt to establish a connection between the same two sockets, the sequence of events is as illustrated in Fig(b). The result of these events is that just one connection is established, not two, because connections are identified by their end points. If the first setup results in a connection identified by (*x*, *y*) and the second one does too, only one table entry is made, namely, for (*x*, *y*).



(a) TCP connection establishment in the normal case.

(b) Call collision.

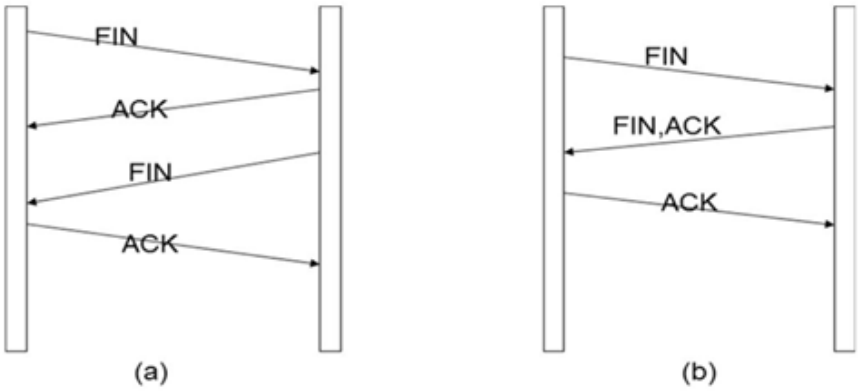
TCP Connection Release

Each simplex connection is released independently of its sibling. To release a connection, either party can send a TCP segment with the *FIN* bit set, which means that it has no more data to transmit. When the *FIN* is acknowledged, that direction is shut down for new data. Data may continue to flow indefinitely in the other direction, however. When both directions have been shut down, the connection is released. Normally, four TCP segments are needed to release a connection: one *FIN* and one *ACK* for each direction. However, it is possible for the first *ACK* and the second *FIN* to be contained in the same segment, reducing the total count to three. Just as with telephone calls in which both people say goodbye and hang up the phone simultaneously, both ends of a TCP connection may send *FIN* segments at the same time. These are each acknowledged in the usual way, and the connection is shut down. There is, in fact, no essential difference between the two hosts releasing sequentially or

simultaneously. To avoid the two-army problem, timers are used. If a response to a *FIN* is not forthcoming within two maximum packet lifetimes, the sender of the *FIN* releases the connection.

TCP Transmission Policy

- Window management is not directly tied to acks as in data link protocols



(a) Normally, four TCP segments are needed to release a connection

(b) It is possible for the first ACK and the second FIN to be contained in the same segment

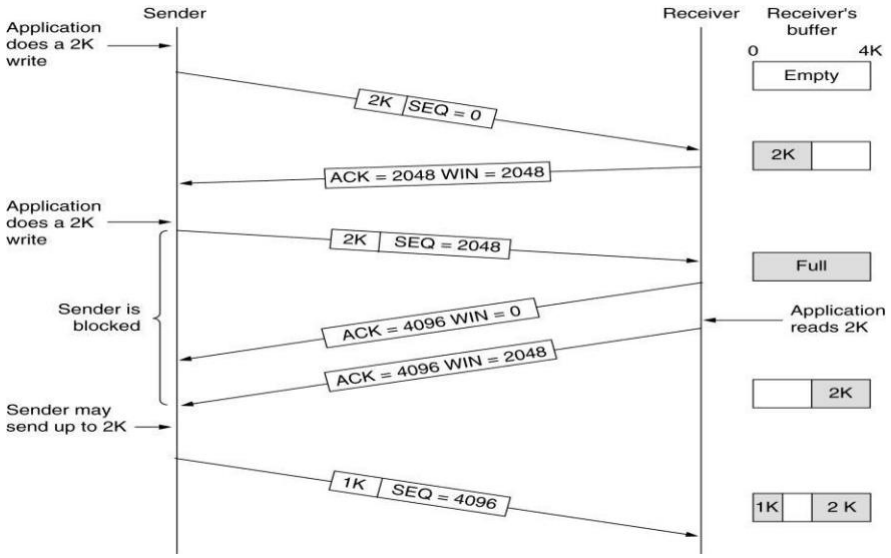
- Exclusive buffer messages manage the transmission
- If no buffer at receiver, then no transmission by sender except
  - Urgent data may be sent
  - One byte segment can be sent to ask receiver to renounce its buffer status (to prevent deadlock)
- Sender and receiver are not forced to transmit or receive as soon as they receive data from the application. This improves performance as follows:
  - If one byte messages are sent (like in TELNET) then use NAGLE's algorithm
    - Send first byte and keep the rest until ack comes back
    - As ack comes in send the rest and keep the further incoming bytes until ack is received

TCP Sliding Window

- Window management in TCP decouples the issues of acknowledgement of the correct receipt of segments and receiver buffer allocation. For example, suppose the receiver has a 4096-byte buffer, as shown in Fig. If the sender transmits a 2048-byte segment that is correctly received, the receiver will acknowledge the segment. However, since it now has only 2048 bytes of buffer space (until the application removes some data from the buffer), it will advertise a window of 2048 starting at the next byte expected. Now the sender transmits another 2048

bytes, which are acknowledged, but the advertised window is of size 0. The sender must stop until the application

process on the receiving host has removed some data from the buffer, at which time TCP can advertise a larger window and more data can be sent.

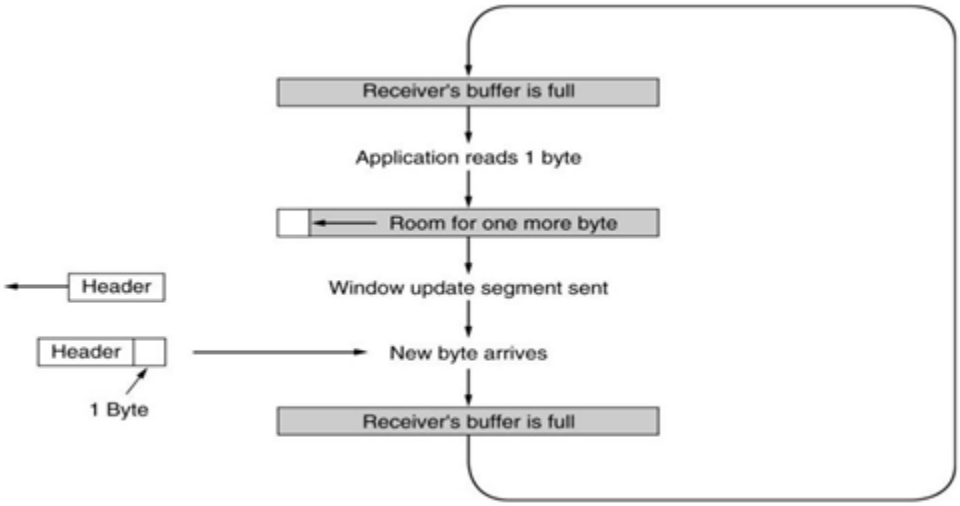


Window management in TCP

Nagle's Algorithm and Silly Window Syndrome

Silly Window Syndrome: Receive bytes one by one and send window messaging accordingly

- Clark’s Solution to Silly Window Syndrome
  - – Prevent receiver from sending one byte updates and make it wait until decent amount of space available before it sends buffer messages
  - – Sender may also pospone sending messages
- Nagle’s Algorithm and Clark’s solution are complementary and they can be used at the same time

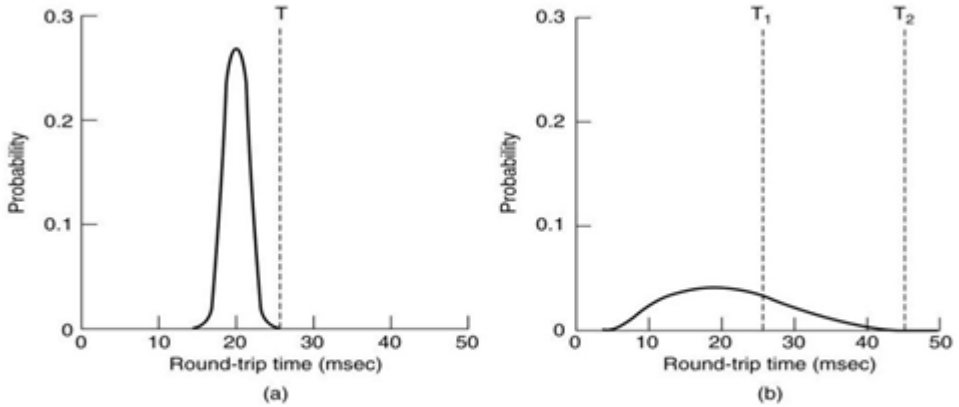


Silly window syndrome

TCP Timer Management

The solution of deciding the retransmission timer is to use a highly dynamic algorithm that constantly adjusts the timeout interval, based on continuous measurements of network performance. TCP uses multiple timers (at least conceptually) to do its work. The most important of these is the **RTO** (**Retransmission TimeOut**). When a segment is sent, a retransmission timer is started. If the segment is acknowledged before the timer expires, the timer is stopped. If, on the other hand, the timer goes off before the acknowledgement comes in, the segment is retransmitted (and the timer os started again). This problem is much more difficult in the transport layer than in data link protocols such as 802.11. In the latter case, the expected delay is measured in microseconds and is highly predictable (i.e., has a low variance), so the timer can be set to go off just slightly after the acknowledgement is expected, as shown in Fig(a).

Since acknowledgements are rarely delayed in the data link layer (due to lack of congestion), the absence of an acknowledgement at the expected time generally means either the frame or the acknowledgement has been lost. TCP is faced with a radically different environment. The probability density function for the time it takes for a TCP acknowledgement to come back looks more like Fig(b) than Fig(a). It is larger and more variable. Determining the round-trip time to the destination is tricky. Even when it is known, deciding on the timeout interval is also difficult. If the timeout is set too short, say, *T*1 in Fig(b), unnecessary retransmissions will occur, clogging the Internet with useless packets. If it is set too long (e.g., *T*2), performance will suffer due to the long retransmission delay whenever a packet is lost.



(a) Probability density of ACK arrival times in the data link layer.

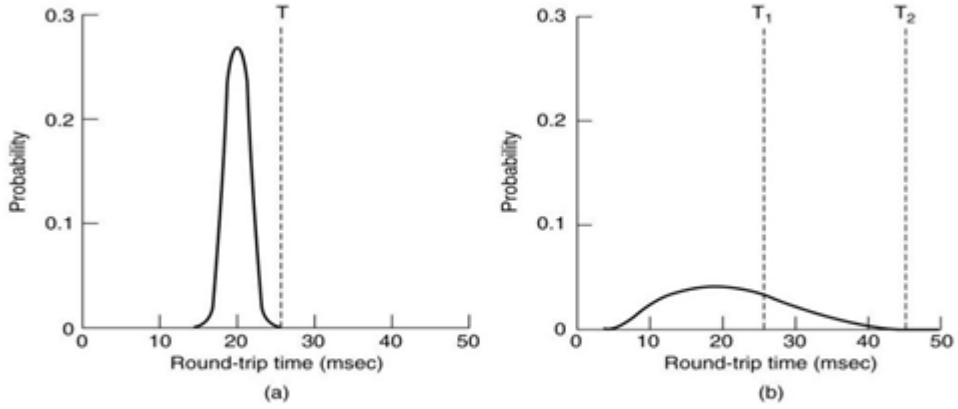
(b) Probability density of ACK arrival times for TCP.

Timeout

For each connection, TCP maintains a variable, RTT, that is the best current estimate of the round-trip time to the destination in question. When a segment is sent, a timer is started, both to see how long the acknowledgement takes and to trigger a retransmission if it takes too long. If the acknowledgement gets back before the timer expires, TCP measures how long the acknowledgement took, say, M. It then updates RTT according to the formula.

$RTT=\alpha\ RTT+(1-\alpha)M$

Typically a = 7/8.



(a) Probability density of ACK arrival times in the data link layer.

(b) Probability density of ACK arrival times for TCP.

Another smoothed variable, D, is the deviation, that is | RTT - M |.

$D=\alpha\ D+(1-\alpha)\ |\ RTT - M\ |$

Timeout=RTT+4D

For each connection, TCP maintains a variable, *SRTT* (Smoothed Round-Trip Time), that is the best current estimate of the round-trip time to the destination in question. When a segment is sent, a timer is started, both to see how long the acknowledgement takes and also to trigger a retransmission if it takes too long. If the acknowledgement gets back before the timer expires, TCP measures how long the acknowledgement took, say, *R.* It then updates *SRTT* according to the formula

$$SRTT = \alpha\, SRTT + (1 - \alpha)\, R$$

where α is a smoothing factor that determines how quickly the old values are forgotten. Typically, α = 7/8. This kind of formula is an **EWMA** (**Exponentially Weighted Moving Average**) or low-pass filter that discards noise in the samples.

Jacobson proposed making the timeout value sensitive to the variance in round-trip times as well as the smoothed round-trip time. This change requires keeping track of another

smoothed variable, *RTTVAR* (Round- Trip Time variation) that is updated using the formula

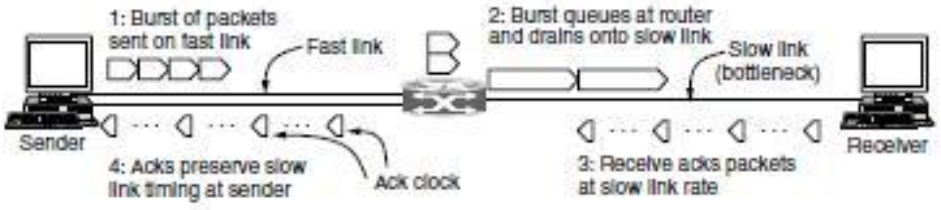
$$RTTVAR = \beta\, RTTVAR + (1 - \beta)\, |SRTT - R|$$

This is an EWMA as before, and typically β = 3/4. The retransmission timeout, *RTO,* is set to be

$$RTO = SRTT + 4 \times RTTVAR$$

TCP Congestion Control

What happens when a sender on a fast network (the 1-Gbps link) sends a small burst of four packets to a receiver on a slow network (the 1- Mbps link) that is the bottleneck or slowest part of the path. Initially the four packets travel over the link as quickly as they can be sent by the sender. At the router, they are queued while being sent because it takes longer to send a packet over the slow link than to receive the next packet over the fast link. But the queue is not large because only a small number of packets were sent at once. Note the increased length of the packets on the slow link. The same packet, of 1 KB say, is now longer because it takes more time to send it on a slow link than on a fast one.

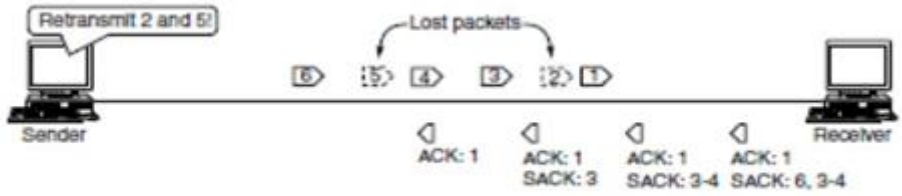


A burst of packets from a sender and the returning ack clock

Then the packets get to the receiver, where they are acknowledged. The times for the acknowledgements reflect the times at which the packets arrived at the receiver after crossing the slow link. They are spread out compared to the original packets on the fast link. As these acknowledgements travel over the network and back to the sender they preserve this timing.

A simple fix is the use of **SACK** (**Selective ACKnowledgements**), which lists up to three ranges of bytes that have been received. A receiver uses the TCP Acknowledgement number field in the normal manner, as a cumulative acknowledgement of the highest in-order byte that has been received. When it receives packet 3 out of order (because packet 2 was lost), it sends a SACK

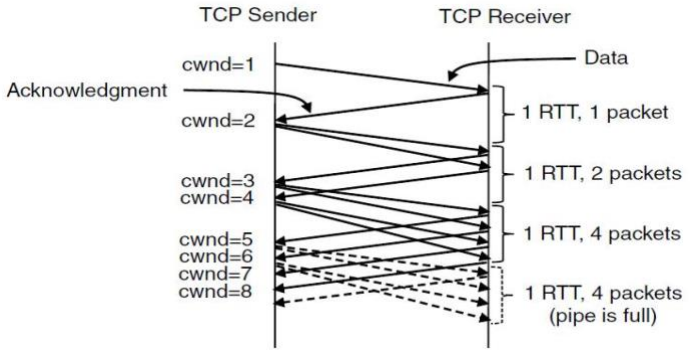
option for the received data along with the (duplicate) cumulative acknowledgement for packet 1. The SACK option gives the byte ranges that have been received above the number given by the cumulative acknowledgement. The first range is the packet that triggered the duplicate acknowledgement. The next ranges, if present, are older blocks. Up to three ranges are commonly used. By the time packet 6 is received, two SACK byte ranges are used to indicate that packet 6 and packets 3 to 4 have been received, in addition to all packets up to packet 1. From the information in each SACK option that it receives, the sender can decide which packets to retransmit. In this case, retransmitting packets 2 and 5 would be a good idea.



Selective Acknowledgements

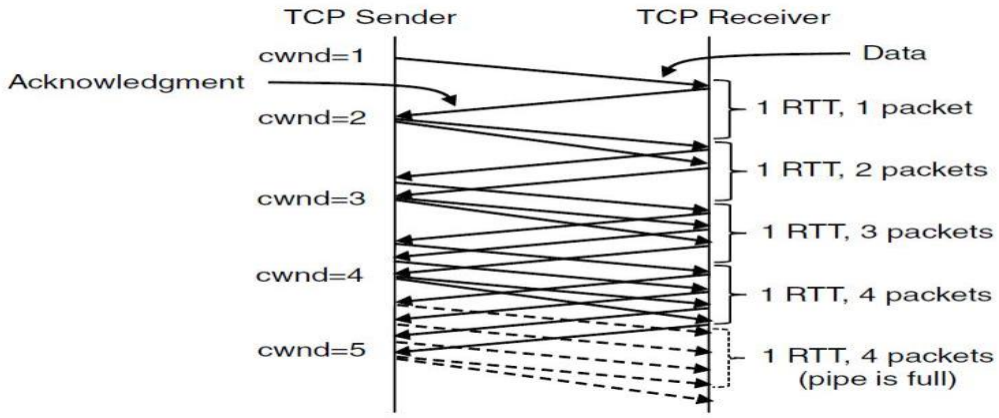
In the first round-trip time, the sender injects one packet into the network (and the receiver receives one packet). Two packets are sent in the next round-trip time, then four packets in the third round-trip time. When the sender gets an acknowledgement, it increases the congestion window by one and immediately sends two packets into the network. (One packet is the increase by one; the other packet is a replacement for the packet that has been acknowledged and left the network. At all times, the number of unacknowledged packets is given by the congestion window.) However, these two packets will not necessarily arrive at the receiver as closely spaced as when they were sent.

When the queues are full, one or more packets will be lost. After this happens, the TCP sender will time out when an acknowledgement fails to arrive in time. There is evidence of slow start growing too fast in Fig. After three RTTs, four packets are in the network. These four packets take an entire RTT to arrive at the receiver. That is, a congestion window of four packets is the right size for this connection. However, as these packets are acknowledged, slow start continues to grow the congestion window, reaching eight packets in another RTT. Only four of these packets can reach the receiver in one RTT, no matter how many are sent. That is, the network pipe is full. Additional packets placed into the network by the sender will build up in router queues, since they cannot be delivered to the receiver quickly enough. Congestion and packet loss will occur soon. To keep slow start under control, the sender keeps a threshold for the connection called the **slow start threshold**.



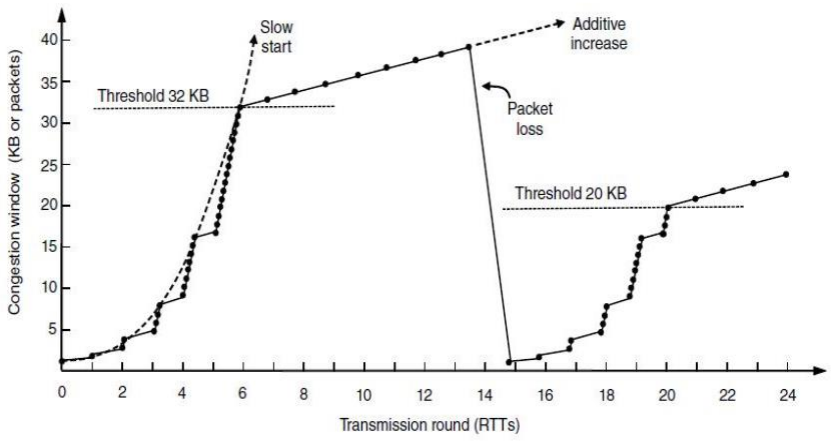
Slow start from an initial congestion window of 1 segment

Additive increase is shown in Fig. For the same situation as slow start. At the end of every RTT, the sender's congestion window has grown enough that it can inject an additional packet into the network. Compared to slow start, the linear rate of growth is much slower. It makes little
difference for small congestion windows, as is the case here, but a large difference in the time taken to grow the congestion window to 100 segments



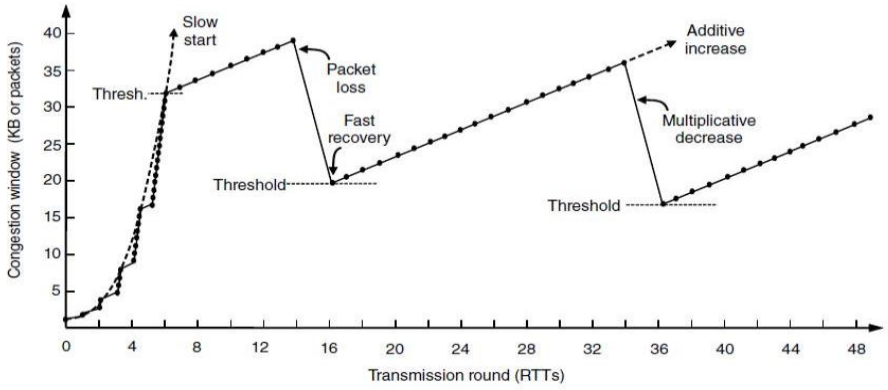
Additive increase from an initial congestion window of 1 segment

There is a quick way for the sender to recognize that one of its packets has been lost. As packets beyond the lost packet arrive at the receiver, they trigger acknowledgements that return to the sender. These acknowledgements bear the same acknowledgement number. They are called **duplicate acknowledgements**. Each time the sender receives a duplicate acknowledgement, it is likely that another packet has arrived at the receiver and the lost packet still has not shown up. Because packets can take different paths through the network, they can arrive out of order. This will trigger duplicate acknowledgements even though no packets have been lost. However, this is uncommon in the Internet much of the time. When there is reordering across multiple paths, the received packets are usually not reordered too much. Thus, TCP somewhat arbitrarily assumes that three duplicate acknowledgements imply that a packet has been lost. The identity of the lost packet can be inferred from the acknowledgement number as well. It is the very next packet in sequence. This packet can then be retransmitted right away, before the retransmission timeout fires. This heuristic is called **fast retransmission**.



Slow start followed by additive increase in TCP Tahoe.

Fast recovery is the heuristic that implements this behavior. It is a temporary mode that aims to maintain the ack clock running with a congestion window that is the new threshold, or half the value of the congestion window at the time of the fast retransmission. To do this, duplicate acknowledgements are counted (including the three that triggered fast retransmission) until the number of packets in the network has fallen to the new threshold. This takes about half a round-trip time. From then on, a new packet can be sent for each duplicate acknowledgement that is received. One round-trip time after the fast retransmission, the lost packet will have been acknowledged. At that time, the stream of duplicate acknowledgements will cease and fast recovery mode will be exited. The congestion window will beset to the new slow start threshold and grows by linear increase



Fast recovery and the sawtooth pattern of TCP Reno

UNIVERSITY QUESTIONS

2Marks:

1. Define UDP.(Nov 2013)
2. Define Flow Control.(Nov 2013)
3. Different types of Multiplexing(April 2014)
4. Explain Connection establishment in TCP.(Nov 2015)
5. Define Sockets.(Nov 2015)

11Marks:

1. Explain Various Timer used in TCP.(Nov 2013)
2. Explain TCP Time management(Nov 2013)
3. Explain TCP Congestion control methods. (Nov 2013)
4. Explain transport Layer services.(Nov2015)
5. Explain Multiplexing.(Nov2015)
6. Explain Flow control.(Nov2015)

UNIT-V

Application Layer – DNS – Name space – Resource records – name servers – e-mail - Architecture and Services - The User Agent - Message Formats - Message Transfer - Final Delivery – WWW – Architecture - Static Web Pages - Dynamic Web Pages and Web Applications - HTTP– Network Security - Introduction to Cryptography - Substitution Ciphers - Transposition Ciphers – Public key algorithms – RSA – Authentication Protocols - Authentication Using Kerberos.

2 MARKS

1. **What is Application Layer?**

   An **application layer** is an abstraction **layer** that specifies the shared protocols and interface methods used by hosts in a communications network.

2. **What are the responsibilities of Application Layer?**

   The Application Layer enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as e-mail, shared database management and other types of distributed information services

   - Network virtual Terminal
   - File transfer, access and Management (FTAM)
   - Mail services
   - Directory Services

3. **Write down the Principles of Network Applications.**

   - Electric Mail
   - The Web
   - Instant messaging
   - Login into a remote computer
   - P2P file sharing
   - File transfer between two accounts on two computers
   - Multi-user networked games
   - Streaming stored video clips
   - Internet phone

Real-time video conferencing

4. **Define DNS?**

   - Domain Name Servers (**DNS**) are the Internet's equivalent of a phone book. They maintain a directory of domain names and translate them to Internet Protocol (IP) addresses. This is necessary because, although domain names are easy for people to remember, computers or machines, access websites based on IP addresses.
   - The naming system on which DNS is based is a hierarchical and logical tree structure called the domain namespace . This List of **DNS record** types provides an overview of types of **resource records** (database **records**) stored in the zone files of the **Domain Name System** (**DNS**).

5. **Define E-Mail?**

   - Electronic mail, most commonly referred to as email or e-mail since c 1993, is a method of exchanging digital messages from an author to one or more recipients. Modern email operates across the Internet or other computer networks.
   - Short for electronic mail, email (or e-mail) is defined as the transmission of messages over communications networks. Typically the messages are notes entered from the keyboard or electronic files stored on disk. Most mainframes, minicomputers, and computer networks have an email system.

6. **What is User Agent?**

Mail user agent(MUA) The program that allows the user to compose and read electronic mail messages. The MUA provides the interface between the user and the Message Transfer Agent

- Plain text This is a format that all email applications support. Plain text messages don't support bold, italic, colored fonts, or other text formatting. .
- Outlook Rich Text format (RTF) This is a Microsoft format that only the following email applications support

7. **What is Message Transfer?**

A message transfer agent receives mail from either another MTA, a mail submission agent (MSA), or a mail user agent (MUA). The transmission details are specified by the Simple Mail Transfer Protocol (SMTP).

8. **What is SMTP?**

Simple Mail Transfer Protocol (**SMTP**) is an Internet standard for electronic mail (e-mail) transmission. First defined by RFC 821 in 1982, it was last updated in 2008 with the Extended **SMTP** additions by RFC 5321 - which is the protocol in widespread use today. **SMTP** by default uses TCP port 25.

9. **What is the function of SMTP?**

- SMTP functions in two ways.

  Firstly, it **verifies the configuration of the computer** from where the email is being sent and grants permission for the process.

  Secondly, it sends out the message and follows the successful delivery of the email. If the email cannot be delivered, it's returned-to-sender or bounces back.

10. **What is Message Delivery Agent?**

A **mail delivery agent** or **message delivery agent** (**MDA**) is a computer software component that is responsible for the delivery of e-mail messages to a local recipient's mailbox. Also called an **LDA**, or **local delivery agent**. In the Internet mail architecture, local message delivery is achieved through a process of handling messages from the message transfer agent, and storing mail into the recipient's environment.

11. **What is the use of Internet Control Message Protocol?**

The **Internet Control Message Protocol** (**ICMP**) is one of the main protocols of the Internet Protocol Suite. It is **used** by network devices, like routers, to send error messages indicating, for example, that a requested service is not available or that a host or router could not be reached.

12. **Define World Wide Web?**

The World Wide Web (**www**, W3) **is** an information system of interlinked hypertext documents that are accessed via the Internet and built on top of the Domain Name System. It has also commonly become known simply as the Web.

13. **Define Static Web Page?**

A **static web page** (sometimes called a flat **page**/stationary **page**) is a **web page** that is delivered to the user exactly as stored, in contrast to dynamic **web pages** which are generated by a **web** application.

14. **What is Dynamic Web Page?**

A server-side **dynamic web page** is a **web page** whose construction is controlled by an application server processing server-side scripts. In server-side scripting, parameters determine how the assembly of every new **web page** proceeds, including the setting up of more client-side processing.

15. **What is a CGI?**

Common Gateway Interface (**CGI**) is a standard method used to generate dynamic content on Web pages and Web applications. **CGI**, when implemented on a Web server, provides an interface between the Web server and programs that generate the Web content.

16. Define Web Application?

A **web application** or **web** app is any computer program that runs in a **web** browser. It is created in a browser-supported programming language (such as the combination of JavaScript, HTML and CSS) and relies on a **web** browser to render the **application.**

17. **Define HTTP?**

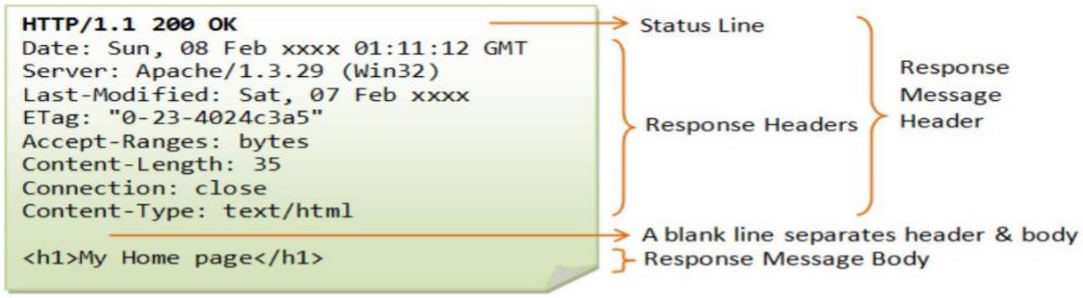
The Hypertext Transfer Protocol (**HTTP**) is an application protocol for distributed, collaborative, hypermedia information systems. **HTTP** is the foundation of data communication for the World Wide Web. Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text.

18. **Why HTTP is said to be stateless protocol?**

Because a **stateless protocol** does not require the server to retain session information or status about each communications partner for the duration of multiple requests. **HTTP** is a **stateless protocol**, which means that the connection between the browser and the server is lost once the transaction ends.

19. Give the format of HTTP response message?

Each request message sent by an HTTP client to a server prompts the server to send back a response message.



20. **What is meant by Network Security?**

Network security consists of the provisions and policies adopted by a **network** administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer **network** and **network**-accessible resources.

21. Who are the people who cause security problem?

- Outside people and hackers
- The people who work for your company
- The applications that your users use to perform their business tasks
- The operating systems that run on your users' desktops and your servers, as well as the equipment employed
- The network infrastructure used to move data across your network, including devices such as routers, switches, hubs, firewalls, gateways, and other devices

22. **Define Cryptography?**

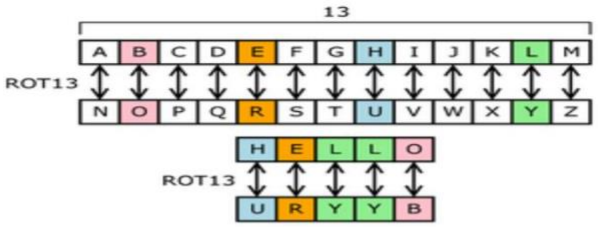
Cryptography is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). Cryptography is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. Cryptography is closely related to the disciplines of cryptology and cryptanalysis.

23. **What is Cipher Text?**

Ciphertext is encrypted text. Plaintext is what you have before encryption, and ciphertext is the encrypted result. The term cipher is sometimes used as a synonym for ciphertext, but it more properly means the method of encryption rather than the result.

24. **Define Substitution cipher?**

In cryptography, a **substitution cipher** is a method of encoding by which units of plaintext are replaced with **ciphertext**, according to a regular system; the "units" may be single letters (the most common), pairs of letters, triplets of letters, mixtures of the above, and so forth

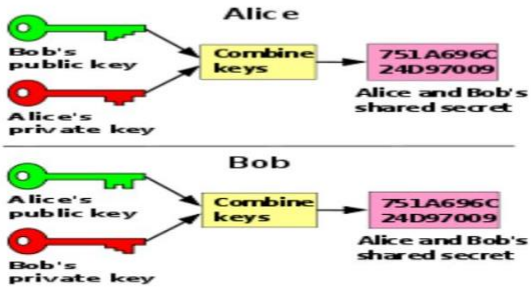


25. **Define Transposition Cipher?**

In cryptography, a **transposition cipher** is a method of encryption by which the positions held by units of plaintext (which are commonly characters or groups of characters) are shifted according to a regular system, so that the **ciphertext** constitutes a permutation of the plaintext.

26. **Define Public Key Cryptography?**

Public-key cryptography, also known as asymmetric cryptography, is a class of cryptographic protocols based on **algorithms** that require two separate **keys**, one of which is secret (or private) and one of which is **public**.



27. **Define RSA?**

RSA is one of the first practical public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and differs from the decryption key which is kept secret.

28. **What is Authentication Protocol?**

An **authentication protocol** is a type of cryptographic **protocol** with the purpose of authenticating entities wishing to communicate securely. There are different authentication protocols such as: AKA. CAVE-based_authentication. Challenge-handshake authentication protocol (CHAP).

29. **Define Kerberos?**

Kerberos is a secure method for authenticating a request for a service in a computer network. **Kerberos** was developed in the Athena Project at the Massachusetts Institute of Technology (MIT). The name is taken from Greek mythology; **Kerberos** was a three-headed dog who guarded the gates of Hades

11 MARKS

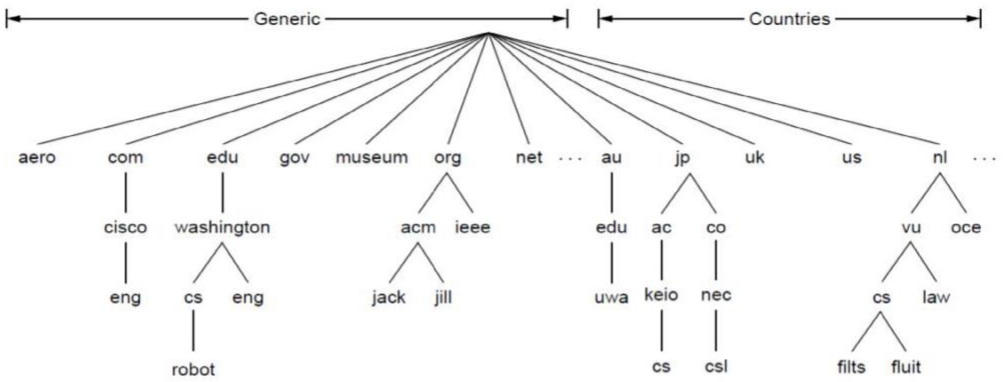
1. Discuss the services provided by the Internet's domain name system (DNS).

Many millions of PCs were connected to the Internet, everyone involved with it realized that this approach could not continue to work forever. To solve these problems, **DNS** (**Domain Name System**) was invented in 1983. It is primarily used for mapping host names to IP addresses but can also be used for other purposes. DNS is defined in RFCs 1034, 1035, 2181, and further elaborated in many others.

- **The DNS name space**
- **Domain Resource records**
- **Name servers**

The DNS Name Space

- A hostname consists of the computer name followed by the domain name
- csc.villanova.edu is the domain name
  - A domain name is separated into two or more sections that specify the organization, and possibly a subset of an organization, of which the computer is a part
  - Two organizations can have a computer named the same thing because the domain name makes it clear which one is being referred to
- The very last section of the domain is called its top-level domain (TLD) name
- Organizations based in countries other than the United States use a top-level domain that corresponds to their two-letter country codes
- The domain name system (DNS) is chiefly used to translate hostnames into numeric IP addresses
  - DNS is an example of a distributed database
  - If that server can resolve the hostname, it does so
  - If not, that server asks another domain name server



A portion of the Internet domain name space.

Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No
aero	Air transport	2001	Yes

Generic top-level domains

Domain Resource Records

Every domain whether it is a single host or a top level domain can have a set of resource records associated with it. Whenever a resolver (this will be explained later) gives the domain name to DNS it gets the resource record associated with it. So DNS can be looked upon as a service which maps domain names to resource records. Each resource record has five fields and looks as below:

Domain Name	Class	Type	Time to Live	Value

- Domain name: the domain to which this record applies.
- Class: set to IN for internet information. For other information other codes may be specified.
- Type: tells what kind of record it is.
- Time to live: Upper Limit on the time to reach the destination
- Value: can be an IP address, a string or a number depending on the record type.

A **Resource Record** (RR) has the following:

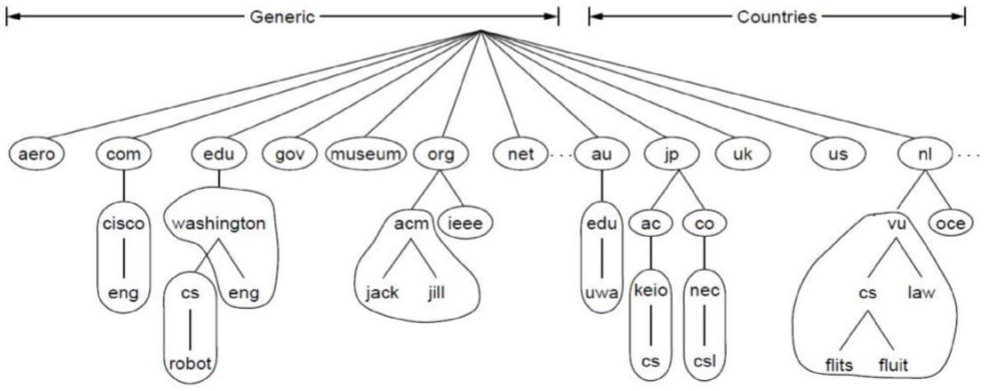
- **owner** which is the domain name where the RR is found.
- **type** which is an encoded 16 bit value that specifies the type of the resource in this resource record. It can be one of the following:

- **A** a host address
- **CNAME** identifies the canonical name of an alias
- **HINFO** identifies the CPU and OS used by a host
- **MX** identifies a mail exchange for the domain.
- **NS** the authoritative name server for the domain
- **PTR** a pointer to another part of the domain name space
- **SOA** identifies the start of a zone of authority class which is an encoded 16 bit value which identifies a protocol family or instance of a protocol.

- **class** One of: **IN** the Internet system or **CH** the Chaos system
- **TTL** which is the time to live of the RR. This field is a 32 bit integer in units of seconds, an is primarily used by resolvers when they cache RRs. The TTL describes how long a RR can be cached before it should be discarded.
- **RDATA** Data in this field depends on the values of the type and class of the RR and a description for each is as follows:
  - for A: For the IN class, a 32 bit IP address For the CH class, a domain name followed by a 16 bit octal Chaos address.
  - for CNAME: a domain name.
  - for MX: a 16 bit preference value (lower is better) followed by a host name willing to act as a mail exchange for the owner domain.
  - for NS: a host name.
  - for PTR: a domain name.
  - for SOA: several fields.

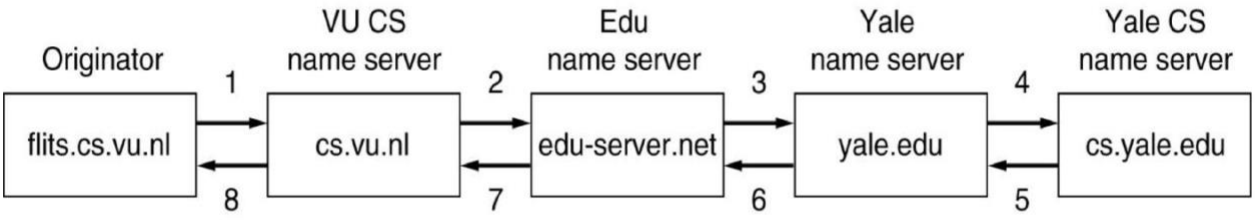
Name Servers

- DNS database is partitioned into zones.
- Each zone contains part of the DNS tree.
- Name servers store information about the name space in units called “zones”
- Zone <-> name server.
  - Each zone may be served by more than 1 server.
  - A server may serve multiple zones.
- Primary and secondary name servers.



Part of the DNS name space divided into zones (which are circled)

- Application wants to resolve name.
- Resolver sends query to local name server.
  - Resolver configured with list of local name servers.
  - Select servers in round-robin fashion.
- If name is local, local name server returns matching authoritative RRs.
  - Authoritative RR comes from authority managing the RR and is always correct.
  - Cached RRs may be out of date.
- If information not available locally (not even cached), local NS will have to ask someone else.
  - It asks the server of the top-level domain of the name requested.
- Recursive query:
  - Each server that doesn‟t have info forwards it to someone else.
  - Response finds its way back.
- Alternative:
  - Name server not able to resolve query, sends back the name of the next server to try.
  - Some servers use this method.
  - More control for clients.
- Suppose resolver on flits.cs.vu.nl wants to resolve linda.cs.yale.edu.
  - Local NS, cs.vu.nl, gets queried but cannot resolve it.
  - It then contacts .edu server.
  - .edu server forwards query to yale.edu server.
  - yale.edu contacts cs.yale.edu, which has the authoritative RR.
  - Response finds its way back to originator.
  - cs.vu.nl caches this info.
    - Not authoritative (since may be out-of-date).
    - RR TTL determines how long RR should be cached.



How a resolver looks up a remote name in eight steps.

2. **Explain the basic function of Electronic Mail (e-mail) system.**

- Many user applications use client-server architecture
- Electronic mail client accepts mail from user and delivers to server on destination computer
- Many variations and styles of delivery

Electronic mail paradigm

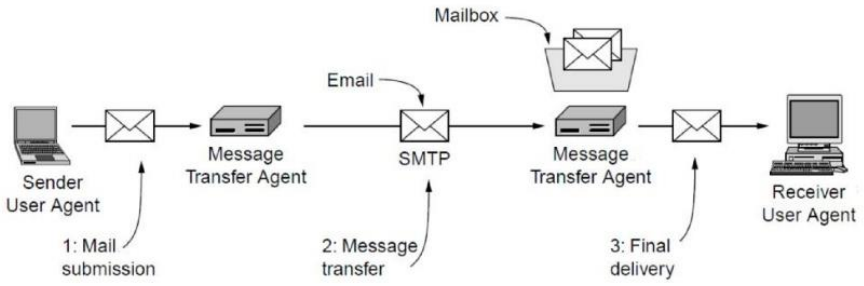
- Electronic version of paper-based office memo
    - Quick, low-overhead written communication
    - Dates back to time-sharing systems in 1960s
- Because e-mail is encoded in an electronic medium, new forms of interaction are possible
    - Fast
    - Automatic processing - sorting, reply
    - Can carry other content

- **Architecture and services**
- **The user agent**
- **Message formats**
- **Message transfer**
- **Final delivery**

Architecture and Services

It consists of two kinds of subsystems:

- The **user agents**, which allow people to read and send email.
- The **message transfer agents**, which move the messages from the source to the destination. Refer to message transfer agents informally as **mail servers**.
- The user agent is a program that provides a graphical interface, or sometimes a text- and command based interface that lets users interact with the email system.
- The act of sending new messages into the mail system for delivery is called **mail submission**.
- Their job is to automatically move email through the system from the originator to the recipient with **SMTP** (**Simple Mail Transfer Protocol**).
- **Mailboxes** store the email that is received for a user. They are maintained by mail servers. User agents simply present users with a view of the contents of their mailboxes.
- A key idea in the message format is the distinction between the **envelope** and its contents.
- The envelope encapsulates the message. It contains all the information needed for transporting the message, such as the destination address, priority, and security level, all of which are distinct from the message itself.
- The message inside the envelope consists of two separate parts: the **header** and the **body**. The header contains control information for the user agents. The body is entirely for the human recipient.



Architecture of the email system

The User Agent

- A user agent is a program (sometimes called an **email reader**) that accepts a variety of commands for composing, receiving, and replying to messages, as well as for manipulating mailboxes.
- There are many popular user agents, including Google gmail, Microsoft Outlook, Mozilla Thunderbird, and Apple Mail. They can vary greatly in their appearance.
- Most user agents have a menu- or icon driven graphical interface that requires a mouse, or a touch interface on smaller mobile devices.

Message Formats

- Lines of text in format keyword: information
- keyword identifies information; information can appear in any order
- Essential information:
  - To: list of recipients
  - From: sender
  - Cc: list of copy recipients
- Useful information:
  - Reply-to: different address than From:
  - Received-by: for debugging
- Frivolous information:
  - Favorite-drink: lemonade
  - Phase-of-the-moon: gibbous

E-mail example

```
From: John_Q_Public@foobar.com
To: 912743.253843@nonexist.com
Date: Wed, 4 Sep 96 10:21:32 EDT
Subject: lunch with me?

Bob,

   Can we get together for lunch when you visit next
week?  I'm free on Tuesday or Wednesday -- just let me
know which day you would prefer.

John
```

E-mail headers

- Mail software passes unknown headers unchanged
- Some software may interpret vendor-specific information

Keyword	Meaning
From	Sender's address
To	Recipients' addresses
Cc	Addresses for carbon copies
Date	Date on which message was sent
Subject	Topic of the message
Reply-To	Address to which reply should go
X-Charset	Character set used (usually ASCII)
X-Mailer	Mail software used to send the message
X-Sender	Duplicate of sender's address
X-Face	Encoded image of the sender's face

Data in e-mail

- Original Internet mail carried only 7-bit ASCII data
- Couldn't contain arbitrary binary values; e.g., executable program
- Techniques for encoding binary data allowed transport of binary data
- uuencode: 3 8-bit binary values as 4 ASCII characters (6 bits each)
    - Also carries file name and protection information
    - Incurs 33% overhead
    - Requires manual intervention

MIME (Multipurpose Internet Mail Extensions)

- Extends and automates encoding mechanisms - Multipart Internet Mail Extensions
- Allows inclusion of separate components - programs, pictures, audio clips - in a single mail message

- Sending program identifies the components so receiving program can automatically extract and inform mail recipient
    - Header includes:

MIME-Version: 1.0
Content-Type: Multipart/Mixed; Boundary=Mime_separator

    - Separator line gives information about specific encoding
    - Plain text includes:

Content-type: text/plain

- MIME is extensible - sender and receiver agree on encoding scheme

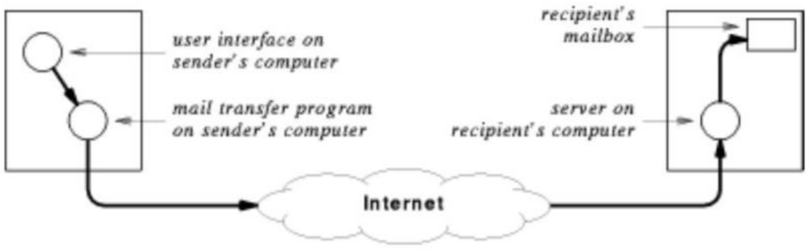
- MIME is compatible with existing mail systems
    - Everything encoded as ASCII
    - Headers and separators ignored by non-MIME mail systems
- MIME encapsulates binary data in ASCII mail envelope

Programs as mail recipients

- Can arrange for e-mailbox to be associated with a program rather than a user's mail reader
- Incoming mail automagically processed as input to program
- Example - mailing list subscription administration
- Can be used to implement client-server processing
    - Client request in incoming mail message
    - Server response in returned mail reply

Message Transfer

- E-mail communication is really a two-part process:
    - User composes mail with an e-mail interface program
    - Mail transfer program delivers mail to destination
        - Waits for mail to be placed in outgoing message queues
        - Picks up message and determines recipient(s)
        - Becomes client and contacts server on recipient's computer
        - Passes message to server for delivery



SMTP

- Simple Mail Transfer Protocol (SMTP) is standard application protocol for delivery of mail from source to destination
- Provides reliable delivery of messages
- Uses TCP and message exchange between client and server
- Other functions:
    - E-mail address lookup
    - E-mail address verification

Multiple recipients on one computer

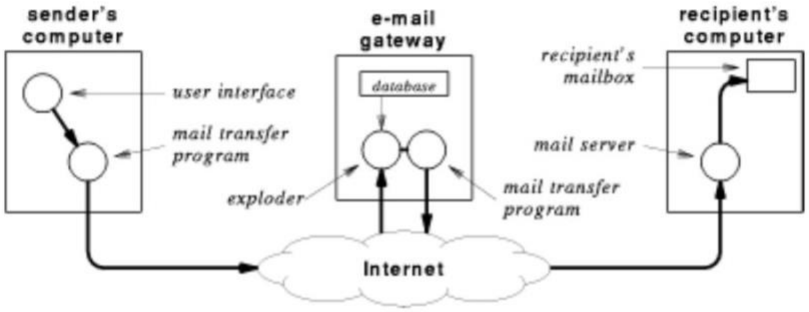
- E-mail addresses can be attached to programs as well as electronic mailboxes
- Mail exploder or mail forwarder resends copies of message to e-mail addresses in mailing list
    - UNIX mail program sendmail provides simple mail aliases
    - Mailing list processor, e.g., listserv, can also interpret subscription management commands

List	Contents
friends	Joe@foobar.com, Jill@bar.gov, Tim@StateU.edu, Mary@acollege.edu, Hank@nonexist.com,
customers	george@xyz.com, VP_Marketing@news.com
bball-interest	hank@nonexist.com, Linda_S_Smith@there.com, John_Q_Public@foobar.com, Connie@foo.edu,

Mail gateways

- Mailing list processing may take significant resources in large organization
- May be segregated to a dedicated server computer: mail gateway
    - Provides single mail destination point for all incoming mail
    - e.g., bucknell.edu

- Can use MX records in DNS to cause all mail to be delivered to gateway



Mail gateways and forwarding

- Users within an organization may want to read mail on local or departmental computer
- Can arrange to have mail forwarded from mail gateway
- Message now makes multiple hops for delivery
- Hops may be recorded in header
- Forwarded mail may use proprietary (non-SMTP) mail system

Mail gateways and e-mail addresses

- Organization may want to use uniform naming for external mail
- Internally, may be delivered to many different systems with different naming conventions
- Mail gateways can translate e-mail addresses

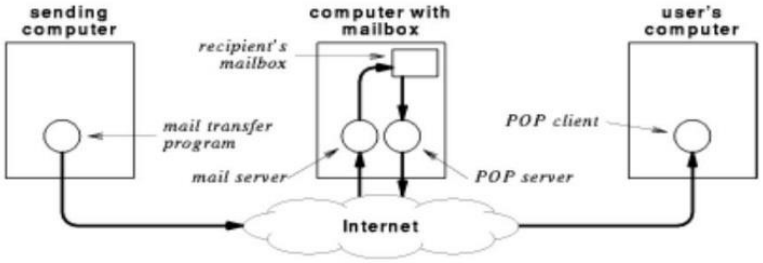
Ralph_Droms droms@regulus.eg.bucknell.edu
Dan_Little dlittle@mail.bucknell.edu
Ruth_Miller miller@charcoal.eg.bucknell.edu

Mailbox access

- Where should mailbox be located?
- Users want to access mail from most commonly used computer
- Can't always use desktop computer as mail server
  - Not always running
  - Requires multitasking operating system
  - Requires local disk storage
- Can TELNET to remote comptuer with mail server

Internet Mail access protocols

- Instead of TELNET, use prtocl that accesses mail on remote computer directly
- TCP/IP protocol suite includes Post Office Protocol (POP) for remote mailbox access
  - Computer with mailboxes runs POP server
  - User runs POP client on local computer
  - POP client can access and retrieve messages from mailbox
  - Requires authentication (password)
  - Local computer uses SMTP for outgoing mail



Final Delivery

- Our mail message is almost delivered. It has arrived at Bob‟s mailbox.
- All that remains is to transfer a copy of the message to Bob‟s user agent for display.
- When the user agent and mail transfer agent ran on the same machine as different processes.
- The mail transfer agent simply wrote new messages to the end of the mailbox file, and the user agent simply checked the mailbox file for new mail.

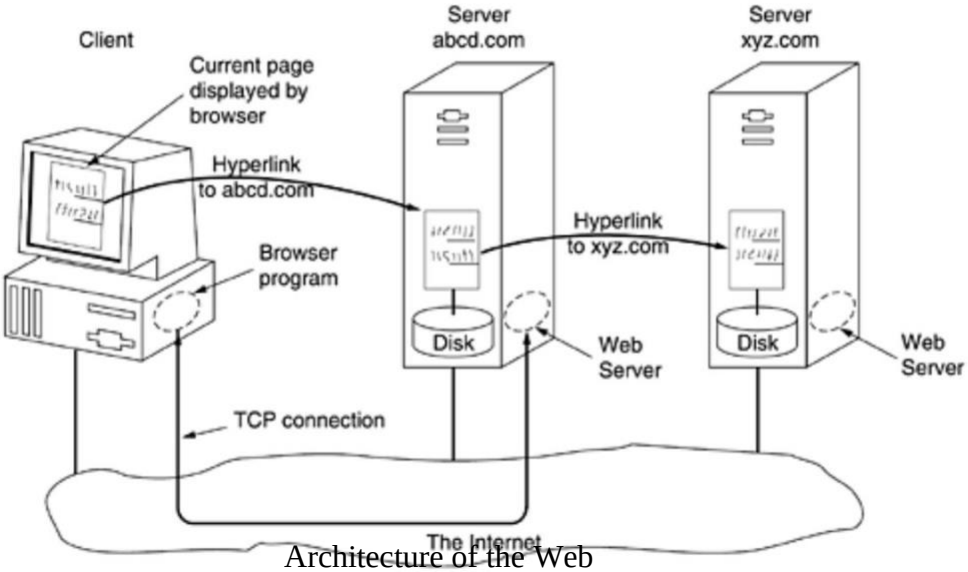
3. Explain in detail about World Wide Web?

- **Architectural overview**
- **Static web pages**
- **Dynamic web pages, web applications**
- **The hypertext transfer protocol**

Architectural Overview

- The Web consists of a vast, worldwide collection of content in the form of **Web pages**, often just called **pages** for short.
- Each page may contain links to other pages anywhere in the world. Users can follow a link by clicking on it, which then takes them to the page pointed to.

- This process can be repeated indefinitely. The idea of having one page point to another, now called **hypertext**, generally viewed with a program called a **browser**.
- This page shows text and graphical elements (that are mostly too small to read).
- A piece of text, icon, image, and so on associated with another page is called a **hyperlink**.



Architecture of the Web

- Here the browser is displaying a Web page on the client machine.
- When the user clicks on a line of text that is linked to a page on the abcd.com server, the browser follows the hyperlink by sending a message to the abcd.com server asking it for the page.
- When the page arrives, it is displayed. If this page contains a hyperlink to a page on the xyz.com server that is clicked on, the browser then sends a request to that machine for the page.

The Client side

When an item is selected, the browser follows the hyperlink and fetches the page selected. Therefore, the embedded hyperlink needs a way to name any other page on the Web. Pages are named using **URLs** (**Uniform Resource Locators**).

URLs have three parts:

This URL consists of three parts: the protocol (http), the DNS name of the host (www.cs.washington.edu), and the path name (index.html).

- The protocol (also known as the **scheme**),
- The DNS name of the machine on which the page is located, and the path uniquely indicating the specific page (a file to read or program to run on the machine).
- In the general case, the path has a hierarchical name that models a file directory structure.

As an example, the URL http://www.cs.washington.edu/index.html

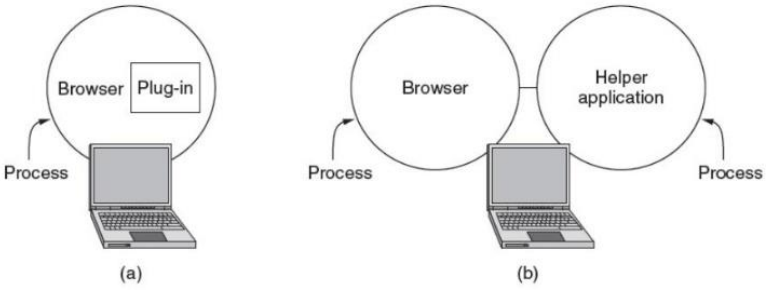
The steps that occur at the client side are:

- The browser determines the URL
- The browser asks DNS for the IP address
- DNS replies with the IP address
- The browser makes a TCP connection to port 80 on the IP address
- It sends a request asking for file
- The site server sends the file
- The TCP connection is released.
- The browser fetches and displays all the text and images in the file.
- Web pages are written in standard HTML language to make it understandable by all browsers.

MIME TYPES

- A **plug-in** is a third-party code module that is installed as an extension to the browser, as illustrated
- A **plugin** is a piece of software that acts as an add-on to a web **browser** and gives the **browser** additional functionality. Plugins can allow a web **browser** to display additional content it was not originally designed to display.

- A **helper application** is an external viewer program launched to display content retrieved using a web browser. Some examples include JPEGview, Windows Media Player, QuickTime Player Real Player and Adobe Reader.



(a)A browser plug-in. (b) A helper application.

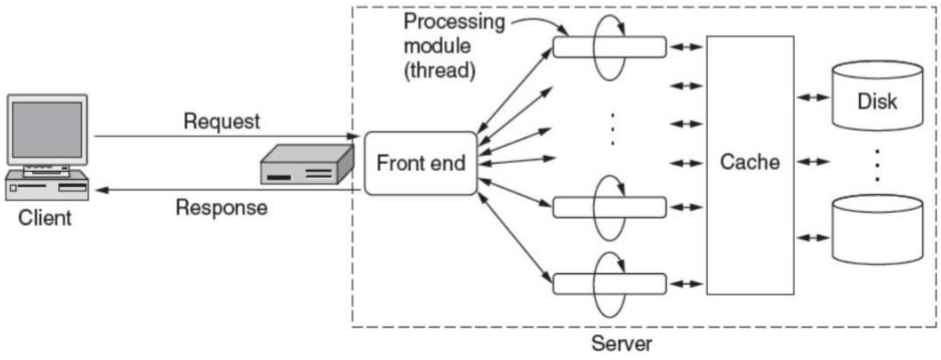
The Server side

The server is given the name of a file to look up and return via the network. In both cases, the steps that the server performs in its main loop are:

- Accept a TCP connection from a client (a browser).
- Get the path to the page, which is the name of the file requested.
- Get the file (from disk).
- Send the contents of the file to the client.
- Release the TCP connection.

Many servers each processing module performs a series of steps. The front end passes each incoming request to the first available module, which then carries it out using some subset of the following steps.These steps occur after the TCP connection and any secure transport mechanism (such as SSL/TLS,) have been established.

- Resolve the name of the Web page requested.
- Perform access control on the Web page.
- Check the cache.
- Fetch the requested page from disk or run a program to build it.
- Determine the rest of the response (e.g., the MIME type to send).
- Return the response to the client.
- Make an entry in the server log.



A multithreaded Web server with a front end and processing modules

Cookies

- Cookies are usually small text files, given ID tags that are stored on your computer's browser directory or program data subfolders.
- Cookies are created when you use your browser to visit a website that uses cookies to keep track of your movements within the site, help you resume where you left off, remember your registered login, theme selection, preferences, and other customization functions.

There are two types of cookies: session cookies and persistent cookies.

- **Session cookies** are created temporarily in your browser's subfolder while you are visiting a website. Once you leave the site, the session cookie is deleted.
- On the other hand, **persistent cookie** files remain in your browser's subfolder and are activated again once you visit the website that created that particular cookie. A persistent cookie remains in the browser's subfolder for the duration period set within the cookie's file.

Domain	Path	Content	Expires	Secure
toms-casino.com	/	CustomerID=297793521	15-10-10 17:00	Yes
jills-store.com	/	Cart=1-00501;1-07031;2-13721	11-1-11 14:22	No
aportal.com	/	Prefs=Stk:CSCO+ORCL;Spt:Jets	31-12-20 23:59	No
sneaky.com	/	UserID=4627239101	31-12-19 23:59	No

Some examples of cookies

HTML-Hypertext Markup Language

HTML is a **markup** language for **describing** web documents (web pages).

- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- A markup language is a set of **markup tags**
- HTML documents are described by **HTML tags**
- Each HTML tag **describes** different document content

A Small HTML Document

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Example Explained

- The **DOCTYPE** declaration defines the document type to be HTML
- The text between **<html>** and **</html>** describes an HTML document
- The text between **<head>** and **</head>** provides information about the document
- The text between **<title>** and **</title>** provides a title for the document
- The text between **<body>** and **</body>** describes the visible page content
- The text between **<h1>** and **</h1>** describes a heading
- The text between ** ** and ** ** describes a paragraph

HTML Tags

HTML tags are **keywords** (tag names) surrounded by **angle brackets**: <tagname>content</tagname>

- HTML tags normally come **in pairs** like and
- The first tag in a pair is the **start tag,** the second tag is the **end tag**

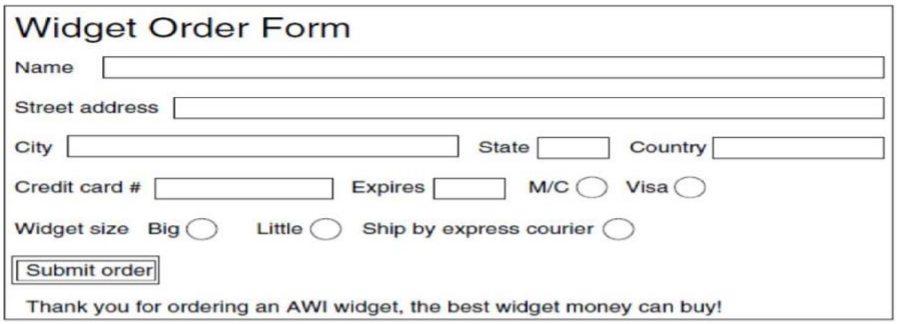
• The end tag is written like the start tag, but with a **slash** before the tag name The start tag is often called the **opening tag**. The end tag is often called the **closing tag**.

Web Browsers

The purpose of a web browser (Chrome, IE, Firefox, Safari) is to read HTML documents and display them. The browser does not display the HTML tags, but uses them to determine how to display the document:

Input And Forms

Input form is an online form which ActionApps users use to manually add data into a slice.. Any input forms a collection (or sequence) of input elements, which correspond to slice Fields. There are various types of input elements, like textarea, select box, simple text box.



The formatted page

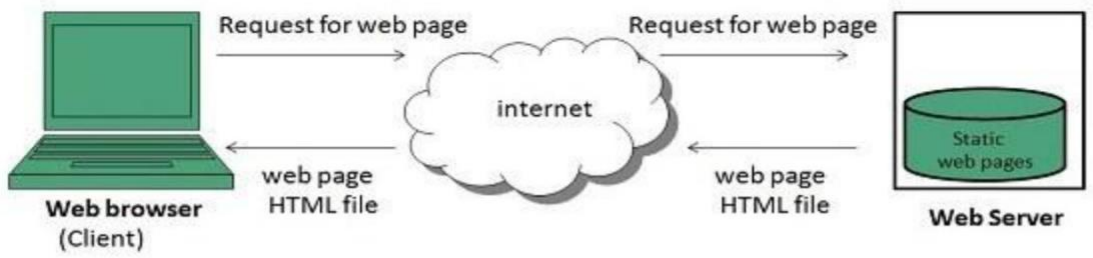
Cascading Style Sheets (**CSS**) is a style sheet language used for describing the look and formatting of a document written in a markup language.

Web Page

- **web page** is a document available on world wide web. Web Pages are stored on web server and can be viewed using a web browser.
- A web page can cotain huge information including text, graphics, audio, video and hyper links. These hyper links are the link to other web pages.
- Collection of linked web pages on a web server is known as **website.** There is unique **Uniform Resource Locator (URL)** is associated with each web page.

Static Web page

- **Static web pages** are also known as flat or stationary web page. They are loaded on the client‟s browser as exactly they are stored on the web server. Such web pages contain only static information. User can only read the information but can‟t do any modification or interact with the information.
- Static web pages are created using only HTML. Static web pages are only used when the information is no more required to be modified.



Dynamic Web page

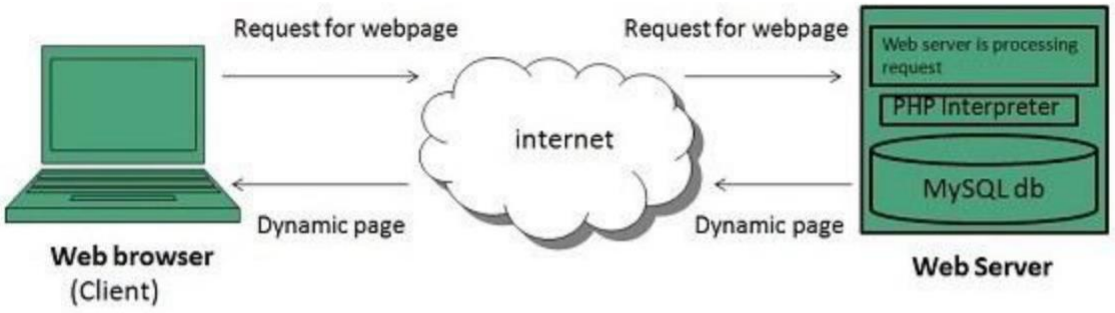
Dynamic web page shows different information at different point of time. It is possible to change a portaion of a web page without loading the entire web page. It has been made possible using **Ajax** technology.

Server-side dynamic web page

It is created by using server-side scripting. There are server-side scripting parameters that determine how to assemble a new web page which also include setting up of more client-side processing.

Client-side dynamic web page

It is processed using client side scripting such as JavaScript. And then passed in to **Document Object Model (DOM).**



4. Explain about Hyper Text Transfer Protocol (HTTP).

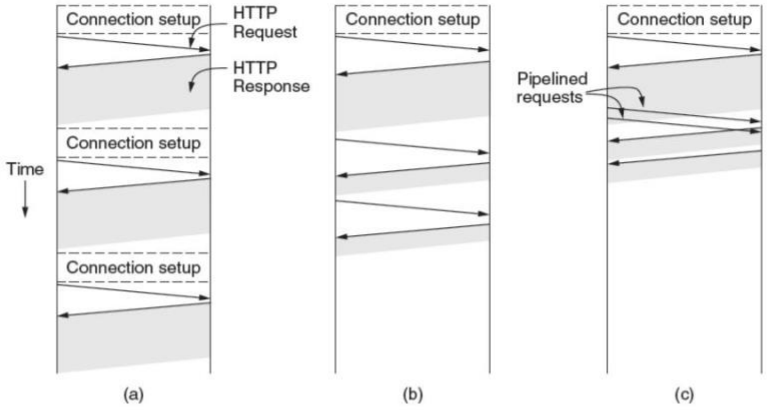
- The Hypertext Transfer Protocol (**HTTP**) is an application protocol for distributed, collaborative, hypermedia information systems. **HTTP** is the foundation of data communication for the World Wide Web.

- HTTP is based on the client-server architecture model and a stateless request/response protocol that operates by exchanging messages across a reliable TCP/IP connection.

- HTTP makes use of the Uniform Resource Identifier (URI) to identify a given resource and to establish a connection.
- Once the connection is established, **HTTP messages** are passed in a format similar to that used by the Internet mail [RFC5322] and the Multipurpose Internet Mail Extensions (MIME) [RFC2045].
- These messages include **requests** from client to server and **responses** from server to client which will have the following format:
- HTTP-message = <Request> | <Response> ; HTTP/1.1 messages

Connections

- Let us consider a Web page with two embedded images on the same server. The URLs of the images are determined as the main page is fetched, so they are fetched after the main page.
- The page is fetched with a persistent connection. That is, the TCP connection is opened at the beginning, then the same three requests are sent, one after the other as before, and only then is the connection closed.
- There is one persistent connection and the requests are pipelined. Specifically, the second and third requests are sent in rapid succession as soon as enough of the main page has been retrieved to identify that the images must be fetched.
- This method cuts down the time that the server is idle, so it further improves performance.



HTTP with (a) multiple connections and sequential requests, (b) A persistent connection and sequential requests, (c) A persistent connection and pipelined requests.

Methods

HTTP - Requests

An HTTP client sends an HTTP request to a server in the form of a request message which includes following format:

- A Request-line
- Zero or more header (General|Request|Entity) fields followed by CRLF
- An empty line (i.e., a line with nothing preceding the CRLF)
- indicating the end of the header fields
- Optionally a message-body

Request-Line

The Request-Line begins with a method token, followed by the Request-URI and the protocol version, and ending with CRLF. The elements are separated by space SP characters.

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

Request Method

The request **method** indicates the method to be performed on the resource identified by the given **Request-URI**. The method is case-sensitive and should always be mentioned in uppercase. The following table lists all the supported methods in HTTP/1.1.

GET

- The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.

HEAD

- Same as GET, but transfers the status line and header section only.

POST

- A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.

PUT

- Replaces all current representations of the target resource with the uploaded content.

DELETE

- Removes all current representations of the target resource given by a URI.

CONNECT

- Establishes a tunnel to the server identified by a given URI.

OPTIONS

- Describes the communication options for the target resource.

TRACE

- Performs a message loop-back test along the path to the target resource.

HTTP - Status Codes

The Status-Code element in a server response, is a 3-digit integer where the first digit of the Status-Code defines the class of response and the last two digits do not have any categorization role. There are 5 values for the first digit:

Code and Description

1xx: Informational:It means the request has been received and the process is continuing.
2xx: Success:It means the action was successfully received, understood, and accepted. **3xx: Redirection**:It means further action must be taken in order to complete the request. **4xx: Client Error**:It means the request contains incorrect syntax or cannot be fulfilled. **5xx: Server Error**:It means the server failed to fulfill an apparently valid request.

Message Types

HTTP messages consist of requests from client to server and responses from server to client.

HTTP-message = Request | Response ; HTTP/1.1 messages

Request and Response messages use the generic message format of RFC 822 [9] for transferring entities (the payload of the message).

Both types of message consist of a start-line, zero or more header fields (also known as "headers"), an empty line (i.e., a line with nothing preceding the CRLF) indicating the end of the header fields, and possibly a message body.

generic-message = start-line
*(message-header CRLF)

CRLF
[message-body]
start-line = Request-Line | Status-Line

Message Headers

HTTP - Header Fields

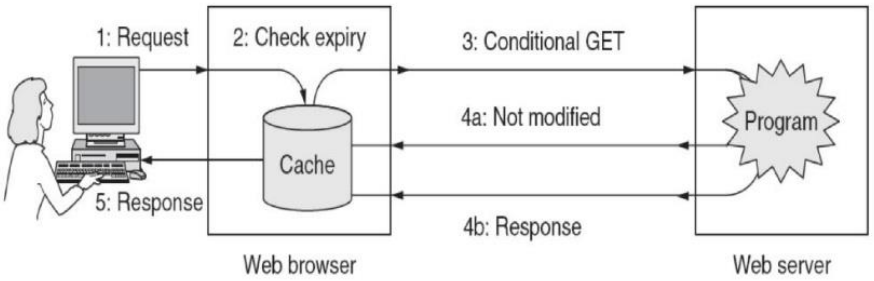
HTTP header fields provide required information about the request or response, or about the object sent in the message body. There are four types of HTTP message headers:

- **General-header:** These header fields have general applicability for both request and response messages.
- **Client Request-header:** These header fields have applicability only for request messages.
- **Server Response-header:** These header fields have applicability only for response messages.
- **Entity-header:** These header fields define meta information about the entity-body or, if no body is present, about the resource identified by the request.

Caching

People often return to Web pages that they have viewed before, and related Web pages often have the same embedded resources. It would be very wasteful to fetch all of these resources for these pages each time they are displayed because the browser already has a copy. Squirreling away pages that are fetched for subsequent use is called **caching**.

- The first strategy is page validation (step 2).
- The cache is consulted, and if it has a copy of a page for the requested URL that is known to be fresh (i.e., still valid), there is no need to fetch it a new from the server.
- It is to ask the server if the cached copy is still valid. This request is a **conditional GET**, and it is shown in Fig(step 3).
- If the server knows that the cached copy is still valid, it can send a short reply to say so (step 4a).
- Otherwise, it must send the full response (step 4b).



HTTP caching.

5. **Write short note on network security.(or)**

Explain in detail about the principles of cryptography?

Cryptography

- **Introduction**
- **Substitution ciphers**
- **Transposition ciphers**

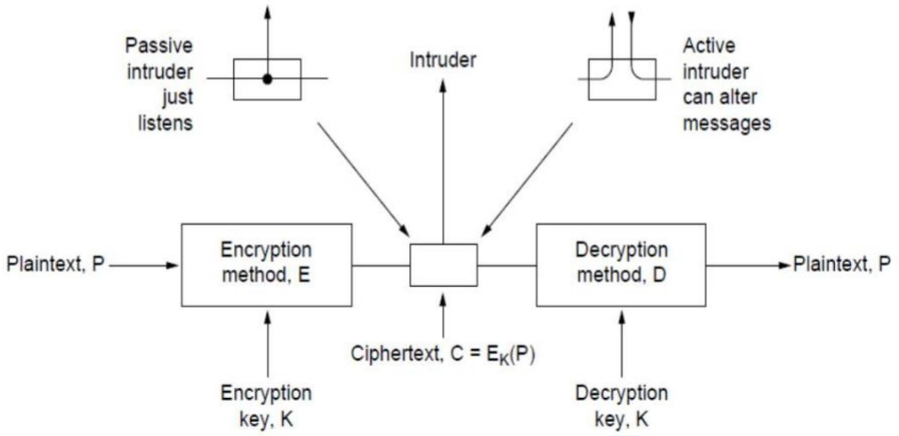
Introduction to Cryptography

- **Cryptography** comes from the Greek words for „„secret writing.""
- A **cipher** is a character-for-character or bit-for-bit transformation, without regard to the linguistic structure of the message. In contrast, a **code** replaces one word with another word or symbol.
- The messages to be encrypted, known as the **plaintext**, are transformed by a function that is parameterized by a **key**.
- The output of the encryption process, known as the **ciphertext**, is then transmitted, often by messenger or radio.
- Sometimes the intruder can not only listen to the communication channel (passive intruder) but can also record messages and play them back later, inject his own messages, or modify legitimate messages before they get to the receiver (active intruder).
- The art of breaking ciphers, known as **cryptanalysis**, and the art of devising them (cryptography) are collectively known as **cryptology**.

It will often be useful to have a notation for relating plaintext, ciphertext, and keys. We will use C =EK(P) to mean that the encryption of the plaintext P using key K gives the ciphertext C. Similarly, P=DK(C) represents the decryption of C to get the plaintext again. It then follows that

$$DK(EK(P)) = P$$

This notation suggests that E and D are just mathematical functions, to distinguish it from the message.



The encryption model (for a symmetric-key cipher)

Substitution Ciphers

- **Substitution Cipher**
  - Changes characters in the plaintext to produce the ciphertext.
  - where letters of plaintext are replaced by other letters or by numbers or symbols
  - or if plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns
- Examples
  - Caesar Cipher
  - Vigenere Cipher
  - One Time Pad
- **Caesar Cipher**
  - Consider the plaintext to be the letters A,B,C,...,Z.
  - Now shift the sequence, say, by 3 to get D,E,F,...Z,A,B,C.
  - Then the cipher text becomes D for A, E for B, and so on.
  - If each letter is represented by integers 0,1,...,25, we can describe this process as C=(M + K) mod 26, where the key is K=3.
  - earliest known substitution cipher
  - by Julius Caesar
  - first attested use in military affairs
  - replaces each letter by 3rd letter on

Example:

meet me after the toga party

PHHW PH DIWHU WKH WRJD SDUWB

- can define transformation as:

  a b c d e f g h i j k l m n o p q r s t u v w x y z

  D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

- mathematically give each letter a number

  a b c d e f g h i j k l m

  0 1 2 3 4 5 6 7 8 9 10 11 12

  n o p q r s t u v w x y Z

  13 14 15 16 17 18 19 20 21 22 23 24 25

- then have Caesar cipher as:

  C = E(p) = (p + k) mod (26)

  p = D(C) = (C – k) mod (26)

Cryptanalysis of Caesar Cipher

- only have 26 possible ciphers
  - A maps to A,B,..Z
- could simply try each in turn
- a **brute force search**
- given ciphertext, just try all shifts of letters
- do need to recognize when have plaintext
- eg. break ciphertext "GCUA VQ DTGCM"

Monoalphabetic Cipher

- rather than just shifting the alphabet
- could shuffle (jumble) the letters arbitrarily
- each plaintext letter maps to a different random ciphertext letter
- hence key is 26 letters long

Plain: abcdefghijklmnopqrstuvwxyz
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN
Plaintext: ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA

Monoalphabetic Cipher Security

- now have a total of 26! = 4 x 1026 keys
- with so many keys, might think is secure
- but would be **!!!WRONG!!!**
- problem is language characteristics

Playfair Cipher

- not even the large number of keys in a monoalphabetic cipher provides security
- one approach to improving security was to encrypt multiple letters
- the **Playfair Cipher** is an example
- invented by Charles Wheatstone in 1854, but named after his friend Baron Playfair

Playfair Key Matrix

- a 5X5 matrix of letters based on a keyword
- fill in letters of keyword (sans duplicates)
- fill rest of matrix with other letters
- eg. using the keyword MONARCHY

MONAR
CHYBD
EFGIK
LPQST
UVWXZ

Polyalphabetic Ciphers

- another approach to improving security is to use multiple cipher alphabets
- called **polyalphabetic substitution ciphers**
- makes cryptanalysis harder with more alphabets to guess and flatter frequency distribution
- use a key to select which alphabet is used for each letter of the message
- use each alphabet in turn
- repeat from start after end of key is reached

Vigenère Cipher

- The Vigenère cipher chooses a sequence of keys, represented by a string.
- Key letters are applied to successive plaintext.
- When the end of the key sequence is reached, the key starts over again.
- The length of the key is called the period of the cipher.
- simplest polyalphabetic substitution cipher is the **Vigenère Cipher**
- effectively multiple caesar ciphers
- key is multiple letters long K = k1 k2 ... kd
- i^{th} letter specifies i^{th} alphabet to use
- use each alphabet in turn
- repeat from start after d letters in message
- decryption simply works in reverse

Example

- write the plaintext out
- write the keyword repeated above it
- use each key letter as a caesar cipher key
- encrypt the corresponding plaintext letter
- eg using keyword deceptive

key: deceptivedeceptivedeceptive

plaintext: wearediscoveredsaveyourself

- ciphertext:ZICVTWQNGRZGVTWAVZHCQYGLMGJ

One-Time Pad

- A variant of the Vigenère cipher.
- The key is chosen at random.
- The length of the key is at least as long as that of the message, and so it does not repeat.
- if a truly random key as long as the message is used, the cipher will be secure
- called a One-Time pad

- is unbreakable since ciphertext bears no statistical relationship to the plaintext
- since for **any plaintext** & **any ciphertext** there exists a key mapping one to other
- can only use the key **once** though
- have problem of safe distribution of key

Transposition Ciphers

- now consider classical **transposition** or **permutation** ciphers
- these hide the message by rearranging the letter order
- without altering the actual letters used
- can recognise these since have the same frequency distribution as the original text

Rail Fence cipher

- write message letters out diagonally over a number of rows
- then read off cipher row by row
- eg. write message out as:

m e m a t r h t g p r

y e t e f e t e o a a t

- giving ciphertext

MEMATRHTGPRYETEFETEOAAT

Row Transposition Ciphers

- a more complex scheme
- write letters of message out in rows over a specified number of columns
- then reorder the columns according to some key before reading off the rows

Key: 3 4 2 1 5 6 7

Plaintext: a t t a c k p

o s t p o n e

d u n t i l t

w o a m x y z

Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ

7. **Discuss briefly about Public-key Algorithms(RSA).**

RSA(Rivest,Shamir,adleman)

RSA Algorithm

- It was developed by Rivest, Shamir and Adleman. This algorithm makes use of an expression with exponentials.
- Plaintext is encrypted in blocks, with each block having a binary value less than some number n.
- That is, the block size must be less than or equal to log2 (n); in practice, the block size is k-bits, where $2k < n < 2k+1$.
- Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C:

$C = Me \bmod n$

$M = Cd \bmod n = (Me \bmod n) \bmod n$

$= (Me)\, d \bmod n$

$= Med \bmod n$

Both the sender and receiver know the value of n. the sender knows the value of e and only the receiver knows the value of d. thus, this is a public key encryption algorithm with a public key of KU = {e, n} and a private key of KR = {d, n}.

For this algorithm to be satisfactory for public key encryption, the following requirements must be met:

It is possible to find values of e, d, n such that $Med = M \bmod n$ for all $M<n$.

- It is relatively easy to calculate Me and Cd for all values of $M<n$.
- It is infeasible to determine d given e and n.

Let us focus on the first requirement. We need to find the relationship of the

$$Med = M \bmod n$$

A corollary to Euler‟s theorem fits the bill: Given two prime numbers p and q and two integers, n and m, such that $n=pq$ and $0<m<n$, and arbitrary integer k, the following relationship holds

$$mk\Phi(n) +1 = mk(p-1)(q-1) +1 = m \bmod n$$

where $\Phi(n)$ – Euler totient function, which is the number of positive integers less than n and relatively prime to n.

we can achieve the desired relationship, if

$$ed = k\Phi(n)+1$$

This is equivalent to saying:

$ed \equiv 1 \bmod \Phi(n)$
$d = e\text{-}1 \bmod \Phi(n)$

That is, e and d are multiplicative inverses mod Φ(n). According to the rule of modular arithmetic, this is true only if d (and therefore e) is relatively prime to Φ(n). Equivalently, gcd(Φ(n), d) = 1. The steps involved in RSA algorithm for generating the key are

- Select two prime numbers, p = 17 and q = 11.
- Calculate n = p*q = 17*11 = 187
- Calculate Φ(n) = (p-1)(q-1) = 16*10 = 160.
- Select e such that e is relatively prime to Φ(n) = 160 and less than Φ(n); we choose e = 7.
- Determine d such that ed ≡ 1 mod Φ(n) and d<160. the correct value is d = 23, because 23*7 = 161 = 1 mod 160.

The RSA algorithm is summarized below.

Key Generation

- Select p, q p ,q both prime p!=q
- Calculate n = p x q
- Calculate Φ (n) = (p -l)(q - 1)
- Select integer e gcd(Φ (n), e) = 1; 1< e< Φ (n)
- Calculate d d= e-1mod Φ (n)

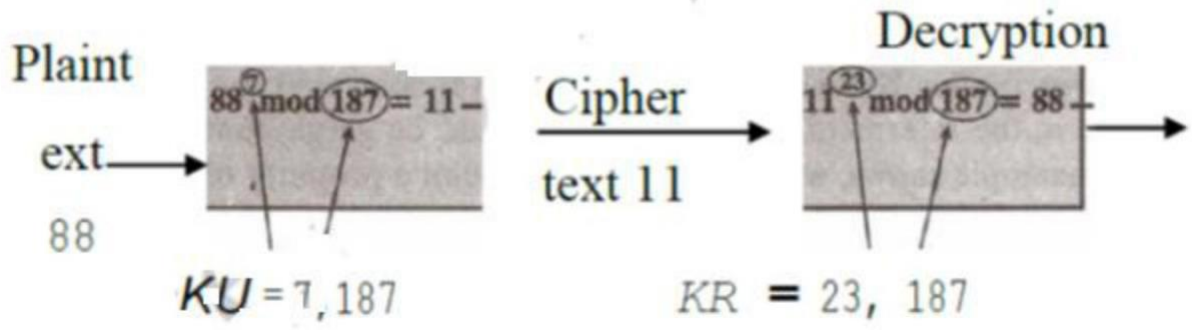
Public key KU = { e,n}
Private key KR = {d,n}

Encryption

Plaintext M < n
Ciphertext C = Me (mod n)

Decryption

Ciphertext C
Plaintext M = Cd (mod n)



Security of RSA

There are three approaches to attack the RSA:

- brute force key search (infeasible given size of numbers)
- mathematical attacks (based on difficulty of computing ø(N), by factoring modulus N)
- timing attacks (on running time of decryption)

8. What is Authentication? How it is different from authorization? Explain in brief about different authentication protocols.

Authentication

Authentication is any process by which a system verifies the identity of a User who wishes to access it. Since Access Control is normally based on the identity of the User who requests access to a resource, **Authentication** is essential to effective Security.

Authorization

Authorization is the process of giving someone permission to do or have something. Authorization or authorisation is the function of specifying access rights to resources related to information security and computer security in general and to access control in particular.

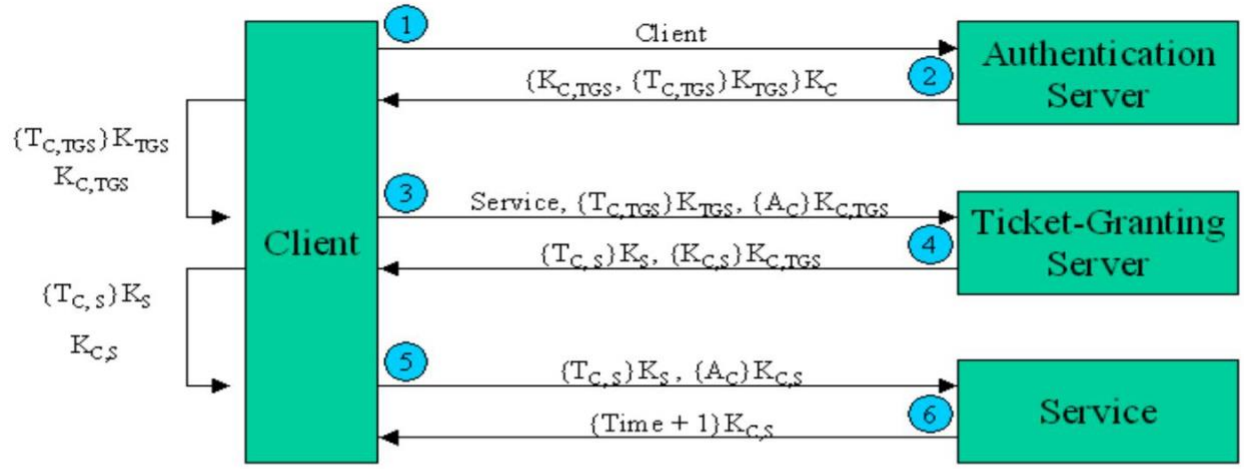
Authentication Protocol-Kerberos

Kerberos was created by Massachusetts Institute of Technology as a solution to many network security problems. It is being used in the MIT campus for reliability. The basic features of Kerberos may be put as:

- It uses symmetric keys.
- Every user has a password (key from it to the Authentication Server)
- Every application server has a password.
- The passwords are kept only in the Kerberos Database.
- The Servers are all physically secure.(No unauthorized user has access to them.)
- The user gives the password only once.
- The password is not sent over the network in plain text or encrypted form.
- The user requires a ticket for each access.

1. Authentication Server (AS): Verifies users during login.
2. Ticket-Granting Server (TGS): Issues „„proof of identity tickets.““
3. Bob the server: Actually does the work Alice wants performed.

A diagrammatic representation of the interfaces involved in Kerberos may be put as:



$T_{C,S}$ = {client, service, IP address, timestamp, lifetime, $K_{C,S}$}
= Ticket for Client to use Service

A_C = {Client, IP Address, Timestamp}
= Client Authenticator

The exchanges of information between the want of transaction by a User with the application server and the time that they actually start exchanging data may be put as:

1. **Client to the Authentication Server(AS):** The following data in plain text form are sent:
   - Username.
   - Ticket Granting Server(TGS) name.
   - A nonce id 'n'.
2. Response from the **Authentication Server(AS) to the Client:** The following data in encrypted form with the key shared between the AS and the Client is sent:
   - The TGS session key.
   - The Ticket Granting Ticket. This contains the following data encrypted with the TGS password and can be decrypted by the TGS only.
     - Username.
     - The TGS name.
     - The Work Station address.
     - The TGS session key.
   - The nonce id 'n'.
3. **Client to the Ticket Granting Server:** This contains the following data
   - The Ticket Granting ticket.
   - Authenticator.
   - The Application Server.
   - The nonce id 'n'
4. **Ticket Granting Server to the Client:** The following data encrypted by the TGS session key is sent:
   - The new session key.
   - Nonce id 'n'

- Ticket for the application server- The ticket contains the following data encrypted by the application servers' key:
  - Username
  - Server name
  - The Workstation address
  - The new session key.

After these exchanges the identity of the user is confirmed and the normal exchange of data in encrypted form using the new session key can take place. The current version of Kerberos being developed is Kerberos V5.

Types of Tickets

1. **Renewable Tickets:** Each ticket has a timer bound , beyond that no authentication exchange can take place . Applications may desire to hold tickets which can be valid for long periods of time.
2. **Post Dated Tickets:** Applications may occasionally need to obtain tickets for use much later, e.g., a batch submission system would need tickets to be valid at the time the batch job is serviced. **Proxiable Tickets:** At times it may be necessary for a principal to allow a service to perform an operation on its behalf. The service must be able to take on the identity of the client, but only for a particular purpose
3. **Forwardable Tickets:** Authentication forwarding is an instance of the proxy case where the service is granted complete use of the client's identity.

Time Stamps:

- **Authentication:** This is the time when i first authenticated myself .
- **Start:** This is the time when valid period starts.
- **End:** This is the time when valid period ends.
- **Renewal time:** This is the time when ticket is renewed.
- **Current time:** This time is for additional security. This stops using old packets. Here we need to synchronize all clocks.

Cross Realm Authentication

- The Kerberos protocol is designed to operate across organizational boundaries. A client in one organization can be authenticated to a server in another.
- Each organization wishing to run a Kerberos server establishes its own "realm".
- The name of the realm in which a client is registered is part of the client's name, and can be used by the end-service to decide whether to honor a request.

Limitations of Kerberos

- **Password Guessing:** Anyone can get all privileges by cracking password.
- **Denial-of-Service Attack:** This may arise due to keep sending request to invalid ticket.
- **Synchronization of Clock:** This is the most significant limitation to the kerberos.