Unit 1

INTRODUCTION

A collection of interconnected computers are known as Computer Networks.

Mention the uses of computer networks. (Or)
List some uses of computer networks. (5 marks)
Uses of Computer Networks:

1) Business Applications
- Resource sharing
- Client-server model
- Electronic mail
- Electronic commerce

2) Home Applications
- Access to remote information
- Person – to – person communication
- Interactive entertainment
- Electronic commerce

3) Mobile Users
4) Social Issues

1) Business Applications

Resource Sharing

To make all data's, programs and equipments available to anyone on the network without regard to the physical location of the resource and the user. For example, office, a group of workers to share a common printer.

Client – Server Model

A company information system consists of one or more databases and number of employees who need to access them remotely. The data are stored on computers called servers. The employees have machine called clients, which they access remote data. The client and server machines are connected by a network as shown below.

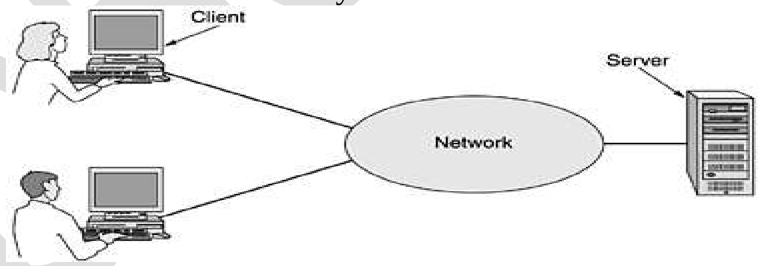


Fig: A network with two clients and one server

This is called client- server model.
- The two processes are involved, one on the client machine and one on the server machine.
- The client process sends a message over the network to the server process.
- The client process then waits for a reply message.
- When the server process gets the request, it performs the requested work or looks up the requested data and sends back a reply.

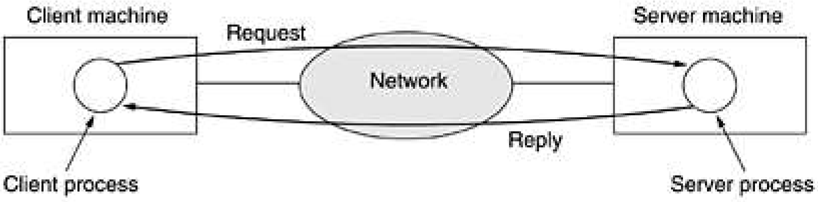


Fig: The client-server model involves requests and replies.

Electronic Mail (E-Mail)

- It is a powerful communication medium among employees.
- Daily communications among employees are easy.
- With a network, it is easy for two or more people who work to write a report together. When one worker makes a change to an on-line document, the other can see the change immediately.
- Speedup makes cooperation among far-flung groups of people easy.

Electronic Commerce (E-Commerce)

- It is doing business with consumers over the internet.
- Many companies provide catalogs of their goods and services on-line and take orders on-line.
- For example, Airlines, bookstores, etc.

2) Home Applications

Access to remote information

- It can be surfing the World Wide Web for information.
- Information available includes the arts, business, cooking, government, health, history, hobbies, recreation, science, travel and many others.
- Many newspapers have gone on-line and can be personalized.
- Beyond newspaper the on-line digital library are used.

Person – to –person Communication

- E-Mail is used as a daily basis by millions of people all over the world.
- It contains audio, video, text and pictures.
- Newsgroups - one person posts a message and all the other subscribes to the newsgroups can read it.
- Peer – to –peer communication with one or more other people, there is no fixed clients and servers.

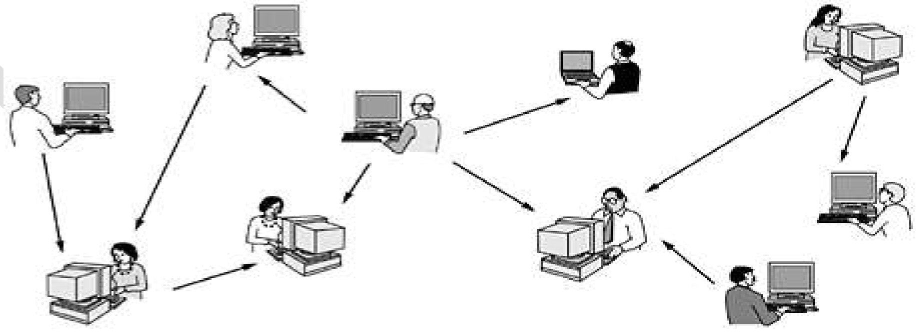


Fig: In a peer-to-peer system there are no fixed clients and servers

Interactive Entertainment

- Live television may also become interactive, with the audience participating in quiz shows, choosing among contestants and so on.
- The killer applications here is video on demand.
- New films become interactive

Electronic Commerce

- Home shopping is very popular.
- The most popular e-commerce is listed below:

Business – to – Consumer	Ordering books on-line
Business – to –Business	Car manufacturer ordering tires from supplier
Government – to – Consumer	Government distributing tax forms electronically
Consumer – to – Consumer	Auctioning second- hand products on-line
Peer- to- Peer	File sharing

3) Mobile Users

- Mobile computers, such as notebook computers and personal digital assistants (PDAs) are one of the fastest- growing segments of the computer industry.
- Wireless networks are important to the military.

4) Social Issues

- The widespread introduction of networking has introduced new social, ethical and political problems.
- The trouble comes when newsgroups are set politics, religion, etc.,

Explain the dimensions of network hardware (Or)
Write the two dimension of network hardware (5 marks)

Network Hardware

The two dimensions of network hardware are:

1) Transmission technology
2) Scale

Transmission Technology

There are two types of transmission technology are: 1) Broadcast links and 2) Point – to –Point links.

Broadcast links

- Broadcast network have a single communication channel that is shared by all the machines on the network.
- Short messages or packets can sent by any machine are received by all the others.
- An address field within the packet specifies the intended recipient.
- Upon receiving a packet is intended processes the packet, if the packet is intended for some other machine, it is just ignored.
- When a packet with this code is transmitted, it is received and processes by every machine on the network. This mode of operation is called broadcasting

Point – to –point links

- Point – to point network consist of many connections between individuals pairs of machines.
- To go from the source to the destination, a packet on this type of network may have to first one or more intermediate machines.
- Often multiple routes; choose the shortest path to send the packets.
- Point – to –Point transmission with one sender and one receiver is sometimes called unicasting.

Scale

The types of networks are classified in their scale are shown below:

Inter process distance	**Processors located in same**	**Example**
1m	Square Meter	Personal Area Network
10m	Room	LAN
100m	Building	LAN
1km	Campus	LAN

10km	City	MAN
100km	Country	WAN
1000km	Continent	WAN
10,000km	Planet	Internet

Explain in details the types of Networks. (Or)
Explain briefly the network hardware (10 marks)

Types of Networks

1) Local Area Network
2) Metropolitan Area Network
3) Wide Area Network
4) Wireless Area Network
5) Home Network
6) InterNetwork

1) Local Area Network (LAN)

- LANs are privately owned networks within a single building or campus of up to a few kilometers in size.
- They are widely used to connect personal computer and workstation in company offices and factories to share resources and exchange information.
- The characteristics are
  - their size
  - their transmission technology
  - their topology
- Various topologies are possible for broadcast LANs. They are 1) Bus and 2) Ring

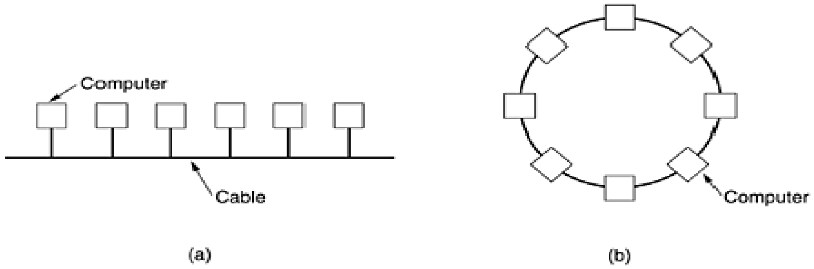


Fig: Two broadcast networks. (a) Bus (b) Ring

Bus Network

- In a bus network, one machine is the master and is allowed to transmit.
- All other machines are required to refrain from sending.

Ring Network

- In a ring network, each bit propagates around on its own, not waiting for the packet to which it belongs.
- Each bit is a ring can takes the time to transmit a few bits, often before the complete packet has even been transmitted.

2) Metropolitan Area Network (MAN)

- MAN covers a city.
- There were locally designed ad-hoc systems.
- Entire channels designed for cables only.
- The cable channels were highly specialized, such as all news, all sports, all cooking, all gardening and so on.
- Both television signals and internet being fed into the centralized head end for subsequent distribution to people’s home.

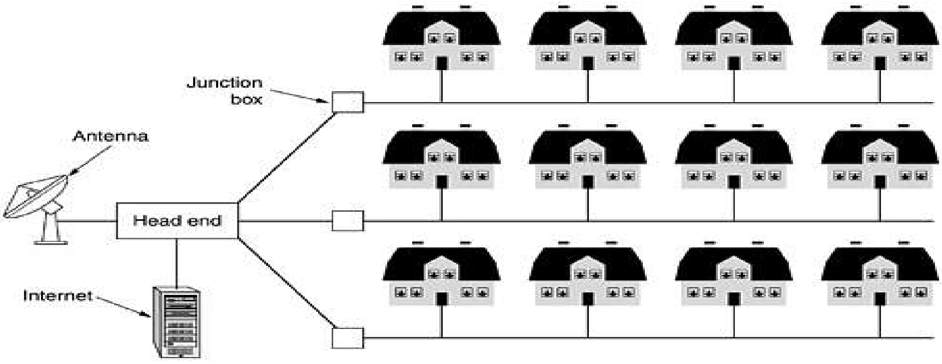


Fig: A metropolitan area network based on cable TV

The best example is the cable television network.

3) Wide Area Networks (WAN)

- WAN spans a large geographical area.
- It contains a collection of machines intended for running user programs. These machines are called hosts.
- The hosts are connected by a communication subnet or subnet.
- The hosts are owned by the customers.
- The subnet is operated by Telephone Company or Internet Service Provider.
- The subnet is to carry message from host to host.
- The subnet consists of two distinct components 1) Transmission Technology and 2) Switching Elements.
- Transmission lines move bits between machines.
- They can be made of copper wire, optical, fiber or even radio links.
- Switching elements are specialized computers that connect two or three transmission links.
- When data arrive on an incoming line, the switching elements must choose an outgoing line on which to forward them.
- These switching elements are called Routers.

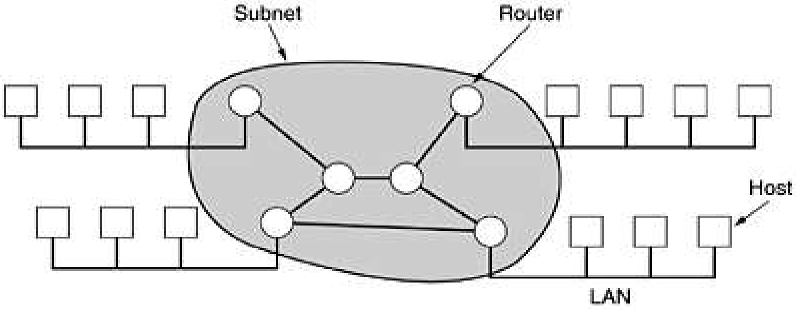


Fig: Relation between hosts on LANs and the subnet

- Each host is connected to a LAN on which a router is present and a host can be connected directly to the router.
- A collection of communication lines and routers from the subnet.
- A collection of routers and communication lines that moved packets from the source host to the destination host.
- When a packet is sent from one router to another via one or more intermediate routers, the packet is received at each intermediate router, stored there until the required output line is free, and then forward.
- This is called a store- and – forward or packet – switched subnet.

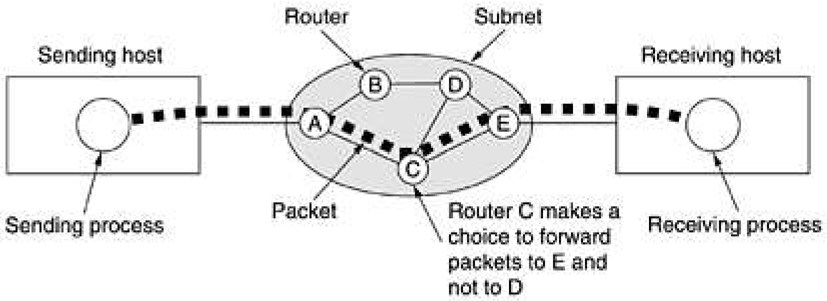


Fig: A stream of packets from sender to receiver

When a process on some host has sent a message to receiving process on some other host, the sending host first cuts the message into packets, each one bearing its number in the sequence. These packets are injected into the network one at a time. The packets are transported individually over the network and deposited at the receiving hosts, where they are reassembled into the original message and delivered to the receiving process. Choose the shortest path and sends a packets from source to destination host.

4) Wireless Networks

Wireless Networks can be divided into:

i) System Interconnected ii) Wireless LANs and iii) Wireless WANs

i) System Interconnected

System interconnected is all interconnecting the components of a computer using short-range radio. Every computer has a monitor, Keyboard, mouse, and printer connected to the main units by cables. Some companies got together to design a short-range wireless network called Bluetooth to connect these components without wires.

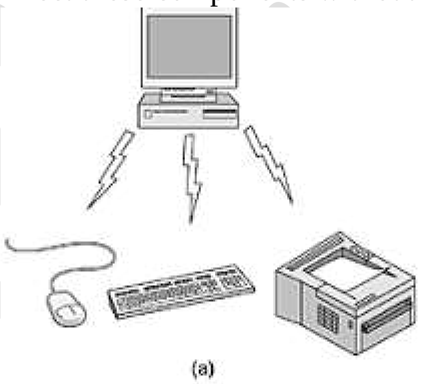


Fig: a) Bluetooth configuration.

Bluetooth also allows digital cameras, headsets, scanners, and other devices to connect to a computer within the range. No cables, no driver installation, just put them down, turn them on and they work.

ii) Wireless LANs

Every computer has a radio modem and antenna with which it can communicate with other systems. If the systems are close enough, they can communicate directly with one another in a peer - to – peer configuration. The standard wireless LANs called IEEE 802.11

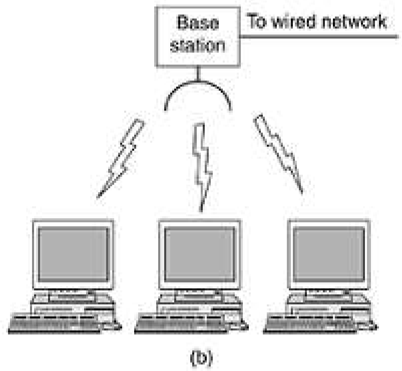


Fig: (b) Wireless LAN

iii) Wireless WANs

The radio network used for cellular telephones is an example for low-bandwidth wireless systems. These wireless systems have three generations are: i) First generation was analog and for voice only.
ii) Second generation was digital and for voice only. iii) Third generation is digital and is for both voice and data.

5) Home Networks

Every device in the home will be capable of communicating with every other device and all of them will be accessible over the internet. Some categories are:
1) Computer (Desktop PC, Notebook PC, PDA, Shared Peripherals)
2) Entertainment (TV, DVD, VCR, camcorder, camera, stereo, MP3)
3) Telecommunications (telephone, mobile telephone, intercom, fax)
4) Appliances (microwave, refrigerator, clock, furnace, airco, lights)
5) Telemetry (utility meter, smoke/burglar alarm, thermostat, babycam)

6) InterNetworks

- A collection of interconnected networks is called an Internetwork or internet.
- A common form of internet is a collection of LANs connected by a WAN.
- Subnet makes the collection of routers and communication lines owned by the network operator.
- A combination of subnet and its hosts forms a network.
- An Internetwork is formed when distinct networks are interconnected.
- Connecting a LAN and a WAN or connecting two LANs forms an internet.

Explain the Network Software. (Or)
Describe in detail the Network Software. (10 marks)

The methods of Network Software are:
1) Protocol Hierarchies
2) Design issues for the layers
3) Connection-oriented and Connectionless services
4) Service primitives
5) The relationship of services to protocols

1) Protocol Hierarchies

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 each layer, the contents of each layers and the function of each layer differ from network to network. The purpose of each layer is to services to the higher layers. A protocol is an agreement between the communicating parties on how communication is to proceed. A five-layer network is illustrated:

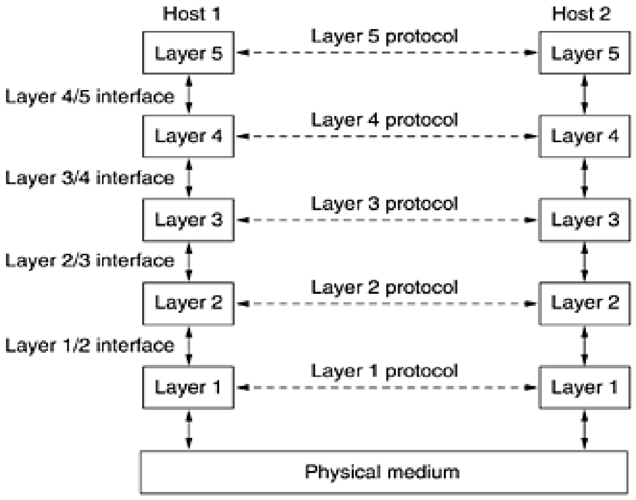


Fig: Layers, protocols, and interfaces

The entities comprising the corresponding layers on different machines are called peers. Peers may be process, hardware devices, or even human beings. Peers that communicate by using the protocol.

No data are directly transferred from layer n on one machine to layer n on another machine. Each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer is the physical medium through which actual communication occurs.

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. A set of layer and protocols is called a network architecture. A list of protocols used by a certain systems. One protocol per layer is called protocol stack.

2) Design Issues for the layers

Design issues that occur in computer networks are present in several layers. Every layer needs a mechanism for identifying senders and receiver. Network has many computers.

They have rules for data transfer. In some systems, data only travels in one direction, in others, data can go both ways.

Error control is an important issue because physical communication circuits are not perfect. The receiver must have some way of telling the sender which message has been correctly received and which has not.

Some kinds of feedback from the receiver to the sender, either directly or indirectly, about the receiver’s current situation. This subject called flow control.

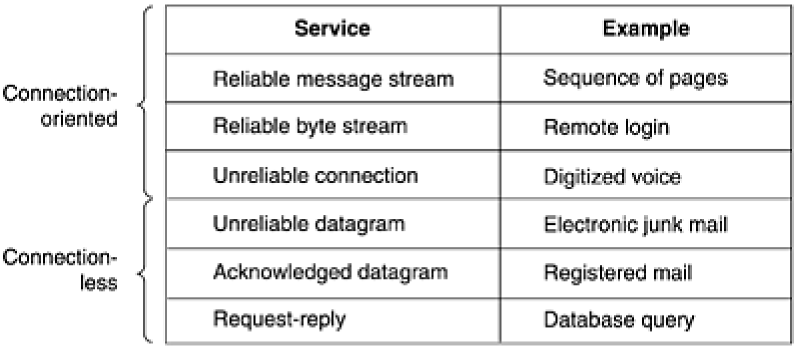
A problem in sending large message for disassembling, transmitting, and then reassembling messages. When there are multiple paths between source and destination a route must have be chosen.

3) Connection-oriented and Connectionless services

Connection-oriented service is modeled after the telephone system. To talk to someone, you pick up the phone, dial the number, talk and then hang up. To use a connection –oriented network service, the service users first establish a connection, user the connection, and then release the connection.

Connection-less service is modeled after the postal system. Each message carries the full destination address, and each one is routed through the system independent of all the others.

Each service can be characterized by a quality of service. There are six different types of services are:

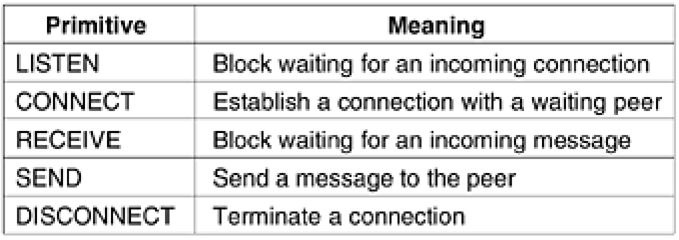


	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Remote login
	Unreliable connection	Digitized voice
Connection-less	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Registered mail
	Request-reply	Database query

Fig: Six different types of services

4) Service primitives

A service is specified by a set of primitives available to a user process to access the service. The five service primitives for implementing a simple connection –oriented services are:



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

Fig: Five service primitives for implementing a simple connection-oriented service

These primitive might be used as follows:

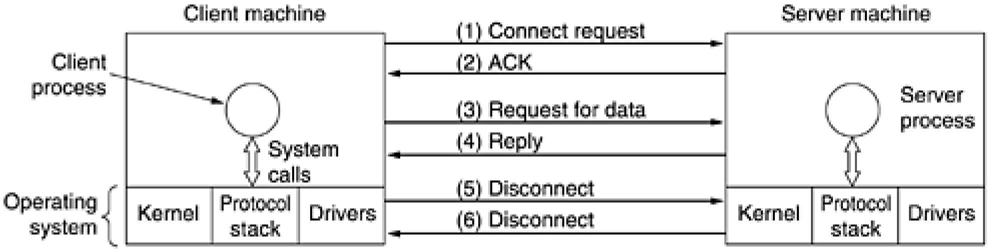


Fig: Packets sent in a simple client-server interaction on a connection-oriented network

- The server executes LISTEN to indicate that it is prepared to accept incoming connections. The connection is to make it a blocking system call.
- The client process executes CONNECT to establish a connection with the server.
- The servers to execute RECEIVE to prepare to accept the first request. The RECEIVE call blocks the server.
- The client machine asks the request for data to the server.
- The server reply the request asks from the client as an acknowledgement.
- SEND is to sends a message from client to server.
- The client sends the DISCONNECT to the server, they reply DICONNECT to client.

5) The relationship of services to protocols

- A service is a set of primitives that a layer provides to the layer above it.
- A Protocol is a set of rules governing the format and meaning of the packets, or messages that are exchanged by the peer entities within a layer.

- Service relate to the interface between layers.
- Protocol relate to the packets send between peer entities on different machines.
- A service is an object-oriented language or an abstract data type.
- A protocol relates to implementation of the service and is not visible to the user of the service.
- Service primitive SEND PACKET with the user providing a pointer to a fully assembled packet.
- All changes to the protocol were immediately visible to the users.

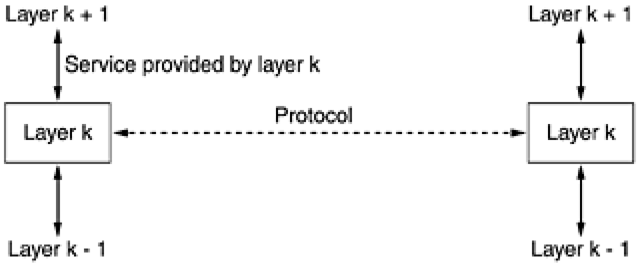


Fig: The relationship between a service and a protocol

Discuss about the OSI Reference Model (Or)
Explain briefly the ISO-OSI Reference Model. (10 Marks)

The OSI model is based on a proposed developed by the International Standards Organization (ISO) towards international standardization of the protocols used in the various layers. The model is called ISO -

OSI (Open Systems Interconnected) reference model because it deals with connecting open system.

The principles of seven layers are:

1) A layer should be created where a different abstraction is needed.
2) Each layer should perform a well-defined function.
3) The function of each layer should be chosen with an eye.
4) The layer boundaries should be chosen to minimize the information flow across the interfaces.
5) The number of layers should be large enough.

The OSI model has seven layers.

1) Physical Layer
2) Data link Layer
3) Network Layer
4) Transport Layer
5) Session Layer
6) Presentation Layer
7) Application Layer

1) Physical Layer

The physical layer is concerned with transmitting raw bits over a communication channel. When one side sends a 1 bit, it is received by the other side as a 1 bit, not as a 0 bit.

2) Data link Layer

The data link layer is to transform a raw transmission facility into a line that appears free undetected transmission errors to the network layer. Senders break up the input data into data frames and transmit the frame sequentially.

3) Network Layer

The network layer controls the operation of the subnet. A key design issue is determining how packets are routed from source to destination. When a packet has to travel from one network to another to get to its destination, many problems can arise. In broadcast networks, the routing problem is simple.

4) Transport Layer

The transport layer is to accept data from above, split it up into smaller units pass these to the network layer and all arrive correctly at the other end. Transport connection is an error- free point-to-point channel that delivers messages or bytes in the order in which they were sent. The transport layer is a true
end-to-end layer, all the way from source to destination.

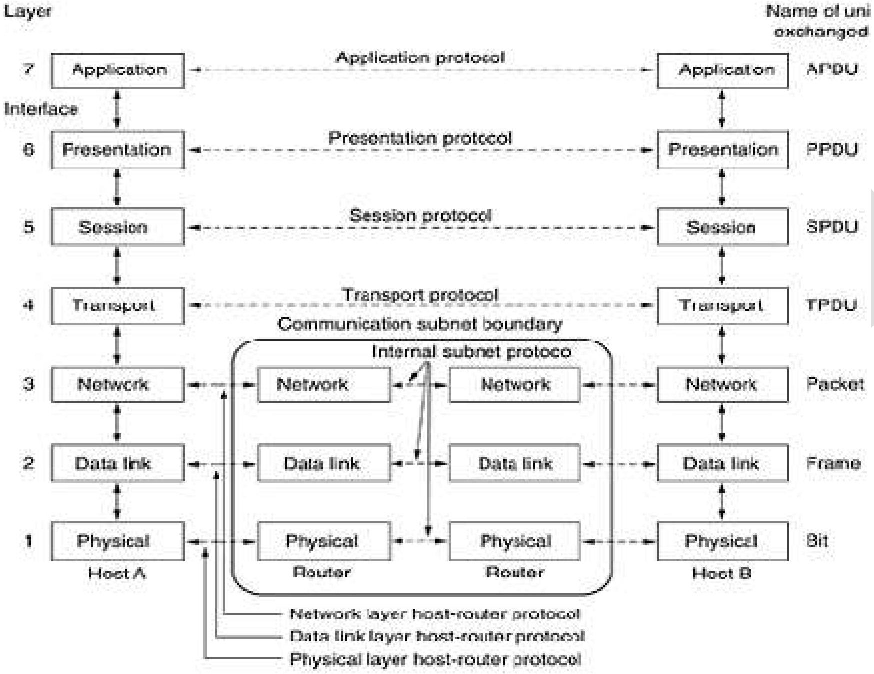


Fig: The OSI reference model

5) Session Layer

The session layer allows users on different machines to establish session between them sessions offer various services, including dialog control, token management and synchronization.

Dialog-control – keeping track of whose turn it is to transmit

Token management – preventing two parties from attempting the same critical operation at the same time

Synchronization – check pointing long transmissions to allow them to continue from where there were after a crash.

6) Presentation Layer

The Presentation Layer is concerned with the syntax and semantics of the information transmitted. They manage these abstract data structures and allow higher-level data structures to be defined and exchanged.

7) Application Layer

The application layer contains a variety of protocols that are commonly needed by users. Application protocol is HTTP (Hypertext Transfer Protocol), which is the basis for the World Wide Web. When a browser wants a web page, it sends the name of the page it wants to the server using HTTP. The server then sends the page back. Other application protocols are used for file transfer, electronic mail and network news.

Write short notes on TCP/IP Reference Model. (5 Marks)

TCP/IP Reference Model

When satellite and radio networks were added later, the existing protocols had trouble interworking with them, so a new reference architecture was needed. This architecture is known as TCP/IP Reference Model.

1) The Internet Layer

This entire requirement led to the choice of a packet switching network based on a connectionless internetwork layer. This layer called the internet layer. Its job is to permit hosts to inject packets into any network and have them travel independently to the destination.

The internet layer defines an official packet format and protocol called IP. The job of the internet layer is to deliver IP packets where they are supposed to go.

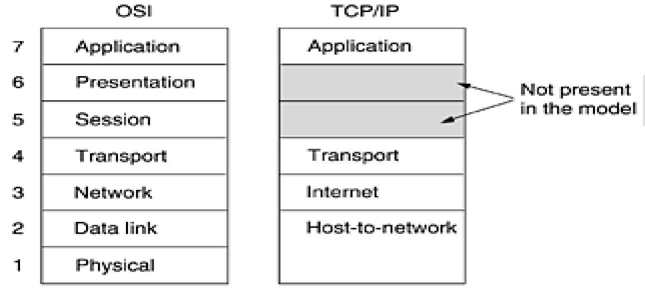


Fig: The TCP/IP reference model

2) The Transport Layer

Transport Layer is designed to allow peer entities on the source and destination hosts to carry on a conversation. Two end – to –end transport protocols have been defined here. TCP is 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. At the destination the receiving TCP process reassembles the received messages into the output stream.

The second protocol in this layer, UDP (User Datagram Protocol) is an unreliable, connectionless protocol for application that do not wants TCP's sequencing or flow control and wish to provide their own. The relation of IP, TCP and UDP is shown below:

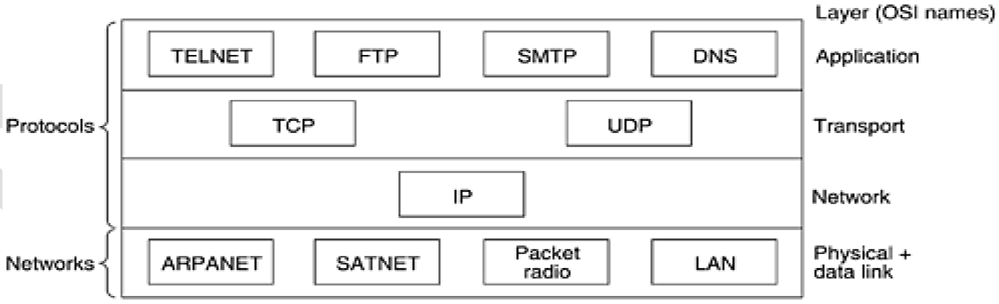


Fig: Protocols and networks in the TCP/IP model initially

3) The Application Layer

The TCP/IP model does not have session or presentation layer on top of the transport layer in the application layer. It include virtual terminal (TELNET), file transfer (FTP) and electronic mail (SMTP). The virtual terminal protocol allows a user on one machine to log onto a distant machine and work there.

The file transfer protocol provides a way to move data efficiently from one machine to another. Electronic mail was originally just a land of file transfer, but later a specialized protocol (SMTP) was developed for it.

4) The Host-to-Network Layer

TCP/IP reference model does not really say much about what happens here. This protocol is not defined and varies from host to host and network to network.

Write down the difference between the OSI and TCP Reference Model. (5 marks)
OSI Reference Model

- It has seven layers are Physical, Data Link, Network, Transport, Session, Presentation & Application.
- It supports both connectionless and connection-oriented communication in the network layer.
- It supports only connection-oriented communication in the transport layer.
- It clearly distinguishes between service, interface and protocol.
- Based on the concept of a stack of independent protocols.

TCP/IP Reference Model

- It has four layers are Internet, Transport, Application and Host-to-Network.
- It supports only connectionless communication in the network layer.
- It supports both connection-oriented and connection-less communication in the transport layer.
- It did not clearly distinguish between service, interface and protocol.
- Based on the concept of a stack of independent protocols.

Discuss the critique of the OSL Reference Model. (5 marks)
The critique of the OSI Reference model

1) Bad timing: The amount of activity surrounding new subjects. When the subject is first discovered, there is a burst of research activity in the form of discussion, papers and meetings. The subject was poorly understood, the result is bad standards are effectively ignored.
2) Bad Technology: The seven layers was more political than technical and two of the layers (session and presentation) are nearly empty, whereas other ones (data link and network) are overfull. OSI has some functions such as addressing, flow control and error control, reappear again and again in each layer.
3) Bad Implementation: The complexity of the model and the protocols it will come as no surprise that the initial implementation were huge unwidely and slow.
4) Bad Politics: The poor researches and programmers down in the trenches actually developing computer network did not help much.

Write the critique of the TCP/IP Reference Model. (5 Marks)
The critique of the TCP/IP Reference Model

- The model does not clearly distinguish the concepts of service, interface and protocol.
- The model is not all general and is poorly suited to describing any protocol stack other than TCP/IP.
- The host-to-network layer is not really a layer at all in the normal sense of the term in the context of layered protocol.
- The model does not distinguish the physical and data link layers. These are completely different.
- The IP and TCP protocols were carefully thought out and well implemented, many of the other protocols were ad-hoc.

Discuss about the various physical media. (5 0r 10 Marks)
The various physical media are:

1) Magnetic media 2) Twisted pair 3) Coaxial cable 4) Fiber optics.

1) Magnetic Media

The most common ways to transport data from one computer to another is to write them onto magnetic tape or removable media, (eg: recordable DVDs) physically transport the tape or disks to the destination machine and read them back it again.

2) Twisted Pair

A twisted pair consists of two insulated copper wires, typically about 1mm (main memory) thick. The wires are twisted together in a helical form. 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 wires radiates less effectively.

The most common application of the twisted pair is the telephone system. Twisted pair can run several kilometers without amplifications, but for longer distances, repeaters are needed. Twisted pairs can be used for transmitting either analog or digital signals.

Twisted pair cabling comes in several varieties: 1) Category 3 twisted pairs and 2) Category 5 twisted pairs.

The category 3 twisted pairs consist of two insulated wires gently twisted together. For such pairs are typically grouped in a plastic sheath to protect the wires and keep them together.

. The category 5 twisted pairs has 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.

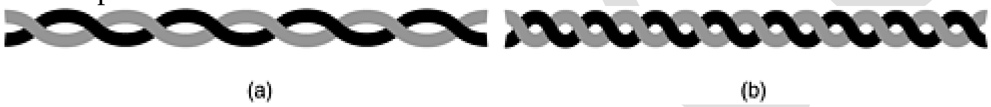


Fig: (a) Category 3 UTP. (b) Category 5 UTP

3) Coaxial Cable

- It can span longer distances at higher speeds.
- Two kinds of coaxial cable are: 1) 50-ohm cable and 2) 75- ohm cable.
- 50 –ohm cable is used when it is intended for digital transmission from the start.
- 75-ohm cable is used for analog transmission and cable television.

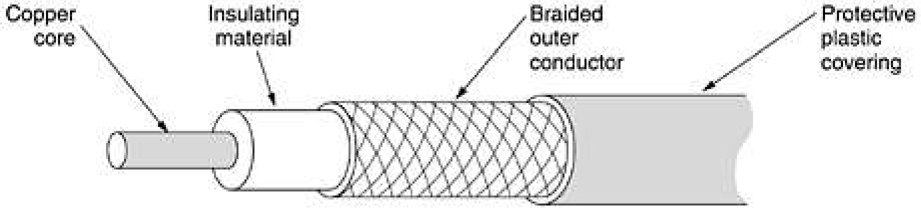


Fig: A coaxial cable

- 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.
- The outer conductor is covered in a protective plastic sheath.
- 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 is widely used for cable television and metropolitan area network.

4) Fiber Optics

Fiber optic cables are at the center is the glass core through which the light propagates.
In multimode fibers, the core is 50 microns in diameter, about the thickness of a human hair.
In single mode fibers, the core is 8 to 10 microns.

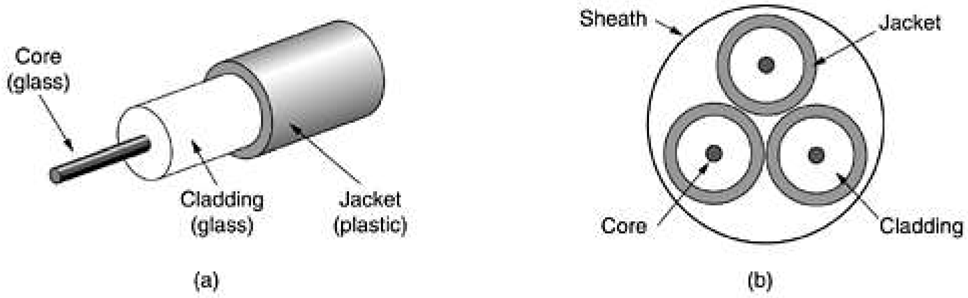


Fig: a) Side view of a single fiber. (b) End view of a sheath with three fibers

- 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 grouped in bundles, protected by an outer sheath.
- 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.
- Third, two pieces of fiber can be fused to form a solid connection. A fusion splice is almost as good as a single drawn fiber, but even there, a small amount of attenuation occurs.

------- END OF UNIT I ---------

UNIT : II

List out the Wireless Transmission. (5 or 10 marks)

The Wireless Transmission are:

1) Electromagnetic spectrum
2) Radio transmission
3) Microwave transmission
4) Infrared and millimeter waves
5) Light wave transmission

1) Electromagnetic Spectrum

When electrons move, they create electromagnetic waves that can propagate through space. The number of oscillations per second of a wave is called its frequency f, and is measured in Hz (Hertz). The distance between two consecutive maxima (or minima) is called the wavelength, λ (lambda).

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 $3x10^8$ m/sec or about 1 foot (30 cm) per nanosecond.

The fundamental relation between f, λ , and C is: $\lambda f = C$. For example: $\lambda f = 300$. The Electro Magnetic Spectrum is shown:

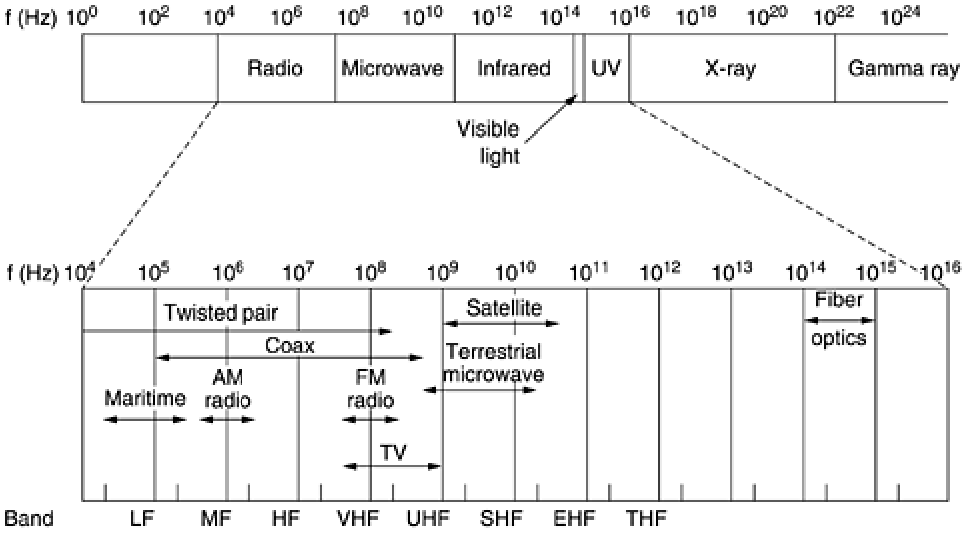

f (Hz) 10^0 10^2 10^4 10^6 10^8 10^{10} 10^{12} 10^{14} 10^{16} 10^{18} 10^{20} 10^{22} 10^{24}
Radio
Microwave
Infrared
UV
X-ray
Gamma ray
Visible light
f (Hz) 10^4 10^5 10^6 10^7 10^8 10^9 10^{10} 10^{11} 10^{12} 10^{13} 10^{14} 10^{15} 10^{16}
Twisted pair
Coax
Satellite
Terrestrial microwave
Fiber optics
Maritime
AM radio
FM radio
TV
Band
LF
MF
HF
VHF
UHF
SHF
EHF
THF

Fig: The electromagnetic spectrum and its uses for communication

The radio, micro wave, 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. 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. The higher bands were later named the Very, Ultra, Super, Extremely and Tremendously high frequency bands.

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 are omni directional.

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 others electrical equipment.

In the VLF, LF, and MF bands, radio waves follow the ground is illustrated:

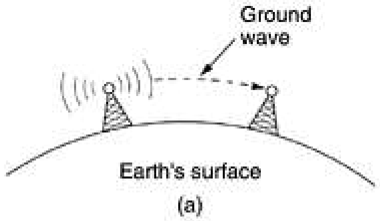


Fig: (a) In the VLF, LF, and MF bands, radio waves follow the curvature of the earth.

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 the HF and VHF bands, the ground waves tend to be absorbed by the earth. The waves that search the ionosphere, a layer of charged particles circling the earth at a height at a height of 100 to 500 km are refracted by it and sent back to earth as shown:

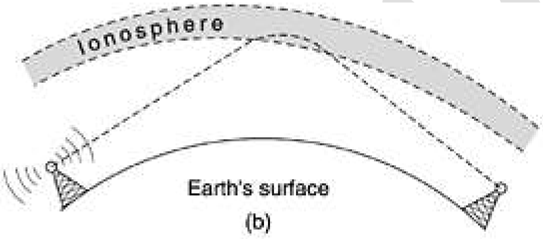


Fig: (b) In the HF band, they bounce off the ionosphere.

Amateur radio operators 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 recently straight lines and can be narrowly focused concentrating all the energy into a small beam by means of a parabolic antenna gives a much higher signal-to-noise must be accurately aligned with each other. The microwaves travel in a straight line.

Microwaves do not pass through buildings. 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.

Bands up to 10GHz are now in routine use, but at about 4 GHz a new problem sets in: absorption by water. These waves are only a few centimeter long and are absorbed by rain.

Microwave communication is widely used for long distance telephone communication, mobile phones, television distribution and other uses that a severe shortage of spectrum has developed. Microwave is also relatively inexpensive.

4) Infrared and millimeter waves

Unguided infrared and millimeter waves are widely used for short-range communication. The remote controls used on television, 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 walls well is also a plus. It means that an infrared system in one room of a building will not interface with a similar system in adjacent rooms or buildings.

You cannot control your neighbor's television with your remote control. No government license is needed to operate an infrared system, in contrast to radio systems. Infrared communication has a limited use on the desktop, for example, connecting notebook computer and printers.

5) Light wave Transmission

The application is to connect the LANs in two buildings via lasers mounted on their rooftops. Coherent optional signaling using lasers in inherently unidirectional, each building needs its own laser and its own photodetector. This scheme offers very high bandwidth and very low cost. The laser's strength, a very narrow beam is weakness here.

A disadvantage is laser beam cannot penetrate rain or thick fog, but they normally work well on sunny days. Heat from the sun during the day time caused convection currents to rise up from the roof of the building. This turbulent air diverted the beam and made it dance around the detector. Atmosphere “seeing” like this makes the stars twinkle. It is also responsible for shimmering roads on a hot day and the wavy images seen when one looks out above a hot radiator.

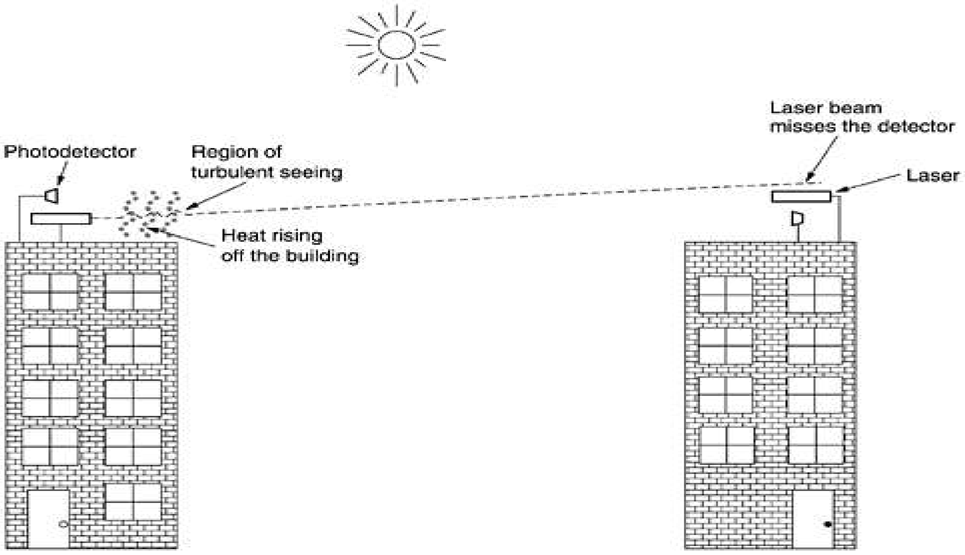

Photodetector
Region of turbulent seeing
Heat rising off the building
Laser beam misses the detector
Laser

Fig: Convection currents can interfere with laser communication systems. A bidirectional system with two lasers is pictured here

Describe in detail the geostationary satellite. (5 marks)

Geostationary Satellite

Geostationary satellites including the orbits, solar panels, radio frequencies and launch procedures. Satellites were impractical due to the impossibility of putting power-hungry, fragile, vacuum tube amplifiers into orbit. Communication satellites have become a multibillion dollar business and the only aspect of outer space that has become highly profitable. These high-flying satellites are often called GEO.

Modern satellites can be quite large, weighing up to 4000 kg and consuming several kilowatts of electric power produced by the solar panels. The effects of solar, lunar and planetary gravity tend to move them away from their assigned orbit slots and orientations, an effect countered by on-board rocket motors. This fine tuning activity is called stationary keeping. ITU has allocated certain frequency bands to satellite users. The main ones are listed below:

Band	Downlink	Uplink	Bandwidth	Problems
L	1.5 GHz	1.6 GHz	15 MHz	Low bandwidth: crowded
S	1.9 GHz	2.2 GHz	70 MHz	Low bandwidth; crowded
C	4.0 GHz	6.0 GHz	500 MHz	Terrestrial interference
Ku	11 GHz	14 GHz	580 MHz	Rain
Ka	20 GHz	30 GHz	3500 MHz	Rain; Equipment cost

A modern satellite has around 40 transponders, each with an 80 MHz bandwidth. Each transponder operates as a bent pipe, but recent satellites have some on-board processing capacity, allowing more sophisticated operation.

The first geostationary satellites had a single spatial beam that illuminated about 1/3 of the earth’s surface, called its footprint. Each satellite is equipped with multiple antennas and multiple transponders. Each downward beam can be focused on a small geographical area, so multiple upward and downward transmissions can take place simultaneously. Typically, these so-called spot beams are elliptically shaped and can be small as few hundred km in diameter.

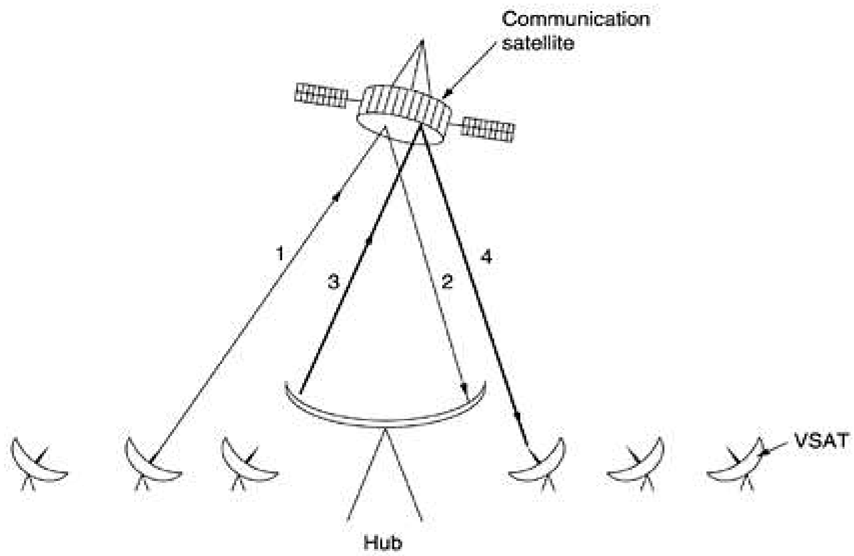


Fig: VSATs using a hub.

A new development in the communication satellite world is the development of low-cost micro stations, sometimes called VSATs (Very Small Aperture Terminals). Direct broadcast satellite television user this technology for one way transmissions.

In many VSAT systems, the micro stations do not have enough power to communicate directly with one another. The hub, with a large, high gain antenna is needed to relay traffic between VSATs. In this mode of operation either the sender or the receiver has a large antenna and a powerful amplifier. The trade-off is a longer delay in return for having cheaper and user stations.

Another important property of satellites is that they are inherently broadcast media. It does not cost more to send a message to thousands of stations within a transponder's footprint than it does to send to one.

Write down the difference between satellites and fibers. (5 Marks)

Communication satellites have some major niche markets that does not address. We have a few of these:

1) A first while a single fiber has more potential bandwidth is not available to most users. The fibers that are now being installed are within the telephone system to handle many long distance calls at once, not to provide individual users with high bandwidth.

2) A second niche is for mobile communications many people nowadays want to communicate while jogging, driving, sailing and flying. Terrestrial fiber optic links are of no use of them, but satellite links potentially are.

3) A third niche is for situations in which broadcasting is essential. A message sent by satellite can be received by thousands of ground stations at once.

4) A fourth niche is for communication in places with hostile terrain or a poorly developed terrestrial infrastructure.

5) A fifth niche market for satellites is to cover areas where obtaining the right of way for laying fiber is difficult or unduly expensive.

6) A sixth niche, when rapid deployment is critical as in military communication systems in time of war, satellites wins easily.

Discuss in detail the structure of the telephone systems. (10 marks)

The public switched telephone network

When the distances are large or there are many computers on the cables have to pass through a public road or other public right of way, the costs of running private cables are usually prohibitive. The network designers must rely on the existing telecommunication facilities. These facilities especially the PSTN (Public Switched Telephone Network).

Structure of the telephone systems

If the telephone owner wanted to talk to a other telephone owners, separate wires had to be string to all n houses. Within a year, the cities were covered with wires passing over houses and trees in a wild jumble. It became immediately obvious that the model of connecting every telephone to every other telephone, as shown (Fig: a) was not going to work.

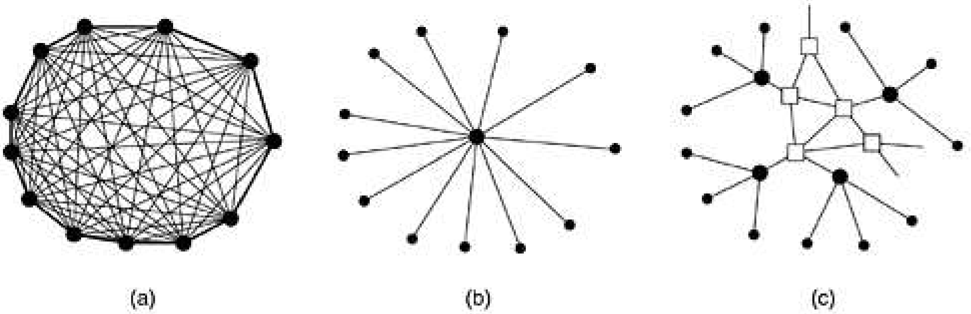


Fig: (a) Fully-interconnected network. (b) Centralized switch. (c) Two-level hierarchy

To make a call, the customer would crank the phone to make a ringing sound in the telephone company office to attract the attention of an operator, who would then manually connect the caller to the callee by using a jumper cable. The model of a single switching office is illustrated (Fig b).

The switching offices were singing up everywhere and people wanted to make long-distance calls between cities, begin to connect the switching offices. To connect every switching office to every other switching office by means of a wire between them quickly became unmanageable, so second-level switching offices were invented. After a while multiple second-level offices were needed as illustrated (Fig: c).

The three major parts of the telephone system were in place: the switching offices, the wires between the customers and the switching offices and the long-distance connection between the switching offices.

Each telephone has two copper wires coming out of it that go directly to the telephone company's nearest end office(also called a local central office). The distance is typically 1 to 10 km, being shorter in cities than in rural areas. The two-wire connections between each subscriber's telephone and the end office are known in the trade as the local loop.

If a subscriber attached to a given end office call another subscriber attached to the same end office, the switching mechanism between the two local loops. This connection remains intact for the duration of the call.

If the called telephone is attached to another end office, a different procedure has to be used. Each end office has a number of outgoing lines to one or more nearly switching centers, called toll offices. These lines are called toll connecting trunks. If both the caller's and callee's end offices happen to have a toll connecting trunk to the same toll office, the connection may be established within the toll offices. A telephone network consists only of telephones, end offices and toll offices as shown:

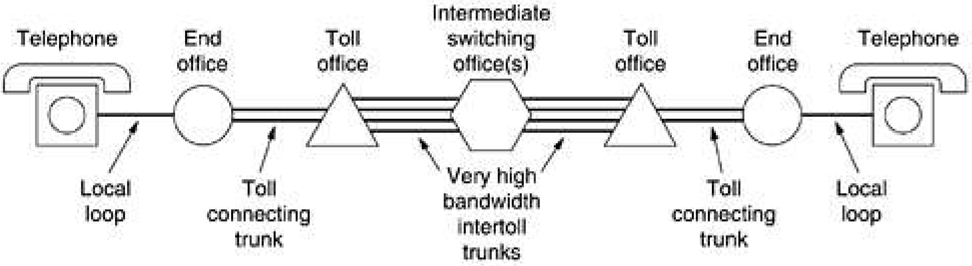


Fig: A typical circuit route for a medium-distance call

If the caller and callee do not have a toll office in common, the path will have to be established somewhere higher up in the hierarchy. Primary, sectional and regional exchanges communicate with each other via high bandwidth inter toll trunks (also called inter office trunks). The number of different kinds of switching centers and their topology.

The telephone system consists of three major components:

1. Local loops (analog twisted pairs going into houses and business)

2. Trunks (digital fiber optics connecting the switching offices)
3. Switching offices (where calls are moved from one trunk to another)

Explain MODEM. (Or)
Write short notes on Modulation and Demodulation. (5 marks)

The square waves used in digital signals have a wide frequency spectrum and subject to strong attenuation and delay distortion. These effects make base band (DC) signaling unsuitable except at slow speeds and over short distances.

To get around the problems associated with DC signaling, especially on telephone lines, AC signaling is used. A continuous tone in the 1000 to 2000 Hz range called a sine wave carrier is introduced. Its amplitude, frequency, or phase can be modulated to transmit information.

In amplitude modulation, two different amplitudes are to represent 0 and 1 respectively. In frequency modulation, also known as frequency shift keying, two (or more) different tones are used. In the simplest form of phase modulation, the carrier wave is systematically shifted 0 or 180 degrees at uniformly speed intervals. A better scheme is to use shifts of 45, 135, 225 or 315 degrees to transmit 2 bits of information per time interval.

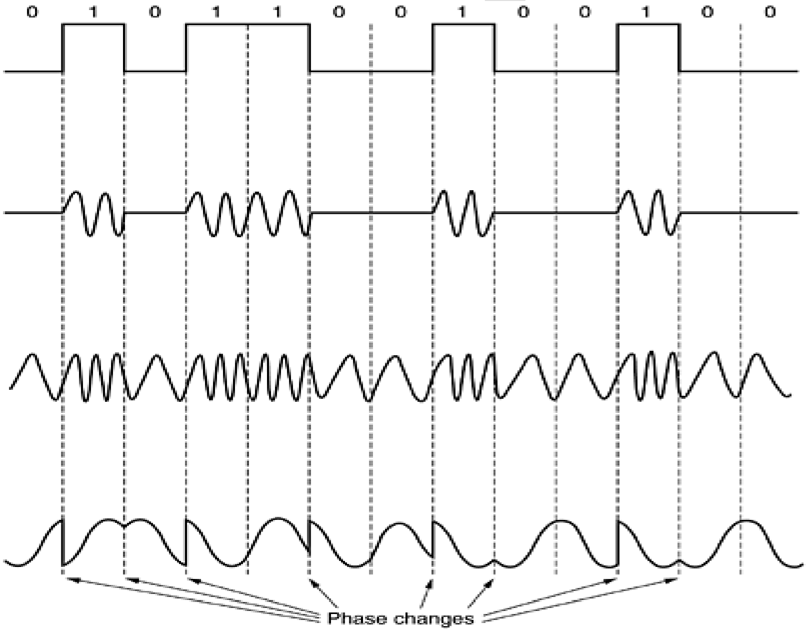


Fig: (a) A binary signal. (b) Amplitude modulation. (c) Frequency modulation. (d) Phase modulation

A device that accepts a serial stream of bits as input and produces a carrier modulated by one of these methods is called a modem (for modulation – demodulation). The modem is inserted between the (digital) computer and the (analog) telephone system.

All modern modems allow traffic in both directions at the same time. A connection that allows traffic in both directions simultaneously is called full duplex. A two-lane road is full duplex. A connection that allows traffic either way, but only one way at a time is called half duplex. A connection that allows traffic only one way is called simplex. A one-way street is simplex.

Explain the trunks and multiplexing. (10 marks)

To install and maintain a high-bandwidth trunk as a low-bandwidth trunk between two switching offices. Telephone companies have developed elaborate schemes for multiplexing many conversations over a single physical trunk. These multiplexing schemes can be divided into two basic categories: FDM (Frequency Division Multiplexing) and TDM (Time Division Multiplexing). In FDM, the frequency spectrum is divided into frequency bands, with each user having exclusive possession of some band. In TDM, the users take turns, each one periodically getting the entire bandwidth for a little burst of time.

Frequency Division Multiplexing

The three voice-grade telephone channels are multiplexed using FDM. Filters limit the usable bandwidth to about 3100 Hz per voice-grade channel. When many channels are multiplexed together, 4000 Hz is allocated to each channel to keep them well separated. First the voice channels are raised in frequency, each by a different amount. Then they can be combined because no two channels now occupy the same portion of the spectrum. There is gaps between the channels, there is some overlap between adjacent channels because the filters do not have sharp edges. This overlaps means that a strong spike at the edge of one channel will be felt in the adjacent one as non-thermal noise.

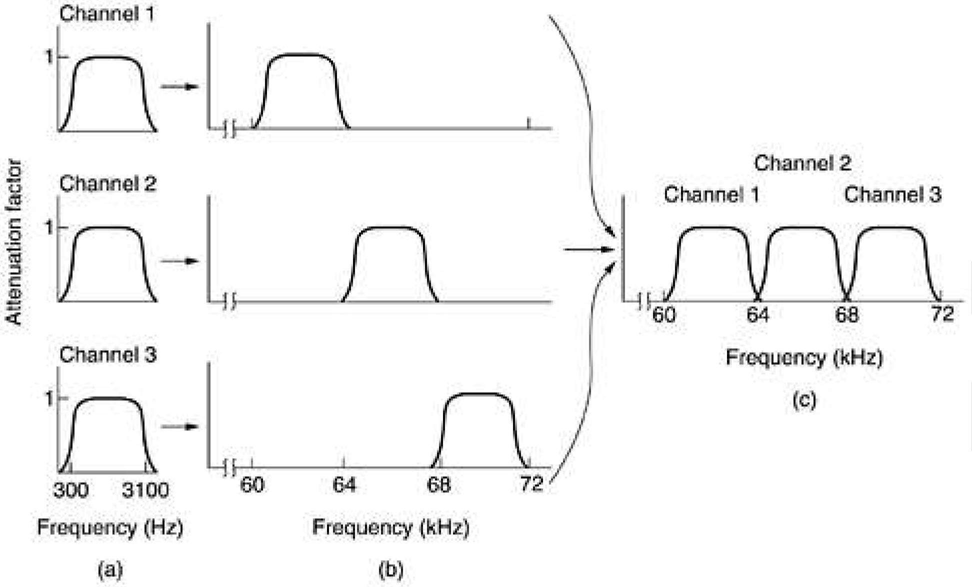


Fig: Frequency division multiplexing.
(a) The original bandwidths. (b) The bandwidths raised in frequency. (c) The multiplexed channel

A widespread standard is twelve 4000-Hz voice channels multiplexed into the 60 to 108 KHz band. This unit is called a group. Five groups can be multiplexed to form a super group. The next unit is the master group, which are five super groups or ten super groups.

Wavelength Division Multiplexing

For fiber optic channels, a variation of frequency division multiplexing is used. It is called WDM (Wavelength Division Multiplexing). Four fibers come together at an optical combiner, each with its energy present at a different wavelength. The four beams are combined onto a single shared fiber for transmission to a distant destination. At the far end, the beam is split up over a many fibers as there were on the input side. Each output fiber contains a short, specially constructed core that filters out all but one wavelength. The resulting signals can be routed to their destination or recombined in different ways for additional multiplexed transport.

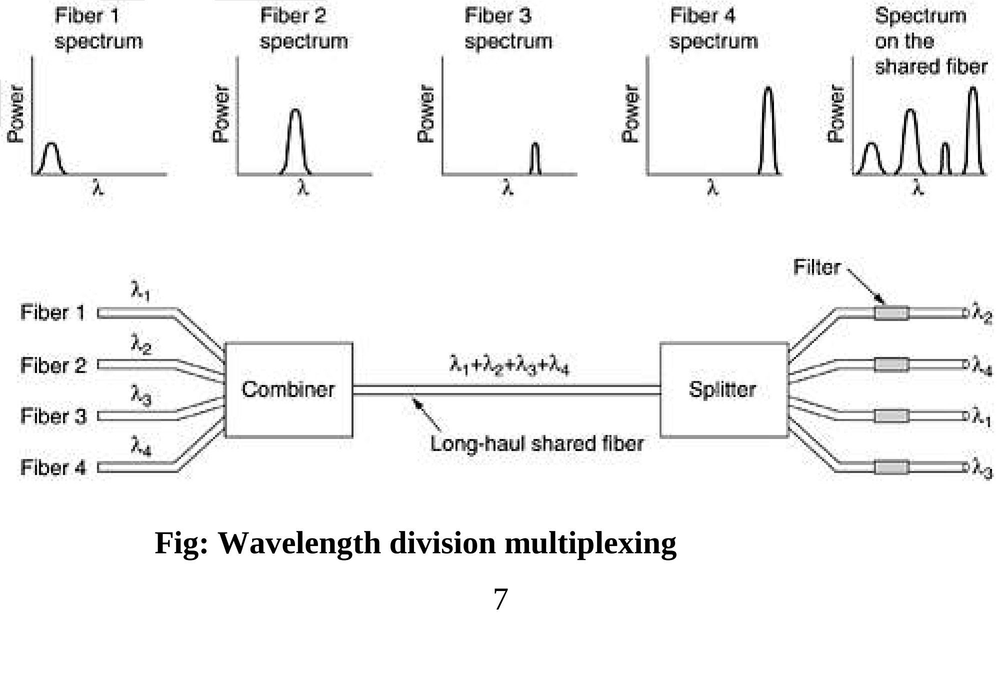


Fig: Wavelength division multiplexing

As long as each channel has its own frequency range and all the ranges and disjoint, they can be multiplexed together on the long haul fiber. The only difference with electrical FDM is that an optical system using a diffraction grating is completely passive and thus highly reliable.

Time Division Multiplexing

TDM can be handled entirely by digital electronics. It can only be used for digital data. The local loops produce analog signal a conversion is needed from analog to digital in the end office, where all the individual local loops come together to be combined onto outgoing trunks.

How multiple analog voice signals are digitalized and combined onto a single outgoing digital analog. The analog signals are digitalized in the end office by a device called codec (coder-decoder). At a lower sampling rate, information would be lost, at a higher one, no extra information would be gained. This technique is called PCM (Pulse Code Modulation).

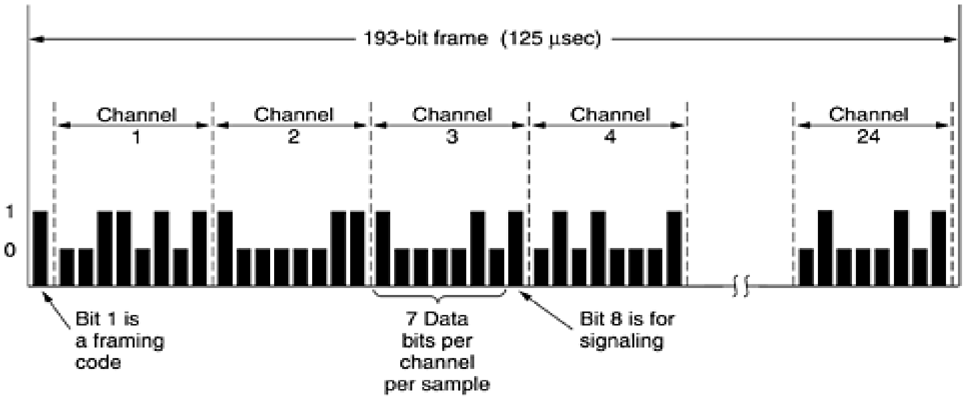


Fig: The T1 carrier (1.544 Mbps).

he T1 carrier consists of 24 voice channels multiplexed together. The analog signals are sampled on a round-robin basis with the resulting analog stream being fed to the codec rather than having 24 separate codec and then merging the digital output. Each of the 24 channels, in turn gets to insert 8 bits into output stream. Seven bits are data and one is for control yielding 7*8000=56,000 bps of data, and 1*8000=8000 bps of signaling information per channel.

A frame consists of 24*8=192 bits plus one extra bit for framing, yielding 193 bits every 125 λsec. This gives a gross data rate of 1.544 mbps. The 193rd bit is used for frame synchronization.

Explain in detail the switching. (Or) Write about the types of switching. (10 marks)

Switching

The phone system is divided into two parts: outside plant (the local loops and trunks, since they are physically outside the switching office) and inside plant (the switches), which are inside the switching offices.

The different switching techniques are: 1) Circuit Switching 2) Message Switching and 3) packet switching.

1) Circuit Switching

When the computer places a telephone call, the switching equipment within the telephone system seeks out a physical path all the way from your telephone to the receiver’s telephone. This technique is called Circuit Switching.

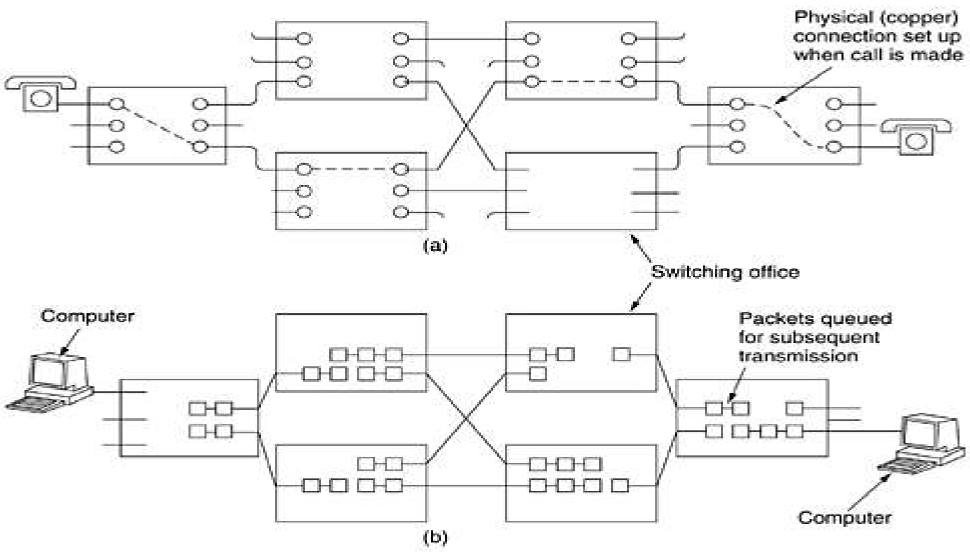

Physical (copper) connection set up when call is made
(a)
Switching office
Computer
Packets queued for subsequent transmission
Computer
(b)

Fig: (a) Circuit switching. (b) Packet switching

Each of the six rectangles represents a carrier switching office (end office, toll office, etc.). Each office has three incoming lines and three outgoing lines. When a call passes through a switching office, a physical connection is established between the line on which the call came in and one of the output links as shown by the dotted lines.

The alternative to circuit switching is packet switching as shown fig(b). Individual packets are sent as need be, with no dedicated path being set up in advance. It is up to each packet to find its way to the destination on its own. An important property of circuit switching is needed to set up an end-to-end path before any data can be sent. The elapsed time between the end of dialing and the start of ringing can easily be 10 sec, more on long-distance or international calls.

Message Switching

When this form of switching is used, no physical path is established in advance between sender and receiver. Instead, when the sender has a block of data to be sent, it is stored in the first switching office (i.e., router) and then forwarded later, one hop at a time. Each block is received in its entirely inspected for error, and then retransmitted. A network using this technique is called a store-and –forward network.

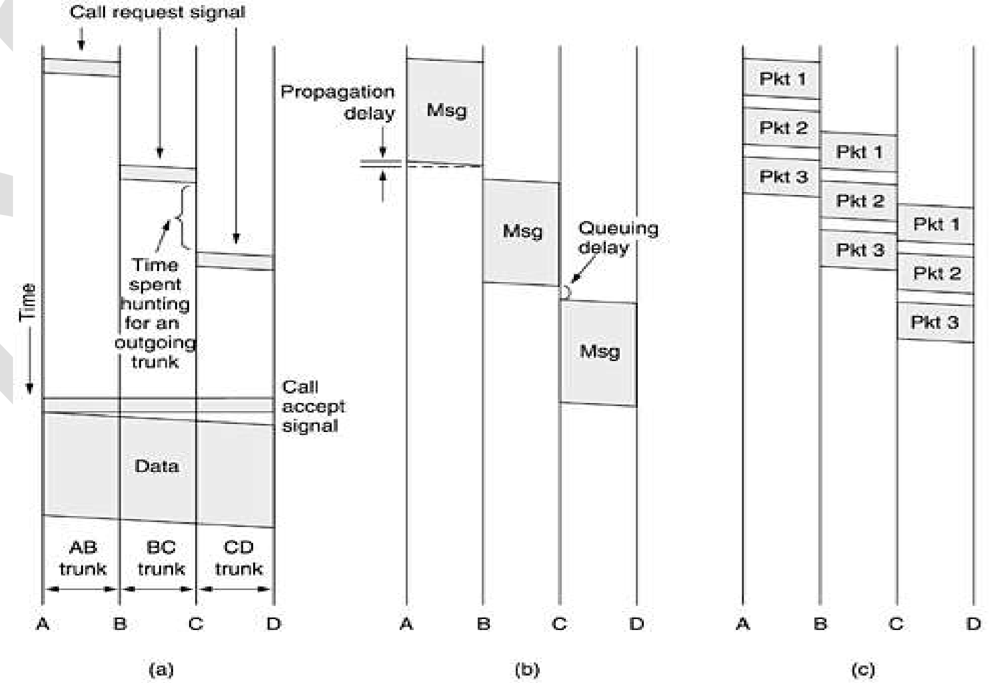


Fig: Timing of events in (a) circuit switching, (b) message switching, (c) packet switching

Packet Switching

With message switching there is no limit at all on block size, which means that routers must have disks to buffer long blocks. It also means that a single block can tie up a router – router line for minutes, rendering message switching useless for interactive traffic. The packet switching was invented packet-switching networks place a tight upper limit on block size, allowing packets to be buffered in router main memory instead of m disk.

The packet switching over message switching as shown fig b and fig c. The first packet of a multi-packet message can be forwarded before the second one has fully arrived, reducing delay and improving throughput. Computer networks are usually packet switched, occasionally circuit switched, put never message switched.

Write the difference between circuit switching and packet switching (5 Marks)

Item	Circuit switching	Packet switching
Call setup	required	Not needed
Dedicated physical path	yes	no
Each packet follows the same route	yes	no
Packets arrives in order	yes	no
Is a switch crash fatal	yes	no
Bandwidth available	fixed	dynamic
Time of possible congestion	At setup time	On every packet
Potentially wasted bandwidth	yes	no
Store-and-forward transmission	no	yes
Transparency	yes	no
Charging	Per minute	Per packet

Explain in detail the mobile telephone systems. (Or)
Discuss the generation of mobile phones. (10 marks)

Mobile phones have gone through three distinct generations, with different technologies:
1. Analog voice 2. Digital voice 3. Digital voice and data (Internet, e-mail, etc.,)

First generation mobile phones: Analog voice

Mobile radiotelephones were used sporadically for maritime and military communication. This system used a single large transmitter on top of a tall building and had a single channel, used for both sending and receiving. To talk, the user had to push a button that enabled the transmitter and disabled the receiver. Such systems known as push – to- talk systems.

IMTS (Improved Mobile Telephone System) used a high-powered (200-watt) transmitter, on top of a hill, but now had two frequencies, one for sending and one for receiving, so the push-to-talk button was no longer needed. IMTS supported 23 channels spread out from 150 MHz to 450 MHz.

Advanced mobile phone system

In all mobile phone systems, a geographic region is divided up into cells. In AMPS, the cells are typically 10 to 20 km across, in digital systems, the cells are smaller. Each cell uses some set of frequencies not used by any of its neighbors.

The cells are all the same size. They are grouped in units of seven cells. Each letter indicates a group of frequencies. Notice that for each frequency set, there is a buffer about two cells wide where that frequency is not reused, providing for good separation and low interference as shown fig a.

In an area were the number of users has grown to the point that the system is overloaded, the power is reduced and the overload cells are split into smaller micro cells to permit more frequency reuse as shown fig b.

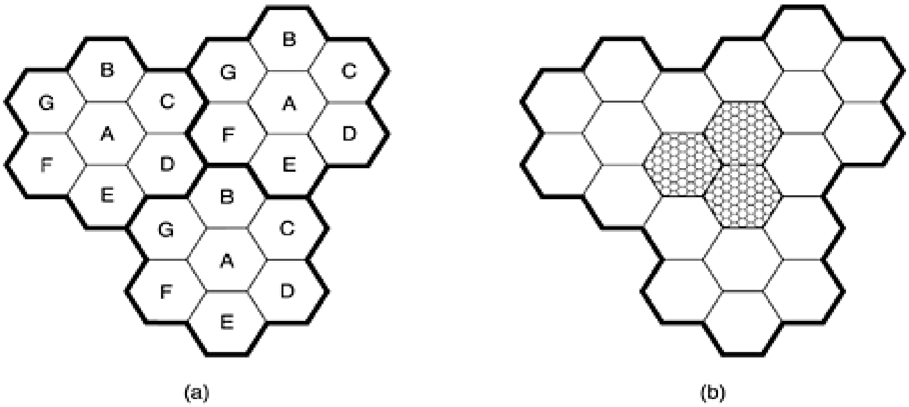


Fig: (a) Frequencies are not reused in adjacent cells. (b) To add more users, smaller cells can be used

At the center of each cell is a base station to which all the telephone in the cell transmit. The base station consists of a computer and transmitter/receiver connected to an antenna. In a small device called an MSTO (mobile telephone switching officer) or MSC (mobile switching center)

Second generation mobile phones: Digital voice

The four systems are in use: 1) D-AMPS, 2) GSM, 3) CDMA and 4)PDC.

1) D-AMPS

- Digital Advanced Mobile Phone System is fully digital.
- It uses the same 30 KHz channels.
- One channel can be analog and one can be digital.
- It can change channel types dynamically
- Voice signal picked up by the microphone is digitalized and compressed using a model.
- The compression is done by a circuit called a vocoder.

D-AMPS, three users can share a single frequency pair using time division multiplexing. Each frequency pair supports 25 frames/sec of 40 msec each. Each frame is divided into six time slots of 6.67 msec each, as illustrated for the lowest frequency pair.

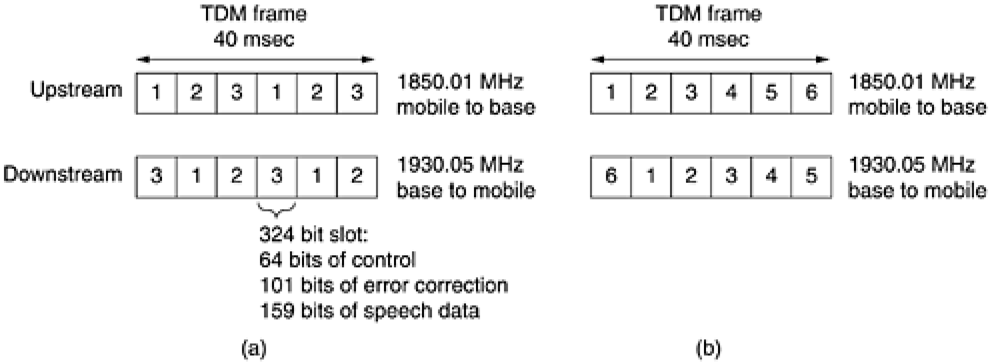


Fig: (a) A D-AMPS channel with three users. (b) A D-AMPS channel with six users

Each frame holds three users who take turns using the upstream and downstream links. During slot 1 of fig (a), for example, user 1 may transmit to the base station and user 3 is receiving from the base station. Each slot is 324 bits long, of which 64 bits are used for guard times, synchronization, and control purposes, leaving 260 bits for the user payload.

2) GSM

The Global System for Mobile communication is a cellular system. Frequency division multiplexing is used with each mobile transmitting one frequency and receiving on a higher frequency. A single frequency pair is split by time-division multiplexing into time slots shared by multiple mobiles.

Each frequency band is 200 KHz wide. A GSM system has 124 pairs of simplex channels. Each simplex channel is 200 KHz wide and support eight separate connections on it, using time division multiplexing. Each currently active station is assigned one time slot on one channel pair.

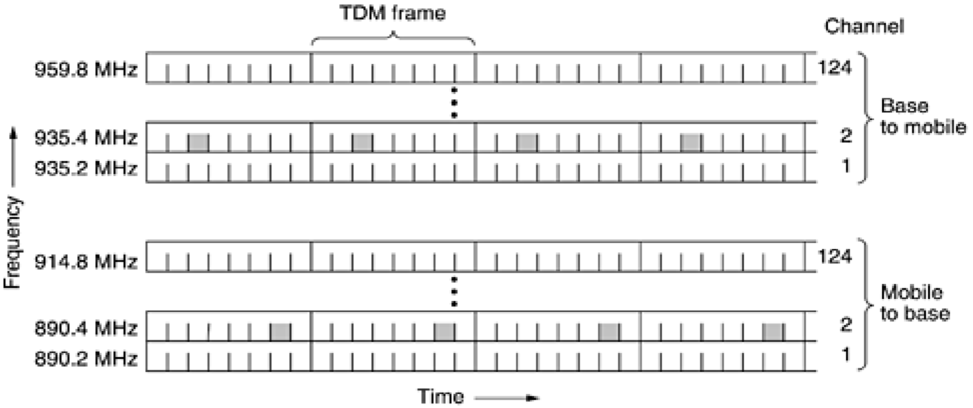


Fig: GSM uses 124 frequency channels, each of which uses an eight-slot TDM system

Each TDM slot has a specific structure and groups of TDM slots form multiframes, also with a specific structure.

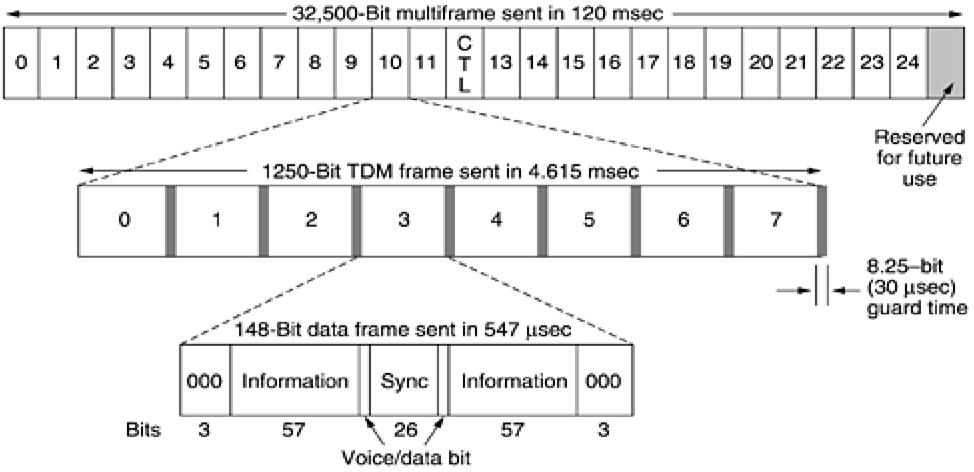
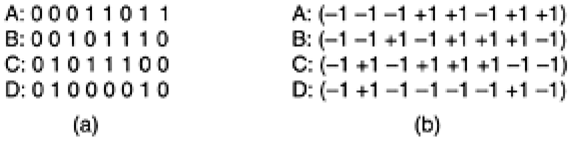


Fig: A portion of the GSM framing structure.

Each TDM slot consists of a 148-bit data frame that occupies the channel for 577 µsec. Each data frame starts and ends with three 0 bits. It contains two 57-bit information fields, each one having a control bit that indicates whether the following information field is for voice or data. Between the information fields in a 26-bit sync filed that is used by the receiver to synchronize to the sender’s frame boundaries.

3) CDMA

In CDMA, each bit time is subdivided into m short intervals called chips. There are 64 or 128 chips per bit. Each station is assigned a unique m-bit code called a chip sequence. To transmit a 1 bit, a station sends its chip sequence. To transmit a 0 bit, it sends the one’s complement of its chip sequence. No other patterns are permitted for m=8, if station A is assigned the chip sequence 00011011, it sends a 1 bit by sending 00011011 and a 0 bit by sending 11100100.



A: 0 0 0 1 1 0 1 1
B: 0 0 1 0 1 1 1 0
C: 0 1 0 1 1 1 0 0
D: 0 1 0 0 0 0 1 0
(a)

A: (−1 −1 −1 +1 +1 −1 +1 +1)
B: (−1 −1 +1 −1 +1 +1 +1 −1)
C: (−1 +1 −1 +1 +1 +1 −1 −1)
D: (−1 +1 −1 −1 −1 −1 +1 −1)
(b)

Fig: (a) Binary chip sequences for four stations. (b) Bipolar chip sequences

It is more convenient to use a bipolar notation, with binary 0 being -1 and binary 1 being +1. A chip sequences in parentheses, so a 1 bit for station A now becomes (-1 -1 -1 +1 +1 -1 +1 +1). Fig (a) shows binary chip sequences for four example stations. Fig (b) shows them in bipolar notation.

Third Generation mobile phones: Digital voice and data

To get a bit more specific and issue a blueprint are called IMT-2000 (International Mobile Telecommunications). The number 2000 sized for three things: 1) the year it was supposed to go into service. 2) the frequency it was supposed to operate at (in MHz) and 3) the bandwidth the service should have (in KHz).

The basic services that IMT-2000 network is supposed to provide to its users are:

1. High-quality voice transmissions
2. Messaging (replacing e-mail, fax, sms, chat, etc.,)
3. Multimedia (playing music, viewing videos, films, televisions, etc.,)
4. Internet Access (web surfing, including pages with audio & video)

GPRS allows mobile stations to send and receive IP packets in a cell running a voice system. When GPRS is in operation, some time slots on some frequencies are reserved for packet traffic. The number and location of the time slots can be dynamically managed by the base station, depending on the ratio of voice to data traffic in the cell.

One logical channel is for downloading packets from the base station to some mobile station, with each packet indicating who it is destined for. To send an IP packet, a mobile station requests one or more time slots by sending a request to the base station. If the request arrives without damage, the base station announces the frequency and time slots allocated to the mobile for sending the packet. Once the packet has arrived at the base station, it is transferred to the Internet by a wired connection. GPRS is one of the 2.5 G scheme.

Discuss the design issues in data link layer. (10 marks)

Data Link Layer Design Issues:

The data link layer has a number of specific functions it can carry out. These functions include:

1) Providing a well-defined service interface to the network layer.
2) Dealing with transmission errors.
3) Regulating the flow of data so that slow receives are not swamped by fast senders.

1) Service provided to the network layer

The function of the data link layer is to provide services to the network layer. The service is transferring data from the network layer on the source machine to the network layer on the destination machine. The data link layer is to transmit the bits to the destination machine so they can be handled over to the network layer as shown fig (a). The actual transmission follows path of fig (b), but the two data link layer process communicating using a data link protocol.

Three reasonable possibilities that are commonly provided are:

1) Unacknowledgement connectionless service.
2) Acknowledgement connectionless service
3) Acknowledgement connection-oriented service

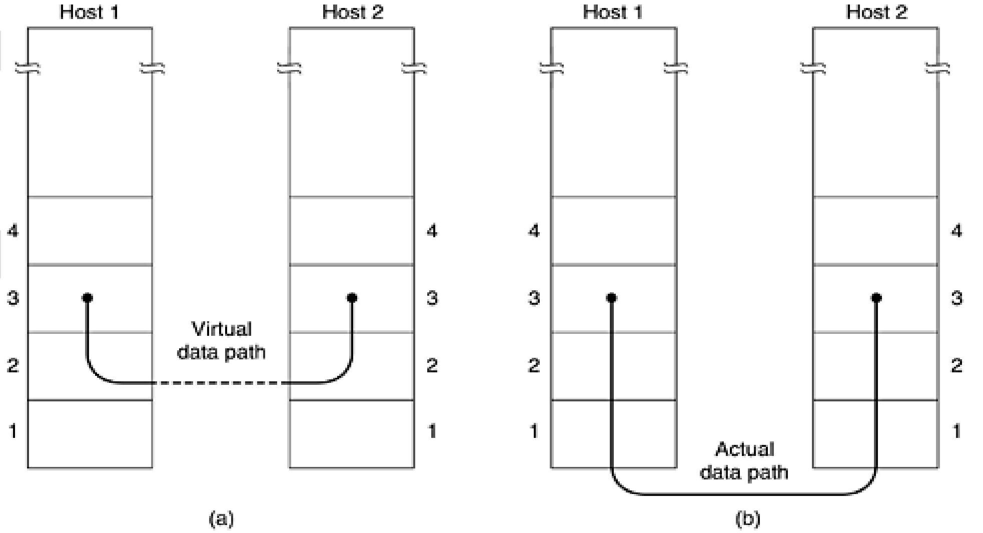


Fig: (a) Virtual communication. (b) Actual communication

Unacknowledgement connectionless service consists of having the source machine send independent frames to the destination machine without having the destination machine acknowledge them. No logical connection is established. If a frame is lost due to noise on the line most LANs are used.

Acknowledged connectionless service, there is no logical connections used but each frame sent is individually acknowledged. The sender knows whether a frame has arrived correctly. If it is not arrived within a specified time interval, it can be sent again. This service is useful over unreliable channels, such as wireless systems.

Acknowledged connection-oriented service, the source and destination machines establish a connection before any data are transferred. Each frame sent over the connection is numbered and the data link layer guarantees that each frame sent is indeed received.

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 gives the packet contained in the payload field to the routing software. The routing software then choose 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:

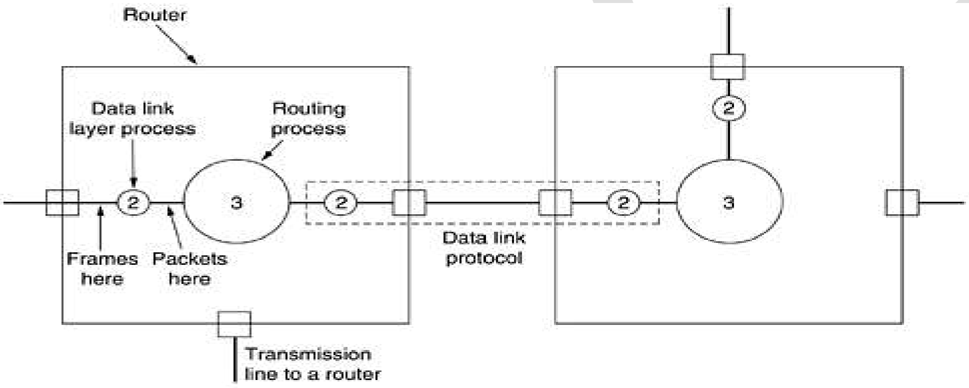


Fig: Placement of the data link protocol

With reliable, sequenced connections on each of the point –to-point lines. It does not want to be bothered too often with packets that got lost on the way. It is up to the data link protocol, shown in the dotted rectangle, to make unreliable communication lines look perfect or, at least, fairly good.

2) Framing

There are four methods of framings are:

i. character count
ii. Flag bytes with byte stuffing
iii. Starting and ending flags, with bit stuffing
iv. Physical layer coding violations

i) Character count

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 the end of the frame is. This technique is shown fig (a) for four frames of size 5, 5, 8 and 8 characters, respectively.

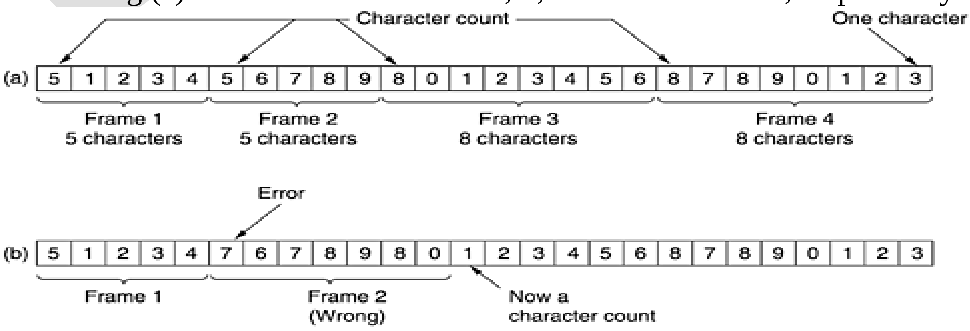


Fig: A character stream. (a) Without errors. (b) With one error

For example, if the characters count of 5 in the second frame of fig (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.

ii) Flag bytes with byte stuffing

In the past, starting and ending bytes were different, but in recent years most protocols have used the same byte, called a flag byte, as both the starting and ending delimiter as shown fig (a) as FLAG. If the receiver ever loses synchronization, it can just search for the flag byte to find the end of the current frame. Two consecutive flag bytes indicate the end of one frame and start of the next one.

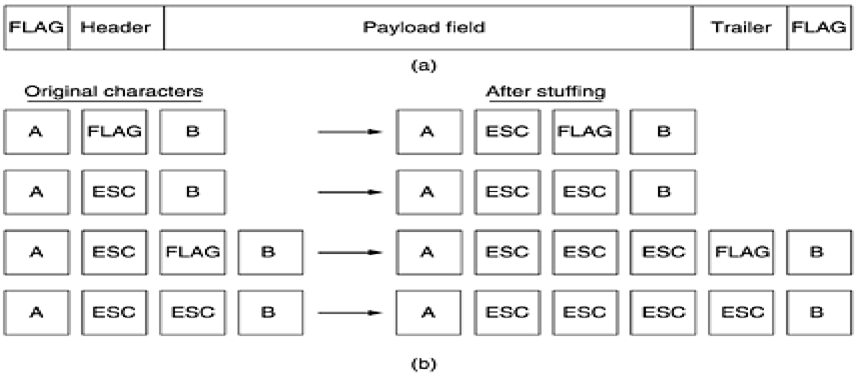


Fig: a) A frame delimited by flag bytes b) Four examples of byte sequences before and after byte stuffing

One way to solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just each "accidental" flag byte in the data. The data link layer on the receiving end removes the escape byte before data are given to the network layer. This technique is called byte stuffing or character stuffing.

iii) Starting and ending flags, with bit stuffing

Each frame begins and ends with a special bit pattern, 0111110. Whenever the sender's data link layer encounters five consecutives 1s in the data, it automatically stuffs a 0 bit into the outing bit stream. This bit stuffing is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing characters stream before a flag byte in the data.

When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (i.e., deletes) the 0 bit. If the user data contain the flag pattern 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110.

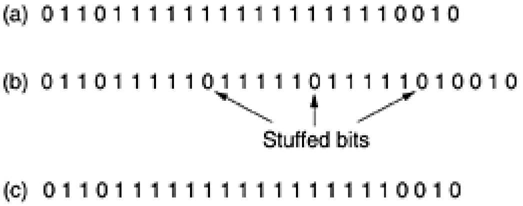

(a) 011011111111111111110010
(b) 011011111011111011111010010
Stuffed bits
(c) 011011111111111111110010

Fig: Bit stuffing. A) The original data. B) The data as they appear on the line. c) The data as they are stored in the receiver's memory after destuffing.

iv) Physical layer coding violations

Framing is 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.

3) Error Control

The protocol calls for the receiver to send back special control frames bearing positive or negative acknowledgements about the incoming frames. If the sender receives a positive acknowledgement about a frame, it knows the frame has arrived safely. A negative acknowledgement means that something has gone wrong, and the frame must be transmitted again. This is known as Error control.

4) Flow Control

When the sender is running on a fast computer and the receiver is running on a slow machine. The sender keeps pumping the frames out at a high rate until the receiver is completely swamped.

Two approaches are commonly used. The first, feedback-based flow control, the 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. The second, rate-based flow control, the protocol has a built-in mechanism that limits the rate at which sender may transmit data, without using feedback from the receiver.

Describe the Error detection and correction (5 marks)

Error-correcting codes

Network designers have developed two basic strategies foe dealing with errors. One way is to include enough redundant information along with each block of data sent, to enable the receiver to deduce what the transmitted data must have been. The other way is to include only enough redundancy to allow the receiver to deduce that an error occurred, but not which error, and have it request a retransmission. This use of error- correcting codes is often referred to as forward-error correction.

A frame consists of m data (i.e., Message) bits and r redundancy, or check bits. The total length be n (i.e., n= m+r). An n-bit unit containing data and check bits is often referred to as an n-bit codeword.

Given any two codeword, say, 10001001 and 10110001, it is possible to determine how many corresponding bits differ. In this case, 3 bits differ. To determine how many bits differ just exclusive-OR the two codewords and count the number of 1 bits in the result, for example:

```
10001001
10110001
________
00111000
```

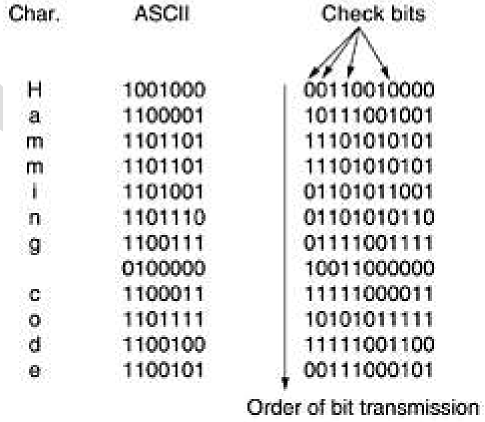
The number of bit position in which two codewords differ is called the Hamming Distance. Its significance is that if two codewords are a hamming distance d apart, it will require d single-bit errors to convert one into the other.

Hamming Method

The bits of the codeword are numbered consecutively, starting with bit 1 at the left end, bit 2 to its immediate right, and so on. The bits that are powers of 2 (1, 2, 4, 8, 16, etc.) are check bits. The rest (3, 5, 6, 7, 9, etc.) are filled up with the m data bits. Each check bit forces the parity of some collection of bits, including itself, to be even (or odd).

When a codeword arrives, the receiver initializes a counter to zero. It then examines each check bit, k (k = 1, 2, 4, 8, ...), to see if it has the correct parity. If not, the receiver adds k to the counter. If the counter is zero after all the check bits have been examined, the codeword is accepted as valid. If the counter is nonzero, it contains the number of the incorrect bit.

For example, if check bits 1, 2, and 8 are in error, the inverted bit is 11, because it is the only one checked by bits 1, 2, and 8.



Char.	ASCII	Check bits
H	1001000	00110010000
a	1100001	10111001001
m	1101101	11101010101
m	1101101	11101010101
i	1101001	01101011001
n	1101110	01101010110
g	1100111	01111001111
	0100000	10011000000
c	1100011	11111000011
o	1101111	10101011111
d	1100100	11111001100
e	1100101	00111000101

Order of bit transmission

Fig: Use of hamming code to correct burst errors.

The above figure shows 7-bit ASCII characters encoded as 11-bit code words using a Hamming code. Remember that the data are found in bit positions 3,5,6,7,9,10 and 11.

Hamming codes can only correct single error. A sequence of k consecutive codewords is arranged as a matrix, one codeword per row. The data would be transmitted one codeword at a time, from left to right. To correct burst error, the data should be transmitted one column at a time, starting with the leftmost column. When all k bits have been sent, the second column is sent, and so on. When the frame arrives at the receiver, the matrix is reconstructed, one column at a time. If a burst error of length k occurs, at most 1 bit in each of the k code words will have been affected, but the hamming code can correct one error per codeword, so the entire block can be restored. This method uses Kr check bits to make blocks of Km data bits immune to a single burst error of length K or less.

Error-detecting codes

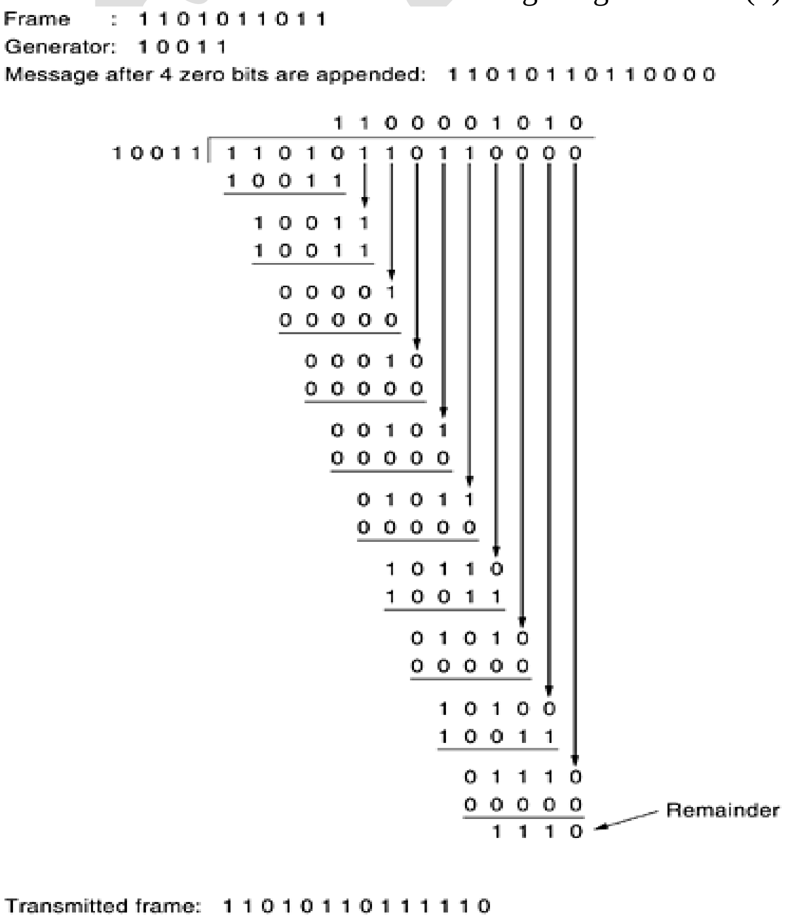
A copper wire or fiber, the error rate is much lower, so error detection and retransmission is usually more efficient there for dealing with the occasional error.

For example, consider a channel on which errors are isolated and the error rate is 10^{-6} per bit. Let the block size be 1000 bits. To provide error correction for 1000-bit blocks, 10 check bits are needed, a megabit of data would require single 1 bit-error, one parity bit per block will suffice. Once every 1000 blocks, an extra block (1001 bits) will have to be transmitted. The total overhead for the error detection + retransmission method is only 2001 bits per megabit of data, versus 10,000 bits for a Hamming code.

The polynomial codes also known as CRC (Cyclic Redundancy Check). Polynomial codes are based upon treating bit strings as representation of polynomials with coefficients of 0 and 1 only. A k-bit frame is regarded as the coefficient list for a polynomial with k terms, ranging from x^{k-1} to x^0. Such a polynomial is said to be of degree k-1. The high-order (leftmost) bit is the coefficient of x^{k-1}, the next bit is the coefficient of x^{k-2} and so on. For example, 110001 have 6 bits and thus represent a six-term polynomial with coefficients 1, 1, 0, 0, 0, and 1: $x^5 + x^4 + x^0$.

When the polynomial code method is employed, the sender and receiver must agree upon a generator polynomial, G(x). Both the high- and low-order bits of the generator must be 1. To compute the checksum for some frame with m bits, corresponding to the polynomial M(x), the frame must be longer than the generator polynomial. The idea is to append a checksum to the end of the frame in such a way that the polynomial represented by the checksummed frame is divisible by G(x). When the receiver gets the checksummed frame, it tries dividing it by G(x). If there is a remainder, there has been a transmission error.

To illustrate the calculation for a frame 1101011011 using the generator $G(x) = x^4 + x + 1$.



Frame : 1 1 0 1 0 1 1 0 1 1

Generator: 1 0 0 1 1

Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0

Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0

Fig: Calculation of the polynomial code checksum

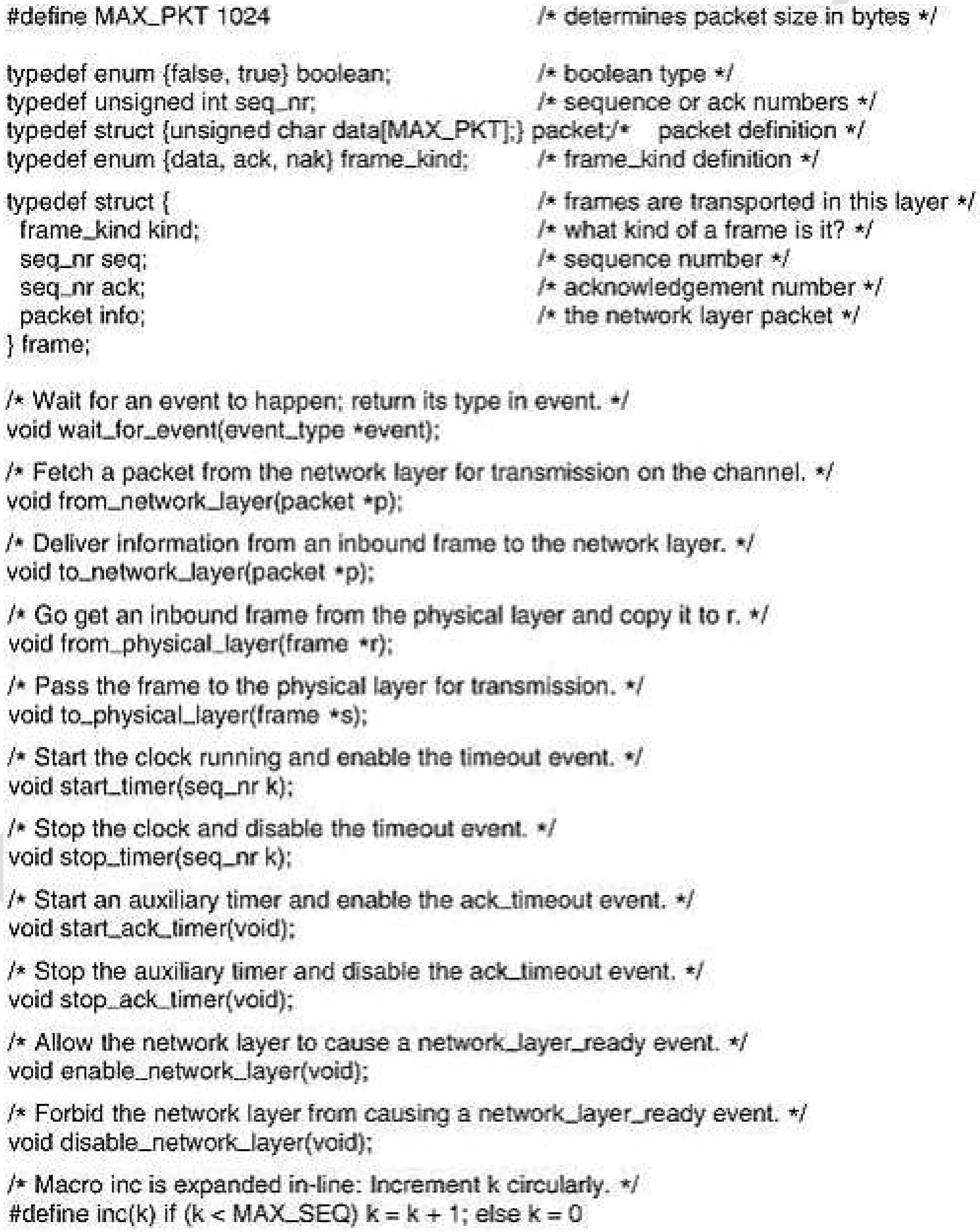
It should be clear that T(x) is divisible (modulo 2) by G(x). In any division problem, if you diminish the dividend by the remainder, what is left over is divisible by the divisor. For example, in base 10, if you divide 210,278 by 10,941, the remainder is 2399. By subtracting 2399 from 210,278, what is left over (207,879) is divisible by 10,941.

UNIT III

Explain the elementary data link protocol (Or)   (5 marks)
Explain the unrestricted simplex protocol (or)   (10 marks)
Explain the stop and wait protocol (or)   (10 marks)
Explain the simplex protocol for a noisy channel   (10 marks).

Elementary Data Link Protocol

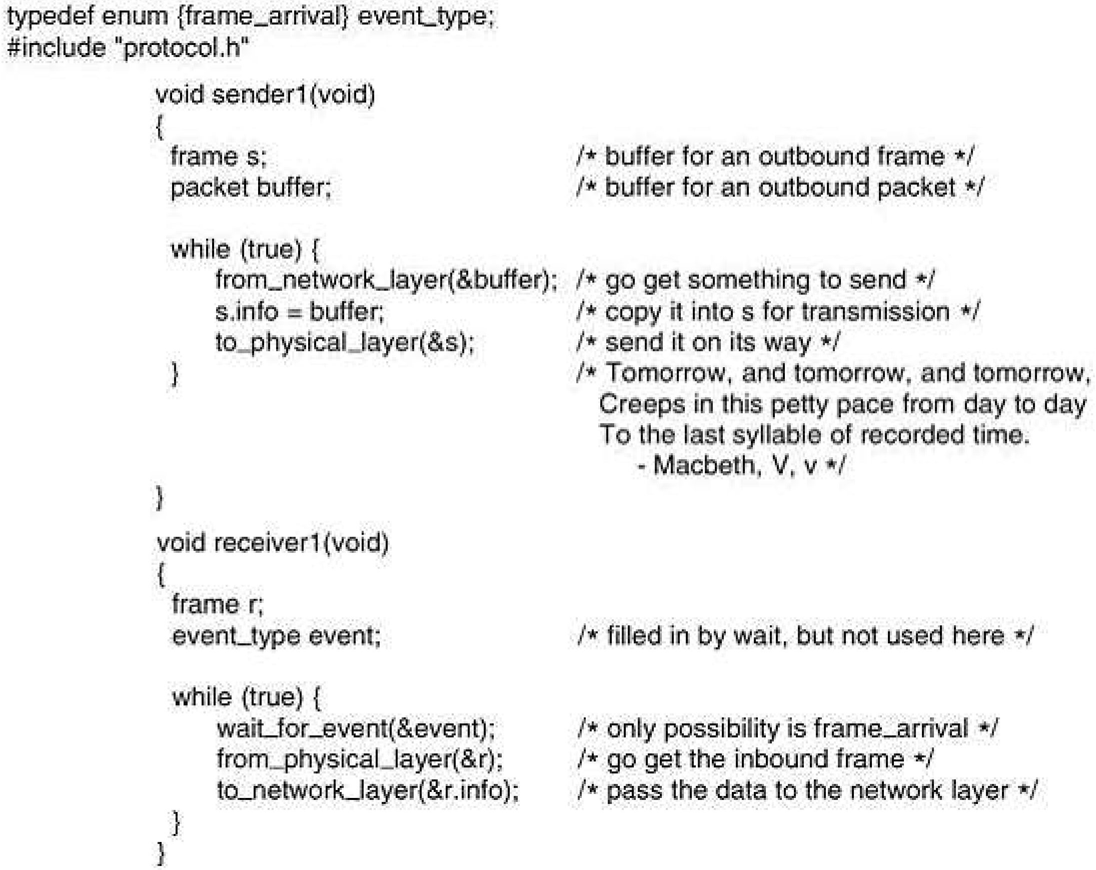
These definitions are located in the file protocol.h.



```
#define MAX_PKT 1024                                /* determines packet size in bytes */

typedef enum {false, true} boolean;                 /* boolean type */
typedef unsigned int seq_nr;                        /* sequence or ack numbers */
typedef struct {unsigned char data[MAX_PKT];} packet;/*   packet definition */
typedef enum {data, ack, nak} frame_kind;           /* frame_kind definition */

typedef struct {                                    /* frames are transported in this layer */
  frame_kind kind;                                  /* what kind of a frame is it? */
  seq_nr seq;                                       /* sequence number */
  seq_nr ack;                                       /* acknowledgement number */
  packet info;                                      /* the network layer packet */
} frame;

/* Wait for an event to happen; return its type in event. */
void wait_for_event(event_type *event);

/* Fetch a packet from the network layer for transmission on the channel. */
void from_network_layer(packet *p);

/* Deliver information from an inbound frame to the network layer. */
void to_network_layer(packet *p);

/* Go get an inbound frame from the physical layer and copy it to r. */
void from_physical_layer(frame *r);

/* Pass the frame to the physical layer for transmission. */
void to_physical_layer(frame *s);

/* Start the clock running and enable the timeout event. */
void start_timer(seq_nr k);

/* Stop the clock and disable the timeout event. */
void stop_timer(seq_nr k);

/* Start an auxiliary timer and enable the ack_timeout event. */
void start_ack_timer(void);

/* Stop the auxiliary timer and disable the ack_timeout event. */
void stop_ack_timer(void);

/* Allow the network layer to cause a network_layer_ready event. */
void enable_network_layer(void);

/* Forbid the network layer from causing a network_layer_ready event. */
void disable_network_layer(void);

/* Macro inc is expanded in-line: Increment k circularly. */
#define inc(k) if (k < MAX_SEQ) k = k + 1; else k = 0
```

An Unrestricted simplex protocol

Data are transmitted in one direction only. Both the transmitting and receiving network layers are always ready. Processing time can be ignored. Infinite buffer space is available. The communication channel between the data link layers never damages or loses frames.



```
typedef enum {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 */
            }                                 /* Tomorrow, and tomorrow, and tomorrow,
                                                 Creeps in this petty pace from day to day
                                                 To the last syllable of recorded time.
                                                      - Macbeth, V, v */
          }

          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 */
            }
          }
```

Fig: An Unrestricted simplex protocol

The protocol consists of two distinct procedures a sender and a receiver. The sender runs in the data link layer of the source machine, and the receiver runs in the data link layer of the destination machine. No sequence numbers or acknowledgements are used here, so MAX_SEQ is not needed. The only event type possible is frame_arrival (i.e., the arrival of an undamaged frame)

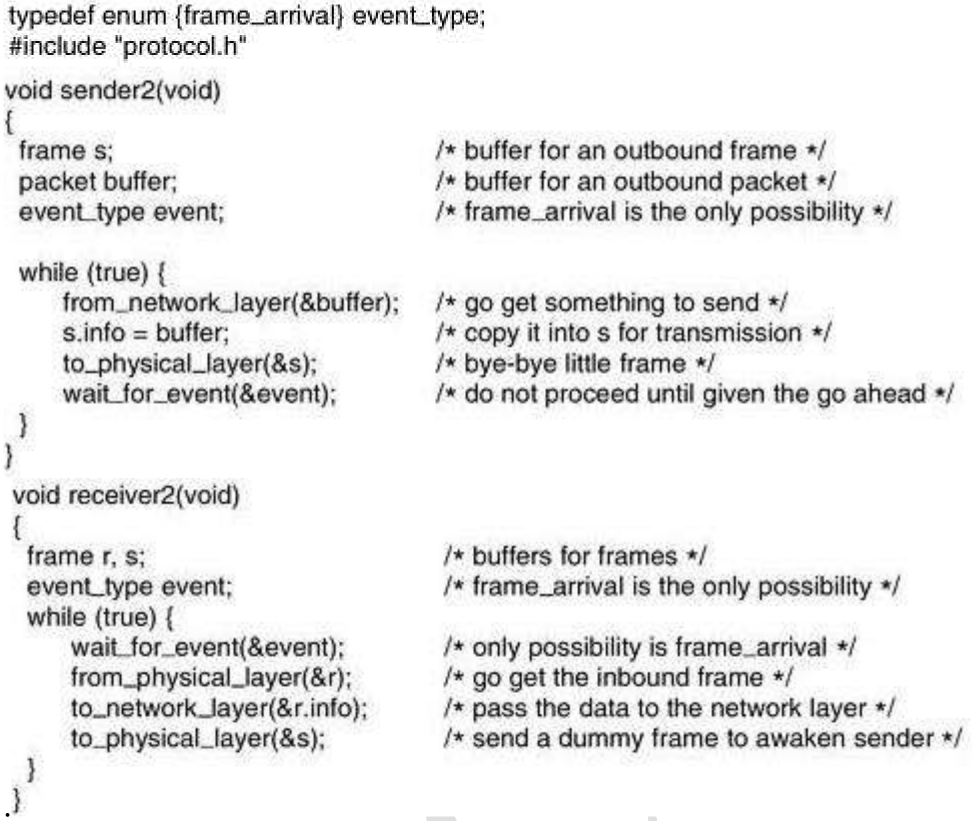
The body of the loop consists of three actions: go fetch a packet from the network layer, construct an outbound frame using the variable s, and send the frame on its way. The receiver is equally simple.

The call to from_physical_layer removes the newly arrived frame from the hardware buffer and puts it in the variable r, where the receiver code can get at it. The data portion is passed on to the network layer and the data link settles back to wait for the next frame, effectively suspending itself until the frame arrives.

A simplex stop and wait protocol

The stop and wait provides for a one-directional flow of data from sender to receiver. The communication channel is once again assumed to be error free. The receiver has only a finite buffer capacity and a finite processing speed, so the protocol must explicit prevent the sender from flooding the receiver with data faster that it can be handled.

The data traffic is simplex. A general solution is to have the receiver provide feedback to the sender. After having passed a packet to its network layer, the receiver sends a little dummy frame back to the sender, which gives the sender permission to transmit the next frame. After having sent a frame, the sender is required by the protocol to bide its time until the little dummy (i.e., acknowledgement) frame arrives



```
 typedef enum {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 */
   }
 }
```

.

Fig: A simplex stop and wait protocol

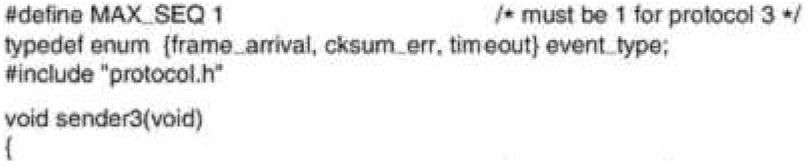
Protocols in which the sender one frame and then waits for an acknowledgement before proceeding are called stop and wait.

The data traffic is simplex, going only from the sender to the receiver, frames do travel in both directions. The sender sends a frame first and then the receiver sends a frame, then the sender sends another frame, then the receiver ends another one, and so on. The incoming frame is always an acknowledgement.

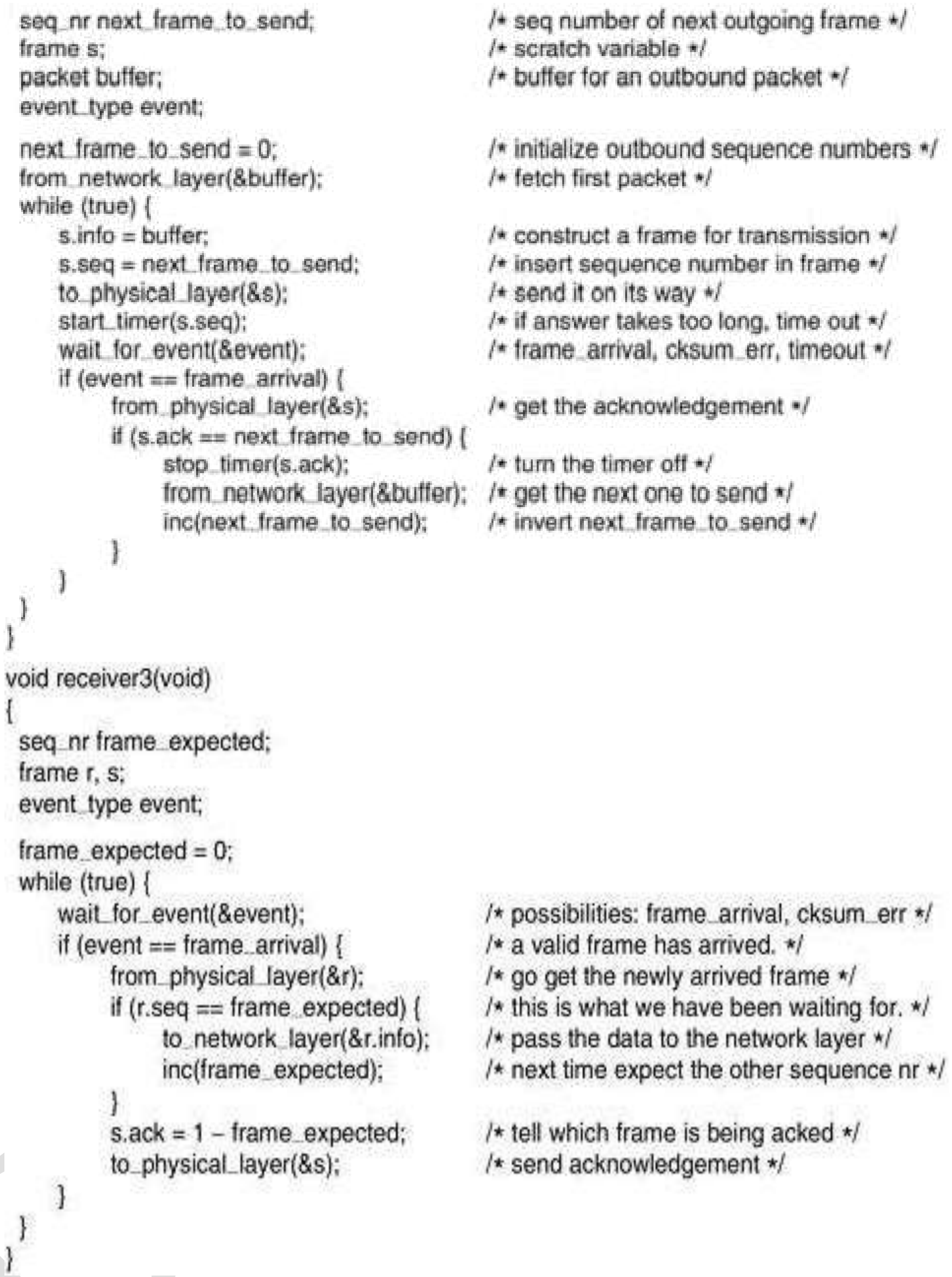
A simplex protocol for a noisy channel

The data link layer processes to provide error_free, transparent communication between network layer processes. The network layer on machine A gives a series of packets to its data link layer, which must

ensure that an identical series of packets are delivered to the network layer on machine B by its data link layer. The network layer on B has no way of knowing that a packet has been lost or duplicated, the data link layer must guarantee that no combination of transmission errors, however can cause a duplicate packet to be delivered to a network layer.



```
#define MAX_SEQ 1                          /* must be 1 for protocol 3 */
typedef enum  {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 */
     }
 }
}
```

Fig: A positive acknowledgement with retransmission protocol

Protocol in which the sender waits for a positive acknowledgement before advancing to the next data item are often called PAR (Positive Acknowledgement with Retransmission) or ARQ (Automatic Repeat reQuest). Transmit data only in one direction.

The sender remembers the sequence number of the next frame to send in next_frame_to_send; the receiver remembers the sequence number of the next frame expected in frame_expected; each protocol has a short initialization phase before entering the infinite loop.

After transmitting a frame and starting the timer, the sender waits for something exciting to happen. Only three possibilities exist: an acknowledgement frame arrives undamaged, a damaged acknowledgement frame staggers in, or the timer expires.

When a valid frame arrives at the receiver, to its sequence number is checked to see if it is a duplicate. If not, it is accepted, passed to the network layer and an acknowledgement are generated. Duplicates and damaged frames are not passed to the network layer.

Discuss about sliding window protocols. (Or) (5 marks)
Explain the one-bit sliding window protocols (Or) (5 marks)
Write a note on protocol using Go back N. (Or) (5 marks)
Describe the selective repeat protocol (5 marks)

Sliding Window Protocols

When a data frame arrives, instead of immediately sending a separate control frame, the receiver itself and waits until the network layer passes it the next packet. The acknowledgement is attached to the outgoing data frame. The acknowledgement gets a free ride on the next outgoing data frame. The technique of temporarily delaying outgoing acknowledgement so they can be hooked onto the next outgoing data frame is known as piggybacking.

How long should the data link layer wait for a packet onto which to piggyback the acknowledgement. If the data link layer waits longer than the sender's timeout period, the frame will be retransmitted, defeating the whole purpose of having acknowledgements.

The bidirectional protocols that belong to a class called sliding window protocols. The essence of all sliding window protocols is that at any instant of time, the sender maintains a set of sequence numbers corresponding to frames it is permitted to send. These frames are said to fall within the sending window. The receiver also maintains a receiving a receiving window corresponding to the set of frames it is permitted to accept.

The sequence numbers within the sender's window represent frames that have been sent or can be sent, but are not acknowledged. Whenever a new packet arrives from the network layer, it is given the next highest sequence number, and the upper edge of the window is advanced by one. When an acknowledgement comes in, the lower edge is advanced by one.

If the maximum window size is n, the sender needs n buffer to hold the unacknowledged frames. If the window ever grows to its maximum size, the sending data link layer must forcibly shut off the network layer until another buffer becomes free.

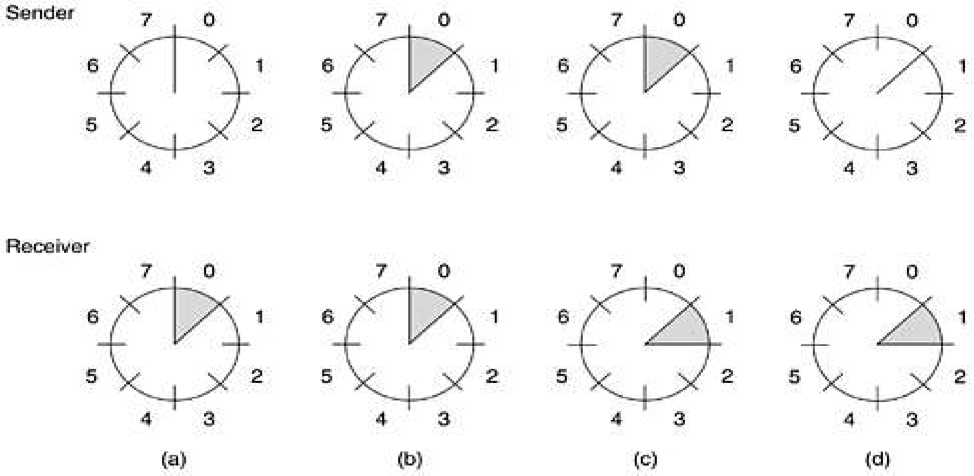


Fig: A sliding window of size 1, with a 3-bit sequence number. (a) Initially. (b) After the first frame has been sent. (c) After the first frame has been received. (d) After the first acknowledgement has been received

The receiving data link layer window corresponding to the frame it may accept. Any frame falling outside the window is discarded without comment. When a frame whose sequence number is equal to the lower edge of the window is received, it is passed to the network layer, an acknowledgement is generated, and the window is rotated by one.

A one-bit sliding window protocol

Assume that computer A is trying to send its frame 0 to computer B and that B is trying to send its frame 0 to A. Suppose that A sends a frame to b but A's timeout interval is short. A may timeout repeatedly, sending a series of identical frames, all with seq=0 and ack=1.

When the first valid frame arrives at computer B, it will be accepted and frame_expected will be set to 1. All the subsequent frames will be rejected because B is expecting frames with sequence number 1, not 0. All the duplicates have ack=1 and B is still waiting for an acknowledgement of 0, B will not fetch a new packet from its network layer.

After every rejected duplicate comes in, B sends A, a frame containing seq=0 and ack=0. One of these arrives correctly at A, causing A to begin sending the next packet. No combination of lost frames or premature timeouts can cause the protocol to deliver duplicate packets to either network layer, to skip a packet, or to deadlock.

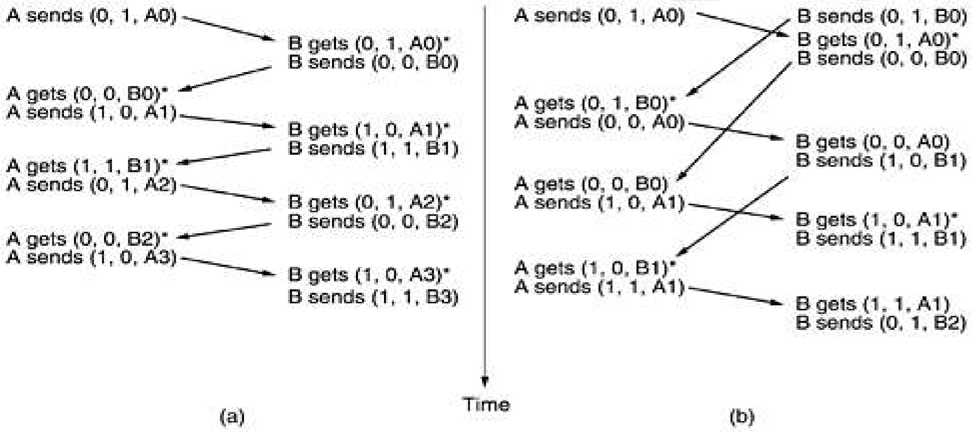


Fig: Two scenarios for protocol 4. (a) Normal case. (b) Abnormal case. The notation is (seq, ack, packet number). An asterisk indicates where a network layer accepts a packet.

In fig (a), the normal operation of the protocol. In fig (b) the peculiarity is illustrated. If B waits for A's first frame before sending one of its own, the sequence is as shown in (a), and every frame is accepted. If A and B simultaneously initiate communication, their first frame cross, and the data link layer then get into situation (b). In (a) each frame arrival brings a new packet for the network layer, there are no duplicates. In (b) half of the frames contain duplicates, even though there are no transmission errors. Similar situations can occur as a result of premature timeouts, even when one side clearly starts first. If multiple premature timeouts occur, frames may be sent three or more times.

A protocol using Go Back N

The product of the pipe and the sender needs the ability to fill it without stopping in order to operate at peak efficiency. This technique is known as pipelining.

Two basic approaches are available for dealing with errors in the presence of pipelining. One way, called Go Back N, is for the receiver simply to discard all subsequent frames, sending no acknowledgements for the discarded frames. The strategy corresponds a receive window of size 1. Which the receiver's window is large, frames 0 and 1 are correctly received and acknowledged frame 1, however is damaged or lost. The sender unaware of this problem, continues to send frames until the timer for frame 2 expires. Then it backs up to frame 2 and starts all over with it, sending 2,3,4, etc., all over again.

The other general strategy for handling errors when frame are pipelined is called selective repeat. When it is used, a bad frame that is received is discarded, but good frames received after it is buffered. When the sender times out, only the oldest unacknowledged frame is retransmitted. If that frame arrives correctly, the receiver can deliver to the network layer, all the frames it has buffered. Selective repeat is often combined with having the receiver send a negative acknowledgement (NAK) when it detects an error. For example, when it receives a checksum error or a frame out of sequence. NAKs stimulate retransmission before the corresponding timer expires and improve performance.

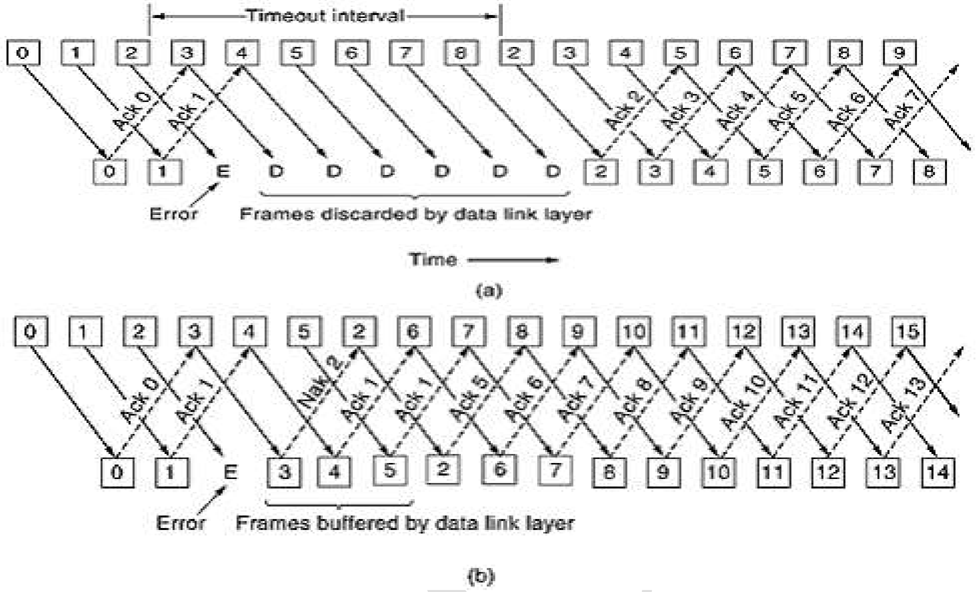


Fig: Pipelining and error recovery. Effect of an error when (a) receiver's window size is 1 and (b) receiver's window size is large

Frames 0 and 1 are again correctly received and acknowledged and frame 2 is lost. When frame 3 arrives at the receiver, the data link layer there notices that is has missed a frame, so it sends back a NAK for 2 but buffers 3. When frames 4 and 5 arrive, they are buffered by the data link layer instead of being passed to the network layer. The NAK 2 gets back to the sender, which immediately resends frame 2. When that arrives, the data link layer now has 2,3,4 and 5 can pass all of them to the network layer in the correct order.

A Protocol using Selective Repeat

Suppose, a 3 bit sequence number, the sender is permitted to transmit up to seven frames before being required to wait for an acknowledgement. Initially, the sender’s and receiver’s windows are shown in fig (a). The sender now transmits frames 0 through 6. The receiver’s window allows it to accept any frame with sequence number between 0 and 6 inclusive. All seven frames arrive correctly, so the receiver acknowledges them and advances its window to allow receipt of 7,0,1,2,3,4 or 5, as shown in fig (b). All seven buffers are marked empty.

The sender eventually times out and retransmits frame 0. When this frame arrives at the receiver, a check is made to see if it falls within the receiver’s window. In fig (b), frame 0 is within the new window, so it will be accepted. The receiver sends a piggybacked acknowledgement for frame 6, since 0 through 6 have been received.

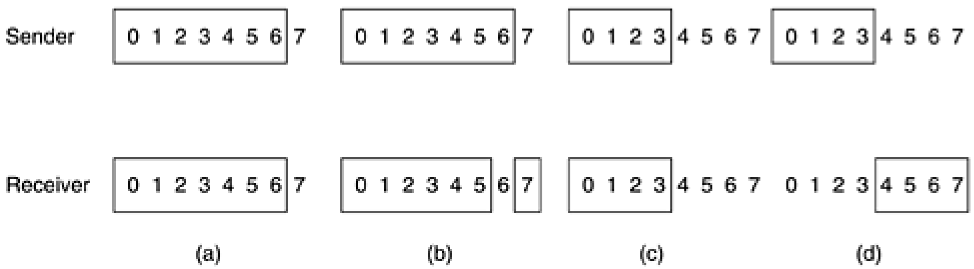


Fig: (a) Initial situation with a window of size seven. (b) After seven frames have been sent and received but not acknowledged. (c) Initial situation with a window size of four. (d) After four frames have been sent and received but not acknowledged.

To ensure that there is no overlap, the maximum window size should be at most half the range of the sequence numbers, as shown in fig (c) and fig (d). For example, if 4 bits are used for sequence numbers, these will range from 0 to 15. Only eight acknowledged frames should be outstanding at any instant. If the receiver has just accepted frames 0 through 7 and advanced its window to permit acceptance of frames 8 through 15, it can unambiguously tell if subsequent frames are retransmissions (0 through 7) or new ones (8 through 15). In general, the window size for protocol 6 will be (MAX_SEQ + 1)/2. For 3-bit sequence numbers, the window size is four.

Describe in detail the finite state machine models. (5 or 10 marks)

Each protocol machine (i.e., sender or receiver) is always on a specific state at every instant of time. Its state consists of all the values of its variables, including the program counter. A large number of states can be grouped for purposes of analysis.

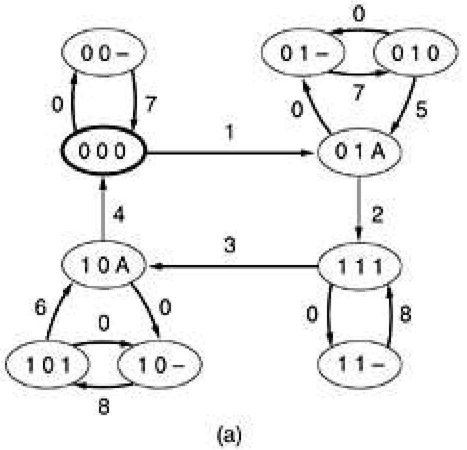
For each state, there are zero or more possible transitions to other states. Transitions occur when some event happens. For a protocol machine, a transition might occur when a frame is sent, when a frame arrives, when a timer expires, when an interrupt occurs, etc. For the channel, typical events are insertion of a new frame onto the channel by a protocol machine, machine, delivery of a frame to a protocol machine, or loss of a frame due to noise.

From the initial state, some all of the other states can be reached by a sequence of transitions. Using well-known techniques from graph theory (e.g., computing the transitive closure of a graph), it is possible to determine which states are reachable and which are not. This technique is called reachability analysis.

A finite state machine model of a protocol can be regarded as a quadruple (S, M, I, T) where:

S is the set of states the processes and channel can be in.
M is the set of frames that can be exchanged over the channel.
I is the set of initial states of the processes.
T is the set of transitions between states.

As an example of a finite state machine model, consider fig(a).



Transition	Who runs?	Frame accepted	Frame emitted	To network layer
0	–	(frame lost)		–
1	R	0	A	Yes
2	S	A	1	–
3	R	1	A	Yes
4	S	A	0	–
5	R	0	A	No
6	R	1	A	No
7	S	(timeout)	0	–
8	S	(timeout)	1	–

(b)

Fig: (a) State diagram for protocol 3. (b) Transitions

This graph corresponds to protocol 3. Each protocol machine has two states and the channel has four states. A total of 16 states exist, not all of them reachable from the initial one. The unreachable ones are not shown in the figure. Checksum errors are also ignored here for simplicity.

Each state is labeled by three characters, SRC, where S is 0 or 1, corresponding to the frame the sender is trying to send, R is also 0 or 1, corresponding to the frame the receiver expects, and C is 0,1,A or empty (-), corresponding to the state of the channel. In this example the initial state has been chosen as (000) (SRC).

Nine kinds of transitions are shown in the fig (b). Transition 0 consists of the channel losing its contents. Transition 1 consists of the channel correctly delivery packet 0 to the receiver, with the receiver then changing its state to expect frame 1 and emitting an acknowledgement. Transition 1 also corresponds to the receiver delivery packet 0 to the network layer.

During normal operation, transitions 1, 2, 3, and 4 are repeated in order over and over. In each cycle, two packets are delivered, bringing the sender back to the initial state to trying to send a new frame with sequence number 0. If the channel loses frame 0, it makes a transition from state (000) to state (00-). The sender times out (transition 7) and the system moves back to (000). The lost of an acknowledgement is more complicated, required two transitions, 7 and 5, or 8 and 6, to repair the damage.

Discuss in detail Petri net models. (5 or 10 marks)

A Petri net has four basic elements: places, transitions, arcs and tokens. A place represents a state which the system may be in. A Petri net with two places, A and B, both as circles. The system is currently in state A, indicated by the token in place A. A transition is indicated by a horizontal or vertical bar. Each transition has zero or more input arcs coming from its input places and zero or more output arcs, going to its output places.

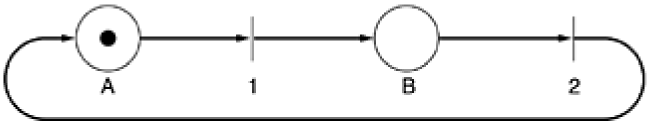


Fig: A Petri net with two places and two transitions.

A transition is enabled if there is at least one input token in each of its input places. Any enabled transition may fire, removing one token from each input place and depositing a token in each output place. If the number of input arcs and output arcs differ, token will not be conserved. If two or more transitions are enabled, any one of them may fire. It can be used to model any two-phase process.

The Petri net model has the sender's state, channel state and receiver's state are represented separately. Transitions 1 and 2 correspond to transmission of frame 0 by the sender, and on a timeout respectively. Transition 3 and 4 are analogous for frame 1. Transition 5, 6 and 7 correspond to the loss of frame 0, an acknowledgement and frame 1, respectively. Transitions 8 and 9 occur when a data frame with the wrong sequence number arrives at the receiver of the next frame in sequence and its delivery to the network layer.

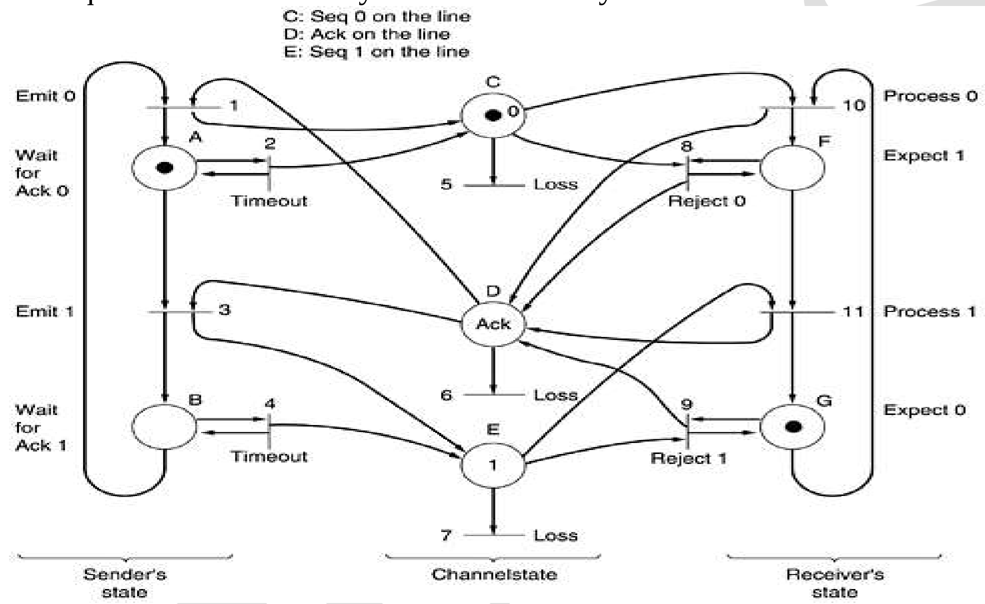


Fig: A Petri net model for protocol 3

Petri nets can be represented in convenient algebraic form resembling a grammar. Each transition contributes one rule to the grammar. Each rule specifies the input and output places of the transition. It has 11 transitions; its grammar has 11 rules, numbered 1-11, each one corresponding to the transition with the same number. The grammar for the Petri net as follows:

1. BD-> AC
2. A-> A
3. AD->BE
4. B->B
5. C->
6. D->
7. E->
8. CF-> DF
9. EG->DG
10. CG->DF
11. EF->DG

Discuss in detail Bluetooth and its Architecture. (10 marks)

Bluetooth Architecture

The basic unit of a Bluetooth system is a pico net, which consists of a master node and up to seven active slave nodes within a distance of 10 meters. Multiple pico nets can exist in the same room and can even be connected via a bridge node. An interconnected collection of pico nets is called a scatter net.

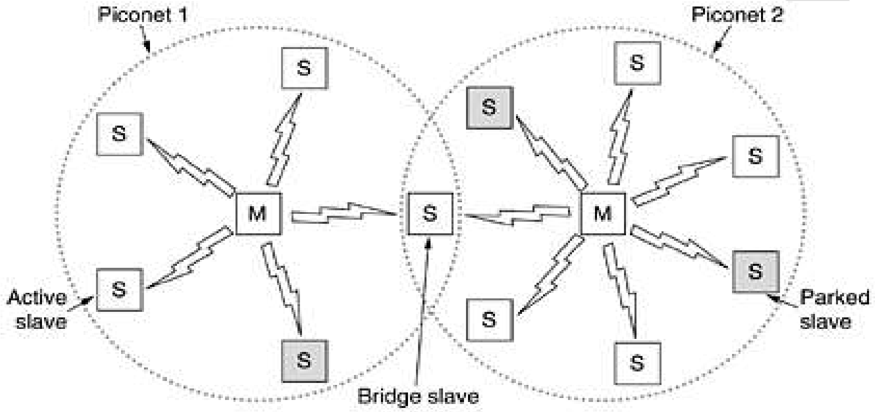


Fig: Two piconets can be connected to form a scatternet

There can be up to 255 parked nodes in the net. These are devices that the master has switched to a lower-power state to reduce the drain on their batteries. In parked state, a device cannot do anything except respond to an activation or beacon signal from the master. There are also two intermediate power states, hold and sniff.

A pico net is a centralized TDM system, with the master controlling the clock and determining which device gets to communicate in which time slot. All communication is between the master and a slave, direct slave-slave communication is not possible.

Bluetooth Applications

The 13 applications are called profiles. They are listed below:

Name	**Description**
Generic Access	Procedures for link management
Services discovery	Protocol for discovering offered services
Serial port	Replacement for a serial port cable
Generic Object Exchange	Defines Client-server relationship for object movement
LAN Access	Protocol between a mobile computer and a fixed LAN
Dial-up networking	Allows a notebook computer to call via a mobile phone
Fax	Allows a mobile fax machine to talk to q mobile phone
Cordless Telephony	Connects a handset and its local base station
Intercom	Digital Walkie-Talkie
Headset	Allows hands-free voice communication
Object Push	Provides a way to exchange simple objects
File Transfer	Provides a more general file transfer facility
Synchronization	Permits a PDA to synchronize with another computer

- The generic access profile is to provide a way to establish and maintain secure links (channels) between the master and the slave.

- The generic is the service discovery profile, which is used by devices to discover what services other devices have to offer.
- The serial port is a transport protocol and is especially useful for legacy applications that expect a serial line.
- The generic object exchange profile defines a client-server relationship for moving data around.
- The next group of three profiles is for networking.
    1) The LAN access profile allows a Bluetooth device to connect to a fixed network.
    2) The dial-up networking profile allows a notebook computer to connect to a mobile phone containing a built-in modem without wires.
    3) The fax profile allows wireless fax machines to send and receive faxes using mobile phones without a wire between the two.
- The next three profiles are for telephony.

1) The cordless telephony profile provides a way to connect the handset of a cordless telephone to the base station.
2) In Intercom profile allows two telephones to connect as walkie-talkies.
3) The headset profile provides hands-free voice communication between the headset and its base station.

- The next three profiles are for actually exchanging objects between two wireless devices. These could be business cards, pictures, or data files.

The Bluetooth Protocol Stack

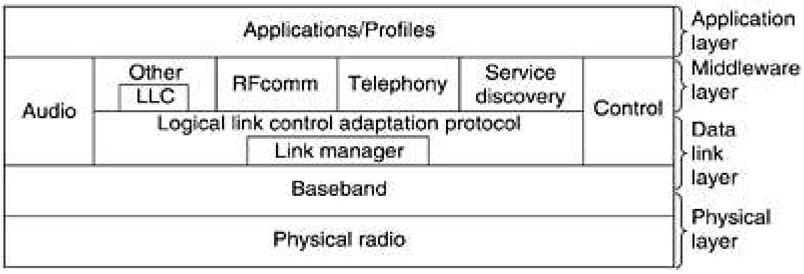


Fig: The 802.15 version of the Bluetooth protocol architecture

The Bluetooth protocol stack is:

- The bottom layer is the physical radio layer, which corresponds fairly well to the physical layer in the OSI and 802 models. It deals with radio transmission and modulation.
- The baseband layer is somewhat analogous to the MAC sublayers but also includes elements of the physical layer. It deals with how the master controls time slots and how these slots are grouped into frames.
- The link manager handles the establishment of logical channels between devices, including power management, authentication and quality of service.
- The logical link control adaptation protocol shields the upper layers from the details of transmission.
- The audio and control protocols deal with audio and control.
- The middleware layer, which contains a mix of different protocols. The 802 LLC was inserted by IEEE for compatibility with its other 802 networks. The RFComm, telephony and service discovery protocols are native. The RFComm (Radio Frequency Communication) is the protocol that emulates the standard serial port found on PC's for connecting the keyboard, mouse and modem among other devices.

- The telephony protocol is a real-time protocol used for the three speech-oriented profiles. It also manages call setup and termination.
- The service discovery protocol is used to locate services within the network.
- The applications profile to make use of the protocols in lower layers to get their work done.

The Bluetooth Radio Layer

The radio layer moves the bits from master to slave or vice-versa. It is a low-power system with a range of 10 meters operating in the 2.4GHz ISM band. The band is divided into 79 channels of 1 MHz each.

Both 802.11 and Bluetooth operate in the 2.4 GHz ISM band on the same 79 channels, they interface with each other. Both are IEEE standards..

The Bluetooth Baseband Layer

The master in each piconet defines a series of 625 µsec time slots, with the master's transmissions starting in the even slots and the slave's transmission starting in the odd ones. Frames can be 1, 3 or 5 slots long.

For a single-slot frame, 366 of the 625 bits are left over. 126 are for an access code and the header, learning 240 bits for data. When five slots are stung together, only one setting period is needed and a slightly shorter setting period is used, the 5*625=3125 bits in five time slots, 2781 are available to the baseband layer.

Each frame is transmission over a logical channel called a link between the master and a salve. Two kinds of links are ACL (Asynchronous Connection Oriented) and SCO (Synchronous Connection Oriented).

The ACL is used for packet-switched data available at irregular intervals. These data come from the L2CAP layer on the sending side and are delivered to the L2CAP layer on the receiving side. Frames can be lost and may have to be retransmitted.

The SCO link for real time data, such as telephone connections. This type of channel is allocated a fixed slot in each direction. Frames sent over them are never retransmitted. A slave may have up to three SCO links with its master. Each SCO link can transmit one 64,000 bps PCM audio channel.

The Bluetooth L2CAP layer

The L2CAP layer has three major functions.

1) It accepts packets of up to 64 KB from the upper layers and breaks them into frames for transmission. At the far end, the frames are reassembled into packets again.

2) It handles the multiplexing and de-multiplexing of multiple packet sources. When a packet has been reassembled, the L2CAP layer determines which upper-layer protocol to hand it to. For example, RFComm or telephony.

3) L2CAP handles the quality of service requirements, both when links are established and during normal operation.

The Bluetooth frame Structure

The structure begins with an access code that usually identifies the master so that slaves within radio range of two masters can tell which traffic for them. Next comes a 54-bit header containing typical MAC sublayer fields. Then comes the data field, of up to 2744 bits (for a five-slot transmission). For a single time slot, the format is the same except that the data field is 240 bits.

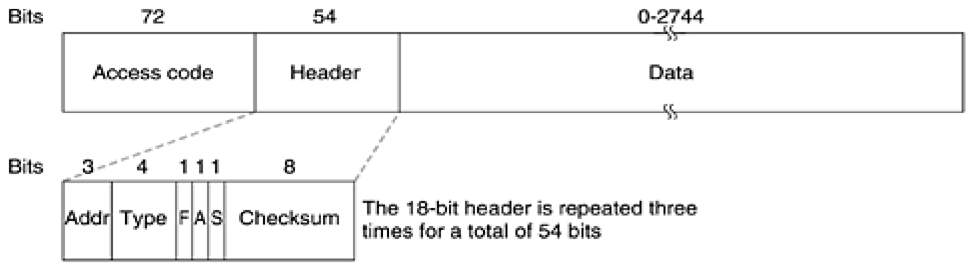


Fig: A typical Bluetooth data frame

The header has the address field, type field, flow bit, acknowledgement bit, sequence bit and checksum.

The address field identifies which of the eight active devices the frame is intended for.

The type field identifies the frame type (ACL, SCO, Poll or null), the type of error correction used in the data field and how many slots long the frame is.

The flow bit is asserted by a slave when its buffer is full and cannot receive any more data. This is a primitive form of flow control.

The acknowledgement bit is used to piggybacking an ACK onto a frame.

The sequence bit is used to number the frames to detect retransmissions. The protocol is stop-and-wait, so 1 bit is enough.

The 8-bit is header checksum.

The entire 18-bit header is repeated three times to form the 54-bit header.

Explain the Local Internetworking. (5 Marks)

A transparent bridge operates in promiscuous mode, accepting every frame transmitted on all the LANs to which it is attached. For example, Bridge B1 is connected to LANs 1 and 2, and Bridge B2 is connected to LANs 2, 3 and 4. A frame arriving at bridge B1 on LAN1 destined for A can be discarded immediately, because it is already on the correct LAN, but a frame arriving on LAN1 for C or F must be forwarded.

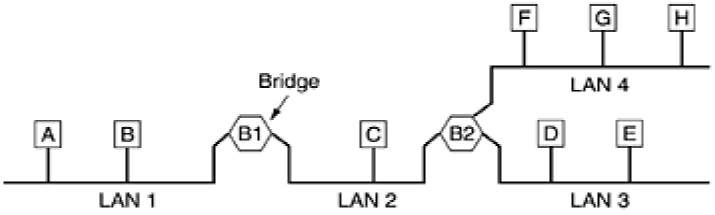


Fig: A configuration with four LANs and two bridges

When a frame arrives, a bridge must decide whether to discard or forward it, and which LAN to put the frame. This decision is made by looking up the destination address in a big (hash) table inside the bridge. The table can list each possible destination and tell which output line (LAN) it belongs on.

When the bridges are first plugged in, all the hash tables are empty. None of the bridge knows where any of the destinations are, so they use a flooding algorithm: every incoming frame for an unknown destination is output on all the LANs to which the bridge is connected except the one it arrived on.

Once a destination is known, frames destined for it are put on only the proper LAN and are not flooded. The algorithm used by the transparent bridges is backward learning. The bridges operate in promiscuous mode, every frame sent on any of their LANs. At the source address, they can tell which machine is accessible on which LAN.

The routing procedure for n incoming frame depends on the LAN it arrives on (the source LAN) and the LAN its destination is on (the destination LAN) as follows:

1) If destination and source LANs are the same, discard the frame.
2) If the destination and source LANs are different, forward the frame.
3) If the destination LAN is unknown, use flooding.

Explain the Spanning Tree Bridges. (5 Marks)

To increase reliability, some sites use two or more bridges in parallel between pairs of LANs. A simple example of these problems can be seen by observing how a frame F, with unknown destination is handled. Each bridge following the normal rules for handling unknown destinations, uses flooding, which means just copying it to LAN2. Bridge 1 sees F2, a frame with an unknown destination, which it copies to LAN1, generating F3 (not shown). Bridge 2 copies F1 to LAN1 generating F4 (not shown). Bridge 1 now towards F4 and bridge 2 copies F3. This cycle goes on forever.

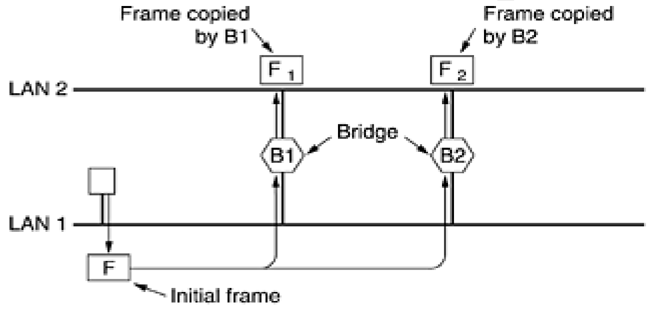


Fig: Two parallel transparent bridges

We see nine LANs interconnected by ten bridges. This Configuration can be abstracted into a graph with LANs as the nodes. An arc connects any two LANs that are connected by a bridge. The graph can be reduced to a spanning tree by dropping the arcs shown as dotted lines.

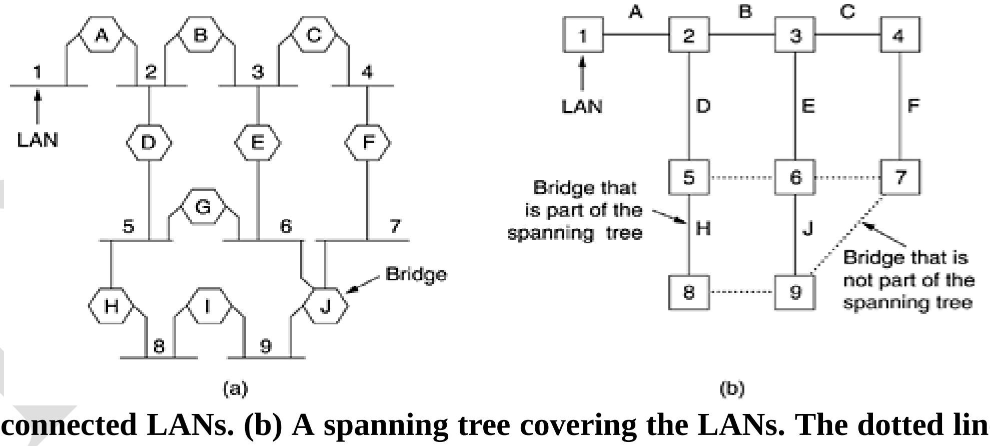

A
B
C
1
2
3
4
LAN
D
E
F
G
5
6
7
H
I
J
Bridge
8
9
(a)
A
B
C
1
2
3
4
LAN
D
E
F
5
6
7
Bridge that is part of the spanning tree
H
J
Bridge that is not part of the spanning tree
8
9
(b)

Fig: (a) Interconnected LANs. (b) A spanning tree covering the LANs. The dotted lines are not part of the spanning tree.

Using this spanning tree, there is exactly one path from every LAN to every other LAN. Once the bridges have agreed on the spanning tree, all forwarding between LANs follows the spanning tree. There is a unique path from each source to each destination, loops are impossible.

To build the spanning tree, first the bridges have to choose one bridge to be the root of the tree. They make this choice by having each one broadcast its serial number, installed by the manufacturer and guaranteed to be unique worldwide. The bridge with the lowest serial number becomes the root. A tree of shortest paths from the root to every bridge and LAN is constructed.

This tree is the spanning tree. If a bridge or LAN fails, a new one is computed. The result of this algorithm is that a unique path is established from every LAN to the root and thus to every other LAN.

Explain the Remote Bridges. (5 Marks)

A bridge is to connect two (or more) distant LANs. For example, a company might have plants in several cities, each with its own LAN. All the LANs should be interconnected, the complete system acts like one large LAN.

This goal can be achieved by putting a bridge on each LAN and connecting the bridges pair wise with point-to-point lines (eg, lines leased from a telephone company). Simple systems, with three LANs are shown below:

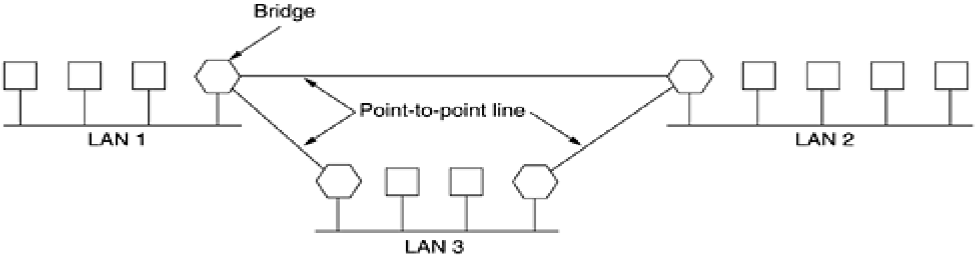


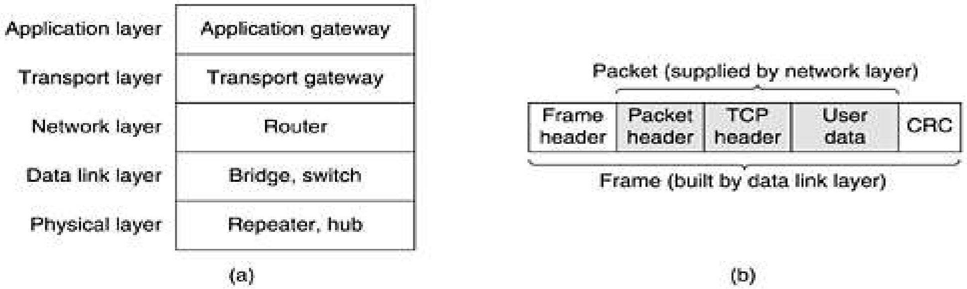
Fig: Remote bridges can be used to interconnect distant LANs

A normal system of six LAN interconnected by four bridges. Various protocols can be used on the point-to-point lines. One possibility is to choose some standard point-to-point data link protocol such as PPP, putting complete MAC frames in the payload field. If LANs are identical, and the only problem is getting frames to the correct LAN. A new MAC header and trailer can then be generated at the destination bridge. A disadvantage of this approach is that the checksum that arrives, at the destination host is not the one computed by the source host, so errors caused by bad hits in a bridge's memory may not be detected.

Explain the various Common Devices in networks. (Or) (5 Marks)

Write short notes on Repeaters, Hubs, Bridges, Switches, Routers and Gateways. (5 Marks)

The Repeaters, Hubs, Bridges, Switches, Routers and Gateways are devices operate in different layers as shown:



Application layer	Application gateway
Transport layer	Transport gateway
Network layer	Router
Data link layer	Bridge, switch
Physical layer	Repeater, hub

(a)
Packet (supplied by network layer)
Frame header
Packet header
TCP header
User data
CRC
Frame (built by data link layer)
(b)

Fig: (a) Which device is in which layer. (b) Frames, packets, and headers

The user generates some data to be sent to remote machine. Those data are passed to the transport layer, which then adds a header. Then the packet goes to the data link layer, which adds its own header and checksum (CRC) and gives the resulting frame to the physical layer for transmission. For example, over a LAN.

As the bottom, in the physical layer, to find the repeaters. These are analog devices that are connected to two cable segments. A signal appearing on one of them is amplified and put out on the other. Repeaters do not understand frames, packets or headers. They understand volts.

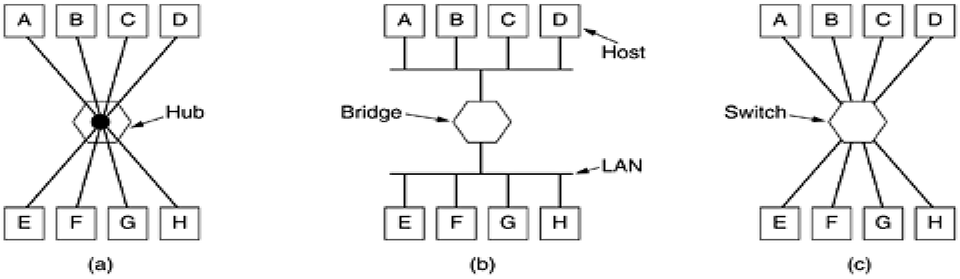


Fig: (a) A hub. (b) A bridge. (c) A switch.

A hub has a number of input lines that it joins electrically. Frames arriving on any of the lines are sent out on all the others. If two frames arrive at the same time, they will collide, just as on a coaxial cable. All the lines coming into hub must operate at the same speed.

A bridge connects two or more LANs. When a frame arrives, software in the bridge extracts the destination address from the frame header and looks it up in a table to see where to send the frame.

Switches are similar to bridges in that both route on frame addresses. The main difference is that a switch is most often to connect individual computers. The switch must actively forward the frame from A to B because there is no other way for the frame to get there. Each switch port usually goes to a single computer. Each port is its own collision domain; switches never lose frames to collisions. These switches do not use store-and-forward switching. They are referred to as cut-through switches.

When a packet comes into a router, the frame header and trailer are stripped off and the packet located in the frame's payload field is passed to the routing software. This software uses the packet header to choose an output file. For an IP packet, the packet header will contain a 32-bit (IPV4) or 128-bit (IPV6) address, but not a 48-bit 802 address. The routing software does not see the frame addresses and does not even know whether the packet came in on a LAN or a point- to-point line.

Application gateways understand the format and contents of the data and translate message from one format to another. For example, an e-mail gateway could translate internet message into SMS message for mobile phones.

End of UNIT -III

UNIT – IV

Describe the design layer in Network Layer. (10 Marks)

1) Store – and - forward Switching:

The major components of the system are the carrier's equipment (routers connected by transmission lines), shown inside the shaded oval and the customers' equipment, shown outside the oval. Host H1 is directly connected to one of the carrier's routers, A, by a leased line. H2 is on a LAN with a router, F, owned and operated by the customer. We have shown F as being outside the oval because it does not belong to the carrier, but in terms of construction, software and protocols, it is probably no different from the carrier's routers.

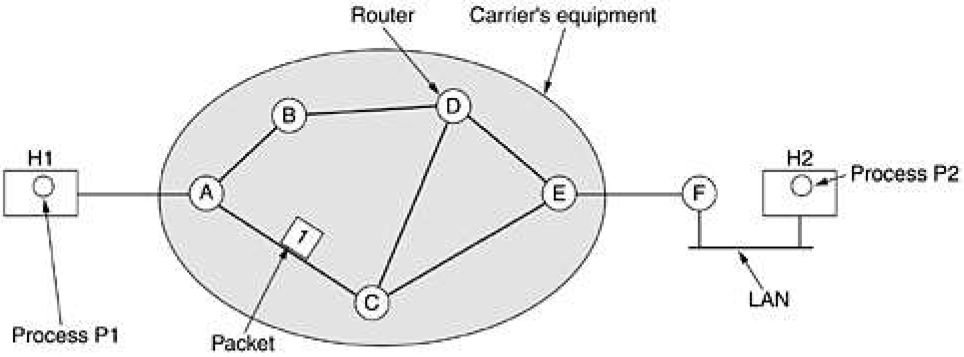


Fig: The environment of the network layer protocols

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.

2) Services provided to the transport layer:

The network layer provides services to the transport layer at the network layer / transport layer interface. The network layer services have been designed with the following:

1) The services should be independent of the router technology.

2) The transport layer should be shielded from the number, type and topology of the routers present.

3) The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

One camp (represented by the Internet Community) argues that the routers job is moving packets around and nothing else. The subnet is inherently unreliable, no matter how it is designed. The host should accept the fact the network is unreliable and do errors control (i.e. error detection and corrections) and flow control themselves.

The other camp (represented by the telephone companies) argues that the subnet should provide a reliable, connection-oriented service. In this view, quality of service is the dominant factor and without connections in the subnet, quality of service is very difficult to achieve, especially for real-time traffic such as voice and video.

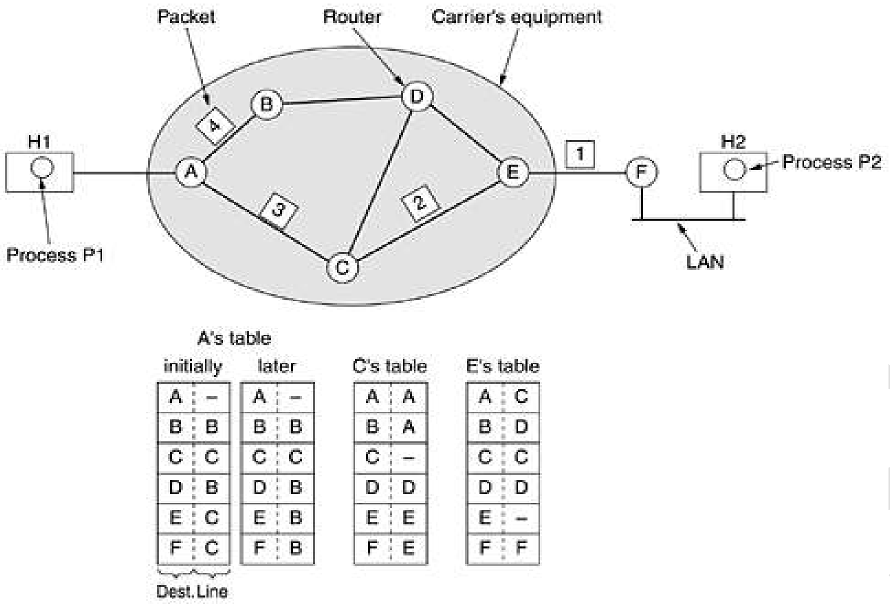
The Internet offers connectionless network-layer service and ATM network offers connection-oriented network-layer service.

3) Implementation of connectionless service:

The types of service are: 1) Connectionless and Connection-oriented.

If connectionless service is offered, packets are injected into the subnet individually and rounded independently of each other. No advance setup is needed. The packets are frequently called datagram (in analogy with telegrams) 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), in analogy with the physical circuits set up by the telephone system and the subnet is called a virtual circuit subnet.



Dest.	A's table initially	A's table later	C's table	E's table
A	–	–	A	C
B	B	B	A	D
C	C	C	–	C
D	B	B	D	D
E	C	B	E	–
F	C	B	E	F

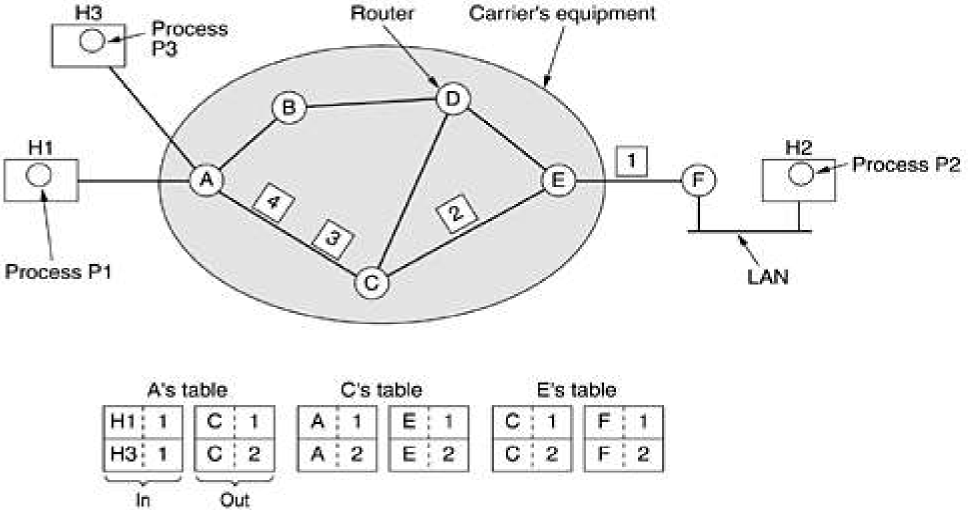
Fig: Routing within a datagram subnet.

How a datagram subnet works. Suppose that the process P1 in the above diagram has a long message for P2. It hands the message to the transport layer with instructions to deliver it to process P2 on host H2. the transport layer code runs on H1, typically within the operating system.

Let us assume that the message is four times longer than the maximum packet size, the network layer has to break it into four packets 1,2,3 and 4 and sends each of them in turn to route A using some point-to-point protocol, for example, PPP. At this point the carrier takes over. Every router has an internal table telling it where to send packets for each possible destination. Each table entry is a pair consisting of a destination and the outgoing line to use for that destination. Only directly-connected lines can be used. The algorithm that manages the tables and makes the routing decisions is called the routing algorithm.

4) Implementation of connection-oriented service:

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 following over the connection, exactly the same way that the telephone system works. When the connection is released, the virtual circuit is also terminated. With connection-oriented service, each packet carries an identifies telling which virtual circuit it belongs to.



A's table In	A's table Out	C's table In	C's table Out	E's table In	E's table Out
H1 1	C 1	A 1	E 1	C 1	F 1
H3 1	C 2	A 2	E 2	C 2	F 2

Fig: Routing within a virtual-circuit subnet.

As an example, host H1 has established connection 1 with host H2. The first line of A's table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1. The first entry at C routes the packet to E, also with connection identifier 1.

Let us consider, what happens if H3 also wants to establish a connection to H2. It chooses connection identifier 1 and tells the subnet to establish the virtual circuit. This leads to the second row in the tables. We have a conflict here because although A can easily distinguish connection 1 packets from H1 from connection 1 packet from H3, C cannot do this. For this reason, A assigns a different connection identifier to the outgoing traffic for the second connection. This is called label switching.

5) 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 VC that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated for each VC.
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC.

Explain the routing algorithms. (5 marks)

The function of the network layer is routing packets from the source machine to the destination machine.

Routing algorithms can be grouped into two major classes: non-adaptive and adaptive.

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

Adaptive algorithms change their routing decisions to reflect changes in the topology, and usually the traffic as well. Adaptive algorithms differ in where they get their information (e.g., locally from adjacent routers, or from all routers), when they change the routes (e.g., every ΔT sec, when the load changes or when the topology changes) and what metric is used for optimization (e.g., distance, number of hops, or estimated transit time). This procedure is called dynamic routing.

Explain the optimality principle. (5 Marks)

One can make a general statement about optional routes without regard to network topology as traffic. This statement is known as the optimality principle. It states that 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.

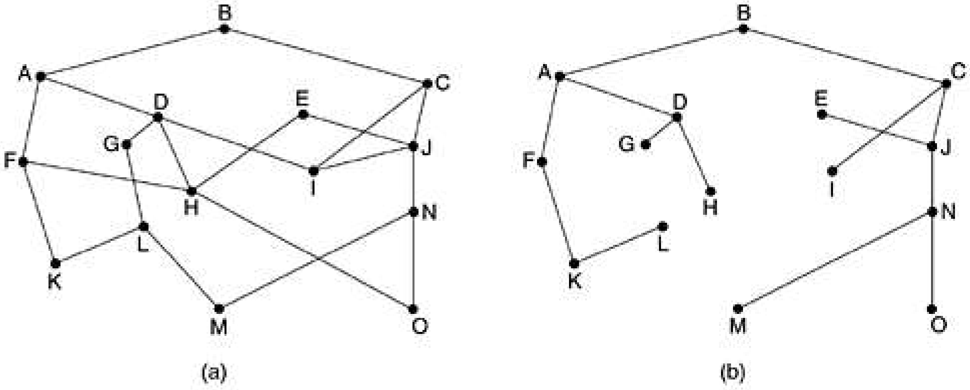


Fig: (a) A subnet. (b) A sink tree for router B

The set of optimal routes from all sources to a given destination from a tree rooted at the destination. Such a tree is called a sink tree. Where the distance metric is the number of hops. The goal of all routing algorithms is to discover and use the sink trees for all routers.

Since a sink tree is indeed a tree; it does not contain any loops, so each packet will be delivered within a finite and bounded number of hops.

Explain the shortest path routing. (5 marks)

To build a graph of the subnet, with each node of the graph representing a router and each arc of the graph is representing a communication line (often called a line). To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.

To illustrate how the labeling algorithm works, look at the weighted, undirected graph, where the weights represent, for example, distance, we want to find the shortest path from A to D. We start out by marking node A as permanent, indicated by a filled-in circle.

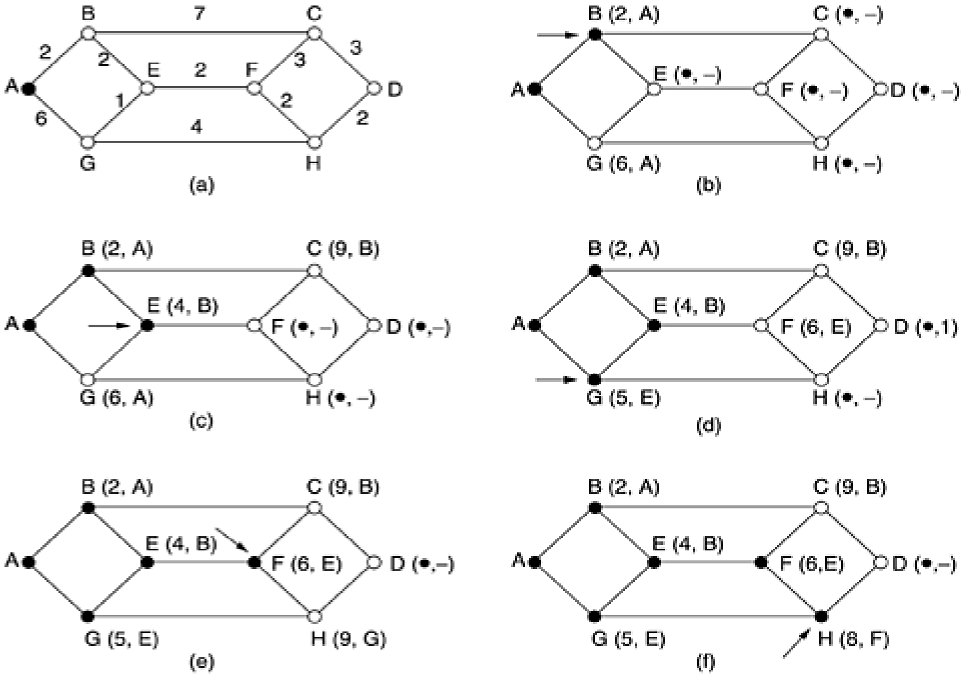


Fig: The first five steps used in computing the shortest path from A to D. The arrows indicate the working node.

Each of the nodes adjacent to A (the working node), relabeling each one with the distance to A. Whenever a node is relabeled, we also label it with the node from which the probe was made so that we can reconstruct the final path later. Having examined each of the nodes adjacent to A, we examine all the tentatively labeled nodes in the whole graph and make the one with the smallest label permanent, as shown in fig (b). This one becomes the new working node.

We now start at B and examine all nodes adjacent to it. If the sum of the label on B and the distance from B to the node being considered is less than the label on that node, we have a shorter path, so the node is relabeled.

After all the nodes adjacent to the working node have been inspected and the tentative labels changed if possible, the entire graph is searched for the tentatively-labeled node with the smallest value. This node is made permanent and becomes the working node for the next round.

Explain the flooding. (5 marks)

The static algorithm is flooding, in which every incoming packet is sent out on every outgoing line except the one it arrived on. Flooding obviously generates vast numbers of duplicate packets. 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.

To achieve this goal is to have the source router put a sequence number in each packet it receives from its hosts. Each router then needs a list per source router telling which sequence numbers originating at that source have already been seen. If an incoming packet is on the list, it is not flooded.

For example, it can be used in military applications, distributed applications and wireless networks.

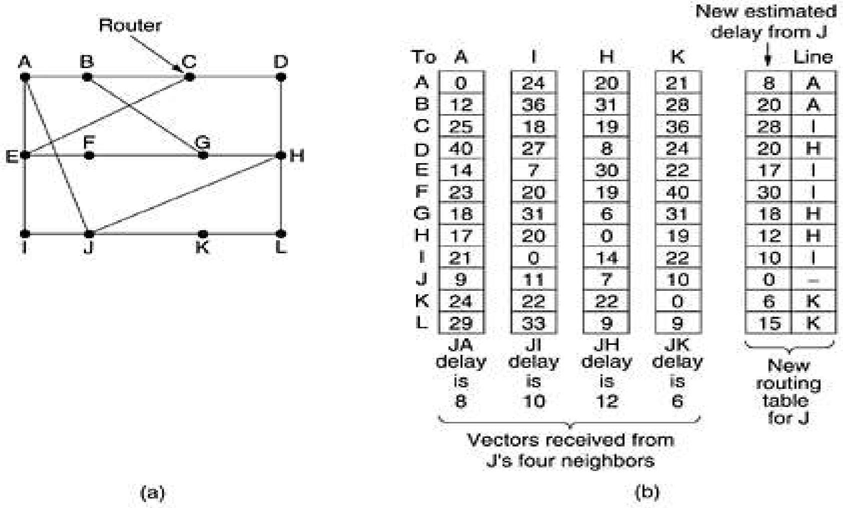
Discuss the distance vector routing. (5 marks)

Distance vector routing algorithm 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.

The distance vector routing algorithm is sometimes called the distributed Bellman-Ford routing algorithm and the Ford-Fulkerson algorithm. It was the original ARPANET routing algorithm.

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.

The router is assumed to know the "distance" to each of its neighbors. If the metric is hops, the distance is just one hop. If the metric is queue length, the router simply examines each queue. If the metric is delay, the router can measure it directly with special ECHO packets that the receiver just timestamps and sends back as fast as it can.



To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

Vectors received from J's four neighbors

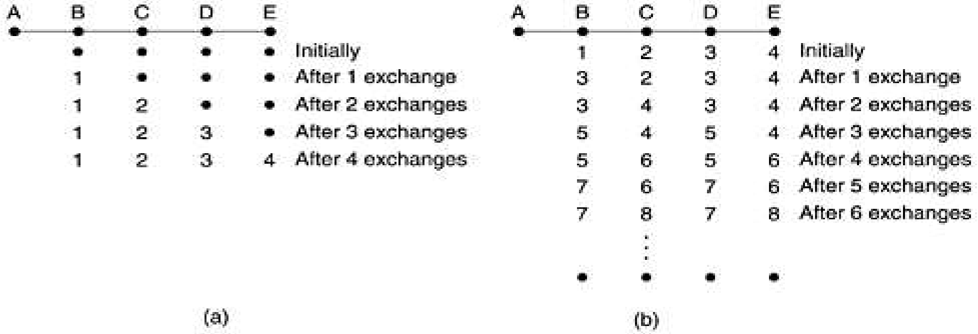
Fig: (a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.

The fig(a) shows a subnet. The first four columns of part (b) show the delay vector received from the neighbors of router J. A claims to have 12-msec delay to B, a 25-msec delay to C, a 40-msec delay to D, etc. 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.

Consider how J computes its new route to router G. It knows that it can get to A in 8-msec, and A claims to be able to get to G in 18-msec, so J knows it can count on a delay of 26-msec to G if it forwards packets bound for G to A. It computes the delay to G via I, H and K as 41 (31+10), 18 (6+12) and 37 (31+6) msec respectively. The best of these values is 18, so it makes an entry in its routing table that the delay to G is 18-msec and that the route to use is via H. The same calculation is performed for all the other destinations, with the new routing table.

The Count-to-Infinity Problem

Consider the five-node (linear) subnet of fig. where the delay metric is the number of hops. Suppose A is down initially and all the other routers know this. In other words, they have all recorded the delay to A as infinity.



A	B	C	D	E	
	•	•	•	•	Initially
	1	•	•	•	After 1 exchange
	1	2	•	•	After 2 exchanges
	1	2	3	•	After 3 exchanges
	1	2	3	4	After 4 exchanges

(a)

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			
	•	•	•	•	

(b)

Fig: The count-to-infinity problem

When A comes up, the other routers learn about it via the vector exchanges.

Let us consider the situation of fig(b), in which all the lines and routers are initially up. Routers B, C, D, and E have distances to A of 1, 2, 3, and 4, respectively. Suddenly A goes down, or alternatively, the line between A and B is cut, which is effectively the same thing from B's point of view.

At the first packet exchange, B does not hear anything from A. Fortunately, C says: Do not worry; I have a path to A of length 2. Little does B know that C's path runs through B itself. For all B knows, C might have ten lines all with separate paths to A of length 2. As a result, B thinks it can reach A via C, with a path length of 3. D and E do not update their entries for A on the first exchange.

On the second exchange, C notices that each of its neighbors claims to have a path to A of length 3. It picks one of the them at random and makes its new distance to A 4, as shown in the third row of Fig(b). Subsequent exchanges produce the history shown in the rest of Fig.(b).

All routers work their way up to infinity, but the number of exchanges required depends on the numerical value used for infinity. For this reason, it is wise to set infinity to the longest path plus 1. If the metric is time delay, there is no well-defined upper bound, so a high value is needed to prevent a path with a long delay from being treated as down. This problem is known as the count-to-infinity problem.

Explain the link state routing (10 marks)

The link state routing is simple and has five parts. Each router has:

1. Discover its neighbors and learn this network addresses.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

1.Learning about the neighbors:

When a router is booted, its first task is to learn who its neighbors. 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. The three routers are all connected to F, it is essential that it can determine whether all three mean the same F.

When two or more routers are connected by a LAN, the situation is slightly more complicated. Fig(a) illustrates a LAN to which three routers A, C, and F, are directly connected. Each of these routers is connected to one or more additional routers, as shown.

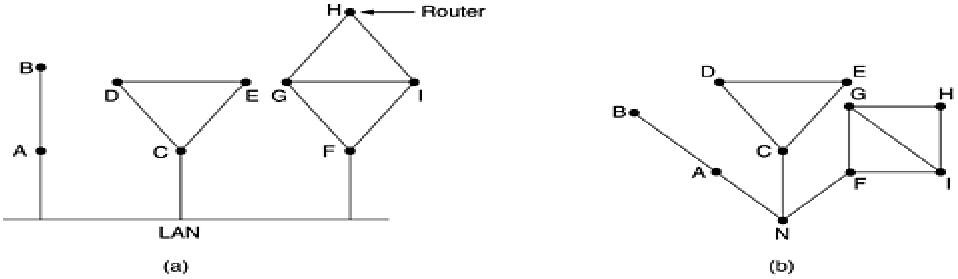


Fig: (a) Nine routers and a LAN. (b) A graph model of (a).

One way to model the LAN is to consider it as a node itself, as shown fig(b). We have introduced a new, artificial node, N, to which A, C and F are connected. The fact that it is possible to go from A to C on the LAN is represented by the path ANC here.

2. 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 id 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.

Consider the subnet of fig 5.12, which is divided into two parts, East and West, connected by two lines CF and EI.

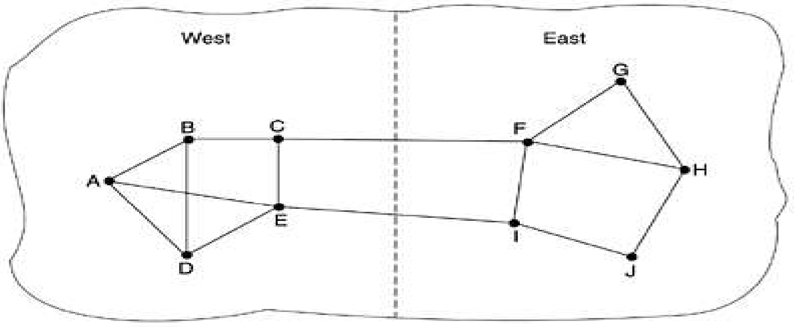
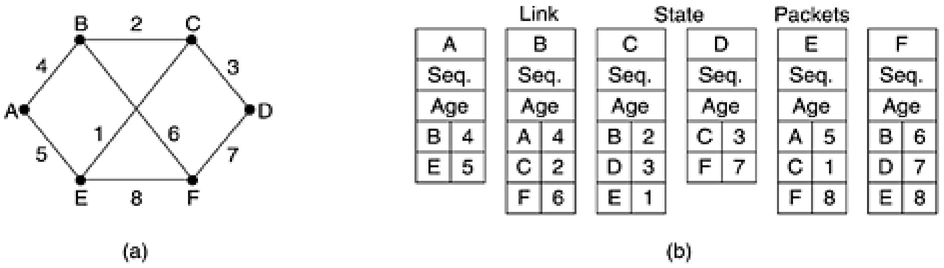


Fig: A subnet in which the East and West parts are connected by two lines.

Suppose that most of the traffic between east and west is using line CF, and as a result, this line is heavily loaded with ling delays. Including queuing delay in 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, overloading this line. 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 is considered, this problem does not occur.

3. Building link state packets:

The packet starts with the identity of the sender, followed by a sequence number and age, and a list of neighbors. For each neighbors, the delay to that neighbors is given. An example subnet is given in fig(a) with delays shown as labels on the links. The corresponding link state packets for all six routers are shown in fig(b).



A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

Fig: (a) A subnet. (b) The link state packets for this subnet

Building the link state packet is easy. The hard part is determining when to build them. One possibility is to build them periodically, that is, at regular intervals. 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.

4. Distributing the link state packets:

The packets are distributed and installed, the routers getting the first ones will change their routes. The different routers may be using different versions of the topology, which can lead to inconsistencies, loops, unreachable machines and other problems.

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. If it is new, it is forwarded on all lines except the one it arrived on. If it is a duplicate, it is discarded.

This algorithm has a few problems, but they are manageable. First, 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, if a router even 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, if a sequence number is ever corrupted and 65,540 is received instead of 4(a-1-bit error), packet 5 through 65,540 will be rejected as obsolete, since the current sequence number is thought to be 65, 540.

5. 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. Every link is, in fact, represented twice, once for each direction. For a subnet with n routers, each of which has k neighbors, the memory required to store the input data is proportional for kn. For large subnets, this can be a problem.

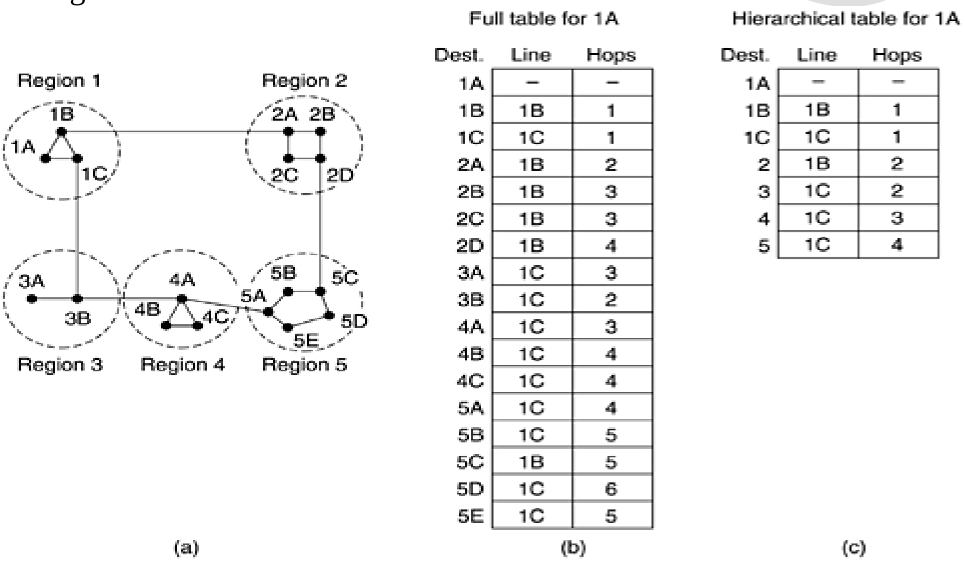
For example, if a router claims to have a line it does not have or forgets a line it does have, the subnet graph will be incorrect. If a router fails to forward packets or concepts them while forwarding them, trouble will arise. Finally, if it runs out of memory or does the routing calculation wrong bad things will happen.

Explain the Hierarchical routing. (10 marks)

The routers are divided into 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. When different networks are interconnected, each one as a separate region in order to free the routers in one network from the topological structure of the other ones.

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.

Fig. gives a quantitative example of routing in a two-level hierarchy with five regions. The full routing table for routers 1A has 17 entries, as fig (b). When routing is done hierarchically, as fig(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.



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

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

Fig: Hierarchical routing

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.

When a single network becomes very large, an interesting question is: How many levels should the hierarchy have? For example, consider a subnet with 720 routers. If there is no hierarchy, each router needs 720 routing table entries. If the subnet is partitioned into 24 regions of 30 routers each, each router needs 30 local entries plus 23 remote entries for a total of 53 entries. If a three-level hierarchy is chosen, with eight clusters, each containing a regions of 10 routers, each router needs 10 entries for local routers, 8 entries for routing to other regions within its own cluster, and 7 entries for distant clusters, for a total of 25 entries.

Explain the broadcasting routing. (10 marks)

Some applications, hosts need to send messages to many or all other hosts. For example, a service distributing weather reports, stock market updates, or live radio programs might work best by broadcasting to all machines and letting those that are interested read the data. Sending a packet to all destinations simultaneously is called broadcasting.

The first method, which requires no special features from the subnet is for the source to simply send a distinct packet to each destination.

The second method, flooding is ill-suited for ordinary point-to-point communication for broadcasting it might rate serious consideration, especially if none of the methods described below are applicable.

The third method is multi-destination routing. When a packet arrives at a router, the router checks all the destinations to determine the set of output lines that will be needed. The router generates a new copy of the packet for each output line to be used and includes in each packet only those destinations that are to use the line. After a sufficient number of hops, each packet will carry only one destination and can be treated as a normal packet.

The fourth method, a spanning tree is a subnet of the subnet that includes all the routers but contains no loops. If each router knows which of its lines belong to the spanning tree, it can copy an incoming broadcast packet onto all the spanning tree lines except the one it arrived on. The problem is that each router must have knowledge of some spanning tree for the method to be applicable. Sometimes this information is available (e.g., with link state routing) but sometimes it is not (e.g., with distance vector routing).

The last method is reverse path forwarding, 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. 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. The router forwards copies of it onto all lines except the one it arrived on. The broadcast packet arrived on a line other than the preferred one for reaching the source; the packet is discarded as a likely duplicate.

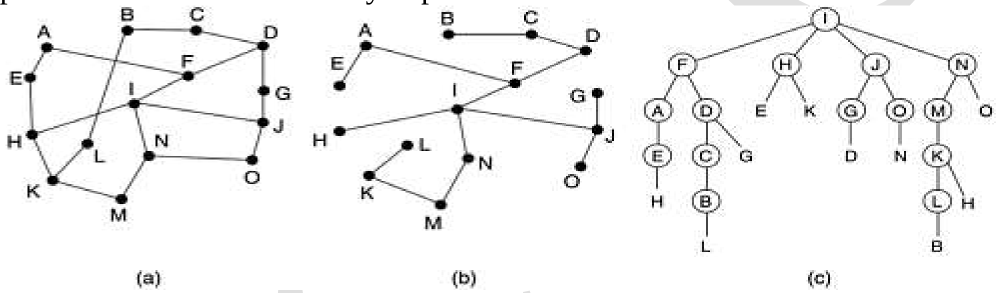


Fig: Reverse path forwarding. (a) A subnet. (b) A sink tree. (c) The tree built by reverse path forwarding

For example, part (a) shows a subnet, part (b) shows a sink tree for router I of the subnet, and part (c) shows how the reverse path algorithm works. On the first hop, I send packets to F, H, J, and N, as indicated by the second row of the tree. Each of these packets arrives on the preferred path to I (assuming that the preferred path falls along the sink tree) and indicated by a circle around the letter. On the second hop, eight packets are generated, two by each of the routers that received a packet on the first hop. All of these arrive at previously unvisited routers, and five of these arrive along the preferred line. Of the six packets generated on the third hop, only three arrive on the preferred path (at C, E, and K), the others are duplicates. After five hops and 24 packets, the broadcast terminates, compared with four hops and 14 packets had the sink tree been followed exactly.

Explain the multicast routing. (10 marks)

Some applications require that widely separately processes work together in groups, for example, a group of processes implementing a distributed database system. One process to send a message to all the other member of the group. Sending a message to a group is called multicasting and its routing algorithm is called multicast routing.

In multicast routing, each router computes a spanning tree covering all other routers. For example, in fig (a) we have two groups, 1 and 2. Some routers are attached to hosts that belong to one or both of these groups. A spanning tree for the leftmost router is shown in fig (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. Fig (c) shows the pruned spanning tree for group 1. Fig (d) shows the pruned spanning tree for group 2. Multicast packets are forwarded only along the appropriate spanning tree.

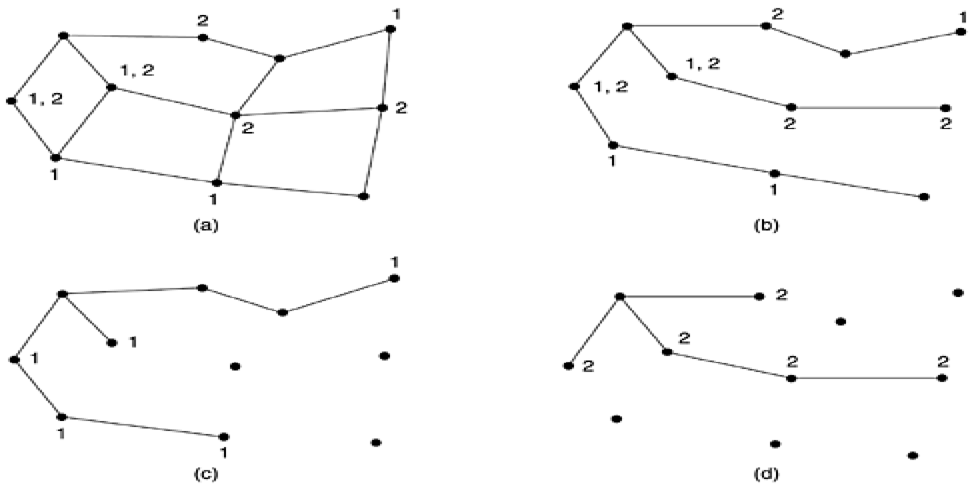


Fig: (a) A network. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.

Explain the routing for mobile hosts. (10 marks)

We have a WAN consisting of routers and hosts. Connected to the WAN are LANs, MANs and wireless cells.

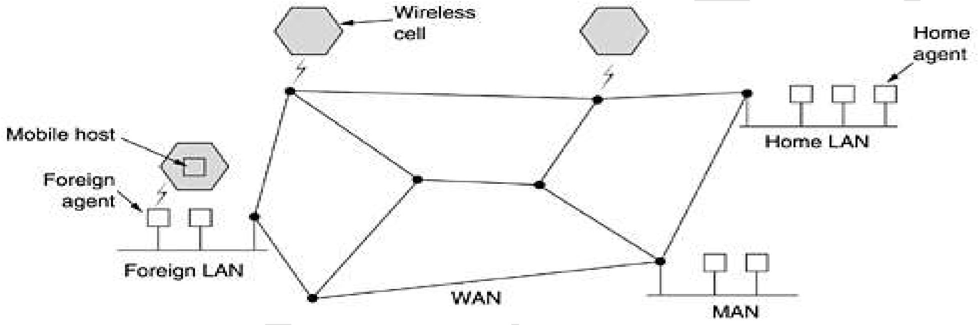


Fig: A WAN to which LANs, MANs, and wireless cells are attached

Hosts that never move are said to be stationary. They are connected to the network by copper wires or fiber optics. Roaming hosts actually compute on the run and want to maintain their connections as they move around.

The world is divided up (geographically) into small units. Let us call them areas, where an area is typically a LAN or wireless cell. Each area has one or more foreign agents, which are processes that keep track of all mobile hosts visiting the area. In addition, each area has a home agent, which keeps track of hosts whose home is in the area, but who are currently visiting another area.

When a new host enters an area, either by connecting to it (e.g., plugging into the LAN) or just wandering into the cell, his computer must register itself with the foreign agent there. The registration procedure typically works like this:

1. Each foreign agent broadcast a packet announcing its existence and address.
2. The mobile host registers with the foreign agent, giving its home address, current data link layer address, and some security information.
3. The foreign agent contacts the mobile host's home agent and says: one of your hosts is over here.
4. The home agent examines the security information, which contains a timestamp, to prove that it was generated within the part few seconds.
5. When the foreign agent gets the acknowledgement from the home agent, it makes an entry in its tables and informs the mobile host that it is now registered.

When a packet is sent to a mobile host, it is routed to the host's home LAN because that is what the address says should be done, as in step1 of fig. The sender, in the northwest city of Seattle, wants to send a packet to a host normally across the United States in New York. Packets sent to the mobile host on its home LAN in New York are intercepted by the home agent there. The home agents then look up the mobile host's new (temporary) location and find the address of the foreign agent handling the mobile host, in Los Angles.

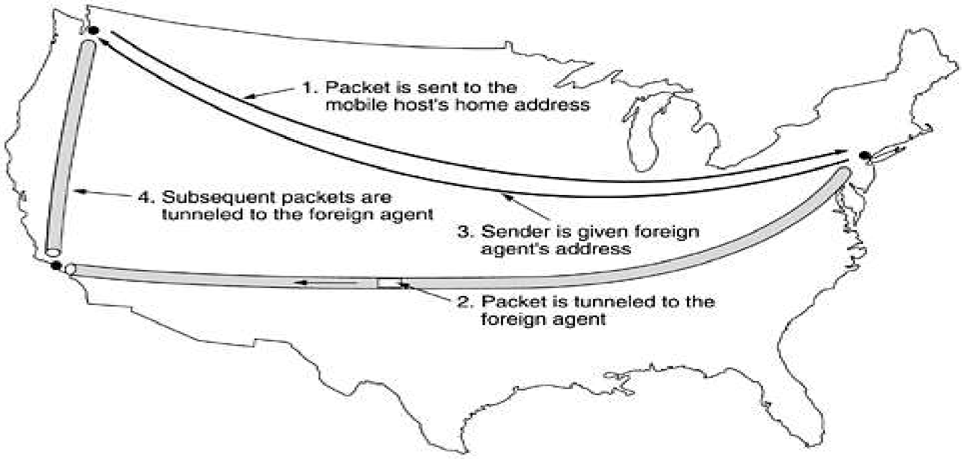

1. Packet is sent to the mobile host's home address
4. Subsequent packets are tunneled to the foreign agent
3. Sender is given foreign agent's address
2. Packet is tunneled to the foreign agent

Fig: Packet routing for mobile hosts

The home agent then does two things. First, it encapsulates the packet in the payload field of an outer packet and sends the latter to the foreign agent (step 2). This mechanism is called Tunneling. After getting the encapsulated packet, the foreign agent removes the original packets from the payload field and sends it to the mobile host as a data link frame.

Second, the home agent tells the sender to henceforth send packets to the mobile host by encapsulating them in the payload of packets explicitly addressed to the foreign agent instead of just sending them to the mobile host's home address (step 3). Subsequent packets can now be routed directly to the host via the foreign agent (step 4), bypassing the home location entirely.

Discuss in detail the routing in Ad Hoc networks. (10 marks)

Each node consists of a router and a host, usually on the same computer. Networks of nodes that just happen to be near each other are called ad hoc networks or MANETs (Mobile Ad hoc NETworks). The AODV (Adhoc On-demand Distance Vector), it determines a route to some destination only when somebody wants to send a packet to that destination.

Route Discovery:

An adhoc network of fig, in which a process at node A wants to send a packet to node I. The AODV algorithm maintains a table at each node, keyed by destination, giving information about that destination, including which neighbor to send packets to in order to reach the destination. Suppose that A looks in its table and does not find an entry for I. It now has discover a route to I. This property of discovering routes only when they are needed is what makes this algorithm "on demand".

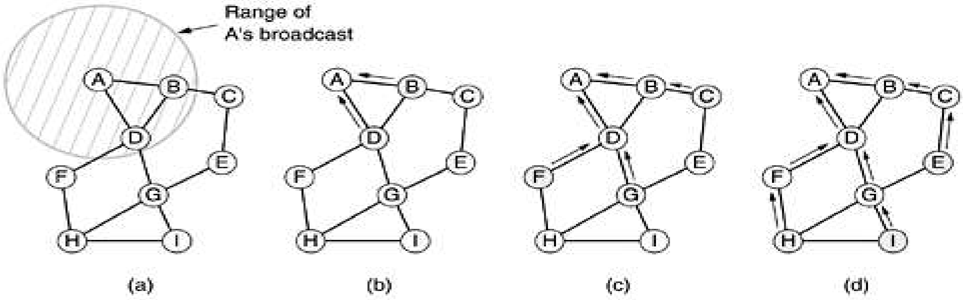
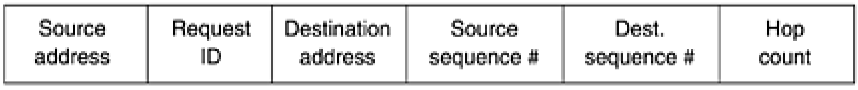


Fig: (a) Range of A's broadcast. (b) After B and D have received A's broadcast. (c) After C, F, and G have received A's broadcast. (d) After E, H, and I have received A's broadcast. The shaded nodes are new recipients. The arrows show the possible reverse routes.

To locate I, A constructs a special ROUTE REQUEST packet and broadcasts it. The packet reaches B and D, as illustrated in fig(a). The reason B and D are connected to A in the graph is that they can receive communication from A. F, for example, is not shown with an arc to A because it cannot receive A's radio signal. F is not connected to A. The format of the ROUTE REQUEST packet is shown in fig. 5.21.



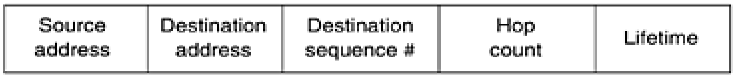
Source address	Request ID	Destination address	Source sequence #	Dest. sequence #	Hop count

Fig: Format of a ROUTE REQUEST packet

It contains the source and destination addresses, their IP addresses, which identify who is looking for whom. It also contains a request ID, which is a local counter maintained separately by each node and incremented each time a ROUTE REQUEST is broadcast. The fourth field of A's sequence counter. The fifth field is the most recent value of I's sequence number that A has seen. The final field, Hop count, will keep track of how many hops the packet has made. It is initialized to 0.

Neither B nor D knows where I is, so each of them creates a reverse route entry pointing back to A, as shown by the arrows in fig, and broadcasts the packet with hop count set to 1. The broadcast from B reaches C and D. C makes an entry for it in its reverse route table and rebroadcasts it. D rejects it as a duplicate. D’s broadcast is rejected by B. D’s broadcast is rejected by b. D’s broadcast is accept by F and G and stored, as shown in fig (c). After E, H, and I receive the broadcast, the ROUTE REQUEST finally reaches a destination that knows where I is, namely, I itself, as shown in fig (d).

In response to the incoming requests, I build a ROUTE REPLY packet as shown in fig. The source address, destination address, and hop count are copied from the incoming request, but the destination sequence number taken from its counter in memory. The Hop count field is set to 0. The lifetime field controls how long the route is valid. This packet is unicast to the node that the ROUTE REQUEST packets come from G. It then follows the reverse path to D and finally to A. At each node, Hop count is incremented so the node can see how far from the destination (I) it is.



Source address	Destination address	Destination sequence #	Hop count	Lifetime

Fig: Format of a ROUTE REPLY packet

At each intermediate node on the way back, the packet is inspected. It is entered into the local routing table as a route to I if one or more of the following three conditions are met:

1. No route to I is known.
2. The sequence number for I in the ROUTE REPLY packet is greater than the value in the routing table.
3. The sequence numbers are equal but the new route is shorter.

Write the congestion control algorithms (5 marks)

Write down the general principles of congestion control. (5 marks)

Congestion control Algorithm:

When too many packets are present in the subnet, performance degrades. This situation is called congestion. Fig depicts the symptom. When the number of packets dumped into the subnet by the hosts is within its carrying capacity, they are all delivered and the number delivered is proportional to the number sent. 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 trafffic, performance collapses completely and almost no packets are delivered.

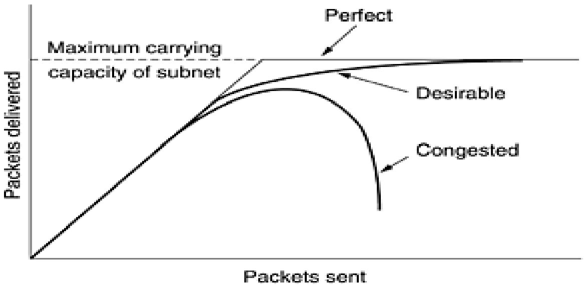


Fig: When too much traffic is offered, congestion sets in and performance degrades sharply

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.

- Low-bandwidth lines can also cause congestion. Upgrading the lines but not changing the processors, or vice versa, often helps a little, but frequently just shifts the bottleneck.

The difference between congestion control and flow control are:

Congestion control has to do with making sure the subnet is able to carry the offered traffic

Flow control relates to the point-to-point traffic between a given sender and a given receiver. Its job is to make sure that a fast sender cannot continually transmit data faster than the receiver is able to absorb it. Flow control frequently involves some direct feedback from the receiver to the sender to tell the sender how things are doing at the other end.

General Principles of 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.

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.

They begin by dividing all algorithms into open loop or closed loop. They further divide the open loop algorithms into ones that act at the source versus ones that act at the destination. The closed loop algorithms are also divided into two subcategories: explicit feedback versus implicit feedback. In explicit feedback algorithms, packets are sent back from the point of congestion to warn the source. In implicit algorithms, the source deduces the existence of congestion by making local observations, such as the time needed for acknowledgements to come back.

Write a note on congestion prevention policies. (5 marks)

Congestion prevention policies:

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 queuing 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

Discuss about the Virtual-circuit subnets control and datagram subnets in congestion. (10 marks)

Virtual-Circuit subnets:

It 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. Attempts to set up new transport layer connections fail. While this approach is crude, it is simple and easy to carry out. In the telephone system, when a switch gets overloaded, it also policies admission control by not giving dial tones.

For example, consider the subnet of fig(a) in which two routers are congested.

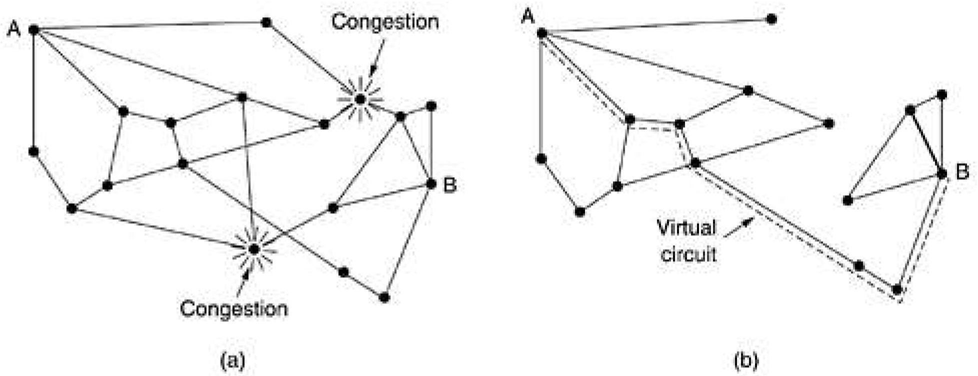


Fig: a) A congested subnet. (b) A redrawn subnet that eliminates the congestion. A virtual circuit from A to B is also shown.

Suppose that a host attached to router A wants to set up a connection to a host attached to router B. This connection would pass through one of the congested routers. To avoid this situation, we can redraw the subnet as shown in fig(b), omitting the congested routers and all of their lines. The dashed line shows a possible route for the virtual that avoids the congested routers.

Datagram subnets:

Each router can easily monitor the utilization of its output 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

$$U_{new}=qu_{old} + (1-a)f$$

Where the constant a determines how fast the router forgets recent history.

The warning bit:

The old DECNET architecture signaled the warning state by setting a special bit in the packet’s header. So does frame delay. When the packet arrived at its destination, the transport entity copied the bit into the next acknowledgement sent back to the source. The source then cut back on traffic.

As long as the route was in the warning state, it continued to set the warning bit, which meant that the source continued to get acknowledgements with it set. Every router along the path could set the warning bit, traffic increased only when no router was in trouble.

Choke Packets:

The router sends a choke packet back to the source host, giving it the destination found in the packet. The original packet is tagged 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. 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 few 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 few unless trouble occurs.

Hop-by-Hop choke packets:

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. For example, a host in San Francisco (router A in fig) that is sending traffic to a host in New York host (router D in fig.) 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.

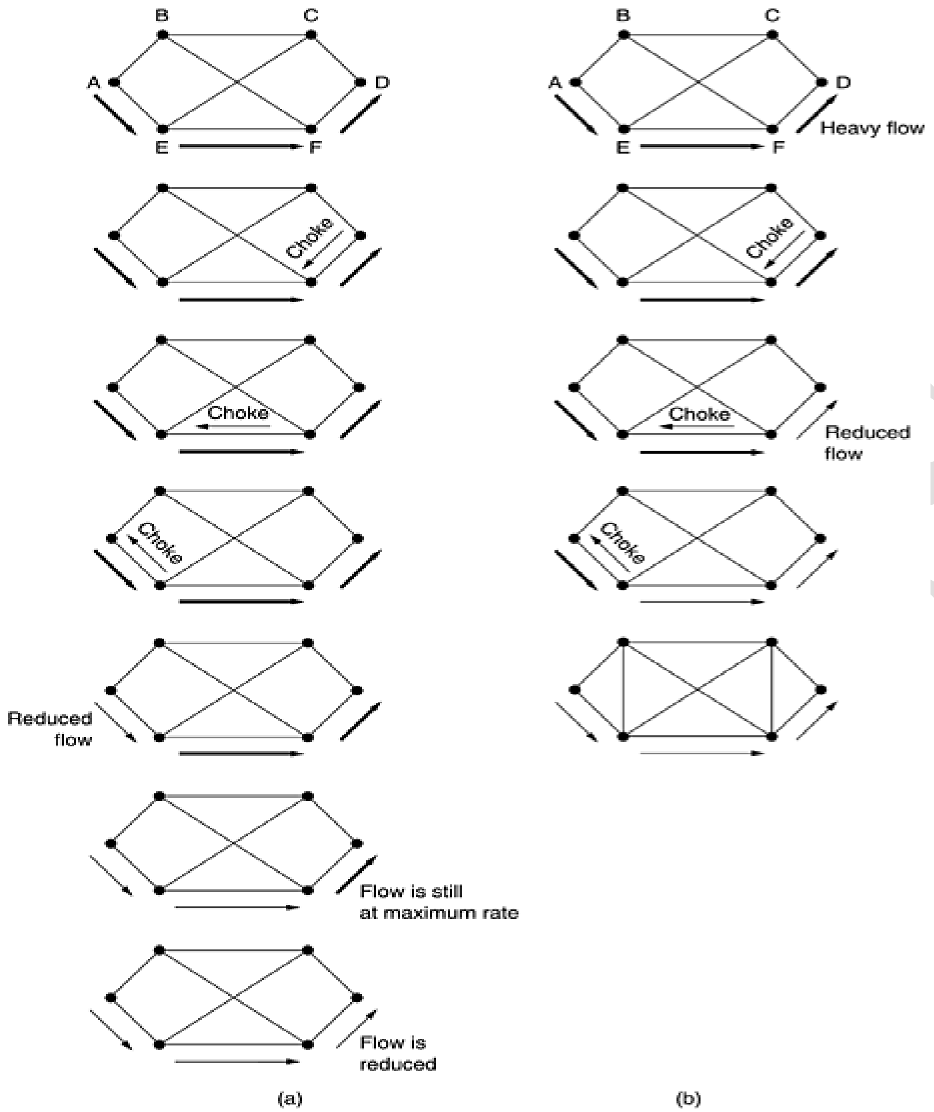


Fig: (a) A choke packet that affects only the source. (b) A choke packet that affects each hop it passes through.

The choke packet propagation is shown as the second, third and fourth steps in fig (a). In those 30 msec, another 4.6 megabits will have been sent. Even if the host in San Francisco completed 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 (a) will the New York router notice a slower flow.

An alternative approach is to have the choke packet take effect at every hop it passes through as shown in the sequence of fig (b). Choke packet reaches F, F is required to reduce the flow to D. In the next step, the choke packet reaches E, which tells E to reduce the flow to F. Finally the choke packet reaches A and the flow genuinely slows down.

Load shedding:

Load shedding is a fancy way of saying that when routers are being inundated 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. For file transfer, an old packet is worth more than a new one because dropping packet 6 and keeping packets 7 through 10 will cause a gap at the receiver that may force packets 6 through 10 to be retransmitted. In a 12 packet file, dropping 6 may require 7 through 12 to be retransmitted, whereas dropping 10 may require only 10 through 12 to be retransmitted. In contrast, for multimedia, a new packet is more important than old one. The former policy (old is better than new) is often called wine and the latter (new is better than old) is often called milk.

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.

Since the router probably cannot tell which source is causing most of the trouble, picking a packet at random from the queue that triggered the action is probably as good as it can do.

How should the router tell the source about the problem? One way is to send it a choke packet. A problem with that approach is that it puts even more load on the already congested network. A different strategy is to just discard the selected packet and not report it. In wireless networks, where most losses are due to noise on the air link, this approach cannot be used.

Write short notes on Jitter Control. (5 marks)

For application such as audio and video streaming, it does not matter much if the packets take 20 msec or 30 msec to be delivered, as long as the transmit time is constant. The variation in the packet arrival times is called jitter. High jitter, for example, having some packets taking 20 msec and others taking 30 msec to arrive will give an uneven quality to the sound or movie. Jitter is illustrated in fig. An agreement that 99 percent of the packets be delivered with a delay in the range of 24.5 msec to 25.5 msec might be acceptable.

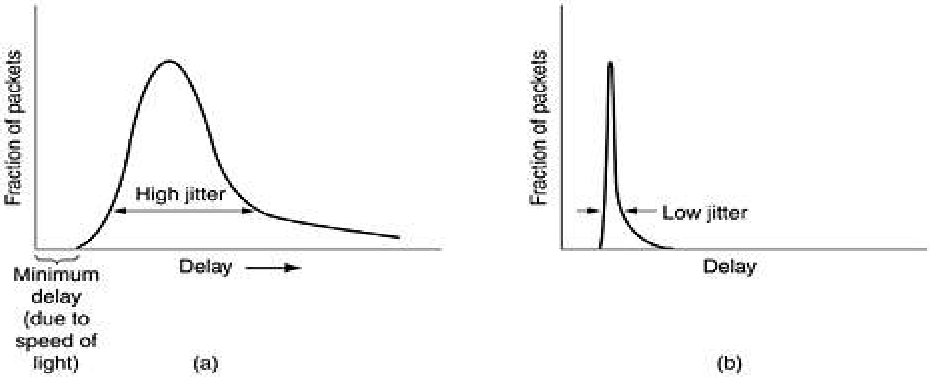


Fig:(a) High jitter. (b) Low jitter

The jitter can be bounded by computing the expected transit time for each hop along the path. When a packet arrives at a router, the router checks to see how much the packet is behind or ahead of its schedule. The information is stored in the packet and updated at each hop. If the packet is ahead of schedule, it is held just long enough to get it back on schedule. If it is behind schedule, the router tries to get it out the door quickly.

Explain the Quality of Service. (10 marks)

1) Requirements:

A stream of packets from a source to a destination is called a flow. In a connection-oriented network, all the packets belonging to a flow follow the same route; in a connectionless network, they may follow different routes. The needs of each flow can be characterized by four primary parameters: reliability, delay, jitter and bandwidth. Together these determine the QoS (Quality of Service) the flow requires. Several applications and their requirements are listed below:

Applications	Reliability	Delay	Jitter	Bandwidth
E-mail	High	Low	Low	Low
File transfer	High	Low	Low	Medium
Web access	High	Medium	Low	Medium
Remote login	High	Medium	Medium	Low
Audio on demand	Low	Low	High	Medium
Video on demand	Low	Low	High	High
Telephony	Low	High	High	Low
Video conferencing	Low	High	High	High

The first four applications have stringent requirements on reliability. This goal is usually achieved by check summing each packet and verifying the checksum at the destination. The four final (audio/video) applications can tolerate errors, so no checksums are computed or verified.

File transfer applications, including e-mail and video are not delay sensitive. Interactive applications, such as web surfing and remote login, are more delay sensitive. Real-time applications, such as telephony and videoconferencing have strict delay requirements playing audio or video files from a server does not require low delay.

The first three applications are not sensitive to the packets arriving with irregular time intervals between them. Remote login suffers much jitter. Video and especially audio are extremely sensitive to jitter. The application differ in their bandwidth needs, with e-mail and remote login not needing much, but video in all forms needing a great deal. ATM networks classify flows in four broad categories with respect to their QoS demands as follows:

1) Constant bit rate (e.g., telephony)
2) Real-time variable at rate (e.g., compressed video conferencing)
3) Non-real-time variable bit rate (e.g., watching a movie over the Internet)
4) Available bit rate (e.g., File transfer)

2) Techniques for achieving good quality of service:

Some of the techniques system designers use to achieve QoS:

Overprovisioning:

An easy solution is to provide so much router capacity, buffer space and bandwidth that the packets just fly through easily. The trouble with this solution is that it is expensive.

Buffering:

In fig, we see stream a packets being delivered with substantial jitter. Packet 1 is sent from the server at t=0 sec and arrives at the client at t=1 sec. Packet 2 undergoes more delay and takes 2 sec to arrive. As the packets arrive, they are buffered on the client machine.

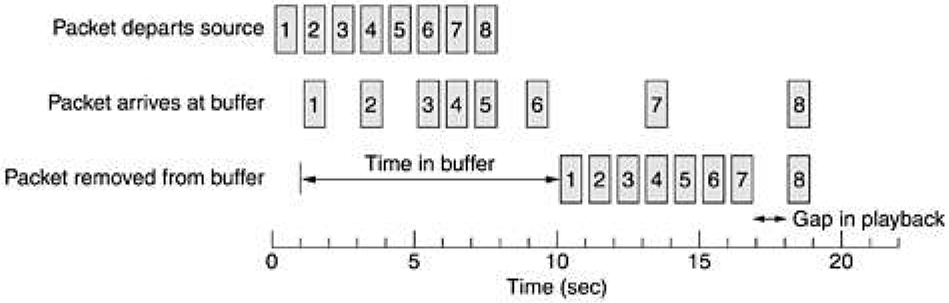


Fig: Smoothing the output stream by buffering packets.

At t=0 sec, playback begins. At this time, packets 1through 6 have been buffered so that they can be removed from the buffer at uniform intervals for smooth play. Packet 8 has been delayed so much that it is not available when its play slot comes up, so playback must stop until it arrives, creating an annoying gap in the music or movie.

Traffic Shaping:

The source outputs the packets with a uniform spacing between them, but in other cases, they may be emitted irregularly, which may cause congestion to occur in the network. Nonuniform output is common if the server is handling many streams at once, and it also allows other actions, such as fast forward and rewind, uses authentication, and so on. We used here (buffering) is not always possible, for example, with videoconferencing. However, if something could be done to make the server (and hosts in general) transmit at a uniform rate, quality of service would be better. With a technique, traffic shaping, which smooth out the traffic on the server side, rather than on the client side.

Traffic shaping is about regulating the average rate of data transmission. In contrast, the sliding window protocols, the amount of data in transit at once, not the rate at which it is sent. When a connection is set up, the user and the subnet (i.e., the customer and the carrier) agree on a certain traffic pattern (i.e., shape) for that circuit. Sometimes is called a service level agreement.

The leaky bucket algorithm:

Imagine a bucket with a small hole in the bottom, as illustrated in fig (a). No matter the rate at which water enters the bucket, the overflow is at a constant rate, p. When there is any water in the bucket and zero when the bucket is empty. Once the bucket is full, any additional water entering it spills over the sides and is lost.

The same idea can be applied to packets, as shown in Fig (b). Each host is connected to the network by an interface containing a leaky bucket, a finite interval queue. If a packet arrives at the queue when it is full, the packet is discarded. This arrangement can be built into the hardware interface or stimulated by the host operating system. It was first proposed by Turner (1986) and is called the leaky bucket algorithm.

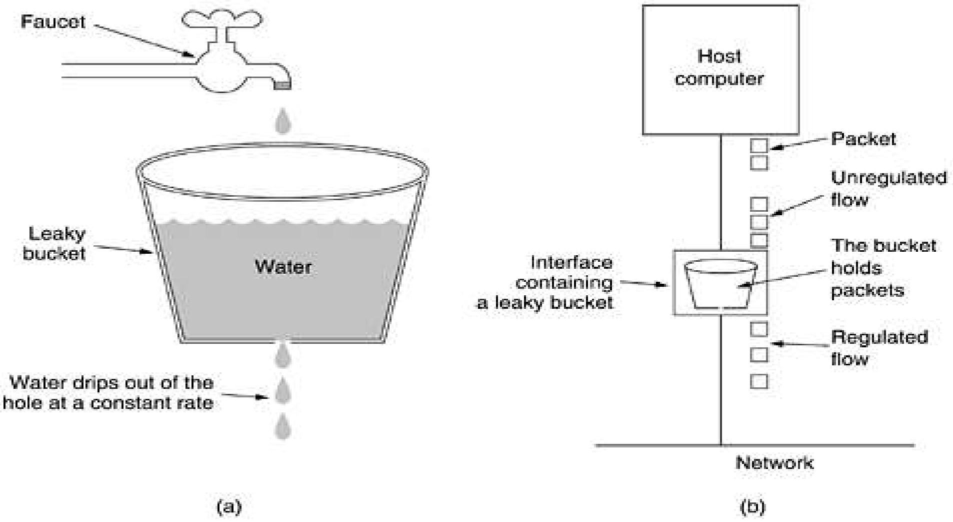


Fig: (a) A leaky bucket with water. (b) A leaky bucket with packets

Implementing the original leaky bucket algorithm is easy. The leaky bucket consists of a finite queue. When a packet arrives, if there is room on the queue it is appended to the queue, otherwise, it is discarded. At every clock tick, one packet is transmitted (unless the queue is empty).

The byte-counting leaky bucket is implemented almost the same way. At each tick, a counter is initialized to n. If the first packet on the queue has fewer bytes than the current value of the counter, it is transmitted, and the counter is decremented by that number of bytes. Additional packets may also be sent, as long as the counter is high enough. When the counter drops below the length of the next packet on the queue, transmission stops until the next tick, at which time the residual byte count is reset and the flow can continue.

The token bucket algorithm:

In fig (a) we see a bucket holding three tokens, with five packets waiting to be transmitted. For a packet to be transmitted, it must capture and destroy one token. In fig (b) we see that three of the five packets have gotten through, but the other two are stuck waiting for two more tokens to be generated.

The token bucket algorithm does allow saving, upto the maximum size of the bucket n. This property means that burst of upto n packets can be sent at once, allowing some bushiness in thin the output stream and giving faster response to sudden burst of input. A host can make the host stop sending when the rules say it must. Telling a router to stop sending while its input keeps pouring in may result in lost data.

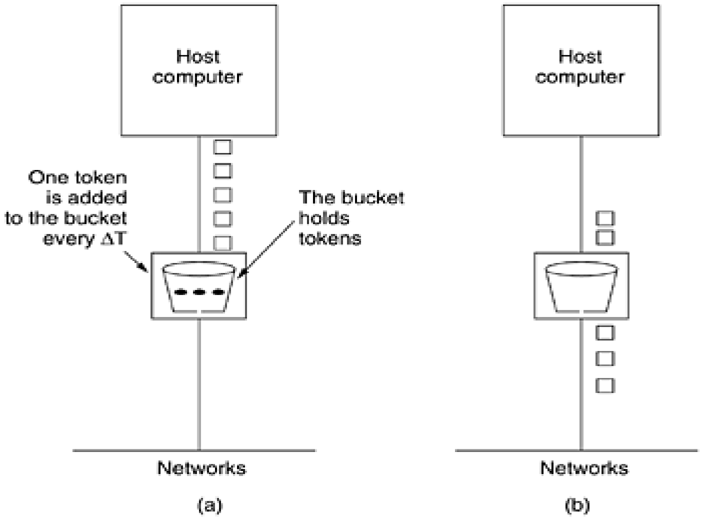


Fig: The token bucket algorithm. (a) Before. (b) After

The implementation of the basic token bucket algorithm is just a variable that counts token. The counter is incremented by one every ΔT and decremented by one whenever a packet is sent. When the counter hits zero, no packets may be sent. In the byte-count variant, the counter is incremented by k bytes ΔT and decremented by the length of each packet sent.

Explain the Internetworking (10 marks)

When two or more networks are connected to from an Internet. An example of how different networks might be connected, consider the fig. We see a corporate network with multiple locations tied together by a wide area ATM network. At one of the locations, an FDDI optical backbone is used to connect an Ethernet, an 802.11 wireless LAN, and the corporate data center's SNA mainframe network.

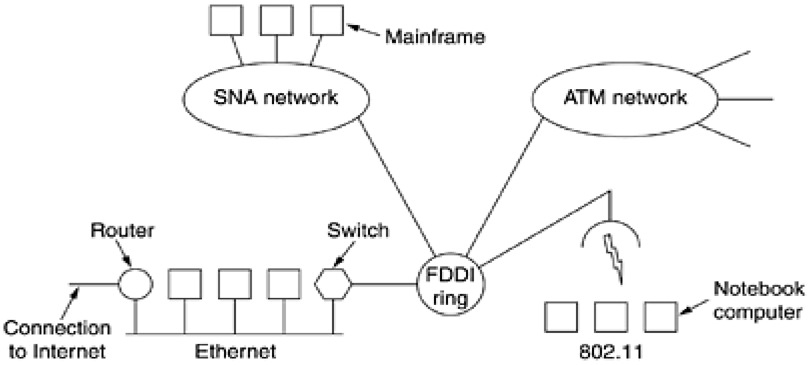


Fig: A collection of interconnected networks

The purpose of interconnecting all these networks is to allow users on any of them to communicate with users on all the other ones and also to allow users on any of them to access data on any of them. This goal means sending packets from one network to another. Since networks often differ in important ways, getting packets from one network to another is not always so easy.

How network differ:

Networks can differ in many ways. Some of the differences, such as different modulation techniques or frame formats, are in the physical and data link layers. We lost some of the differences that can occur in the network layer.

Item	**Some possibilities**
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)`
Packet size	Every network has its own maximum
Quality of service	Present or absent; many different kinds
Error handling	Reliable, ordered and unordered delivery

Flow control	Sliding window, rate control, other, or none
Congestion control	Leaky bucket, token bucket, RED, choke packets, etc
Parameters	Different timeouts, flow specifications, etc
Security	Privacy rules, encryption, etc
Accounting	By connect time, by packet, by byte, or not at all

How networks can be connected:

In physical layer, networks can be connected by repeaters or hubs, which just move the bits from one network to an identical network. In data link layer, operates bridges and switches. They can accept frames, examine the MAC addresses and forward the frames to a different network. In network layer, we have routers that can connect two networks. If two networks have dissimilar network layers, the router may be able to translate between the packet formats. A router that can handle multiple protocols is called a multi-protocol router.

In transport layer we find transport gateways, which can interface between two transport connections. In the application layer, application gateways translate message semantics.

To see how that differs from switching in the data link layer, examine fig. In fig (a), the source machine, S, wants to send a packet to the destination machine, D. These machines are on different Ethernets, connected by a switch. S encapsulates the packet in a frame and sends it on its way. The frame arrives at the switch, which then determines that the frame has to go to LAN 2 by looking at its MAC address. The switch just removes the frame from LAN 1 and deposits it on LAN 2.

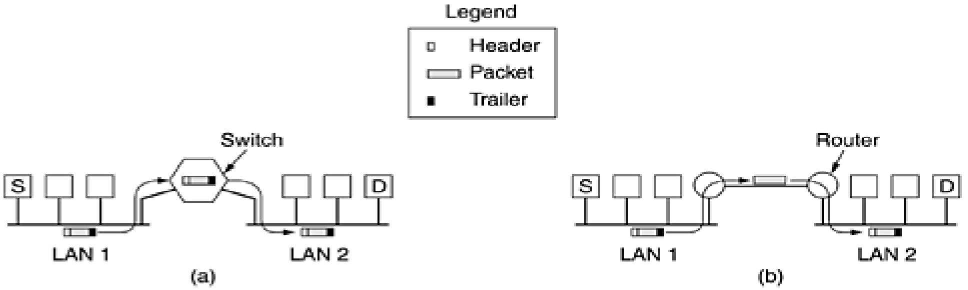


Fig: (a) Two Ethernets connected by a switch. (b) Two Ethernets connected by routers.

Let us consider the fig (b), the same situation but with the two Ethernets connected by a pair of routers instead of a switch. The routers are connected by a point-to-point line, possibly a leased line thousands of kilometers long. The frame is picked up by the router and the packet removed from the frame's data field. The router examines the address in the packet (e.g., an IP address) and looks up this address in its routing table. Based on this address, it decides to send the packet to the remote router, potentially encapsulated in a different kind of frame, depending in the line protocol. At the far end, the packet is put into the data field of an Ethernet frame and deposited onto LAN 2.

Concatenated Virtual Circuits:

In the concatenate virtual-circuit model, shown in fig, a connection to a host in a distant network is set up in a way similar to the way connections are normally established. The subnet sees that the destination is remote and builds a virtual circuit to the router nearest the destination network. Then it constructs a virtual circuit from that router to an external gateway. This gateway records the existence of the virtual circuit in its tables and proceeds to build another virtual circuit to a router in the next subnet. This process continues until the destination host has been reached.

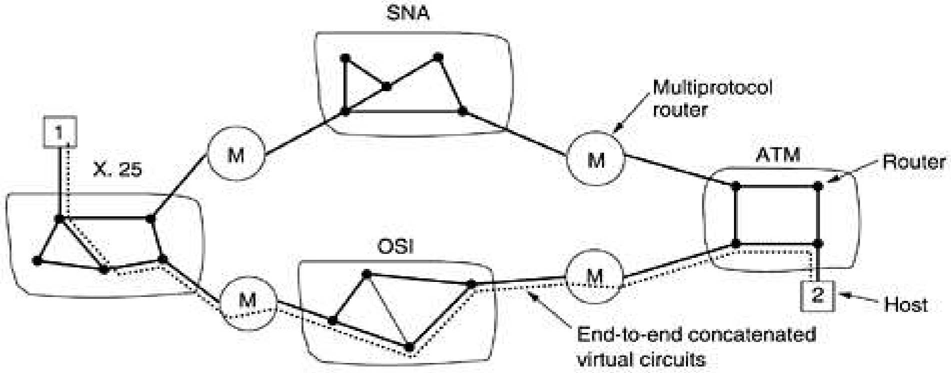


Fig: Internetworking using concatenated virtual circuits

Once data packets begin flowing along the path, each gateway relays incoming packets, converting between packet formats and virtual-circuit numbers as needed. All data packets must traverse the same sequence of gateways packets in a flow are never recorded by the network.

The sequence of virtual circuits is set up from the network source through one or more gateways to the destination. Each gateway maintains tables telling which virtual circuits pass through it, where they are to be routed, and what the new virtual circuit numbers is concatenated virtual circuits are also common in the transport layer.

Connectionless Internetworking:

In fig datagram's from host 1 to host 2 are shown taking different routes through the Internetwork. A routing decision is made separately for each packet, possibly depending on the traffic at the moment the packet is sent. This strategy can use multiple routes and thus achieve a higher bandwidth than the concatenated virtual-circuit model. There is no guarantee that the packets arrive at the destination in order, assuming that they arrive at all.

For one thing, if each network has its own network layer protocol, it is not possible for a packet from one network to transit another one. One could imagine the multiprotocol routers actually trying to translate from one format to another, but unless the two formats are close relatives with the same information fields, such conversions will always be incomplete and often doomed to failure.

A second, and more serious, problem is addressing, a host on the internet is trying to send an IP packet to a host on an adjoining SNA network. The IP and SNA addresses are different one would need a mapping between IP and SNA addresses in both directions. The concept of what is addressable in different. In IP, hosts (actually, interface cards) have address. In SNA, entities other than hosts (e.g., hardware devices) can also have addresses.

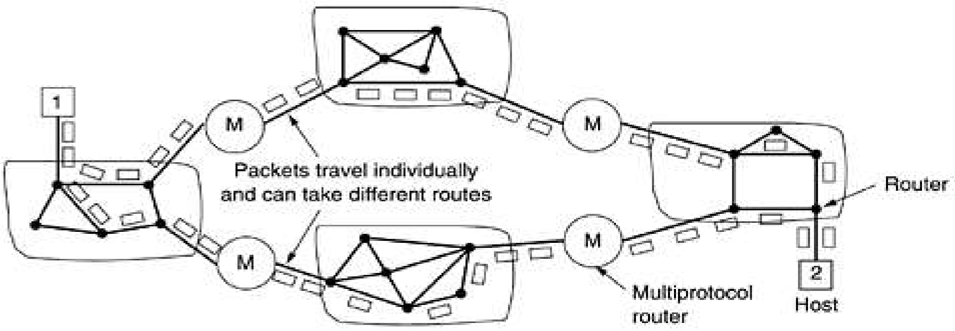


Fig: A connectionless internet

A major advantage of the datagram approach to internetworking is that it can be used over subnets that do not use virtual circuits inside. Many LANs, mobile networks (e.g., aircraft and naval fleets), and even some WANs fall into this category.

Tunneling:

In making two different networks interwork is exceedingly difficult. There is a common special case that is manageable. This case is where the source and destination hosts are on the same type of network, but there is a different network in between. As an example, think of an international bank with a TCP/IP based Ethernet in Paris, a TCP/IP based Ethernet in London, and a non-IP wide area network (e.g., ATM) in between, as shown in fig.

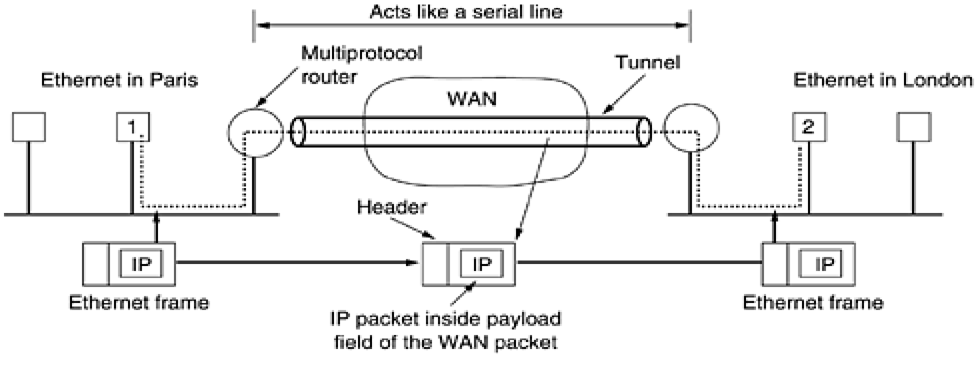


Fig: Tunneling a packet from Paris to London

The solution to this problem is a technique called tunneling. To send an IP packet to host 2, host 1 constructs the packet containing the IP address of host 2, inserts it into an Ethernet frame addressed to the pair’s multi-protocol router, and puts it on the Ethernet. When the multi-protocol router gets the frame, it removes the IP packet, inserts it in the payload field of the WAN address of the London multi-protocol router. When it gets there, the multi-protocol removes the IP packet and sends it to host 2 inside it an Ethernet frame.

The WAN can be seen as a big tunnel extending from one multi-protocol router to the other. The IP packet just travels from one end of the tunnel to the other, sung in its nice box. It does not have to worry about dealing with the WAN at all. Neither do the hosts on either Ethernet. Only the multi-protocol router has to understand IP and WAN packets. The entire distance from the middle of one multi-protocol router to the middle of the other acts like a serial line.

Internetworking Routing:

Routing through an internetworking is similar to routing within a single subnet, but with some added complications. Consider, for example, the internetwork of fig(a) in which five networks are connected by six 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 (a) can directly access A and C via network 2 and also D via network 3. This leads to the graph of fig (b).

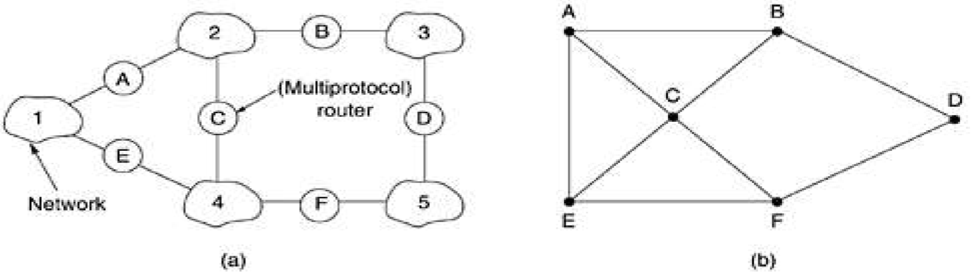


Fig: (a) An internetwork. (b) A graph of the internetwork

Once the graph has been constructed, known routing algorithms, such as the distance vector and line state algorithm, can be applied to the set of multi-protocol routers. This gives a two-level routing algorithm: within each network an interior gateway protocol is used, but between the network, an exterior gateway protocol is used (“gateway” is an older term for “router”).

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 buffer 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 retransmission to some level.
6) Desire to prevent one packet from occupying the channel too long.

The only solution to the problem is to allow gateways to break up packets into fragments, sending each fragment as a separate Internet packet.

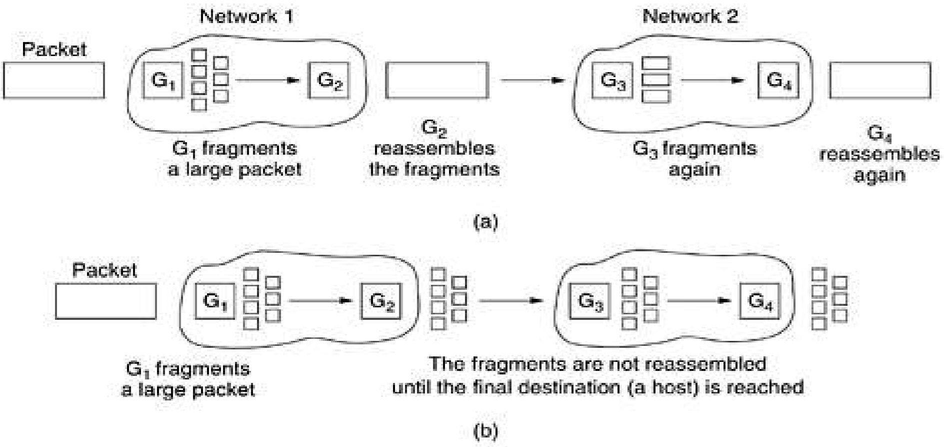

Network 1
Network 2
Packet
G1
G2
G3
G4
G1 fragments a large packet
G2 reassembles the fragments
G3 fragments again
G4 reassembles again
(a)
Packet
G1
G2
G3
G4
G1 fragments a large packet
The fragments are not reassembled until the final destination (a host) is reached
(b)

Fig: (a) Transparent fragmentation. (b) Nontransparent fragmentation

Two opposing strategies exist for recombining the fragments back into the original packet. The first strategy is to make fragmentation caused by a "small packet" network transparent to any subsequence networks through which the packet must pass on its way to the ultimate destination. This option is shown in fig 5.50 (a). In this approach, the small packet network has gateways that interface to other networks. When an oversized packet arrives at a gateway, the gateway breaks it up into fragments. Each fragment is addressed to the same exit gateway, where the pieces are recombined. In this way passage through the small-packet network has been made transparent. Subsequent networks are not even aware that fragmentation has occurred. ATM networks, for example, have special hardware to provide transparent fragmentation of packets into cells and then reassembly of cells into packets.

The other fragmentation strategy is to refrain from recombining fragments at any intermediate gateways. Once a packet has been fragmented, each fragment is treated as though it were an original packet. All fragments are passed through the exit gateway, as shown in fig 5.50 (b). Recombination occurs only at the destination host IP works this way.

Explain the IP Protocol. (5 marks)

An IP datagram consists of a header part and a text part. The header has a 20-byte fixed part and a variable length optional part. The header format is shown in fig.

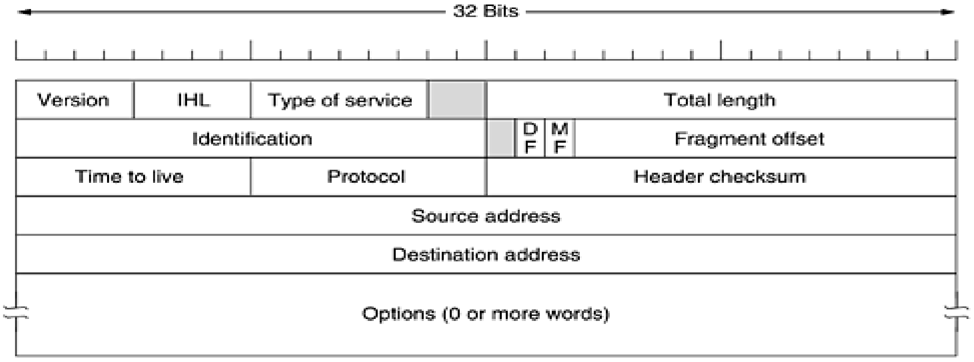


Fig: The IPv4 (Internet Protocol) header

The header length is not constant, a field in the header, IHL, is provided to tell how long the header is, in 32-bit words. The maximum value is 5,, which applies when no options are present. The maximum value of this 4-bit field is 15, which limits the header to 60 bytes and thus the options field to 40 bytes.

The type of service field is one of the few fields that have changed its meaning over the years. The 6-bit filed contained (from left to right), a three bit precedence field and three flags, D, T and R. the precedence field was a priority from 0 (normal) to 7 (network control packet). The three flag bits allowed the host to specify what it cared most about from the set (Delay, Throughput, Reliability).

The total length includes everything in the datagram both header and data. The maximum length is 65,535 bytes.

The identification field is needed to allow the destination host to determine which datagram a newly arrived fragment belongs to.

Next comes an unused bit and then two 1-bit fields. DF stands for Don't Fragment. It is an order to the routers not to fragment the datagram because the destination is incapable of putting the pieces back together again. MF stands for More Fragment. All fragments except the last one have this bit set. It is needed to know when all fragments of a datagram have arrived.

The fragment offset tells where in the current datagram this fragment belongs. There is a maximum of 8192 fragments per datagram, giving a maximum datagram length of 65,536 bytes.

The time to live field is a counter used to limit packet lifetimes. It is supposed to count time in seconds, allowing a maximum lifetime of 255 sec.

The protocol field tells it which transport process to give it to. TCP is one possibility, but so are UDP and some others.

The header checksum verifies the header only. Such a checksum is useful for detecting errors generated by bad memory words inside a router.

The source address and destination address indicate the network number and host number.

The options field is padded out to a multiple of four bytes. Five options defined as:

Option	**Description**
Security	Specifies how secret the datagram is
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed
Record route	Makes each router append its IP address
Timestamp	Makes each router append its address and timestamp

Fig 5.54 some of the IP options

Explain the IP addresses. (5 marks)

Every host and router on the Internet has an IP address, which encodes its network number and host number. The combination is unique; no two machines on the Internet have the same IP address. An IP addresses are 32 bits long and are used in the source address and destination address fields of IP packets. It must have two IP addresses. In practice, most hosts are on one network and have one IP address.

IP addresses were divided into five categories listed in Fig 5.55. This allocation has come to be called class full addressing. The class A, B, C and D formats for up to 128 networks with 16 million host each, 16, 384 network with up to 64k hosts, and 2 million networks (e.g. LANs) with up to 256 hosts each. Also supported is multicast, in which a datagram is directed to multiple hosts. Over 500,000 networks are now connected to the Internet, and the number grows every year. Network numbers are managed by a nonprofit corporation called ICANN (Internet Corporation for Assigned Names and Numbers).

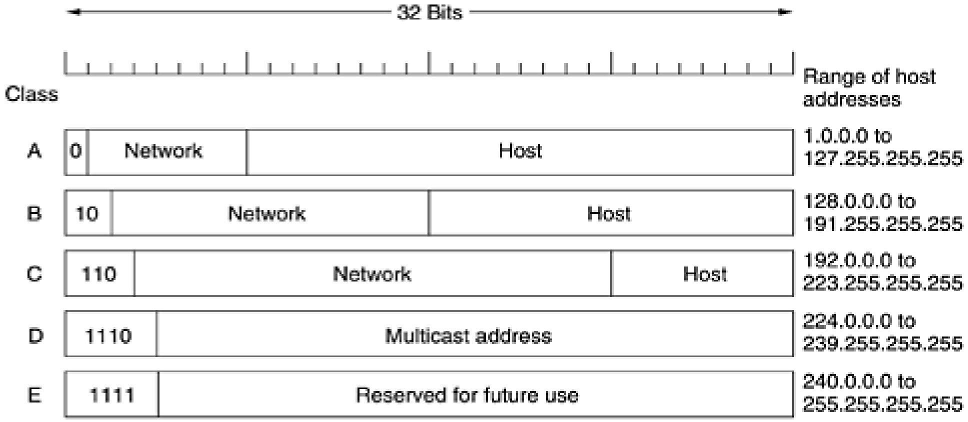

32 Bits
Class
Range of host addresses
A
0
Network
Host
1.0.0.0 to 127.255.255.255
B
10
Network
Host
128.0.0.0 to 191.255.255.255
C
110
Network
Host
192.0.0.0 to 223.255.255.255
D
1110
Multicast address
224.0.0.0 to 239.255.255.255
E
1111
Reserved for future use
240.0.0.0 to 255.255.255.255

Fig: IP address formats

Networks addresses, which are 32-bit number, are usually written in doted decimal notation. Each of the 4 bytes is written in decimal, from 0 to 255. For example, the 32-bit hexadecimal address 10290614 is written as 192.41.6.20. The lowest IP address is 0.0.0.0 and the highest is 255.255.255.255.

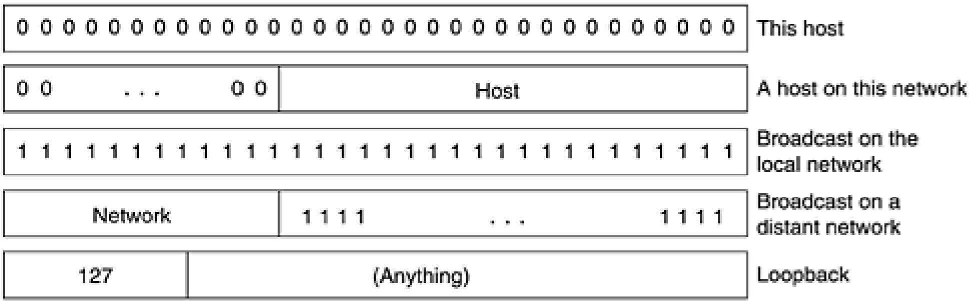


Fig: Special IP addresses

The values 0 and -1 (all 1s) have special meanings, as shown in fig 5.56. 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.

The IP address 0.0.0.0 is used by hosts when they are being booted. IP addresses with 0 as network number refer to the current network. These addresses allow machines to refer to their own network without knowing its number. The address consisting of all 1s allows broadcasting on the local network, typically a LAN. The addresses with a proper network number and all 1s in the host field allow machines to send broadcast packets to distant LANs anywhere in the Internet. Finally, all addresses of the form 127.xx.yy.zz are reserved for loopback testing

.

Explain the Internet multicasting (5 marks)

It is useful for a process to be able to send to a large number of receivers simultaneously. Examples are updating replicated, distributed databases, transmitting stock quotes to multiple brokers and handling digital conference (i.e. multiparty) telephony calls.

IP supports multicasting, using class D addresses. Each class D address identifies a group of hosts. Twenty-eight bits are available for identifying groups, so over 250 million groups can exist at the same time. When a process sends a packet to a class D address, a best-efforts attempt is made to deliver it to all the member of the group addressed, but no guarantees are given. Some numbers may not yet the packet.

Two kinds of group addresses are supported: permanent addresses and temporary ones. A permanent group is always there and does not have to be set up. Each permanent group has a permanent group address. Some examples of permanent group addresses are:

224.0.0.1 All systems on a LAN
224.0.0.2 All routers on a LAN
224.0.0.5 All OSPP routers on a LAN
224.0.0.6 All designated OSPF routers on a LAN

Temporary groups must be created before they can be used. A process can ask its host to join a specific group. It can also ask its host to leave the group. When the last process on a host leaves a group, that group is no longer present on the host. Each host keeps track of which groups its processes currently belong to.

About once a minute, each multicast router sends a hardware (i.e. data link layer) multicast to the hosts on its LAN (address 224.0.0.1) asking them to report back on the groups their processes currently belong to. Each host sends back responses for all the class D addresses it is interested in. These query and response packets use a protocol called IGMP (Internet Group Management Protocol)

Explain the Mobile IP. (5 marks)

Every IP address contains a network number and a host number. For example, consider the machine with IP address 160.80.40.20/16. The 160.80 gives the network number (8272 in decimal), the 40.20 is the host number (10260 in decimal). Routers all over the world having routing tables telling which line to use to get to network 160.80. Whenever a packet comes in with a destination IP address of the form 160.80. xxx.yyy, it goes out on that line.

The machine with that address is carted off to some distant site, the packets for it will continue to be routed to its home LAN (or router). The owner will no longer get e-mail, and so on. Giving the machine a new IP address corresponding to its new location is unattractive because large numbers of people, programs and databases would have to be informed to the change.

The major ones were:

- Each mobile host must be able to use its home IP address anywhere.
- Software changes to the fixed hosts were not permitted.
- Changes to the router software and tables were not permitted.
- Most packets for mobile hosts should not make detours on the way.
- No overhead should be incurred when a mobile host is at home.

Every site that wants to allow its users to roam has to create a home agent. Every site that wants to allow visitors has to create a foreign agent. When a mobile host shows up at a foreign site, it contacts the foreign host there and registers. The foreign host then contacts the user's home agent and gives it a care-of-address.

When a packet arrives at the user's home LAN, it comes in at some router attached to the LAN. The router then tries to locate the host in the usual way, by broadcasting an ARP packet asking, for example: what is the Ethernet address of 160.80.40.20? The home agent responds to this query by giving its own Ethernet address. The router then sends packets for 160.80.40.20 to the home3 agent. In turn, tunnels them to the care-of-address by encapsulating them in the payload field of an IP packet addressed to the foreign agent. The foreign agent then decapsulates and delivers them to the data link address of the mobile host. The home agent gives the care-of-address to the sender, so feature packets can be tunneled directly to the foreign agent.

Imagine an airplane with an on-board Ethernet used by the navigation and avionics computers. On this Ethernet is a standard router that talks to the wired Internet on the ground over a radio link. One fine day, some clever marketing executive gets the idea to install Ethernet connectors in all the arm rests sp passengers with mobile computers can also plug in.

We have two levels of mobility: the aircraft's own computers; which are stationary with respect to the Ethernet, and the passengers' computers, which are mobile with respect to it.

End of UNIT - IV

UNIT – V

Explain the services provided in the upper layer in transport layer. (5 marks)

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 transport entity can be located in the operating system kernel, in a separate user process, in a library package bound into network applications, or conceivably on the network interface card. The relationship of the network, transport and application layers is shown in fig.

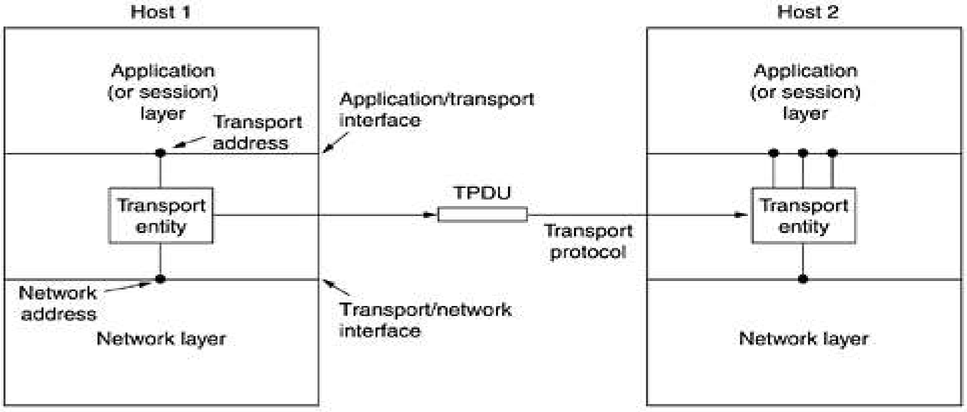


Fig: The network, transport, and application layers

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. Connections have three phases: establishment, data transfer and release. Addressing and flow control are similar in both layers. The connectionless transport service is also very similar to the connectionless network service.

The obvious question is then this: If the transport layer service is so similar to the network layer service, why are there two distant layers? Why is one layer not adequate? The answer is subtle, but crucial. The transport code runs entirely on the users' machines, but the network layer mostly runs on the routers, which are operated by the carrier. What happens if the network layer offers inadequate service? Suppose that it frequently loses packets? What happens if routers crash from to time?

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. The transport service primitive's can be implemented as calls to library procedure in order to make them independent to the network service primitives. The network service calls may vary considerably from network to network (e.g., connectionless LAN service may be quite different from connection-oriented WAN service). By hiding network service behind a set of transport service primitives, changing the network service merely requires replacing one set of library procedures by another one that does the same thing with a different underlying service.

Write short notes on the transport service primitives? (10 marks)

To allow users to access the transport service, the transport layer must layer provide some operations to applications programs, that is, the transport service interface. Each transport service has its own interface. 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. Real networks can lose packets, so the network service is generally unreliable.

The (connection-oriented) transport layer service is reliable. Real networks are not error-free, but that is preciously the purpose of the transport layer – to provide a reliable service on top of an unreliable network.

Consider two processes connected by pipes in UNIX. They assume the connection between them is perfect. They do not want to know about acknowledgements, lost packets, congestion, or anything like that.

What they want is a 100 percent reliable connection. Process A puts data into one end of the pipe, and process B takes it out of the other.

A second difference between the network service and transport service is that the services are intended for. The network service is used only by the transport entities. Few users write their own transport entities, and thus few users or programs ever see the bare network services.

Consider the five primitives listed in fig. This transport interface is truly bare bones, but it gives the essential flavor of what a connection-oriented transport interface has to do. It allows application programs to establish, use, and then release connections, which is sufficient for many applications.

Primitive	**Packet Sent**	**Meaning**
LISTEN	(None)	Block until some process tries to connect
CONNECT	CONNECTION REQUEST	Actively attempt to establish a connection
SEND	DATA	Send Information
RECEIVE	(None)	Block until a DATA packet arrives
DISCONNECT	DISCONNECT REQUEST	This side wants to release the connection

Fig: The primitives for a simple transport service

Consider an application with a server and a number of remote clients. To start with, the server executes a LISTEN primitive, typically by calling a library procedure that makes a system call to block 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.

TPDU (Transport Protocol Data Unit) for messages sent from transport entity to transport entity. TPDUs (exchanged by the transport layer) are contained in packets (exchanged by the network layer). 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.

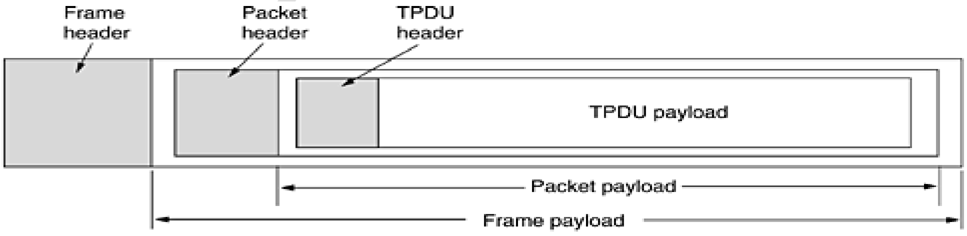


Fig: Nesting of TPDUs, packets, and frames

Getting back to our client-server example the client's CONNECT call causes a CONNECTION REQUEST TPDU 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 interest in handling requests). It then unblocks the server and sends a CONNECTION ACCEPTED TPDU back to the client. When this TPDU arrives, the client is unblocked and the connection is established.

Data can now be exchanged using the SEND and RECEIVE primitives. In this simplest form, either party can do a (blocking) RECEIVE to wait for the other party to do a SEND. When the TPDU arrives, the receiver is unblocked. It can then process the TPDU and send a reply.

When a connection is no longer needed, it must be released to free up table space within the two transport entities. Disconnection has two variants: asymmetric and symmetric. In the asymmetric variant, either transport user can issue a DISCONNECT primitive, which results in a DISCONNECT TPDU being sent to the remote transport entity. Upon arrival, the connection is released. In the symmetric variants, each direction is closed separately, independently of the other one. When one side does a DISCONNECT, that means it has no more data to sent but it is small willing to accept data from its partner. In this model, a connection is released when both sides have done a DISCONNECT.

Write short notes on Berkeley sockets. (5 marks)

The socket primitives used in Berkeley UNIX for TCP. These primitives are widely used for internet programming. They are listed below in fig.

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

Fig: The socket primitives for TCP

The first four primitives in the list are executed in that order by servers. The socket primitive creates a new end point and allocates table space for it within the transport entity. Newly-created sockets do not have network addresses. These are assigned using the BIND primitive. Once a server has bound an address to a socket, remote clients can connect to it.

The LISTEN call, which allocates space to queue incoming calls for the case that several clients try to connect at the same time. To block waiting for an incoming connection, the server executes an ACCEPT primitive. When a TPDU asking for a connection arrives, the transport entity creates a new socket with the same properties as the original one and returns a file descriptor for it. ACCEPT returns a normal file descriptor, which can be used for reading and writing in the standard way, the same as for files.

The CONNECT primitive blocks the caller and actively starts the connection process. Both sides can now use SEND and RECV to transmit and receive data over the full-duplex connection. Connection release with sockets is symmetric. When both sides have executed a CLOSE primitive, the connection is released.

Discuss about the elements of transport protocols. (10 marks)

The transport service is implemented by a transport protocol used between the two transport entities. These differences are due to major dissimilarities between the environments in which the two protocols operate, as shown in the fig.

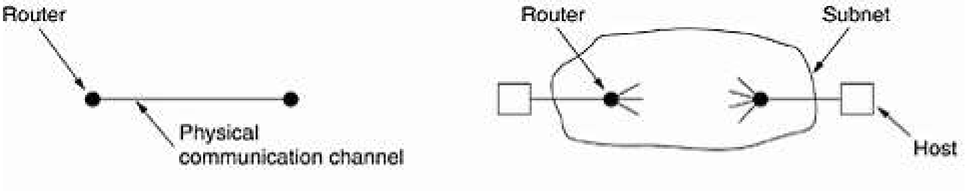


Fig: (a) Environment of the data link layer. (b) Environment of the transport layer.

All the data link layer, two routers communicate directly via a physical channel, whereas qt the transport layer, this physical channel is replaced by the entire subnet.

For one thing, in the data link layer, it is not necessary for a router to specify which router it wants to talk to each outgoing line uniquely specifies a particular router. In the transport layer, explicit addressing of destinations is required.

For another thing, the process of establishing a connection over the wire of fig (a) is simple: the other end is always there.

A final difference between the data link and transport layers is one of amount rather than of kind. Buffering and flow control are needed in both layers, but the presence of a large and dynamically varying number of connections in the transport layer may require a different approach than we used in the data link layer.

1) Addressing:

When an application process wishes to set up 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 end points are called ports. In ATM networks, they are called AAL-SAPs. The generic term TSAP (transport service access point). The analogous and points in the network layer (i.e., network layer addresses) are then called NSAPs. IP addresses are examples of NSAPs.

Fig. illustrates the relationship between the NSAP, TSAP and transport connection. Application processes, both clients and server, can attach themselves to a TSAP to establish a connection to a remote TSAP. These connections run through NSAPs on each host, as shown. 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 end points that share that NSAP.

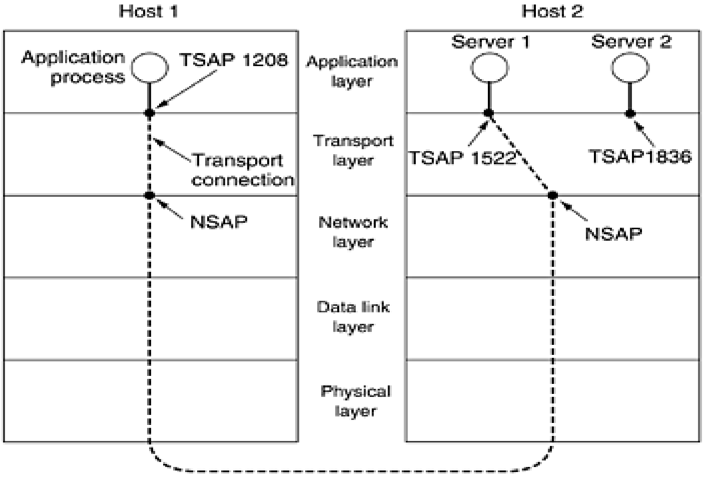


Fig: TSAPs, NSAPs, and transport connections

A possible scenario for a transport connection is as follows:

- A time of day server process on host 2 attaches itself to TSAP 1522 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.
- An application process on host1 wants to find out the time-of-day, so it issues a CONNECT request specifying TSAP 1208 as the source and TSAP 1522 as the destination. This action ultimately results in transport connection being established between the application process on host1 and server1 on host2.
- The application process then sends over a request for the time.
- The time server process responds with the current time.
- The transport connection is then released.

2) Connection Establishment:

The connection establishment looks very simple and straight forward. But the problem occurs when the network can lose, or store duplicate packets. Where there is heavy congestion on the subnet, the acknowledgement will not get back in time. Due to this delay, the packets are retransmitted two or three times. After some time the original packets may arrive at destination following different route. These duplicate create a lot of problem and confusion in real time applications.

The solutions are proposed to avoid duplicate packets, some of them are:

Throw away transport address: If there is disconnection. Each time when transport address is needed, a new one is generated. When connection is released, the address is discarded and never used again.

Using connection identifier: the connection identifier is given to each connection. Connection identifier is a sequence number incremented for each connection established.

The above two approaches fails. Because each transport entity has to maintain certain amount of history information indefinitely. If a machine crashes and loses its memory, it will no longer know which connection identifiers have already been used.

Packet life time can be restricted to known maximum criteria using the following techniques:

- Restricted subnet design
- Putting a hop counter in each packet
- Time stamping each packet

In restricted subnet design, packets are prevented from looping, combined with some way of bounding congestion delay over the longest possible path.

In second method hop count is initialized to some appropriate value and decremented each time, when packet is forwarded. When packet hop count becomes zero, packets are discarded.

In third method, each packet is added with time it was created. The routers agree to discard any packet older than some agreed upon time.

The three protocol scenarios for establishing a connection using three ways handshake is explained with three cases:

Case 1: Normal operation

Case 2: Old duplicate CONNECTION_REQUEST

Case 3: Duplicate CONNECTION_REQUEST and duplicate ACK

Case 1: Normal setup as shown in fig (a)

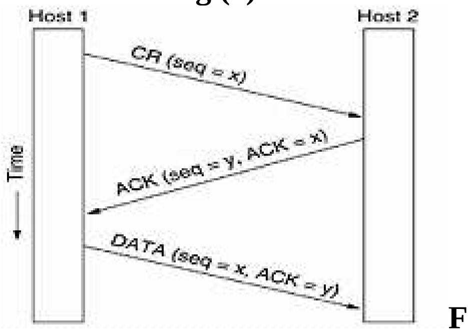


Fig: (a) Normal operation.

Host1 choose a sequence number k and sends a CONNECTION_REQUEST TPDU to host2. Host2 receives CONNECTION_REQUEST TPDU and replies with ACK (acknowledgement) TPDU acknowledgement x and assigns its own initial sequence number y.

Finally, host1 acknowledges host2's choice of an initial sequence number in the first data TPDU. The DATA TPDUs has sequence number x, indicating the connection identifier for that connection.

Case 2: Delayed duplicate CONNECTION_REQUEST TPDUs

First, TPDU is a delayed duplicate CONNECTION_REQUEST from an old connection as shown in the fig (b). This delayed CR TPDU arrives at host 2 without the knowledge of host1.

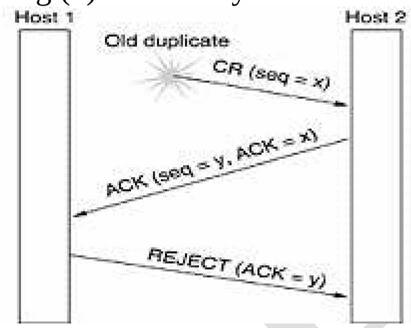


Fig :(b) Old duplicate CONNECTION REQUEST appearing out of nowhere

Host2 replies to this delayed TPDU by sending ACK TPDU to host1. Host1 gets ACK TPDU without sending CR TPDU. This is because host1 does not have the knowledge of CR TPDU have been sent because it is delayed one. So, host1 rejects host 2's attempt to establish a connection, host2 realizes that is was a delayed duplicate and abandons the connection. In this way delayed duplicate does not any damage.

Case 3: Duplicate CONNECTION_REQUEST and duplicate ACK

This situation arises, when both CONNECTION_REQUEST and ACK TPDUs are delayed. This case is shown in the fig c)

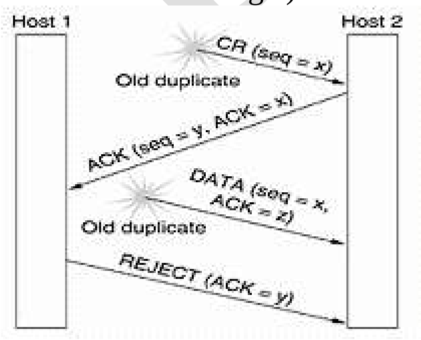


Fig: (c) Duplicate CONNECTION REQUEST and duplicate ACK

Host2 gets a delayed CONNECTION_REQUEST and replies to it. The CONNECTION_REQUEST is acknowledged by host2 by assigning its own sequence number y.

At this point, second delayed duplicate from old connection with acknowledge sequence number z arrives at host2. But, the sequence number y is not acknowledged from host1, because host1 is not aware of the CONNECTION_REQUEST sent to host2. This indicates that both CR and old duplicate DATA with ACK z are duplicate TPDUs. So, host1 send REJECT TPDUs for reject connection to host2.

3. Connection Release:

Connection release is easier than establishing the connection. The connections are released in two ways: asymmetric release and symmetric release.

In asymmetric release, either side can release connection or there may be chances of losing the data.

In symmetric release, the connection is treated as two separate unidirectional connections and requires either side to be released separately. If user on one side release connection, but still it wait for other side to release connection. Here there is no chance of loosing the data.

Four protocol scenarios for releasing a connection are discussed with the following cases:

- Normal connection release
- Final ACK lost
- Response lost
- Both response lost and subsequent DISCONNECTION_REQUEST lost

Case (i) Normal Connection release:

One of the users sends a DISCONNECTION_REQUEST TPDU to initiate the connection release as shown in the fig 6.10(a)

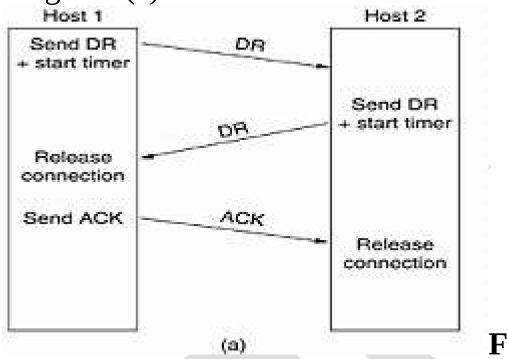


Fig:(a) Normal case of three-way handshake.

When DR arrives at host2, it sends back a DR TPDU indicating its willingness to release. Timers are started when DR TPDU is sent, to keep track of the time. When DR TPDU arrives at host1, the original sender sends back an ACK TPDU and releases the connection. Finally, ACK TPDU arrives at the host2 and also releases the connection.

Case (ii) Final ACK TPDU is lost:

When final ACK TPDU is lost, the timer will save the situation. Host2 will wait for time out. When the timer expires, the connection is released anyway. The case (ii) shown in the fig(b)

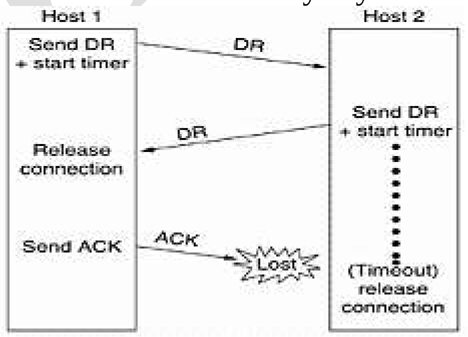


Fig:(b) Final ACK lost.

Case (iii) Response lost:

This is the case, when second DR TPDU is lost. This is shown in the fig (c)

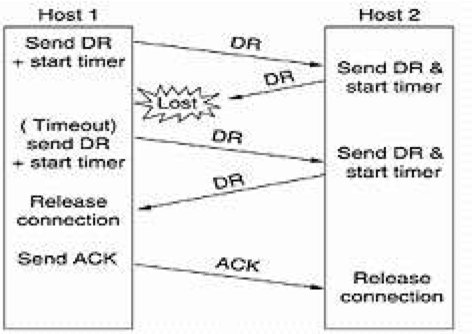


Fig:(c) Response lost.

The host1 indicating the disconnection will not receive the excepted response. This is due to second DR from host2 is lost. At host1, time out occurs and will start all over again. i.e., once again DR is sent. Host2 upon arrival of DR replies back. Host1 receives second DR and releases connections and send ACK to host2. Host2 upon receiving the ACK, host2 releases connection.

Case (iv) Both response and subsequent DR are lost:

In this case, assume all the repeated attempts to retransmit the DR also fails due to lost TPDUs as shown in the fig (d).

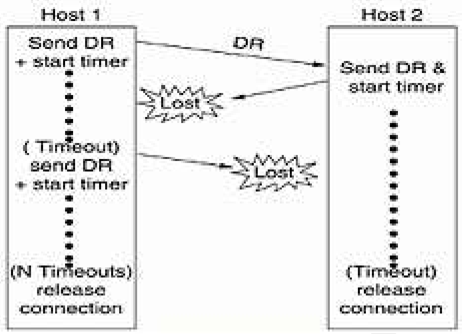

Host 1
Host 2
Send DR + start timer
DR
Send DR & start timer
Lost
(Timeout) send DR + start timer
Lost
(N Timeouts) release connection
(Timeout) release connection

Fig:(d) Response lost and subsequent DRs lost

After N retries, the sender just gives up and releases the connection. The receiver times out and also exits. The sender side will give up and release the connection, while other side does not know about the attempts to disconnect. This situation results in a half open connection.

Half open connection can be avoided by not allowing the sender to give up after N retries. But if other side is allowed to time out, the sender will not release connection for ever. Another way is to have ruled if no TPDUs have arrived for a certain number of seconds, then connection is automatically disconnection. If one side ever disconnects, the other side will detect lack of activity and also disconnect.

4. Flow control and Buffering:

Flow control problem on the transport layer is same as in the data link layer and other issues are different. In both the layer sliding window scheme is used to keep a fast transmitter from over running a slow receiver. Flow control scheme used at data link layer and transport layer are different. In transport layer receiver may maintain a single pool shared by all connections.

When TPDUs comes in, new buffer is acquired for that connection. If buffer is available, the TPDU is accepted, otherwise, it is discarded. Even if TPDU is discarded, no harm because sender is prepared to retransmit lost TPDUs by the subnet. The problem is resources are wasted.

The buffering at receiver side has some problems. It is very difficult to allocate buffer size at the receiver. If the all TPDUs are in same sizes, then it is easy to organize the buffer. Here buffer can be a pool of identically size buffers, with one TPDU per buffer.

The problem with fixed size buffers are: if there is wide variation in TPDU from few characters to thousands of characters, then fixed size buffers fails. For few characters TPDUs space is wasted and for long TPDU, it overflow as shown in the fig (a)

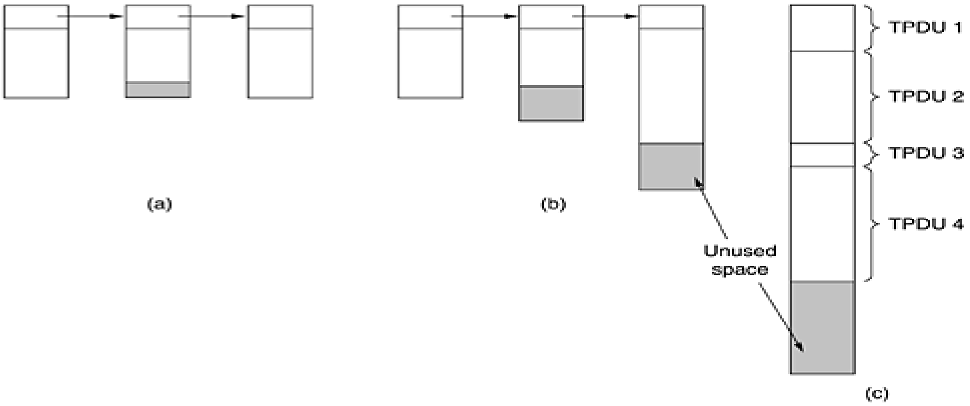


Fig: (a) Chained fixed-size buffers. (b) Chained variable-sized buffers. (c) One large circular buffer per connection.

If the buffer size is chosen equal to the largest possible TPDU, space will be wasted when a short TPDU arrives. If the buffer size is less than the maximum TPDU size, multiple buffers will be needed for long TPDUs, with incoming the complexity.

In this approach, variable sized buffers are used as shown in the fig(b).

Advantage of using variable sized buffers is better memory utilization and disadvantage is more complicated buffer management.

Third approach uses a single large circular buffer per connection as shown in fig(c).

This approach makes good use of memory, provided that all connections are heavily loaded. Utilization of memory is very poor, if the connections are slightly loaded.

5. Multiplexing:

Multiplexing conversation linked to one or distributed to many connections can be called ad multiplexing with respect to transport layers. Multiple users may be using different port to get their required services. The multiplexing can be classified into:

- Upward Multiplexing
- Downward Multiplexing

1) Upward Multiplexing:

In upward multiplexing, the multiple connections are multiplexed on to a single connection as shown in fig(a)

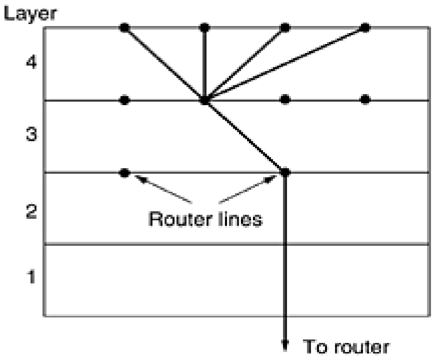


Fig: (a) Upward multiplexing.

For example, all transport connections on the host must use only one network address available.
Four distinct transport connections all use the same network connection (IP address) to the remote host.
If more number of users is multiplexed, then performance is degraded. If there are less users multiplexed, the service will be expensive.

(ii) Downward Multiplexing:

In downward multiplexing, a single connection is split and distributed among multiple connections as shown in the fig(b)

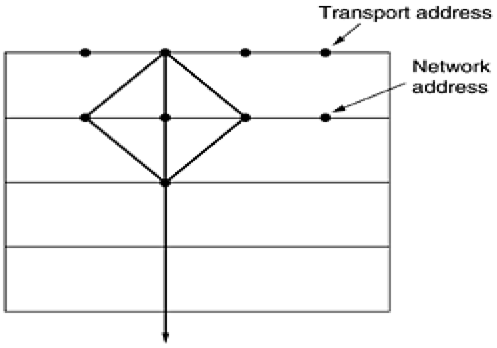


Fig: (b) Downward multiplexing.

For example, if subnet uses virtual circuit internally and imposes maximum data rate on each one. If user needs a more bandwidth than one virtual circuit can provide, open multiple network connections and distribute traffic among them on round robin basis.

Example for downward multiplexing occurs with home users with ISDN line. If the line provides 64 kbps of two separate connections each. Using both of them by dividing the traffic over both lines makes it possible to achieve an effective bandwidth of 128 kbps. So, the throughput can be improved.

6) Crash Recovery:

When hosts and routers are subject to crashes, recoveries from these crashes are difficult. It may be desirable for clients to be able to continue working when servers crash and then quickly reboot.

The difficult task in crash recovery is to recover the previous status of server. One way is to broadcast. TPDU to all the other hosts, announcing that it had just crashed and requesting its client inform it of the status of all open connections.

Clients can be in any one of the state:

S1 -> one TPDU outstanding and
S0 -> no TPDU outstanding

Based on this state information, client must decide whether to retransmit the most recent TPDU. The client should retransmit, if and only if it has an unacknowledged TPDU outstanding S1, when it learns about the crash.

There are some difficulties in this approach. If crash occurs after the acknowledgement has been sent but before the write has been done at server side. But client will receive the acknowledgement and it will be in state S0, when the crash recovery announcement arrives. The client will not retransmit thinking that the TPDU has arrived. This method client and TPDU will be lost. The solution of problem can give as ‘first write and send acknowledgement later’. But if there is crash after write and acknowledgement cannot be sent. But client will be in state S1 and retransmission leads to duplicate TPDU in the output stream to the server process.

Client and server are programmes in different ways, but there are some situations where the protocol fails to recover properly.

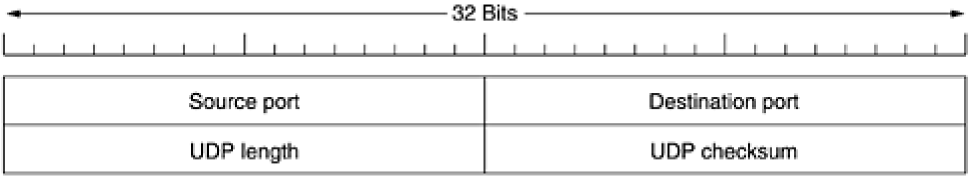
Server can be programmed in two ways:
1) Acknowledge first and
2) Write first

Client can be programmed in four ways:

- always retransmit the last TPDU
- never retransmit the last TPDU
- retransmit only in state S0
- retransmit only in state S1

Explain the internet transport protocol UDP and TCP . (5 marks)

The internet transport protocol has two important protocols in the transport layer. They are UDP and TCP. User datagram protocol is the connection less protocol. UDP transmit segments consisting of 8 byte header followed by the payload. The header is shown below:



Source port	Destination port
UDP length	UDP checksum

Fig: The UDP header

Two ports identify the end points within source and destination machines. With the help of ports transport layer sends segment correctly.

1) Source port and destination port:

This field is required by the receiver, when reply is needed to be sent back to the source machine. Receiving machine copies the incoming segment source port to outgoing segments destination port. Destination port is needed to reach the destination machine.

2) UDP length:

This field is 16 bit in length and used to get the length of UDP datagram. The length includes 8 byte header plus data.

3) Checksum:

This field is also 16 bit and used for error detection. This field is optional and stored as 0 if not computed. Checksum is used to determine whether bits within the UDP segment have been altered due interference or noise in the links as it moves from source to destination.

The internet transport protocols TCP:

The internet's transport layer, connection oriented, reliable protocol is TCP (transmission control protocol). TCP was specifically designed to provide a reliable end to end byte stream over an unreliable internetwork (IP). The internetwork differs from single network because they may have different topologies, bandwidth, delays, packet sizes and other parameters.

Characteristics of TCP:

Connection-oriented: Before actual data transfer, two communicating process must exchange control segments to establish a connection. This means processes must first hand shake each other. Communication is reliable one.

Full duplex: the connection is established in bi-direction. So, data transfer will takes place in both directions.

Point-to-point: the connection is established between the single sender and single receiver. TCP is not good for multicasting.

Explain the remote procedure call. (10 marks)

Sending a message to a remote host and getting a reply back is a lot like making a function call in a programming language. In both cases you start with one or more parameters and you get back a result. This observation has led people to try to arrange request-reply interactions on networks to be cast in the form of procedure calls. Such an arrangement makes network applications much easier to program and more familiar to deal with. For example, just imagine a procedure named get_IP_address (host_name) that works by sending a UDP packet to a DNS server and waiting for the reply, timing out and trying again if one is not forthcoming quickly enough. In this way, all the details of networking can be hidden from the programmer.

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 programmer. This technique is known as RPC (Remote Procedure Call) and has become the basis for many networking applications. The calling procedure is known as the client and the called procedure is known as the server.

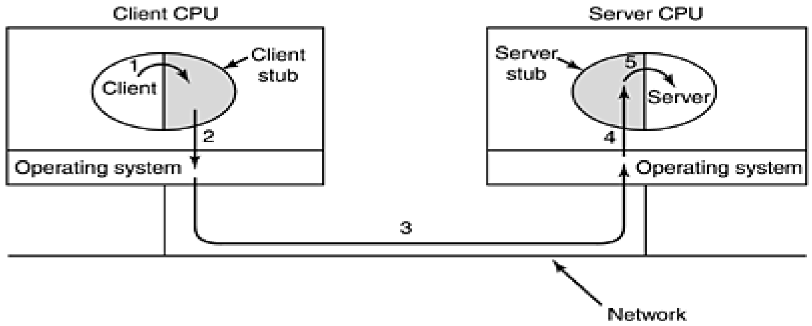


Fig: Steps in making a remote procedure call. The stubs are shaded

The actual steps in making an RPC are shown in Fig. 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 kernel sending the message from the client machine to the server machine. Step 4 is the kernel 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.

Write short notes on the TCP service model. (5 marks)

TCP service is obtained by both sender and receiver creating end point called sockets. Each socket is identified with a pair of addresses. 32 bit IP address of the host and 16 bit number local to host called a port.

Port is the TCP name for TSAP (transport service access point). TCP service is obtained by establishing connection between a socket on sending machine and a socket on receiving machine.

Socket may be used for multiple connections at the same time. Connections are identified by the socket identifiers at both ends will be in form of (socket1, socket2).

Port numbers are assigned from 1024. The port numbers 1024 are reserved for some standard services and they are called well known ports. For example, file transfer FTP uses port 21, email SMTP uses port 25 and internet HTTP uses port 80. List of all well known ports are given at www.iana.org. The list of some of the well known ports are shown in the table.

Port	Protocol	Application
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	Email
69	TFTP	Trivial FTP
79	Finger	Lookup information about user
80	HTTP	World wide web
110	POP3	Remote e-mail access
119	NNTP	USENET news

Table: Reserved Ports.

All TCP connections are full duplex and point to point. Full duplex means that traffic can go in both directions at the same time. Point to point means that each connection has two end points. TCP does not support multicasting or broadcasting so, only unicasting.

TCP connection is byte stream connection. The data is delivered to the receiving process in multiple of bytes. For example, if sending process wants to writes four 512 byte or TCP stream. These data may be delivered to the receiving process as four 512 byte chunks or two 1024 byte chunks or one 2048 byte chunk or in some other way. Receiver cannot detect the units in which the data were written.

Write a note on the TCP protocol. (5 marks)

Every byte in a TCP connection has its own 32 bit sequence number. The sequence numbers used for acknowledgements and for window mechanism are separate. TCP entities exchange the data in the form of segments. TCP segment consists of fixed 20 byte header (plus an optical part) followed by zero or more data bytes. The length of segments will be decided by TCP software. TCP software can split data into one or multiple segments or accumulate data into one segment. There are two limits which restricts the segment size.

First, each segment including the TCP header, must fit in the 65,515 byte IP payload.

Second limitation in each network has a maximum transfer unit (MTU) and each segment fit in the MTU. Generally MTU is 1500 bytes (Ethernet payload size).

TCP entities use sliding window protocol. When a sender transmits a segment and starts timer segment arrives at the destination and receiving TCP entity sends back segment data bearing the acknowledgement number equal to the next end, if timer goes off before receiving the acknowledgements, then sender retransmits the segment again.

Write short notes on TCP segment header. (10 marks)

Every TCP segment begins with a fixed format, 20 byte header. The fixed header may be followed by header options. Segments without data are commonly used for acknowledgements and control messages.

TCP header fields are discussed below:

i) Source port and destination port: these fields identify the local end points of the connection. A 16 bit port number plus its host's 32 bit IP address forms a 48 bit unique end point. The source and destination end points together identify the connection. These port numbers are used for multiplexing/demultiplexing data from/to upper layer applications.

ii) Sequence number and acknowledgement number: these fields are used by the TCP sender and receiver in implementing a reliable data transfer service. Both are 32 bits long because every byte of data is numbered in TCP stream.

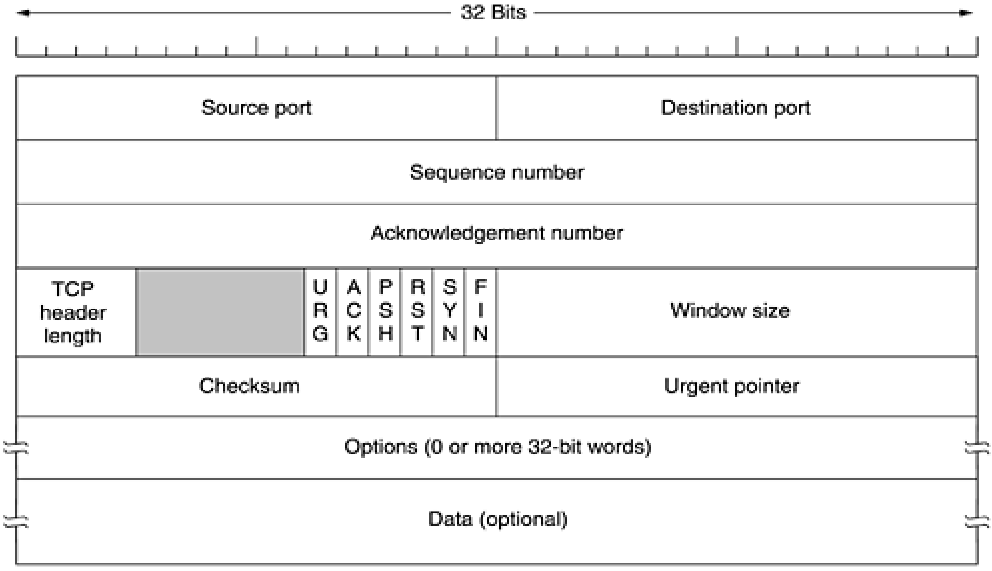


Fig: The TCP header

iii) TCP header length: this field specifies the length of the TCP header in 32 bit words. TCP header can be variable length due to the TCP options field.

iv) URG (urgent pointer bit): URG is set to 1 if the urgent pointer is in use urgent pointer is used to indicate a byte offset from the current sequence number at which urgent data are to be found. This means there is data in this segment that sending side upper layer entity has marked as 'urgent'.

v) ACK (acknowledgement): this bit is set to indicate that the acknowledgement number is valid. If ACK is 0, the segment does not contain an acknowledgement number field is ignored.

vi) PSH (Pushed data): if this bit is set, indicate that the receiver should pass the data to the upper layer immediately.

vii) RST (Reset bit): this bit 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 connection.

viii) SYN: this bit is used to establish connections. Connection request has SYN=1 and ACK =0. This indicates that the piggyback acknowledgement field is not used. Connection request has SYN=1 and ACK=1. This indicates the connection reply does bear an acknowledgement.

ix) FIN (Final release connection): this bit is used to release a connection. This indicates that the sender has no more data to transmit. After closing a connection, the closing process may continue to receive data indefinitely.

x) Window size: this 16 bit window size field is used for flow control. It is used to indicate the number of bytes that a receiver is willing to accept. TCP uses variable sized sliding window.

xi) Checksum: this field is used for extra reliability. The checksum is algorithm is simply to add-up all the 16 bit words in one's complement and one's complement sum is taken. At the receiver end, receiver perform the calculation on the entire segment, including checksum field, the result should be zero.

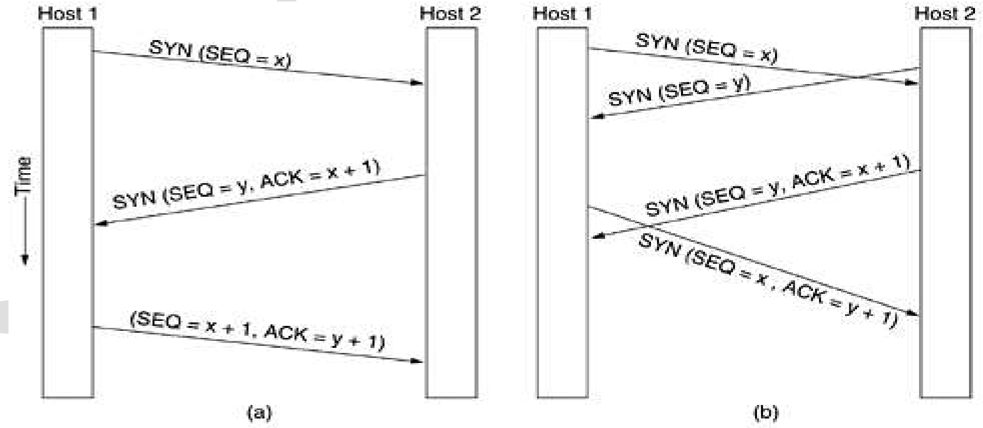
xii) Urgent data pointer field: this field is to indicate a byte offset from the current sequence number at which urgent data are to be found. TCP must inform the receiving side upper layer when urgent data exists and pass it a pointer to the end of the urgent data.

xiii) Options field: this provides extra facilities that are not covered by the regular header. The most important option is the one that allows each host to specify the maximum TCP payload it is willing to accept.

Write short notes on TCP connection establishment and release. (5 marks)

TCP establishes connection by three way handshake messages. Servers wait passively for an incoming by executing the LISTEN and ACCEPT primitives. At the client side, it executes a CONNECT primitive, specifying the IP address and port to which it wants to connect, maximum TCP segment size and some user data.

The CONNECT primitive sends a TCP segment with the SYN bit on and ACK bit off and waits for response. When segment arrives at the destination, TCP entity there checks a process that has executed LISTEN on the port given in the destination port field. If not, destination reply with setting RST bit and rejects the connection.


Host 1
Host 2
SYN (SEQ = x)
Time
SYN (SEQ = y, ACK = x + 1)
(SEQ = x + 1, ACK = y + 1)
(a)
Host 1
Host 2
SYN (SEQ = x)
SYN (SEQ = y)
SYN (SEQ = y, ACK = x + 1)
SYN (SEQ = x , ACK = y + 1)
(b)

(a) TCP connection establishment in the normal case. (b) Call collision.

The sequence of TCP segments sent in normal case is shown in the fig (a). SYN segment consumes one byte of sequence space so that it can be acknowledged without any problems.

If two hosts simultaneously attempt to establish a connection between the same sockets, the sequences of events are shown in the fig(b).

The result of these events is only one connection is established, not two. Because connections are identified by their end points. Only one table entry is made for (x, y).

TCP connection release:

TCP connections are full duplex and it can be seen as pair of simplex connections. Each simplex connection is released independent entity of other.

To release connection, either part can send a TCP segment with FIN bit set, which means no more data to transmit, when FIN is acknowledged, that direction connection is shut down. Data may shut down, the connection is released.

From TCP segments are needed to release a connection. One FIN and one ACK for each direction, it is also possible to combine first ACK and second FIN in the same segment and reduces segment count to three.

Explain the wireless TCP and UDP. (10 marks)

All TCP implementation nowadays assume that timeouts are caused by congestion, not by lost packets. When a timer goes off, TCP slows down and sends less vigorously. Wireless transmission links are highly unreliable. They lose packets all the time. The proper approach to dealing with lost packets is to send them again, and as quickly as possible. Slowing down just makes matter worst. If 20 percent of all packets are lost, then when the sender transmits 100 packets/sec, the throughput is 80 packet/sec. if the sender slows down to 50 packets/sec, the throughput drops to 40 packets/sec.

When a packet is lost on a wired network, the sender should slow down. When one is one on a wireless network, the sender should try harder. When the sender does not know what the network is, it is difficult to make the correct decision.

The path from sender and receiver is heterogeneous. The first 1000 km might be over a wired network, but the last 1 km might be wireless. Indirect TCP, is to split the TCP connection into two separate connections, as shown in the fig.

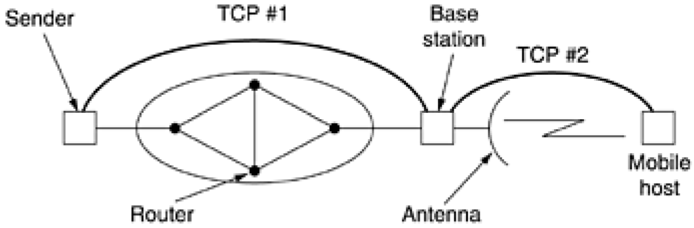


Fig: Splitting a TCP connection into two connections.

The first connection goes from the sender to the base station. The sender one goes from the base station to the receiver. The base station simply copies packets between the connections in both directions.

The advantage is both connections are homogeneous. Timeouts on the first connection can slow the sender down, whereas timeouts on the second can speed it up. Other parameters can also be tuned separately for the two connections. The disadvantage is that it violates the semantics of TCP. Each part of the connection is a full TCP connection; the base station acknowledges each TCP segment in the usual way. Now receipt of an acknowledgement by the sender does not mean that the receiver got the segment, only that the base station got it.

While UDP does not suffer from then same problems as TCP, wireless communication also introduces difficulties fro it. The main trouble is that programs use UDP expecting it to be highly reliable. They know that no guarantees are given but they still expect it to be near perfect. In a wireless environment, UDP will be far from perfect. For programs that can recover from lost UDP messages but only at considerable cost, suddenly going from an environment where messages theoretically can be lost but rarely are, to one in which they are constantly being lost can result in a performance disaster.

Write short notes on Transactional TCP. (5 marks)

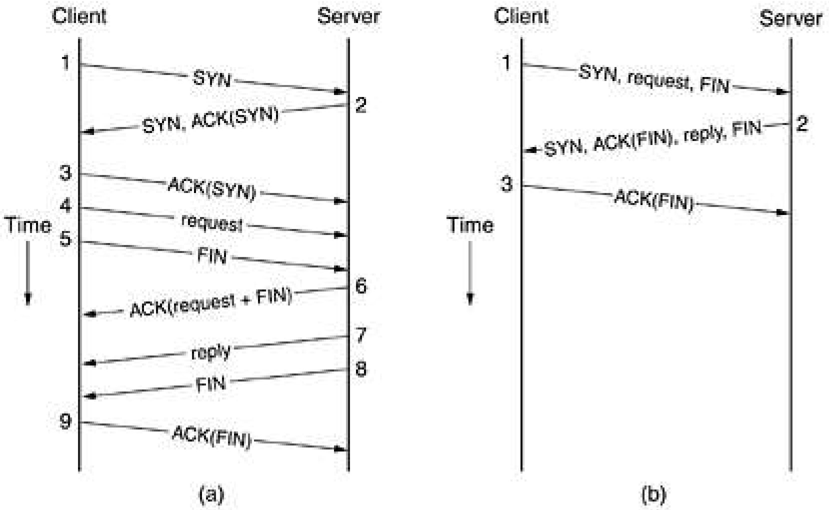
In transactional TCP, remote procedures call as a way to implement client-server systems. If both the request and reply are small enough to fit into single packets and the operation is idempotent, UDP can simply be used. If these conditions are not met, using UDP is less attractive. For example, if the reply can be quite large, then the pieces must be sequenced and a mechanism must be devised to retransmit lost pieces.

The normal sequence of packets for doing an RPC over TCP is shown in the fig (a). Nine packets are required in the best case. The nine packets are as follows:

1) The client sends a SYN packet to establish a connection.
2) The server sends an ACK packet to acknowledge the SYN packet.
3) The client completes the three-way handshake
4) The client sends the actual request.
5) The client sends a FIN packet to indicate that it is done sending
6) The server acknowledges the request and the FIN

7) The server sends the reply back to the client
8) The server sends a FIN packet to indicate that it is also done.
9) The client acknowledges the server's FIN.

In the worst case, the client's request and FIN are acknowledged separately as are the server's reply and FIN.



(a) RPC using normal TCP. (b) RPC using T/TCP

There is some way to combine the efficiency of RPC using UDP with the reliability of TCP. The answer is: Almost. It can be done with an experimental TCP variant called T/TCP (transactional TCP).

To modify the standard connection setup sequence slightly to allow the transfer of data during setup. The T/TCP protocol is illustrated in the fig (b). The client's first packet contains the SYN bit, the request itself, and the FIN. In effect it says: I want to establish a connection, here is the data, and I am done.

When the server gets the request, it looks up or computes the reply, and chooses how to respond. If the reply fits in one packet, it gives the reply of fig (b), which says: I acknowledge your FIN, here is the answer, and I am done. The client then acknowledges the server's FIN and the protocol terminates in three messages.

Explain in detail the Domain Name System. (10 marks)

Domain Name System:

On the internet, each host is identified by address. These addresses are hard and difficult for people to remember. People started preferring names instead of addresses. We need a system that can map a ASCII name to an address.

When internet was small, this mapping was accomplished by a simple file hosts.txt. The host file had two columns comprising name and IP address. When a program or a user wanted to map a name to an address, host refers the host file and the mapping was found. As the internet started growing rapidly it was impossible to have one single host file and every address with a name. The host file would be too large to store in every host. There was also a problem of updating the host file, whenever there is a change. There may be chances of host name conflicts due to distributed managing of host file. To all these problems, DNS (Domain Name System) was invented. DNS is a hierarchical, domain-based naming scheme with a distributed database system. DNS was mainly used for mapping host names and email address to IP addresses.

The DNS name space:

In the internet domains are divided into 200 top levels, where each domain covers many hosts. Each domain is further partitioned into subdomains and these subdomains into further subdomain and so on. A namespace that maps each address to a unique name.

The top-levels domains are classified into two categories: (1) Generic domain and (2) countries domain.

The generic domains are:
.com (commercial)
.edu (educational institutions)
.gov (government)
.int (some international organizations)
.mil (US military forces)
.net (network providers) and
.org (non profit organizations)

The country domains include one entry for every country, for example India's domain is .in, Australia has .au, etc. all these domain can be represented by a tree, as shown below:

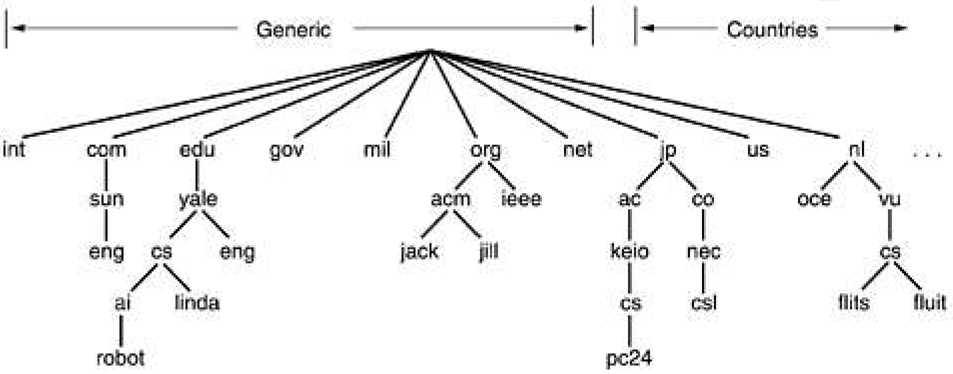


Fig: A portion of the Internet domain name space.

Resource Records:

Each domain name is associated with a record called the resource record. The server database consists of resource records. These records are returned by the server to the client. Server is a DNS server which returns resource records associated with that name. the primary function of DNS is to map domain names onto resource records.

Format of resource record:

Resource record consists of five tuple and all fields are encoded in binary form for efficiency. Resource records are represented as ASCII, text, one line per resource record. The five tuple are domain name, time to live, class, type and value.

1) Domain Name:

This variable length field tells the domain to which this record applies. This field is used as primary search key to satisfy queries.

2) Time – to –live:

This field is 32-bit that defines the number of seconds the answer is valid. If the information is highly stable is assigned with a large value and highly volatile information is assigned with a small value. The receiver can cache the answer for this period of time. If this filed is zero, then the resource record is used only for single transaction and it is not stored for future use.

3) Domain class:

This field identifies domain class of every resource record. For example, internet information, it is always IN and for non-internet information other codes can be used.

4) Domain Type:

This field tells what type of resource record it is. There are various types of resource records are:

Domain type	Meaning	Value
SoA	Start of Authority	Parameter for this zone
A	IP address of a host	32-bit Integer
MX	Mail Exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for IP address

HINFO	Host Description	CPU and OS in ASCII
SRV	Service Available	Defines services available in zone eg. Idap, http, etc.
SIG	Signature	Signature contains data authenticated in a source DNS
TXT	Text information	Text information associated name
WKS	Well known services	Well known services depreciated in favour of SRV

SoA (Start of Authority):

Record starts with SoA which provides the name of primary source information. This information may be name server’s zone or email address of administrator, a unique serial number, various flags and timeouts.

A (Address):

This record is most important record type holds a 32-bit IP address for some host. Each host on the internet is identified or addressed by atleast one IP address. This IP address is used by other machine for communication. Even if network connections are more than one for some hosts, but they will have only one type of A resource per IP address.

MX (Mail Exchange):

The MX record provides the name of the host prepared to accept e-mail for the specified domain. MX record is used because, every machine is not prepared to accept e-mail. It redirects mail to a mail server.

NS (Name Server) record:

The NS records are used to specify the name servers. Every DNS database normally has an NS record for each of the top-level domains.

CNAME (canonical Name) record:

CNAME records will have domain name as value. CNAME records allow aliases to be created. Sometime address will not be correct. For example, a person familiar with Internet naming wants to send a message to his friend whose name is X in the computer science department at iisc. He might guess that x@cs.iisc.edu will work. But the actual address is x@cse.iisc.edu. Making CNAME entry, one can do the job in the following way:

Cs.iisc.edu   864001N   CNAME   cse.iisc.edu

PTR record:

PTR is a regular DNS data type whose interpretation depends on the context in which it is found. PTR is used to associate a name with an IP address. For a given IP address it returns the name of the corresponding machine. This mechanism is known as reverse lookups.

HINFO record:

This record gives the type of machine and operating system a domain corresponds to. It gives the host description with type of CPU and OS.

TXT record:

This record contains uninterpreted ASCII text and allows domains to identify themselves in arbitrary ways.

5) Domain value:

This field can be a number, a domain name or an ASCII string. The semantics depend on the record type.

Name Servers:

The DNS name was divided into non over lapping zones. Each zone contains some part of the true and also contains name servers holding the information about that zone. Zone will have one primary name server and one or more secondary name servers. Primary name servers get their information from a file on its disk and secondary name servers get their information from the primary name servers. The figure shows one of the possible ways to divide the name space.

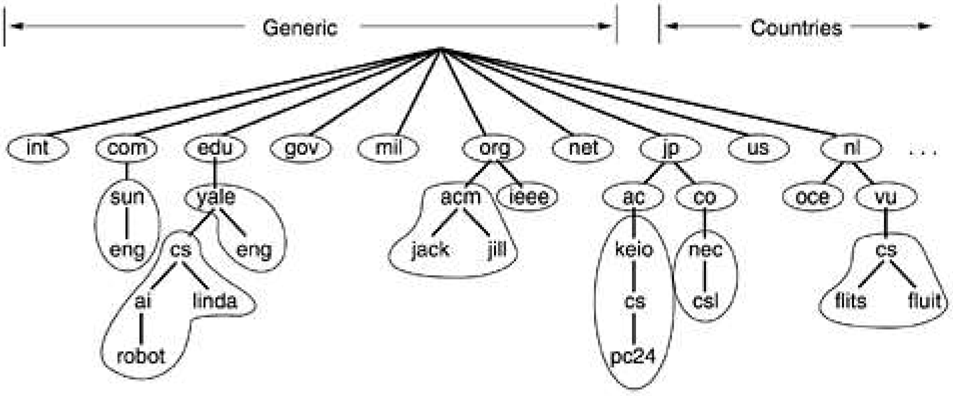


Fig: Part of the DNS name space showing the division into zones

The domain is remote and no information about the requested domain is available locally, then name server sends a query message to the top level name server for the domain requested.

Let us consider the fig. to explain the process of resolving remote name.

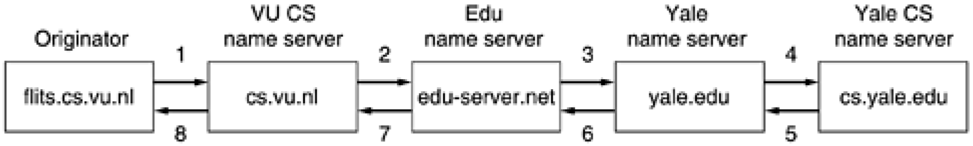


Fig: How a resolver looks up a remote name in eight steps

A resolves on tgb.iete.org wants to know the IP address of the host ravi.iit.cs.edu.

Step 1:

Originator tgb.iete.org sends a query to local name server iete.org. Local servers have never had a query for this domain before and ask near by name servers.

Step 2:

It sends a UDP packet to the server for edu given its database edu.server.net.

Step 3:

Edu.server.net forward the request to the name server for cs.edu..

Step 4:

Edu.server.net forwards the request to iisc.cs.edu which must have authoritative resource records.

Step 5 to Step 8:

Each request is from a client to a server, the resource requested works its way back in step 5 to step 8.

Explain the E-mail. (10 marks)

E-mail or electronic mail is one of the most popular network services. The email system simply consists of file transfer protocols with the convention that the first line of each message, contained in the recipient's address.

There were some limitations and problems of using file transfer protocol. They are:

- It was not possible or difficult to send message group of people.
- No internal structure of messages, which makes computer processing difficult.
- There was no way to intimate the arrival of new email messages to the senders.
- There was no facility of re-directing messages to secretaries, when some one was away on business.
- Poor user interface.
- Not possible to create and send messages containing a combination of text, images, voices and facsimile.

Architecture and Services:

The email system consists of two subsystems: 1) user agents (UA) and 2) message transfer Agents (MTA).

User agents will allow people to read and send mail, whereas message transfer agents more the messages from the source to the destination MTAs are typically system daemons. Daemons are the process which runs in the background and their job is to move email through the system. User agents are program at the client side that provide a command based, menu based or graphical based method for interacting with the email systems.

Email systems support five basic functions. These basic functions are:

1) Compositions:

The process of creating or writing a messages and answers. The email system itself will support to compose a mail. After writing a mail address and other header field can be attached to each message.

2) Transfer:

This refers to transferring a mail from sender to the recipient. We need to establish a connection to the destination or intermediate machine. After transferring the messages the connection can be released. The email system will automatically connects/disconnects without the intervention of the user.

3) Reporting:

This process will inform the sender about the email sent. This information can be whether mail was delivered or rejected or lost. Reporting helps in providing confirmation about the email sent.

4) Displaying:

This support is provided to show or display the email received. People can read their emails. Email cannot be viewed directly. Conversion is required or special viewer tools are needed to get the messages.

5) Disposition:

After reading the mail what the recipient want to do. The mail may be read and deleted or not read or read and saved so on. The emails are saved, whenever it is need it can be reread or retrieved or forwarded.

Addition to the basic services, some email systems advanced features. They are:

1) Mail Boxes:

These are created to store incoming email explicit commands are needed to create and destroy mailboxes, check the contents of mailboxes, insert and delete messages from mail boxes and so on.

2) Mailing list:

The mailing list is a list of email addresses. When a email is sent to this mailing list, the same copies are delivered to everyone on the list.

3) Advanced features:

The advanced features like carbon copies (CC), blind carbon copies (Bcc), higher priority email, encrypted email, automated reply email and so on are developed.

The User Agents:

User Agents (UA) is a part of e-mail systems used at client side. A user agent is a program that accepts a variety of commands. These commands are used to compose, receive, send, delete and move mails to a folder, etc.

Send e-mail:

Email can be sent through User Agent (UA) by creating mail that looks very similar to postal or snail mail. It has an envelope and a message. A user must provide destination address, message and other parameters. The message can be prepared by text editor or work processing like program which is built into the user agent.

The envelope contains the sender address, receiver address and other related information. The message contains the header and the body. The header of the message contains the sender, the receiver and subject of the message. The body contains the actual information to be read by the receipt.

The destination address of the receipt must be in the form of username@dns-address.

Receive e-mail:

User agent checks the mail boxes periodically for incoming email. If a user has a mail in mailbox then the UA informs the user first by giving a notice or number of messages in the mailbox. If the user is ready to read the mail, a list is displayed in which each line contains a summary of the information about a particular message in the mailbox as shown below:

Sl.no	Flags	Sender address	Subject	Size
1	K	Raj	Hello	413K
2		Ravi	Conference	20k
3	KA	roopa@yahoo.com	Re:CSE Dept	612K

4	KF	Raghu	Request	212K

Table: Screenshot of the contents of a mail box

The first field is the message number. Second field contain flags K, A and F. flag K indicates that message is not new and was read already. Flag KA indicates that message is already read and answered. Flag KF indicates that message was read and forwarded to someone. The third field tells who has sent the message. The field may contain only first name or email address or full names. The next field, subject gives the brief summary of what the message is about. Finally, the last field tells the size of the message is bytes.

Message Formats:

The format of the email messages which is described in RFC 822. The message consist of: 1) primitive envelope, 2) header fields 3) blank line and 4) message body.

The header fields related to the message transport have the following fields as shown in below:

Header	**Meaning**
To	Filed gives email addresses of the primary recipient(s)
Cc	Gives the addresses of any secondary recipient(s)
Bcc	Email addresses for blind carbon copies
From	Who wrote or created the message
Sender	Email address of the actual sender
Receiver	Line added by each transfer agent along the route
Return-path	Can be used to identify a path black to the sender

Table: RFC 822 Header fields

The RFC 822 headers are described below:

1) To field:

This field gives the email address of the primary recipient (to whom message has to be sent).

2) CC field:

This field gives the email addresses of the any secondary recipients. Cc stand for carbon copy. There is no specific distinction between the primary and secondary recipients.

3) Bcc field:

This field is referred as blind carbon copy, it is similar to cc field, except this line is deleted from all the copies sent to the primary and secondary recipients. So that primary and secondary recipients cannot know the copies sent from Bcc field.

4) From field:

This field tells who wrote the mail or from whom message has been received.

5) Sender field:

This field tells who has sent the mail. For example, boss may write a message, but his assistant may be one who actually sends it. In this case boss would be listed in from field and his assistant in the sender field.

6) Received field:

This field is added by each message transfer agent along the way. The line contains the agent’s identify, the date and time the message was received.

7) Return-path:

This field is added by the final message transfer agent and was used to tell how to get back to the sender.

In addition, RFC 822 messages may also contain a variety of header fields. The important fields are listed on the table below:

Header	**Meaning**
Date	Date and time the message was sent
Reply to	Email address to which replies should be sent
Message-id	Unique number for referring this message later
In-reply to	Message-id of the message to which this is a reply
References	Other relevant message-ids
Keywords	User chosen keywords

Subject	Short summary of the message for the one-line display

Write short notes on Multipurpose Internet Mail Extensions. (5 marks)

MIME:

There are some limitations in the message format of RFC 822. on the internet, there were some problems in sending and receiving with:

- Messages in languages with accents (e.g. French and German)
- Messages in non-Latin alphabets (e.g. Hebrew and Russian)
- Messages in language without alphabets (e.g. Chinese and Japanese)
- Messages cannot be used to send binary files
- Messages with audio or video or images.

The solution was proposed in RFC 1341 and updated in RFC 2045-2049. This solution is called MIME.

MIME continued to the same RFC 822 format, but added structure to the body and defined encoding rules for non-ASCII messages. All MIME messages can be sent using mail program and protocols. MIME is not a mail protocol and it is only extension of SMTP.

MIME defines five messages headers, as shown below:

Header	**Meaning**
MIME-version	Identifies version of the MIME used
Content-Description	Human readable string tell what is the message
Content-id	Unique identifier
Content-transfer-encoding	Method to encode body for transmission
Content-type	Type and format of the content

MIME Version

The header tells the useragent receiving the message that is dealing with a MIME and which version of MIME it uses. E.g. MIME-version 7.1

Content-Description:

This header defines whether the body of the message is image, audio or video. The recipient will know whether it is wroth decoding and reading the message. E.g. content-description <description>

Content-ID:

This header uniquely identifies the content of the message. Content-id follows the same format as the standard message-id header.

Content-transfer-encoding:

This header defines the method to encode the message into binary form for transmission through the network. Five schemes are provided to encode which is shown below:

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

Content-Type:

This header is used to specify the type or nature of the message body. The content type will be further content subtype. They are separated by a slash. Depending on the subtype. The header may contain other parameters.

Format: content-type: <type/subtype: Parameters>

MIME types and subtypes are listed below:

Type	**Subtype**	**Meaning**
Text	Plain Enchriched	Unformatted text Text with simple formatting commands
Message	RFC 822 External body Partial	Body is an encapsulated message Body is a reference to another message Body is a fragment of a bigger message
Image	JPEG GIF	Image is in JPEG format Image is in GIF format
Audio	Basic	Single channel encoding of voice at 8KHz
Video	MPEG	Video is in MPEG format (moving picture)
Application	Octet-stream Postscript	General binary data (uninterrupted) Printable document in postscript

Message Transfer: Message transfer system is related with sending messages from originator to the recipient. The connection is established from the source machine to the destination machine. After or once connection was established, messages can be transferred. The TCP/IP protocol that supports email on the Internet is called SMTP (Simple Mail Transfer Protocol).

SMTP (Simple Mail Transfer Protocol):

SMTP is a simple ASCII protocol, which uses TCP connection with part 25 of the destination machine, email daemon (background process) listen to the port 25, accepts incoming connections and transfer messages from them into the appropriate mailboxes. If the message cannot be delivered to the intended recipient, then error report of undeliverable message is returned to the sender or originator.

A sample illustration of transferring a message from boy@abcd.com to girl@xyz.com is given in steps. The line starting with C sent by the client and S by the server.

```
S:  220              xyz.com          //SMTP service ready
C:  HELLO          abcd.com   //Command from client
S:  250              xyz.com          //Says hello to abcd.com
C:  MAIL FROM:   <body@abcd.com>
S:  250              sender ok
C:  RCPT TO:       <girl@xyz.com>
S:  250 receipt ok                   // only one RCPT command because only one receipient
C:  DATA
S:  354 sendmail;    //end with “.” On a line by itself
C:  from: boy@abcd.com
C:  to:  girl@xyz.com
C:  MIME-Version: 1.0
C:  message-id:        <0703716182. BA01474@abcd.com>
C:  content-type: Multipart/alternative
C:  subject: wishes
C:  Happy Birthday to you
C:   -
C:  S: 250 message accepted
C:  QUIT
S:   221 xyz.com closing connection
```

There are some limitation of SMTP protocol, they are:

- Some older implementations cannot handle message exceeding 64 KB.
- If the client and server have different timeouts, one of them may give up while other is still busy, unexpectedly terminates the connection.
- Infinite mails forms can be generated which increases email traffic.

Final Delivery:

SMTP establishes a TCP connection to the receiver and then transfer email over it. If the recipient is not online, then connection cannot be established and email is not delivered. One solution to this problem is to have a Message Transfer Agent (MTA) on ISP (Internet Service Provider) machine accept email for its customer and store it in their mail boxes on ISP machine.

There are currently two mail access protocols: Post Office Protocol (POP3) and Internet Mail Access Protocol (IMAP)

POP3: (Post Office Protocol Version 3)

POP3 protocol is used to pull or receive e-mail from the ISP's message transfer agent and allow email to be copied from the ISP to the user. POP3 is described in RFC 1939. There are two solutions in which POP3 protocol works. They are:

- when both sender and receiver are online (connected)
- when sender is currently online but receiver is not

Case(i):

When both sender and receiver are online the arrangement is shown below:

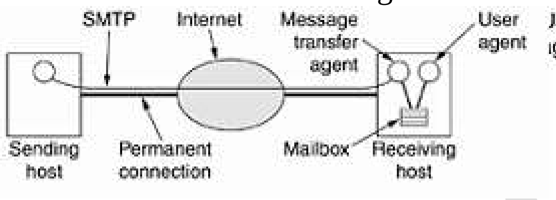


Fig: Sending and reading mail when the receiver has a permanent Internet connection and the user agent runs on the same machine as the message transfer agent.

User starts mail reader, in turn mail reader calls up the ISP and establishes a TCP connection with the message transfer agent at port 110. POP3 protocols performs three functions once the connection has been established. The three functions are:

Authorization – deals with user login

Transactions – deals with the user collecting the emails and making them for deletion from the mail box.

Update – cause the emails to be deleted.

Case(ii):

When sender is currently online, but receiver is not. The arrangement is shown below:

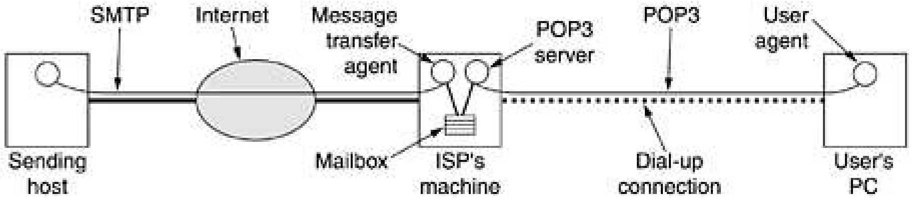


Fig: Reading e-mail when the receiver has a dial-up connection to an ISP.

When sending host is currently online, SMTP establishes TCP connection with the ISP's machine. The email (message) is sent to the message transfer agent (MTA), and mail is transferred to user's respective mailboxes. ISP machine will hold all the messages in user's respective mailboxes. When receiver tries to connect to the ISP's machine via dial up connection POP3 protocols starts working. POP3 server software is installed on user's PC. Through UA (User Agent) user is allowed to connect to POP3 server and starts reading or receiving the mails which are available in mailbox. The problem of sending the messages to offline receiver is solved with the help of ISP's machine.

IMAP (Internet Mail Access Protocol)

IMAP is similar to POP3, but has more features. IMAP is more powerful and complex. POP3 allows all stored messages at each contact and this result in user's email quickly gets speak over multiple machines. To overcome this disadvantages IMAP was developed. IMAP is defined in RFC 2060. IMAP server listens to port 143. Some of the limitations of POP3 are listed below:

- POP3 does not allow the user to organize their mails on the server.
- The user cannot have different folder on the server.
- POP3 does not allow the users to partially check the contents of the mail before downloading.
- IMAP provides extra functions over POP3. They are:
- User can check the email header before downloading it.

- IMAP provides mechanism for creating, destroying and manipulating multiple mailboxes on the server.
- User can create a hierarchy of mail boxes in a folder for email storage.
- User can also download email partially. This feature is useful when bandwidth is limited and email contains multimedia which needs high bandwidth.
- User can also search the contents of the email for a specific character before downloading.
- IMAP can also accept outgoing email for transferring to destination as well as deliver incoming email.

Explain in detail the Cryptography. (5 marks)
Write a note on Substitution Ciphers. (5 marks)
Write a note on Transposition Ciphers. (5 marks)
Cryptography:

Cryptography comes from the Greek words for "secret writing". Four groups of people have used and contributed to the art of cryptography: the military, the diplomatic corps, diarists, and lovers. 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 messages or radio. We assume that the enemy, or intruder, hears and accurately copies down the complete ciphertext. Unlike the intended recipient, he does not know that the decryption key is and so cannot decrypt the ciphertext easily. 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 message before they get to the receiver (active intruder). The art of breaking ciphers, called cryptanalysis, and the art devising them (cryptography) is collectively known as cryptology.

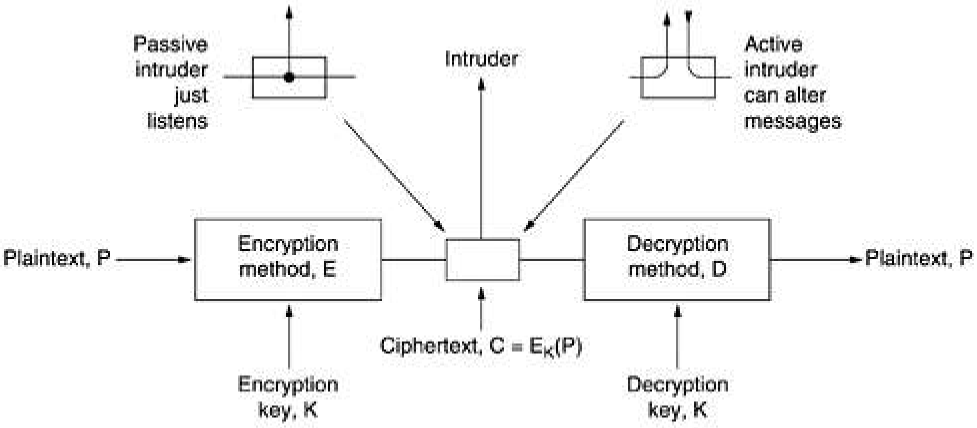


Fig: The encryption model (for a symmetric-key cipher).

It will be often be useful to have a notation for relating plaintext, ciphertext and keys. We will use C= E_k (P) to mean that the encryption of the plaintext P using key k gives the ciphertext C. P = D_K(C) represents the decryption of C to get the plaintext again. It then follows that D_K (E_K(P)) = P.

A fundamental rule of cryptography is that one must assume that the cryptanalyst knows the methods used for encryption and decryption. In other words, the cryptanalyst knows hoe the encryption methods, E and decryption, d, of fig. work in detail. The amount of effort necessary to invest, test, and install a new algorithm every time the old method is compromised (or thought to be comprised) has always made it impractical to keep the encryption algorithm secret. Thinking it is secret when it is not more harm than good.

Substitution Ciphers:

In a substitution cipher each letter or group of letters is replaced by another letter or group of letters to disguise it. One of the oldest known ciphers is the Caesar cipher, attributed to Julius Caesar. In this method, a becomes D, b becomes E, c becomes F, …… and z becomes C. for example, attack becomes DWWDFN. In examples, plaintext will be given in lowercase letters and cipher text in uppercase letters.

Caesar cipher allows the ciphertext alphabet to be shifted by k letters, instead of always 3. In this case k becomes a key to the general method of circularly shifted alphabets. The Caesar cipher may have fooled Pompey, but it has not fooled anyone since.

The next improvement is to have each of the symbols in the plaintext, the 26 letters for simplicity map onto some other letter, for example:

Plaintext: 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
Ciphertext: Q W E R T Y U I O P A S D F G H J K L Z X C V B N M

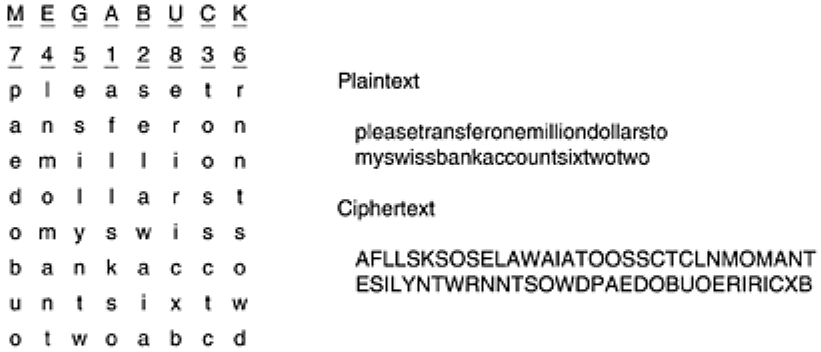
The general system of symbol-for-symbol substitution is called a monoalphabetic substitution, with the key being the 26 letter string corresponding to the full alphabet. For the key above, the plaintext attack would be transformed into the ciphertext QZZQEA.

The might appear to be a safe system because although the cryptanalyst knows the general system (letter-for-letter substitution), he does not know which of the 26! = $4*10^{26}$ possible keys is in use. In contrast with the Caesar cipher, trying all of them is not a promising approach. Even at 1 nsec per solution, a computer would take 10^{10} years to try all the keys.

A small amount of ciphertext, the cipher can be broken easily. The basic attack takes advantages of the statistical properties of natural languages. In English, for example, e is the most common letter, followed by t, o, a, n, i, etc. the most common two-letter combinations or diagrams, are the ia, er, re and an. The most common three-letter combinations or trigrams are the ing, and and ion.

Transposition ciphers:

Transposition ciphers, recorder the letter but do not disguise them. Fig depicts a common transposition cipher, the columnar transposition. The cipher is keyed by a word or phrase not containing any repeated letters. In this example, MEGABUCK is the key. The purpose of the key is to number the columns, column1 being under the key letter closest to the start of the alphabet, and so on. The plaintext is written horizontally, in rows, padded to fill the matrix if need be. The ciphertext is read out by columns, starting with the column whose key letter is the lowest.



M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Plaintext

pleasetransferonemilliondollarsto
myswissbankaccountsixtwotwo

Ciphertext

AFLLSKSOSELAWAIATOOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

Fig: A transposition cipher

To break a transposition ciphers, the cryptanalysis must first be aware that he is dealing with a transposition cipher. By loading at the frequency of E, T, A, O, I, N, etc, it is easy to see of they fit the normal pattern for plaintext. The cipher is clearly a transposition cipher, because in such a cipher every letter represents itself. Keeping the frequency distribution intact.

Transposition ciphers accept a fixed-length block of input and produce a fixed-length block of output. These ciphers can be completely described by giving a list telling the order in which the characters are to be output. For example, the cipher of fig 8.3 can be seen as a 64 characters block cipher. Its output is 4, 12, 20, 28, 36, 44, 52, 60, 5, 13… 62. In other words, the forth input character, a is the first to be output, followed by the twelfth, f, and so on.

Explain the symmetric key algorithms (5 marks)

Symmetric-key algorithms are used the same key for encryption and decryption. Block ciphers take an n-bit block of plaintext as input and transform it using the key into n-bit block of ciphertext.

Fig (a) shows a device, known as a p-box (p stands for permutation), used to effect a transposition on an 8-bit input. If the 8 bits are designed from top to bottom as 01234567, the output of this particular p-box is 36071245. A p-box can be made to perform any transposition and do it at practically the speed of light, since no computation is involved, just signal propagation. This design follows kerckheff’s principle: the attacker knows that the general method is permuting the bits. What he does know is which bit goes where, which is the key.

Substitutions are performed by S-boxes, as shown in fig (b). In this example a 3-bit plaintext is entered and a 3-bit ciphertext is output. The 3-bit input selects one of the eight lines exiting from the first stage and sets it to 1, all the other lines are 0. The second stage is a P-box. The third stage encodes the selected input line in binary again. If the eight octal numbers 01234567 were input one after another, the output sequence would be 24506713. 0 has been replaced by 2, 1 has been replaced by 4, etc.

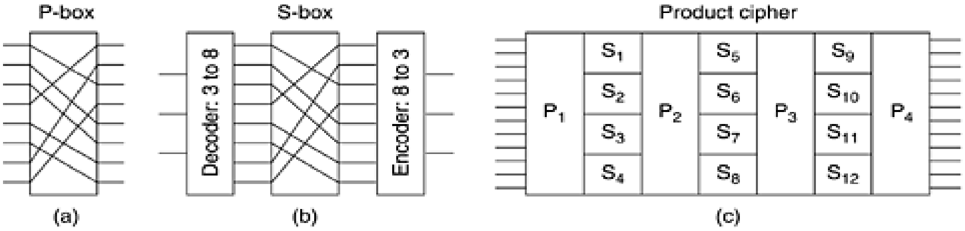


Fig: Basic elements of product ciphers. (a) P-box. (b) S-box. (c) Product

The real power of these basic elements only becomes apparent when we cascade a whole series of boxes to form a product cipher, as shown in fig (c). In this example, 12 input lines are transposed by the first stage (P_1). It would be possible to have the second stage be an S-box that mapped a 12-bit number onto another 12-bit number. Such a device would need 2^{12}=4096 crossed wires in its middle stage. The input is broken up into four groups of 3 bits, each of which is substituted independently of the others.

Explain the DES. (10 marks)

An outline of DES is shown in fig(a), plaintext is encrypted in block of 64 bits, yielding 64 bits of ciphertext. The algorithm, which is parameterized by a 56-bit key, has 19 distinct stages. The first stage is a key-independent transposition on the 64-bit plaintext. The last stage is the exact inverse of this transposition. The stage prior to the last one exchanges the leftmost 32 bits with the rightmost 32 bits. The remaining 16 stages are functionally identical but are parameterized by different functions of the key. The algorithm has been designed to allow decryption to be done with the same key as encryption, a property needed in any symmetric key algorithm. The steps are just run in the reverse order.

The operation of one of these intermediate stages is illustrated in fig (b). Each stage takes two 32-bit inputs and produces two 32-bit output. The left output is simply a copy of the right input. The right output is the bitwise XOR of the left input and a function of the right input and the key for this stage, k_i. All the complexity lies in this function.

The function consists of four steps, carried out in sequence. First, a 48-bit number, E, is constructed by expanding the 32-bit r_{i-1} according to a fixed transposition and duplication rule. Second, E and K_i are XORed together. This output is then partitioned into eight groups of 6 bits each, each of which is fed into a different S-box. Each of the 64 possible inputs to an s-box is mapped onto a 4-bit output. Finally 8*4 bits are passed through a P-box.

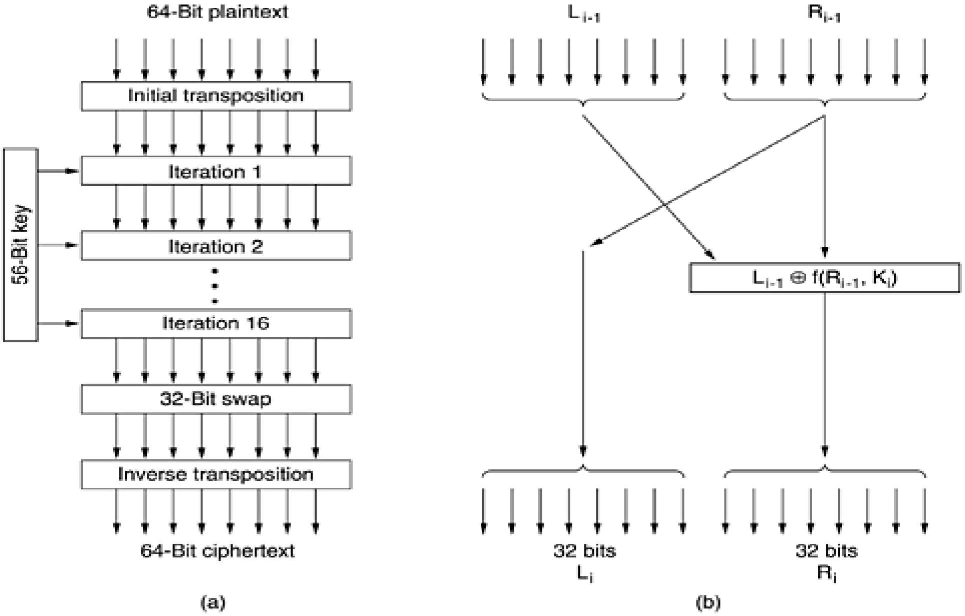


Fig: The data encryption standard. (a) General outline. (b) Detail of one iteration. The circled + means exclusive OR

In each of the 16 iterations, a different key is used. Before the algorithm starts, a 56-bit transposition is applied to the key. Just before each iteration, the key is partitioned into two 28-bit units, each of which is rotated left by a number of bits dependent on the iteration number. K_i is derived from this rotated key by applying yet another 56-bit transposition to it. A different 48-bit subset of the 56-bits is extracted and permuted on each round. A technique that is sometimes used to make DES stronger is called whitening.

Triple DES:

Triple DES, which has since been incorporated in International standard 8732, is illustrated in Fig. Here two keys and three stages are used. In the first stage, the plaintext is encrypted using DES in the usual way with k_1. In the second stage, DES is run in decryption mode, using k_2 as the key. Finally, another DES encryption is done with k_1.

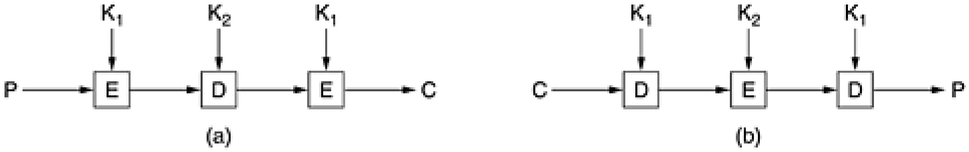


Fig: (a) Triple encryption using DES. (b) Decryption

This design immediately gives risk to two questions. First, why are only two keys used, instead of three? Second, why is EDE (Encrypt Decrypt Encrypt) used, instead of EEE (Encrypt Encrypt Encrypt)? The reason that two keys are used is that even the most paranoid cryptographers believe that 112 bits is adequate for commercial applications for the time being.

The reason for encrypting, decrypting and then encrypting again is backward compatibility with existing single-key DES systems. Both the encryption and decryption functions are mappings between sets of 64-bit numbers. From a cryptographic point of view, the two mappings are equally strong. By using EDE, instead of EEE, a computer using triple encryption can speak to one using single encryption by just setting k_1=k_2. This property allows triple encryption to be phased in gradually, something of no concern to academic cryptographers, but of considerable importance to IBM and its customers.

Explain the public key algorithm of RSA. (10 marks)

In 1976, two researchers at standard university, Diffie and Hellman, [proposed a radically new kind of cryptosystem, one in which the encryption and decryption keys were different, and the decryption key could not feasibly be derived from the encryption key. In their proposal, the (keyed) encryption algorithm, E, and the (keyed) decryption algorithm, D, had to meet three requirements. These requirements can be stated simply as follows:

D(E(P))=P

It is exceedingly difficult to deduce D from E.

E cannot be broken by a chosen plaintext attack.

The first requirement says that if we apply D to an encrypted message, E(P), we get the original plaintext message, P, back. Without this property, the legitimate receiver could not decrypt the ciphertext. The second requirement speaks for itself. The third requirement is needed because, as we shall see in a moment, intruders may experiment with the algorithm to their hearts content. Under these conditions, there is no reason that the encryption key cannot be made public.

RSA:

One good method was discovered by a group at M.I.T. (Rivest et al 1978). It is known by the initials of the three discoverers (Rivest, Shamir, Adleman) RSA. The RSA method is based on some principles from number theory.

Choose two large primes, P and Z (typically 1024 bits)

Compute n=p*q and z=(p-1) *(q-1)

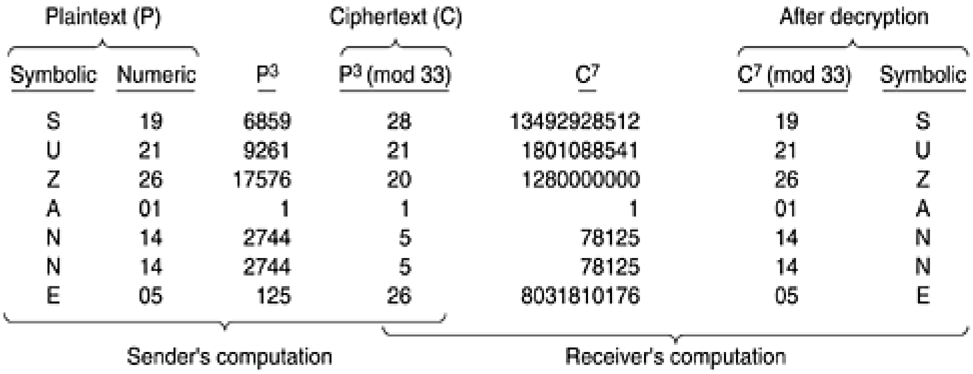
Choose a number relatively prime to Z and call it d.

Find e such that e*d=1 mod Z.

We are ready to begin encryption. Divide the plaintext into blocks, so that each plaintext message, P, falls in the interval 0<p<n. Do that by grouping the plaintext into blocks of k bits, where k is the largest

integer for which 2^k<n is true. To encrypt a message, P, compute $C=P^e$(mod n). To decrypt C, Compute $P=C^d$ (mod n). To perform the encryption, you need e and n. to perform the decryption, you need d and n. the public key consists of the pair (e,n) and the private key consists of (d,n).

A trivial pedagogical example of how the RSA algorithm works is given in fig 8.17. For this example we have chosen P=3 and q=11, giving n=33 and z=20. A suitable value for d is d=7, since 7 and 20 have no common factors. With these choices, e can be found by solving the equation 7e=1(mod 20), which yields e=3. The ciphertext, C, for a plaintext message, P, is given by C=P3 (mod 33). The ciphertext is decrypted by the receiver by making use of the rule p=c7 (mod 33). The figure shows the encryption of the plaintext "SUZANNE" as an example.



Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	P^3 (mod 33)	C^7	C^7 (mod 33)	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E

Sender's computation

Receiver's computation

Fig: An example of the RSA algorithm

Because the primes chosen for this example are so small, P must be less than 33, so each plaintext block can contain only a single character. The result is a monoalphabetic substitution cipher, not very impressive. If instead we had chosen p and $q=2^{512}$, we would have $n=2^{1024}$, so each block could be up to 1024 bits or 129 eight-bit characters, versus 8 characters for DES and 16 characters for AES.

Explain briefly the Social Issues. (10 marks)

1) Privacy:

Do keep have a right to privacy? Good question. The fourth amendment to the U.S. Constitution prohibits the government from searching people's houses, papers, and effects without good reason and goes on to restrict the circumstances under which search warrants shall be issued. Privacy has been on the public agenda for over 200 years, at least in the U.S.

What have changed in the past decade is both the ease with which governments can spy on their citizens and the ease with which the citizens can prevent such spying. In the 18th century, for the government to search a citizen's papers, it had to send out a policeman on a horse to go to the citizen's farm demanding to see certain documents. It was a cumbers one procedure. Telephone companies and Internet provides readily provide wiretaps when presented with search warrants. It makes life much easier for the policeman and there is no danger of falling off the horse.

Anonymous Remailers:

PGP, SSL and other technologies make it possible for two parties to establish secure, authenticated communication, free from third-party surveillance and interference. Sometimes privacy is best served by not having authentication, in fact by making communication anonymous. The anonymity may be desired for point-to-point messages, newsgroups, or both.

Let us consider some examples. First, political dissidents living under authoritarian regimes often wish to communicate anonymously to escape being jailed or killed. Second, wrong doing in many corporate, educational, governmental and other organizations has often been exposed by whistle blower, who frequently prefers to remain anonymously to avoid retribution. Third, people with unpopular social, political, or religious views may with to communicate with each other via email or newsgroup without exposing themselves. Fourth, people may wish to discuss alcoholism, mental illness, sexual harassment, child abuse, or being a number of a persecuted minority in a newsgroup without having to go public.

Many users who wish anonymity chain their requests through multiple anonymous remailers, as shown in fig. Alice wants to send Bob a really, recalls, really anonymous Valentine's Day card, so she uses three remailers. She composes the message, M, and puts a header on it containing Bob's email address. Then she encrypts the whole thing with remailer 3's public key, E_3. To this she prepends a header with remailer 3's email address in plaintext. This is the message shown between remailers 2 and 3 in the figure.

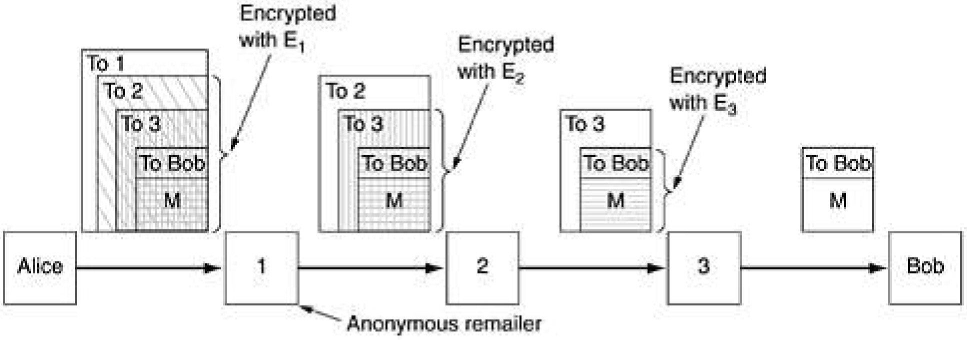


Fig: How Alice uses 3 remailers to send Bob a message.

Then she encrypts this message with remailer 2's public key, E_2 and prepends a plaintext header containing remailer 2's email address. This message is shown between 7 and 2 in Fig. Finally, she encrypts the entire message with remailer 1's public key, E_1, and prepends a plaintext header with remailer 1's email address. This is the message shown to the right of Alice in the figure and this is the message she actually transmits.

When the message hits remailer 1, the outer header is stripped off. The body is decrypted and then e-mailed to remailer 2. Similar steps occur at the other two remailers.

2) Freedom to speech: A second key social issue is freedom to speech, and its opposite, censorship, which is about government wanting to restrict what individuals can read and publish. With the web containing millions of pages, it has become a censor's paradise. Depending on the native and ideology of the regime, banned material may include web sites containing any of the following:

1. Material inappropriate for children or teenagers
2. Hate aimed at various ethics, religious, sexual or other groups.
3. Information about democracy and democratic values.
4. Accounts of historical events contradicting the government's version.
5. Manuals for picking locks, building weapons, encrypting messages, etc.

Steganography: People who want to communicate secretly often try to hide the fact that any communication at all is taking place. The science of hiding messages is called steganography, from the Greek words for "Covered Writing".

The full text of the five plays and a short notice add up to 734,891 bytes. This text was first compresses to about 274 KB using a standard compression algorithm.

To use steganography for undetected communication, dissidents could create a web site bursting with politically-correct pictures, such as photographs of the Great Ceadess, local sports, movie, and television on stars, etc. the pictures would be riddled with steganographic messages. If the messages were first compresses and then encrypted, even someone who suspected their presence would have immense difficulty in distinguishing the messages from white noise.

Images are by no means the only carrier for steganographic messages. Audio files also work fine. Video files have a huge steganographic bandwidth. Even the layout and ordering of tags in an HTML file can carry information.

We have examined steganography in the context of free speech, it has numerous other uses. One common use is for the owners of images to encode secret messages in them stating their ownership rights. It such an image is stolen and placed on a web site, the lawful owner can reveal the steganographic message in count to prove whose image it is. This technique is called water marking.

3) Copyright: A third one is copyright. Copyright is the granting to the creators of IP (Intellectual Property), including writers, artists, composers, musicians, photographers, cinematographers, choreographers and others, the exclusive right to exploit their IP for some period of time, typically the life of the author plus 50 years or 75 years in the case of corporate ownership.

For example, consider a peer-to-peer network in which people share legal files (public domain music, home videos, religious tracts that are not trade secrets, etc.) and perhaps a few that are copyrighted. Assume that everyone is on-line all the time via ADSL or cable. Each machine has an index of what is on the hard disk, plus a list of other members. Someone looking for a specific item can pick a random member and see if he has it. If not, he can check out all the members in that person's list and all the members in their lists and so on. Computers are good at this kind of work having found the item the requester just copies it.

End of UNIT - V