COMPUTER NETWORKS(18BIT42C)

UNIT I: Introduction- The Uses of Computer Networks – Networks hardware – Network software – Reference models.
UNIT II: The Physical Layer **-** Transmission Media – Communication satellites – Wireless transmission –The public switched telephone system
UNIT III: The Data Link layer **-** Data link layer Design Issues – Error Detection and Correction Elementary Data link protocols. Medium Access Sub Layers The channel allocation problem – Multiple access protocols Carrier sense multiple access protocols, collision –free protocols, Limited contention protocols
UNIT IV: The Network Layer – Network Layer Design Issues – Routing Algorithms The optimality principle, shortest path routing, flooding, and distance vector routing, routing for mobile hosts
UNIT V: The Transport Layer – The Transport service – Services provided to the upper layers, transport service primitives – Elements of Transport protocols. Application Layer – DNS – The Domain Name System – Electronic mail – Architecture and services, the user agent.

TEXT BOOK

1. Andrew S. Tanenbaum, “Computer Networks”, 4th Edition, Pearson Education Publ. 2014.

REFERENCE BOOKS

1. Miller, “Data and Network Communications”, Vikas Publ., 2001.
2. William A Shay, “Understanding data communications and Networks”, 2nd Edition, Vikas Publ., 2001.

STUDY MATERIAL

COURSE : II B.Sc IT
SUBJECT : COMPUTER NETWORKS
SEMESTER : IV
UNIT : I

SYLLABUS: Introduction-The use of Computer Networks-Network Hardware-Network Software-Reference Models.

TEXT BOOK :

Andrew S. Tanenbaum, “Computer Networks”, 4th Edition, Pearson Education Publ. 2014.

INTRODUCTION

Computer Network means an “interconnected collection of autonomous Computers”. Two Computers are said to be interconnected if they can exchange information. The connection usually will be based on a communication medium like copper wire, fiber optics etc., Master/Slave relationship in which one computer forcibly start, stop, or control another computer is not a network, the computers should be autonomous.

Difference between a Computer Network and Distributed System.

Computer Network	Distributed System
The Existence of autonomous computers are transparent (they are visible).	The Existence of autonomous computers are NOT transparent (they are not visible).
The autonomous computer performs the operation requested by the user.	The best processor is selected by the operating system for carrying out the operations requested by the user.
The user is aware of his working environment.	The user is not aware of his working environment, which is multiple processor in nature but looks like a virtual uniprocessor.
All operations (allocation of jobs to processors, files to disks, movement of files) are done explicitly.	All operations (allocation of jobs to processors, files to disks, movement of files) are done automatically without the user’s knowledge.
Regulation software is enough for computer networks.	A software that gives a high degree of cohesiveness and transparency is needed since distributed system is built on top of a network

USES OF COMPUTER NETWORKS

The use of Computer Networks can be divided into three ways.

1. Network for Companies
2. Network for People
3. Social Issues

Network for Companies: Many companies have a substantial number of computers, for examples a company may have separate computers to monitor production, keep track of inventories, do the payroll. Each of these computers may have worked in isolation from the others, but at some point, management may have decided to connect them to extract and correlate information about entire company.

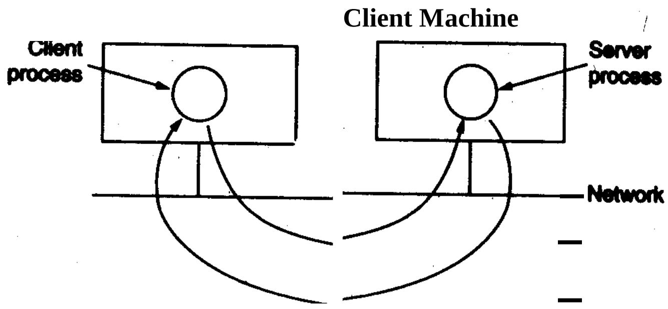
Use of Computer Network for business can be classified as

1. Resource Sharing
2. High Reliability
3. Saving Money
4. Scalability
5. Communication Medium

The goal is **resource sharing**, to make all programs, equipment, and especially data available to anyone on the network without regard to the physical location of the resource and the user.

A second goal is to provide **high reliability** by having alternative sources of supply. All files could be replicated on two or three machines, so if one of them is unavailable (due to a hardware failure), the other copies could be used. In addition, the presence of multiple CPUs means that if one goes down, the others may be able to take over its work, although at reduced performance. Military, banking, air traffic control, nuclear reactor safety, and many other applications, the ability to continue operating in the face of hardware problems is of utmost importance.

Another goal is **saving money**. Small computers have a much better price/performance ratio than large ones. Mainframes (room-size computers) are faster than personal computers, cost more. This imbalance has lead to the idea of connecting personal computers, with data kept on one or more shared **file server** machines. In this model, the users are called **clients**, and the whole arrangement is called the **client server model.** In the client-server model, communication generally takes the form of a request message from the client to the server asking for some work to be done.



Client Server Model

The server then does the work and sends back the reply. Usually, there are many clients using a small number of servers.

Another networking goal is **scalability**, the ability to increase system performance gradually as the workload grows just by adding more processors. With the client-server model, new clients and new servers can be added as needed.

A computer network can provide a powerful **Communication medium** among widely separated employees. Using a network, it is easy for two or more people who live far apart to write a report together. When one worker makes a change to an on-line document, the others can see the change immediately, instead of waiting several days for a letter. Such a speedup makes cooperation among far-flung groups of people easy where it previously had been impossible.

Networks for People: The use of Computer Networks for people can be classified as

1. Access to remote information.
2. Person-to-person communication.
3. Interactive entertainment.

Access to remote information comes in many forms. Information available includes the Arts, Business, Cooking, Government, Health, History, Hobbies, Recreation, Science, Sports, Travel and many others. **Newspapers** will go on-line and be personalized. It will be possible to download the areas of interest of a person say, politics, big fires, scandals involving celebrities, and epidemics. The next step beyond newspapers is the **on-line digital library**. All of the above applications involve interactions between a person and a remote database.

The second broad category of network use will be **person-to-person communication** Electronic mail or **email** is already widely used by millions of people and will soon contain audio and video as well as text. Smell in messages will take a bit longer to perfect. **Instant messaging** allows two people to type messages at each other in real time. A multiperson version of this idea is the chat room, in which a group of people can type messages for all to see.

Real-time email will allow remote users to communicate with no delay, possibly seeing and hearing each other as well. This technology makes it possible to have virtual meetings, called **videoconference,** among far-flung people. Virtual meetings could be used for remote school, getting medical opinions from distant specialists, and numerous other applications. **Worldwide newsgroups**, with discussions on every conceivable topic are common among a select group of people, and this will grow to include the population at large. Here one person posts a message and all the other subscribers to the newsgroup can read it and can respond with an answer.

Our third category is **entertainment**, which is a huge and growing industry. The killer application here is **video on demand.** Live television may also become interactive, with the audience participating in quiz shows, choosing among contestants, and so on. **Game playing** is an important application of computer network for people. Multiperson real-time simulation games, like hide-and-seek in a virtual dungeon, and flight simulators with the players on one team trying to shoot down the players on the opposing team, 3-dimensional real-time, photographic-quality moving images, virtual reality games are few to mentio

Tag	**Full name**	**Example**
B2C	Business-to-Consumer	Ordering books on-line
B2B	Business-to-Business	Car manufacturer ordering tires from supplier
G2C	Government-to-Consumer	Government distributing tax forms electronically
C2C	Consumer-to-Consumer	Auctioning second-hand products on line
P2P	Peer-to-Peer	File sharing

Mobile Users Mobile computers, such as notebook computers and personal digital assistants (PDAs), are one of the fastest-growing segments of the computer industry. A common reason is the portable office. People on the road want to use their portable electronic equipment to send and receive telephone calls, faxes and electronic mail, surf the web, access remote files, and log on remote machines and they want to do this from anywhere on land, sea or air. Wireless networks are of great value of fleets of trucks, taxis, delivery vehicles and repair persons for keeping in contact with home. For example in many cities, taxi drivers are independent business men rather than being employee's of a taxi company. The taxi has a display the driver can see, when a customer calls up, a central dispatcher types in the pick-up and destination points. This information is displayed on a driver's displays and a beep sounds. The first driver to hit a button on the display gets the call. Wireless network are also important to the military.

Distinction between **Fixed wireless** and **mobile wireless**

Wireless	Mobile	Applications
No	No	Desktop computers in offices
No	Yes	A notebook computer used in a hotel room
Yes	No	Networks in older, unwired buildings
Yes	Yes	Portable offices; PDA for store inventory

(Refer class notes)

Social issues The widespread introduction of networking has led to

1. Social
2. Ethical and
3. Political problems.

The trouble arises when newsgroups are set up on topics that people contradicting views. Views posted to such groups may be deeply offensive to some people. Thus the debate rages. Users rights are violated and freeness of speech is barred. Computer networks offer the potential for sending anonymous messages, a way to express views without fear of reprisals. This newfound freedom brings with it many unsolved social, political, and moral issues. (refer class notes)

NETWORK HARDWARE:- There is no generally accepted taxonomy into which all computer networks fit, but two dimensions Stand out as important.

1. Transmission Technology
2. Scale

Transmission Technology: Broadly speaking, there are two types of transmission technology:

1. Broadcast networks.
2. Point-to-point Networks

Broadcast networks have a single communication channel that is shared by all the machines on the network. Short messages, called **packets** in certain contexts sent by any machine are received by all the others. An address field within the packet specifies for whom it is intended. Upon receiving a packet, a machine checks the address field. If the packet is intended for itself, it processes the packet; if the packet is intended for some other machine, it is just ignored. Although the packet may actually be received by many systems, only the intended one responds. The others just ignore it.

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

This mode of operation is called **broadcasting.** Some broadcast systems also support transmission to a subset of the machines, something known as **multicasting.** One possible scheme is to reserve one bit to indicate multicasting. The remaining *n* - 1 address bits can hold a group number. Each machine can "subscribe" to any or all of the groups. When a

packet is sent to a certain group, it is delivered to all machines subscribing to that group.

In contrast, **point-to-point** networks consist of many connections between individual pairs of machines. To go from the source to the destination, a packet on this type of network may have to first visit one or more intermediate machines.

Often multiple routes, of different lengths are possible, so routing algorithms play an important role in point-to-point networks. As a general rule smaller, geographically localized networks tend to use broadcasting, whereas larger networks usually are point-to-point. Point-to-Point transmission with one sender and one receiver is called **Unicasting.**

Inter processor Distances	Processors located in same	Examples
1 m	Square meter	Personal Area Network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan Area Network
100 km	Country	Wide Area Network
1000 km	Continent	
10000 km	Planet	The Internet

Classification of interconnected processors by scale.

An alternative criterion for classifying networks is their scale. Multiple processor systems can be arranged by their physical size. At the top are **data flow machines,** highly parallel computers with many functional units all working on the same program. Next come the **multicomputers,** systems that communicate by sending messages over very short, very fast buses. Beyond the multicomputers are the true networks, computers that communicate by exchanging messages over longer cables. These can be divided into local, metropolitan, and wide area networks, finally, the connection of two or networks are called an internetwork. The worldwide Internet is a well-known example of an internetwork.

Local Area Networks:- **Local area networks,** generally called LANs, are privately-owned network within a single building or campus of up to a few kilometers in size. They are widely used to connect personal computers and workstations in company offices and factories to share resources (e.g., printers) and exchange information.

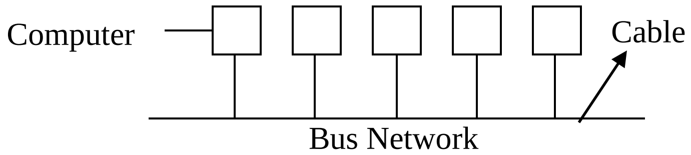
LANs are distinguished from other kinds of networks by three characteristics:

(1) size.

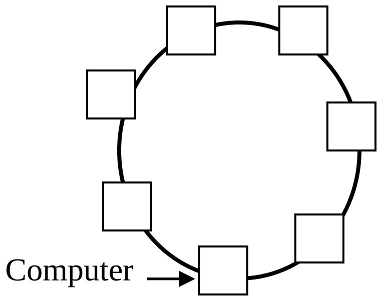
(2) transmission technology, and

(3) topology.

LANs are restricted in size, which means that the worst-case transmission time is bounded and known in advance. It simplifies network management. LANs often use a transmission technology consisting of a cable to which all the machines are attached. Traditional LANs run at speeds of 10 to *100* Mbps, have low delay (tens of microseconds), and make very few errors. Newer LANs may operate at higher speeds, up to hundreds of megabits/sec. Various topologies are possible for broadcast LANs. In a bus (i.e., a linear cable) network, at any instant one machine is the Master and is allowed to transmit. All other machines are required to refrain from sending. An arbitration mechanism is needed to resolve conflicts when two or more machines want to transmit simultaneously. The arbitration mechanism may be centralized or distributed. IEEE 802.3, popularly called Ethernet™, for example, is a bus-based broadcast network with decentralized control operating at 10 or 100 Mbps. Computers on an Ethernet can transmit whenever they want to; if two or more packets collide, each computer just waits a random time and tries again later.



Bus Network



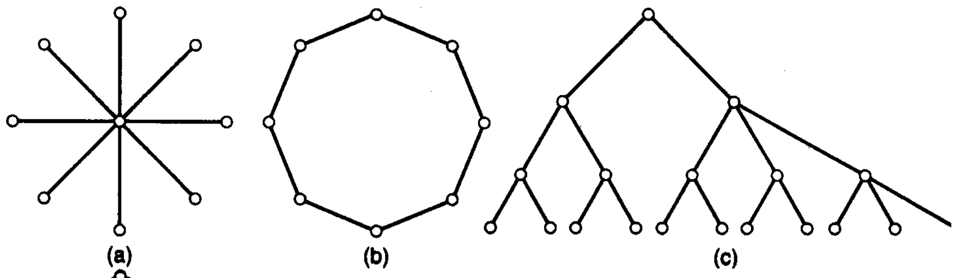
Ring Network

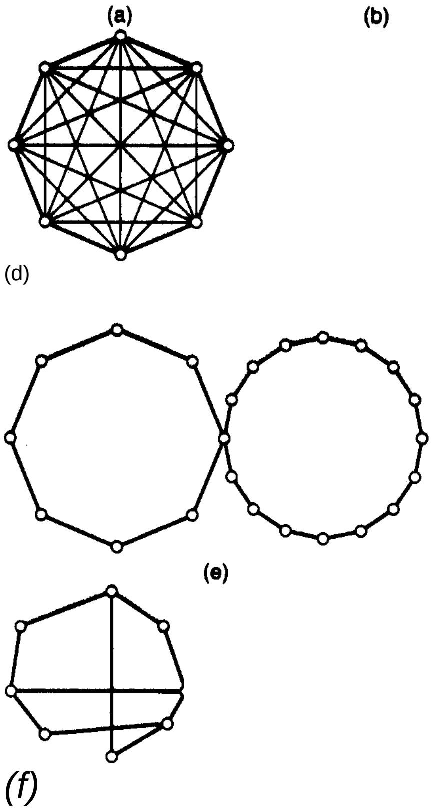
A second type of broadcast system is the ring. In a ring, each bit propagates around on its own, not waiting for the rest of the packet to which it belongs. Typically, each bit circumnavigates the entire ring in the time it takes to transmit a few bits, often before the complete packet has even been transmitted. Like all other broadcast systems, some rule is needed for arbitrating simultaneous accesses to the ring. The IBM token ring, is a popular ring-based LAN operating at 4 and 16 Mbps.

Broadcast networks can be further divided into static and dynamic, depending on how the channel is allocated. A typical static allocation would be to divide up time into discrete intervals and run a round robin algorithm, allowing each machine to broadcast only when its time slot comes up. Static allocation wastes channel capacity when a machine has nothing to say during its allocated slot, so most systems attempt to allocate the channel dynamically (i.e., on demand).

Dynamic allocation methods for a common channel are either centralized or decentralized. In the centralized channel allocation method, there is a single entity, for example a bus arbitration unit, which determines who goes next. It might do this by accepting requests and making a decision according to some internal algorithm.

In the decentralized channel allocation method, there *is* no central entity; each machine must decide for itself whether or not to transmit.





Topology for Point to Point Network

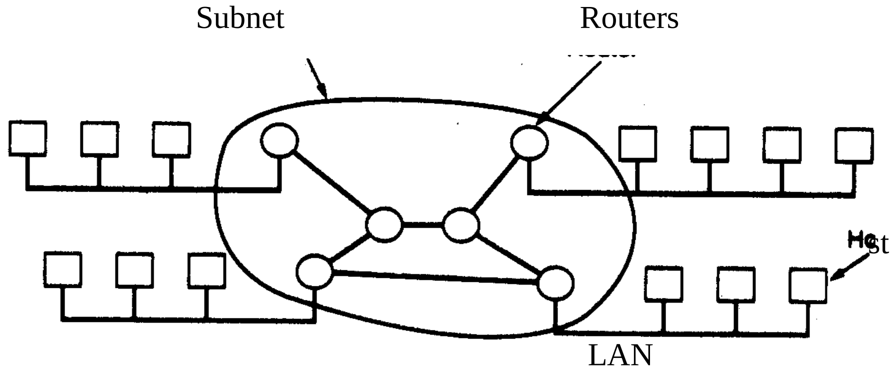
a) Star
b) Ring
c) Tree
d) Complete
e) Intersecting Rings
f) Irregular

Metropolitan Area Networks

Metropolitan Area Networks or Man covers a city. The best-known example of MAN it is the cable television network available in many cities. In early systems a large antenna was placed on top of a near by hill and signal was piped to the subscribers houses. At first, these were locally-designed, ad hoc systems. The next step was television and even entire channels designed for cable only, starting when the internet attracted a mass audience a cable TV network operator begun to realize that with some changes to the system, they could provide two-way internet service in un used parts of the spectrum, we see both television signals and internet being fed into the centralized **head end** for subsequent distribution to homes. (Refer diagram in notes)

Wide Area Networks

Wide Area Networks or **WAN** spans a large geographical area often a country or continent. It contains collections of machines for running user programs called **Hosts.** The hosts are connected by a communication subnet. The hosts are owned by the customers whereas the communication subnet owned and operated by Telephone Company or ISP. The job of the subnet is to carry messages from host to host. The subnet consists of two distinct components: **transmission lines** and **switching elements**.**Transmission lines** move bits between machines that are made of copper wire, optical fiber, or even radio links. **Switching elements** are specialized computers that connect 3 or more transmission lines. When data arrive on an incoming line the switching element must choose an outgoing

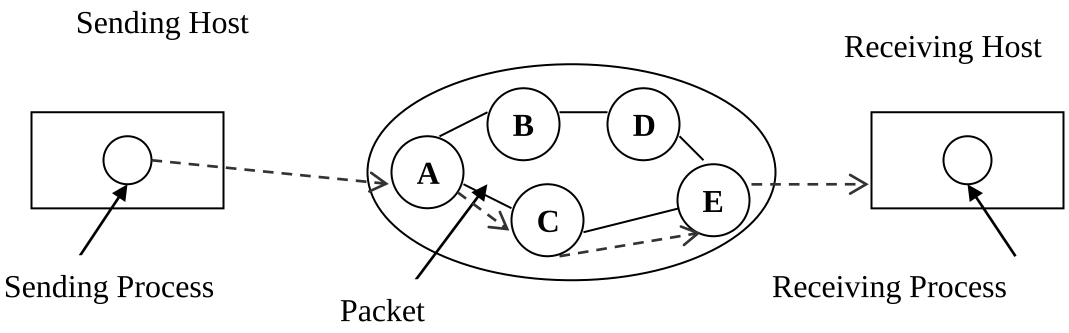


Relation between hosts and the subnet

line to forward them. The switching elements are also called router. The collection communication lines and routers (but not the hosts) form the subnet. A short subnet is a collection of communication lines that moved packets from the source host to the destination host. In most WAN the network contains numerous transmission lines, each one connecting a pair of routers.

If two routers that do not share a transmission line wish to communicate, they must do this indirectly, via other routers. When a packet is sent from one router to another via one or more intermediate routers the packet is received at each intermediate routers and stored there until the required line free and then forwarded. A subnet organized according to this principle is called store and forward or packet switched subnet. When the packets are small and all the same size they are often called as **cells.**

The principle of packet switched WAN, when a process on some host as a message to be sent to a process on some other host the sending host first cuts the message into packets, each one bearing its number in the sequence. The packets are then transported individual over the network and deposited at the receiving hosts where they are reassembled into the original message and delivered to the receiving process



A second possibility for a WAN is a satellite or ground radio system. Each router has an antenna through which it can send and receive. Sometimes the routers are connected to a substantial point-to-point subnet, with only some of them having a satellite antenna. Satellite networks are inherently broadcast and are most useful when the broadcast property is important.

Wireless networks: Wireless network can be divided into three main categories:

- System interconnection
- LANs
- Wireless WANs

System interconnection: System interconnection is all about interconnecting the components of a computer using short range radio. Every computer has monitor, keyboard, mouse and printer connected to main unit by cables. New users have a hard time plugging all the cables into right little holes. Some companies got together to design short range wireless network called **Bluetooth** to connect these components without wires. Bluetooth also allows digital cameras, headsets, scanners and other devices to connect the computer about within range. No cables, no driver installation just put down them on and they work.

Wireless LANs: These are systems in which every computer has a radio modem and antenna which it can communicate with other systems. If systems are close enough they can communicate directly with one another with peer-to-peer configuration. Wireless LANs are becoming increasingly common in small offices and homes. There is a standard for wireless LANs called **IEEE 802.11.**

Wireless WANs: The radio network used for cellular telephones is an example of low bandwidth wireless systems. System has already gone through 3 generation. The first generation was analog and for voice only. The second generation was digital and for voice only. The third generation is digital and it is for both voice and data. Wireless LANs can operate rate up to 15 Mbps over distance of ten of meters. Cellular system operate below 10 Mbps but the distance between way station and the computer or telephone is measured in kilometers rather than meters. A standard for a called **IEEE 802.16**. For example an airplane with number peoples using modem and seat-back telephones to call the office. Each call is independent of other ones. Next case a flying LAN each seats comes equipped with an Ethernet connection into which passengers can plug their computers. A single router on the aircraft maintain radio link with some router on the ground, changing router as its flies along.

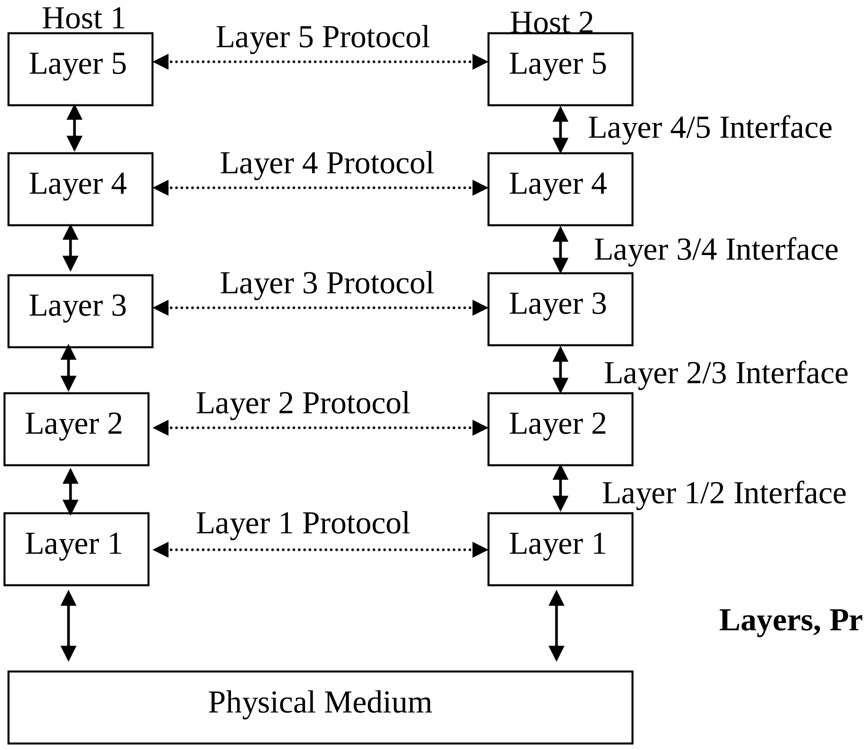
Home network: Home network is on the horizon. The fundamental idea is that in the future most homes will be setup for networking. Every device in home will be capable of communicating with every other device, and all of them will be accessible over the internet. Many devices are capable of being a network some of more obvious categories are as follows:

- Computers (Desktop PC, Notebook PC, PDA, Shared Peripherals)
- Entertainment (TV, DVD, VCR, Camcorder, Camera, Stereo, MP3)
- Telecommunication (Telephone, Mobile telephone, Intercom, Fax)
- Appliances (Microwave, Refrigerator, Clock, Furnace, Airco, Lights)
- Telemetry (Utility meter, Smoke/burglar alarm, Thermostat, Babycam)

Internetworks: A collection of interconnected networks called internetwork or internet. A common form of internet is a collection of LANs connected by WANs. Subnet makes the most sense in the context of wide area network, Where it refers to collection of routers to the communication lines owned by network operator. Telephone system consist of telephone switching offices connected to one another by high speed lines and houses and businesses by low speed lines. Lines and equipment owned by telephone companies form the subnet of telephone system. The combination of a subnet and its host forms a network. An internetwork is formed when distinct network are interconnected.

NETWORK SOFTWARE The first computer networks were designed with the hardware as the main concern and the software as an afterthought. Network software is now highly structured.

Protocol Hierarchies To reduce their design complexity, most networks are organized as a series 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 layer, and the function of each 1ayer differ from network to network. However, in all networks, the purpose of each layer is to offer certain services to the higher layers, shielding those layers from the details of how the offered services are actually implemented. Layer *n* on one machine carries on a conversation with layer *n* on another machine. The rules and conventions used in this conversation are collectively known as the layer *n* **protocol.** Basically, a protocol is an agreement between the communicating parties on how communication is to proceed. Violating the protocol will make communication more difficult if not impossible. The entities comprising the corresponding layers on different machines are called **peers.** The peers communicate using protocol.

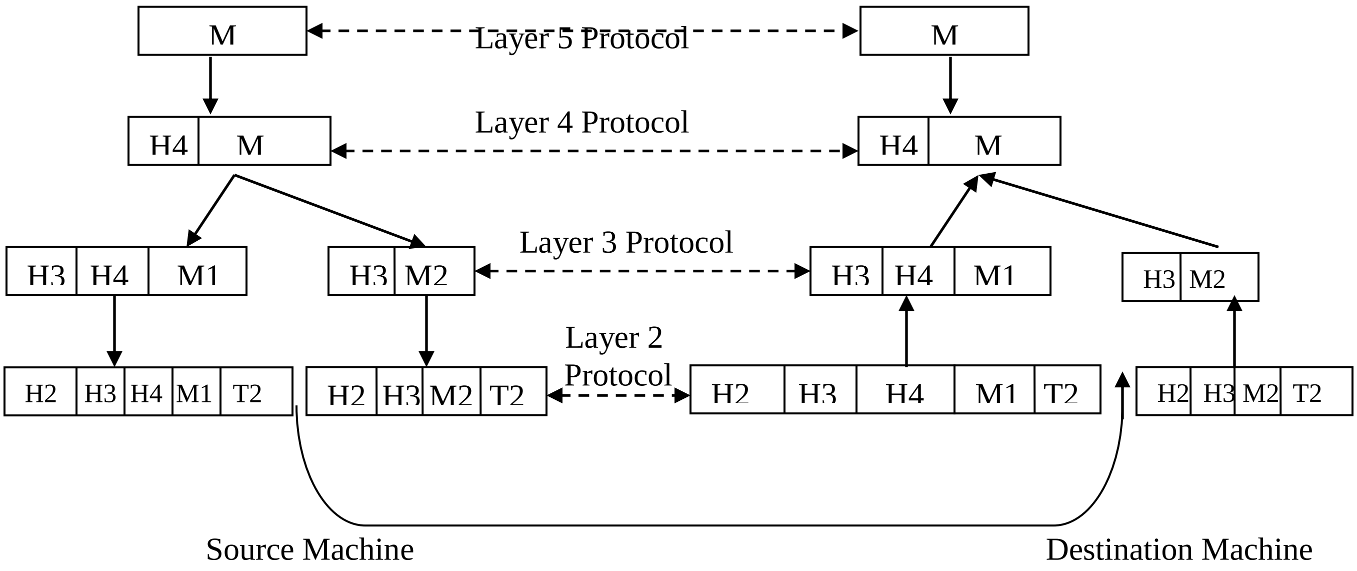


Layers, Protocols and Interfaces

In reality, no data are directly transferred from layer n on one machine to layer *n on* another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs. Between each pair of adjacent layers there is an **interface.** The interface defines which primitive operations and services the lower layer offers to the upper one. One of the most important considerations is defining clean interfaces between the layers. Doing so, in turn, requires that each layer perform a specific collection of well-understood functions. In addition to minimizing the amount of information that must be passed between layers, clean-cut interfaces also make it simpler to replace the implementation of one layer with a completely different implementation because all that is required of the new implementation is that it offers exactly the same set of services to its upstairs neighbor as the old implementation did.

A set of layers and protocols is called **network architecture**. The specification of architecture contains enough information to build the hardware/software for each layer so that it correctly obeys the appropriate protocol. A list of protocols used by a certain system, one protocol per layer, is called a protocol stack. A message M, produced by the application process puts a header in front of the message to identify the message and passes the result to the next layer. The header includes control information, such as a sequence numbers to allow the next layer in the destination machine to deliver messages in the right order. Headers may also contain sizes, times and other control fields. The layers break up the incoming messages into smaller units, packets.

For example, message *M* is split into two parts, m1 and m2. A Layer decides which of the outgoing lines to use and passes the packets to next layer. This Layer adds not only a header to each piece, but also a trailer, and give the resulting unit to layer below it for physical transmission. At the receiving machine the message moves upward, from layer to layer, with headers being stripped off as it progresses. None of the headers for layers below *n* are passed up to layer *n.* The peer process abstraction is crucial to all network design. Using it, the unmanageable task of designing the complete network can be broken into several smaller, manageable, design problems, namely the design of the individual layers.



Design Issues for the Layers

Some of the key design issues that occur in computer networking are present in several Layers. Every layer needs a mechanism for identifying senders and receivers. Since a network normally has many computers, some of which have multiple processes, a means is needed for a process on one machine to specify with whom it want to talk. As a consequence of having multiple destinations, some form of **addressing** is needed in order to specify a specific destination.

Another set of design decisions concerns the rules for **data transfer**. In some systems, data only travel in one direction **(simplex communication).** In others they can travel in either direction, but not simultaneously **(half-duplex communication).** In still others they travel in both directions at once **(full-duplex communication).** The protocol must also determine how many logical channels the connection corresponds to, and what their priorities are. Many networks provide at least two logical channels per connection, one for normal data and one for urgent data.

Error control is an important issue because physical communication circuits are not perfect. Many error-detecting and error-correcting codes are known, but both ends of the connection must agree on which one is being used. In addition the receiver must have some way of telling the sender which messages have been correctly received and which has not. Not all communication channels preserve the order of messages sent on them to deal with a possible loss of sequencing; the protocol must make explicit provision for the receiver to allow the pieces to be put back together properly. An issue that occurs at every level is how to keep a **fast sender from swamping a slow receiver with data**. Some of them involve some kind of feedback from the receiver to the sender, either directly or indirectly, about the receiver's current situation. This subject is called **flow control.**

Another problem that must be solved at several levels is the inability of all processes to accept arbitrarily long messages. This property leads to mechanisms for disassembling, transmitting, and then reassembling messages. A related issue is what to do when processes insist upon transmitting data in units that are so small that sending each one separately is inefficient. Here the solution is to gather together several small messages heading toward a common destination into a single large message and dismember the large message at the other side. When it is inconvenient or expensive to set up a separate connection for each pair of communicating processes, the underlying layer may decide to use the same connection for multiple, unrelated conversations. As long as this **multiplexing** and **de-multiplexing** is done transparently, it can be used by any layer. Multiplexing is needed in the physical layer, for example, where all the traffic for all connections has to be sent over at most a few physical circuits. When there are multiple paths between source and destination, a. route must be chosen. Sometimes this decision must be split over two or more layers.

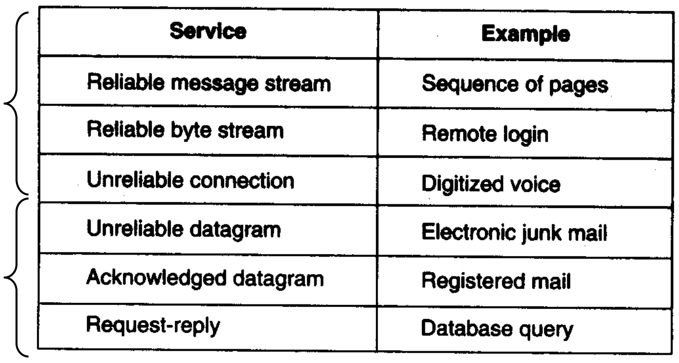
Connection-Oriented and Connectionless Services Layers can offer two different types of service to the layers above them:

1.connection-oriented and
2.connectionless.

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. Similarly, to use a connection-oriented network service, the service user first establishes a connection, uses the connection, and then releases the connection. The essential aspect of a connection is that it acts like a tube: the sender pushes objects (bits) in at one end, and the receiver takes them out in the same order at the other end.

Connectionless 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. Normally, when two messages are sent to the same destination, the first one sent will be the first one to arrive. However, it is possible that the first one sent can be delayed so that the second one arrives first. Each service can be characterized by a **quality of service**. Some services are reliable in the sense that they never lose data. Usually, a reliable service is implemented by having the receiver acknowledge the receipt of each message, so the sender is sure that it arrived. The acknowledgement process introduces overhead and delays, which are often worth it but are sometimes undesirable.

A typical situation in which a reliable connection-oriented service is appropriate is file transfer. The owner of the file wants to be sure that all the bits arrive correctly and in the same order they were sent. Reliable connection-oriented service has two minor variations: message sequences and byte streams. In the former, the message boundaries are preserved when two 1-KB messages are sent, they arrive as two distinct 1-KB messages never as one 2-KB message. In the latter, the connection is simply a stream of bytes, with no message boundaries. Not all applications require connections. Unreliable (not acknowledged) connectionless service is often called **datagram service,** which does not provide an acknowledgement back to the sender. In other situations, the convenience of not having to establish a connection to send one short message is desired, but reliability is essential. The **acknowledged datagram** service can be provided for these applications. Still another service is the **request-reply** service. In this service, the sender transmits a single datagram containing a request, the reply contains the answer. Request-reply is commonly used to implement communication in the client-server model: the client issues a request and the server responds to it.



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

Six different types of Service

Service Primitives:- A service is formally specified by a set of **primitives** (operations) available to a user or other entity to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity. One way to classify the service primitives is to divide them into four classes:

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

Five classes of service primitives.

First the server executes LISTEN to indicate that is prepared to accept the incoming connection. A common way to implement LISTEN is make it a blocking system call. After executing primitive a server process a block until a request for connection appears. A client process executed CONNECT to establish a connection with the server. The CONNECT call needs to specify who to connect to, parameter gives the servers address.

The operating system sends the packet to the peer asking it to connect as shown by (1) fig (refer class notes). The client process is connected until there is a response. When a packet arrives at the server it is processed by the operating system. When a system sees the packet is requesting a connection, it checks to see there is a listener. Unblock the listener and sends back the acknowledgement (2). The arrival of acknowledgement releases the client. At this point the client and server are both are running and they have a connection established. Next step for the server to execute RECEIVE to prepare to accept the first request.

The server does this immediately upon being released from the LISTEN, before the acknowledgement can get back to the client. The RECEIVE call blocks the srever. The client executes send to transmit the request (3) followed by the execution to receive to get the reply. The arrival of the request packet at the server machine unblocks the server process so it can process the request. After it has done the work it uses SEND to return the answer to the client (4). If it is done it use DISCONNECT to terminate the connection. An initial is a blocking call, suspending the client and sending a packet to the server saying that connection is no longer needed (5). When the server gets the packet it also issues a DISCONNECT of its own, acknowledging the client and releasing the connection when the servers packet (6) gets back to the client machine, the client process is released and connection is broken.

The Relationship of Services to Protocols

A ***service*** is a set of primitives (operations) that a layer provides to the layer above it. The service defines what operations the layer is prepared to perform or behalf of its users, but it says nothing at all about how these operations are implemented. A service relates to an interface between two layers, with the lower layer being the service provider and the upper layer being the service user.

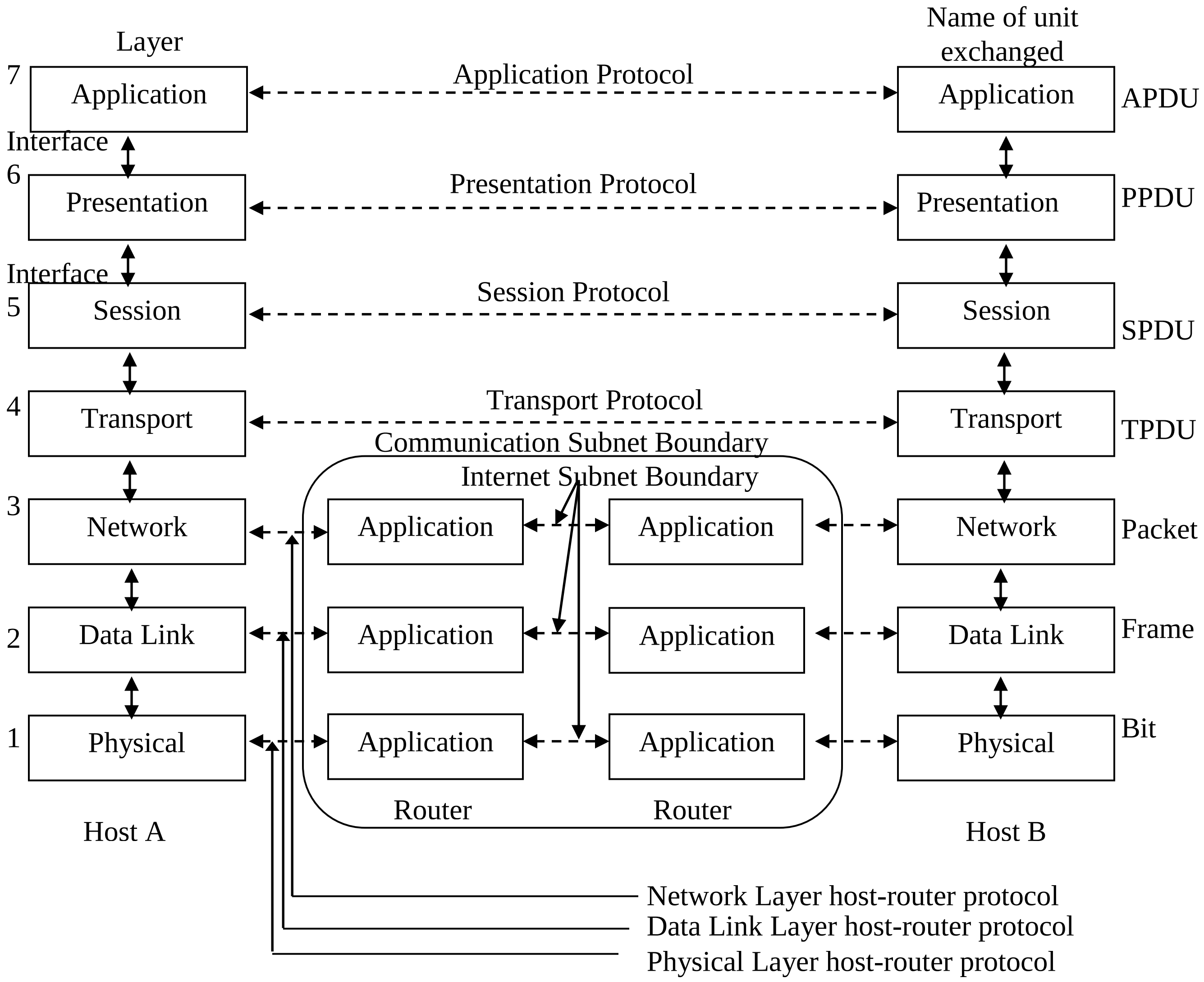
A ***protocol***, in contrast, is a set of rules governing the format and meaning of the frames, packets, or messages that are exchanged by the peer entities within a layer. Entities use protocols in order to implement their service definitions. They are free to change their protocols at will, provided they do not change the service visible to their users. In this way, the service and the protocol are completely decoupled.

A service is like an abstract data type or an object in an object-oriented language. It defines operations that can be performed on an object but does not specify how these operations are implemented. A protocol relates to the *implementation* of the service and as such is not visible to the user of the service.

The OSI Reference Model

This model is based on a proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the protocols used in the various layers The model is called the **ISO OSI (Open Systems Interconnection) Reference Model** because it deals with connecting open systems—that is, systems that are open for communication with . The OSI model has seven layers. The principles that were applied to arrive at the seven layers are as follows:

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



OSI Reference Model

The Physical Layer

The **physical layer** is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit, it is received by the other side as a 1 bit, not as a 0 bit. Typical questions here are how many volts should be used to represent a 1 and how many for a 0, how many microseconds a bit lasts, whether transmission may proceed simultaneously in both directions, how the initial connection is established and how it is torn down when both sides are finished, etc., the design issues here largely deal with mechanical, electrical, and procedural interfaces, and the physical transmission medium, which lies below the physical layer.

The Data Link Layer

The main task of the **data link layer** is to take a raw transmission facility and transform it into a line that appears free of undetected transmission errors to the network layer. It accomplishes this task by having the sender break the input data up into **data frames** transmit the frames sequentially, and process the **acknowledgement frames** sent back by the receiver. Since the physical layer merely accepts and transmits a stream of bits without any regard to meaning or structure, it is up to the data link layer to create and recognize frame boundaries. This can be accomplished by attaching special bit patterns to the beginning and end of the frame. If these bit patterns can accidentally occur in the data, special care must be taken to make sure these patterns are not incorrectly interpreted as frame delimiters. A noise burst on the line can destroy a frame completely. In this case, the data link layer software on the source machine can retransmit the frame.

However, multiple transmissions of the same frame introduce the possibility of duplicate frames. A duplicate frame could be sent if the acknowledgement frame from the receiver back to the sender were lost. It is up to this layer to solve the problems caused by damaged, lost, and duplicate frames. The data link layer may offer several different service classes to the network layer, each of a different quality and with a different price.

Another issue that arises in the data link layer is how to keep a fast transmitter from drowning a slow receiver in data. Some traffic regulation mechanism must be employed to let the transmitter know how much buffer space the receiver has at the moment. Frequently, this flow regulation and the error handling are integrated. If the line can be used to transmit data in both directions, this introduces a new complication that the data link layer software must deal with. The problem is that the acknowledgement frames for *A* to *B* traffic compete for the use of the line with data frames for the B to A traffic.

Broadcast networks have an additional issue in the data link layer: how, to control access to the shared channel. A special sub layer of the data link layer, the medium access sublayer, deals with this problem.

The Network Layer

The **network layer** is concerned with controlling the operation of the subnet. A key design issue is determining how packets are routed from source to destination. Routes can be based on static tables that are "wired into" the network and rarely changed. They can also be determined at the start of each conversation, for example a terminal session. Finally, they can be highly dynamic, being determined a new for each packet, to reflect the current network load. If too many packets are present in the subnet at the same time, they will get in each other's way forming bottlenecks. The control of such congestion also belongs to the network layer. There should be software that must count how many packets or characters or bits are sent by each customer. When a packet crosses between layers, with different rates on each side, the accounting can become complicated.

When a packet has to travel from one network to another to get to its destination, many problems can arise. The addressing used by the second network may be different from the first one. The second one may not accept the packet because it is too large, the protocols may differ, and so on. It is up to the network layer to overcome all these problems to allow heterogeneous networks get interconnected. In broadcast networks, the routing problem is simple, so the network layer often is thin or even nonexistent.

The Transport Layer

The basic function of the **transport layer** is to accept data from the session layer, split it up into smaller units if needed, pass these to the network layer, and ensure that the pieces all arrive correctly at the other end. Under normal conditions, the transport layer creates a distinct network connection for each transport connection required by the session layer. If the transport connection requires a high throughput, however, the transport layer might create multiple network connections, dividing the data among the network connections to improve throughput. On the other hand, if creating or maintaining network connection is expensive, the transport layer might multiplex several transport connections onto the same network connection to reduce the cost. In cases, the transport layer is required to make the multiplexing transparent to the session layer. The transport layer also determines what type of service to provide the session layer and ultimately, the users of the network.

The most popular type of transport connection is an error-free point-to-point channel that delivers messages or bytes in the order in which they were sent. However, other possible kinds of transport service are transport of isolated messages with no guarantee about the order of delivery, and broadcasting of messages to multiple destinations. The type of service is determined when the connection is established. The transport layer is a true end-to-end layer, from source to destination. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message headers and control messages. In the lower layers, the protocols are between each machine and its immediate neighbors, and not by the ultimate source and destination machines, which may be separated by many routers.

Many hosts are multiprogrammed, which implies that multiple connections will be entering and leaving each host. There needs to be some way to tell which message belongs to which connection. In addition to multiplexing several message streams onto one channel, the transport layer must take care of establishing and deleting connections across the network. This requires some kind of naming mechanism, so that a process on one machine has a way of describing with whom it wishes to converse. There must also be a mechanism to regulate the flow of information, so that a fast host cannot overrun a slow one. Such a mechanism is called **flow control** and plays a key role in the transport layer.

The Session Layer

The session layer allows users on different machines to establish **sessions** between them. A session allows ordinary data transport, as does the transport layer, but it also provides enhanced services useful in some applications. A session might be used to allow a user to log into a remote timesharing system or to transfer a File between two machines. One of the services of the session layer is to manage dialogue control. Sessions can allow traffic to go in both directions at the same time, or in only one direction at a time.

A related session service is **token management.** For some protocols, it is essential that both sides do not attempt the same operation at the same time. To manage these activities, the session layer provides tokens that can be exchanged. Only the side holding the token may perform the critical operation.

Another session service is **synchronization**, the session layer provides a way to insert checkpoints into the data stream, so that after a crash, only the data transferred after the last checkpoint have to be repeated.

The Presentation Layer

The **presentation layer** performs certain functions that are requested sufficiently often to warrant finding a general solution for them, rather than letting each user solve the problems. The presentation layer is concerned with the syntax and semantics of the information transmitted. A typical example of a presentation service is encoding data in a standard way. Most user programs do not exchange random binary bit strings. They exchange things such as people's names, dates, amounts of money, and invoices. These items are represented as character strings, integers, floating-point numbers, and data structures composed of several simpler items. Different computers have different codes for representing character strings (e.g., ASCII and Unicode), integers (e.g., one's complement and two's complement), and these different representations should be made to communicate. The presentation layer manages these abstract data structures and converts from the representation used inside the computer to the network standard representation and back.

The Application Layer

The **application layer** contains a variety of protocols that are commonly needed. For example, there are hundreds of incompatible terminal types in the world. Consider the plight of a full screen editor that is supposed to work over a network with many different terminal types, each with different screen layouts, escape sequences for inserting and deleting text, moving the cursor, etc. One way to solve this problem is to define an abstract **network virtual terminal** that editors and other programs can be written to deal with. To handle each terminal type, a piece of software must be written to map the functions of the network virtual terminal onto the real terminal. For example, when the editor moves the virtual terminal's cursor to the upper left-hand corner of the screen, this software must issue the proper command sequence to the real terminal to get its cursor there too. All the virtual terminal software is in the application layer. Another application layer function is file transfer. Different file systems have different file naming conventions, different ways of representing text lines, and so on. Transferring a file between two different systems requires handling these and other incompatibilities. This work, too, belongs to the application layer, as do electronic mail, remote job entry, directory lookup, and various other general purpose and special-purpose facilities.

The TCP/IP Reference model

ARPANET was a research network sponsored by the **DOD** (U.S Department of Defense), connecting hundreds of universities and government installations, using leased telephone lines. When satellite and radio networks were added later, the existing protocols had trouble interworking with them, so new reference architecture was needed. Thus, the ability to connect multiple networks in a seamless way was one of the major design goals from the very beginning. This architecture later became known as the TCP/IP Reference model. DOD wanted connections to remain intact as long as the source and destination machines were functioning.

The INTERNET Layer

All these requirements led to the choice of a packet-switching network based on a connectionless internetwork layer. This layer, called the internet layer, is the linchpin that holds the whole architecture together. Its job is to permit hosts to inject packets into any network and have them travel independently to the destination. They arrive in a different order than they were sent, the job of higher layers to rearrange them. The analogy here is with the (snail) mail system. A person can drop a sequence of international letters into a mail box in one country and with a little luck, most of them will be delivered to the correct address in the destination country, letters will travel through one or more international mail gateways. Each country (i.e., each network) has its own stamps, preferred envelope sizes and delivery rules is hidden from the users. The internet layer defines an official packet format and protocol called **IP** (**INTERNET PROTOCOL**). The job of the internet layer is to deliver IP packets where they are supposed to go.

The TRANSPORT Layer

The layer above the internet layer in the TCP/IP model is the **transport layer**. It is designed to allow peer entities on the source and destination hosts to carry on a conversation, just as OSI transport layer. Two end-to-end transport protocols are defined, first one TCP (**Transmission Control Protocol**), is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet. It fragments the incoming byte stream into discrete messages and passes each one to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles **flow control** to make fast sender cannot swamp slow receiver with more messages than it can handle.

OSI	**TCP/IP**
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data link	Host-to-network
Physical	

Not present in the mode

The second protocol is **UDP** (**USER DATAGRAM PROTOCOL)** is an un-reliable, connectionless protocol for applications protocol for applications that do not want TCP's sequencing or flow control and wish to provide their own. It is also widely used for one-shot, client-server-type request-reply queries and applications in which prompt delivery is more important than accurate delivery,

The application layer:

On the top of the transport layer is application layer. It contains all the higher level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP). The virtual terminal allows a user on one machine to log on to a distant machine and work there. The file transfer protocol provides the way to move data efficiently from one machine to another. Electronic mail was originally just a kind of file transfer but later a specialized protocol (SMTP) was developed for it. Some other protocol are Domain Name System (DNS) for mapping host names on to their network address, NNTP, the protocol for moving USENET news article around, HTTP, the protocol for fetching pages on the World Wide Web.

The Host-to-Network layer:- It tells only to point out that the host has to connect to the network using some protocol so it can send IP packets to it.

A Comparison of the OSI and TCP/IP Reference Models:-

The OSI and TCP/IP reference model have much in common. Both are based on the concept of stack of independent protocols. Also the functionality of the layers is roughly similar. For example in the both models the layer up thru & including the transport layer are there to provide an end-to-end, network independent transport services to processes to wishing to communicate. The layers form the transport provider. Again in both models, the layers above transport are application-oriented users of the transport service.

Three concepts are central to the OSI Model:

1. Services
2. Interfaces
3. Protocols

The *services* definition tells what the layer does, not how entities above it access it or how the layer works. It defines the layer's semantics. A layer's *interface* tells the processes above it how to access it. It specifies what the parameter are and what results to expect. Finally, the peer *protocols* used in a layer are the layer's own business. It can use any protocol it wants to, as long as it gets the job done. It can also change them at will without affecting software in higher layers.

The TCP/IP model did not distinguish between service, interface, and protocol. For example the only real services offered by the internet layer are SEND IP PACKET and RECEIVE IP PACKET.

- The OSI reference model was devised before the corresponding protocols were invented.
- With TCP/IP the reverse was true: the protocol came first, and the model was really just a description of the existing protocols.
- OSI model has seven layers and the TCP/IP has four layers. Both have (inter)network, transport, and application layers, but the other layers are different.
- OSI model supports both connection less and connection oriented communication in the network layer, but only connection oriented communication in the transport layer.
- The TCP/IP model has only one mode in the network layer but supports both modes in the transport layer giving the user a choice. This choice is especially important for simple request response protocols.

A Critique of the OSI Model and protocols:-

These lessons can be summarized as:

1. Bad timing.
2. Bad technology
3. Bad implementations
4. Bad politics

Bad timing: - The time at which a standard is established is absolutely critical to its success. When the subject is first discovered, there is a burst of research activity in the form of discussions, papers and meetings. After a while this activity subsides, corporation discover the subject, and the billion-dollar wave of investment hits. It is essential that the standard be return in the trough in between the two elephants. If the standards are written too early, before the research is finished, the subject may still be poorly understood; the result is bad standards. If they are written too late so many companies may have already made major investments in different ways of doing things that the standards are effectively ignored.

Bad technology: - The choice of seven layers was more political than technical, and two of the layers (session and presentation) are nearly empty, whereas two other ones (data link and network) are overfull. The OSI model, along with the associated service definition and protocol, is extraordinarily complex. Another problem with OSI that some functions, such as addressing, flow control and error control, reappear again and again in each layer.

Bad Implementation: - Complexity of the model and the protocols, it will come as no surprise the initial implementation were huge, unwieldy and slow.

Bad Politics: - OSI was widely thought to be the creature of the European telecommunication ministries, the European community, and later the U.S. Government. Some people viewed this development in the same light as IBM announcing in the 1960's that PL/I was the language of the future or DoD correcting this later by announcing it was actually Ada.

A Critique of the TCP/IP Reference Model: - The TCP/IP model and protocols have their problems too.

- 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 a term as used in the context of layered protocol. It is an interface between the network and the data link layer.
- TCP/IP model does not distinguish the physical and data link layer and they are completely different. The physical layer has to do with the transmission characteristic of copper wire, Fiber optics, wireless communication. The data link layer job's is to delimit the start and end of frames and get them from one side to the other with the desired degree of reliability.

Application Layer
Transport Layer
Network Layer
Data link Layer
Physical Layer

The Hybrid reference model to be used in this book.