Q1. Explain about Local Area Network (LAN).

Local Area Networks:

Local area networks, generally called LANs, are privately-owned networks within a single building or campus of up to a few kilometres 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) Their size,
(2) Their transmission technology, and
(3) Their topology.

LANs are restricted in size, which means that the worst-case transmission time is bounded and known in advance. Knowing this bound makes it possible to use certain kinds of designs that would not otherwise be possible. It also simplifies network management.

LANs may use a transmission technology consisting of a cable to which all the machines are attached, like the telephone company party lines once used in rural areas. Traditional LANs run at speeds of 10 Mbps to 100 Mbps, have low delay (microseconds or nanoseconds), and make very few errors. Newer LANs operate at up to 10 Gbps

Various topologies are possible for broadcast LANs. Figure1 shows two of them. In a bus (i.e., a linear cable) network, at any instant at most 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, usually operating at 10 Mbps to 10 Gbps. 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.

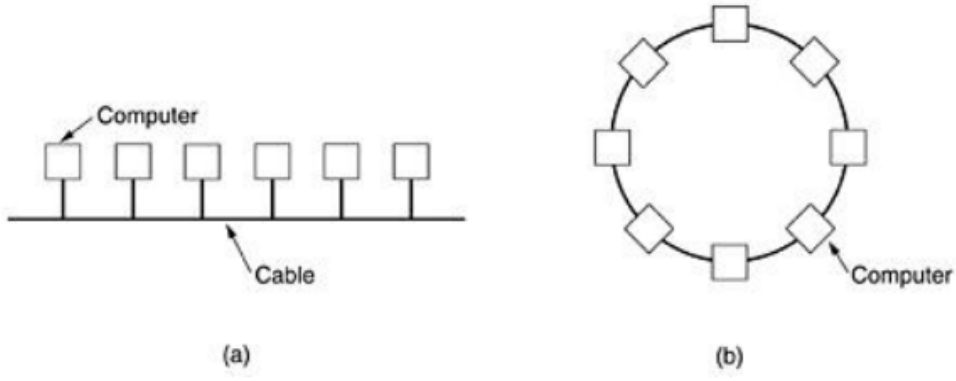


Fig.1: Two broadcast networks . (a) Bus. (b) Ring.

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. As with all other broadcast systems, some rule is needed for arbitrating simultaneous accesses to the ring. Various methods, such as having the machines take turns, are in use. IEEE 802.5 (the IBM token ring), is a ring-based LAN operating at 4 and 16 Mbps. FDDI is another example of a ring network.

Q2. Explain about Metropolitan Area Network (MAN).

Metropolitan Area Network:

A metropolitan area network, or MAN, covers a city. The best-known example of a MAN is the cable television network available in many cities. This system grew from earlier community antenna systems used in areas with poor over-the-air television reception. In these early systems, a large antenna was placed on top of a nearby hill and signal was then piped to the subscribers' houses.

At first, these were locally-designed, ad hoc systems. Then companies began jumping into the business, getting contracts from city governments to wire up an entire city. The next step was television programming and even entire channels designed for cable only. Often these channels were highly specialized, such as all news, all sports, all cooking, all gardening, and so on. But from their inception until the late 1990s, they were intended for television reception only.

To a first approximation, a MAN might look something like the system shown in Fig.2. In this figure both television signals and Internet are fed into the centralized head end for subsequent distribution to people's homes.

Cable television is not the only MAN. Recent developments in high-speed wireless Internet access resulted in another MAN, which has been standardized as IEEE 802.16.

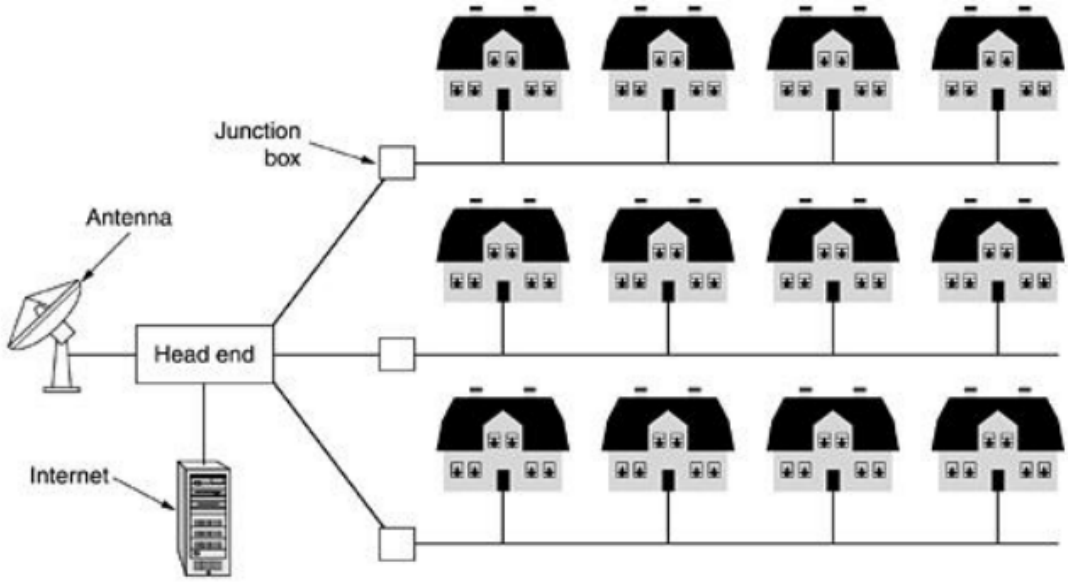


Fig.2: Metropolitan area network based on cable TV.

A MAN is implemented by a standard called DQDB (Distributed Queue Dual Bus) or IEEE 802.16. DQDB has two unidirectional buses (or cables) to which all the computers are attached.

Q3. Explain about Wide Area Network (WAN).

Wide Area Network:

A wide area network, or WAN, spans a large geographical area, often a country or continent. It contains a collection of machines intended for running user (i.e., application) programs. These machines are called as hosts. The hosts are connected by a communication subnet, or just subnet for short. The hosts are owned by the customers (e.g., people's personal computers), whereas the communication subnet is typically owned and operated by a telephone company or Internet service provider. The job of the subnet is to carry messages from host to host, just as the telephone system carries words from speaker to listener. Separation of the pure communication aspects of the network (the subnet) from the application aspects (the hosts), greatly simplifies the complete network design. In most wide area networks, the subnet consists of two distinct components: transmission lines and switching elements. Transmission lines move bits between machines. They can be made of copper wire, optical fiber, or even radio links.

In most WANs, 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 router in its entirety, stored there until the required output line is free, and then forwarded. A subnet organized according to this principle is called a store-and-forward or packet-switched subnet. Nearly all wide area networks (except those using satellites) have store-and-forward subnets. When the packets are small and all the same size, they are often called cells.

The principle of a packet-switched WAN is so important. Generally, when a process on some host has 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. These packets are then injected into the network one at a time in quick succession. The packets are transported individually over the network and deposited at the receiving host, where they are reassembled into the original message and delivered to the receiving process. A stream of packets resulting from some initial message is illustrated in Fig.3.1.

In this figure, all the packets follow the route ACE, rather than ABDE or ACDE. In some networks all packets from a given message must follow the same route; in others each packed is routed separately. Of course, if ACE is the best route, all packets may be sent along it, even if each packet is individually routed.

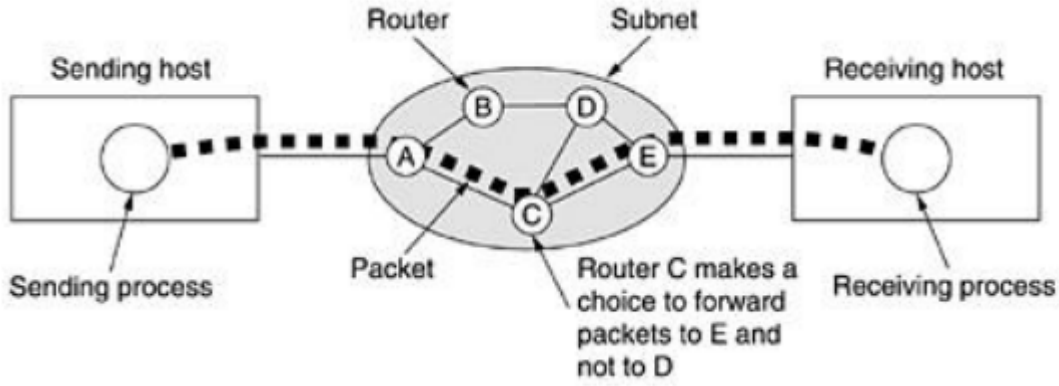


Fig.3.1: A stream of packets from sender to receiver.

Not all WANs are packet switched. A second possibility for a WAN is a satellite system. Each router has an antenna through which it can send and receive. All routers can hear the output from the satellite, and in some cases they can also hear the upward transmissions of their fellow routers to the satellite as well. 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.

Q4. Explain in detail about ISO-OSI reference Model.

The OSI Reference Model:

The OSI model (minus the physical medium) is shown in Fig 4. 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 (Day and Zimmermann, 1983). It was revised in 1995(Day, 1995). 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 other systems.

The OSI model has seven layers. The principles that were applied to arrive at the seven layers can be briefly summarized as follows:

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 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.

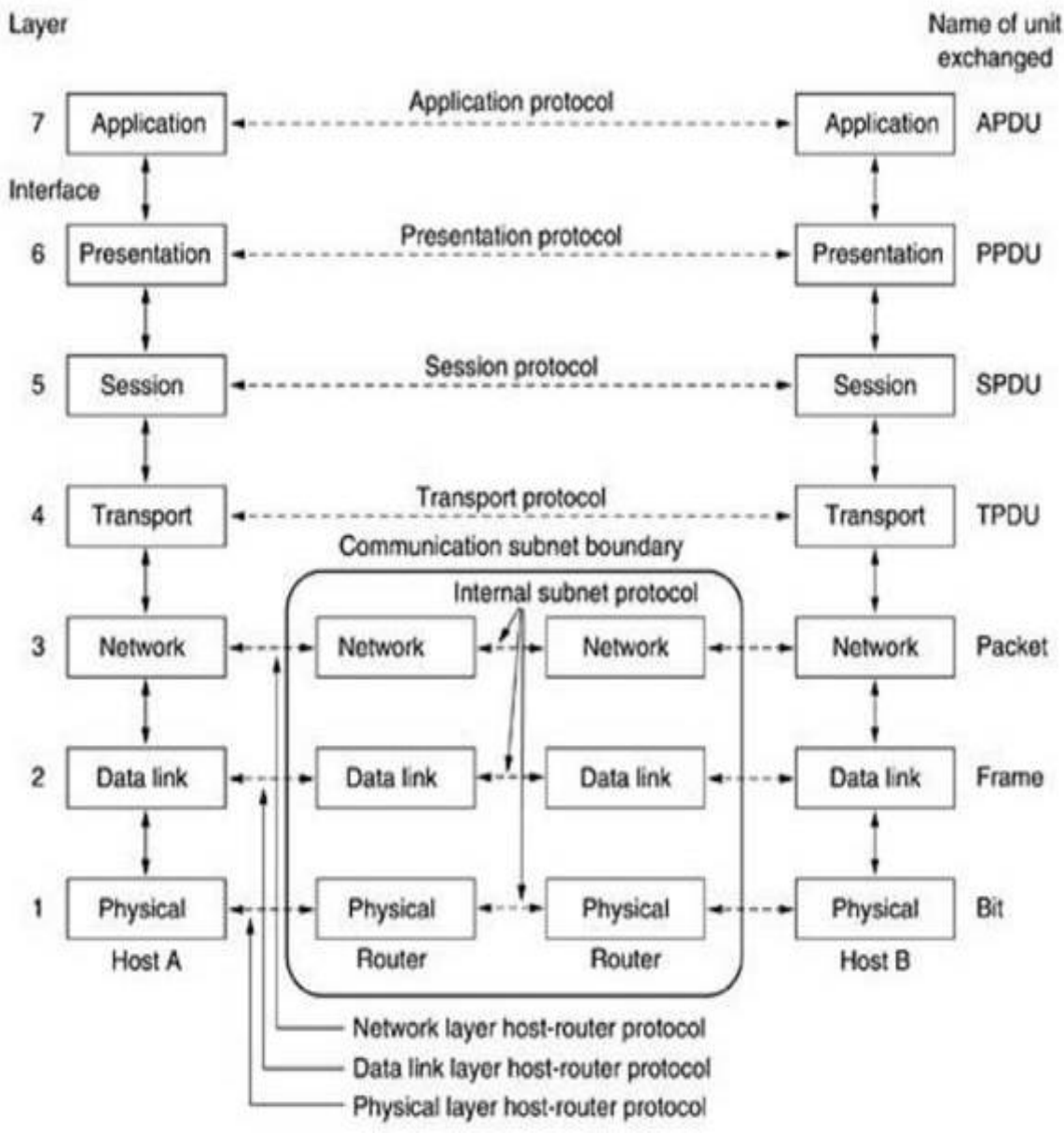


Fig.4: The OSI reference model.

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.

The Data Link Layer:

The main task of the data link layer is to transform a raw transmission facility into a line that appears free of undetected transmission errors to the network layer. It accomplishes this task by having the sender break up the input data into data frames (typically a few

hundred or a few thousand bytes) and transmits the frames sequentially. If the service is reliable, the receiver confirms correct receipt of each frame by sending back an acknowledgement frame.

Another issue that arises in the data link layer (and most of the higher layers as well) is how to keep a fast transmitter from drowning a slow receiver in data. Some traffic regulation mechanism is often needed 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.

The 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. 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 (e.g., a login to a remote machine). Finally, they can be highly dynamic, being determined anew 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 one another's way, forming bottlenecks. The control of such congestion also belongs to the network layer. More generally, the quality of service provided (delay, transit time, jitter, etc.) is also a network layer issue.

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 at all 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 to be interconnected. In broadcast networks, the routing problem is simple, so the network layer is often thin or even nonexistent.

The Transport Layer:

The basic function of the transport layer is to accept data from above, split it up into smaller units if need be, pass these to the network layer, and ensure that the pieces all arrive correctly at the other end. Furthermore, all this must be done efficiently and in a way that isolates the upper layers from the inevitable changes in the hardware technology.

The transport layer also determines what type of service to provide to the session layer, and, ultimately, to 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 the transporting of isolated messages, with no guarantee about the order of delivery, and the 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, all the way from the source to the 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 neighbours, and not between the ultimate source and destination machines, which may be separated by many routers.

The Session Layer:

The session layer allows users on different machines to establish sessions between them. Sessions offer various services, including 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), and synchronization (check pointing long transmissions to allow them to continue from where they were after a crash).

The Presentation Layer:

The presentation layer is concerned with the syntax and semantics of the information transmitted. In order to make it possible for computers with different data representations to communicate, the data structures to be exchanged can be defined in an abstract way, along with a standard encoding to be used "on the wire." The presentation layer manages these abstract data structures and allows higher-level data structures (e.g., banking records), to be defined and exchanged.

The Application Layer:

The application layer contains a variety of protocols that are commonly needed by users. One widely-used 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.

Q6. Explain the TCP/IP Reference Model.

The TCP/IP Reference Model:

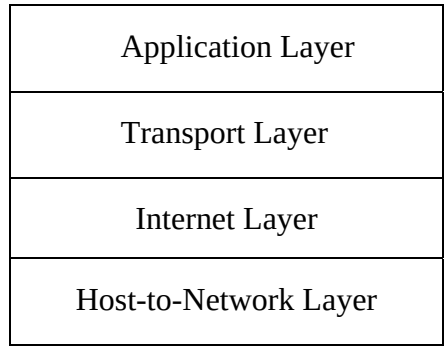
The TCP/IP reference model was developed prior to OSI model. The major design goals of this model were,

1. To connect multiple networks together so that they appear as a single network.
2. To survive after partial subnet hardware failures.
3. To provide a flexible architecture.

Unlike OSI reference model, TCP/IP reference model has only 4 layers. They are,

1. Host-to-Network Layer

2. Internet Layer
3. Transport Layer
4. Application Layer



TCP/IP Reference model

Host-to-Network Layer:

The TCP/IP reference model does not really say much about what happens here, except to point out that the host has to connect to the network using some protocol so it can send IP packets to it. This protocol is not defined and varies from host to host and network to network.

Internet 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 they travel independently to the destination (potentially on a different network). They may even arrive in a different order than they were sent, in which case it is the job of higher layers to rearrange them, if in-order delivery is desired. Note that "internet" is used here in a generic sense, even though this layer is present in the Internet.

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. Packet routing is clearly the major issue here, as is avoiding congestion. For these reasons, it is reasonable to say that the TCP/IP internet layer is similar in functionality to the OSI network layer. Fig.6.1 shows this correspondence.

The Transport Layer:

The layer above the internet layer in the TCP/IP model is now usually called the transport layer. It is designed to allow peer entities on the source and destination hosts to

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

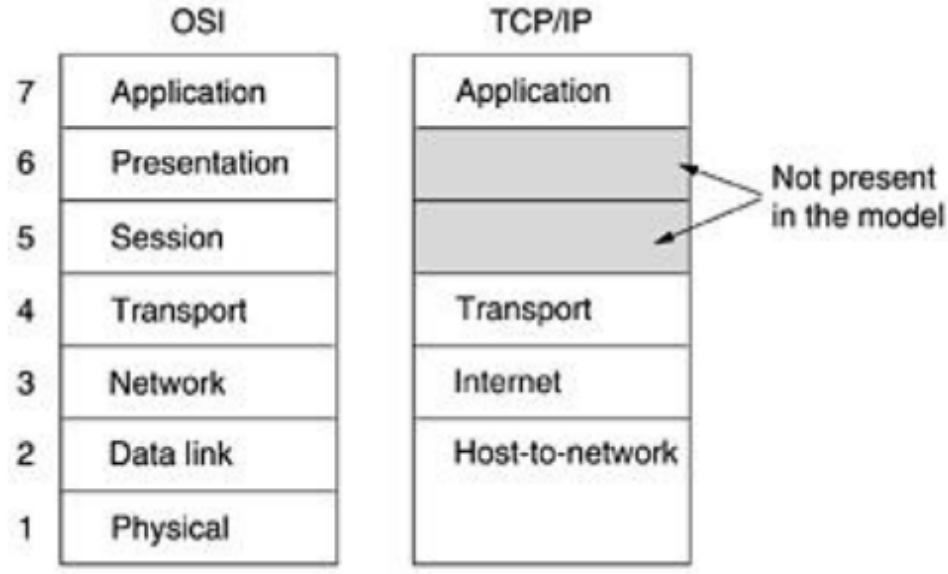


Fig.6.1: The TCP/IP reference model.

The second protocol in this layer, UDP (User Datagram Protocol), is an unreliable, connectionless 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, such as transmitting speech or video. The relation of IP, TCP, and UDP is shown in Fig. 6.2. Since the model was developed, IP has been implemented on many other networks.

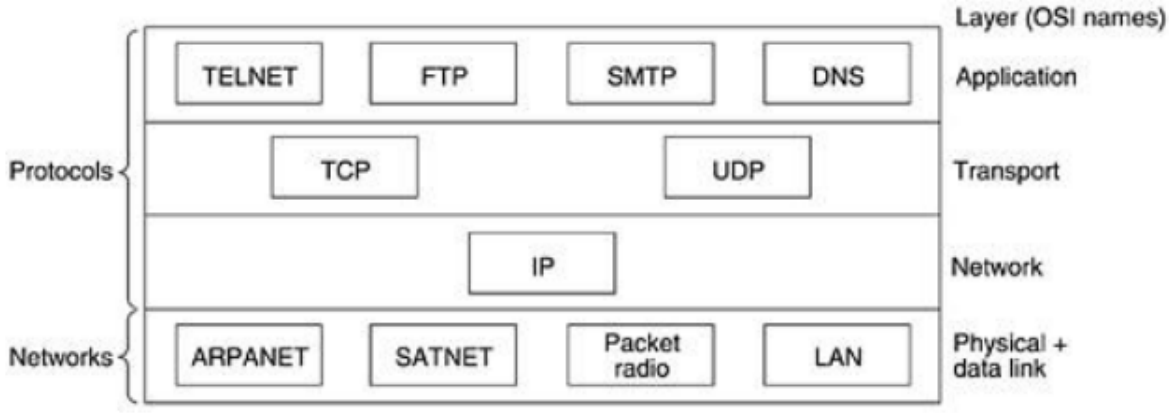


Fig.6.2: Protocols and networks in the TCP/IP model initially.

The Application Layer:

The TCP/IP model does not have session or presentation layers.

On top of the transport layer is the application layer. It contains all the higher-level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP), as shown in Fig.6.2. 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 kind of file transfer, but later a specialized protocol (SMTP) was developed for it.

Many other protocols have been added to these over the years: the Domain Name System (DNS) for mapping host names onto their network addresses, NNTP, the protocol for moving USENET news articles around, and HTTP, the protocol for fetching pages on the World Wide Web, and many others.

Q7. What are the comparisons of the OSI and TCP/IP Reference Models?

Comparison of the OSI and TCP/IP Reference Models:

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

Despite these fundamental similarities, the two models also have many differences Three concepts are central to the OSI model:

1. Services.

2. Interfaces.
3. Protocols.

Probably the biggest contribution of the OSI model is to make the distinction between these three concepts explicit. Each layer performs some services for the layer above it. The service definition tells what the layer does, not how entities above it access it or how the layer works. It defines the layer's semantics.

A layer's interface tells the processes above it how to access it. It specifies what the parameters are and what results to expect. It, too, says nothing about how the layer works inside.

Finally, the peer protocols used in a layer are the layer's own business. It can use any protocols it wants to, as long as it gets the job done (i.e., provides the offered services). It can also change them at will without affecting software in higher layers.

The TCP/IP model did not originally clearly distinguish between service, interface, and protocol, although people have tried to retrofit it after the fact to make it more OSI-like. For example, the only real services offered by the internet layer are SEND IP PACKET and RECEIVE IP PACKET.

As a consequence, the protocols in the OSI model are better hidden than in the TCP/IP model and can be replaced relatively easily as the technology changes. Being able to make such changes is one of the main purposes of having layered protocols in the first place. The OSI reference model was devised before the corresponding protocols were invented.

This ordering means that the model was not biased toward one particular set of protocols, a fact that made it quite general. The downside of this ordering is that the designers did not have much experience with the subject and did not have a good idea of which functionality to put in which layer.

Another difference is in the area of connectionless versus connection-oriented communication. The OSI model supports both connectionless and connection-oriented communication in the network layer, but only connection-oriented communication in the transport layer, where it counts (because the transport service is visible to the users). The TCP/IP model has only one mode in the network layer (connectionless) but supports both modes in the transport layer, giving the users a choice. This choice is especially important for simple request-response protocols.

Q8. Explain the problems in TCP/IP Reference Model.

Problems of the TCP/IP Reference Mode:

First, the model does not clearly distinguish the concepts of service, interface, and protocol. Good software engineering practice requires differentiating between the specification and the implementation, something that OSI does very carefully, and TCP/IP

does not. Consequently, the TCP/IP model is not much of a guide for designing new networks using new technologies.

Second, the TCP/IP model is not at all general and is poorly suited to describing any protocol stack other than TCP/IP. Trying to use the TCP/IP model to describe Bluetooth, for example, is completely impossible.

Third, the host-to-network layer is not really a layer at all in the normal sense of the term as used in the context of layered protocols. It is an interface (between the network and data link layers). The distinction between an interface and a layer is crucial, and one should not be sloppy about it.

Fourth, the TCP/IP model does not distinguish (or even mention) the physical and data link layers. These are completely different. The physical layer has to do with the transmission characteristics of copper wire, fiber optics, and wireless communication. The data link layer's job is to delimit the start and end of frames and get them from one side to the other with the desired degree of reliability. A proper model should include both as separate layers. The TCP/IP model does not do this.

Finally, although the IP and TCP protocols were carefully thought out and well implemented, many of the other protocols were ad hoc, generally produced by a couple of graduate students hacking away until they got tired. The protocol implementations were then distributed free, which resulted in their becoming widely used, deeply entrenched, and thus hard to replace. Some of them are a bit of an embarrassment now. The virtual terminal protocol, TELNET, for example, was designed for a ten-character per second mechanical Teletype terminal. It knows nothing of graphical user interfaces and mice. Nevertheless, 25 years later, it is still in widespread use.

Q9. Explain the problems in OSI Model and Protocols.

Problems of the OSI Model and Protocols:

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

1. Bad Timing:

The time at which a standard is established is absolutely critical to its success. David Clark of M.I.T. has a theory of standards that he calls the apocalypse of the two elephants, which is illustrated in Fig.9.

This figure shows the amount of activity surrounding a new subject. 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, corporations discover the subject, and the billion-dollar wave of investment hits.

It is essential that the standards be written 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. If the interval between the two elephants is very short(because everyone is in a hurry to get started), the people developing the standards may get crushed.

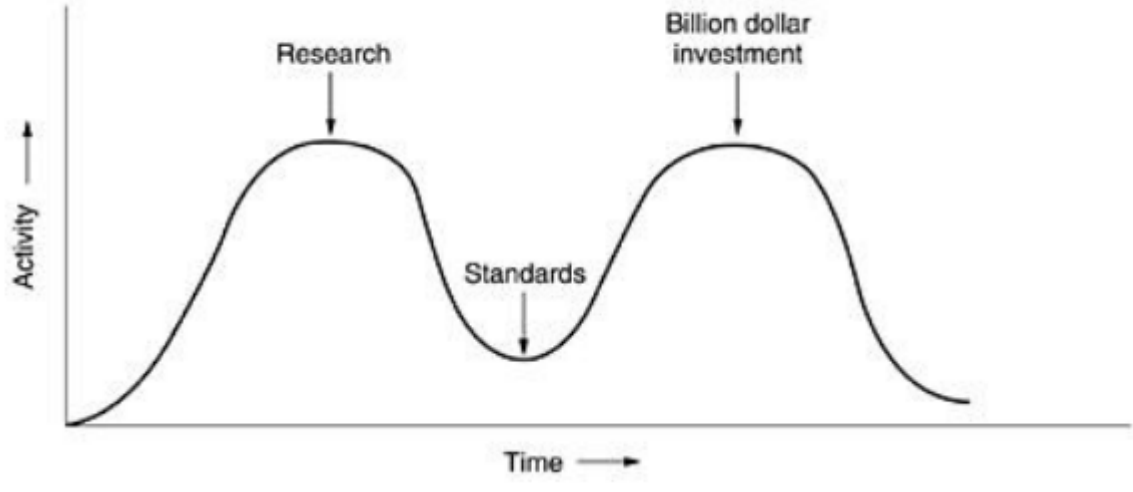


Fig.9: The apocalypse of the two elephants

2. Bad Technology:

The second reason that OSI never caught on is that both the model and the protocols are flawed. 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 definitions and protocols, is extraordinarily complex. When piled up, the printed standards occupy a significant fraction of a meter of paper. They are also difficult to implement and inefficient in operation. In addition to being incomprehensible, another problem with OSI is that some functions, such as addressing, flow control, and error control, reappear again and again in each layer.

3. Bad Implementations:

Given the enormous complexity of the model and the protocols, it will come as no surprise that the initial implementations were huge, unwieldy, and slow. Everyone who tried them got burned. It did not take long for people to associate "OSI" with "poor quality." Although the products improved in the course of time, the image stuck.

4. Bad Politics:

On account of the initial implementation, many people, especially in academia, thought of TCP/IP as part of UNIX, and UNIX in the 1980s in academia was not unlike parenthood (then incorrectly called motherhood) and apple pie.

OSI, on the other hand, was widely thought to be the creature of the European telecommunication ministries, the European Community, and later the U.S. Government. This belief was only partly true, but the very idea of a bunch of government bureaucrats trying to shove a technically inferior standard down the throats of the poor researchers and programmers down in the trenches actually developing computer networks did not help much. Some people viewed this development in the same light as IBM announcing in the 1960s that PL/I was the language of the future, or DoD correcting this later by announcing that it was actually Ada.

Q10. Explain about ARPANET.

ARPANET:

The subnet would consist of minicomputers called IMPs (Interface Message Processors) connected by 56-kbps transmission lines. For high reliability, each IMP would be connected to at least two other IMPs. The subnet was to be a datagram subnet, so if some lines and IMPs were destroyed, messages could be automatically rerouted along alternative paths.

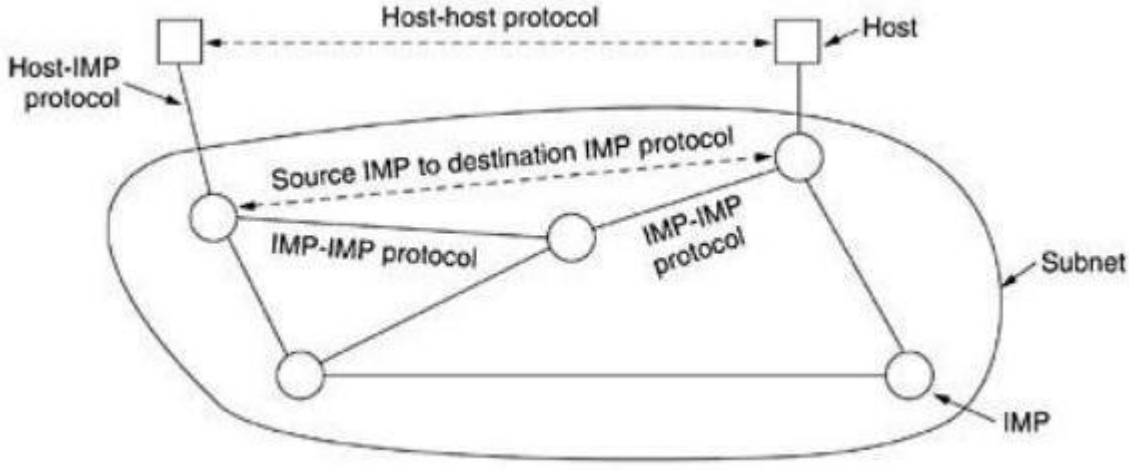


Fig.10: Original ARPANET design

Each node of the network was to consist of an IMP and a host, in the same room, connected by a short wire. A host could send messages of up to 8063 bits to its IMP, which would then break these up into packets of at most 1008 bits and forward them independently toward the destination. Each packet was received in its entirety before being forwarded, so the subnet was the first electronic store-and-forward packet-switching network.

ARPA then put out a tender for building the subnet. Twelve companies bid for it. After evaluating all the proposals, ARPA selected BBN, a consulting firm in Cambridge, Massachusetts, and in December 1968, awarded it a contract to build the subnet and write the subnet software. BBN chose to use specially modified Honeywell DDP-316 minicomputers with 12K 16-bit words of core memory as the IMPs. The IMPs did not have disks, since moving parts were considered unreliable. The IMPs were interconnected by 56-kbps lines leased from telephone companies. Although 56 kbps is now the choice of teenagers who cannot afford ADSL or cable, it was then the best money could buy.

The software was split into two parts: subnet and host. The subnet software consisted of the IMP end of the host-IMP connection, the IMP-IMP protocol, and a source IMP to destination IMP protocol designed to improve reliability. The original ARPANET design is shown in Fig.10. Outside the subnet, software was also needed, namely, the host end of the host-IMP connection, the host-host protocol, and the application software. It soon became clear that BBN felt that when it had accepted a message on a host-IMP wire and placed it on the host-IMP wire at the destination, its job was done.

Q11. Explain the architecture of the Internet.

Architecture of the Internet:

Let us assume client calls his or her ISP over a dial-up telephone line, as shown in Fig. 11. The modem is a card within the PC that converts the digital signals the computer produces to analog signals that can pass unhindered over the telephone system. These signals are transferred to the ISP's POP (Point of Presence), where they are removed from the telephone system and injected into the ISP's regional network. From this point on, the system is fully digital and packet switched. If the ISP is the local Telco, the POP will probably be located in the telephone switching office where the telephone wire from the client terminates. If the ISP is not the local Telco, the POP may be a few switching offices down the road.

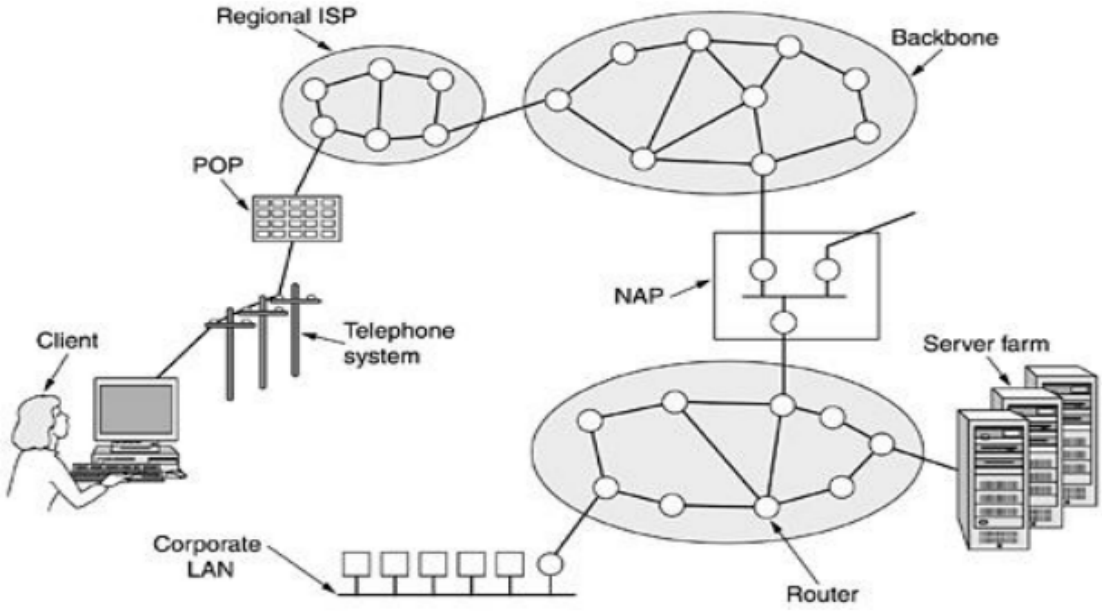


Fig.11: Overview of the Internet

The ISP's regional network consists of interconnected routers in the various cities the ISP serves. If the packet is destined for a host served directly by the ISP, the packet is delivered to the host. Otherwise, it is handed over to the ISP's backbone operator.

At the top of the food chain are the major backbone operators, companies like AT&T and Sprint. They operate large international backbone networks, with thousands of routers connected by high-bandwidth fiber optics. Large corporations and hosting services that run server farms (machines that can serve thousands of Web pages per second) often connect directly to the backbone. Backbone operators encourage this direct connection by renting space in what are called carrier hotels, basically equipment racks in the same room as the router to allow short, fast connections between server farms and the backbone.

If a packet given to the backbone is destined for an ISP or company served by the backbone, it is sent to the closest router and handed off there. However, many backbones, of varying sizes, exist in the world, so a packet may have to go to a competing backbone. To allow packets to hop between backbones, all the major backbones connect at the NAPs discussed earlier. Basically, a NAP is a room full of routers, at least one per backbone. A LAN in the room connects all the routers, so packets can be forwarded from any backbone to any other backbone.

In addition to being interconnected at NAPs, the larger backbones have numerous direct connections between their routers, a technique known as private peering. One of the many paradoxes of the Internet is that ISPs who publicly compete with one another for customers often privately cooperate to do private peering (Metz, 2001).

Q12. Explain the ATM Reference Model.

ATM Reference Model:

ATM has its own reference model, different from the OSI model and also different from the TCP/IP model. This model is shown in Fig. 12.1. It consists of three layers, the physical, ATM, and ATM adaptation layers, plus whatever users want to put on top of that.

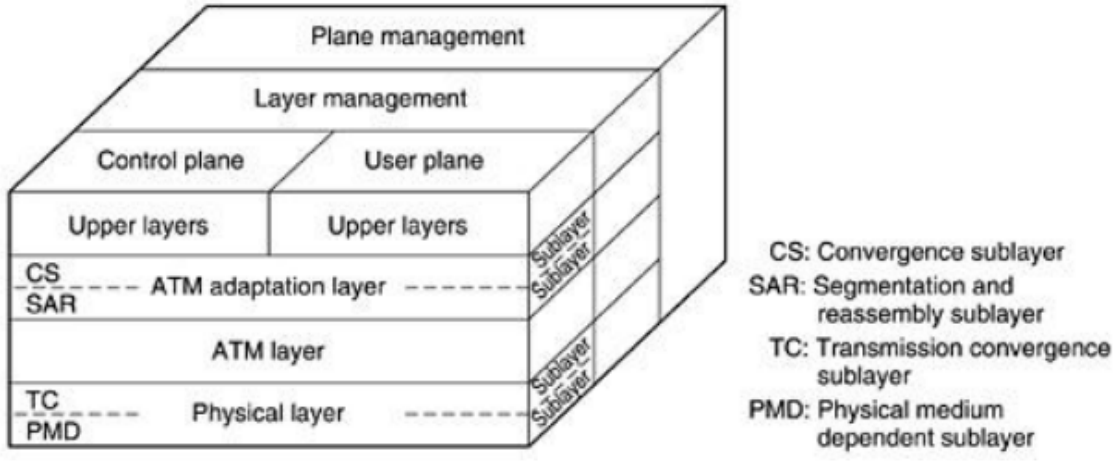


Fig.12.1: The ATM Reference model

The physical layer deals with the physical medium: voltages, bit timing, and various other issues. ATM does not prescribe a particular set of rules but instead says that ATM cells can be sent on a wire or fiber by themselves, but they can also be packaged inside the payload of other carrier systems. In other words, ATM has been designed to be independent of the transmission medium.

The ATM layer deals with cells and cell transport. It defines the layout of a cell and tells what the header fields mean. It also deals with establishment and release of virtual circuits. Congestion control is also located here.

Because most applications do not want to work directly with cells (although some may), a layer above the ATM layer has been defined to allow users to send packets larger than a cell. The ATM interface segments these packets, transmits the cells individually, and reassembles them at the other end. This layer is the AAL (ATM Adaptation Layer).

Unlike the earlier two-dimensional reference models, the ATM model is defined as being three-dimensional, as shown in Fig. 12.1. The user plane deals with data transport, flow control, error correction, and other user functions. In contrast, the control plane is concerned with connection management. The layer and plane management functions relate to resource management and interlayer coordination.

The physical and AAL layers are each divided into two sub layers, one at the bottom that does the work and a convergence sub layer on top that provides the proper interface to the layer above it. The functions of the layers and sub layers are given in Fig. 12.2.

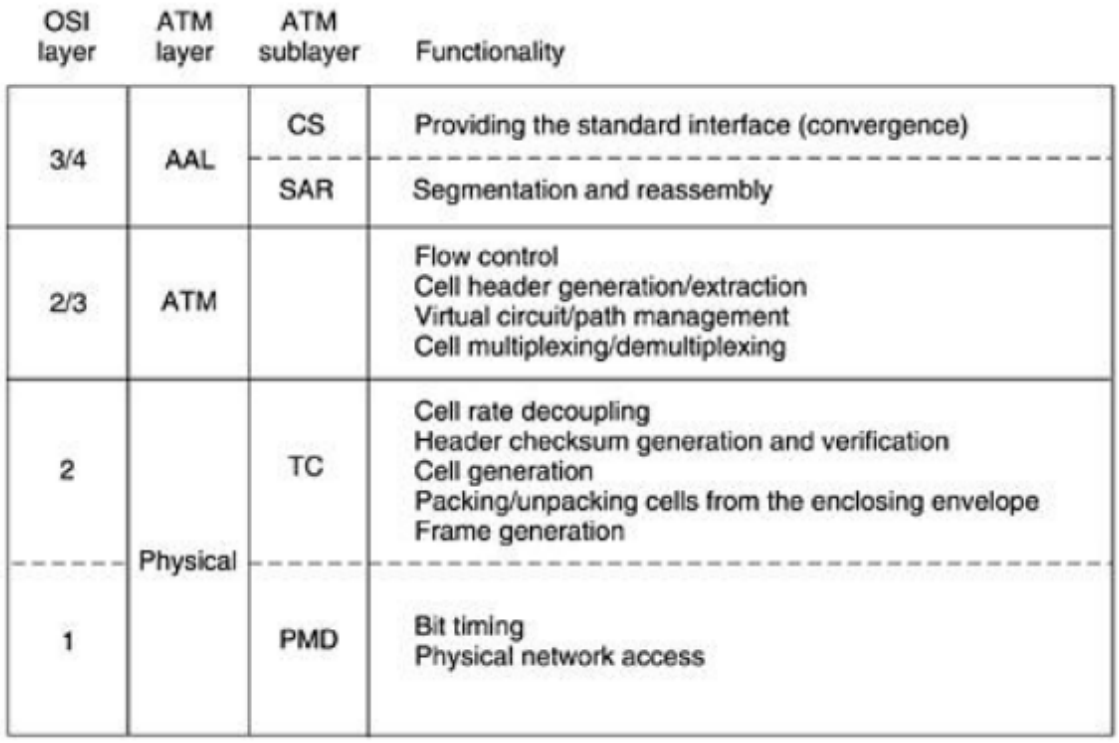


Fig.12.2: The ATM layers and sub layers, and their functions

The PMD (Physical Medium Dependent) sub layer interfaces to the actual cable. It moves the bits on and off and handles the bit timing. For different carriers and cables, this layer will be different.

The other sub layer of the physical layer is the TC (Transmission Convergence) sub layer. When cells are transmitted, the TC layer sends them as a string of bits to the PMD layer. Doing this is easy. At the other end, the TC sub layer gets a pure incoming bit stream from the PMD sub layer. Its job is to convert this bit stream into a cell stream for the ATM layer. It handles all the issues related to telling where cells begin and end in the bit stream. In the ATM model, this functionality is in the physical layer. In the OSI model and in pretty much all other networks, the job of framing, that is, turning a raw bit stream into a sequence of frames or cells, is the data link layer's task.

The ATM layer manages cells, including their generation and transport. Most of the interesting aspects of ATM are located here. It is a mixture of the OSI data link and network layers; it is not split into sub layers.

The AAL layer is split into a SAR (Segmentation And Reassembly) sub layer and a CS (Convergence Sub layer). The lower sub layer breaks up packets into cells on the transmission side and puts them back together again at the destination. The upper sub layer makes it possible to have ATM systems offer different kinds of services to different applications (e.g., file transfer and video on demand have different requirements concerning error handling, timing, etc.).

Q13. Explain about client-server model.

Client-server model:

In this model, the data is stored on powerful computers called servers. Often these are centrally housed and maintained by a system administrator. In contrast, the employees have simpler machines, called clients, on their desks, with which they access remote data, for example, to include in spreadsheets they are constructing. (Sometimes we will refer to the human user of the client machine as the "client," but it should be clear from the context whether we mean the computer or its user.)

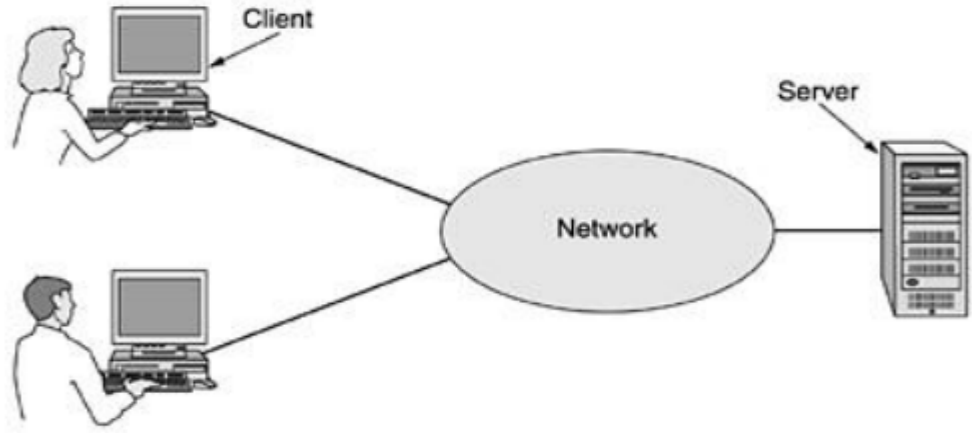


Fig.13.1: A network with two clients and one server

The client and server machines are connected by a network, as illustrated in Fig. 13.1..

This whole arrangement is called the client-server model. It is widely used and forms the basis of much network usage. It is applicable when the client and server are both in the same building (e.g., belong to the same company), but also when they are far apart. For example, when a person at home accesses a page on the World Wide Web, the same model is employed, with the remote Web server being the server and the user's personal computer being the client. Under most conditions, one server can handle a large number of clients.

In the client-server model two processes are involved, one on the client machine and one on the server machine. Communication takes the form of the client process sending 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. These messages are shown in Fig.13.2.

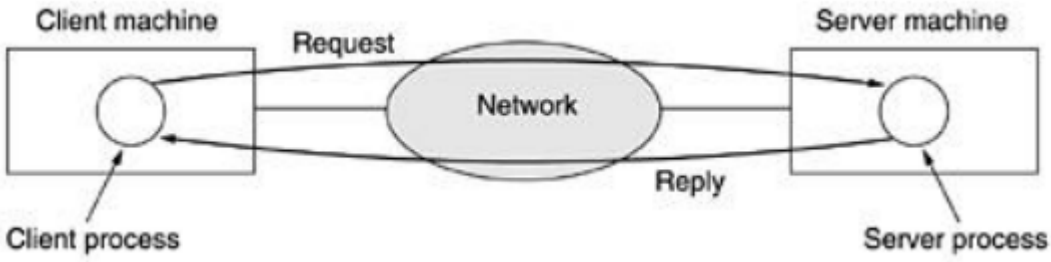


Figure 13.2: The client-server model involves requests and replies.

Q14. Explain about Peer-to-peer Communication.

Peer-to-peer:

The peer-to-peer Communication is a type of person-to-person communication. In this form, individuals who form a loose group can communicate with others in the group, as shown in Fig. 14. Every person can, in principle, communicate with one or more other people; there is no fixed division into clients and servers.

Peer-to-peer communication really hit the big time around 2000 with a service called Napster, which at its peak had over 50 million music fans swapping music, in what was probably the biggest copyright infringement in all of recorded history. The idea was fairly simple. Members registered the music they had on their hard disks in a central database maintained on the Napster server. If a member wanted a song, he checked the database to see who had it and went directly there to get it. By not actually keeping any music on its machines, Napster argued that it was not infringing anyone's copyright. The courts did not agree and shut it down.

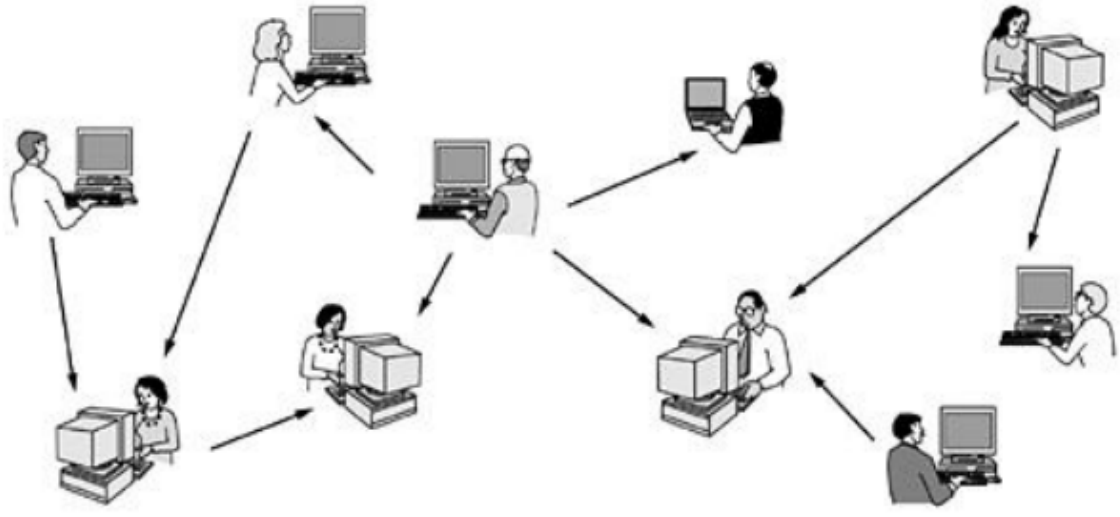


Figure 14: In a peer-to-peer system there are no fixed clients and servers.

However, the next generation of peer-to-peer systems eliminates the central database by having each user maintain his own database locally, as well as providing a list of other nearby people who are members of the system.

A new user can then go to any existing member to see what he has and get a list of other members to inspect for more music and more names. This lookup process can be repeated indefinitely to build up a large local database of what is out there. It is an activity that would get tedious for people but is one at which computers excel.

Legal applications for peer-to-peer communication also exist. For example, fans sharing public domain music or sample tracks that new bands have released for publicity purposes, families sharing photos, movies, and genealogical information, and teenagers playing militiaperson on-line games. In fact, one of the most popular Internet applications of all, e-mail, is inherently peer-to-peer.

Q15. Explain about transmission technologies.

Transmission technologies:

There are two types of transmission technology that are in widespread use. They are as follows:

1. Broadcast links.

2. Point-to-point links.

Broadcast networks have a single communication channel that is shared by all the machines on the network. Short messages, called packets in certain contexts, sent by any machine are received by all the others. An address field within the packet specifies the intended recipient.

Upon receiving a packet, a machine checks the address field. If the packet is intended for the receiving machine, that machine processes the packet; if the packet is intended for some other machine, it is just ignored.

As an analogy, consider someone standing at the end of a corridor with many rooms off it and shouting "Watson, come here. I want you." Although the packet may actually be received(heard) by many people, only Watson responds. The others just ignore it. Another analogy is an airport announcement asking all flight 644 passengers to report to gate 12 for immediate boarding.

Broadcast systems generally also allow the possibility of addressing a packet to all destinations by using a special code in the address field. When a packet with this code is transmitted, it is received and processed by every machine on the network. This mode of operation is called broadcasting. Some broadcast systems also support transmission to a subset of the machines, something known as multicasting. One possible scheme is to reserve one bit to indicate multicasting. The remaining n - 1 address bits can hold a group number.

Each machine can "subscribe" to any or all of the groups. When a packet is sent to a certain group, it is delivered to all machines subscribing to that group.

In contrast, point-to-point networks consist of many connections between individual pairs of Machines. To go from the source to the destination, a packet on this type of network may have to first visit one or more intermediate machines. Often multiple routes, of different lengths, are possible, so finding good ones is important in point-to-point networks. As a general rule (although there are many exceptions), 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 sometimes called uncasing.

Q15. Differentiate Connection-Oriented and Connectionless Services.

Layers can offer two different types of service to the layers above them:

1. Connection-oriented and

2. Connectionless.

In this section we will look at these two types and examine the differences between them.

1. Connection-oriented service:

Connection-oriented service is modelled after the telephone system. To talk to someone, 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 at the other end. In most cases the order is preserved so that the bits arrive in the order they were sent.

In some cases when a connection is established, the sender, receiver, and subnet conduct a negotiation about parameters to be used, such as maximum message size, quality of service required, and other issues. Typically, one side makes a proposal and the other side can accept it, reject it, or make a counterproposal. 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.

2. Connectionless service:

In contrast, connectionless service is modelled after the postal system. Each message (letter) 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.

Not all applications require connections. For example, as electronic mail becomes more common, electronic junk is becoming more common too. The electronic junk-mail sender probably does not want to go to the trouble of setting up and later tearing down a connection just to send one item. Nor is 100 percent reliable delivery essential, especially if it costs more All that is needed is a way to send a single message that has a high probability of arrival, but no guarantee. Unreliable (meaning not acknowledged) connectionless service is often called datagram service, in analogy with telegram service, which also does not return an acknowledgement to the sender.

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. Very few file transfer customers would prefer a service that occasionally scrambles or loses a few bits, even if it is much faster.

Q16. Write short notes on interface, service and protocol.

Protocol Hierarchies:

To reduce their design complexity, most networks are organized as a stack of layers or levels, each one built upon the one below it. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network. The purpose of each layer is to offer certain services to the higher layers, shielding those layers from the details of how the offered services are actually implemented. In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it.

This concept is actually a familiar one and used throughout computer science, where it is variously known as information hiding, abstract data types, data encapsulation, and object-oriented programming. The fundamental idea is that a particular piece of software (or hardware) provides a service to its users but keeps the details of its internal state and algorithms hidden from them.

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. As an analogy, when a woman is introduced to a man, she may choose to stick out her hand. He, in turn, may decide either to shake it or kiss it, depending, for example, on whether she is an American lawyer at a business meeting or a European princess at a formal ball. Violating the protocol will make communication more difficult, if not completely impossible.

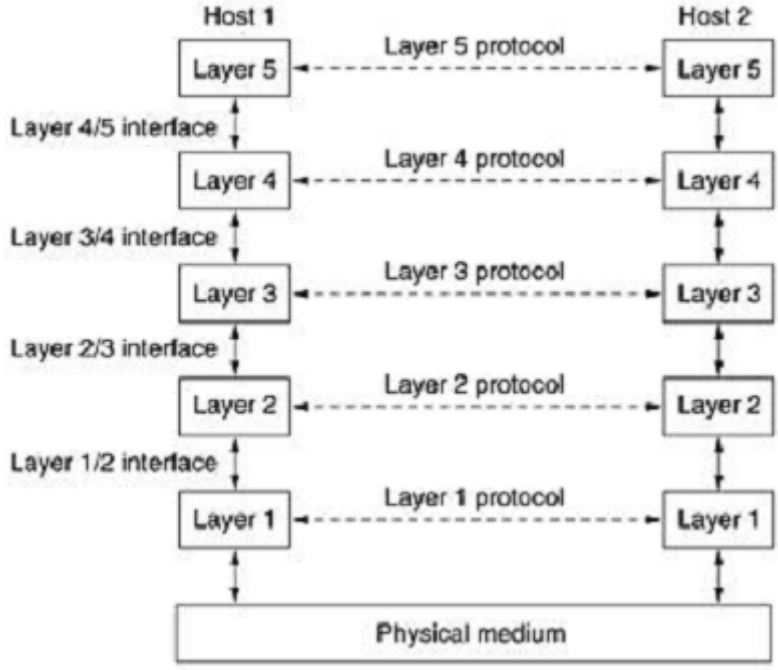
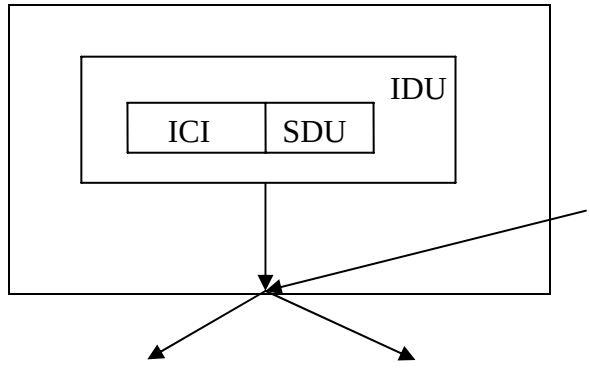


Fig.16.1: Layers, protocols, and interfaces.

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. In Fig.16.1, virtual communication is shown by dotted line.

Interfaces:

Between each pair of adjacent layers is an interface. The interface defines which primitive operations and services the lower layer makes available to the upper one. When network designers decide how many layers to include in a network and what each one should do, one of the most important considerations is defining clean interfaces between the layers. Doing so, in turn, requires that each layer perform a specific collection of well-understood functions.



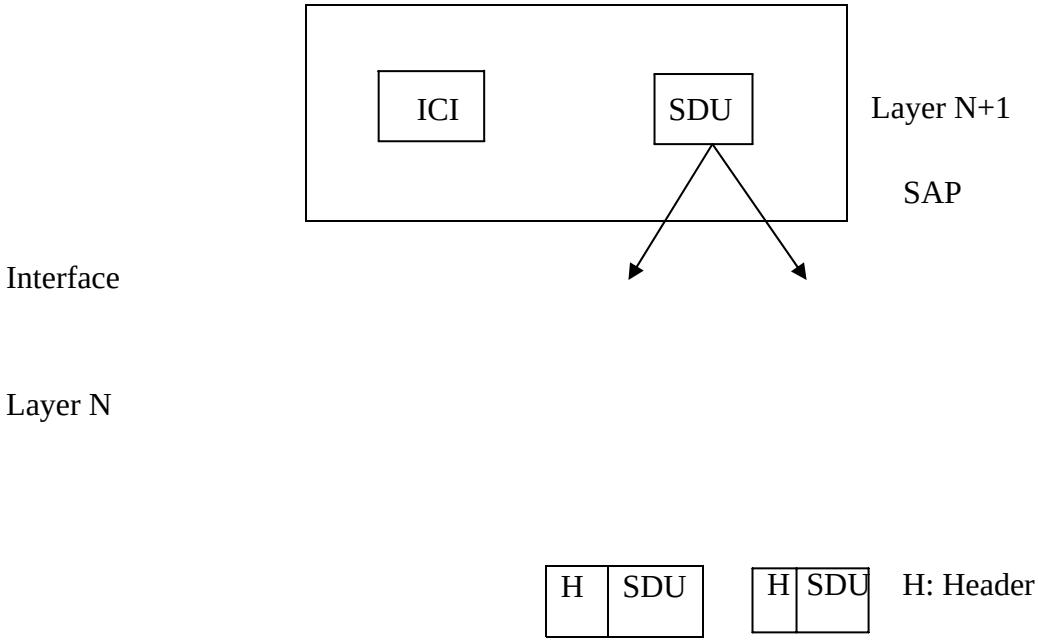


Fig.16.2: Relation between Layers at an Interface

Services:

Layers can offer two different types of service to the layers above them are,

1. Connection-oriented and
2. Connectionless.

Request-reply is commonly used to implement communication in the client-server model: the client issues a request and the server responds to it. Figure 16.3 summarizes the types of services discussed above.

	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.16.3: Six different types of service.

Service primitives:

A service is formally specified by a set of primitives (operations) available to a user process to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity. If the protocol stack is located in the operating system, as it often is, the primitives are normally system calls. These calls cause a trap to kernel mode, which then turns control of the machine over to the operating system to send the necessary packets. Five service primitives for implementing a simple connection-oriented service are shown in fig.16.4.

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.16.4: Five service primitives for implementing a simple connection-oriented service.

17. Write a brief note on frame relay and X.25 networks.

In the 1980s, the X.25 networks were largely replaced by a new kind of network called frame relay. The essence of frame relay is that it is a connection-oriented network with no error control and no flow control. Because it was connection-oriented, packets were delivered in order (if they were delivered at all). The properties of in-order delivery, no error control, and no flow control make frame relay akin to a wide area LAN. Its most important application is interconnecting LANs at multiple company offices. Frame relay enjoyed a modest success and is still in use in places today.

Our first example of a connection-oriented network is X.25, which was the first public data network. It was deployed in the 1970s at a time when telephone service was a monopoly everywhere and the telephone company in each country expected there to be one data network per country—theirs. To use X.25, a computer first established a connection to the remote computer, that is, placed a telephone call. This connection was given a connection number to be used in data transfer packets (because multiple connections could be open at the same time). Data packets were very simple, consisting of a 3-byte header and up to 128 bytes of data. The header consisted of a 12-bit connection number, a packet sequence number, an acknowledgement number, and a few miscellaneous bits. X.25 networks operated for about a decade with mixed success.

18. What are the applications of computer networks?

1. Information:

One of the applications of computer networks *is* the ability to provide *access* to remote information.

- Pay bills; carry out transactions on bank accounts etc.
- Shop from home by inspecting the catalogs of thousands of companies available online.
- Ask the newspaper for full information about your interesting topics such as corrupt politicians, big fires, football and so on.
- Access information about health, science, art, business, cooking, sports, travel, and government and so on. All this is available on the information systems like the World Wide Web (WWW).

2. Communication:

The popular application of computer networks is electronic mail or e-mail which widely used by millions of people to send and receive text messages. With real-time e-mail, remote users can Communicate even by see and hear each other at the same time. It is also possible to have virtual meetings called videoconference on-line among remote users.

3. Entertainment:

A huge and growing application is entertainment. It entertains people by allowing video demand, and has multiple real-time games etc.

19. Write any four reasons for using layered protocols.

1. Design:

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

2. Change:

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

3. Learning:

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

4. Communication:

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

5. Standards:

It is the most important reason for using a layered model. A layered model provides a guideline and framework not a rigid standard to be used by the various vendors when creating

their products. This is important for interoperability between the various vendors products hat perform different data communication tasks.

20. **With suitable example explain simplex, half-duplex and full-duplex communication.**

Communication between two hosts can be in one of the following ways,

(i) Simplex

(ii) Half-duplex

(iii) Full-duplex.

(i) Simplex Mode:

In this type of transmission mode, the data can be transmitted only in one direction at any time i.e., this type of communication is unidirectional in which one host transmits the data and the other host receives it. The complete channel bandwidth is utilized. Monitors and keyboards are the examples of simplex devices because at any lime the input can be inserted through the keyboard which is accepted by the monitor.

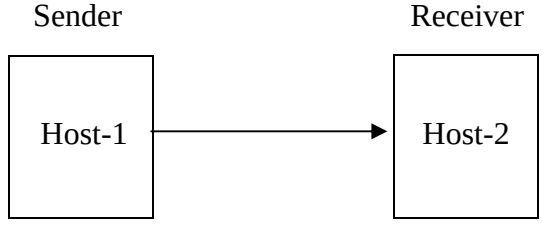


Fig.20.1: Simplex Date Transmission

(ii) Half-duplex Mode:

In this type of transmission mode, the data can be transmitted in both the directions but only one host can transmit at any time. ibis type of data transmission is used when no data has to be transmitted in both directions simultaneously. The entire channel bandwidth can be used for each direction.

Walkie-talkies and citizens based radios are the examples of half-duplex devices.

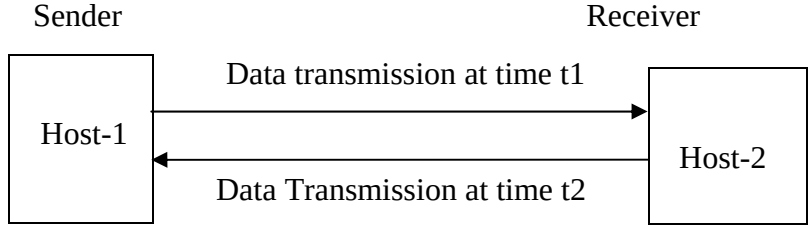


Figure (2): Half duplex Transmission

(iii) Full-duplex Mode:

In this type of data transmission, the data can be transmitted in both the directions at the same time. This mode is suitable for simultaneous bidirectional communication. The entire channel bandwidth is divided among the two directions in which the data is transmitted.

Telephone conversation is an example of full-duplex data transmission in which both the persons can talk and listen at the same time.

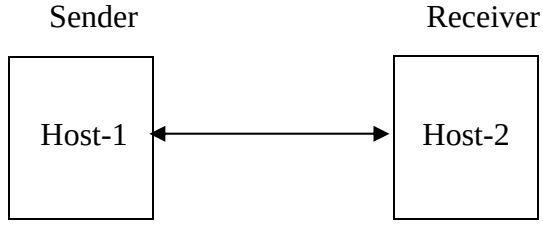


Figure (3): Full duplex Transmission

21. Give a detailed description of the Novell NetWare IPX packet.

Novell NetWare is a popular network in the PC world. It was designed for companies who wanted to have a network of PCs. In this system some PCs functioned as clients and other powerful PCs functioned as servers. It is based on the idea of client server model.

The architecture of Novell NetWare is shown in figure. It looks more like TCP/IP reference model than OSI reference model.

SAP	File server	
NCP	SPX	
IPX		
Ethernet	Token Ring	FDDI
Ethernet	Token Ring	FDDI

Novell NetWare model

The various standards such as internet, IBM token ring, FDDI, PPP and ARC net can be chosen for physic and data link layer.

An unreliable connectionless internetwork protocol called IPX (Internet Packet Exchange) runs on the network layer. It is used to transfer the packets from sour to destination. The source and destination can reside ii different networks. IPX is similar o IP (Internet Protocol in functionality. It uses 12-byte address whereas IP uses only 4-bytes addresses.

On the transport layer a connection-oriented transport protocol called NCP (Network Core Protocol) runs. NCP is the heart or Novell NetWare that provide various services in addition to the data tratisol1. A second protocol that can run above IPX is SPX (Sequenced Packet Exchange). It provides only the data transport. Another option available is TCP (Transmission Control Protocol Any one of the three can be chosen by an application. For example, the tile system application uses NCP and SPX used by Lotus Notes.

There are no session and presentation layers in the NetWare. On the application layers various application layers various application protocols are available such as SAP, file server etc.

22. What are the various types of network topology? What are the implications of having different topology?

Network topologies:

Network topology defined as the logical connection of various computers in the network. The six basic network topologies are: bus, ring, star, tree, mesh and hybrid.

1. Bus Topology:

In bus topology all the computers are connected to a long cable called a bus. A node that wants to send data puts the data on the bus which carries it to the destination node. In this topology any computer can data over the bus at any time. Since, the bus is shared among all the computers. When two or more computers to send data at the same time, an arbitration mechanism is needed to prevent simultaneous access to the bus.

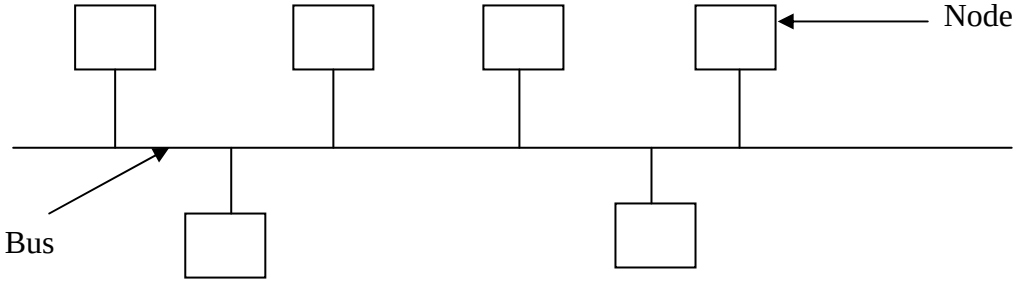


Figure 22.1: Bus Topology

A bus topology is easy to install but is not flexible i.e., it is difficult to add a new node to bus. In addition to this the bus stops functioning even if a portion of the bus breaks down. It is also very difficult to isolate fault.

2. Ring Topology:

In ring topology, the computers are connected in the form of a ring. Each node has exactly two adjacent neighbors. To send data to a distant node on a ring it passes through many intermediate nodes to reach to its ultimate destination.

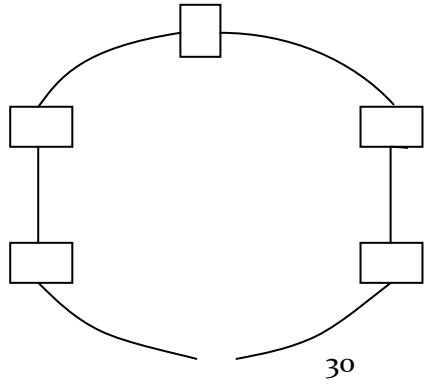


Figure 22.2: Ring Topology

A ring topology is as to install and reconfigure. In this topology, fault isolation is easy because a signal that circulates all the time in a ring helps in identifying a faulty node.

The data transmission takes place in only one direction. When a node fails in ring, it breaks down the whole ring. To overcome this drawback some ring topologies use dual rings. The topology is not useful to connect large number of computers.

3. Star Topology:

In star topology all the nodes are connected to a central node called a hub. A node that wants to send some six data to some other node on the network, send data to a hub which in turn sends it the destination node. A hub plays a major role in such networks.

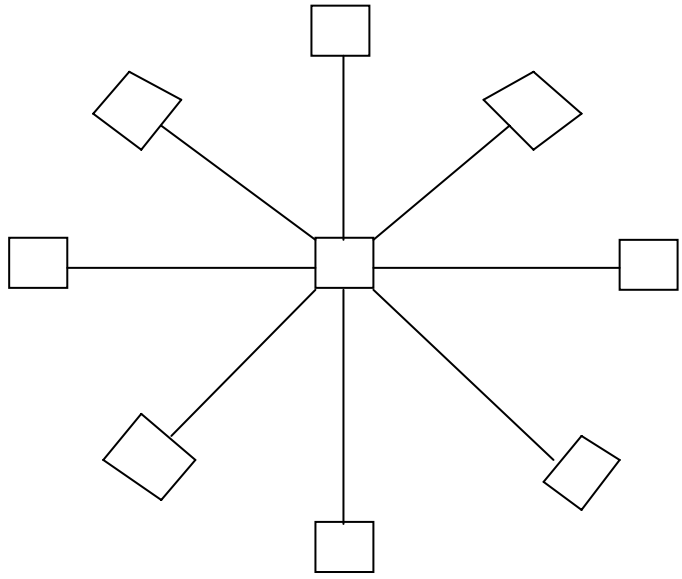
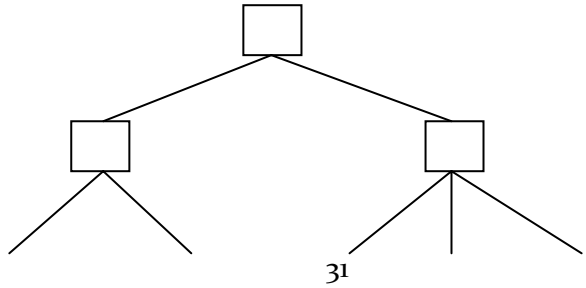


Figure 22.3: Star Topology

Star topology is easy to install and reconfigure. If a link fails then it separates the node connected to link from the network and the network continues to function. However, if the hub goes down, the entire network collapses.

4. Tree Topology:

Tree topology is a hierarchy of various hubs. The entire nodes are connected to one hub or the other. There is a central hub to which only a few nodes are connected directly.



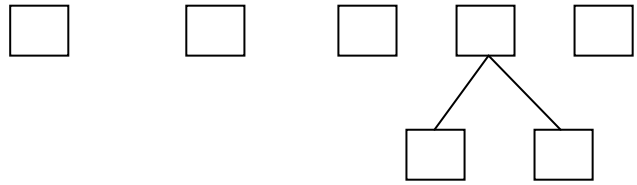


Figure 22.4: Tree topology

The central hub, also called active hub, looks at the incoming bits and regenerates them so that they can traverse over longer distances. The secondary hubs in tree topology may be active hubs or passive hubs. The failure of a transmission line separates a node from the network.

5. Mesh Topology:

A mesh topology is also called complete topology. In this topology, each node is connected directly to every oilier node in the network. That is if there are n nodes then there would be n(n — 1)/2 physical links in the network.

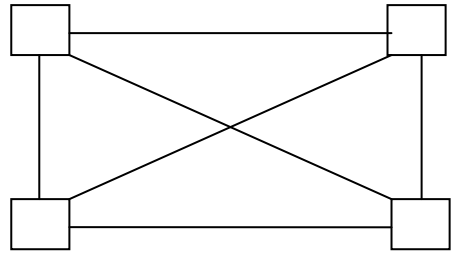


Figure 22.5: Mesh Topology

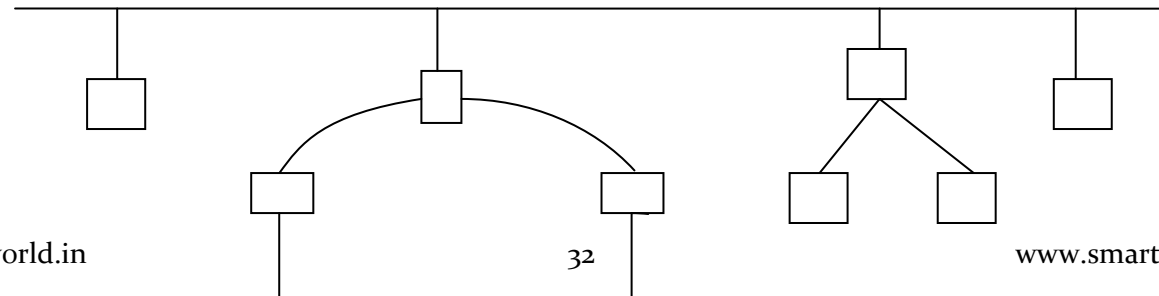
As there are dedicated links, the topology does not have congestion problems. Further it does not need a special Media Access Control (MAC) protocol to prevent simultaneous access to the transmission media since links are dedicated, not shared. The topology also provides data security.

The network can continue to function even in the failure of one of the links. Fault identification is also easy.

The main disadvantage of mesh topology is the complexity of the network and the cost associated with the cable length. The mesh topology is not useful for medium to large networks.

6. Hybrid Topology:

Hybrid topology is formed by connecting two or more topologies together. For example, hybrid topology can be created by using the bus, star and ring topologies, as shown in figure 22.6.



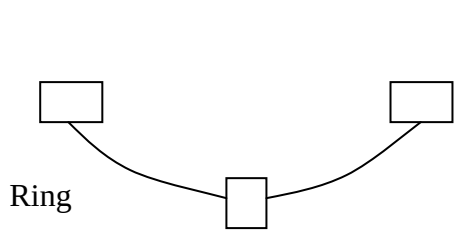


Figure 22.6: Hybrid Topology