Module 1
Introduction: - Types of Computer Networks, Network Software - Protocol Hierarchies, Connection oriented and Connection less hierarchies, Reference Models - ISO-OSI Reference Model, TCP/IP Reference Model – Comparison of OSI and TCP/IP reference models. **Physical Layer: -** Guided Transmission Media– Twisted Pair, Coaxial and Fiber Optics, Wireless Transmission- Radio and Microwave transmission, Communication Satellites – GEO,MEO, LEO. Comparison of Network hardware - Repeaters, Routers, Bridges, Gateways, Hub and Cable Modem

COMPUTER NETWORK

A **computer network** is a set of computers connected together for the purpose of sharing resources. Computer Network is combination of hardware, software, and cabling, which together allow multiple computing devices to communicate (movement of meaningful **information**) with each other

Advantages of Network

- **Speed**. Sharing and transferring files within Networks are very rapid. Thus saving time, while maintaining the integrity of the file.
- **Cost**. Individually licensed copies of many popular software programs can be costly. Networkable versions are available at considerable savings. Shared programs, on a network allows for easier upgrading of the program on one single file server, instead of upgrading individual workstations.
- **Security**. Sensitive files and programs on a network are passwords protected or designated as "copy inhibit," so that you do not have to worry about illegal copying of programs.
- **Centralized Software Management**. Software can be loaded on one computer (the file server) eliminating that need to spend time and energy installing updates and tracking files on independent computers throughout the building.
- **Resource Sharing.** Resources such as, printers, fax machines and modems can be shared.
- **Electronic Mail.** E-mail aids in personal and professional communication.
- **Flexible Access.** Access their files from computers throughout the firm.
- **Workgroup Computing**. Workgroup software (such as Microsoft BackOffice) allows many users to work on a document or project concurrently.

Categories (Types) and connections of networks

TYPES OF NETWORK

Networks are discussed in terms of their (i) transmission technology/type of connection and their (ii) scale / size

Types of Network Based on the *Transmission Technology* / Type of Connection

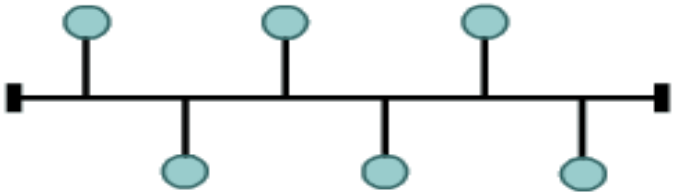
1 .Broadcast Networks (Multi point)

2 .Point-to-point Networks

1 .Broadcast Networks (Multi point)

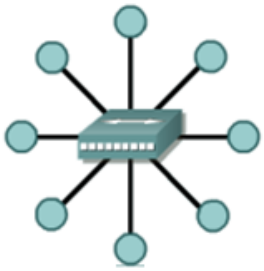
Single communication channel shared by all computers

- Packets (short messages) sent by a computer contains an address specifying the destination computer
- All computers connected to the network receive the packet. The destination computer processes the packet while all other computers discard the packet



- In multipoint (multi drop) connection the capacity of channel is shared either temporary or spatially

2. Point-to-Point Networks



- Consists of many connections between individual pairs of computers
- Sending a packet between two computer may involve the packet being forwarded by intermediate computers
- Multiple routes of different lengths possible, therefore routing algorithms are important

Types of Network Based on the SCALE / SIZE

- PAN (Personal Area Network)
- LAN (Local Area Network),
- MAN (Metropolitan area network)
- WAN (Wide Area Network)

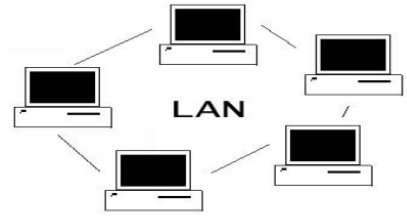
PAN (Personal Area Network)

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

A wireless personal area network (WPAN) is a low-powered PAN carried over a short-distance wireless network technology such as Bluetooth, ZigBee etc.

LAN (Local area networks)

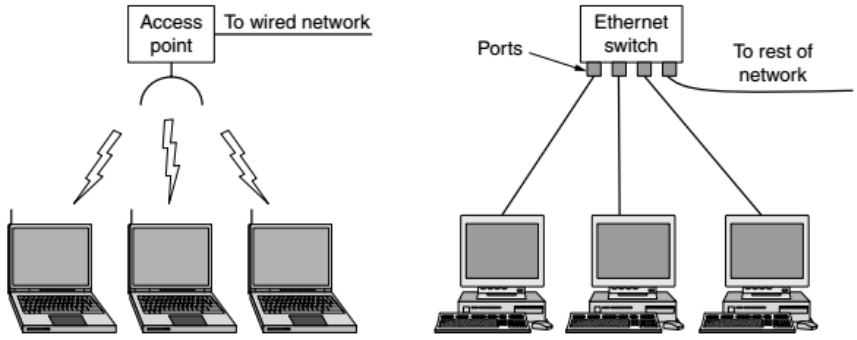
Generally called **LANs**, are privately-owned **networks within a single building or campus of up to a few kilometers in size.** They are widely used to connect personal computers and workstations in company offices and factories to share resources (e.g., printers) and exchange information.



LAN configuration consists of:

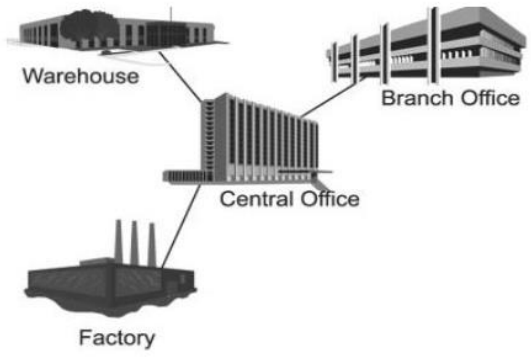
- A file server
- A workstation
- Cables

The topology of many wired LANs is built from point-to-point links. IEEE 802.3, popularly called **Ethernet**



Wireless and wired LANs. (a) 802.11. (b) Switched Ethernet.

MAN (Metropolitan area network)

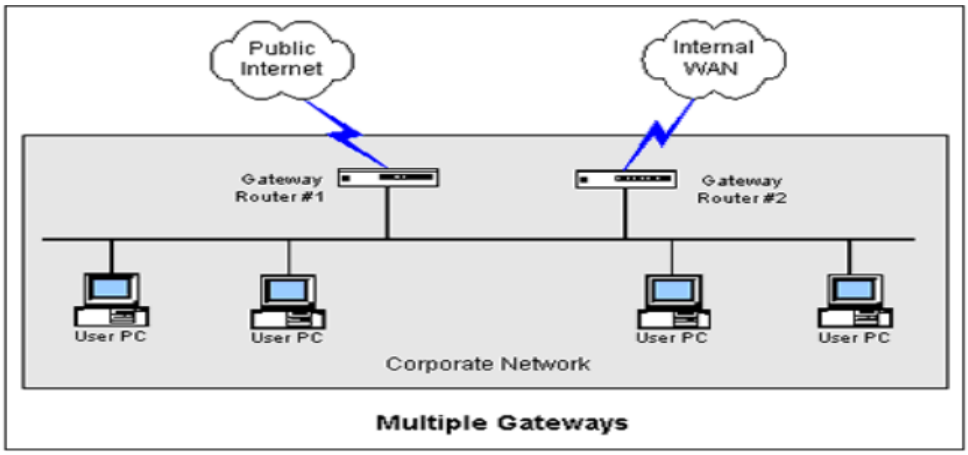


A metropolitan area network (MAN) is a large **computer network that usually spans a city or a large campus.** A MAN usually interconnects a number of local area networks (LANs) using a high-capacity backbone technology, such as fiber-optical links, and provides uplink services to wide area networks and the Internet

Eg. A metropolitan area network based on cable TV.

WAN (Wide Area Network)

A **WAN** spans a **large geographic area, such as a state, province or country.** WANs often connect multiple smaller networks, such as local area networks (LANs) or metro area networks (MANs). Eg. ISP network.



Internetwork or internet

A collection of interconnected networks is called an internetwork or internet. These terms will be used in a generic sense, in contrast to the worldwide Internet (which is one specific internet, which we will always capitalize. The Internet uses ISP networks to connect enterprise networks, home networks, and many other networks.

Connecting a LAN and a WAN or connecting two LANs is the usual way to form an internetwork, A gateway is used to performs necessary translation, both in terms of hardware and software to establish a connection between two or more networks.

Types of Network Based on the SCALE / SIZE (Summary)

	Name of Network	Inter-processor distance	Processor Located in the same
1	PAN	1m	Square meter
2	LAN	10m	Room
		100m	Building
		1Km	Campus
3	MAN	10 Km	City
4	WAN	100 Km	Country
		1000 Km	Continent
5	Internet	10,000 km	Planet

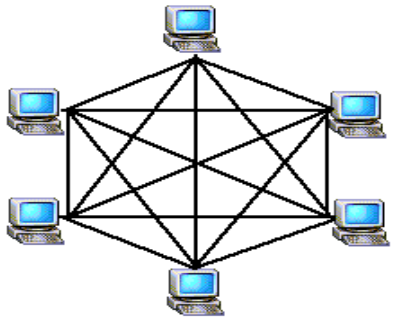
NETWORK TOPOLOGIES

Topology refers to the way a network is laid out either physically or logically. Two or more devices connect to a link; two or more links form a topology. It is the geographical representation of the relationship of all the links and linking devices to each other.

1. Mesh Topology
2. Star Topology
3. Tree (Extended Star) Topology
4. Bus Topology
5. Ring Topology
6. Hybrid Topology

1. Mesh Topology:

Here every device has a dedicated point to point link to every other device. A fully connected mesh can have n(n-1)/2 physical channels to link n devices. It must have n-1 IO ports.



Advantages:

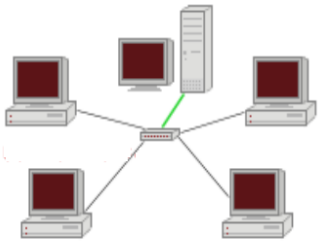
1. They use dedicated links so each link can only carry its own data load. So traffic problem can be avoided.
2. It is robust. If any one link get damaged it cannot affect others
3. It gives privacy and security
4. Fault identification and fault isolation are easy.

Disadvantages:

1. The amount of cabling and the number IO ports required are very large. Since every device is connected to each other devices through dedicated links.
2. The sheer bulk of wiring is larger then the available space
3. Hardware required to connect each device is highly expensive.

2. STAR TOPOLOGY:

Here each device has a dedicated link to the central 'hub'. There is no direct traffic between devices. The transmission are occurred only through the central controller namely hub.



Advantages:

1. Less expensive then mesh since each device is connected only to the hub.
2. Installation and configuration are easy.
3. Less cabling is need then mesh.
4. Robustness.
5. Easy to fault identification & isolation.

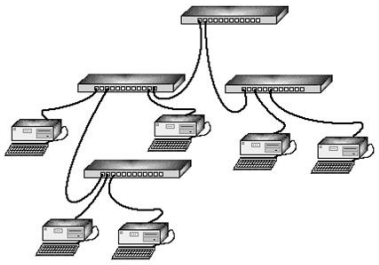
Disadvantages:

1. Even it requires less cabling then mesh when compared with other topologies it still large.

3. TREE TOPOLOGY (Extended Star Topology)

It is a variation of star. Instead of all devices connected to a central hub here most of the devices are connected to a secondary hub that in turn connected with central hub. The central hub is an active hub. An active hub contains a repeater, which regenerate the received bit pattern before sending.

The secondary hub may be active or passive. A passive hub means it just precedes a physical connection only

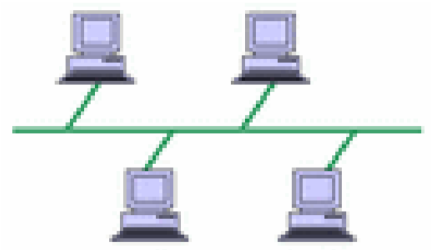


Advantages:

1. Can connect more than star.
2. The distance can be increased.
3. Can isolate and prioritize communication between different computers.

4. BUS TOPOLOGY:

A bus topology is multipoint. Here one long cable is act as a backbone to link all the devices are connected to the backbone by drop lines and taps. A drop line is the connection between the devices and the cable. A tap is the splice into the main cable or puncture the cover.



Advantages:

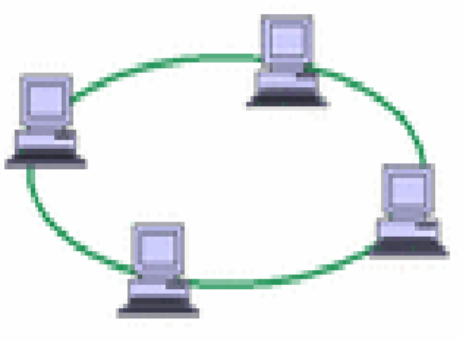
1. Ease of installation.
2. Less cabling

Disadvantages:

1. Difficult reconfiguration and fault isolation.
2. Difficult to add new devices.
3. Signal reflection at top can degradation in quality
4. If any fault in backbone can stops all transmission

5. RING TOPOLOGY

Each node is connected to exactly two other nodes, forming a ring. Can be visualized as a circular configuration. Requires at least three nodes



Advantages:
1. Easy to install.
2. Easy to reconfigure.
3. Fault identification is easy.

Disadvantages:
1. Unidirectional traffic.
2. Break in a single ring can break entire network

6. HYBRID TOPOLOGY

A combination of any two or more network topologies.

NETWORK SOFTWARE

Protocol Hierarchies

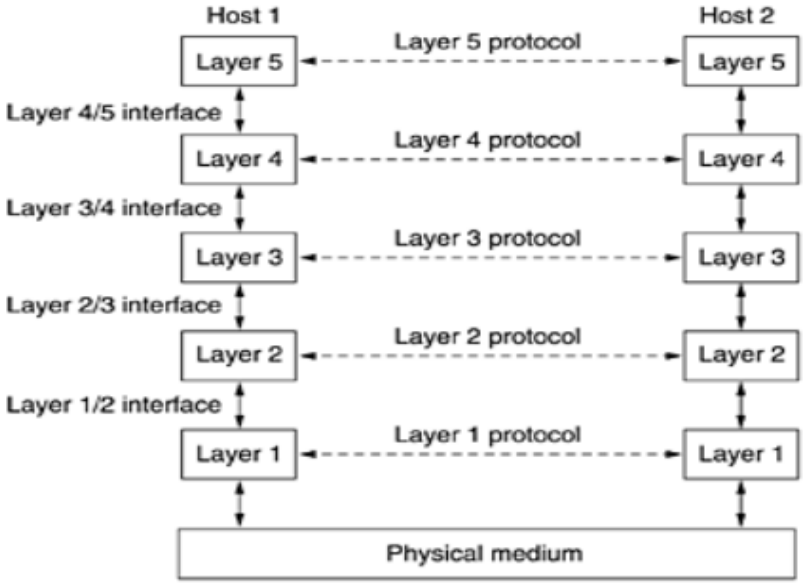
Layering

- 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.
- 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.

Layering provides two nice features.

- It decomposes the problem of building a network into more manageable components. Rather than implementing a monolithic piece of software that does everything implement several layers, each of which solves one part of the problem
- It provides more modular design. To add some new service, it is enough to modify the functionality at one layer, reusing the functions provided at all the other layers.

A five-layer network is illustrated in figure below. The entities comprising the corresponding layers on different machines are called peers. The peers may be processes, hardware devices, or even human beings. In other words, it is the peers that communicate by using the protocol.



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.

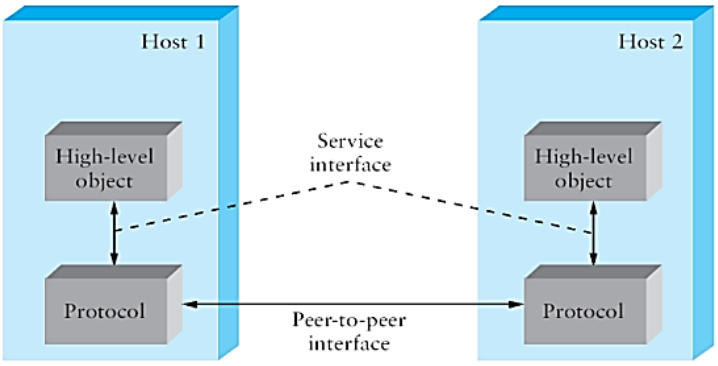
No data are directly transferred from layer n on one machine to layer n on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs. A set of layers and protocols is called **network architecture**. A list of protocols used by a certain system, one protocol per layer, is called a **protocol stack**.

Protocols

A protocol is an agreement between the communicating parties on how communication is to proceed. A protocol is a set of rules that governs data communication. It defines what is communicated, how it is communicated, and when it is communicated. The key elements of a protocol are syntax, semantics and timing.

Each protocol defines two different interfaces.

- Service interface - to the other objects on the same computer that want to use its communication services. This service interface defines the operations that local objects can perform on the protocol.
- Peer interface - to its counterpart (peer) on another machine. It also defines the form and meaning of messages exchanged between protocol peers to implement the communication service.



Encapsulation

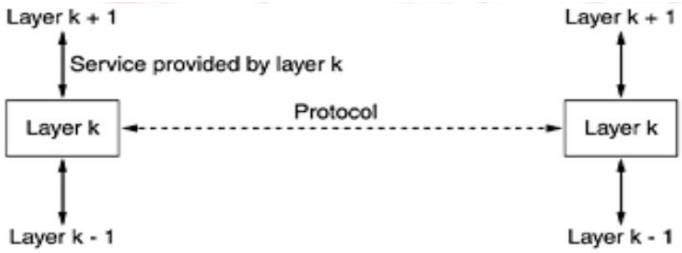
Control information must be added with the data to instruct the peer how to handle with the received message. It will be added into the header or trailer.

- Header - Small data structure from few bytes to few kilobytes attached to the front of message.
- Trailer – Information will be added at the end of the message
- Payload or message body – Data send by the program In this case data is encapsulated with new message created by protocol at each level.

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 on 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, is a set of rules governing the format and meaning of the packets, or messages that are exchanged by the peer entities within a layer. Entities use protocols 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.



CONNECTION-ORIENTED VERSUS CONNECTIONLESS SERVICE

Layers can offer two different types of service to the layers above them: connection-oriented and connectionless.

Connection-oriented service (modeled after the telephone system): to use it, 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 (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. Such connectionless services are often called datagram services.

Quality of service - some services are reliable in the sense that they never lose data. Reliability is usually implemented by having the receiver acknowledge the receipt of each message. The acknowledgment process is often worth but introduces sometimes undesirable overheads and delays.

Reliable connection-oriented service has two minor variation:

- Message sequences - the message boundaries are preserved.
- Byte streams - the connection is simply a stream of bytes, with no message boundaries.

Applications where delays introduced by acknowledgment are unacceptable:

- Digitized voice traffic,
- Video film transmission.

The use of connectionless services:

- Electronic junk mail (third class mail as advertisements) - this service is moreover unreliable (meaning not acknowledged).
- Acknowledged datagram services - connectionless datagram services with acknowledgment.
- Request-reply 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.

NETWORK ARCHITECTURE

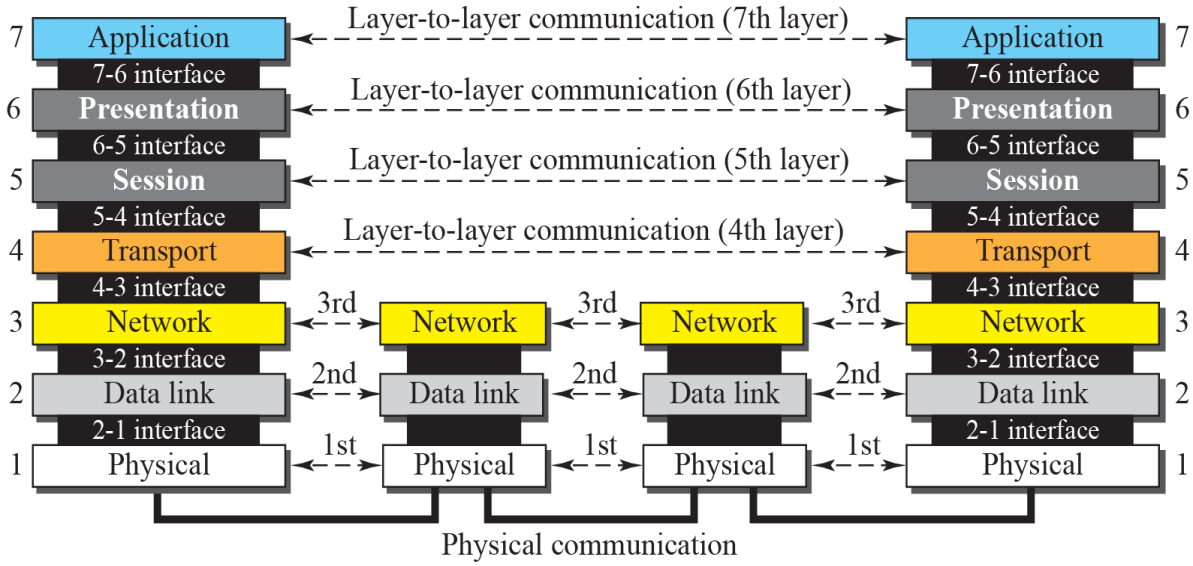
Networks do not remain fixed at single point in time, but it must evolve to accommodate changes based on the technologies on which they are based and demands made by application programmer. Network architecture guides the design and implementation of network. Two commonly used architecture are

- **OSI Architecture**
- **Internet or TCP/IP architecture**

OSI ARCHITECTURE

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



Organization of the layers

The 7 layers can be grouped into 3 subgroups

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

Functions of the Layers

1. PHYSICAL LAYER

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

The physical layer is concerned with the following:

- **Physical characteristics of interfaces and media -** The physical layer defines the characteristics of the interface between the devices and the transmission medium.
- **Representation of bits -** To transmit the stream of bits, it must be encoded to signals. The physical layer defines the type of encoding.
- **Data Rate or Transmission rate -** The number of bits sent each second – is also defined by the physical layer.
- **Synchronization of bits -** The sender and receiver must be synchronized at the bit level. Their clocks must be synchronized.
- **Line Configuration -** In a point-to-point configuration, two devices are connected together through a dedicated link. In a multipoint configuration, a link is shared between several devices.
- **Physical Topology -** The physical topology defines how devices are connected to make a network. Devices can be connected using a mesh, bus, star or ring topology.
- **Transmission Mode -** The physical layer also defines the direction of transmission between two devices: simplex, half-duplex or full-duplex.

2. DATA LINK LAYER

It is responsible for transmitting frames from one node to next node.
The other responsibilities of this layer are

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

3. NETWORK LAYER

This layer is responsible for the delivery of packets from source to destination. It is mainly required, when it is necessary to send information from one network to another.
The other responsibilities of this layer are

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

4. TRANSPORT LAYER

- It is responsible for **Process to Process** delivery.
- It also ensures whether the message arrives in order or not. The other responsibilities of this layer are
- **Port addressing** - The header in this must therefore include a address called port address. This layer gets the entire message to the correct process on that computer.
- **Segmentation and reassembly** - The message is divided into segments and each segment is assigned a sequence number. These numbers are arranged correctly on the arrival side by this layer.
- **Connection control** - This can either be **connectionless or connection-oriented.** The connectionless treats each segment as a individual packet and delivers to the destination. The connection-oriented makes connection on the destination side before the delivery. After the delivery the termination will be terminated.
- **Flow and error control** - Similar to data link layer, but process to process take place.

5. SESSION LAYER

This layer establishes, manages and terminates connections between applications. The other responsibilities of this layer are

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

6. PRESENTATION LAYER

It is concerned with the syntax and semantics of information exchanged between two systems.
The other responsibilities of this layer are

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

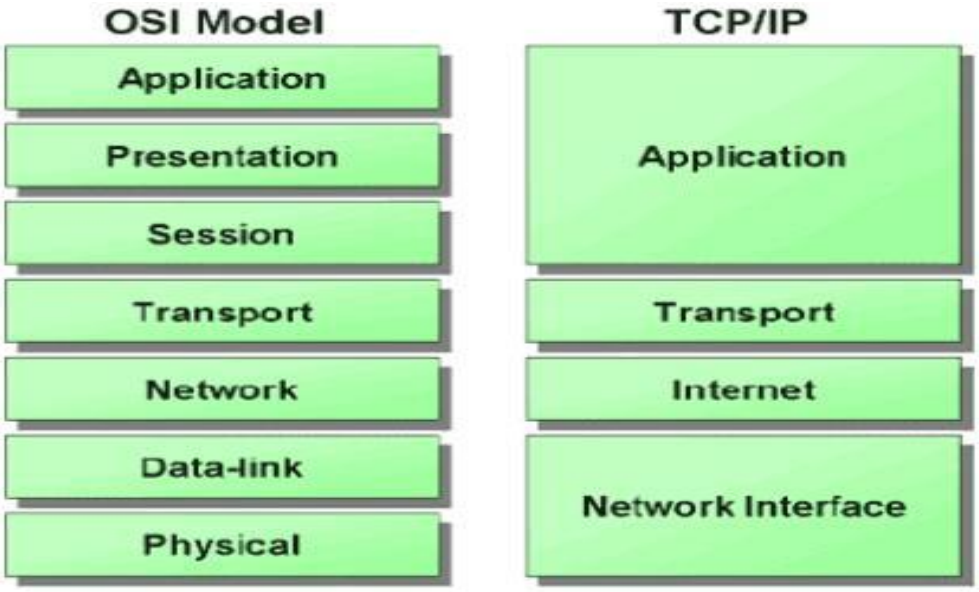
7. APPLICATION LAYER

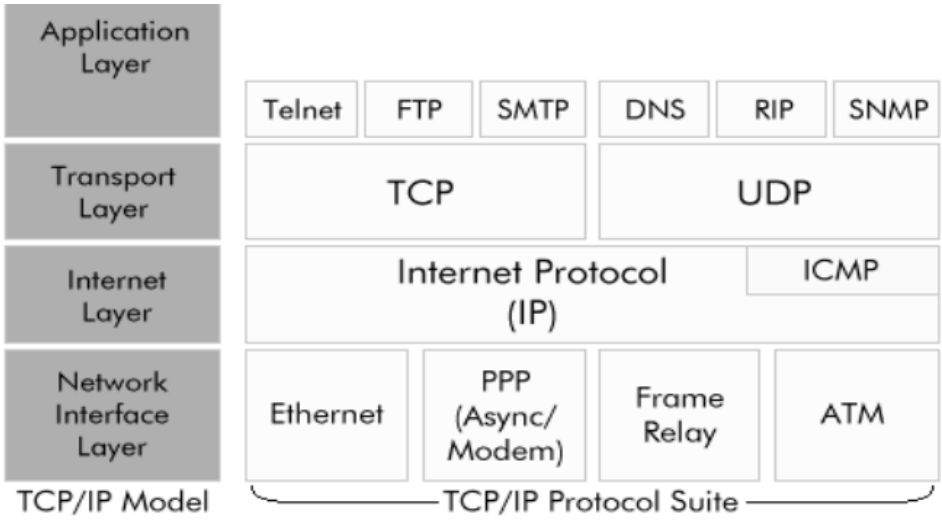
This layer enables the user to access the n/w. This allows the user to log on to remote user. The other responsibilities of this layer are

- **FTAM (file transfer, access, mgmt) -** Allows user to access files in a remote host.
- **Mail services -** Provides email forwarding and storage.
- **Directory services -** Provides database sources to access information about various sources and objects.
- **Network virtual terminal (Remote log-in)**
- **Accessing the World Wide Web**

INTERNET ARCHITECTURE (TCP/IP ARCHITECTURE)

TCP/IP protocols map to a four-layer conceptual model known as the *DARPA model* , named after the U.S. government agency that initially developed TCP/IP. The four layers of the DARPA model are: Application, Transport, Internet, and Network Interface. Each layer in the DARPA model corresponds to one or more layers of the seven-layer Open Systems Interconnection (OSI) model.





1. The Host to Network Layer:

Below the internet layer is great void. The TCP/IP reference model does not really say such about what happen here, except to point out that the host has connect to the network using some protocol so it can transmit IP packets over it. This protocol is not specified and varies from host to host and network to network.

2. Internet layer:

It is a connectionless internetwork layer forming a base for a packet-switching network. Its job is to allow hosts to insert packets into any network and have them to deliver independently to the destination. They may appear in a different order than they were sent in each case it is job of higher layers to rearrange them in order to deliver them to proper destination. TCP/IP internet layer is very similar in functionality to the OSI network layer.

Packet routing is very essential task in order to avoid congestion. For these reason it is say that TCP/IP internet layer perform same function as that of OSI network layer.

The internet layer defines an official packet format and protocol called IP (Internet Protocol) and its provides

- Best-effort delivery
  - No error checking
  - No tracking
- IP is a host-to-host protoco**l.**

3. Transport layer:

In the TCP/IP model, the layer above the internet layer is known as transport layer. It is developed to permit entities on the source and destination hosts to carry on a conversation. It specifies 2 end-to-end protocols

i) TCP (Transmission Control Protocol)
ii) UDP (User Datagram Protocol)

TCP
It is a reliable connection-oriented protocol that permits a byte stream originating on one machine to be transported without error on any machine in the internet. It divides the incoming byte stream into discrete message and passes each one onto the internet layer. At the destination, the receiving TCP process collects the received message into the output stream. TCP deals with flow control to make sure a fast sender cannot swamp a slow receiver with more message than it can handle.

UDP
It is an unreliable, connectionless protocol for applications that do not want TCP's sequencing on flow control and wish to offer their own. It is also used for client-server type request-reply queries and applications in which prompt delivery is more important than accurate delivery such as transmitting speech or video.

4. Application Layer:

In TCP/IP model, session or presentation layer are not present. Application layer is present on the top of the Transport layer. It includes all the higher-level protocols which are virtual terminal (TELNET), file transfer (FTP) and electronic mail (SMTP).
The virtual terminal protocol permits a user on one machine to log into a distant machine and work there. The file transfer protocol offers a way to move data efficiently from one machine to another. Electronic mail was used for file transfer purpose but later a specialized protocol was developed for it.

The Application Layer defines following protocols

i) File Transfer Protocol (FTP)

It was designed to permit reliable transfer of files over different platforms. At the transport layer to ensure reliability, FTP uses TCP. FTP offers simple commands and makes the differences in storage methods across networks transparent to the user. The FTP client is able to interact with any FTP server; therefore the FTP server must also be able to interact with any FTP client. FTP does not offer a user interface, but it does offer an application program interface for file transfer. The client part of the protocol is called as FTP and the server part of the protocol is known as FTPd. The suffix "d" means Daemon this is a legacy from Unix computing where a daemon is a piece of software running on a server that offers a service.

ii) Hyper Text Transfer Protocol

HTTP permits applications such as browsers to upload and download web pages. It makes use of TCP at the transport layer again to check reliability. HTTP is a connectionless protocol that sends a request, receives a response and then disconnects the connection. HTTP delivers HTML

documents plus all of the other components supported within HTML such as JavaScript, Visual script and applets.

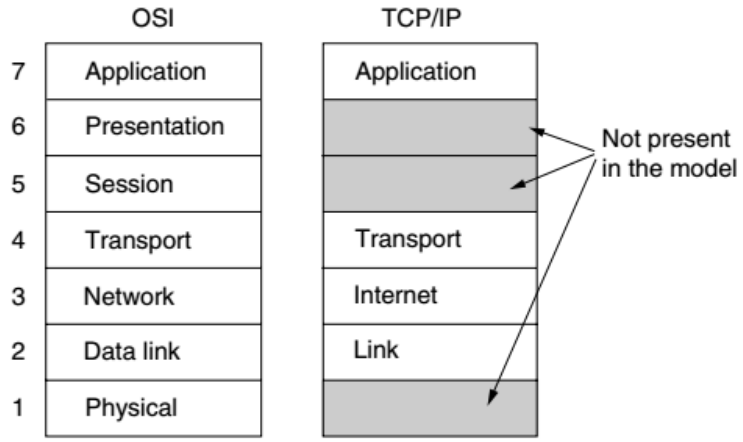
iii) Simple Mail Transfer Protocol

By using TCP, SMTP sends email to other computers that support the TCP/IP protocol suite. SMTP provides extension to the local mail services that existed in the early years of LANs. It supervises the email sending from the local mail host to a remote mail host. It is not reliable for accepting mail from local users or distributing received mail to recipients this is the responsibility of the local mail system. SMTP makes use of TCP to establish a connection to the remote mail host, the mail is sent, any waiting mail is requested and then the connection is disconnected. It can also return a forwarding address if the intended recipient no longer receives email at that destination. To enable mail to be delivered across differing systems, a mail gateway is used.

iv) Simple Network Management Protocol

For the transport of network management information, SNMP is used as standardized protocol. Managed network devices can be cross examined by a computer running to return details about their status and level of activity. Observing software can also trigger alarms if certain performance criteria drop below acceptable restrictions. At the transport layer SNMP protocol uses UDP. The use of UDP results in decreasing network traffic overheads.

Comparison of OSI and TCP/IP reference models



An obvious difference between the two models is the number of layers: the 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.

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.

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.

Three concepts are central to the OSI model:

- Services.
- Interfaces.
- 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. 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. 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.

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.

With TCP/IP the reverse was true: the protocols came first, and the model was really just a description of the existing protocols. There was no problem with the protocols fitting the model. They fit perfectly. The only trouble was that the model did not fit any other protocol stacks.