COMPUTER NETWORK

UNIT-I

Lecture-1

Computer Network: Definitions

[RGPV June 2013]

A computer network or data network is a telecommunications network which allows computers to exchange data. In computer networks, networked computing devices pass data to each other along data connections (network links). Data is transferred in the form of packets. The connections between nodes are established using either cable media or wireless media. The best-known computer network is the Internet.

Network computer devices that originate, route and terminate the data are called network nodes. Nodes can include hosts such as personal computers, phones, servers as well as networking hardware. Two such devices are said to be networked together when one device is able to exchange information with the other device, whether or not they have a direct connection to each other.

Computer networks differ in the transmission media used to carry their signals, the communications protocols to organize network traffic, the network's size, topology and organizational intent. In most cases, communications protocols are layered on (i.e. work using) other more specific or more general communications protocols, except for the *physical layer* that directly deals with the transmission media.

Computer Network: goals

- Resource and load sharing
  - Programs do not need to run on a single machine
- Reduced cost
  - Several machines can share printers, tape drives, etc.
- High reliability
  - If a machine goes down, another can take over
- Mail and communication

Computer Network: components

[RGPV June 2012]

Computer networks share common devices, functions, and features including servers, clients, transmission media, shared data, shared printers and other hardware and software resources, network interface card(NIC), local operating system(LOS), and the network operating system (NOS).

Servers - Servers are computers that hold shared files, programs, and the network operating system. Servers provide access to network resources to all the users of the network. There are many different kinds of servers, and one server can provide several functions. For example, there are file servers, print servers, mail servers, communication servers, database servers, print servers, fax servers and web servers, to name a few.

Clients - Clients are computers that access and use the network and shared network resources. Client computers are basically the customers(users) of the network, as they request and receive services from the servers.

Transmission Media - Transmission media are the facilities used to interconnect computers in a network, such as twisted-pair wire, coaxial cable, and optical fiber cable. Transmission media are sometimes called channels, links or lines.

Shared data - Shared data are data that file servers provide to clients such as data files, printer access programs and e-mail.

Shared printers and other peripherals - Shared printers and peripherals are hardware resources provided to the users of the network by servers. Resources provided include data files, printers, software, or any other items used by clients on the network.

Network Interface Card - Each computer in a network has a special expansion card called a network interface card (NIC). The NIC prepares(formats) and sends data, receives data, and controls data flow between the computer and the network. On the transmit side, the NIC passes frames of data on to the physical layer, which transmits the data to the physical link. On the receiver's side, the NIC processes bits received from the physical layer and processes the message based on its contents.

Local Operating System - A local operating system allows personal computers to access files, print to a local printer, and have and use one or more disk and CD drives that are located on the computer. Examples are MS-DOS, Unix, Linux, Windows 2000, Windows 98, Windows XP etc.

Network Operating System - The network operating system is a program that runs on computers and servers, and allows the computers to communicate over the network.

Hub - Hub is a device that splits a network connection into multiple computers. It is like a distribution center. When a computer request information from a network or a specific computer, it sends the request to the hub through a cable. The hub will receive the request and transmit it to the entire network. Each computer in the network should then figure out whether the broadcast data is for them or not.

Switch - Switch is a telecommunication device grouped as one of computer network components. Switch is like a Hub but built in with advanced features. It uses physical device addresses in each incoming messages so that it can deliver the message to the right destination or port.

Like a hub, switch doesn't broadcast the received message to entire network, rather before sending it checks to which system or port should the message be sent. In other words, switch connects the source and destination directly which increases the speed of the network. Both switch and hub have common features: Multiple RJ-45 ports, power supply and connection lights.

Lecture-2

Computer Network :Architecture,

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

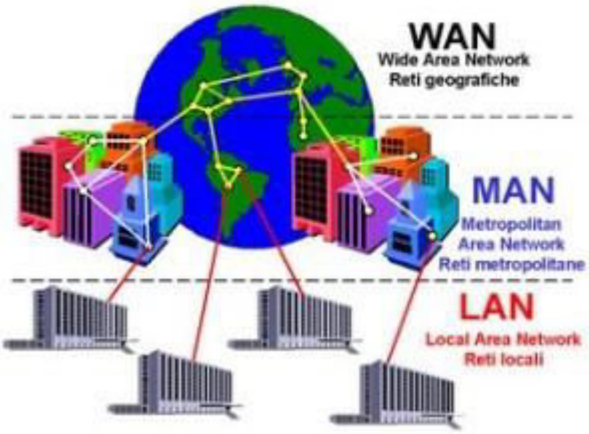
In telecommunication, the specification of a network architecture may also include a detailed description of products and services delivered via a communications network, as well as detailed rate and billing structures under which services are compensated.

The network architecture of the Internet is predominantly expressed by its use of the Internet Protocol Suite, rather than a specific model for interconnecting networks or nodes in the network, or the usage of specific types of hardware links.

Computer Network: Classifications & Types.

There are three types of network classification

1) LAN (Local area network)
2) MAN (Metropolitan Area network)
3) WAN (Wide area network)



1) Local area network (LAN)

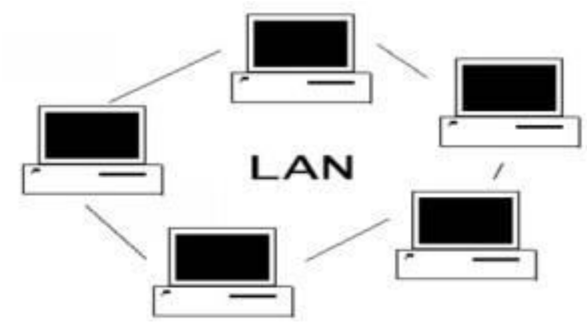
LAN is a group of the computers placed in the same room, same floor, or the same building so they are connected with each other to form a single network to share their resources such as disk drives, data, CPU, modem etc. LAN are limited to some geographical area less than 2 km. Most of LAN is used widely is an Ethernet system of the bus topology.

Characteristics of LAN

LAN connects the computer in a single building, block and they are working in any limited area.

Media access control methods in a LAN, the bus based ehternet, token ring.

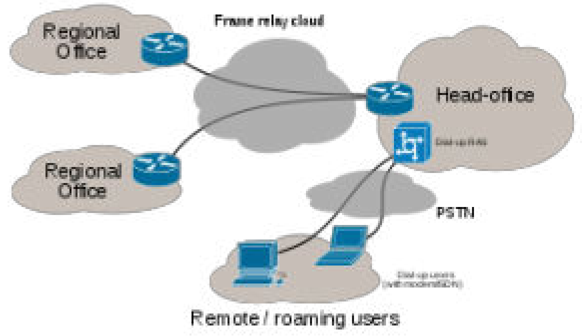
This is private networks, not for subject to tariffs or regulatory controls. LAN is a wireless there is an additional in some countries.



2) Metropolitan Area network (MAN)

The metropolitan area network is a large computer network that expands a Metropolitan area or campus. Its geographic area between a WAN and LAN.its expand round 50km devices used are modem and wire/cable.

Frame-relay network



Characteristics of MAN

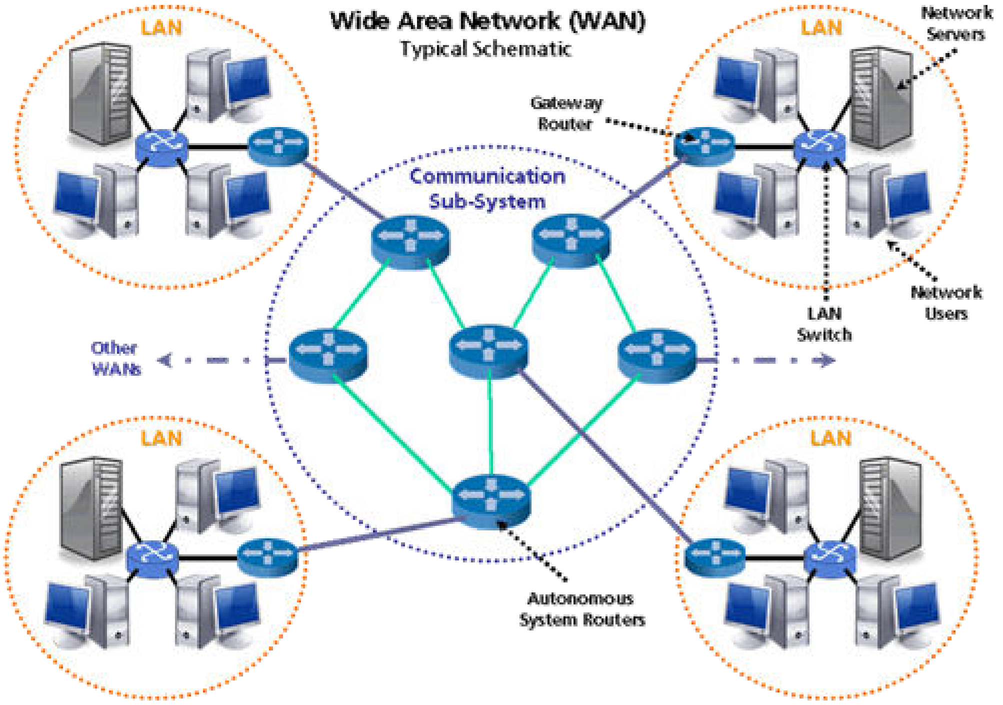
1) Its covers the towns and cities(50km)

2) It is developed in 1980s.

3) MAN is used by the communication medium for optical fiber cables, it also used for other media.

3) Wide area Network (WAN)

The wide area network is a network which connects the countries, cities or the continents, it is a public communications links. The most popular example of a WAN is the internet. WAN is used to connect LAN so the users and the computer in the one location can communicatewith each other.



Characteristics of WAN

1) Its covers the large distances.

2) Communication medium used are satellite, telephones which are connected by the routers.

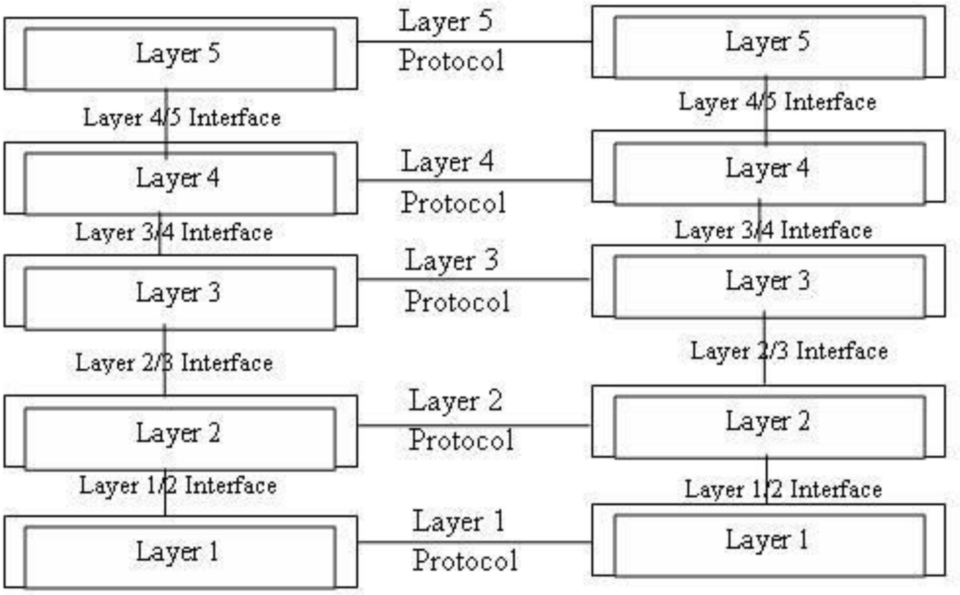
Lecture-3

Layered Architecture: Protocol hierarchy, Design Issues , Interfaces and Services

[RGPV June 2013] , [RGPV June 2012]

To tackle with the design complexity most of the networks are organize as a set of layers or levels. The fundamental idea of layered architecture is to divide the divide the design into small pieces. The layering provides modularity to the network design. The main duty of each layer is to provide offer services to higher layers, and provide abstraction. The main benefits of layered architecture are modularity and clear interfaces.The basic elements of a layered model are services, protocols and Interfaces.

A service is a set of functions that a layer offers to another layer (usually to upper layer)We know that protocol is a set of rules. Here the protocols are used to exchange information with a peer layer. Peers means layers at same level. The protocol consist several rules that deals with the content and the order or structure of the messages exchanged. All the data from one layer to another either upper or lower layer pass through the corresponding interfaces. Suppose we have an n layered network then, layer-n of one machine take conversation with layer-n on another machine. Here layer-n protocol define the rules and orders.



Five Layered Network

Why layered architecture is preferred in computer network design?

Layered architectures have several advantages. Some of them are,

- Modularity and clear interface
- Provide flexibility to modify network services

- Ensure independence of layers
- Management of network architecture is easy
- Each layer can be analyzed and tested independent of other layers

The benefits to layering networking protocol specifications are many including: **Interoperability** - Layering promotes greater interoperability between devices from different manufacturers and even between different generations of the same type of device from the same manufacturer. **Greater Compatibility** - One of the greatest of all of the benefits of using a hierarchal or layered approach to networking and communications protocols is the greater compatibility between devices, systems and networks that this delivers. **Better Flexibility** - Layering and the greater compatibility that it delivers goes a long way to improving the flexibility; particularly in terms of options and choices, that network engineers and administrators alike crave so much. **Flexibility and Peace of Mind** - Peace of mind in knowing that if worst comes to worst and a key core network device; suddenly and without prior warning decides to give up the ghost, you can rest assured that a replacement or temporary stand-by can be readily put to work with the highest degree of confidence that it will do the job. Even though it may not be up to doing the job at the same speed it will still do it; at least, until a better, more permanent solution can be implemented. This is a state of affairs that is much more acceptable than for a lengthy cessation of network services or assets unavailability to occur. 80% is oh so much more pleasing than 0%. **Increased Life Expectancy** - Increased product working life expectancies as backwards compatibility is made considerably easier. Devices from different technology generations can co-exist thus the older units do not get discarded immediately newer technologies are adopted. **Scalability** - Experience has shown that a layered or hierarchal approach to networking protocol design and implementation scales better than the horizontal approach. **Mobility** - Greater mobility is more readily delivered whenever we adopt the layered and segmented strategies into our architectural design **Value Added Features** - It is far easier to incorporate and implement value added features into products or services when the entire system has been built on the use of a layered philosophy. **Cost Effective Quality** - The layered approach has proven time and time again to be the most economical way of developing and implementing any system(s) be they small, simple, large or complex makes no difference. This ease of development and implementation translates to greater efficiency and effectiveness which in turn translates into greater economic rationalization and cheaper products while not compromising quality. **Modularity** - I am sure that you have come across plug-ins and add-ons. These are common and classical examples of the benefits to be derived from the use of a hierarchal (layered) approach to design. **Innate Plasticity** - Layering allows for innate plasticity to be built into devices at all levels and stages from the get-go, to implementation, on through optimization and upgrade cycles throughout a component's entire useful working lifecycle thereafter. **The Graduated, Blended Approach to Migration** - Compatibility enables technologies to co-exist side-by-side which results in quicker uptake of newer technologies as the older asset investments can still continue to be productive. Thus migration to newer technologies and standards can be undertaken in stages or phases over a period of time. This is what is known as the graduated blended approach; which is the opposite of the sudden adoption approach. **Standardization and Certification** - The layered approach to networking protocol specifications facilitates a more streamlined and simplified standardization and certification process; particularly from an "industry" point of view. This is due to the clearer and more distinct definition and demarcation of what functions occur at each layer when the layered approach is taken. **Task Segmentation** - Breaking a large complex system into smaller more manageable subcomponents allows for easier development and implementation of new technologies; as well as facilitating human comprehension of what may be very diverse and complex systems. **Portability** - Layered networking protocols are much easier to port from one system or architecture to another. **Compartmentalization of Functionality** - The compartmentalization or layering of processes, procedures and communications functions gives developers the freedom to concentrate on a specific layer or specific functions within that layer's realm of responsibility without the need for great concern or modification of any other layer. Changes within one layer can be considered to be in self-contained isolation; functionally speaking, from the other layers. Modifications at one layer will not break or compound the other layers. **Side-Kicks** - The development of "Helper" protocols or side-kicks is much easier when a layered approach to networking protocols is embraced. This is especially so when it comes to the development of "helper" protocols that are developed more or less as after-thoughts because the need arose. **Reduced Debugging Time** - The time spent debugging can be greatly reduced as a direct result of taking the layered approach to developing network protocols because debugging is made easier and faster when using the layered approach as opposed to not using it. **Promotion of Multi-Vendor Development** - Layering allows for a more precise identification and delineation of task, process and methodology. This permits a clearer definition of what needs to be done, where it needs to be done, when it needs to be done, how it needs to be done and what or who will do it. It is these factors that promote multi-vendor development through the standardization of networking components at both the hardware and software levels because of the clear and precise delineation of responsibilities that layering brings to the developers' table. **Easier Binding Implementation** - The principle of binding is far easier to implement in layered, tiered, and hierarchal systems. Humans also tend to understand this form easier than the flat model. **Enhanced Troubleshooting and Fault Identification** - Troubleshooting and fault identification are made considerably easier thus resolution times are greatly reduced. Layering allows for examination in isolation of subcomponents as well as the whole. **Enhanced Communications Flow and Support** - Adopting the layered approach allows for improved flow and support for communication between diverse systems, networks, hardware, software, and protocols. **Support for Disparate Hosts** - Communications between disparate hosts is supported more or less seamlessly thus Unix, PC, MAC & Linux to name but a few can freely interchange data. **Reduction of the Domino Effect** - Another very important advantage of a layered protocol system is that it helps to prevent changes in one layer from affecting other layers. This helps to expedite technology development. **Rapid Application Development (RAD)** - Work loads can be evenly distributed which means that multiple activities can be conducted in parallel thereby reducing the time taken to develop, debug, optimize and package new technologies ready for production implementation.

Lecture-4

Connection Oriented & Connectionless Services, Service primitives, Design issues & its functionality

[RGPV June 2014], [RGPV June 2012]

Connection-oriented (CO-mode[) communication is a network communication mode in telecommunications and computer networking, where a communication session or a semi-permanent connection is established before any useful data can be transferred, and where a stream of data is delivered in the same order as it was sent. The alternative to connection-oriented transmission is connectionless communication, for example the datagram mode communication used by the IP and UDP protocols, where data may be delivered out of order, since different packets are routed independently, and may be delivered over different paths.

Connection-oriented communication may be a circuit switched connection, or a packet-mode virtual circuit connection. In the latter case, it may use either a transport layer virtual circuit protocol such as the TCP protocol, allowing data to be delivered in order although the lower layer switching is connectionless, or it may be a data link layer or network layer switching mode, where all data packets belonging to the same traffic stream are delivered over the same path, and traffic flows are identified by some *connection identifier* rather than by complete routing information, allowing fast hardware based switching.

Connection-oriented protocol services are often but not always *reliable* network services, that provide acknowledgment after successful delivery, and automatic repeat request functions in case of missing data or detected bit-errors. ATM, Frame Relay and MPLS are examples of a connection-oriented, unreliable protocol.

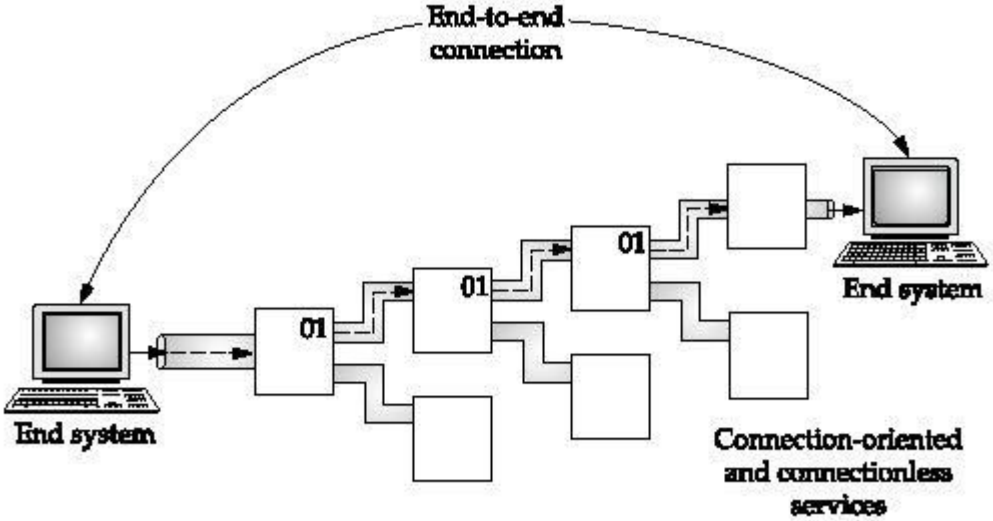
A Connection-Oriented Protocol (COP) is a networking protocol used to establish a data communication session in which endpoint devices use preliminary protocols to establish end-to-end connections and then the subsequent data stream is delivered in sequential transfer mode.

COPs guarantee sequential data delivery but are classed as an unreliable network service because there is no process to ensure that total data received is the same as what was sent. COPs provide circuit-switched connections or virtual circuit connections in packet-switched networks (PSN).

Two distinct techniques are used in data communications to transfer data. Each has its own advantages and disadvantages. They are the connection-oriented method and the connectionless method:

- **Connection-oriented** Requires a session connection (analogous to a phone call) be established before any data can be sent. This method is often called a "reliable" network service. It can guarantee that data will arrive in the same order. Connection-oriented services set up virtual links between end systems through a network, as shown in Figure 1. Note that the packet on the left is assigned the virtual circuit number 01. As it moves through the network, routers quickly send it through virtual circuit 01.

- **Connectionless** Does not require a session connection between sender and receiver. The sender simply starts sending packets (called datagrams) to the destination. This service does not have the reliability of the connection-oriented method, but it is useful for periodic burst transfers. Neither system must maintain state information for the systems that they send transmission to or receive transmission from. A connectionless network provides minimal services.



Connection-oriented methods may be implemented in the data link layers of the protocol stack and/or in the transport layers of the protocol stack, depending on the physical connections in place and the services required by the systems that are communicating. TCP (Transmission Control Protocol) is a connection-oriented transport protocol, while UDP (User Datagram Protocol) is a connectionless network protocol. Both operate over IP.The physical, data link, and network layer protocols have been used to implement guaranteed data delivery. For example, X.25 packet-switching networks perform extensive error checking and packet acknowledgment because the services were originally implemented on poor-quality telephone connections. Today, networks are more reliable. It is generally believed that the underlying network should do what it does best, which is deliver data bits as quickly as possible. Therefore, connection-oriented services are now primarily handled in the transport layer by end systems, not the network. This allows lower-layer networks to be optimized for speed.

LANs operate as connectionless systems. A computer attached to a network can start transmitting frames as soon as it has access to the network. It does not need to set up a connection with the destination system ahead of time. However, a transport-level protocol such as TCP may set up a connection-oriented session when necessary.

The Internet is one big connectionless packet network in which all packet deliveries are handled by IP. However, TCP adds connection-oriented services on top of IP. TCP provides all the upper-level connection-oriented session requirements to ensure that data is delivered properly. MPLS is a relatively new connection-oriented networking scheme for IP networks that sets up fast label-switched paths across routed or layer 2 networks.A WAN service that uses the connection-oriented model is frame relay.

Lecture-5

ISO-OSI Reference Model: Principle, Model, Descriptions of various layers

[RGPV June 2014]

The **Open Systems Interconnection model** (**OSI**) is a conceptual model that characterizes and standardizes the internal functions of a communication system by partitioning it into abstraction layers. The model is a product of the Open Systems Interconnection project at the International Organization for Standardization (ISO), maintained by the identification ISO/IEC 7498-1.The model groups communication functions into seven logical layers. A layer serves the layer above it and is served by the layer below it. For example, a layer that provides error-free communications across a network provides the path needed by applications above it, while it calls the next lower layer to send and receive packets that make up the contents of that path. Two instances at one layer are connected by a horizontal connection on that layer.The recommendation X.200 describes seven layers, labeled 1 to 7. Layer 1 is the lowest layer in this model.

OSI Model

	Layer	**Data unit**	**Function**	**Examples**
Host layers	7. Application	Data	High-level APIs, including resource sharing, remote file access, directory services and virtual terminals	HTTP, FTP, SMTP
	6. Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption	ASCII, EBCDIC, JPEG
	5. Session		Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes	RPC, PAP
	4. Transport	Segments	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing	TCP, UDP
Media layers	3. Network	Packet/Datagram	Structuring and managing a multi-node network, including addressing, routing and traffic control	IPv4, IPv6, IPsec,

			AppleTalk
2. Data link	Bit/Frame	Reliable transmission of data frames between two nodes connected by a physical layer	PPP, IEEE 802.2, L2TP
1. Physical	Bit	Transmission and reception of raw bit streams over a physical medium	DSL, USB

At each level *N* two entities at the communicating devices (layer N *peers*) ex

Layer 1: physical layer

The physical layer has the following major functions:

- It defines the electrical and physical specifications of the data connection. It defines the relationship between a device and a physical transmission medium (e.g., a copper or fiber optical cable). This includes the layout of pins, voltages, line impedance, cable specifications, signal timing, hubs, repeaters, network adapters, host bus adapters (HBA used in storage area networks) and more.
- It defines the protocol to establish and terminate a connection between two directly connected nodes over a communications medium.
- It may define the protocol for flow control.
- It defines transmission mode i.e. simplex, half duplex, full duplex.
- It defines topology.
- It defines a protocol for the provision of a (not necessarily reliable) connection between two directly connected nodes, and the modulation or conversion between the representation of digital data in user equipment and the corresponding signals transmitted over the physical communications channel. This channel can involve physical cabling (such as copper and optical fiber) or a wireless radio link.

The physical layer of Parallel SCSI operates in this layer, as do the physical layers of Ethernet and other local-area networks, such as Token Ring, FDDI, ITU-T G.hn, and IEEE 802.11 (Wi-Fi), as well as personal area networks such as Bluetooth and IEEE 802.15.4.

Layer 2: data link layer

The data link layer provides node-to-node data transfer -- a reliable link between two directly connected nodes, by detecting and possibly correcting errors that may occur in the physical layer. The data link layer is divided into two sublayers:

- Media Access Control (MAC) layer - responsible for controlling how devices in a network gain access to data and permission to transmit it.
- Logical Link Control (LLC) layer - controls error checking and packet synchronization.

The Point-to-Point Protocol (PPP) is an example of a data link layer in the TCP/IP protocol stack.

The ITU-T G.hn standard, which provides high-speed local area networking over existing wires (power lines, phone lines and coaxial cables), includes a complete data link layer that provides both error correction and flow control by means of a selective-repeat sliding-window protocol.

Layer 3: network layer

The network layer provides the functional and procedural means of transferring variable length data sequences (called datagrams) from one node to another connected to the same *network*. It translates logical network address into physical machine address. A network is a medium to which many nodes can be connected, on which every node has an *address* and which permits nodes connected to it to transfer messages to other nodes connected to it by merely providing the content of a message and the address of the destination node and letting the network find the way to deliver ("route") the message to the destination node. In addition to message routing, the network may (or may not) implement message delivery by splitting the message into several fragments, delivering each fragment by a separate route and reassembling the fragments, report delivery errors, etc.Datagram delivery at the network layer is *not* guaranteed to be *reliable*.

A number of layer-management protocols, a function defined in the *management annex*, ISO 7498/4, belong to the network layer. These include routing protocols, multicast group management, network-layer information and error, and network-layer address assignment. It is the function of the payload that makes these belong to the network layer, not the protocol that carries them.

Layer 4: transport layer

The transport layer provides the functional and procedural means of transferring variable-length data sequences from a source to a destination host via one or more networks, while maintaining the quality of service functions.

An example of a transport-layer protocol in the standard Internet stack is Transmission Control Protocol (TCP), usually built on top of the Internet Protocol (IP).

The transport layer controls the reliability of a given link through flow control, segmentation/desegmentation, and error control. Some protocols are state- and connection-oriented. This means that the transport layer can keep track of the segments and retransmit those that fail. The transport layer also provides the acknowledgement of the successful data transmission and sends the next data if no errors occurred. The transport layer creates packets out of the message received from the application layer. Packetizing is a process of dividing the long message into smaller messages.

OSI defines five classes of connection-mode transport protocols ranging from class 0 (which is also known as TP0 and provides the fewest features) to class 4 (TP4, designed for less reliable networks, similar to the Internet). Class 0 contains no error recovery, and was designed for use on network layers that provide error-free connections. Class 4 is closest to TCP, although TCP contains functions, such as the graceful close, which OSI assigns to the session layer. Also, all OSI TP connection-mode protocol classes provide expedited data and preservation of record boundaries.

An easy way to visualize the transport layer is to compare it with a post office, which deals with the dispatch and classification of mail and parcels sent. Do remember, however, that a post office manages the outer envelope of mail. Higher layers may have the equivalent of double envelopes, such as cryptographic presentation services that can be read by the addressee only. Roughly speaking, tunneling protocols operate at the transport layer, such as carrying non-IP protocols such as IBM's SNA or Novell's IPX over an IP network, or end-to-end encryption with IPsec. While Generic Routing Encapsulation (GRE) might seem to be a network-layer protocol, if the encapsulation of the payload takes place only at endpoint, GRE becomes closer to a transport protocol that uses IP headers but contains complete frames or packets to deliver to an endpoint. L2TP carries PPP frames inside transport packet.

Although not developed under the OSI Reference Model and not strictly conforming to the OSI definition of the transport layer, the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) of the Internet Protocol Suite are commonly categorized as layer-4 protocols within OSI.

Layer 5: session layer

The session layer controls the dialogues (connections) between computers. It establishes, manages and terminates the connections between the local and remote application. It provides for full-duplex, half-duplex, or simplex operation, and establishes checkpointing, adjournment, termination, and restart procedures. The OSI model made this layer responsible for graceful close of sessions, which is a property of the Transmission Control Protocol, and also for session checkpointing and recovery, which is not usually used in the Internet Protocol Suite. The session layer is commonly implemented explicitly in application environments that use remote procedure calls.

Layer 6: presentation layer

The presentation layer establishes context between application-layer entities, in which the application-layer entities may use different syntax and semantics if the presentation service provides a big mapping between them. If a mapping is available, presentation service data units are encapsulated into session protocol data units, and passed down the protocol stack.

This layer provides independence from data representation (e.g., encryption) by translating between application and network formats. The presentation layer transforms data into the form that the application accepts. This layer formats and encrypts data to be sent across a network. It is sometimes called the syntax layer.

The original presentation structure used the Basic Encoding Rules of Abstract Syntax Notation One (ASN.1), with capabilities such as converting an EBCDIC-coded text file to an ASCII-coded file, or serialization of objects and other data structures from and to XML.

Layer 7: application layer

The application layer is the OSI layer closest to the end user, which means both the OSI application layer and the user interact directly with the software application. This layer interacts with software applications that implement a communicating component. Such application programs fall outside the scope of the OSI model. Application-layer functions typically include identifying communication partners, determining resource availability, and synchronizing communication. When identifying communication partners, the application layer determines the identity and availability of communication partners for an application with data to transmit. When determining resource availability, the application layer must decide whether sufficient network or the requested communication exists. In synchronizing communication, all communication between applications requires cooperation that is managed by the application layer. Some examples of application-layer implementations include:

- On OSI stack:
    - FTAM File Transfer and Access Management Protocol
    - X.400 Mail
    - Common Management Information Protocol (CMIP)
- On TCP/IP stack:
    - Hypertext Transfer Protocol (HTTP),
    - File Transfer Protocol (FTP),
    - Simple Mail Transfer Protocol (SMTP),
    - Simple Network Management Protocol (SNMP), etc.

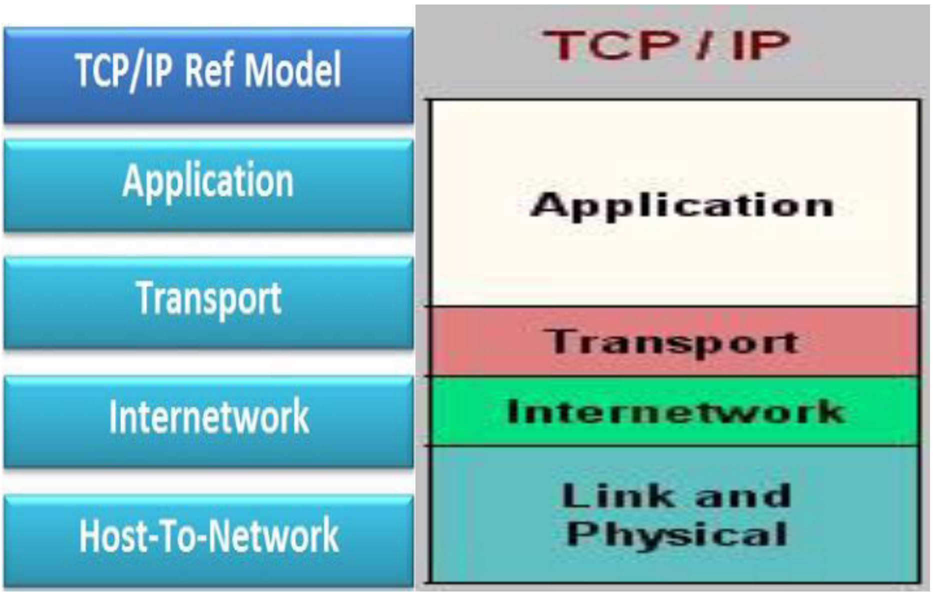
Lecture-6

TCP/IP

[RGPV June 2014], [RGPV June 2013]

The TCP/IP reference model is the network model used in the current Internet architecture. It is considered as the grandfather of the Internet the ARPANET. The reference model was named after two of its main protocols, TCP (Transmission control Protocol) and IP(Internet Protocol).

There are versions of this model with four layers and with five layers. The original four-layer version of the model is shown below.



Layer 4: Process Layer or Application Layer: This is where the "higher level" protocols such as FTP, HTTP, etc. operate. The original TCP/IP specification described a number of different applications that fit into the top layer of the protocol stack. These applications include Telnet, FTP, SMTP and DNS.

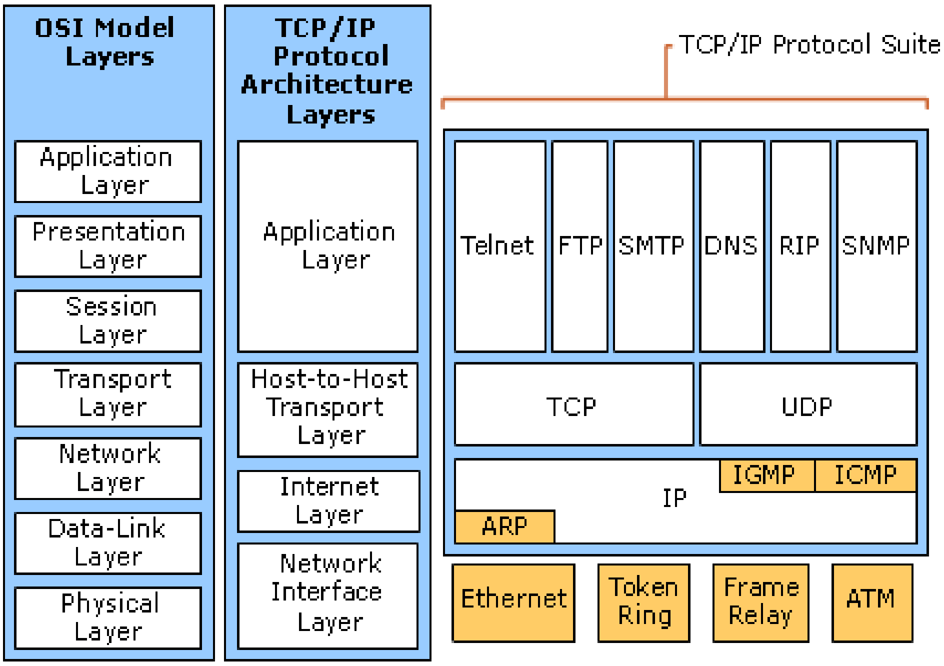
Layer 3: Host-To-Host (Transport) Layer: This is where flow-control and connection protocols exist,, such as TCP. This layer deals with opening and maintaining connection, ensuring that packet are in fact received the transport layer is the interface between the application layer and the complex hardware of the. Two modes are available, full-duplex and half duplex. In full-duplex operation, both sides can transmit and receive data simultaneously, whereas in half duplex, a side can only send or receive at one time.

Layer 2: Internet or Internetworking Layer: This layer defines IP addresses, with many routing schemes for navigating packets from one IP address to another. The job of the network layer is to inject packets into any network and have them travel independently to the destination. Packet routing is a major job of this protocol.

Layer 1: Networking Access Layer: This layer describes the physical equipment necessary for communications, such as twisted pair cables, the signalling used on that equipment, and the low-level protocols using that signalling. That Host-to-Network layer interfaces the

TCP/IP protocol stack to the physical network.

TCP/IP Protocol Suite:



The TCP/IP protocol suite has two sets of protocols at the Internet layer:

- IPv4, also known as IP, is the Internet layer in common use today on private intranets and the Internet.
- IPv6 is the new Internet layer that will eventually replace the existing IPv4 Internet layer.

Lecture-7

Queueing Models: Little's Theorem, Queueing System -1

[RGPV June 2014]

Queueing theory is the mathematical study of waiting lines, or queues. In queueing theory a model is constructed so that queue lengths and waiting times can be predicted. Queueing theory is generally considered a branch of operations research because the results are often used when making business decisions about the resources needed to provide a service.

Queueing theory has its origins in research by Agner Krarup Erlang when he created models to describe the Copenhagen telephone exchange. The ideas have since seen applications including telecommunication, traffic engineering, computing and the design of factories, shops, offices and hospitals.

Single queueing nodes

Single queueing nodes are usually described using Kendall's notation in the form *A*/*S*/*C* where *A* describes the time between arrivals to the queue, *S* the size of jobs and *C* the number of servers at the node. Many theorems in queue theory can be proved by reducing queues to mathematical systems known as Markov chains, first described by Andrey Markov in his 1906 paper.

Agner Krarup Erlang, a Danish engineer who worked for the Copenhagen Telephone Exchange, published the first paper on what would now be called queueing theory in 1909. He modeled the number of telephone calls arriving at an exchange by a Poisson process and solved the M/D/1 queue in 1917 and M/D/k queueing model in 1920. In Kendall's notation

- M stands for Markov or memoryless and means arrivals occur according to a Poisson process
- D stands for deterministic and means jobs arriving at the queue require a fixed amount of service
- *k* describes the number of servers at the queueing node (*k* = 1, 2,...). If there are more jobs at the node than there are servers then jobs will queue and wait for service.

The M/M/1 queue is a simple model where a single server serves jobs that arrive according to a Poisson process and have exponentially distributed service requirements. In an M/G/1 queue the G stands for general and indicates an arbitrary probability distribution. The M/G/1 model was solved by Felix Pollaczek in 1930, a solution later recast in probabilistic terms by Aleksandr Khinchin and now known as the Pollaczek–Khinchine formula.

After World War II queueing theory became an area of research interest to mathematicians. Work on queueing theory used in modern packet switching networks was performed in the early 1960s by Leonard Kleinrock. It was in this period that John Little gave a proof of the formula which now bears his name: Little's law. In 1961 John Kingman gave a formula for the mean waiting time in a G/G/1 queue: Kingman's formula

The matrix geometric method and matrix analytic methods have allowed queues with phase-type distributed interarrival and service time distributions to be considered

Problems such as performance metrics for the M/G/k queue remain an open problem

Queueing networks

Networks of queues are systems in which a number of queues are connected by customer routing. When a customer is serviced at one node it can join another node and queue for service, or leave the network. For a network of *m* the state of the system can be described by an *m*–dimensional vector $(x_1,x_2,...,x_m)$ where x_i represents the number of customers at each node. The first significant results in this area were Jackson networks,[20][21] for which an efficient product-form stationary distribution exists and the mean value analysiswhich allows average metrics such as throughput and sojourn times to be computed.

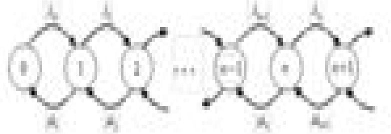
If the total number of customers in the network remains constant the network is called a closed network and has also been shown to have a product–form stationary distribution in the Gordon–Newell theorem.[24] This result was extended to the BCMP networkwhere a network with very general service time, regimes and customer routing is shown to also exhibit a product-form stationary distribution.

Networks of customers have also been investigated, Kelly networks where customers of different classes experience different priority levels at different service nodes.

Another type of network are G-networks first proposed by Erol Gelenbe in 1993:these networks do not assume exponential time distributions like the classic Jackson Network

Example of M/M/1
Birth and Death process

- A/B/C



A:distribution of arrival time
B:distribution of service time
C:the number of parallel servers

Lecture-8

Queueing Models: Little's Theorem, Queueing System -2

We have seen that as a system gets congested, the service delay in the system increases. A good understanding of the relationship between congestion and delay is essential for designing effective congestion control algorithms. Queuing Theory provides all the tools needed for this analysis. This article will focus on understanding the basics of this topic.

Communication Delays

Before we proceed further, lets understand the different components of delay in a messaging system. The total delay experienced by messages can be classified into the following categories:

Processing Delay

- This is the delay between the time of receipt of a packet for transmission to the point of putting it into the transmission queue.
- On the receive end, it is the delay between the time of reception of a packet in the receive queue to the point of actual processing of the message.
- This delay depends on the CPU speed and CPU load in the system.

Queuing Delay

- This is the delay between the point of entry of a packet in the transmit queue to the actual point of transmission of the message.
- This delay depends on the load on the communication link.

Transmission Delay

- This is the delay between the transmission of first bit of the packet to the transmission of the last bit.
- This delay depends on the speed of the communication link.

Propagation Delay

- This is the delay between the point of transmission of the last bit of the packet to the point of reception of last bit of the packet at the other end.
- This delay depends on the physical characteristics of the communication link.

Retransmission Delay

- This is the delay that results when a packet is lost and has to be retransmitted.
- This delay depends on the error rate on the link and the protocol used for retransmissions.

In this article we will be dealing primarily with queueing delay.

Little's Theorem

We begin our analysis of queueing systems by understanding Little's Theorem. Little's theorem states that:

The average number of customers (N) can be determined from the following equation:

$N = \lambda T$

Here lambda is the average customer arrival rate and T is the average service time for a customer.

Proof of this theorem can be obtained from any standard textbook on queueing theory. Here we will focus on an intuitive understanding of the result. Consider the example of a restaurant where the customer arrival rate (lambda) doubles but the customers still spend the same amount of time in the restaurant (T). This will double the number of customers in the restaurant (N). By the same logic if the customer arrival rate remains the same but the customers service time doubles, this will also double the total number of customers in the restaurant.

Queueing System Classification

With Little's Theorem, we have developed some basic understanding of a queueing system. To further our understanding we will have to dig deeper into characteristics of a queueing system that impact its performance. For example, queueing requirements of a restaurant will depend upon factors like:

- How do customers arrive in the restaurant? Are customer arrivals more during lunch and dinner time (a regular restaurant)? Or is the customer traffic more uniformly distributed (a cafe)?
- How much time do customers spend in the restaurant? Do customers typically leave the restaurant in a fixed amount of time? Does the customer service time vary with the type of customer?
- How many tables does the restaurant have for servicing customers?

The above three points correspond to the most important characteristics of a queueing system. They are explained below:

Arrival Process

- The probability density distribution that determines the customer arrivals in the system.
- In a messaging system, this refers to the message arrival probability distribution.

Service Process

- The probability density distribution that determines the customer service times in the system.
- In a messaging system, this refers to the message transmission time distribution. Since message transmission is directly proportional to the length of the message, this parameter indirectly refers to the message length distribution.

Number of Servers

- Number of servers available to service the customers.
- In a messaging system, this refers to the number of links between the source and destination nodes.

Based on the above characteristics, queueing systems can be classified by the following convention:

A/S/n

Where A is the arrival process, S is the service process and n is the number of servers. A and S are can be any of the following:

M (Markov)	Exponential probability density
D (Deterministic)	All customers have the same value
G (General)	Any arbitrary probability distribution

Examples of queueing systems that can be defined with this convention are:

- **M/M/1:** This is the simplest queueing system to analyze. Here the arrival and service time are negative exponentially distributed (poisson process). The system consists of only one server. This queueing system can be applied to a wide variety of problems as any system with a very large number of independent customers can be approximated as a Poisson process. Using a Poisson process for service time however is not applicable in many applications and is only a crude approximation. Refer to M/M/1 Queueing System for details.
- **M/D/n:** Here the arrival process is poisson and the service time distribution is deterministic. The system has n servers. (e.g. a ticket booking counter with n cashiers.) Here the service time can be assumed to be same for all customers)
- **G/G/n:** This is the most general queueing system where the arrival and service time processes are both arbitrary. The system has n servers. No analytical solution is known for this queueing system.

Lecture-9

Queueing Models: Little's Theorem, Queueing System -3

M/M/c queue

In queueing theory, a discipline within the mathematical theory of probability, the M/M/c queue (or Erlang–C modelis a multi-server queueing model. In Kendall's notation it describes a system where arrivals form a single queue and are governed by a Poisson process, there are *c* servers and job service times are exponentially distributed. It is a generalisation of the M/M/1 queue which considers only a single server. The model with infinitely many servers is the M/M/∞ queue.

M/M/1 queue

In queueing theory, a discipline within the mathematical theory of probability, an M/M/1 queue represents the queue length in a system having a single server, where arrivals are determined by a Poisson process and job service times have an exponential distribution. The model name is written in Kendall's notation. The model is the most elementary of queueing models and an attractive object of study as closed-form expressions can be obtained for many metrics of interest in this model. An extension of this model with more than one server is the M/M/c queue.

M/M/∞

In queueing theory, a discipline within the mathematical theory of probability, the **M/M/∞ queue** is a multi-server queueing model where every arrival experiences immediate service and does not wait In Kendall's notation it describes a system where arrivals are governed by a Poisson process, there are infinitely many servers, so jobs do not need to wait for a server. Each job has an exponentially distributed service time. It is a limit of the M/M/c queue model where the number of servers *c* becomes very large.The model can be used to model bound lazy deletion performance.

M/G/1

In queueing theory, a discipline within the mathematical theory of probability, an **M/G/1 queue** is a queue model where arrivals are **M**arkovian (modulated by a Poisson process), service times have a **G**eneral distribution and there is a single server.[1] The model name is written in Kendall's notation, and is an extension of the M/M/1 queue, where service times must be exponentially distributed. The classic application of the M/G/1 queue is to model performance of a fixed head hard disk.

Model definition

A queue represented by a M/G/1 queue is a stochastic process whose state space is the set {0,1,2,3...}, where the value corresponds to the number of customers in the queue, including any being served. Transitions from state i to $i + 1$ represent the arrival of a new customer: the times between such arrivals have an exponential distribution with parameter λ. Transitions from state i to $i − 1$ represent a customer who has been served, finishing being served and departing: the length of time required for serving an individual customer has a general distribution function. The lengths of times between arrivals and of service periods are random variables which are assumed to be statistically independent.

Scheduling policies

Customers are typically served on a first-come, first-served basis, other popular scheduling policies include

- processor sharing where all jobs in the queue share the service capacity between them equally
- last-come, first served without preemption where a job in service cannot be interrupted
- last-come, first served with preemption where a job in service is interrupted by later arrivals, but work is conserved
- generalized foreground-background (FB) scheduling also known as least-attained-service where the jobs which have received least processing time so far are served first and jobs which have received equal service time share service capacity using processor sharing
- shortest job first without preemption (SJF) where the job with the smallest size receives service and cannot be interrupted until service completes
- preemptive shortest job first where at any moment in time the job with the smallest original size is served
- shortest remaining processing time (SRPT) where the next job to serve is that with the smallest remaining processing requirement

Service policies are often evaluated by comparing mean sojourn times in the queue. If service times that jobs require are known on arrival then the optimal scheduling policy is SRPT.Policies can also be evaluated using a measure of fairness

RGPV PAPER QUESTIONS

Q.1 Explain ISO-OSI Model?

Q.2 Explain Queueing Modelsand also Little's Theorem

Q.3 Explain TCP/IP in brief?

Q.4 Explain working of Connection Oriented & Connectionless Services ?

Q.5 What is Computer Network what are the advantages of Computer Network?

Q.6 What is the reason of being using Layered Architecture?

Q.7 Explain in detail TCP/IP model ?

Q.8 What are the components of Computer Network?

Q.9 Describe design issues of layers?

Q.10 Explain Connection Oriented & Connectionless Services