

UNIT I

INTRODUCTION AND PHYSICAL LAYER

Communication - Networks – Network Types – Protocol Layering – TCP/IP Protocol suite – OSI Model – Introduction to Sockets - Application Layer protocols: HTTP – FTP – Email protocols (SMTP - POP3 - IMAP - MIME) – DNS – SNMP

1. INTRODUCTION TO NETWORKS

* Write the parameters used to measure the network performance. (May/June 2016)
(2)

* List the metrics that influence the performance of computer networks.
(Apr/May 2018)(2)

Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable. For data communications to occur, the communicating devices must be part of a communication system made up of a combination of hardware (physical equipment) and software (programs). The effectiveness of a data communications system depends on four fundamental characteristics: delivery, accuracy, timeliness, and jitter.

Delivery. The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user.

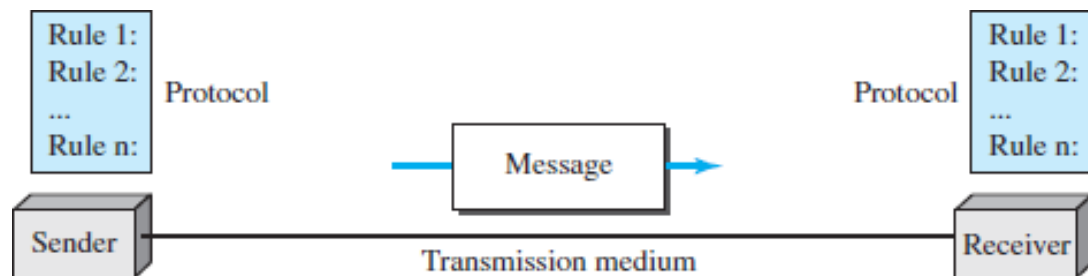
Accuracy. The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable.

Timeliness. The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called real-time transmission.

Jitter. Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets. For example, let us assume that video packets are sent every 30 ms. If some of the packets arrive with 30-ms delay and others with 40-ms delay, an uneven quality in the video is the result.

1.1 Components

A data communications system has five components



Message: The **message** is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.

Sender: The **sender** is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.

Receiver: The **receiver** is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.

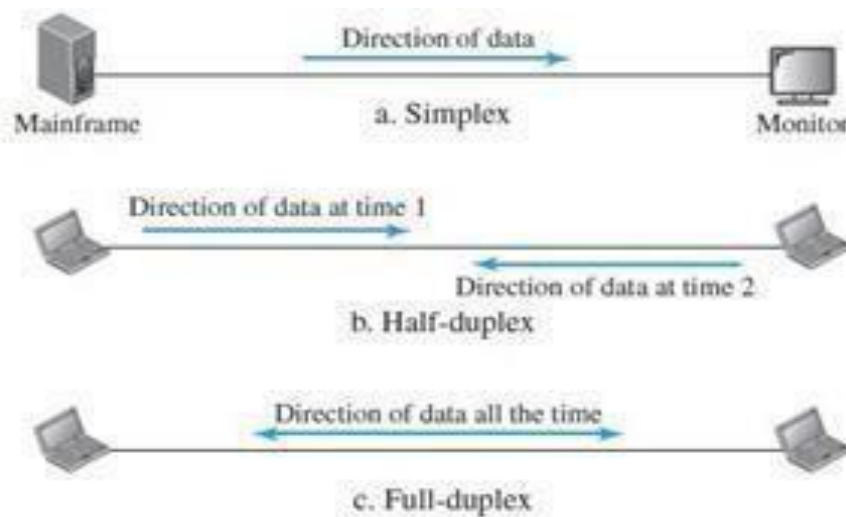
Transmission medium: The **transmission medium** is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves.

Protocol. A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating. The Key elements of a Protocol are as follows,

- **Syntax** – It refers to the structure or format of data meaning the order in which they are presented.
- **Semantics** – It refers to the meaning of each section of bit. How to do interpretation.
- **Timing** – When data should be sent and how fast they can be sent.

1.2 Data Flow

Communication between two devices can be simplex, half-duplex, or full-duplex as shown in Figure



1.3 Simplex

In **simplex mode**, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive

Half-Duplex

In **half-duplex mode**, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa.

In **full-duplex mode** (also called duplex), both stations can transmit and receive simultaneously.

1.4 Network Criteria

A network must be able to meet a certain number of criteria. The most important of these are performance, reliability, and security.

- ▢ Performance is often evaluated by two networking metrics: **throughput** and **delay**.
- ▢ Network **reliability** is measured by the frequency of failure, the time it takes a link
- ▢ Network **security** issues include protecting data from unauthorized access, protecting data from damage and development, and implementing policies and procedures for recovery from breaches and data losses.

Network Performance Metrics:

Bandwidth usage. Bandwidth is the maximum data transmission rate possible on a network.

- ▢ Throughput
- ▢ Latency
- ▢ Packet loss
- ▢ Retransmission
- ▢ Availability
- ▢ Connectivity

1.5 Type of Connection

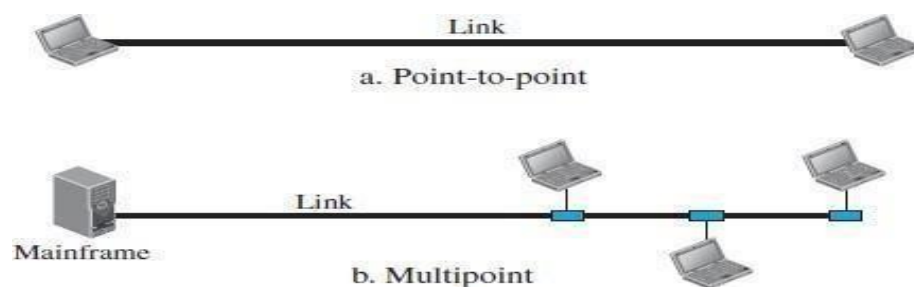
A network is two or more devices connected through links. A link is a communication pathway that transfers data from one device to another.

Point-to-Point

A **point-to-point connection** provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices.

Multipoint

A **multipoint** (also called **multidrop**) **connection** is one in which more than two specific devices share a single link.



1.6 Network Topology

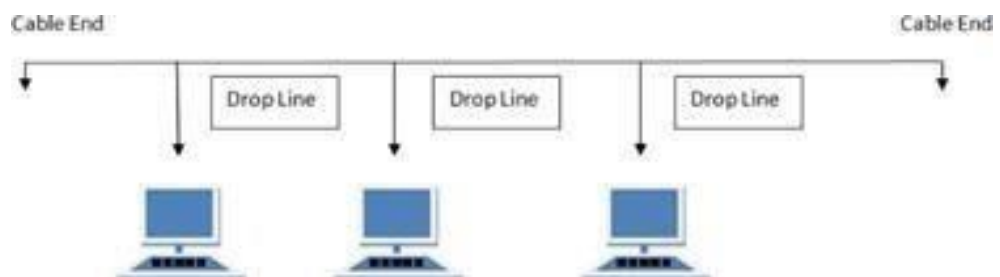
***Illustrate Network topology with neat diagram [13]**

*** What are the types of network topologies? [2]**

- Network Topology refers to the physical layout or geographical orientation and the arrangement of various elements like links, nodes etc.
- The various topologies are Bus topology, Ring topology, Star topology, Mesh Topology and Tree Topology.

Bus Topology

Bus topology is a network type in which every computer and network device is connected to single cable. When it has exactly two endpoints, then it is called Linear Bus topology.



Features of Bus Topology

- ☐ It transmits data only in one direction.
- ☐ Every device is connected to a single cable

Advantages of Bus Topology

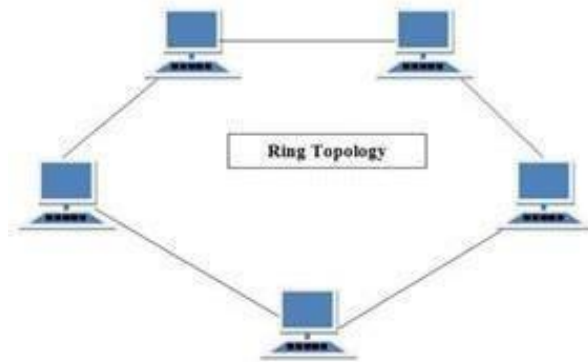
- ☐ It is cost-effective.
- ☐ Cable required is least compared to other network topology.
- ☐ Used in small networks.
- ☐ It is easy to understand.
- ☐ Easy to expand joining two cables together.

Disadvantages of Bus Topology

- ☐ Cables fails then whole network fails.
- ☐ If network traffic is heavy or nodes are more the performance of the network decreases.
- ☐ Cable has a limited length.
- ☐ It is slower than the ring topology.

RING Topology

It is called ring topology because it forms a ring as each computer is connected to another computer, with the last one connected to the first. Exactly two neighbors for each device.



Features of Ring Topology

- ❑ A number of repeaters are used for Ring topology with large number of nodes, because if someone wants to send some data to the last node in the ring topology with 100 nodes, then the data will have to pass through 99 nodes to reach the 100th node. Hence to prevent data loss repeaters are used in the network.
- ❑ The transmission is unidirectional, but it can be made bidirectional by having 2 connections between each Network Node, it is called **Dual Ring Topology**.
- ❑ In Dual Ring Topology, two ring networks are formed, and data flow is in opposite direction in them. Also, if one ring fails, the second ring can act as a backup, to keep the network up.
- ❑ Data is transferred in a sequential manner that is bit by bit. Data transmitted, has to pass through each node of the network, till the destination node.

Advantages of Ring Topology

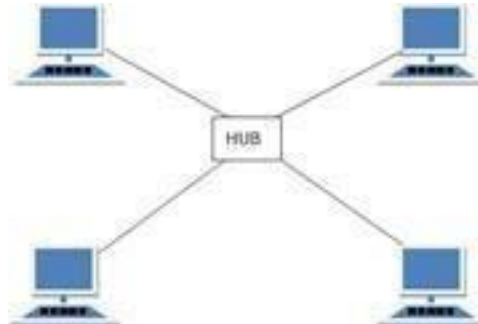
- ❑ Transmitting network is not affected by high traffic or by adding more nodes, as only the nodes having tokens can transmit data.
- ❑ Cheap to install and expand

Disadvantages of Ring Topology

- ❑ Troubleshooting is difficult in ring topology.
- ❑ Adding or deleting the computers disturbs the network activity.
- ❑ Failure of one computer disturbs the whole network

STAR Topology

In this type of topology all the computers are connected to a single hub through a cable. This hub is the central node and all other nodes are connected to the central node.



Features of Star Topology

- ☐ Every node has its own dedicated connection to the hub.
- ☐ Hub acts as a repeater for data flow.
- ☐ Can be used with twisted pair, Optical Fiber or coaxial cable.

Advantages of Star Topology

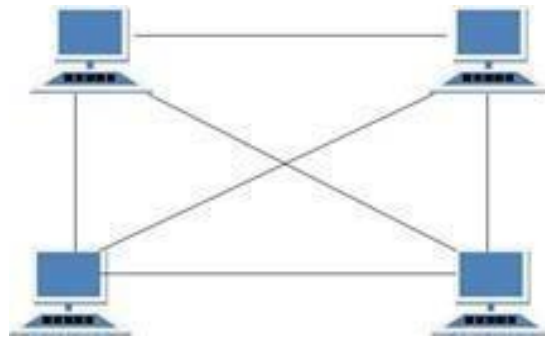
- ☐ Fast performance with few nodes and low network traffic.
- ☐ Hub can be upgraded easily.
- ☐ Easy to troubleshoot.
- ☐ Easy to setup and modify.
- ☐ Only that node is affected which has failed, rest of the nodes can work smoothly.

Disadvantages of Star Topology

- ☐ Cost of installation is high.
- ☐ Expensive to use.
- ☐ If the hub fails then the whole network is stopped because all the nodes depend on the hub.
- ☐ Performance is based on the hub that is it depends on its capacity

MESH Topology

It is a point-to-point connection to other nodes or devices. All the network nodes are connected to each other. Mesh has $\frac{n(n-1)}{2}$ physical channels to link n devices.



Features of Mesh Topology

- ☐ Fully connected.
- ☐ Robust.
- ☐ Not flexible.

Advantages of Mesh Topology

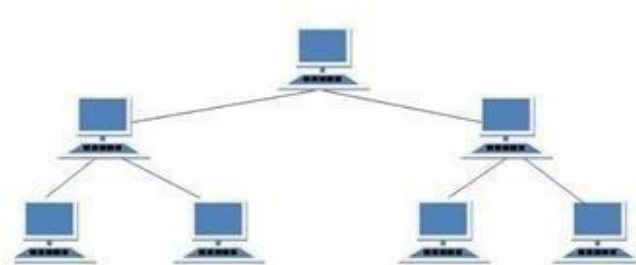
- ☐ Each connection can carry its own data load.
- ☐ It is robust.
- ☐ Fault is diagnosed easily.
- ☐ Provides security and privacy.

Disadvantages of Mesh Topology

- ☐ Installation and configuration is difficult.
- ☐ Cabling cost is more.
- ☐ Bulk wiring is required.

TREE Topology

It has a root node and all other nodes are connected to it forming a hierarchy. It is also called hierarchical topology. It should at least have three levels to the hierarchy.



Features of Tree Topology

- ❑ Ideal if workstations are located in groups.
- ❑ Used in Wide Area Network.

Advantages of Tree Topology

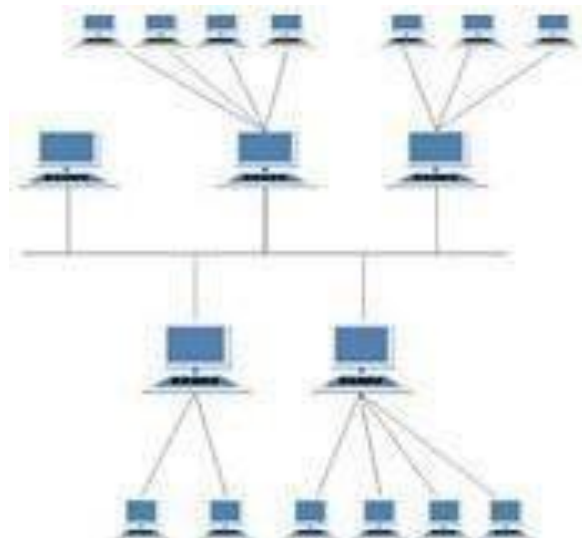
- ❑ Extension of bus and star topologies.
- ❑ Expansion of nodes is possible and easy.
- ❑ Easily managed and maintained.
- ❑ Error detection is easily done.

Disadvantages of Tree Topology

- ❑ Heavily cabled.
- ❑ Costly.
- ❑ If more nodes are added maintenance is difficult.
- ❑ Central hub fails, network fails.

HYBRID Topology

It is two different types of topologies which is a mixture of two or more topologies. For example if in an office in one department ring topology is used and in another star topology is used, connecting these topologies will result in Hybrid Topology (ring topology and star topology).



2. NETWORKS TYPES

***Illustrate Network types with neat diagram (Nov/Dec 2021) (13)**

*** What are the network types? (2)**

The **Network** allows computers to **connect and communicate** with different computers via any medium. LAN, MAN and WAN are the three major types of the network designed to operate over the area they cover.

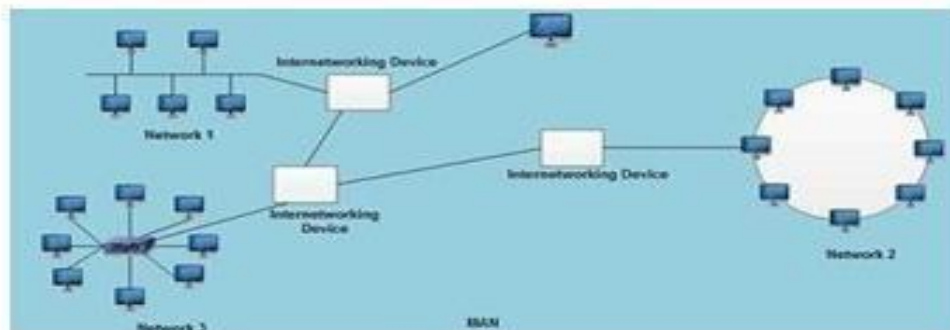
LAN (Local Area Network)

A **Local Area Network** is a privately owned computer **network** covering a **small Networks geographical area**, like a home, office, or groups of buildings e.g. a school Network. A LAN is used to connect the computers and other network devices so that the devices can communicate with each other to share the resources. The resources to be shared can be a hardware device like printer, software like an application program or data. The size of LAN is usually small. The various devices in LAN are connected to central devices called Hub or Switch using a cable.



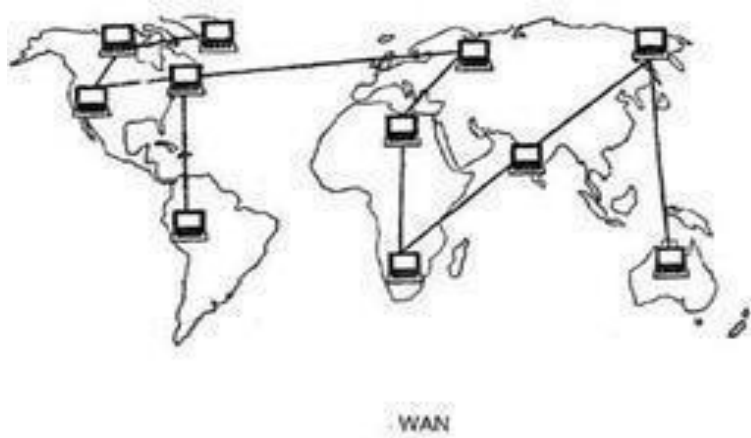
MAN (Metropolitan Area Networks)

MAN stands for Metropolitan Area Networks is one of a number of types of networks. AMAN is a relatively new class of network. MAN is larger than a local area network and as its name implies, covers the area of a single city. MANs rarely extend beyond 100 KM and frequently comprise a combination of different hardware and transmission media. It can be single network such as a cable TV network, or it is a means of connecting a number of LANs into a larger network so that resources can be shared LAN to LAN as well as device to device.



WAN (Wide Area Networks)

A wide area network (WAN) is a telecommunication network. A wide area network is simply a LAN of LANs or Network of Networks. WANs connect LANs that may be on opposite sides of a building, across the country or around the world. WANS are characterized by the slowest data communication rates and the largest distances. WANs can be of two types: an enterprise WAN and Global WAN.



3. PROTOCOL LAYERING

***Differentiate the protocol layering of TCP/IP protocol suite and OSI model. Explain both with the diagram.(NOV/DEC 2021)(13)**

A **protocol** is a set of rules and standards that primarily outline a language that devices will use to communicate. There are an excellent range of protocols in use extensively in networking, and that they are usually implemented in numerous layers.

It provides a communication service where the process is used to exchange the messages. When the communication is simple, we can use only one simple protocol.

When the communication is complex, we must divide the task between different layers, so, we need to follow a protocol at each layer, this technique we used to call protocol layering. This layering allows us to separate the services from the implementation.

Each layer needs to receive a set of services from the lower layer and to give the services to the upper layer. The modification done in any one layer will not affect the other layers.

Basic Elements of Layered Architecture

The basic elements of the layered architecture are as follows –

- ▣ Service – Set of actions or services provided from one layer to the higher layer.
- ▣ Protocol – It defines a set of rules where a layer uses to exchange the information with its peer entity. It is concerned about both the contents and order of the messages used.
- ▣ Interface – It is a way through that the message is transferred from one layer to another layer.

Reasons

The reasons for using layered protocols are explained below –

- ▣ Layering of protocols provides well-defined interfaces between the layers, so that a change in one layer does not affect an adjacent layer.
- ▣ The protocols of a network are extremely complicated and designing them in layers makes their implementation more feasible.

Advantages

The advantages of layered protocols are as follows –

- ▣ Assists in protocol style, as a result of protocols that operate at a particular layer have outlined information that they work and a defined interface to the layers on top of and below.
- ▣ Foster's competition because products from completely different vendors will work along.
- ▣ Prevents technology or capability changes in one layer from touching different layers above and

below.

- ☐ Provides a typical language to explain networking functions and capabilities.

Disadvantages

The disadvantages of layered protocols are as follows –

- ☐ The main disadvantages of layered systems consist primarily of overhead each in computation and in message headers caused by the abstraction barriers between layers. Because a message typically should pass through several (10 or more) protocol layers the overhead of those boundaries is commonly more than the computation being done.
- ☐ The upper-level layers cannot see what is within the lower layers, implying that an application cannot correct where in an exceedingly connection a problem is or precisely what the matter is.
- ☐ The higher-level layers cannot control all aspects of the lower layers, so that they cannot modify the transfer system if helpful (like controlling windowing, header compression, CRC/parity checking, et cetera), nor specify routing, and should rely on the lower protocols operating, and cannot specify alternatives when there are issues.

4. TCP/IP PROTOCOL SUITS

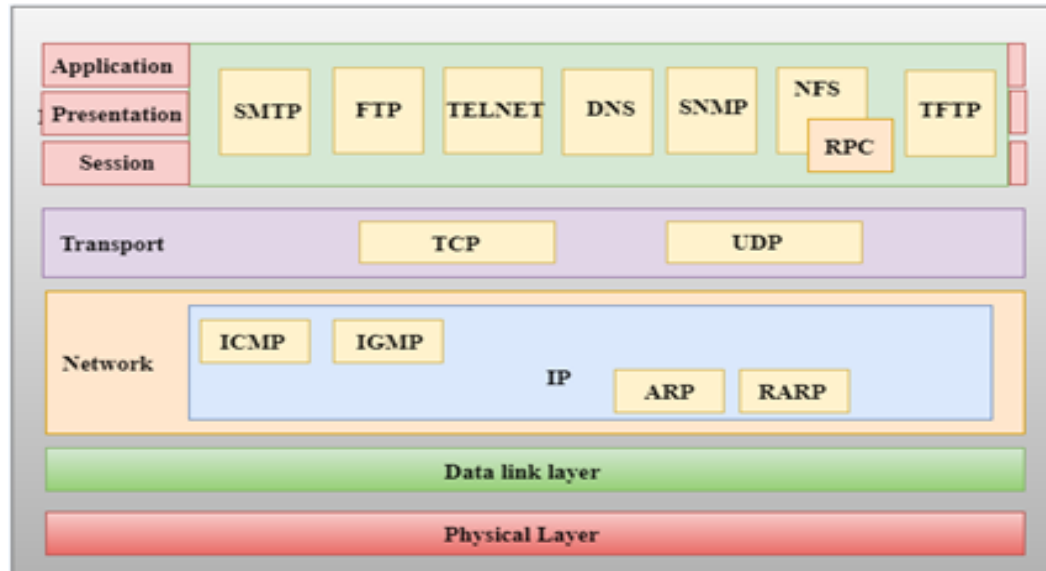
***Differentiate the protocol layering of TCP/IP protocol suite and OSI model. Explain both with the diagram.(NOV/DEC 2021)(13)**

- The TCP/IP model was developed prior to the OSI model.
- The TCP/IP model is not exactly similar to the OSI model.
- The TCP/IP model consists of five layers: the application layer, transport layer, network layer, data link layer and physical layer.
- The first four layers provide physical standards, network interface, internetworking, and transport functions that correspond to the first four layers of the OSI model and these four layers are represented in TCP/IP model by a single layer called the application layer.
- TCP/IP is a hierarchical protocol made up of interactive modules, and each of them provides specific functionality. Here, hierarchical means that each upper-layer protocol is supported by two or more lower-level protocol.

Network Access Layer

- A network layer is the lowest layer of the TCP/IP model.
- A network layer is the combination of the Physical layer and Data Link layer defined in the OSI reference model.
- It defines how the data should be sent physically through the network.
- This layer is mainly responsible for the transmission of the data between two devices on the same network.
- The functions carried out by this layer are encapsulating the IP datagram into frames transmitted by the network and mapping of IP addresses into physical addresses.
- The protocols used by this layer are Ethernet, token ring, FDDI, X.25, frame relay.

Functions of TCP/IP layers:



Internet Layer

- An internet layer is the second layer of the TCP/IP model.
- An internet layer is also known as the network layer.
- The main responsibility of the internet layer is to send the packets from any network, and they arrive at the destination irrespective of the route they take. Following are the protocols used in this layer are:
- **IP Protocol:** IP protocol is used in this layer, and it is the most significant part of the entire TCP/IP suite.
- **Transport Layer :** The transport layer is responsible for the reliability, flow control, and correction of data which is being sent over the network.
- The two protocols used in the transport layer are **User Datagram protocol and Transmission control protocol**.

Application Layer

- An application layer is the topmost layer in the TCP/IP model.
- It is responsible for handling high-level protocols, issues of representation.
- This layer allows the user to interact with the application.
- When one application layer protocol wants to communicate with another application layer, it forwards its data to the transport layer.
- There is an ambiguity in the application layer. Every application cannot be placed inside the application layer except those who interact with the communication.

system. For example: text editor cannot be considered in application layer while web browser using **HTTP** protocol to interact with the network where **HTTP** protocol is an application layer protocol.

Following are the main protocols used in the application layer:

- **HTTP:** HTTP stands for Hypertext transfer protocol. This protocol allows us to access the data over the world wide web. It transfers the data in the form of plain text, audio, video. It is known as a Hypertext transfer protocol as it has the efficiency to use in a hypertext environment where there are rapid jumps from one document to another.
- **SNMP:** SNMP stands for Simple Network Management Protocol. It is a framework used for managing the devices on the internet by using the TCP/IP protocol suite.
- **SMTP:** SMTP stands for Simple mail transfer protocol. The TCP/IP protocol that supports the e-mail is known as a Simple mail transfer protocol. This protocol is used to send the data to another-mail address.
- **DNS:** DNS stands for Domain Name System. An IP address is used to identify the connection of a host to the internet uniquely. But, people prefer to use the names instead of addresses. Therefore, the system that maps the name to the address is known as Domain Name System.
- **TELNET:** It is an abbreviation for Terminal Network. It establishes the connection between the local computer and remote computer in such a way that the local terminal appears to be a terminal at the remote system.
- **FTP:** FTP stands for File Transfer Protocol. FTP is a standard internet protocol used for transmitting the files from one computer to another computer.

5. OSI MODEL

- * Write a neat sketch, explain the architecture of an OSI seven layer model (NOV/DEC 2017)(13)
- * Draw the ISO-OSI architecture and outline the functions performed by each layer(NOV/Dec 2019) (13)
- * What is the use of data link layer in OSI? (NOV/DEC 2015)(13)
- *Define Flow control(May/June 2016)(2)
- *Justify the need for layer five in the OSI model.(NOV/DEC 2021)(2)
- *What is the need for another checking mechanism at the transport layer even though the data link layer is capable of detecting the errors between the hops (NOV/DEC 2020)(13)
- *What are the layers of the ISO/OSI protocol stack? Briefly list out their functions.

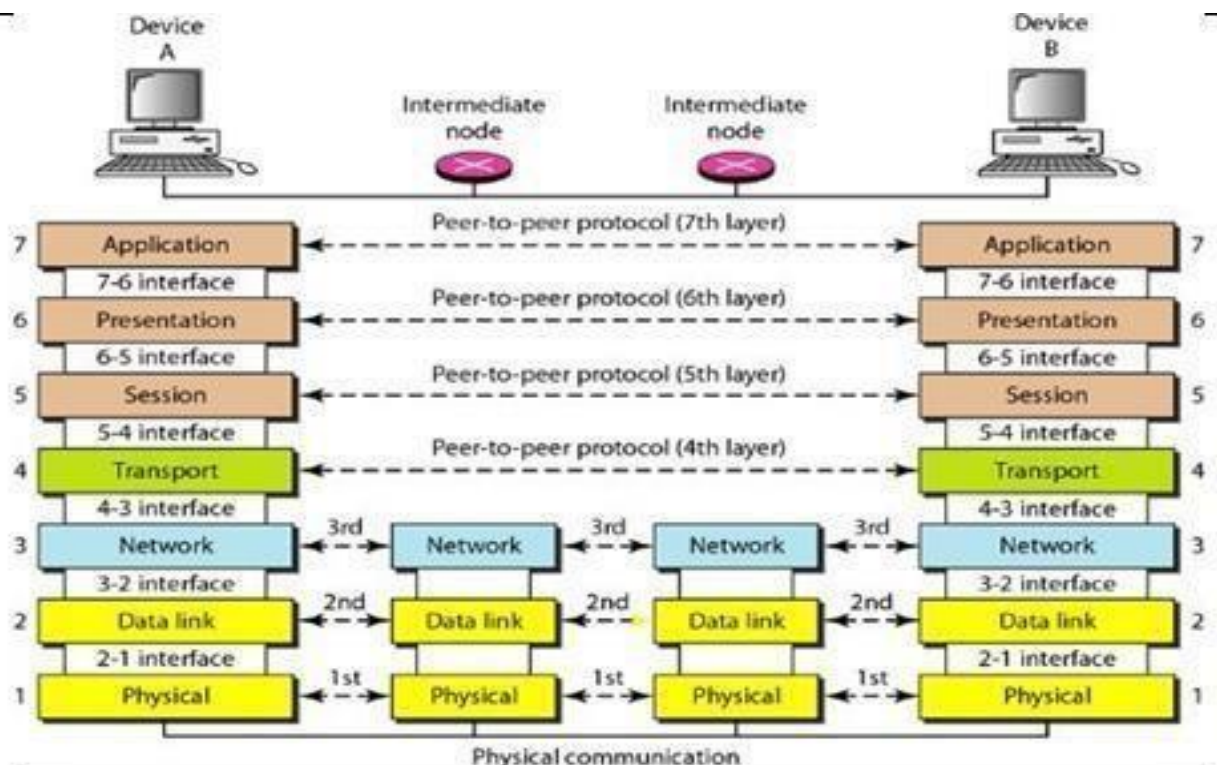
(NOV/DEC 2020)(13)

***What are the layers of the ISO/OSI protocol stack? Briefly list out their functions.(APR/MAY 2021)(13)**

***How are the subgroups of OSI model layers segregated by their functions? (APR/MAY 2020)(2)**

- OSI stands for **Open System Interconnection** is a reference model that describes how information from a software application in one computer moves through a physical medium to the software application in another computer.
- OSI consists of seven layers, and each layer performs a particular network function.
- OSI model was developed by the International Organization for Standardization (ISO) in 1984, and it is now considered as an architectural model for the inter-computer communications.
- OSI model divides the whole task into seven smaller and manageable tasks. Each layer is assigned a particular task.
 - Each layer is self-contained, so that task assigned to each layer can be performed independently.

Physical Layer



- ▮ The main functionality of the physical layer is to transmit the individual bits from one node to another node.
- ▮ It is the lowest layer of the OSI model.
- ▮ It establishes, maintains and deactivates the physical connection.
- ▮ It specifies the mechanical, electrical and procedural network interface specifications.

Functions of a Physical layer:

- ❏ **Line Configuration:** It defines the way how two or more devices can be connected physically.
- ❏ **Data Transmission:** It defines the transmission mode whether it is simplex, half-duplex or full-duplex mode between the two devices on the network.
- ❏ **Topology:** It defines the way how network devices are arranged.
- ❏ **Signals:** It determines the type of the signal used for transmitting the information.

Data-Link Layer

- ❏ This layer is responsible for the error-free transfer of data frames.
- ❏ It defines the format of the data on the network.
- ❏ It provides a reliable and efficient communication between two or more devices.
- ❏ It is mainly responsible for the unique identification of each device that resides on a local network.

***List out the functions of the Data Link Layer (Apr/May 2021)(2)**

Functions of the Data-link layer

- ❏ **Physical Addressing:** The Data link layer adds a header to the frame that contains a destination address. The frame is transmitted to the destination address mentioned in the header.
- ❏ **Flow Control:** Flow control is the main functionality of the Data-link layer. It is the technique through which the constant data rate is maintained on both the sides so that no data gets corrupted. It ensures that the transmitting station such as a server with higher processing speed does not exceed the receiving station, with lower processing speed.
- ❏ **Error Control:** Error control is achieved by adding a calculated value CRC (Cyclic Redundancy Check) that is placed to the Data link layer's trailer which is added to the message frame before it is sent to the physical layer. If any error seems to occur, then the receiver sends the acknowledgment for the retransmission of the corrupted frames.
- ❏ **Access Control:** When two or more devices are connected to the same communication channel, then the data link layer protocols are used to determine which device has control over the link at a given time.

Network Layer

- ❏ It is a layer 3 that manages device addressing, tracks the location of devices on the network.
- ❏ It determines the best path to move data from source to the destination based on the network conditions, the priority of service, and other factors.
- ❏ The Data link layer is responsible for routing and forwarding the packets.
- ❏ Routers are the layer 3 devices, they are specified in this layer and used to provide the routing services within an internetwork.
- ❏ The protocols used to route the network traffic are known as Network layer protocols. Examples of protocols are IP and IPv6.

Functions of Network Layer:

- ❏ **Addressing:** A Network layer adds the source and destination address to the header of the

frame. Addressing is used to identify the device on the internet.

- ▢ **Routing:** Routing is the major component of the network layer, and it determines the best optimal path out of the multiple paths from source to the destination.

Transport Layer

- ▢ The Transport layer is a Layer 4 ensures that messages are transmitted in the order in which they are sent and there is no duplication of data.
- ▢ The main responsibility of the transport layer is to transfer the data completely.
- ▢ It receives the data from the upper layer and converts them into smaller units known as segments.
- ▢ This layer can be termed as an end-to-end layer as it provides a point-to-point connection between source and destination to deliver the data reliably.

Functions of Transport Layer:

- ▢ **Service-point addressing:** Computers run several programs simultaneously due to this reason, the transmission of data from source to the destination not only from one computer to another computer but also from one process to another process. The transport layer adds the header that contains the address known as a service-point address or port address. The responsibility of the network layer is to transmit the data from one computer to another computer and the responsibility of the transport layer is to transmit the message to the correct process.
- ▢ **Segmentation and reassembly:** When the transport layer receives the message from the upper layer, it divides the message into multiple segments, and each segment is assigned with a sequence number that uniquely identifies each segment. When the message has arrived at the destination, then the transport layer reassembles the message based on their sequence numbers.
- ▢ **Connection control:** Transport layer provides two services Connection-oriented service and connectionless service. A connectionless service treats each segment as an individual packet, and they all travel in different routes to reach the destination. A connection-oriented service makes a connection with the transport layer at the destination machine before delivering the packets. In connection-oriented service, all the packets travel in the single route.
- ▢ **Flow control:** The transport layer is also responsible for flow control but it is performed end-to-end rather than across a single link.
- ▢ **Error control:** The transport layer is also responsible for Error control. Error control is performed end-to-end rather than across the single link. The sender transport layer ensures that message reach at the destination without any error.

Session Layer

- ▢ It is a layer 3 in the OSI model.
- ▢ The Session layer is used to establish, maintain and synchronizes the interaction between communicating devices.

Functions of Session layer:

- ▢ **Dialog control:** Session layer acts as a dialog controller that creates a dialog between two processes or we can say that it allows the communication between two processes which can be either half-duplex or full-duplex.
- ▢ **Synchronization:** Session layer adds some checkpoints when transmitting the data in a sequence. If some error occurs in the middle of the transmission of data, then the transmission

will take place again from the checkpoint. This process is known as Synchronization and recovery.

Presentation Layer

- ❑ A Presentation layer is mainly concerned with the syntax and semantics of the information exchanged between the two systems.
- ❑ It acts as a data translator for a network.
- ❑ This layer is a part of the operating system that converts the data from one presentation format to another format.
- ❑ The Presentation layer is also known as the syntax layer.

Functions of Presentation layer:

- ❑ **Translation:** The processes in two systems exchange the information in the form of character strings, numbers and so on. Different computers use different encoding methods, the presentation layer handles the interoperability between the different encoding methods. It converts the data from sender-dependent format into a common format and changes the common format into receiver-dependent format at the receiving end.
- ❑ **Encryption:** Encryption is needed to maintain privacy. Encryption is a process of converting the sender-transmitted information into another form and sends the resulting message over the network.
- ❑ **Compression:** Data compression is a process of compressing the data, i.e., it reduces the number of bits to be transmitted. Data compression is very important in multimedia such as text, audio, video.

Application Layer

- ❑ An application layer serves as a window for users and application processes to access network service.
- ❑ It handles issues such as network transparency, resource allocation, etc.
- ❑ An application layer is not an application, but it performs the application layer functions.
- ❑ This layer provides the network services to the end-users.

Functions of Application layer:

- ❑ **File transfer, access, and management (FTAM):** An application layer allows a user to access the files in a remote computer, to retrieve the files from a computer and to manage the files in a remote computer.
- ❑ **Mail services:** An application layer provides the facility for email forwarding and storage.
- ❑ **Directory services:** An application provides the distributed database sources and is used to provide that global information about various objects.

Problem

Two channels, one with a bit rate of 190 kbps and another with a bit rate 180 kbps are to be multiplexed using pulse stuffing TDM with no synchronization bits. Answer the following questions:

- i. What is the size of a frame in bits? (3)
- ii. What is the frame rate? (3)
- iii. What is the duration of a frame? (NOV/DEC 2020) (3) (3)
- iv. What is the data rate? (4)

Sol: We need to add extra bits to the second source to make both rates = 190 kbps. Now we have two sources, each of 190 Kbps.

a. The frame carries 1 bit from each source. Frame size = 1 + 1 = 2 bits.

b. Each frame carries 1 bit from each 190-kbps source. Frame rate = 190,000 frames/s.

c. Frame duration = $1 / (\text{frame rate}) = 1 / 190,000 = 5.3 \mu\text{s}$.

d. Output data rate = $(190,000 \text{ frames/s}) \times (2 \text{ bits/frame}) = 380 \text{ kbps}$. Here the output bit rate is greater than the sum of the input rates (370 kbps) because of extra bits added to the second source.

consider a point to point link 2km in length at what bandwidth would propagation delay (at speed of $2 \times 10^8 \text{ m/sec}$ equal transmit delay for 100 byte packet? What about 512 byte packet.

- Speed of light is $2 \times 10^8 \text{ m/sec}$.
- 100 bytes = ?
- 512 bytes = ?

Distance = 2 Kms. = $2 \times 10^3 \text{ Mts}$

- $P = D / SL$

$$2 \times 10^3 \text{ m} / 2 \times 10^8 \text{ m/sec} = 1 \times 10^{-5} \times 10 / 10 = 10 \times 10^{-6} \text{ sec} = 10 \mu\text{s}.$$

transmit = size / bandwidth

Bandwidth = size / transmit

$$= 100 \text{ byte} / 10 \times 10^{-6} \text{ sec}$$

$$= 100 \times 8 \text{ bits} / 10 \times 10^{-6} \text{ sec}$$

$$= 80 \times 10^6 \text{ bps}$$

$$= 80 \text{ Mbps}$$

o for 512 byte packet

Bandwidth = size / transmit

$$= 512 \text{ bytes} / 10 \times 10^{-6} \text{ sec}$$

$$= 8 \times 512 \text{ bits} / 10 \times 10^{-6} \text{ sec}$$

$$= 4096 / 10 \times 10^{-6} \text{ sec}$$

$$= 409.6 \times 10^6 \text{ bps}$$

$$= 409.6 \text{ Mbps}$$

- 2. For a Dial up link of 56 Kbps bandwidth and $87 \mu\text{s}$ round trip delay find the delay X Bandwidth product.**

$$\text{Delay} \times \text{Bandwidth} = 86 \times 10^{-6} \times 56 \times 10^3 \text{ b/s}$$

$$= 4872 \times 10^{-3} \text{ bits}$$

$$= 4872 / 1000$$

$$= 4.873 \text{ which is approximately 5 bits}$$

3. Suppose a 128 Kbps point to point link is set up between earth and a river on mars. The distance from earth to mars is approximately 55Gm and data travel over the link at speed of light 3×10^8 m/sec.

- Calculate the minimum RTT for the link
- Calculate the delay X bandwidth product.
- A camera on the river takes a picture of its surroundings and send those to earth. How quickly after a picture is taken can it reach mission control; on earth? Assume that each image is 5 MB in size.

- Minimum RTT = 2 X propagation delay.

$$= 2 \times \text{distance} / \text{speed of light}$$

$$= 2 \times 55 \times 10^9 \text{ m} / 3 \times 10^8 \text{ m/sec}$$

$$= 2 \times 550/3 \text{ sec}$$

$$= 2 \times 184 \text{ sec}$$

$$= 368 \text{ sec.}$$
- Delay X bandwidth

$$= 184 \text{ sec} \times 128 \times 10^3 \text{ bps}$$

$$= 23552 \times 10^3 \text{ bits}$$

$$= 23552 \times 10^3 \times 10^3 / 10^{-3}$$

$$= 23.552 \text{ Mbits}$$

$$= 2.944 \text{ Mbytes.}$$

$$= \text{which is approximately } 3 \text{ Mbytes}$$
- Transmit = size / bandwidth

$$= 5 \text{ MB} / 128 \times 10^3 \text{ bps}$$

$$= 5 \times 8 \text{ bits} / 128 \times 10^3$$

$$= 40 \times 10^6 \text{ bits} / 128 \times 10^6 \text{ bps}$$

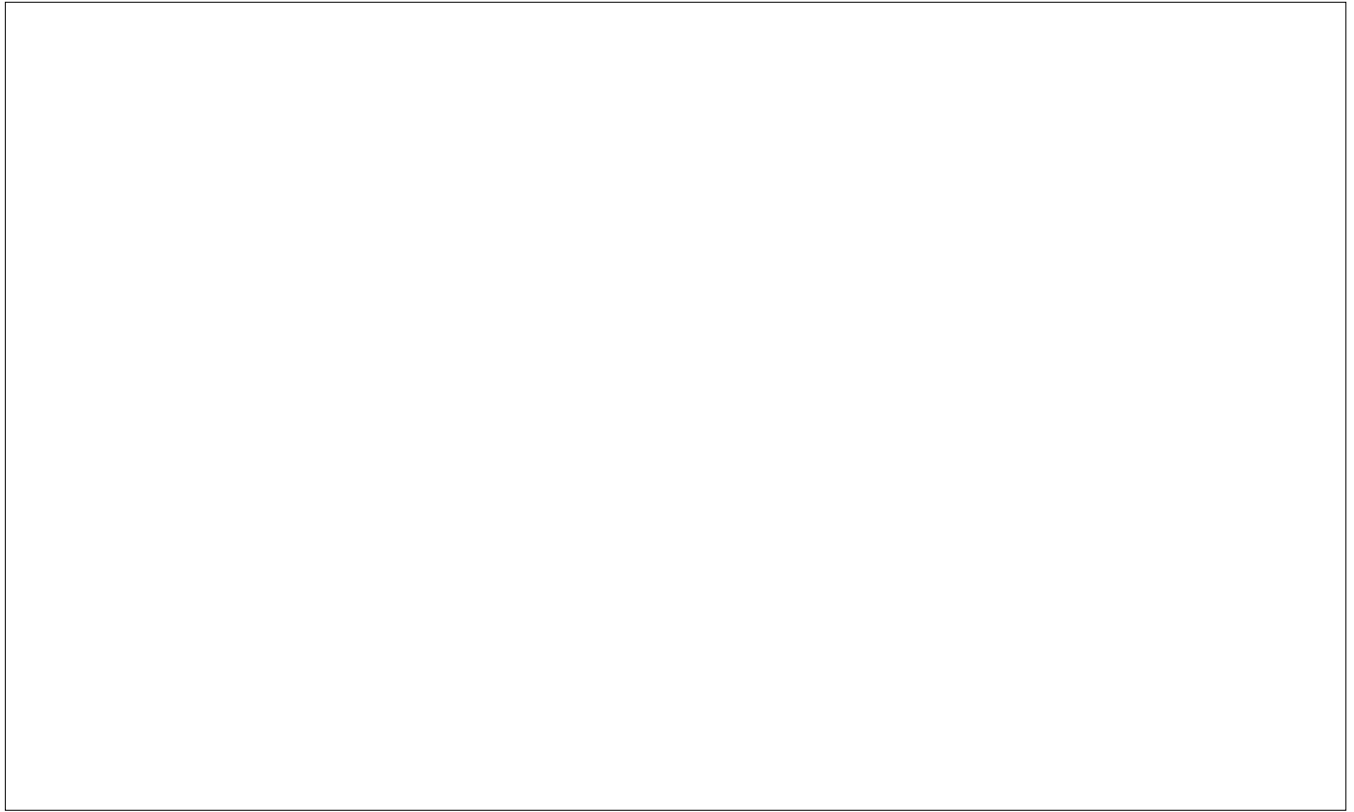
$$= 0.3125 \times 10^3 \text{ sec}$$

$$= 312.5 \text{ sec}$$

Total time = transmit + delay

$$= 312.5 + 184$$

1. Suppose a 100 Mbps point to point link is being set up between earth and a new lunar colony. The distance from moon to earth is approximately 3,85,000 kms and data travels over the link at the speed of light 3×10^8 m/sec.
- Calculate the minimum RTT for the link.
 - Using the RTT as the delay, calculate the delay X bandwidth product for the link.
 - A camera on the lunar base takes pictures of earth and saves them in digital format disk. Suppose mission control on earth wishes to download most current image, which is 25MB. What is the minimum amount of time that will elapse between when the request for the data goes out and the transfer finished?



2. For a wireless LAN of 54 Mbps bandwidth and 0.33 μ s Round trip delay find the delayx bandwidth product.

$$\begin{aligned}\text{Delay X Bandwidth} &= 54 \times 10^6 \text{ bps} \times 0.33 \times 10^{-6} \text{ sec} \\ &= 17.82 \text{ bits which is approximately 18 bits.}\end{aligned}$$

HyperText Transfer Protocol (HTTP)

The HyperText Transfer Protocol (HTTP) is used to define how the client-server programs can be written to retrieve web pages from the Web. An HTTP client sends a request; an HTTP server returns a response. The server uses the port number 80; the client uses a temporary port number. HTTP uses the services of TCP.

Nonpersistent versus Persistent Connections

If the web pages, objects to be retrieved, are located on different servers, we do not have any other choice than to create a new TCP connection for retrieving each object. However, if some of the objects are located on the same server, we have two choices: to retrieve each object using a new TCP connection or to make a TCP connection and retrieve them all. The first method is referred to as a nonpersistent connection, the second as a persistent connection. HTTP, prior to version 1.1, specified nonpersistent connections, while persistent connections are the default in version 1.1, but it can be changed by the user.

Nonpersistent Connections

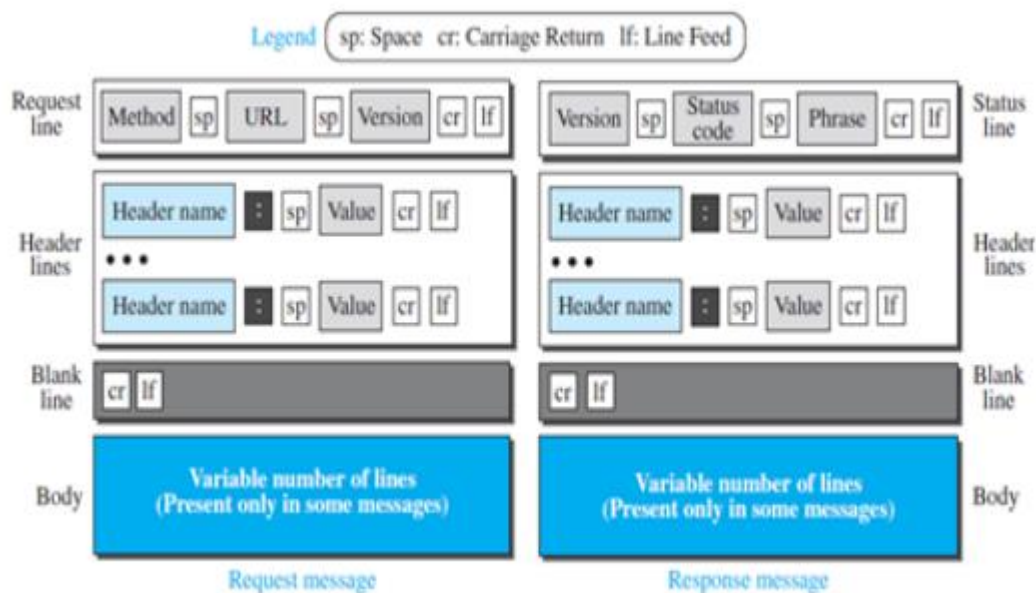
In a nonpersistent connection, one TCP connection is made for each request/response.

The following lists the steps in this strategy:

1. The client opens a TCP connection and sends a request.
2. The server sends the response and closes the connection.
3. The client reads the data until it encounters an end-of-file marker; it then closes the connection. In this strategy, if a file contains links to N different pictures in different files (all located on the same server), the connection must be opened and closed N + 1 times. The nonpersistent strategy imposes high overhead on the server because the server needs N + 1 different buffers each time a connection is opened. Persistent Connections HTTP version 1.1 specifies

a persistent connection by default. In a persistent connection, the server leaves the connection open for more requests after sending a response. The server can close the connection at the request of a client or if a time -out has been reached. The sender usually sends the length of the data with each res ponse.

However, there are some occasions when the sender does not knowthe length of the data. This is the case when a document is created dynamically or actively. In these cases, the server informs the client that the length is not known and closes the connection after sending the data so the client knows that the end of the data hasbeen reached. Time and resources are saved using persistent connections. Only one set of buffers and variables needs to be set for the connection at each site. The round trip time for connection establishment and connection termination is saved. Message Formats The HTTP protocol defines the format of the request and response messages, as shown in Figure below. We have put the two formats next to each other for comparison. Each message is made of four sections. The first section in the request message is called the request line; the first section in the response message is called the status line. The other three sections have the same names in the request and response messages.



Request Message

As we said before, the first line in a request message is called a request line.

There are three fields in this line separated by one space and terminated by two characters (carriage return and line feed) as shown in Figure above.

The fields are called method, URL, and version.

The method field defines the request types.

In version 1.1 of HTTP, several methods are defined, as shown in Table. Most of the time, the client uses the GET method to send a request. In this case, the body of the message is empty.

The HEAD method is used when the client needs only some information about the web page from the server, such as the last time it was modified. It can also be used to test the validity of a URL. The response message in this case has only the header section; the body section is empty.

The PUT method is theinverse of the GET method; it allows the client to post a new web page on the server (if permitted). The POST method is similar to the PUT method, but it is used to send some information to the server to be added to the web page or to modify the web page.

The TRACEmethod is used for debugging; the client asks the server to echo back the request to check whether the server is getting the requests.

The DELETE method allows the client to delete a web page on the server if the client has permission to do so. The CONNECT methodwas originally made as a reserve method; it may be used by proxy servers, as discussed later. Finally, the OPTIONS method allows the client to ask about the properties of a web page. The second field, URL, was discussed earlier in the chapter. It defines the address and name of the corresponding web page. The third field,

version, gives the version of the protocol; the most current version of HTTP is 1.1

<i>Method</i>	<i>Action</i>
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
PUT	Sends a document from the client to the server
POST	Sends some information from the client to the server
TRACE	Echoes the incoming request
DELETE	Removes the web page
CONNECT	Reserved
OPTIONS	Inquires about available options

After the request line, we can have zero or more request header lines. Each header line sends additional information from the client to the server. For example, the client can request that the document be sent in a special format. Each header line has a header name, a colon, a space, and a header value as shown in Figure. Table below shows some header names commonly used in a request. The value field defines the values associated with each header name. The list of values can be found in the corresponding RFCs. The body can be present in a request message. Usually, it contains the comment to be sent or the file to be published on the website when the method is PUT or POST.

<i>Header</i>	<i>Description</i>
User-agent	Identifies the client program
Accept	Shows the media format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
Host	Shows the host and port number of the client
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Cookie	Returns the cookie to the server (explained later)
If-Modified-Since	If the file is modified since a specific date

Response Message

The format of the response message is also shown in Figure . A response message consists of a status line, header lines, a blank line, and sometimes a body.

The first line in a response message is called the status line.

There are three fields in this line separated by spaces and terminated by a carriage return and line feed.

The first field defines the version of HTTP protocol, currently 1.1.

The status code field defines the status of the request. It consists of three digits. Whereas the codes in the 100 range are only informational, the codes in the 200 range indicate a successful request. The codes in the 300 range redirect the client to another URL, and the codes in the 400 range indicate an error at the client site. Finally, the codes in the 500 range indicate an error at the server site.

The status phrase explains the status code in text form. After the status line, we can have zero or more response header lines. Each header line sends additional information from the server to the client. For example, the sender can send extra information about the document. Each header line has a header name, a colon, a space, and a header value.

<i>Header</i>	<i>Description</i>
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Server	Gives information about the server
Set-Cookie	The server asks the client to save a cookie
Content-Encoding	Specifies the encoding scheme
Content-Language	Specifies the language
Content-Length	Shows the length of the document
Content-Type	Specifies the media type
Location	To ask the client to send the request to another site
Accept-Ranges	The server will accept the requested byte-ranges
Last-modified	Gives the date and time of the last change

The body contains the document to be sent from the server to the client. The body is present unless the response is an error message.

Cookies

The World Wide Web was originally designed as a stateless entity. A client sends a request; a server responds. Their relationship is over. The original purpose of the Web, retrieving publicly available documents, exactly fits this design. Today the Web has other functions that need to remember some information about the clients; some are listed below:

- ☐ Websites are being used as electronic stores that allow users to browse through the store, select wanted items, put them in an electronic cart, and pay at the end with a credit card.
- ☐ Some websites need to allow access to registered clients only.
- ☐ Some websites are used as portals: the user selects the web pages he wants to see.
- ☐ Some websites are just advertising agencies. For these purposes, the cookie mechanism was devised.

Creating and Storing Cookies The creation and storing of cookies depend on the implementation; however, the principle is the same.

1. When a server receives a request from a client, it stores information about the client in a file or a string. The information may include the domain name of the client, the contents of the cookie (information the server has gathered about the client such as name, registration number, and so on), a timestamp, and other information depending on the implementation.

2. The server includes the cookie in the response that it sends to the client.

3. When the client receives the response, the browser stores the cookie in the cookie directory, which is sorted by the server domain name. Using Cookies When a client sends a request to a server, the browser looks in the cookie directory to see if it can find a cookie sent by that server. If found, the cookie is included in the request. When the server receives the request, it knows that this is an old client, not a new one. Note that the contents of the cookie are never read by the browser or disclosed to the user. It is a cookie made by the server and eaten by the server.

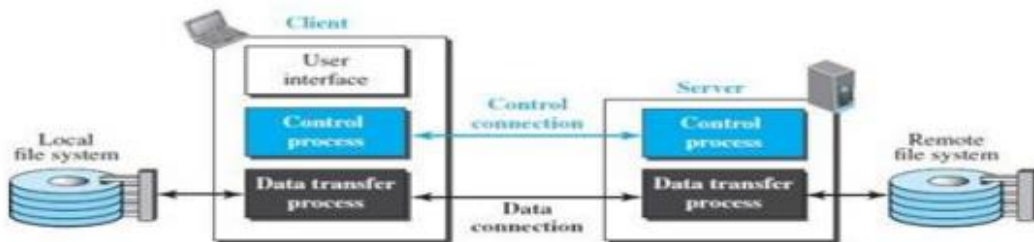
- ☐ An electronic store (e-commerce) can use a cookie for its client shoppers. When a client selects an item and inserts it in a cart, a cookie that contains information about the item, such as its number and unit price, is sent to the browser. If the client selects a second item, the cookie is updated with the new selection information, and so on. When the client finishes shopping and wants to check out, the last cookie is retrieved and the total charge is calculated.

- ☐ The site that restricts access to registered clients only sends a cookie to the client when the client registers for the first time. For any repeated access, only those clients that send the appropriate cookie are allowed.

- ☐ A web portal uses the cookie in a similar way. When a user selects her favorite pages, a cookie is made and sent. If the site is accessed again, the cookie is sent to the server to show what the client is looking for. ☐ A cookie is also used by advertising agencies. An advertising agency can place banner ads on some main website that is often visited by users. The advertising agency supplies only a URL that gives the advertising agency's address instead of the banner itself. When a user visits the main website and clicks the icon of a corporation, a request is sent to the advertising agency. The advertising agency sends the requested banner, but it also includes a cookie with the ID of the user.

FTP

File Transfer Protocol (FTP) is the standard protocol provided by TCP/IP for copying a file from one host to another. Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first. For example, two systems may use different file name conventions. Two systems may have different ways to represent data. Two systems may have different directory structures. All of these problems have been solved by FTP in a very simple and elegant approach. Although we can transfer files using HTTP, FTP is a better choice to transfer large files or to transfer files using different formats.



The client has three components:
the user interface,
the client control process,
and the client data transfer process.

The server has two components: the server control process and the server data transfer process.

The control connection is made between the control processes.

The data connection is made between the data transfer processes. Separation of commands and data transfer makes FTP more efficient.

The control connection uses very simple rules of communication. We need to transfer only a line of command or a line of response at a time.

The data connection, on the other hand, needs more complex rules due to the variety of data types transferred.

Two Connections

The two connections in FTP have different lifetimes.

The control connection remains connected during the entire interactive FTP session. The data connection is opened and then closed for each file transfer activity. It opens each time commands that involve transferring files are used, and it closes when the file is transferred. In other words, when a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred. FTP uses two well-known TCP ports: port 21 is used for the control connection, and port 20 is used for the data connection.

Control Connection

For control communication, FTP uses the same approach as TELNET (discussed later). It uses the NVT ASCII character

set as used by TELNET. Communication is achieved through commands and responses. This simple method is adequate for the control connection because we send one command (or response) at a time. Each line is terminated with a two-character (carriage return and line feed) end-of-line token. During this control connection, commands are sent from the client to the server and responses are sent from the server to the client. Commands, which are sent from the FTP client control process, are in the form of ASCII uppercase, which may or may not be followed by an argument.

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
ABOR		Abort the previous command
CDUP		Change to parent directory
CWD	Directory name	Change to another directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
MKD	Directory name	Create a new directory
PASS	User password	Password
PASV		Server chooses a port
PORT	Port identifier	Client chooses a port
PWD		Display name of current directory
QUIT		Log out of the system
RETR	File name(s)	Retrieve files; files are transferred from server to client
RMD	Directory name	Delete a directory
RNFR	File name (old)	Identify a file to be renamed
RNTO	File name (new)	Rename the file
STOR	File name(s)	Store files; file(s) are transferred from client to server
STRU	F, R, or P	Define data organization (F: file, R: record, or P: page)
TYPE	A, E, I	Default file type (A: ASCII, E: EBCDIC, I: image)
USER	User ID	User information
MODE	S, B, or C	Define transmission mode (S: stream, B: block, or C: compressed)

Every FTP command generates at least one response. A response has two parts: a three-digit number followed by text. The numeric part defines the code; the text part defines needed parameters or further explanation s. The first digit defines the status of the command. The second digit defines the area in which the status applies. The third digit provides additional information. The below table show some common responses.

<i>Code</i>	<i>Description</i>	<i>Code</i>	<i>Description</i>
125	Data connection open	250	Request file action OK
150	File status OK	331	User name OK; password is needed
200	Command OK	425	Cannot open data connection
220	Service ready	450	File action not taken; file not available
221	Service closing	452	Action aborted; insufficient storage
225	Data connection open	500	Syntax error; unrecognized command
226	Closing data connection	501	Syntax error in parameters or arguments
230	User login OK	530	User not logged in

File Type FTP can transfer one of the following file types across the data connection: ASCII file, EBCDIC file, or image file.

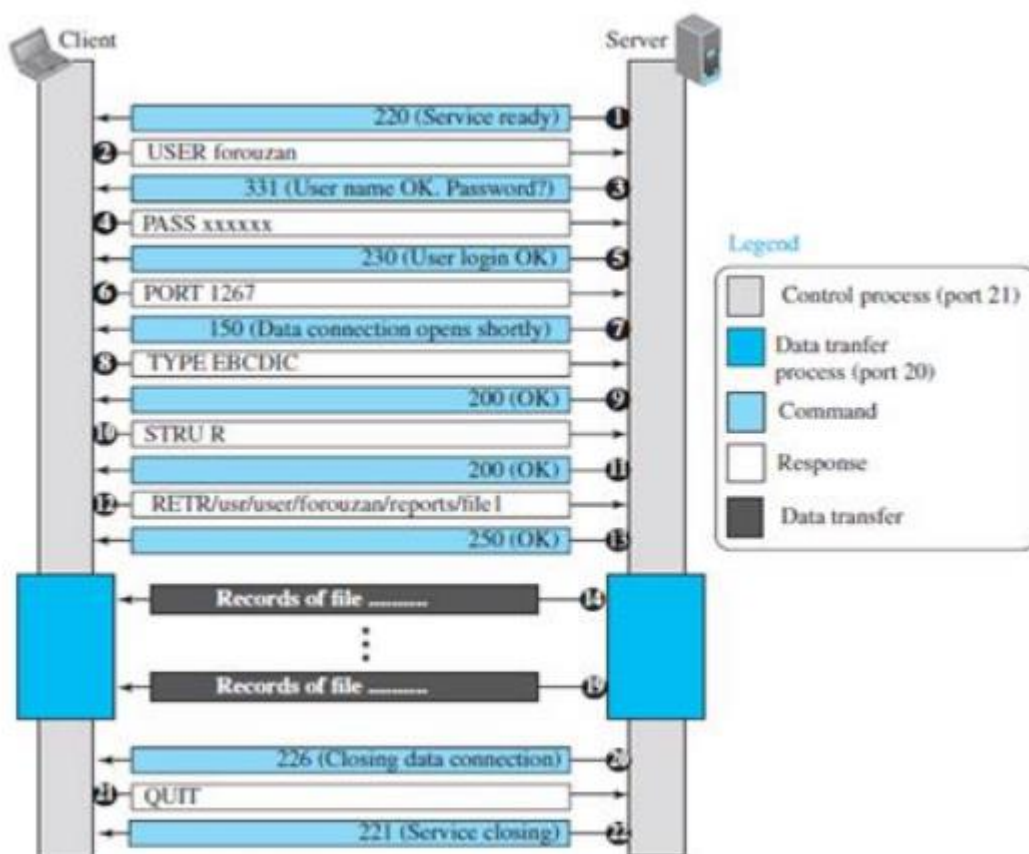
Data Structure FTP can transfer a file across the data connection using one of the following interpretations of the structure of the data: file structure, record structure, or page structure. The file structure format (used by default) has no structure. It is a continuous stream of bytes. In the record structure, the file is divided into records. This can be used only with text files. In the page structure, the file is divided into pages, with each page having a page number and a page header. The pages can be stored and accessed randomly or sequentially.

Transmission Mode

FTP can transfer a file across the data connection using one of the following three transmission modes: stream mode, block mode, or compressed mode. The stream mode is the default mode; data are delivered from FTP to TCP as a continuous stream of bytes. In the block mode, data can be delivered from FTP to TCP in blocks. In this case, each block is preceded by a 3-byte header. The first byte is called the block descriptor; the next two bytes define the size of the block in bytes.

File Transfer File transfer occurs over the data connection under the control of the commands sent over the control connection. However, we should remember that file transfer in FTP means one of three things: retrieving a file (server to client), storing a file (client to server), and directory listing (server to client). Figure below shows an example of using FTP for retrieving a file. The figure shows only one file to be transferred. The control connection remains open all the time, but the data connection is opened and closed repeatedly.

We assume the file is transferred in six sections. After all records have been transferred, the server control process announces that the file transfer is done. Since the client control process has no file to retrieve, it issues the QUIT command, which causes the service connection to be closed.



ELECTRONIC MAIL

* Discuss the working of Email in detail (13)

* Explain in detail how electronic mail application is carried out in a network. Also explain the protocols used in the

application. (13)

Electronic mail (or e-mail) allows users to exchange messages.

The nature of this application, however, is different from other applications discussed so far. In an application such as HTTP or FTP, the server program is running all the time, waiting for a request from a client.

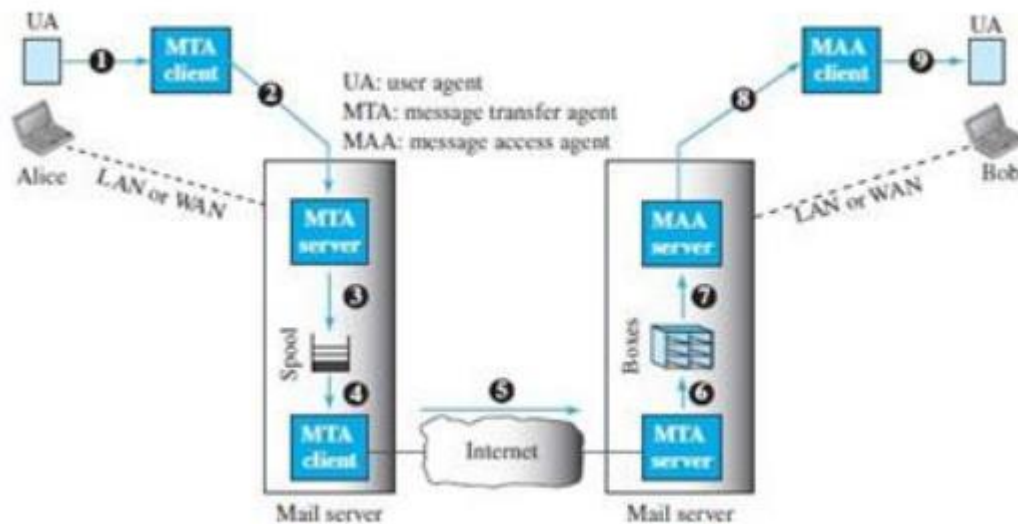
When the request arrives, the server provides the service. There is a request and there is a response. In the case of electronic mail, the situation is different. First, e-mail is considered a one-way transaction.

When Alice sends an email to Bob, she may expect a response, but this is not a mandate. Bob may or may not respond. If he does respond, it is another one-way transaction. Second, it is neither feasible nor logical for Bob to run a server program and wait until someone sends an e-mail to him.

Bob may turn off his computer when he is not using it. This means that the idea of client/server programming should be implemented in another way: using some intermediate computers (servers).

Architecture

To explain the architecture of e-mail, we give a common scenario, as shown in Figure below. Another possibility is the case in which Alice or Bob is directly connected to the corresponding mail server, in which LAN or WAN connection is not required.



In the common scenario, the sender and the receiver of the e-mail, Alice and Bob respectively, are connected via a LAN or a WAN to two mail servers. The administrator has created one mailbox for each user where the received messages are stored. A mailbox is part of a server hard drive, a special file with permission restrictions.

Only the owner of the mailbox has access to it. The administrator has also created a queue (spool) to store messages waiting to be sent. A simple e-mail from Alice to Bob takes nine different steps, as shown in the figure. Alice and Bob use three different agents: a user agent (UA), a message transfer agent (MTA), and a message access agent (MAA).

When Alice needs to send a message to Bob, she runs a UA program to prepare the message and send it to her mail server. The mail server at her site uses a queue (spool) to store messages waiting to be sent. The message, however, needs to be sent through the Internet from Alice's site to Bob's site using an MTA. The electronic mail system needs two UAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server).

User Agent

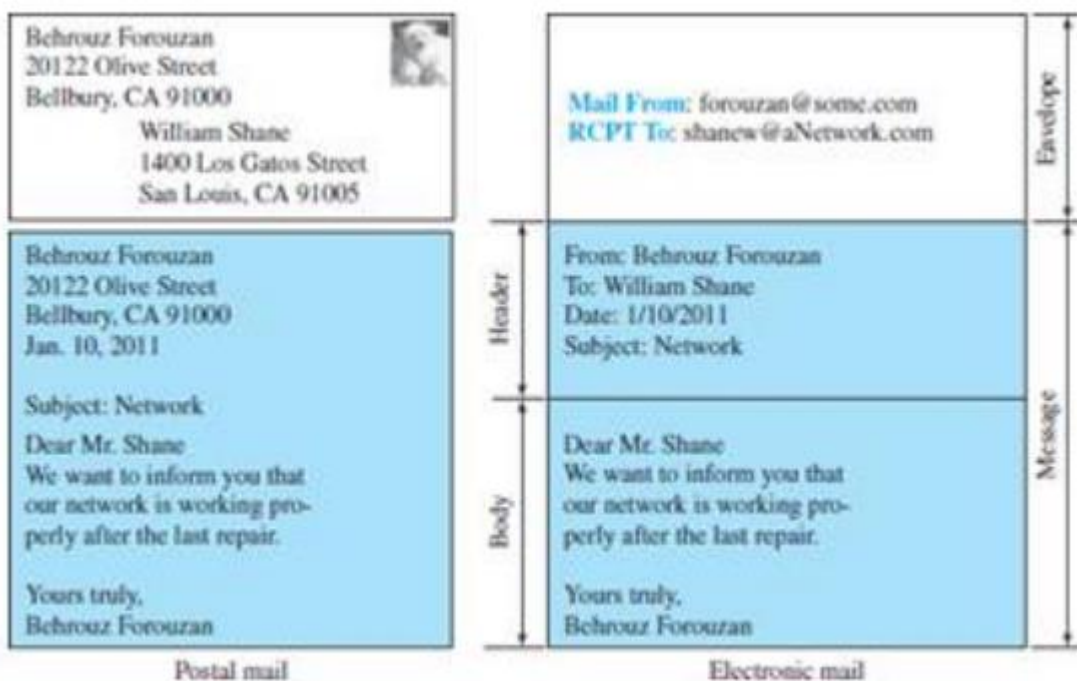
The first component of an electronic mail system is the user agent (UA). It provides service to the user to make the process of sending and receiving a message easier. A user agent is a software package (program) that composes, reads, replies to, and forwards messages. It also handles local mailboxes on the user computers.

There are two types of user agents: command-driven and GUI-based. Command driven user agents belong to the early days of electronic mail. They are still present as the underlying user agents. A command-driven user agent normally accepts a one character command from the keyboard to perform its task.

For example, a user can type the character r, at the command prompt, to reply to the sender of the message, or type the character R to reply to the sender and all recipients. Some examples of command driven user agents are mail, pine, and elm. Modern user agents are GUI-based. They contain graphical user interface (GUI) components that allow the user to interact with the software by using both the keyboard and the mouse. They have graphical components such as icons, menu bars, and windows that make the services easy to access. Some examples of GUI-based user agents are Eudora and Outlook.

Sending Mail

To send mail, the user, through the UA, creates mail that looks very similar to postal mail. It has an envelope and a message. The envelope usually contains the sender address, the receiver address, and other information. The message contains the header and the body. The header of the message defines the sender, the receiver, the subject of the message, and some other information. The body of the message contains the actual information to be read by the recipient. The format of mail is given below:



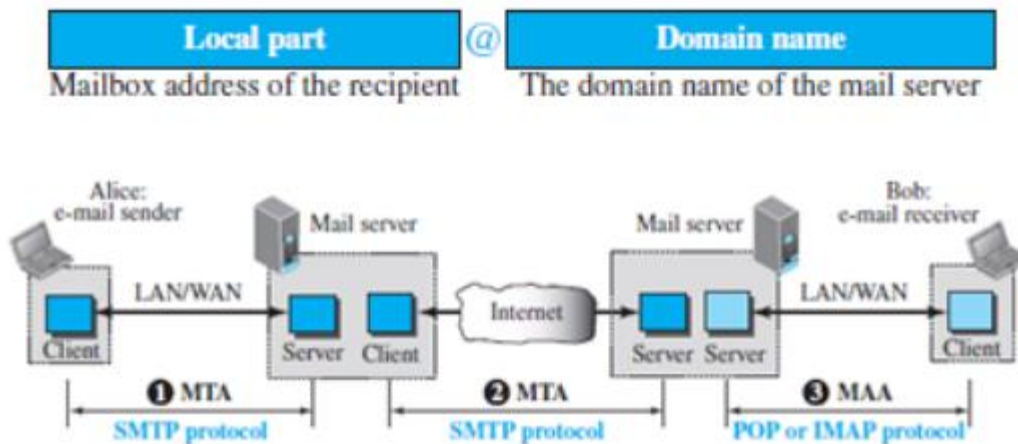
Receiving Mail

The user agent is triggered by the user (or a timer). If a user has mail, the UA informs the user with a notice. If the user is ready to read the mail, a list is displayed in which each line contains a summary of the information about a particular message in the mailbox.

The summary usually includes the sender mail address, the subject, and the time the mail was sent or received. The user can select any of the messages and display its contents on the screen.

Addresses

To deliver mail, a mail handling system must use an addressing system with unique addresses. In the Internet, the address consists of two parts: a local part and a domain name, separated by an @ sign.



Message Transfer Agent: SMTP

The formal protocol that defines the MTA client and server in the Internet is called Simple Mail Transfer Protocol (SMTP). SMTP is used two times, between the sender and the sender's mail server and between the two mail servers. As we will see shortly, another protocol is needed between the mail server and the receiver. SMTP simply defines how commands and responses must be sent back and forth.

Commands and Responses

SMTP uses commands and responses to transfer messages between an MTA client and an MTA server. The command is from an MTA client to an MTA server; the response is from an MTA server to the MTA client. Each command or reply is terminated by a two character (carriage return and line feed) end-of-line token.

Commands

Commands are sent from the client to the server. The format of a command is shown below: Keyword: argument(s) It consists of a keyword followed by zero or more arguments. SMTP defines 14 commands, listed in Table

Keyword	Argument(s)	Description
HELO	Sender's host name	Identifies itself
MAIL FROM	Sender of the message	Identifies the sender of the message
RCPT TO	Intended recipient	Identifies the recipient of the message
DATA	Body of the mail	Sends the actual message
QUIT		Terminates the message
RSET		Aborts the current mail transaction
VRIFY	Name of recipient	Verifies the address of the recipient
NOOP		Checks the status of the recipient
TURN		Switches the sender and the recipient
EXPN	Mailing list	Asks the recipient to expand the mailing list
HELP	Command name	Asks the recipient to send information about the command sent as the argument
SEND FROM	Intended recipient	Specifies that the mail be delivered only to the terminal of the recipient, and not to the mailbox
SMOL FROM	Intended recipient	Specifies that the mail be delivered to the terminal <i>or</i> the mailbox of the recipient
SMAL FROM	Intended recipient	Specifies that the mail be delivered to the terminal <i>and</i> the mailbox of the recipient

Responses Responses are sent from the server to the client. A response is a three-digit code that may be followed by additional textual information. Table below shows the most common response types

<i>Code</i>	<i>Description</i>
Positive Completion Reply	
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
Positive Intermediate Reply	
354	Start mail input
Transient Negative Completion Reply	
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted; insufficient storage
Permanent Negative Completion Reply	
500	Syntax error; unrecognized command

Mail Transfer Phases

The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination.

Connection Establishment

After a client has made a TCP connection to the well-known port 25, the SMTP server starts the connection phase. This phase involves the following three steps:

1. The server sends code 220 (service ready) to tell the client that it is ready to receive mail. If the server is not ready, it sends code 421 (service not available).
2. The client sends the HELO message to identify itself, using its domain name address. This step is necessary to inform the server of the domain name of the client.
3. The server responds with code 250 (request command completed) or some other code depending on the situation.

Message Transfer

After connection has been established between the SMTP client and server, a single message between a sender and one or more recipients can be exchanged. This phase involves eight steps. Steps 3 and 4 are repeated if there is more than one recipient.

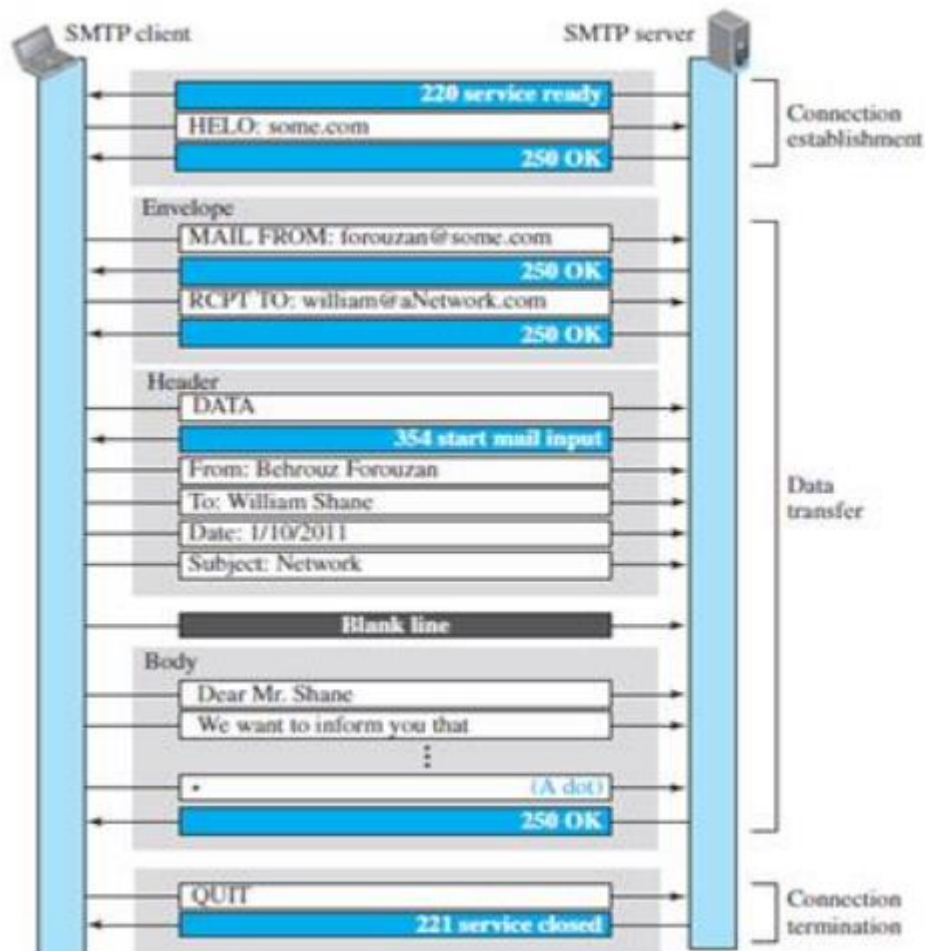
1. The client sends the MAIL FROM message to introduce the sender of the message. It includes the mail address of the sender (mailbox and the domain name). This step is needed to give the server the return mail address for returning errors and reporting messages.
2. The server responds with code 250 or some other appropriate code.
3. The client sends the RCPT TO (recipient) message, which includes the mail address of the recipient.
4. The server responds with code 250 or some other appropriate code.
5. The client sends the DATA message to initialize the message transfer.
6. The server responds with code 354 (start mail input) or some other appropriate message.
7. The client sends the contents of the message in consecutive lines. Each line is terminated by a two-character end-of-line token (carriage return and line feed). The message is terminated by a line containing just one period.
8. The server responds with code 250 (OK) or some other appropriate code.

Connection Termination

After the message is transferred successfully, the client terminates the connection.

This phase involves two steps.

1. The client sends the QUIT command.
2. The server responds with code 221 or some other appropriate code.



Message Access Agent:

POP and IMAP The first and second stages of mail delivery use SMTP. However, SMTP is not involved in the third stage because SMTP is a push protocol; it pushes the message from the client to the server. In other words, the direction of the bulk data (messages) is from the client to the server. On the other hand, the third stage needs a pull protocol; the client must pull messages from the server.

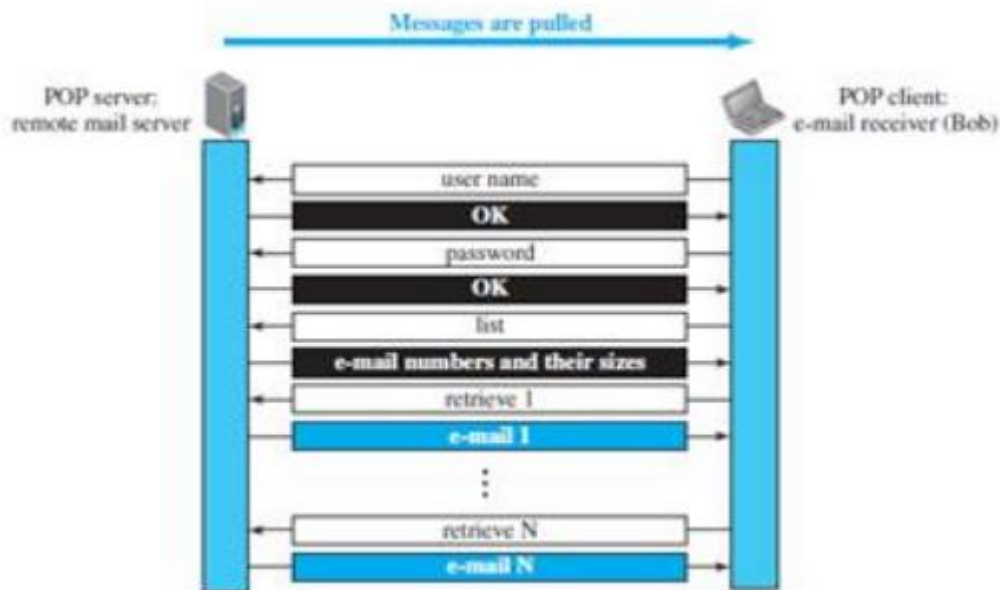
The direction of the bulk data is from the server to the client. The third stage uses a message access agent. Currently two message access protocols are available:

POST OFFICE PROTOCOL, VERSION 3 (POP3) AND INTERNET MAIL ACCESS PROTOCOL, VERSION 4 (IMAP4).

POP3

Post Office Protocol, version 3 (POP3) is simple but limited in functionality. The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server. Mail access starts with the client when the user needs to download its e-mail from the mailbox on the mail server. The client opens a connection to the server on TCP port 110.

It then sends its user name and password to access the mailbox. The user can then list and retrieve the mail messages, one by one.



POP3 has two modes: the delete mode and the keep mode. In the delete mode, the mail is deleted from the mailbox after each retrieval. In the keep mode, the mail remains in the mailbox after retrieval. The delete mode is normally used when the user is working at her permanent computer and can save and organize the received mail after reading or replying. The keep mode is normally used when the user accesses her mail away from her primary computer (for example, from a laptop). The mail is read but kept in the system for later retrieval and organizing.

IMAP4

Another mail access protocol is Internet Mail Access Protocol, version 4 (IMAP4). IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex. POP3 is deficient in several ways. It does not allow the user to organize her mail on the server; the user cannot have different folders on the server. In addition, POP3 does not allow the user to partially check the contents of the mail before downloading.

IMAP4 provides the following extra functions:

- A user can check the e-mail header prior to downloading.

- A user can search the contents of the e-mail for a specific string of characters prior to downloading.

- A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with highband width requirements.

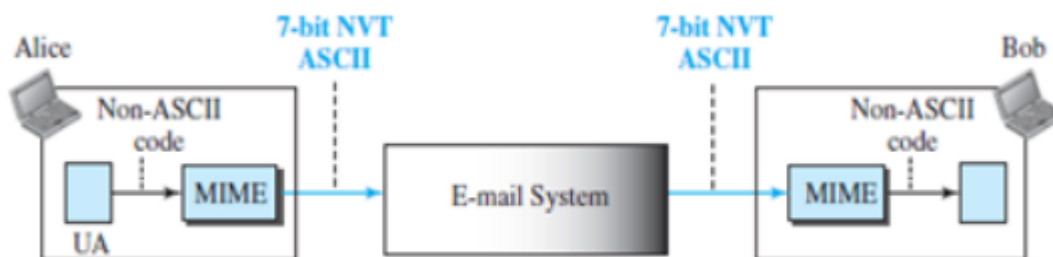
- A user can create, delete, or rename mailboxes on the mail server.

- A user can create a hierarchy of mailboxes in a folder for e-mail storage.

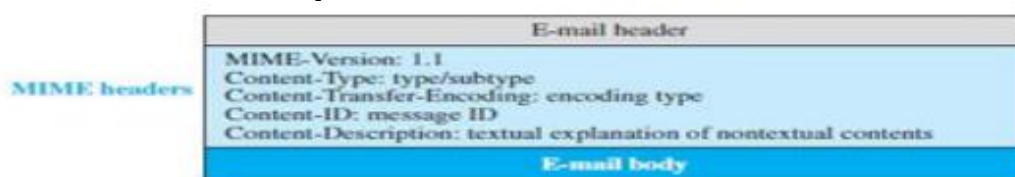
MIME

Electronic mail has a simple structure. Its simplicity, however, comes with a price. It can send messages only in NVT 7-bit ASCII format. In other words, it has some limitations. It cannot be used for languages other than English (such as French, German, Hebrew, Russian, Chinese, and Japanese). Also, it cannot be used to send binary files or video or audio data.

Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that allows non-ASCII data to be sent through e-mail. MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers it to the client MTA to be sent through the Internet. The message at the receiving site is transformed back to the original data. We can think of MIME as a set of software functions that transforms non-ASCII data to ASCII data and vice versa, as shown in Figure.



MIME Headers MIME defines five headers, as shown in Figure which can be added to the original email header section to define the transformation parameters.



MIME-Version This header defines the version of MIME used. The current version is 1.1.

Content-Type This header defines the type of data used in the body of the message. The content type and the content subtype are separated by a slash. Depending on the subtype, the header may contain other parameters. MIME allows seven different types of data, listed in Table

Type	Subtype	Description
Text	Plain	Unformatted
	HTML	HTML format (see Appendix C)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to Mixed, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

Content-Transfer-Encoding This header defines the method used to encode the messages into 0s and 1s for transport. The five types of encoding methods are listed in Table. The last two encoding methods are interesting. In the Base64 encoding, data, as a string of bits, is first divided into 6-bit chunks as shown in Figure

Type	Description
7-bit	NVT ASCII characters with each line less than 1000 characters
8-bit	Non-ASCII characters with each line less than 1000 characters
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equal sign plus an ASCII code

DOMAIN NAME SYSTEM (DNS)

* Briefly explain the Domain Name Service protocol with an example (13)

The last client-server application program we discuss has been designed to help other application programs. To identify an entity, TCP/IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet. However, people prefer to use names instead of numeric addresses.

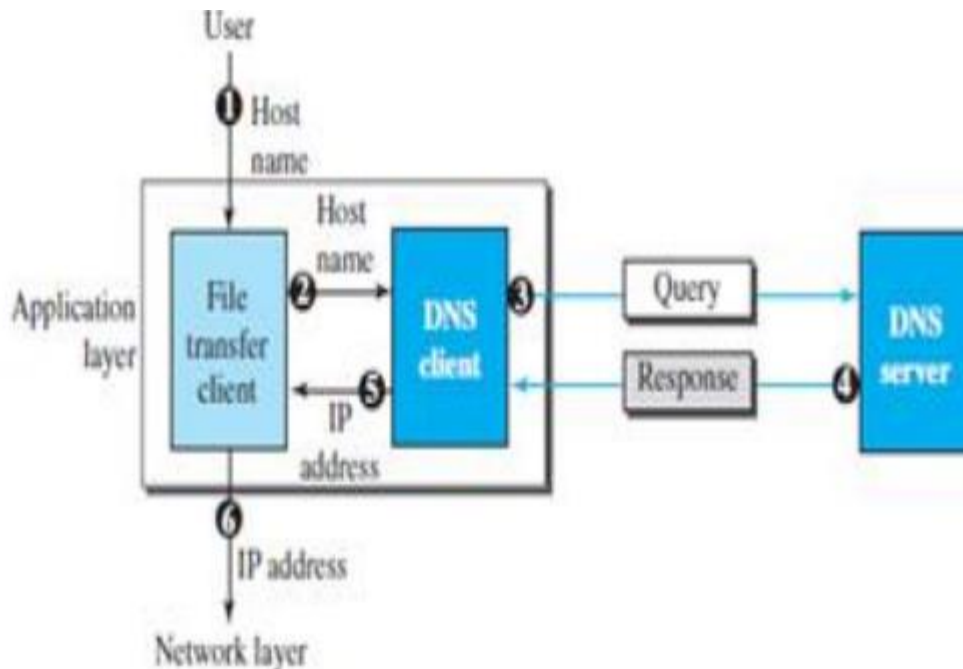
Therefore, the Internet needs to have a directory system that can map a name to an address. This is analogous to the telephone network. A telephone network is designed to use telephone numbers, not names. People can either keep a private file to map a name to the corresponding telephone number or can call the telephone directory to do so. We discuss how this directory system in the Internet can map names to IP addresses. Since the Internet is so huge today, a central directory system cannot hold all the mapping. In addition, if the central computer fails, the whole communication network will collapse.

A better solution is to distribute the information among many computers in the world. In this method, the host that needs mapping can contact the closest computer holding the needed information. This method is used by the Domain Name System (DNS).

Figure below shows how TCP/IP uses a DNS client and a DNS server to map a name to an address. A user wants to use a file transfer client to access the corresponding file transfer server running on a remote host. The user knows only the file transfer server name, such as afileservice.com. However, the TCP/IP suite needs the IP address of the file transfer server to make the connection.

The following six steps map the host name to an IP address:

1. The user passes the host name to the file transfer client.
2. The file transfer client passes the host name to the DNS client.
3. Each computer, after being booted, knows the address of one DNS server. The DNS client sends a message to a DNS server with a query that gives the file transfer server name using the known IP address of the DNS server.
4. The DNS server responds with the IP address of the desired file transfer server.
5. The DNS server passes the IP address to the file transfer client.
6. The file transfer client now uses the received IP address to access the file transfer server



Name Space

To be unambiguous, the names assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses. In other words, the names must be unique because the addresses are unique.

A name space that maps each address to a unique name can be organized in two ways: flat or hierarchical.

In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure. The names may or may not have a common section; if they do, it has no meaning.

The main disadvantage of a flat name space is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication. In a hierarchical name space, each name is made of several parts.

The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization, and so on. In this case, the authority to assign and control the name spaces can be decentralized.

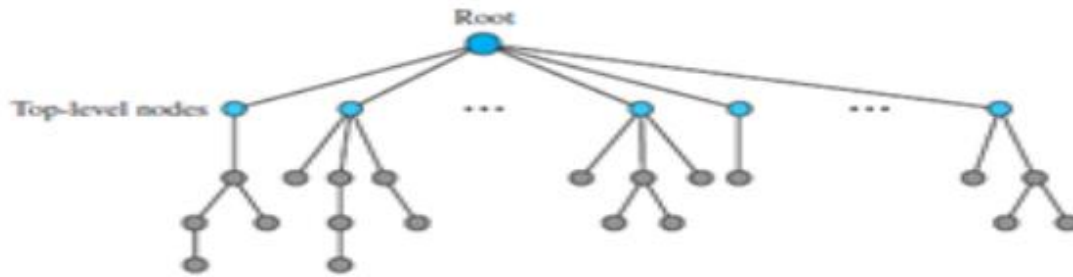
A central authority can assign the part of the name that defines the nature of the organization and the name of the organization.

The responsibility for the rest of the name can be given to the organization itself. The organization can add suffixes (or prefixes) to the name to define its host or resources. The management of the organization need not worry that the prefix chosen for a host is taken by another organization because, even if part of an address is the same, the whole address is different.

For example, assume two organizations call one of their computers caesar. The first organization is given a name by the central authority, such as first.com, the second organization is given the name second.com. When each of these organizations adds the name caesar to the name they have already been given, the end result is two distinguishable names: caesar.first.com and caesar.second.com. The names are unique.

Domain Name Space

To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127



Label

Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string (empty string). DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

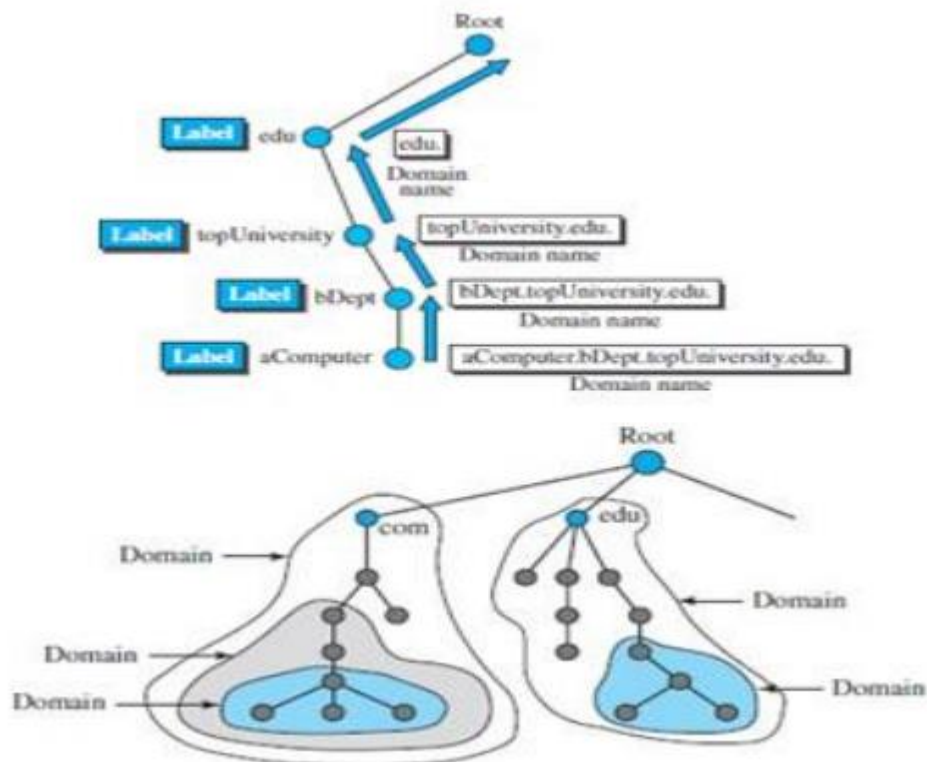
Domain Name

Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.). The domain names are always read from the node up to the root. The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing. If a label is terminated by a null string, it is called a fully qualified domain name (FQDN).

The name must end with a null label, but because null means nothing, the label ends with a dot. If a label is not terminated by a null string, it is called a partially qualified domain name (PQDN). A PQDN starts from a node, but it does not reach the root. It is used when the name to be resolved belongs to the same site as the client. Here the resolver can supply the missing part, called the suffix, to create an FQDN.

Domain

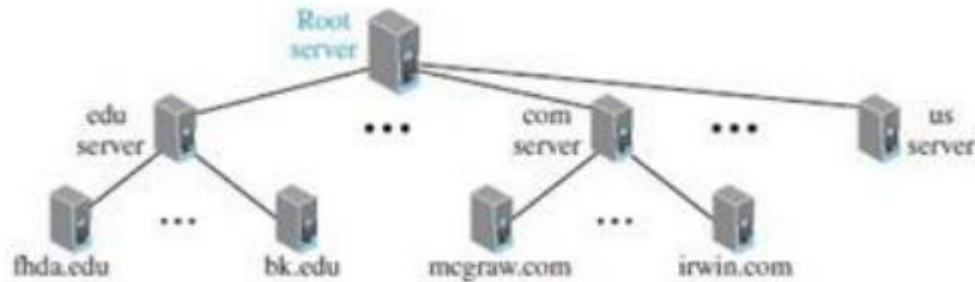
A domain is a subtree of the domain name space. The name of the domain is the name of the node at the top of the subtree



Hierarchy of Name Servers

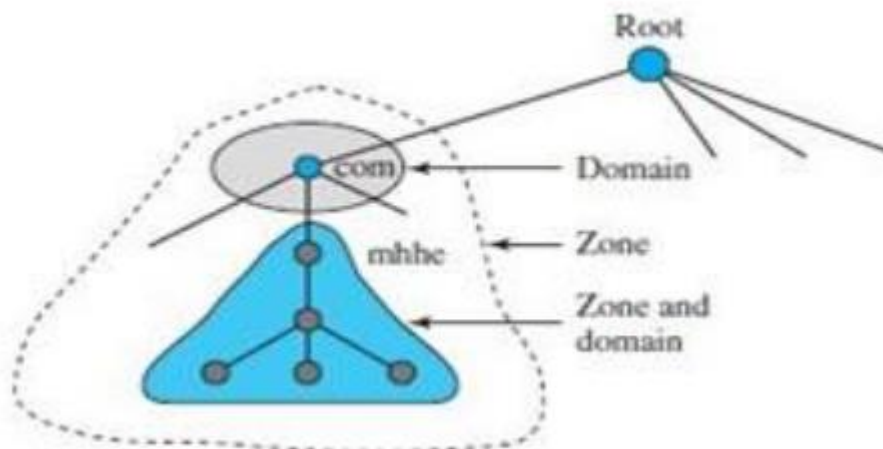
The solution to these problems is to distribute the information among many computers called DNS servers. One way to do this is to divide the whole space into many domains based on the first level.

In other words, we let the root stand alone and create as many domains (subtrees) as there are first-level nodes. Because a domain created this way could be very large, DNS allows domains to be divided further into smaller domains (subdomains). Each server can be responsible (authoritative) for either a large or small domain.



Zone

Since the complete domain name hierarchy cannot be stored on a single server, it is divided among many servers. What a server is responsible for or has authority over is called a zone. We can define a zone as a contiguous part of the entire tree. If a server accepts responsibility for a domain and does not divide the domain into smaller domains, the “domain” and the “zone” refer to the same thing. The server makes a database called a zone file and keeps all the information for every node under that domain



Root Server

A root server is a server whose zone consists of the whole tree. A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers. There are several root servers, each covering the whole domain name space. The root servers are distributed all around the world.

Primary and Secondary Servers

DNS defines two types of servers: primary and secondary. A primary server is a server that stores a file about the zone for which it is an authority.

It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk. A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files. If

updating is required, it must be done by the primary server, which sends the updated version to the secondary.

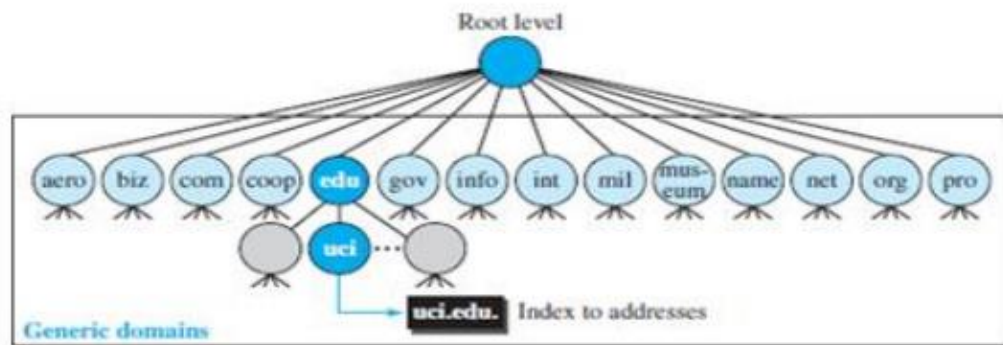
The primary and secondary servers are both authoritative for the zones they serve. The idea is not to put the secondary server at a lower level of authority but to create redundancy for the data so that if one server fails, the other can continue serving clients. Note also that a server can be a primary server for a specific zone and a secondary server for another zone.

DNS in the Internet

DNS is a protocol that can be used in different platforms. In the Internet, the domain name space (tree) was originally divided into three different sections: generic domains, country domains, and the inverse domains. However, due to the rapid growth of the Internet, it became extremely difficult to keep track of the inverse domains, which could be used to find the name of a host when given the IP address. The inverse domains are now deprecated (see RFC 3425). We, therefore, concentrate on the first two.

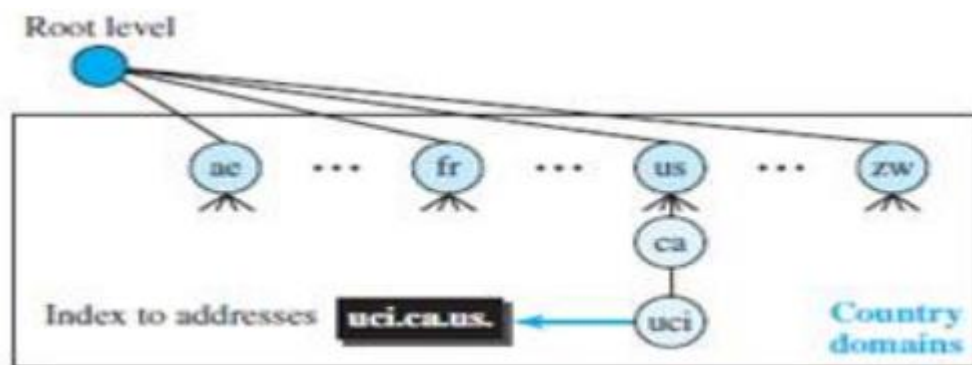
Generic Domains

The generic domains define registered hosts according to their generic behavior. Each node in the tree defines a domain, which is an index to the domain name space database.



Country Domains

The country domains section uses two-character country abbreviations (e.g., us for United States). Second labels can be organizational, or they can be more specific national designations. The United States, for example, uses state abbreviations as a subdivision of us (e.g., ca.us.).



DNS Messages

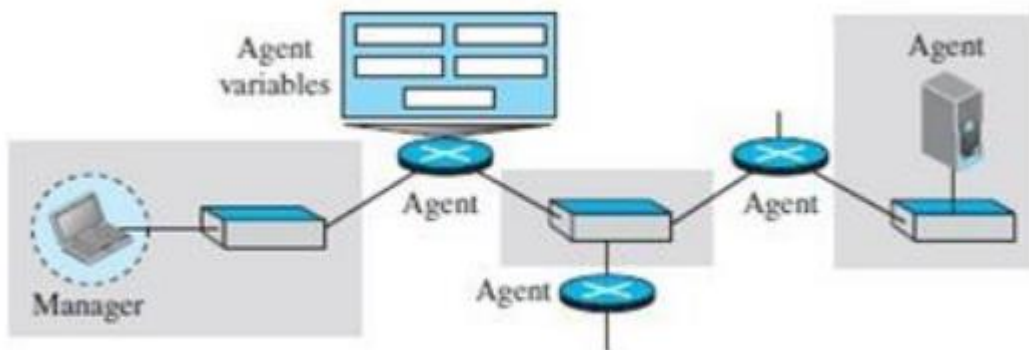
To retrieve information about hosts, DNS uses two types of messages: query and response. Both types have the same format as shown in Figure

SNMP

- Brief about the importance of Simple Network Management Protocol (13)

Several network management standards have been devised during the last few decades. The most important one is Simple Network Management Protocol (SNMP), used by the Internet. We discuss this standard in this section. SNMP is a framework for managing devices in an internet using the TCP/IP protocol suite.

It provides a set of fundamental operations for monitoring and maintaining an internet. SNMP uses the concept of manager and agent. That is, a manager, usually a host, controls and monitors a set of agents, usually routers or servers



SNMP is an application-level protocol in which a few manager stations control a set of agents. The protocol is designed at the application level so that it can monitor devices made by different manufacturers and installed on different physical networks. In other words, SNMP frees management tasks from both the physical characteristics of the managed devices and the underlying networking technology.

Managers and Agents

A management station, called a manager, is a host that runs the SNMP client program. A managed station, called an agent, is a router (or a host) that runs the SNMP server program. Management is achieved through simple interaction between a manager and an agent.

The agent keeps performance information in a database. The manager has access to the values in the database. For example, a router can store in appropriate variables the number of packets received and forwarded. The manager can fetch and compare the values of these two variables to see if the router is congested or not. The manager can also make the router perform certain actions.

For example, a router periodically checks the value of a reboot counter to see when it should reboot itself. It reboots itself, for example, if the value of the counter is 0. The manager can use this feature to reboot the agent remotely at any time. It simply sends a packet to force a 0 value in the counter. Agents can also contribute to the management process.

The server program running on the agent can check the environment and, if it notices something unusual, it can send a warning message (called a Trap) to the manager. In other words, management with SNMP is based on three basic ideas:

1. A manager checks an agent by requesting information that reflects the behavior of the agent.
2. A manager forces an agent to perform a task by resetting values in the agent database.
3. An agent contributes to the management process by warning the manager of an unusual situation.

Management Components

To do management tasks, SNMP uses two other protocols: Structure of Management Information (SMI) and Management Information Base (MIB). In other words, management on the Internet is done through the cooperation of three protocols: SNMP, SMI, and MIB, as shown in Figure.



Role of SNMP

SNMP has some very specific roles in network management. It defines the format of the packet to be sent from a manager to an agent and vice versa. It also interprets the result and creates statistics (often with the help of other management software). The packets exchanged contain the object (variable) names and their status (values). SNMP is responsible for reading and changing these values.

Role of SMI

To use SNMP, we need rules for naming objects. This is particularly important because the objects in SNMP form a hierarchical structure (an object may have a parent object and some child objects). Part of a name can be inherited from the parent. SMI defines the general rules for naming objects, defining object types (including range and length), and showing how to encode objects and values.

Role of MIB

We hope it is clear that we need another protocol. For each entity to be managed, this protocol must define the number of objects, name them according to the rules defined by SMI, and associate a type to each named object. This protocol is MIB. MIB creates a set of objects defined for each entity in a manner similar to that of a database. MIB creates a collection of named objects, their types, and their relationships to each other in an entity to be managed.

SMI

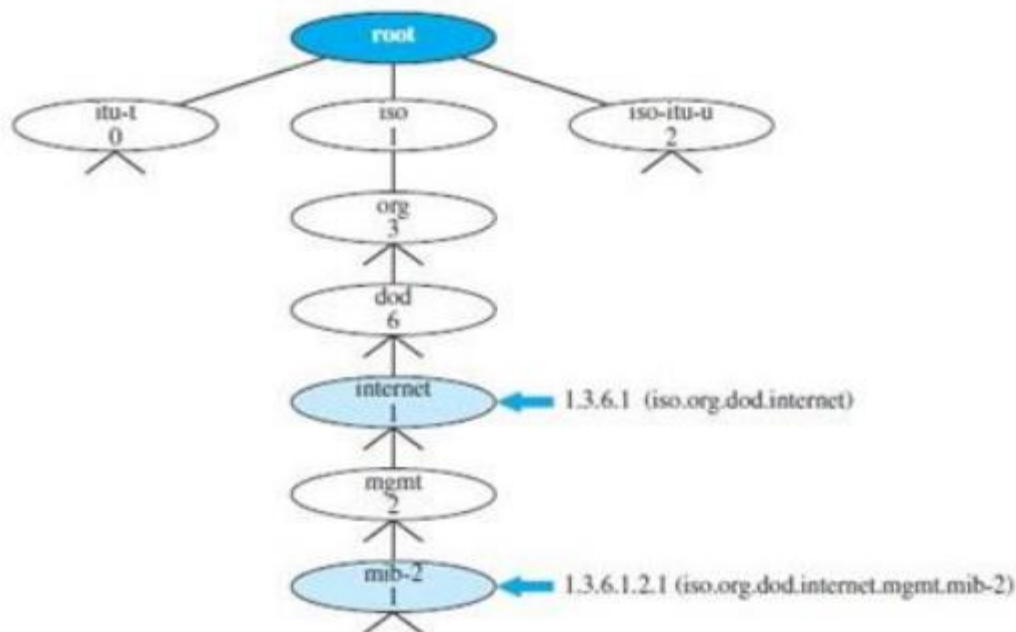
The Structure of Management Information, version 2 (SMIv2) is a component for network management. SMI is a guideline for SNMP. It emphasizes three attributes to handle an object: name, data type, and encoding method. Its functions are:

- ☐ To name objects.
- ☐ To define the type of data that can be stored in an object.
- ☐ To show how to encode data for transmission over the network.

Name

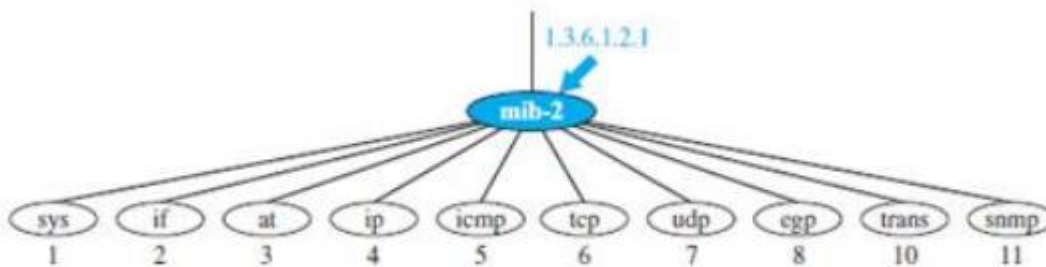
SMI requires that each managed object (such as a router, a variable in a router, a value, etc.) have a unique name.

To name objects globally, SMI uses an object identifier, which is a hierarchical identifier based on a tree structure. The objects that are used in SNMP are located under the mib-2 object, so their identifiers always start with 1.3.6.1.2.1.



MIB

The Management Information Base, version 2 (MIB2) is the second component used in network management. Each agent has its own MIB2, which is a collection of all the objects that the manager can manage.



The objects in MIB2 are categorized under several groups: system, interface, address translation, ip, icmp, tcp, udp, egp, transmission, and snmp (note that group 9 is deprecated). These groups are under the mib-2 object in the object identifier tree. Each group has defined variables and/or tables.

The following is a brief description of some of the objects:

- ☐ sys This object (system) defines general information about the node (system), such as the name, location, and lifetime.
- ☐ if This object (interface) defines information about all of the interfaces of the node including interface number, physical address, and IP address.
- ☐ at This object (address translation) defines the information about the ARP table.
- ☐ ip This object defines information related to IP, such as the routing table and the IP address.
- ☐ icmp This object defines information related to ICMP, such as the number of packets sent and received and total errors created.
- ☐ tcp This object defines general information related to TCP, such as the connection table, time-out value, number of ports, and number of packets sent and received.
- ☐ udp This object defines general information related to UDP, such as the number of ports and number of packets sent

and received.

- ❑ **egp** These objects are related to the operation of EGP.
- ❑ **trans** These objects are related to the specific method of transmission (future use).
- ❑ **snmp** This object defines general information related to SNMP itself.

SNMP uses both SMI and MIB in Internet network management. It is an application program that allows:

- ❑ A manager to retrieve the value of an object defined in an agent.
- ❑ A manager to store a value in an object defined in an agent.
- ❑ An agent to send an alarm message about an abnormal situation to the manager.

Messages

SNMP does not send only PDUs, it embeds each PDU in a message. A message is made of a message header followed by the corresponding PDU, as shown in Figure

