UNIT I INTRODUCTION AND APPLICATION LAYER

Data 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

1.1 Data Communication

When we communicate, we are sharing information. This sharing can be ***local or remote***.

The term telecommunication, which includes telephony, telegraphy, and television, means communication at a distance (tele is Greek for "far").

The word data refers to information presented in whatever form is agreed upon by the parties creating and using the data.

"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:***

I. Delivery: The system must deliver data to the correct destination. Data must be received by the intended device or user .

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

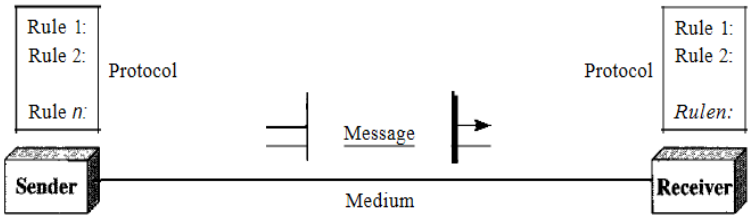
3. Timeliness: The system must deliver data in a timely manner. Data delivered late are useless.

4. Jitter: Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets.

1.1.1 Components

A data communications system has five components

Figure 1.1 *Five components ofdata communication*



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

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

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

4. 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.

5. 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.

1.1.2 Data Representation

Information today comes in different forms such as text, numbers, images, audio, and video.

Text

In data communications, text is represented as a bit pattern, a sequence of bits (O s or 1 s). Different sets of bit patterns have been designed to represent text symbols. Each set is called a code, and the process of representing symbols is called coding.

Unicode, which uses 32 bits to represent a symbol or character used in any language in the world. The American Standard Code for Information Interchange (ASCII), now constitutes the first 127 characters in Unicode and is also referred to as Basic Latin.

Numbers

Numbers are also represented by bit patterns. However, a code such as ASCII is not used to represent numbers; the number is directly converted to a binary number to simplify mathematical operations.

Images

Images are also represented by bit patterns. In its simplest form, an image is composed of a matrix of pixels (picture elements), where each pixel is a small dot. The size of the pixel depends on the resolution. The size and the value of the pattern depend on the image.

For an image made of only black-and-white dots (e.g., a chessboard), a I-bit pattern is enough to represent a pixel. If an image is not made of pure white and pure black pixels, you can increase the size of the bit pattern to include gray scale

There are several methods to represent color images. One method is called RGB, so called because each color is made of a combination of three primary colors: red, green, and blue. The intensity of each color is measured, and a bit pattern is assigned to it. Another method is called YCM, in which a color is made of a combination of three other primary colors: yellow, cyan, and magenta.

Audio

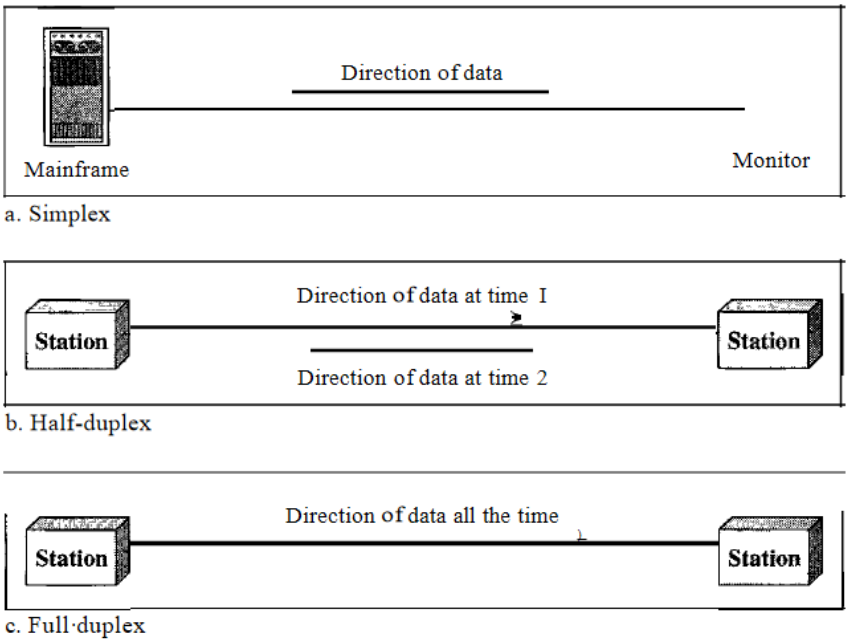
Audio refers to the recording or broadcasting of sound or music. Audio is by nature different from text, numbers, or images.

Video
Video refers to the recording or broadcasting of a picture or movie. Video can either be produced as a continuous entity (e.g., by a TV camera), or it can be a combination of images, each a discrete entity, arranged to convey the idea of motion.

1.1.3 Data Flow / transmission mode

Communication between two devices can be simplex, half-duplex, or full-duplex as shown in Figure 1.2.

Figure 1.2 *Data flow (simplex, half-duplex, and full-duplex)*



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 (see Figure 1.2a). Keyboards and traditional monitors are examples of simplex devices.

Advantage of Simplex mode:

- In simplex mode, the station can utilize the entire bandwidth of the communication channel, so that more data can be transmitted at a time.

Disadvantage of Simplex mode:

- Communication is unidirectional, so it has no inter-communication between devices.

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 (see Figure 1.2b)

Walkie-talkies and CB (citizens band) radios are both half-duplex systems. The half-duplex mode is used in cases where there is no need for communication in both directions at the same time.

Advantage of Half-duplex mode:

- In half-duplex mode, both the devices can send and receive the data and also can utilize the entire bandwidth of the communication channel during the transmission of data.

Disadvantage of Half-Duplex mode:

- In half-duplex mode, when one device is sending the data, then another has towait, this causes the delay in sending the data at the right time.

Full-Duplex

In full-duplex mode (also called duplex), both stations can transmit and receive simultaneously (see Figure 1.2c).The full-duplex mode is like a two-way street with traffic flowing in both directions at the same time. One common example of full-duplex communication is the telephone network. When two people are communicating by a telephone line, both can talk and listen at the same time. The full-duplex mode is used when communication in both directions is required all the time.

Advantage of Full-duplex mode:

- Both the stations can send and receive the data at the same time.

Disadvantage of Full-duplex mode:

- If there is no dedicated path exists between the devices, then the capacity of the communication channel is divided into two parts.

1.2 NETWORKS

A network is a set of devices (often referred to as nodes) connected by communication links. A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.

Distributed Processing

Most networks use distributed processing, in which a task is divided among multiple computers. Instead of one single large machine being responsible for all aspects of a process, separate computers (usually a personal computer or workstation) handle a subset.

1.2.1 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

Performance can be measured in many ways, including transit time and response time. ***Transit time*** is the amount of time required for a message to travel from one device to another. ***Response time*** is the elapsed time between an inquiry and a response. The performance of a network depends on a number of factors, including the number of users,

the type of transmission medium, the capabilities of the connected hardware, and the efficiency of the software. Performance is often evaluated by two networking metrics: ***throughput and delay***. Throughput is an actual measurement of how fast data can be transmitted. Latency/delay is time required for a message to completely arrive at the destination from source. We often need more throughput and less delay. However, these two criteria are often contradictory. If we try to send more data to the network, we may increase throughput but we increase the delay because of traffic congestion in the network.

Reliability

In addition to accuracy of delivery, network reliability is measured by the frequency of failure, the time it takes a link to recover from a failure, and the network's robustness in a catastrophe.

Security

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.

1.2.2 Physical Structures

Type of Connection / Line configuration

A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another. There are two possible types of connections: point-to-point and multipoint.

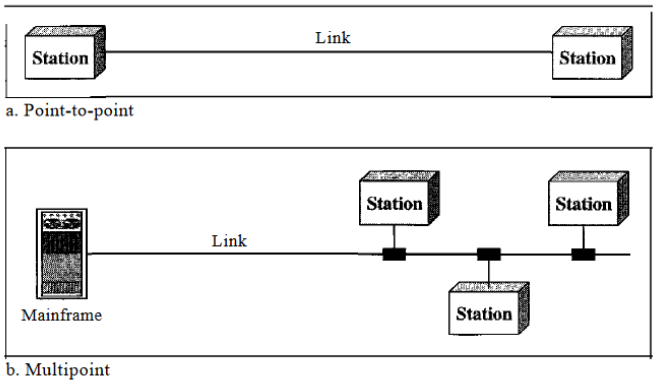
Point-to-Point

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. Most point-to-point connections use an actual length of wire or cable to connect the two ends (see Figure 1.3a). When you change television channels by infrared remote control, you are establishing a point-to-point connection between the remote control and the television's control system.

Multipoint

A multipoint (also called multidrop) connection is one in which more than two specific devices share a single link (see Figure 1.3b). In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a spatially shared connection. If users must take turns, it is a timeshared connection.

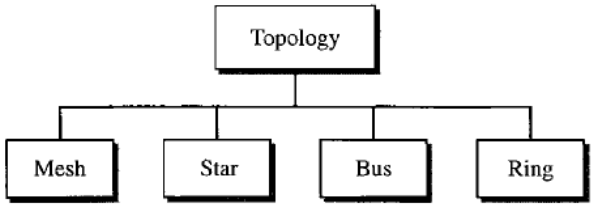
Figure 1.3 *Types of connections: point-to-point and multipoint*



Physical Topology

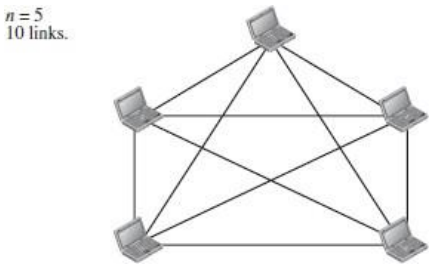
The term physical topology refers to the way in which a network is laid out physically. Two or more devices connect to a link; two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another. There are four basic topologies possible: mesh, star, bus, and ring (see Figure 1.4).

Figure **1.4** *Categories oftopology*



Mesh Topology

- In a mesh topology, every device has a dedicated point-to-point link to everyother device.
- The term dedicated means that the link carries traffic only between the twodevices it connects.
- The number of physical links in a fully connected mesh network with *n* nodes isgiven by *n* (*n* – 1) / 2.

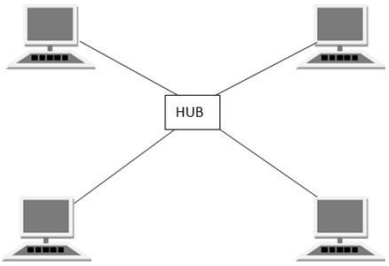


Advantages of Mesh Topology	Disadvantages of Mesh Topology
1. Each connection can carry its own data load. 2. It is robust. 3. Fault is diagnosed easily. 4. Provides security and privacy.	1. Installation and configuration is difficult. 2. Cabling cost is more. 3. Bulk wiring is required.

Star Topology

- In a star topology, each device has a dedicated point-to-point link only to acentral controller, usually called a hub.
- The devices are not directly linked to one another.

- The controller/hub acts as an exchange.
- If one device wants to send data to another, it sends the data to the controller/hub ,which then relays the data to the other connected device.



Advantages of Star Topology	***Disadvantages of Star Topology***
1. Fast performance with few nodes and low network traffic. 2. Hub can be upgraded easily. 3. Easy to troubleshoot. 4. Easy to setup and modify. 5. Only that node is affected which has failed, rest of the nodes can work smoothly	1. Cost of installation is high. 2. Expensive to use. 3. If the hub fails, then the wholenetwork is stopped. 4. Performance is based on the hub thatis it depends on its capacity

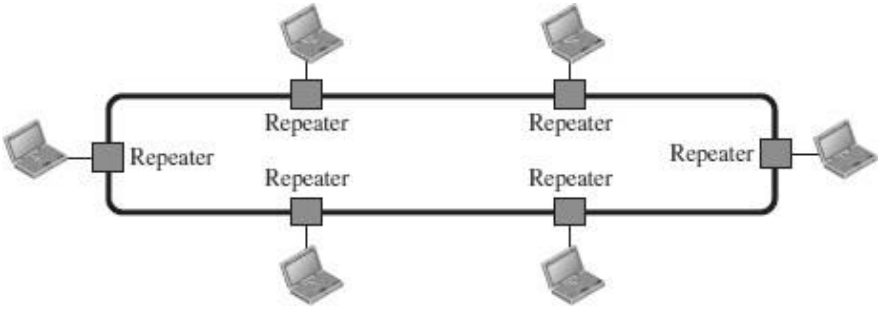
Bus Topology

- Bus topology is a network type in which every computer and network device isconnected to single cable.
- The long single cable acts as a backbone to link all the devices in a network.
- When it has exactly two endpoints, then it is called Linear Bus topology.
- It transmits data only in one direction.

Advantages of Bus Topology	**Disadvantages of Bus Topology**
1. It is cost effective. 2. Cable required is least compared to other network topology. 3. Used in small networks. 4. It is easy to understand. 5. Easy to expand joining two cables together	1. Cables fails then whole network fails. 2. If network traffic is heavy or nodes are more, the performance of the network decreases. 3. Cable has a limited length. 4. It is slower than the ring topology.

Ring Topology

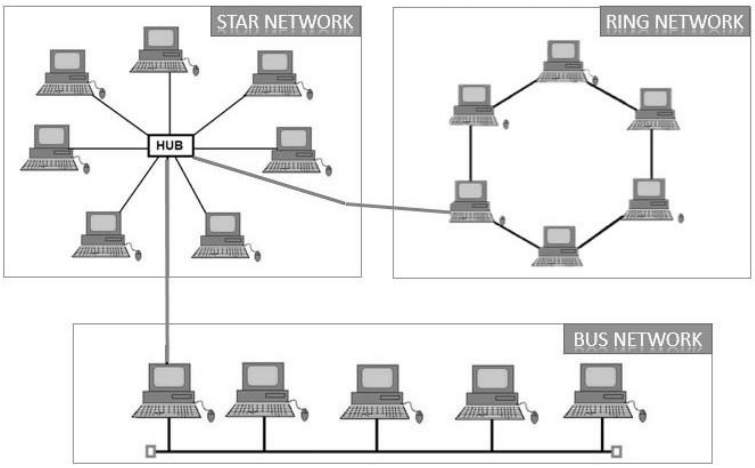
- In a ring topology, each device has a dedicated point-to-point connection withonly the two devices on either side of it.
- A signal is passed along the ring in one direction, from device to device, until itreaches its destination.
- Each device in the ring incorporates a repeater.
- When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along.



Advantages of Ring Topology	Disadvantages of Ring Topology
1. Transmitting network is not affected by high traffic or by adding more nodes, as only the nodes having tokens can transmit data. 2. Cheap to install and expand	1. Troubleshooting is difficult in ring topology. 2. Adding or deleting the computers disturbs the network activity. 3. Failure of one computer disturbs the whole network

Hybrid Topology

- Hybrid Topology is a combination of one or more basic topologies.
- For example if one department in an office uses ring topology, the other departments uses star and bus topology, then connecting these topologies will result in Hybrid Topology.
- Hybrid Topology inherits the advantages and disadvantages of the topologies included.



Advantages of Hybrid Topology	Disadvantages of Hybrid Topology
1. Reliable as Error detecting and trouble shooting is easy. 2. Effective. 3. Scalable as size can be increased easily. 4. Flexible.	1. Complex in design. 2. Costly

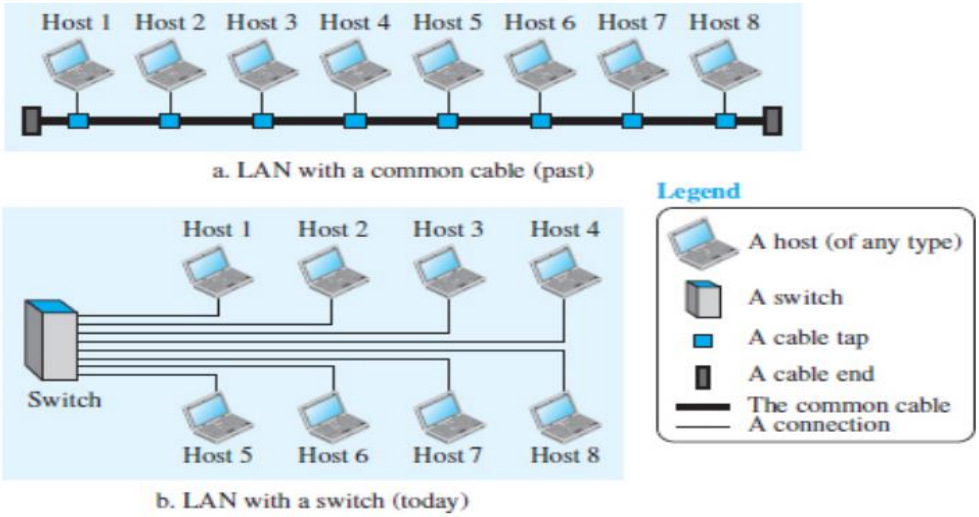
1.3 NETWORK TYPES

Different types of networks: LANs MANs and WANs.

1.3.1 Local Area Network

- Local Area Network is a group of computers connected to each other in a small area such as building, office.
- LAN is used for connecting two or more personal computers through a communication medium such as twisted pair, coaxial cable, etc.
- It is less costly as it is built with inexpensive hardware such as hubs, networkadapters, and ethernet cables.
- The data is transferred at an extremely faster rate in Local Area Network.
- LAN can be connected using a common cable or a Switch

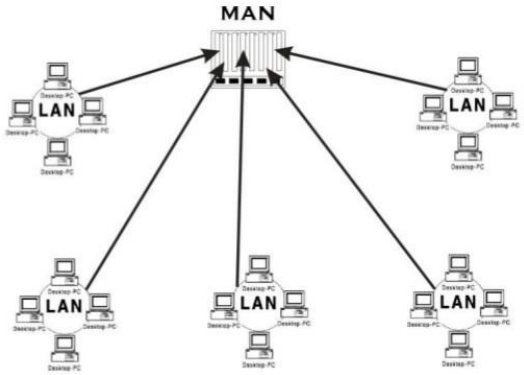
Figure 1.8 *An isolated LAN in the past and today*



Advantages of LAN	Disadvantages of LAN
• Resource Sharing • Software Applications Sharing. • Easy and Cheap Communication • Centralized Data. • Data Security • Internet Sharing	• High Setup Cost • Privacy Violations • Data Security Threat • LAN Maintenance Job • Covers Limited Area

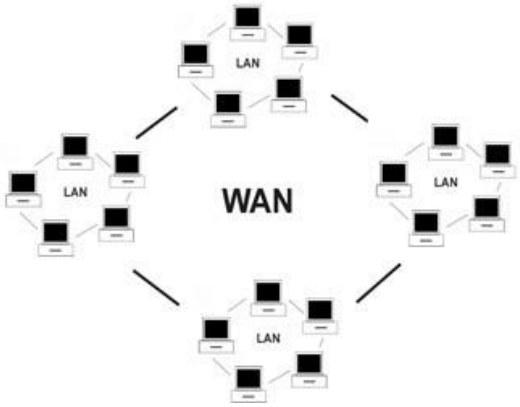
1.3.2 Metropolitan Area Network (MAN)

- A metropolitan area network is a network that covers a larger geographic area by interconnecting a different LAN to form a larger network.
- It generally covers towns and cities (50 km)
- In MAN, various LANs are connected to each other through a telephone exchange line.
- Communication medium used for MAN are optical fibers, cables etc.
- It has a higher range than Local Area Network(LAN).It is adequate for distributed computing applications.



1.3.3 Wide Area Network (WAN)

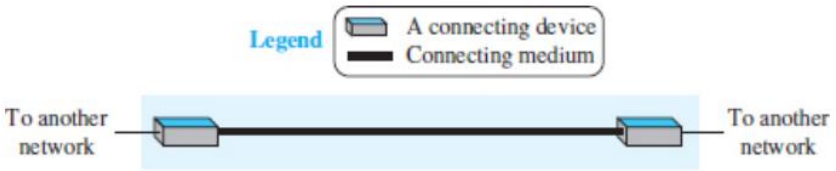
- A Wide Area Network is a network that extends over a large geographical areasuch as states or countries.
- A Wide Area Network is quite bigger network than the LAN.
- A Wide Area Network is not limited to a single location, but it spans over a largegeographical area through a telephone line, fibre optic cable or satellite links.
- The internet is one of the biggest WAN in the world.
- A Wide Area Network is widely used in the field of Business, government, andeducation.
- WAN can be either a point-to-point WAN or Switched WAN.



Point-to-Point WAN

A point-to-point WAN is a network that connects two communicating devices through a transmission medium (cable or air). Figure 1.9 shows an example of a point-to-point WAN.

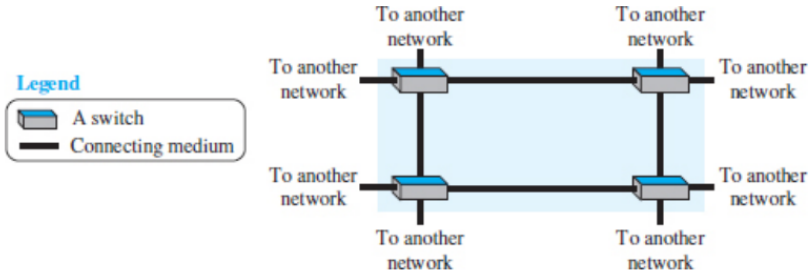
Figure 1.9 *A point-to-point WAN*



Switched WAN

A switched WAN is a network with more than two ends. It is used in the backbone of a global communications network today. Figure 1.10 shows an example of a switched WAN

Figure 1.10 *A switched WAN*



Advantages of Wide Area Network:	**Disadvantages of Wide Area Network:**
○ Large Geographical area ○ Centralized data ○ Exchange messages ○ Sharing of software and resources ○ High bandwidth	○ Security issue ○ Needs Firewall & antivirus software ○ High Setup cost ○ Troubleshooting problems

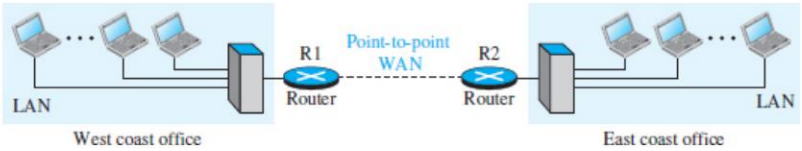
Internetwork

Today, it is very rare to see a LAN or a WAN in isolation; they are connected to one another. When two or more networks are connected, they make an internetwork, or internet. As an example, assume that an organization has two offices, one on the east coast and the other on the west coast.

Each office has a LAN that allows all employees in the office to communicate with each other. To make the communication between employees at different offices possible, the management leases a point-to-point dedicated WAN from a service provider, such as a telephone company, and connects the two LANs.

Now the company has an internetwork, or a private internet (with lowercase i). Communication between offices is now possible. Figure 1.11 shows this internet.

Figure 1.11 *An internetwork made of two LANs and one point-to-point WAN*



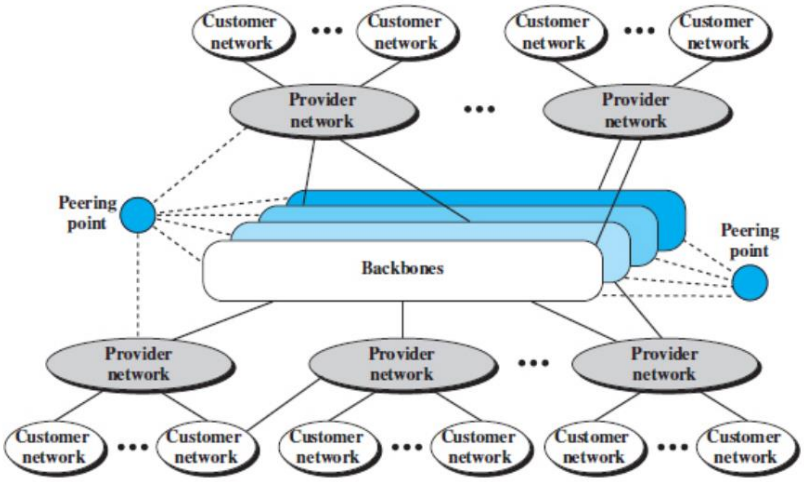
Types of Internetwork

Extranet	***Intranet***
An extranet is used for information sharing. The access to the extranet is restricted to only those users who have login credentials. An extranet is the lowest level of internetworking. It can becategorized as **MAN**, **WAN** or other computer networks. An extranet cannot have a single **LAN**, atleast it must haveone connection to the **external network**.	An intranet belongs to an organization which is only accessible by the **organization's employee** or members. The main aim of the intranet is to share the information and resources among the organization employees. An intranet provides the facility to work in groups and for teleconferences.

1.3.4 The Internet

An internet (note the lowercase i) is two or more networks that can communicate with each other. The most notable internet is called the Internet (uppercase I) and is composed of thousands of interconnected networks. Figure 1.13 shows a conceptual (not geographical) view of the Internet.

Figure 1.13 *The Internet today*



The figure shows the Internet as several backbones, provider networks, and customer networks. At the top level, the backbones are large networks owned by some communication companies. The backbone networks are connected through some complex switching systems, called peering points.

At the second level, there are smaller networks, called provider networks, that use the services of the backbones for a fee. The provider networks are connected to backbones and sometimes to other provider networks. The customer networks are networks at the edge of the Internet that actually use the services provided by the Internet. They pay fees to provider networks for receiving services.

Backbones and provider networks are also called Internet Service Providers (ISPs). The backbones are often referred to as international ISPs; the provider networks are often referred to as national or regional ISPs.

1.3.5 Accessing the Internet

The Internet today is an internetwork that allows any user to become part of it. The user, however, needs to be physically connected to an ISP. The physical connection is normally done through a point-to-point WAN (such as a telephone network, a cable network, a wireless network, or other types of networks).

Using Telephone Networks

Today most residences and small businesses have telephone service, which means they are connected to a telephone network. Because most telephone networks have already connected themselves to the Internet, one option for residences and small businesses to connect to the Internet is to change the voice line between the residence or business and the telephone center to a point-to-point WAN. This can be done in two ways.

❑ Dial-up service. The first solution is to add a modem that converts data to voice to the telephone line. The software installed on the computer dials the ISP and imitates making a

telephone connection. Unfortunately, the dial-up service is very slow, and when the line is used for an Internet connection, it cannot be used for a telephone (voice)connection. It is only useful for small residences and businesses with occasional connection to the Internet.

❑ DSL Service. Since the advent of the Internet, some telephone companies have upgraded their telephone lines to provide higher-speed Internet services to residences or small businesses. The digital subscriber line (DSL) service also allows the line to be used simultaneously for voice and data communications.

Using Cable Networks

More and more residents over the last two decades have begun using cable TV services instead of antennas to receive TV broadcasting. The cable companies have been upgrading their cable networks and connecting to the Internet. A residence or a small business can be connected to the Internet by using this service. It provides a higher-speed connection, but the speed varies depending on the number of neighbors that use the same cable.

Using Wireless Networks

Wireless connectivity has recently become increasingly popular. A household or a small business can use a combination of wireless and wired connections to access the Internet. With the growing wireless WAN access, a household or a small business can be connected to the Internet through a wireless WAN.

Direct Connection to the Internet

A large organization or a large corporation can itself become a local ISP and be connected to the Internet. This can be done if the organization or the corporation leases a high-speed WAN from a carrier provider and connects itself to a regional ISP. For example, a large university with several campuses can create an internetwork and then connect the internetwork to the Internet.

1.4 PROTOCOL LAYERING

- In networking, a protocol **defines the rules** that both the sender and receiver andall intermediate devices need to follow to be able **to communicate effectively**.
- A protocol provides a communication service that the process use to exchange messages.
- When communication is simple, we may need only one simple protocol.
- When the communication is complex, we may need to divide the task between different layers, in which case we need a protocol at each layer, or **protocol layering.**
- Protocol layering is that it allows us to separate the services from the implementation.
- A layer needs to be able to receive a set of services from the lower layer and to give the services to the upper layer.
- Any modification in one layer will not affect the other layers.

Basic Elements of Layered Architecture

- **Service**: It is a set of actions that a layer provides to the higher layer.
- **Protocol**: It defines a set of rules that a layer uses to exchange the information with peer entity. These rules mainly concern about both the contents and order of the messages used.

- **Interface:** It is a way through which the message is transferred from one layer to another layer.

Features of Protocol Layering

1. It decomposes the problem of building a network into more manageable components.
2. It provides a more modular design.

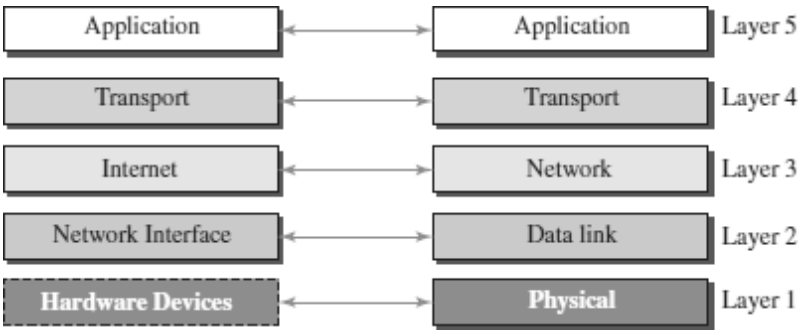
1.4.2 Principles of Protocol Layering

1. The first principle dictates that if we want bidirectional communication, we need to make each layer so that it is able to perform two opposite tasks, one in each direction.
2. The second principle that we need to follow in protocol layering is that the two objects under each layer at both sites should be identical.

1.5 TCP/IP PROTOCOL SUITE (INTERNET ARCHITECTURE)

The TCP/IP architecture is also called as Internet architecture.
It is developed by the US Defense Advanced Research Project Agency (**DARPA**)for its packet switched network (**ARPANET**).
TCP/IP is a protocol suite used in the Internet today.
It is a 5-layer model. The layers of TCP/IP are

1. Application layer
2. Transport Layer (TCP/UDP)
3. Network Layer
4. Datalink Layer
5. Physical Layer

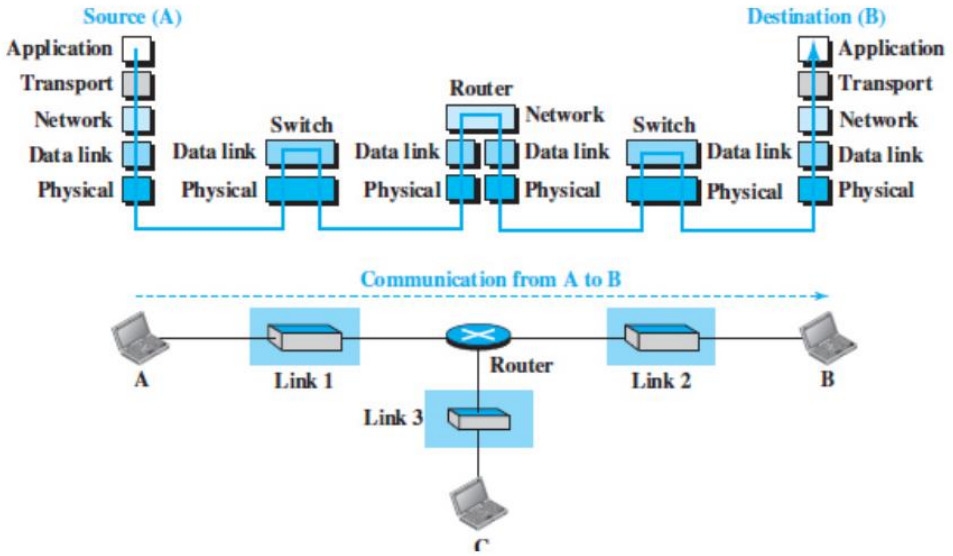


1.5.1 Layered Architecture

To show how the layers in the TCP/IP protocol suite are involved in communication between two hosts, we assume that we want to use the suite in a small internet made up of three LANs (links), each with a link-layer switch. We also assume that the links are connected by one router, as shown in Figure 1.18 (on next page). Let us assume that computer A communicates with computer B.

As Figure 1.18 shows, we have five communicating devices in this communication: source host (computer A), the link-layer switch in link 1, the router, the link-layer switch in link 2, and the destination host (computer B). Each device is involved with a set of layers depending on the role of the device in the internet. The two hosts are involved in all five layers.

Figure 1.18 *Communication through an internet*



1.5.2 Description of Each Layer

After understanding the concept of logical communication, we are ready to briefly discuss the duty of each layer.

Application Layer

- An application layer incorporates the function of top three OSI layers. Anapplication 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.
- Protocols such as FTP, HTTP, SMTP, POP3, etc running in the application layer provides service to other program running on top of application layer

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**.
  - **UDP** – UDP provides connectionless service and end-to-end delivery of transmission. It is an unreliable protocol as it discovers the errors but not specify the error.
  - **TCP** – TCP provides a full transport layer services to applications. TCP is a reliable protocol as it detects the error and retransmits the damaged frames.

Network Layer

- The network layer is the third layer of the TCP/IP model.
- The main responsibility of the network layer is to send the packets from any network, and they arrive at the destination irrespective of the route they take.
- Network layer handle the transfer of information across multiple networks through router and gateway .
- IP protocol is used in this layer, and it is the most significant part of the entire TCP/IP suite.

Data Link Layer

We have seen that an internet is made up of several links (LANs and WANs) connected by routers. When the next link to travel is determined by the router, the data-link layer is responsible for taking the datagram and moving it across the link.

Physical Layer

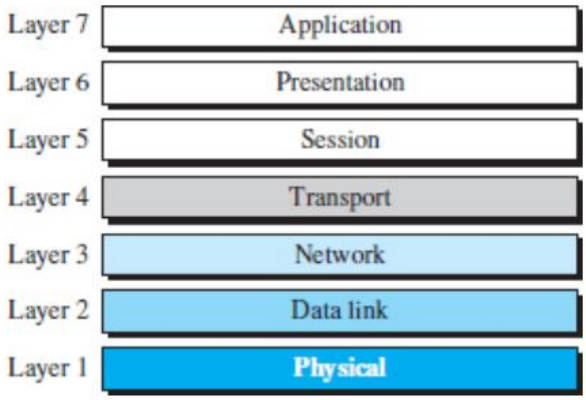
- The physical layer is responsible for carrying individual bits in a frame across the link.
- The physical layer is the lowest level in the TCP/IP protocol suite.
- The communication between two devices at the physical layer is still a logical communication because there is another hidden layer, the transmission media, under the physical layer.

1.6 THE OSI MODEL

An ISO standard that covers all aspects of network communications is the Open Systems Interconnection (OSI) model. It was first introduced in the late 1970s.

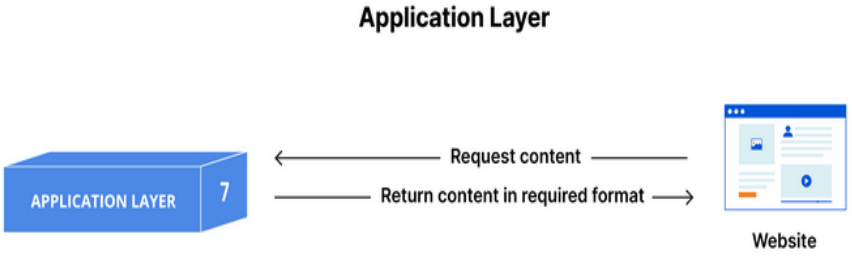
An open system is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture. The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software.

The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable. The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network (see Figure 1.21).



1.6.1 Application Layer

This is the only layer that directly interacts with data from the user. Software applications like web browsers and email clients rely on the application layer to initiate communications. But it should be made clear that client software applications are not part of the application layer; rather the application layer is responsible for the protocols and data manipulation that the software relies on to present meaningful data to the user. Application layer protocols include HTTP as well as SMTP (Simple Mail Transfer Protocol is one of the protocols that enables email communications).



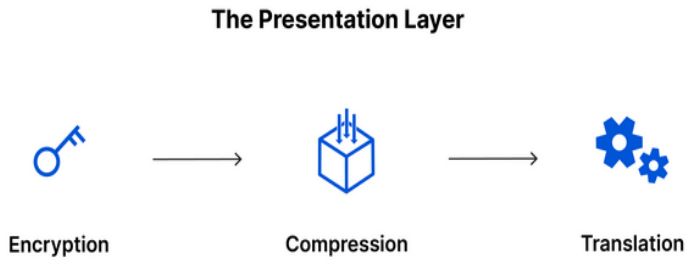
1.6.2 Presentation Layer

This layer is primarily responsible for preparing data so that it can be used by the application layer; in other words, layer 6 makes the data presentable for applications to consume. The presentation layer is responsible for translation, encryption, and compression of data.

Two communicating devices communicating may be using different encoding methods, so layer 6 is responsible for ***translating*** incoming data into a syntax that the application layer of the receiving device can understand.

If the devices are communicating over an ***encrypted*** connection, layer 6 is responsible for adding the encryption on the sender's end as well as decoding the encryption on the receiver's end so that it can present the application layer with unencrypted, readable data.

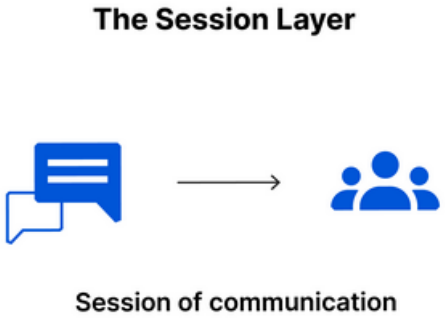
Finally the presentation layer is also responsible for ***compressing*** data it receives from the application layer before delivering it to layer 5. This helps improve the speed and efficiency of communication by minimizing the amount of data that will be transferred.



1.6.3 Session Layer

This is the layer responsible for opening and closing communication between the two devices. The time between when the communication is opened and closed is known as the session. The session layer ensures that the session stays open long enough to transfer all the data being exchanged, and then promptly closes the session in order to avoid wasting resources.

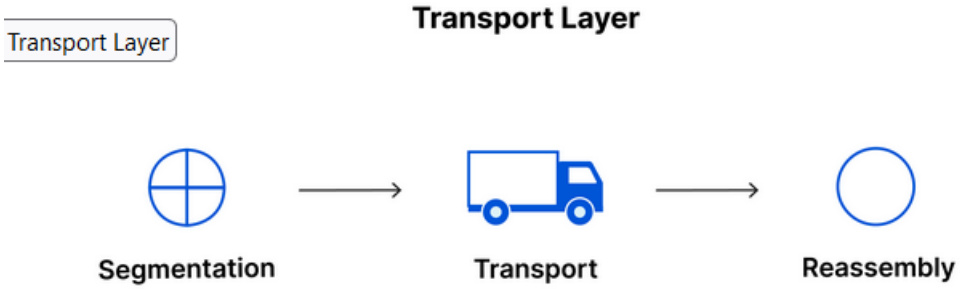
This layer is responsible for the establishment of connection, maintenance of sessions, authentication, and also ensures security.



1.6.4 Transport Layer

Layer 4 is responsible for end-to-end communication between the two devices. This includes taking data from the session layer and breaking it up into chunks called ***segments*** before sending it to layer 3. The transport layer on the receiving device is responsible for reassembling the segments into data the session layer can consume.

The transport layer is also responsible for ***flow control and error control***. Flow control determines an optimal speed of transmission to ensure that a sender with a fast connection does not overwhelm a receiver with a slow connection. The transport layer performs error control on the receiving end by ensuring that the data received is complete, and requesting a retransmission if it isn't.



1.6.5 Network Layer

The network layer works for the transmission of data from one host to the other located in different networks. It also takes care of ***packet routing*** i.e. selection of the shortest path to transmit the packet, from the number of routes available. The sender & receiver's IP addresses are placed in the header by the network layer.

The functions of the Network layer are :

1. **Routing:** The network layer protocols determine which route is suitable from source to destination. This function of the network layer is known as routing.
2. **Logical Addressing:** In order to identify each device on internetwork uniquely, the network layer defines an addressing scheme. The sender & receiver's IP addresses are placed in the header by the network layer. Such an address distinguishes each device uniquely and universally.

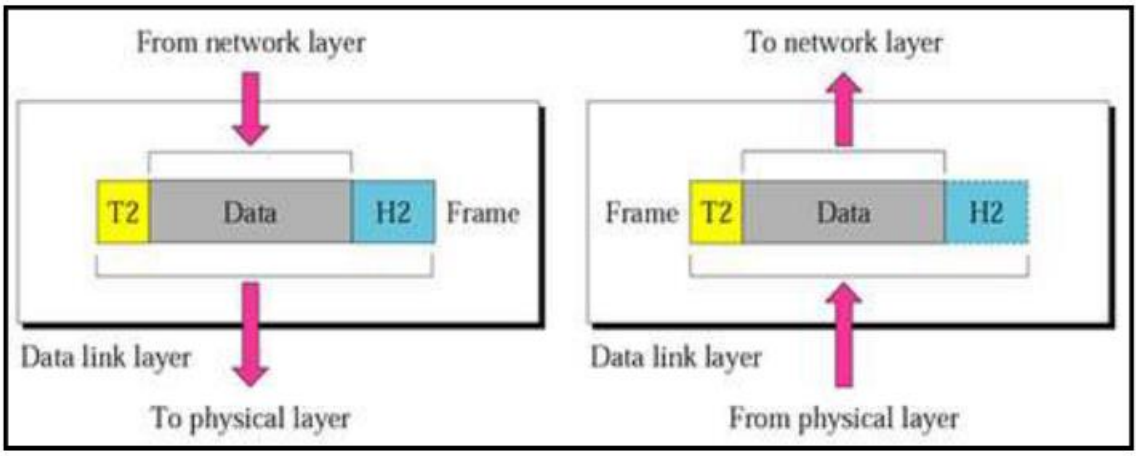
1.6.6 Data Link Layer

The data link layer is responsible for the node-to-node delivery of the message. The main function of this layer is to make sure data transfer is error-free from one node to another, over the physical layer. When a packet arrives in a network, it is the responsibility of DLL to transmit it to the Host using its MAC address.
Data Link Layer is divided into two sublayers:

1. Logical Link Control (LLC)
2. Media Access Control (MAC)

The packet received from the Network layer is further divided into frames depending on the frame size of NIC(Network Interface Card). DLL also encapsulates Sender and Receiver's MAC address in the header.

The Receiver's MAC address is obtained by placing an ARP(Address Resolution Protocol) request onto the wire asking "Who has that IP address?" and the destination host will reply with its MAC address.



1.6.7 Physical Layer

This layer includes the physical equipment involved in the data transfer, such as the cables and switches. This is also the layer where the data gets converted into a bit stream, which is a string of 1s and 0s. The physical layer of both devices must also agree on a signal convention so that the 1s can be distinguished from the 0s on both devices.

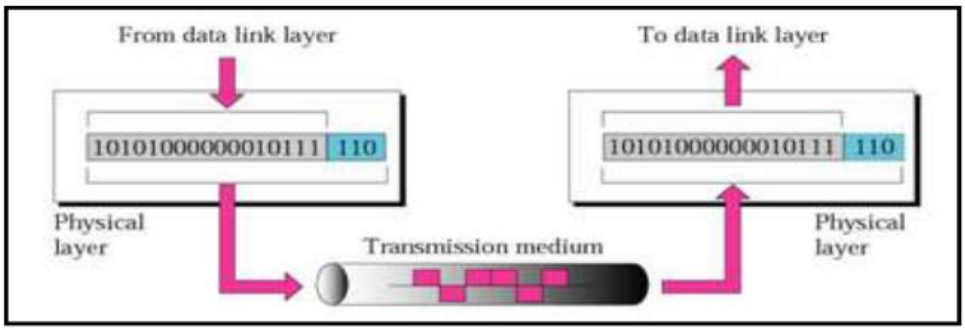
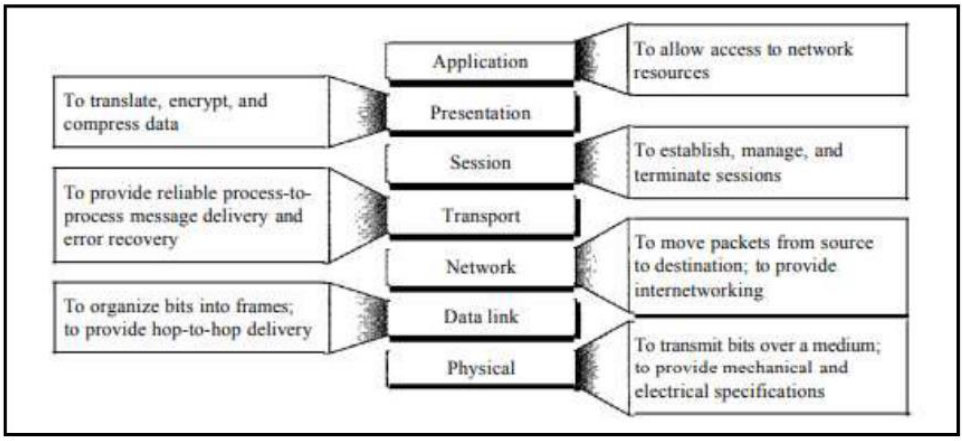
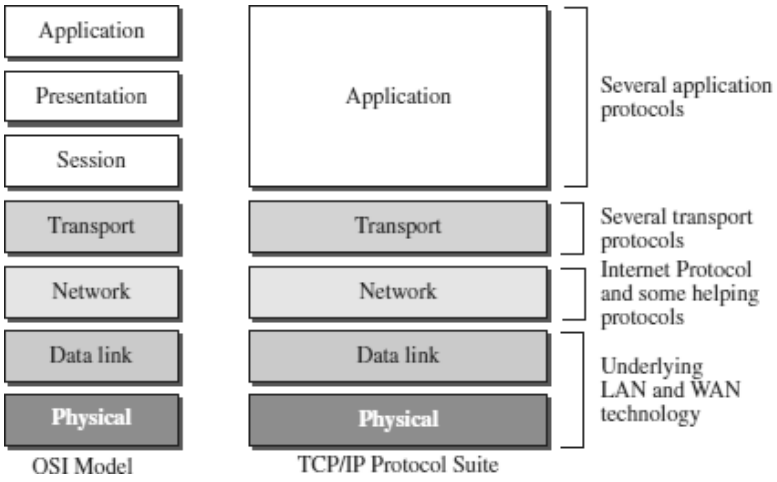


Figure 1.15 *Physical layer*

Summary of Layers



COMPARISON - OSI MODEL AND TCP/IP MODEL



S.No	OSI MODEL	TCP/IP MODEL
1	Defined before advent of internet	Defined after the advent of Internet.
2	Service interface and protocols are clearly distinguished before	Service interface and protocols were notclearly distinguished before
3	Internetworking not supported	TCP/IP supports Internet working
4	Strict layering	Loosely layered
5	Protocol independent standard	Protocol Dependant standard
6	Less Credible	More Credible
7	All packets are reliably delivered	TCP reliably delivers packets, IP doesnot reliably deliver packets

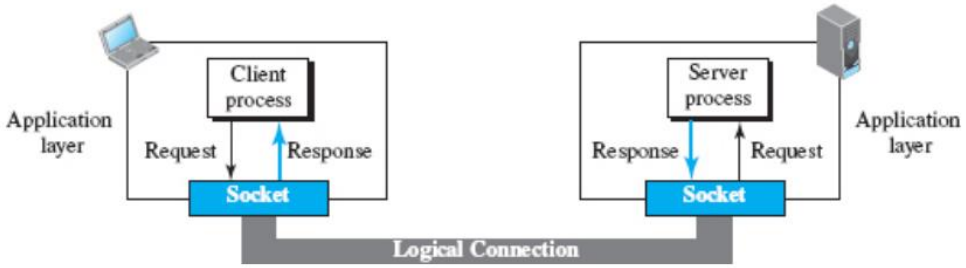
1.7 Introduction to Sockets

A **socket** is one endpoint of a **two way** communication link between two programs running on the network. The socket mechanism provides a means of inter-process communication (IPC) by establishing named contact points between which the communication take place.

Like 'Pipe' is used to create pipes and sockets is created using **'socket'** system call. The socket provides bidirectional **FIFO** Communication facility over the network. A socket connecting to the network is created at each end of the communication. Each socket has a specific address. This address is composed of an IP address and a port number.

Socket are generally employed in client server applications. The server creates a socket, attaches it to a network port addresses then waits for the client to contact it. The client creates a socket and then attempts to connect to the server socket. When the connection is established, transfer of data takes place.

Use of sockets in process-to-process communication



1.7.1 Socket Addresses

The interaction between a client and a server is two-way communication. In a two-way communication, we need a pair of addresses:

local (sender) and remote (receiver).

The local address in one direction is the remote address in the other direction, and vice versa. Because communication in the client/server paradigm is between two sockets, we need a pair of socket addresses for communication:

a local socket address and a remote socket address.

A socket address should first define the computer on which a client or a server is running. A computer in the Internet is uniquely defined by its IP address, a 32-bit integer in the current Internet version. An application program can be defined by a port number, a 16-bit integer. This means that a socket address should be a combination of an IP address and a port number as shown in Figure 10.7.

Figure 10.7 *A socket address*

Because a socket defines the end-point of the communication, we can say that a socket is identified by a pair of socket addresses, a local and a remote.

1.7.2 Finding Socket Addresses

How can a client or a server find a pair of socket addresses for communication? The situation is different for each site.

Server Site

The server needs a local (server) and a remote (client) socket address for communication.

Local Socket Address The local (server) socket address is provided by the operating system. The operating system knows the IP address of the computer on which the server process is running. The port number of a server process, however, needs to be assigned. If the server process is a standard one defined by the Internet authority, a port number is already assigned to it. When a server starts running, it knows the local socket address.

Remote Socket Address The remote socket address for a server is the socket address of the client that makes the connection. Because the server can serve many clients, it does not know beforehand the remote socket address for communication. The server can find this socket address when a client tries to connect to the server. The client socket address, which is contained in the request packet sent to the server, becomes the remote socket address that is used for responding to the client.

Client Site

The client also needs a local (client) and a remote (server) socket address for communication.

Local Socket Address The local (client) socket address is also provided by the operating system. The operating system knows the IP address of the computer on which the client is running. The port number, however, is a 16- bit temporary integer that is assigned to a client process each time the process needs to start the communication. The port number, however, needs to be assigned from a set of integers defined by the Internet authority and called the ephemeral (temporary) port numbers. The operating system, however, needs to guarantee that the new port number is not used by any other running client process.

Remote Socket Address Finding the remote (server) socket address for a client, however, needs more work. When a client process starts, it should know the socket address of the server it wants to connect to. We will have two situations in this case.

Sometimes, the user who starts the client process knows both the server port number and IP address of the computer on which the server is running. This usually occurs in situations when we have written client and server applications and we want to test them

Although each standard application has a well-known port number, most of the time, we do not know the IP address. This happens in situations such as when we need to contact a web page, send an e-mail to a friend, or copy a file from a remote site. In these situations, the server has a name, an identifier that uniquely defines the server process. Examples of these identifiers are URLs, such as www.xxx.yyy, or e-mail addresses, such as xxxx@yyyy.com. The client process should now change this identifier (name) to the corresponding server socket address.

1.8 Application Layer

- The application layer is the highest layer in the protocol suite.
- The application layer provides services to the user.
- Communication is provided using a logical connection, which means that the two application layers assume that there is an imaginary direct connection through which they can send and receive messages.
- The application layer is the only layer that provides services to the Internet user
- The application layer exchange messages with their peers on other machines
- Applications need their own protocols. These applications are part of network protocol.

Types of Application Protocols:

Standard and Nonstandard Protocols

Standard Application-Layer Protocols

o There are several application-layer protocols that have been standardized and documented by the Internet authority.

o Each standard protocol is a pair of computer programs that interact with the user and the transport layer to provide a specific service to the user.

o Two very widely-used standardized application protocols:

SMTP: Simple Mail Transfer Protocol is used to exchange electronic mail.

HTTP : Hyper Text Transport Protocol is used to communicate between Web browsers and Web servers.

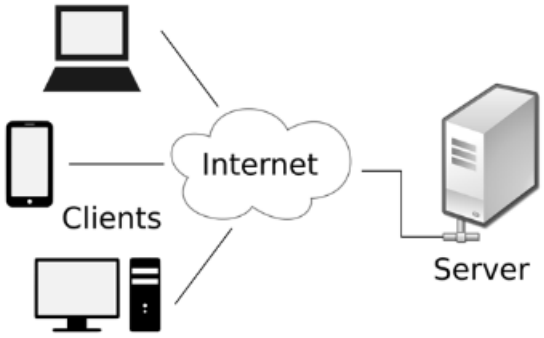
Nonstandard Application-Layer Protocols

o A programmer can create a nonstandard application-layer program if they can write two programs that provide service to the user by interacting with the transport layer.

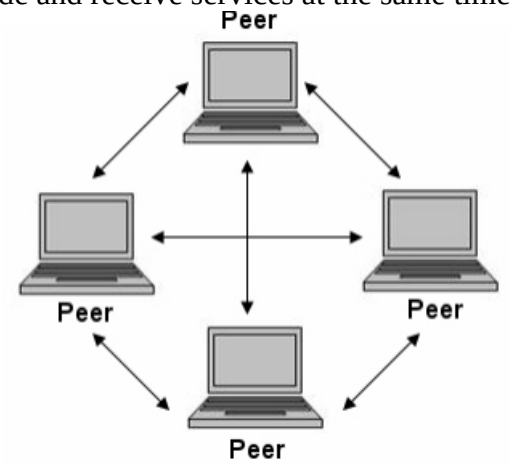
Application-Layer Paradigms

Two paradigms have been developed for Application Layer
1. Traditional Paradigm : Client-Server
2. New Paradigm : Peer-to-Peer

Client-Server Paradigm
o The traditional paradigm is called the client-server paradigm.
o It was the most popular Paradigm.
o In this paradigm, the service provider is an application program, called the server process; it runs continuously, waiting for another application program, called the client process, to make a connection through the Internet and ask for service.
o The server process must be running all the time; the client process is started when the client needs to receive service.
o There are normally some server processes that can provide a specific type of service, but there are many clients that request service from any of these server processes.



Peer-to-Peer(P2P) Paradigm
o A new paradigm, called the peer-to-peer paradigm has emerged to respond to the needs of some new applications.
o In this paradigm, there is no need for a server process to be running all the time and waiting for the client processes to connect.
o The responsibility is shared between peers.
o A computer connected to the Internet can provide service at one time and receive service at another time.
o A computer can even provide and receive services at the same time.



Mixed Paradigm

o An application may choose to use a mixture of the two paradigms by combining the advantages of both.

o For example, a light-load client-server communication can be used to find the address of the peer that can offer a service.

o When the address of the peer is found, the actual service can be received from the peer by using the peer-to-peer paradigm.

1.8.1 The 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.
- It is a protocol used to access the data on the World Wide Web (WWW).
- The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
- HTTP is a stateless request/response protocol that governs client/server communication.
- 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 , a connection-oriented and reliable protocol.
- HTTP is a text-oriented protocol. It contains embedded URL known as links.
- When hypertext is clicked, browser opens a new connection, retrieves file from the server and displays the file.
- Each HTTP message has the general form

START_LINE <CRLF>
MESSAGE_HEADER <CRLF>
<CRLF> MESSAGE_BODY <CRLF>
where <CRLF> stands for carriage-return-line-feed.

Features of HTTP

o ***Connectionless protocol:***

HTTP is a connectionless protocol. HTTP client initiates a request and waits for a response from the server. When the server receives the request, the server processes the request and sends back the response to the HTTP client after which the client disconnects the connection. The connection between client and server exist only during the current request and response time only.

o Media independent:

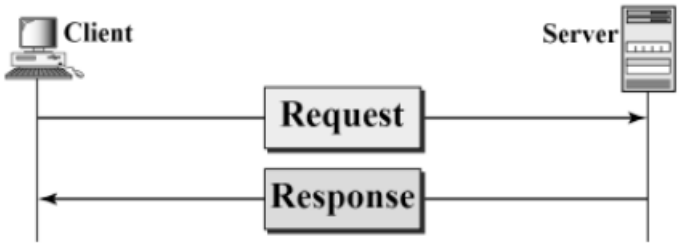
HTTP protocol is a media independent as data can be sent as long as both the client and server know how to handle the data content. It is required for both the client and server to specify the content type in MIME-type header.

o Stateless:

HTTP is a stateless protocol as both the client and server know each other only during the current request. Due to this nature of the protocol, both the client and server do not retain the information between various requests of the web pages.

HTTP Request And Response Messages

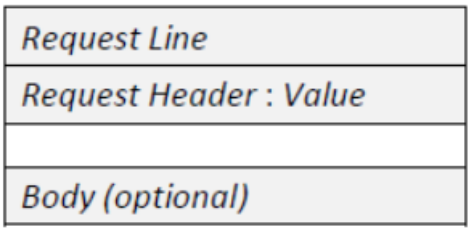
• The HTTP protocol defines the format of the request and response messages.



• Request Message: The request message is sent by the client that consists of a request line, headers, and sometimes a body.

• Response Message: The response message is sent by the server to the client that consists of a status line, headers, and sometimes a body.

HTTP Request Message



• The first line in a request message is called a request line.

• After the request line, we can have zero or more request header lines.

• The body is an optional one. It contains the comment to be sent or the file to be published on the website when the method is PUT or POST.

Request Line

• There are three fields in this request line - Method, URL and Version.

• The Method field defines the request types.

• The URL field defines the address and name of the corresponding web page.

• The Version field gives the version of the protocol; the most current version of HTTP is 1.1.

• Some of the Method types are:

Method	*Action*
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

Request Header

- Each request header line sends additional information from the client to the server.
- Each header line has a header name, a colon, a space, and a header value.
- The value field defines the values associated with each header name.
- Headers defined for request message include:

Header	*Description*
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
If-Modified-Since	If the file is modified since a specific date

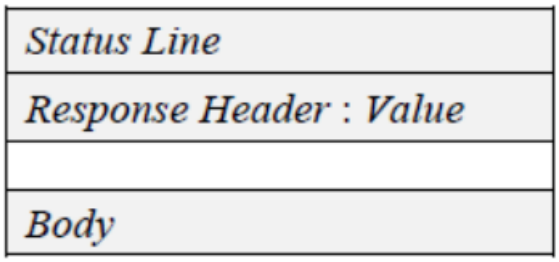
Body

- The body can be present in a request message. It is optional.
- Usually, it contains the comment to be sent or the file to be published on the website when the method is PUT or POST.

Conditional Request

- A client can add a condition in its request.
- In this case, the server will send the requested web page if the condition is met or inform the client otherwise.
- One of the most common conditions imposed by the client is the time and date the web page is modified.
- The client can send the header line If-Modified-Since with the request to tell the server that it needs the page only if it is modified after a certain point in time.

HTTP Response Message



- The first line in a request message is called a status line.
- After the request line, we can have zero or more response header lines.
- The body is an optional one. The body is present unless the response is an error message.

Status Line

- The Status line contains three fields - HTTP version , Status code, Status phrase
- The first field defines the version of HTTP protocol, currently 1.1.
- The status code field defines the status of the request. It classifies the HTTP result. It consists of three digits.

1xx–Informational, 2xx– Success, 3xx–Redirection,
4xx–Client error, 5xx–Server error

- The Status phrase field gives brief description about status code in text form.
- Some of the Status codes are

Code	Phrase	Description
100	Continue	Initial request received, client to continue process
200	OK	Request is successful
301	Moved permanently	Requested URL is no longer in use
404	Not found	Document not found
500	Internal server error	An error such as a crash, at the server site

Response Header

- Each header provides additional information to the client.
- Each header line has a header name, a colon, a space, and a header value.
- Some of the response headers are:

Response Header	Description
Content-type	specifies the MIME type
Expires	date and time up to which the document is valid
Last-modified	date and time when the document was last updated
Location	specifies location of the created or moved document

Body

- 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.

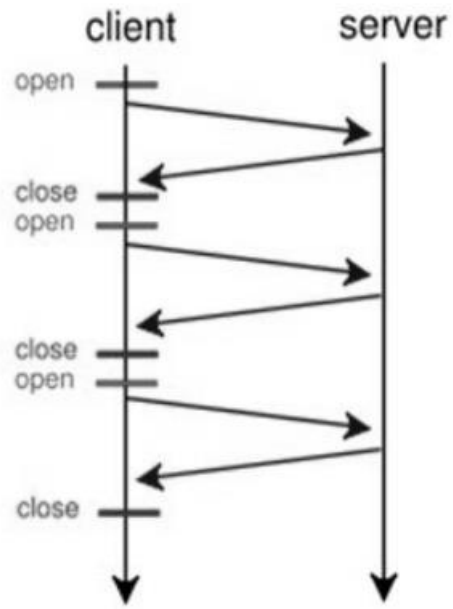
HTTP CONNECTIONS

- HTTP Clients and Servers exchange multiple messages over the same TCP connection.
- 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 non-persistent connection, the second as a persistent connection.
- HTTP 1.0 uses non-persistent connections and HTTP 1.1 uses persistent connections .

Non-Persistent Connections

- In a non-persistent connection, one TCP connection is made for each request/response.
- Only one object can be sent over a single TCP connection
- The client opens a TCP connection and sends a request.
- The server sends the response and closes the connection.

- The client reads the data until it encounters an end-of-file marker.
- It then closes the connection.



Persistent Connections

- HTTP version 1.1 specifies a persistent connection by default.
- Multiple objects can be sent over a single TCP connection.
- 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.
- 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.

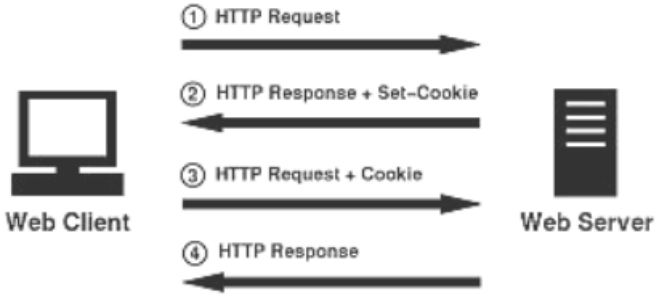
Http Cookies

- An HTTP cookie (also called web cookie, Internet cookie, browser cookie, or simply cookie) is a small piece of data sent from a website and stored on the user's computer by the user's web browser while the user is browsing.

- HTTP is stateless , Cookies are used to add State.

- Cookies were designed to be a reliable mechanism for websites to remember stateful information (such as items added in the shopping cart in an online store) or to record the user's browsing activity (including clicking particular buttons, logging in, or recording which pages were visited in the past).

- They can also be used to remember arbitrary pieces of information that the user previously entered into form fields such as names, addresses, passwords, and credit card numbers.



Components of Cookie

A cookie consists of the following components:

1. Name
2. Value
3. Zero or more attributes (name/value pairs). Attributes store information such as the cookie's expiration, domain, and flags.

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.
- 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.

Types of Cookies

1.Authentication cookies

These are the most common method used by web servers to know whether the user is logged in or not, and which account they are logged in with. Without such a mechanism, the site would not know whether to send a page containing sensitive information, or require the user to authenticate themselves by logging in.

2.Tracking cookies

These are commonly used as ways to compile individuals browsing histories.

3.Session cookie

A session cookie exists only in temporary memory while the user navigates the website. Web browsers normally delete session cookies when the user closes the browser.

4.Persistent cookie

Instead of expiring when the web browser is closed as session cookies do, a persistent cookie expires at a specific date or after a specific length of time. This means that, for the cookie's entire lifespan , its information will be transmitted to the server every time the user visits the website that it belongs to, or every time the user views a resource belonging to that website from another website

Http Caching

¬ HTTP Caching enables the client to retrieve document faster and reduces load on the server.
¬ HTTP Caching is implemented at Proxy server, ISP router and Browser.
¬ Server sets expiration date (Expires header) for each page, beyond which it is not cached.
¬ HTTP Cache document is returned to client only if it is an updated copy by checking against If-Modified-Since header.
¬ If cache document is out-of-date, then request is forwarded to the server and response is cached along the way.
¬ A web page will not be cached if no-cache directive is specified.

HTTP SECURITY

¬ HTTP does not provide security.
¬ However HTTP can be run over the Secure Socket Layer (SSL).
¬ In this case, HTTP is referred to as HTTPS.
¬ HTTPS provides confidentiality, client and server authentication, and data integrity.

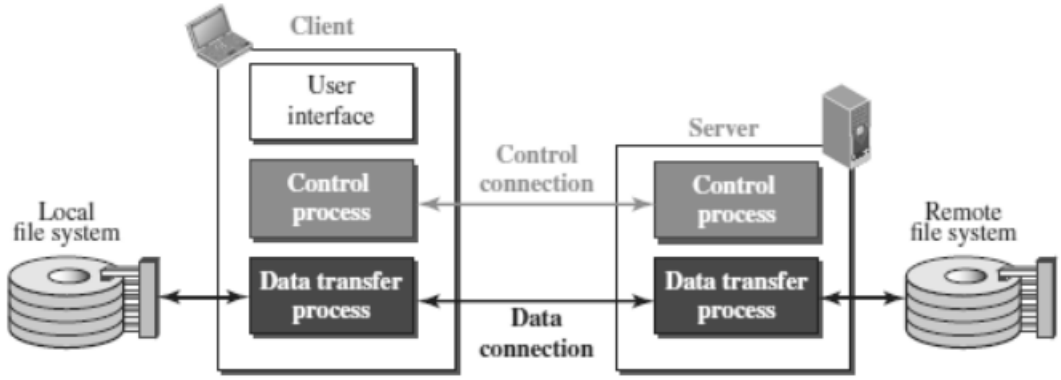
1.8.2 FTP (FILE TRANSFER PROTOCOL)

¬ FTP stands for File transfer protocol.
¬ FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another.
¬ It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet.
¬ It is also used for downloading the files to computer from other servers.
¬ Although we can transfer files using HTTP, FTP is a better choice to transfer large files or to transfer files using different formats.

FTP OBJECTIVES

¬ It provides the sharing of files.
¬ It is used to encourage the use of remote computers.
¬ It transfers the data more reliably and efficiently.

FTP MECHANISM



¬ The above figure shows the basic model of the FTP.
¬ The FTP client has three components:
o user interface, control process, and data transfer process.
¬ The server has two components:
o server control process and server data transfer process.

FTP CONNECTIONS

¬ There are two types of connections in FTP - Control Connection and Data Connection.

¬ The two connections in FTP have different lifetimes.

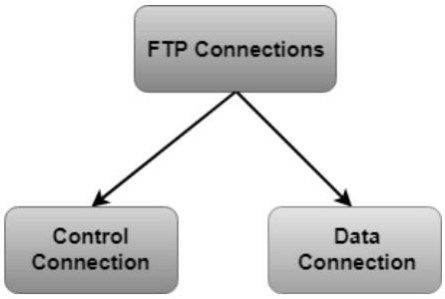
¬ The control connection remains connected during the entire interactive FTP session.

¬ The data connection is opened and then closed for each file transfer activity. 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:

o Port 21 is used for the control connection
o Port 20 is used for the data connection.



Control Connection:

o The control connection uses very simple rules for communication.
o Through control connection, we can transfer a line of command or line of response at a time.
o The control connection is made between the control processes.
o The control connection remains connected during the entire interactive FTP session.

¬ Data Connection:

o The Data Connection uses very complex rules as data types may vary.
o The data connection is made between data transfer processes.
o The data connection opens when a command comes for transferring the files and closes when the file is transferred.

FTP COMMUNICATION

¬ FTP Communication is achieved through commands and responses.
¬ FTP Commands are sent from the client to the server
¬ FTP responses are sent from the server to the client.
¬ FTP Commands are in the form of ASCII uppercase, which may or may not be followed by an argument.
¬ Some of the most common commands are:

Command	*Description*
ABOR	Abort the previous command
CDUP	Change to parent directory
CWD	Change to another directory
DELE	Delete a file
LIST	List subdirectories or files
MKD	Create a new directory
PASS	Password
PASV	Server chooses a port
PORT	Client chooses a port
PWD	Display name of current directory
QUIT	Log out of the system
RETR	Retrieve files; files are transferred from server to client
RMD	Delete a directory
RNFR	Identify a file to be renamed
RNTO	Rename the file
STOR	Store files; file(s) are transferred from client to server
STRU	Define data organization (**F**: file, **R**: record, or **P**: page)
TYPE	Default file type (**A**: ASCII, **E**: EBCDIC, **I**: image)
USER	User information
MODE	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 parameter.

Code	*Description*	*Code*	*Description*
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

FTP FILE TYPE

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

FTP DATA STRUCTURE

¬ FTP can transfer a file across the data connection using one of the following data structure : file structure, record structure, or page structure.
¬ The file structure format is the default one and 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.

FTP 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 the compressed mode, data can be compressed and delivered from FTP to TCP.

FTP FILE TRANSFER

¬ File transfer occurs over the data connection under the control of the commands sent over the control connection.
¬ File transfer in FTP means one of three things:
o retrieving a file (server to client)
o storing a file (client to server)
o directory listing (server to client).

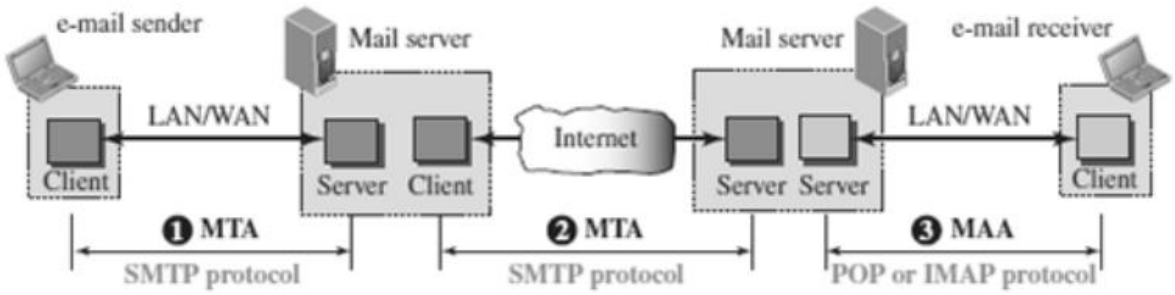
FTP SECURITY

¬ FTP requires a password, the password is sent in plaintext which is unencrypted. This means it can be intercepted and used by an attacker.
¬ The data transfer connection also transfers data in plaintext, which is insecure.

¬ To be secure, one can add a Secure Socket Layer between the FTP application layer and the TCP layer.
¬ In this case FTP is called SSL-FTP.

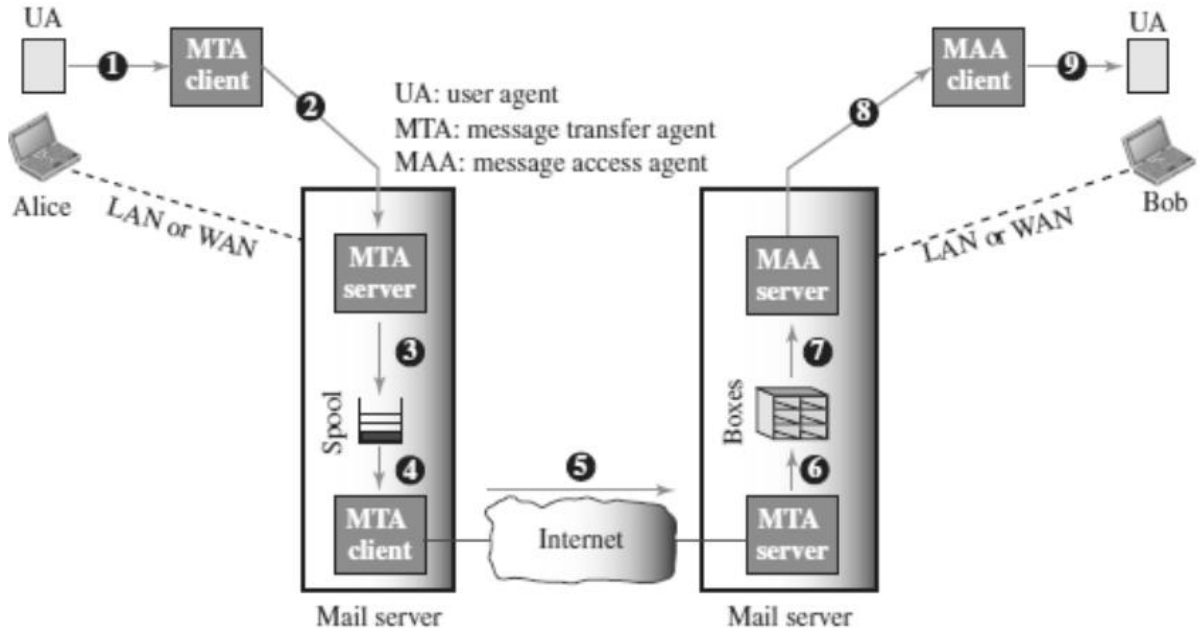
1.8.3 EMAIL (SMTP, MIME, IMAP, POP)

¬ One of the most popular Internet services is electronic mail (E-mail).
¬ Email is one of the oldest network applications.
¬ The three main components of an Email are
1. User Agent (UA)
2. Messsage Transfer Agent (MTA) – SMTP
3. Messsage Access Agent (MAA) - IMAP , POP



¬ When the sender and the receiver of an e-mail are on the same system, we need only two User Agents and no Message Transfer Agent
¬ When the sender and the receiver of an e-mail are on different system, we need two UA, two pairs of MTA (client and server), and two MAA (client and server).

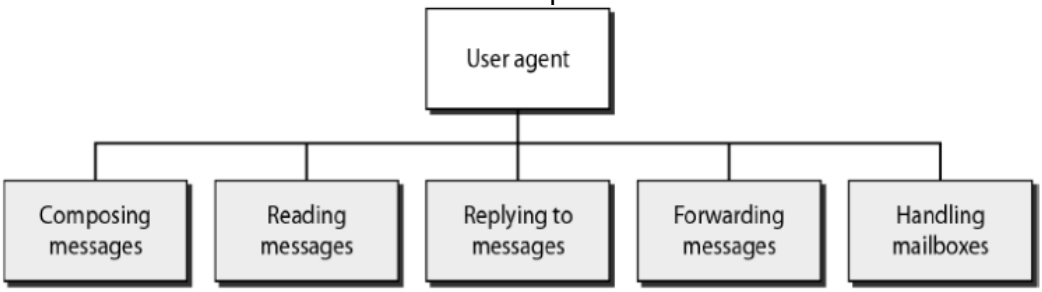
WORKING OF EMAIL



¬ 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.
¬ Here two message transfer agents are needed: one client and one server.
¬ The server needs to run all the time because it does not know when a client will ask for a connection.
¬ The client can be triggered by the system when there is a message in the queue to be sent.
¬ The user agent at the Bob site allows Bob to read the received message.
¬ Bob later uses an MAA client to retrieve the message from an MAA server running on the second server.

USER AGENT (UA)

¬ 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 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

o Command driven user agents belong to the early days of electronic mail.
o A command-driven user agent normally accepts a one character command from the keyboard to perform its task.
o Some examples of command driven user agents are mail, pine, and elm.

GUI-based

o Modern user agents are GUI-based.
o They allow the user to interact with the software by using both the keyboard and the mouse.
o They have graphical components such as icons, menu bars, and windows that make the services easy to access.
o Some examples of GUI-based user agents are Eudora and Outlook.

MESSAGE TRANSFER AGENT (MTA)

¬ The actual mail transfer is done through message transfer agents (MTA).
¬ To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA.
¬ The formal protocol that defines the MTA client and server in the Internet is called Simple Mail Transfer Protocol (SMTP).

MESSAGE ACCESS AGENT (MAA)

¬ MAA is a software that pulls messages out of a mailbox.
¬ POP3 and IMAP4 are examples of MAA.

ADDRESS FORMAT OF EMAIL

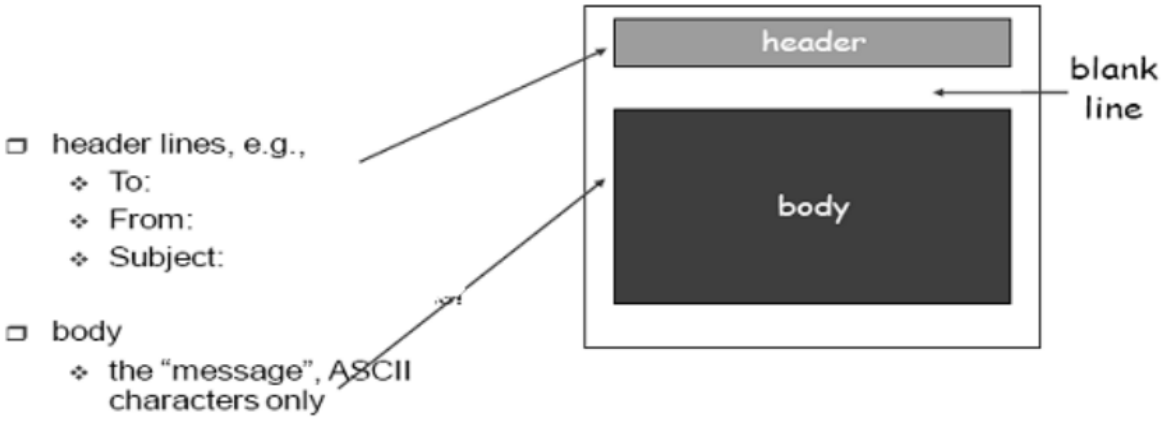
¬ E-mail address is userid @ domain where domain is hostname of the mail server.

MESSAGE FORMAT OF EMAIL

¬ Email message consists of two parts namely header and body.
¬ Each header line contains type and value separated by a colon (:).
¬ Some header contents are:

o From: identifier sender of the message.
o To: mail address of the recipient(s).
o Subject: says about purpose of the message.
o Date: timestamp of when the message was transmitted.

¬ Header is separated from the body by a blank line.
¬ Body contains the actual message.

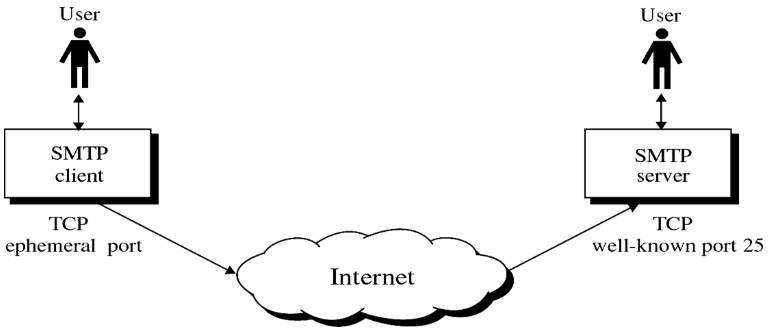


¬ Email was extended in 1993 to carry many different types of data: audio, video, images, Word documents, and so on.
¬ This extended version is known as MIME(Multipurpose Mail Extension).

1.8.4.1 SIMPLE MAIL TRANSFER PROTOCOL (SMTP)

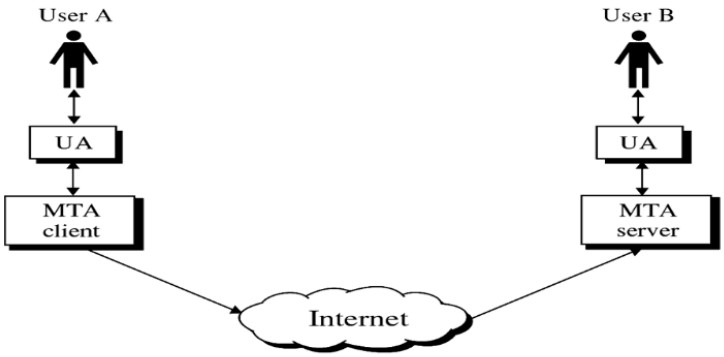
¬ SMTP is the standard protocol for transferring mail between hosts in the TCP/IP protocol suite.

¬ SMTP is not concerned with the format or content of messages themselves.
¬ SMTP uses information written on the envelope of the mail (message header), but does not look at the contents (message body) of the envelope.

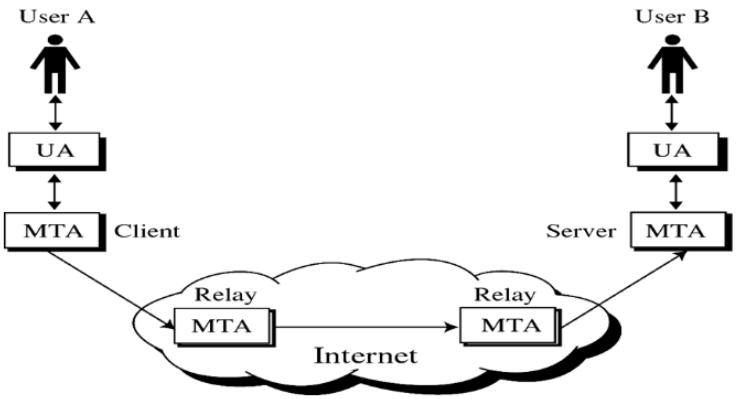


¬ SMTP clients and servers have two main components

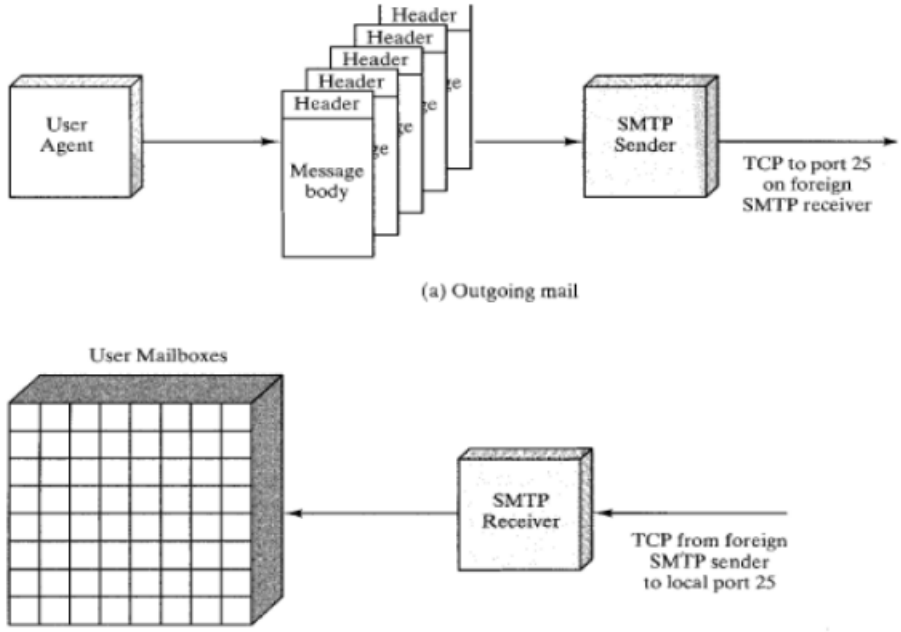
o User Agents(UA) – Prepares the message, encloses it in an envelope.
o Mail Transfer Agent (MTA) – Transfers the mail across the internet



¬ SMTP also allows the use of Relays allowing other MTAs to relay the mail.



SMTP MAIL FLOW



¬ To begin, mail is created by a user-agent program in response to user input.
¬ Each created message consists of a header that includes the recipient's email address and other information, and a message body containing the message to be sent.
¬ These messages are then queued in some fashion and provided as input to an SMTP Sender program.

SMTP COMMANDS AND RESPONSES

¬ The operation of SMTP consists of a series of commands and responses exchanged between the SMTP sender and SMTP receiver.
¬ The initiative is with the SMTP sender, who establishes the TCP connection.
¬ Once the connection is established, the SMTP sender sends commands over the connection to the receiver.
¬ The command is from an MTA client to an MTA server; the response is from an MTA server to the MTA client.

SMTP Commands

¬ Commands are sent from the client to the server. It consists of a keyword followed by zero or more arguments. SMTP defines 14 commands.

SMTP commands

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
VRFY	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 *or* the mailbox of the recipient
SMAL FROM	Intended recipient	Specifies that the mail be delivered to the terminal *and* the mailbox of the recipient

SMTP 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.

SMTP Responses

Code	*Description*
	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
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

SMTP OPERATIONS

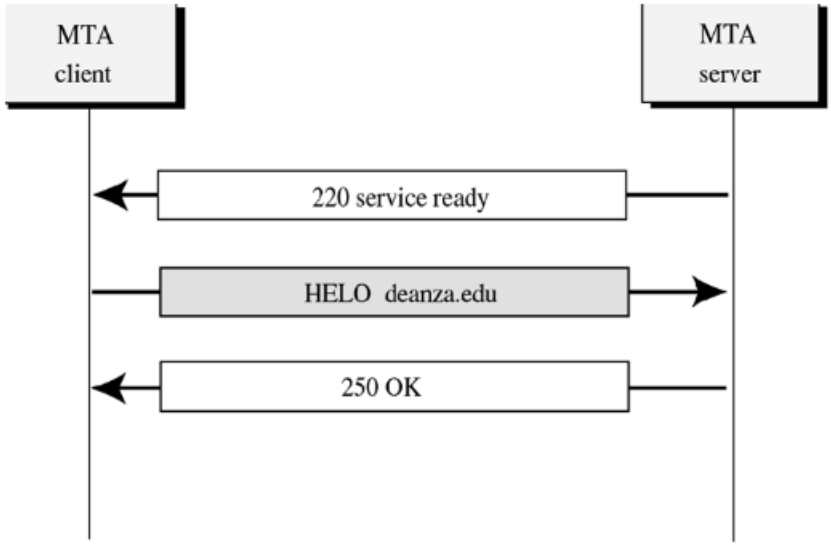
Basic SMTP operation occurs in three phases:
1. Connection Setup
2. Mail Transfer
3. Connection Termination

Connection Setup

¬ An SMTP sender will attempt to set up a TCP connection with a target host when it has one or more mail messages to deliver to that host.

¬ The sequence is quite simple:

1. The sender opens a TCP connection with the receiver.
2. Once the connection is established, the receiver identifies itself with "Service Ready".
3. The sender identifies itself with the HELO command.
4. The receiver accepts the sender's identification with "OK".
5. If the mail service on the destination is unavailable, the destination host returns a "Service Not Available" reply in step 2, and the process is terminated.



Mail Transfer

¬ Once a connection has been established, the SMTP sender may send one or more messages to the SMTP receiver.

¬ There are three logical phases to the transfer of a message:

1. A MAIL command identifies the originator of the message.
2. One or more RCPT commands identify the recipients for this message.
3. A DATA command transfers the message text.

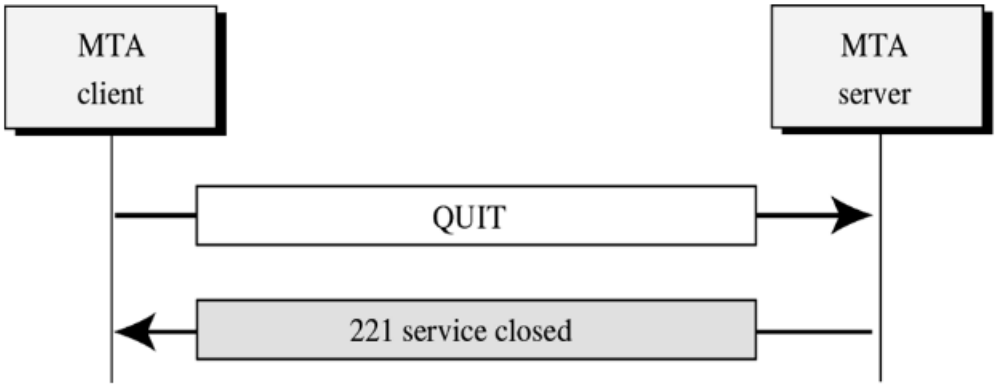
Connection Termination

¬ The SMTP sender closes the connection in two steps.

¬ First, the sender sends a QUIT command and waits for a reply.

¬ The second step is to initiate a TCP close operation for the TCP connection.

¬ The receiver initiates its TCP close after sending its reply to the QUIT command.



Limitations Of Smtp

¬ SMTP cannot transmit executable files or other binary objects.
¬ SMTP cannot transmit text data that includes national language characters, as these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.
¬ SMTP servers may reject mail message over a certain size.
¬ SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems.
¬ Some SMTP implementations do not adhere completely to the SMTP standards defined.
¬ Common problems include the following:
1. Deletion, addition, or recording of carriage return and linefeed.
2. Truncating or wrapping lines longer than 76 characters.
3. Removal of trailing white space (tab and space characters).
4. Padding of lines in a message to the same length.
5. Conversion of tab characters into multiple-space characters.

1.8.4.2 MULTIPURPOSE INTERNET MAIL EXTENSION (MIME)

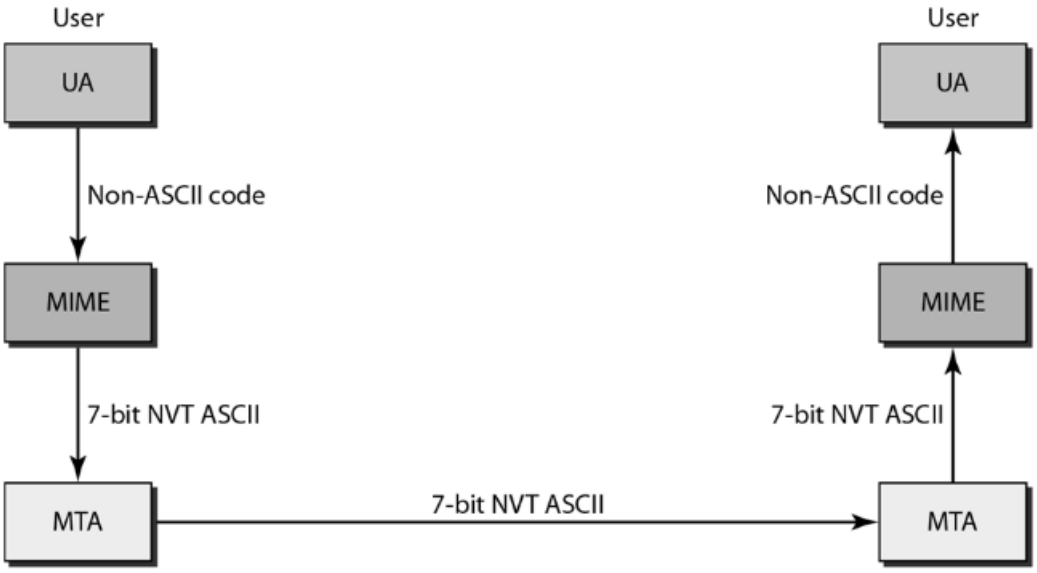
¬ SMTP provides a basic email service, while MIME adds multimedia capability to SMTP.
¬ MIME is an extension to SMTP and is used to overcome the problems and limitations of SMTP.
¬ Email system was designed to send messages only in ASCII format.

- Languages such as French, Chinese, etc., are not supported.
- Image, audio and video files cannot be sent.

¬ MIME adds the following features to email service:

- Be able to send multiple attachments with a single message;
- Unlimited message length;
- Use of character sets other than ASCII code;
- Use of rich text (layouts, fonts, colors, etc)
- Binary attachments (executables, images, audio or video files, etc.), which may be divided if needed.

¬ MIME is a protocol that converts non-ASCII data to 7-bit NVT(Network Virtual Terminal) ASCII and vice-versa.

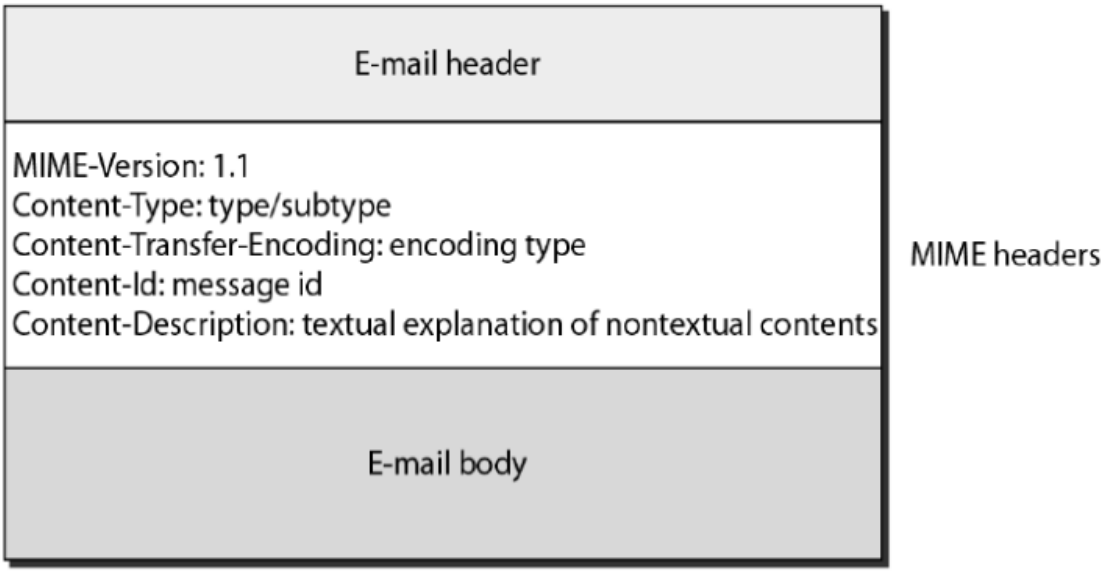


MIME HEADERS

¬ Using headers, MIME describes the type of message content and the encoding used.

¬ Headers defined in MIME are:

- MIME-Version- current version, i.e., 1.1
- Content-Type - message type (text/html, image/jpeg, application/pdf)
- Content-Transfer-Encoding - message encoding scheme (eg base64).
- Content-Id - unique identifier for the message.
- Content-Description - describes type of the message body.



MIME CONTENT TYPES

¬ There are seven different major types of content and a total of 14 subtypes.

¬ In general, a content type declares the general type of data, and the subtype specifies a

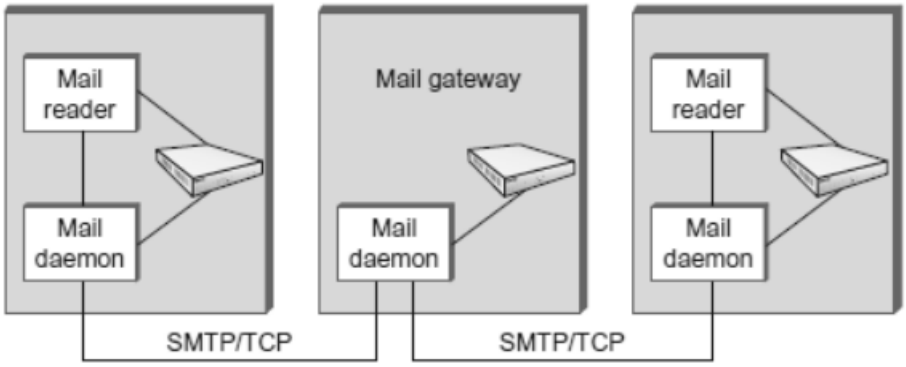
particular format for that type of data.
¬ MIME also defines a multipart type that says how a message carrying more than one data type is structured.
¬ This is like a programming language that defines both base types (e.g., integers and floats) and compound types (e.g., structures and arrays).
¬ One possible multipart subtype is mixed, which says that the message contains a set of independent data pieces in a specified order.
¬ Each piece then has its own header line that describes the type of that piece.
¬ The table below lists the MIME content types:

Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to mixed subtypes, 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 (8-bit bytes)

ENCODING FORMATS OF MIME

¬ MIME uses various encoding formats to convert binary data into the ASCII character set.
¬ To transfer binary data, MIME offers five encoding formats which can be used in the header transfer-encoding:
• 7-bit : 7-bit text format (for messages without accented characters);
• 8-bit : 8-bit text format;
• quoted-printable : Quoted-Printable format, recommended for messages which use a 7-bit alphabet (such as when there are accent marks);
• base-64 : Base 64, for sending binary files as attachments;
• binary : binary format; not recommended.
¬ Since MIME is very open, it can use third-party encoding formats such as:
• BinHex : A proprietary format belonging to Apple
• Uuencode : for UNIX-to-UNIX encoding
• Xencode : for binary-to-text encoding

MESSAGE TRANSFER IN MIME



¬ MTA is a mail daemon (send mail) active on hosts having mailbox, used to send an email.
¬ Mail passes through a sequence of gateways before it reaches the recipient mail server.
¬ Each gateway stores and forwards the mail using Simple mail transfer protocol (SMTP).
¬ SMTP defines communication between MTAs over TCP on port 25.
¬ In an SMTP session, sending MTA is client and receiver is server. In each exchange:
¬ Client posts a command (HELO, MAIL, RCPT, DATA, QUIT, VRFY, etc.)
¬ Server responds with a code (250, 550, 354, 221, 251 etc) and an explanation.
¬ Client is identified using HELO command and verified by the server
¬ Client forwards message to server, if server is willing to accept.
¬ Message is terminated by a line with only single period (.) in it.
¬ Eventually client terminates the connection.

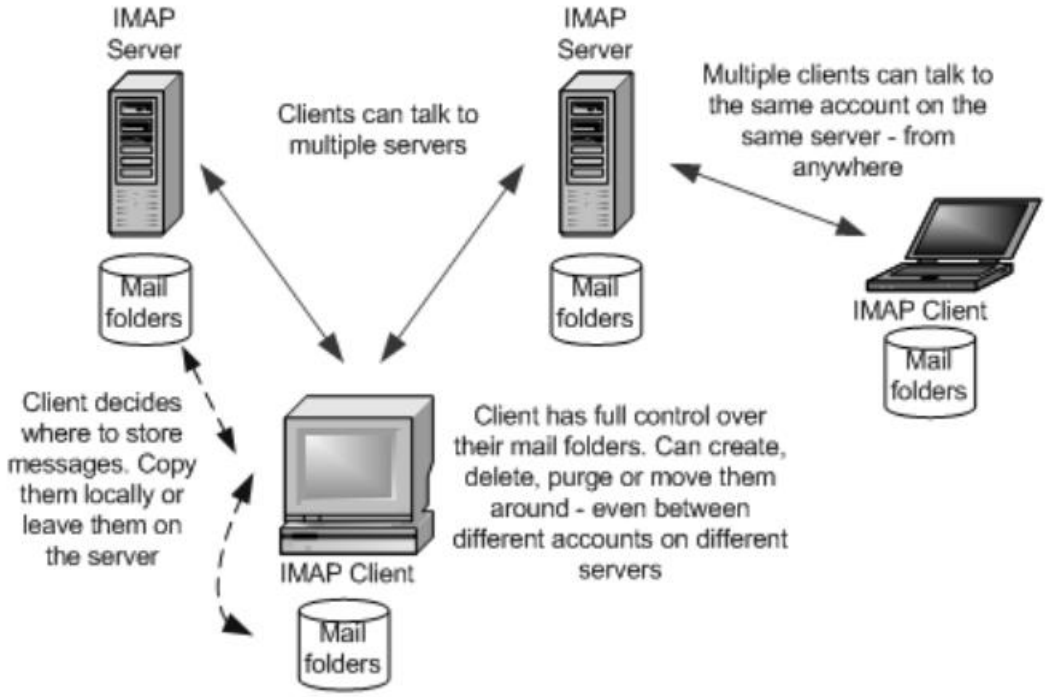
1.8.4.3 IMAP (INTERNET MAIL ACCESS PROTOCOL)

¬ IMAP is an Application Layer Internet protocol that allows an e-mail client to access e-mail on a remote mail server.
¬ It is a method of accessing electronic mail messages that are kept on a possibly shared mail server.
¬ IMAP is a more capable wire protocol.
¬ IMAP is similar to SMTP in many ways.
¬ IMAP is a client/server protocol running over TCP on port 143.
¬ IMAP allows multiple clients simultaneously connected to the same mailbox, and through flags stored on the server, different clients accessing the same mailbox at the same or different times can detect state changes made by other clients.
¬ In other words, it permits a "client" email program to access remote message stores as if they were local.
¬ For example, email stored on an IMAP server can be manipulated from a desktop computer at home, a workstation at the office, and a notebook computer while travelling, without the need to transfer messages or files back and forth between these computers.
¬ IMAP can support email serving in three modes:
♣ Offline
♣ Online
Users may connect to the server, look at what email is available, and access it online. This

looks to the user very much like having local spool files, but they're on the mail server.

♣ Disconnected operation

A mail client connects to the server, can make a "cache" copy of selected messages, and disconnects from the server. The user can then work on the messages offline, and connect to the server later and resynchronize the server status with the cache.



OPERATION OF IMAP

¬ The mail transfer begins with the client authenticating the user and identifying the mailbox they want to access.

¬ Client Commands

LOGIN, AUTHENTICATE, SELECT, EXAMINE, CLOSE, and LOGOUT

¬ Server Responses

OK, NO (no permission), BAD (incorrect command),

¬ When user wishes to FETCH a message, server responds in MIME format.

¬ Message attributes such as size are also exchanged.

¬ Flags are used by client to report user actions.

SEEN, ANSWERED, DELETED, RECENT

IMAP4

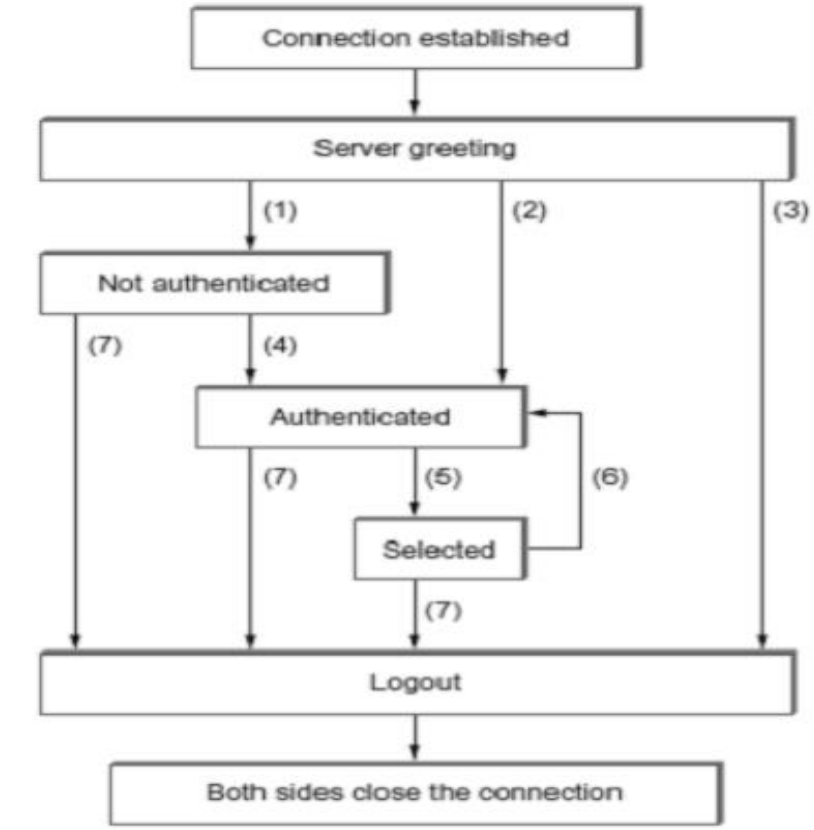
¬ The latest version is IMAP4. IMAP4 is more powerful and more complex.

¬ 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 high bandwidth 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



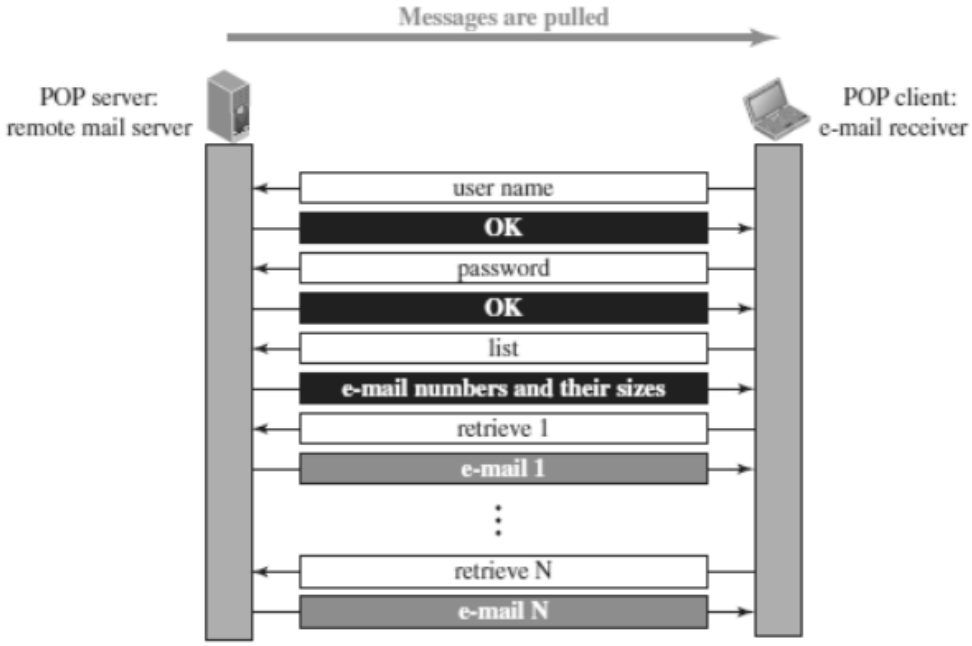
Advantages Of IMAP

¬ With IMAP, the primary storage is on the server, not on the local machine.
¬ Email being put away for storage can be foldered on local disk, or can be foldered on the IMAP server.
¬ The protocol allows full user of remote folders, including a remote folder hierarchy and multiple inboxes.
¬ It keeps track of explicit status of messages, and allows for user-defined status.
¬ Supports new mail notification explicitly.
¬ Extensible for non-email data, like netnews, document storage, etc.
¬ Selective fetching of individual MIME body parts.
¬ Server-based search to minimize data transfer.
¬ Servers may have extensions that can be negotiated.

1.8.4.4 POST OFFICE PROTOCOL (POP3)

¬ Post Office Protocol (POP3) is an application-layer Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection.
¬ There are two versions of POP.
• The first, called POP2, became a standard in the mid-80's and requires SMTP to send messages.
• The current version, POP3, can be used with or without SMTP. POP3 uses TCP/IP port 110.
¬ POP is a much simpler protocol, making implementation easier.
¬ POP supports offline access to the messages, thus requires less internet usage time
¬ POP does not allow search facility.
¬ In order to access the messages, it is necessary to download them.
¬ It allows only one mailbox to be created on server.
¬ It is not suitable for accessing non mail data.
¬ POP mail moves the message from the email server onto the local computer, although there is usually an option to leave the messages on the email server as well.
¬ POP treats the mailbox as one store, and has no concept of folders.
¬ POP works in two modes namely, delete and keep mode.
• In delete mode, mail is deleted from the mailbox after retrieval. The delete mode is normally used when the user is working at their permanent computer and can save and organize the received mail after reading or replying.
• In keep mode, mail after reading is kept in mailbox for later retrieval. The keep mode is normally used when the user accesses her mail away from their primary computer .

¬ POP3 client is installed on the recipient computer and POP server on the mail server.
¬ Client opens a connection to the server using TCP on port 110.
¬ Client sends username and password to access mailbox and to retrieve messages.



POP3 Commands

POP commands are generally abbreviated into codes of three or four letters
The following describes some of the POP commands:
1. UID - This command opens the connection
2. STAT - It is used to display number of messages currently in the mailbox
3. LIST - It is used to get the summary of messages
4. RETR -This command helps to select a mailbox to access the messages
5. DELE - It is used to delete a message
6. RSET - It is used to reset the session to its initial state
7. QUIT - It is used to log off the session

Advantages of IMAP over POP

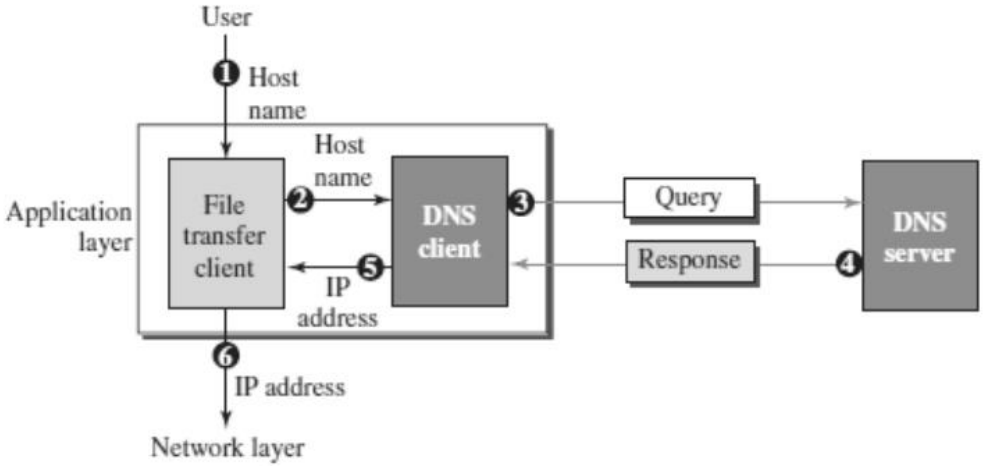
¬ IMAP is more powerful and more complex than POP.
¬ User can check the e-mail header prior to downloading.
¬ User can search e-mail for a specific string of characters prior to downloading.
¬ User can download partially, very useful in case of limited bandwidth.
¬ User can create, delete, or rename mailboxes on the mail server.

1.9 DNS (DOMAIN NAME SYSTEM)

¬ Domain Name System was designed in 1984.
¬ DNS is used for name-to-address mapping.
¬ The DNS provides the protocol which allows clients and servers to communicate with each other.
¬ Eg: Host name like www.yahoo.com is translated into numerical IP addresses like 207.174.77.131
¬ Domain Name System (DNS) is a distributed database used by TCP/IP applications to map

between hostnames and IP addresses and to provide electronic mail routing information.
¬ Each site maintains its own database of information and runs a server program that other systems across the Internet can query.

WORKING OF DNS



The following six steps shows the working of a DNS. It maps 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 address.
¬ 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.

Flat Name Space

- 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 main disadvantage of a flat name space is that it cannot be used in a large system such as Internet because it must be centrally controlled to avoid ambiguity and duplication.

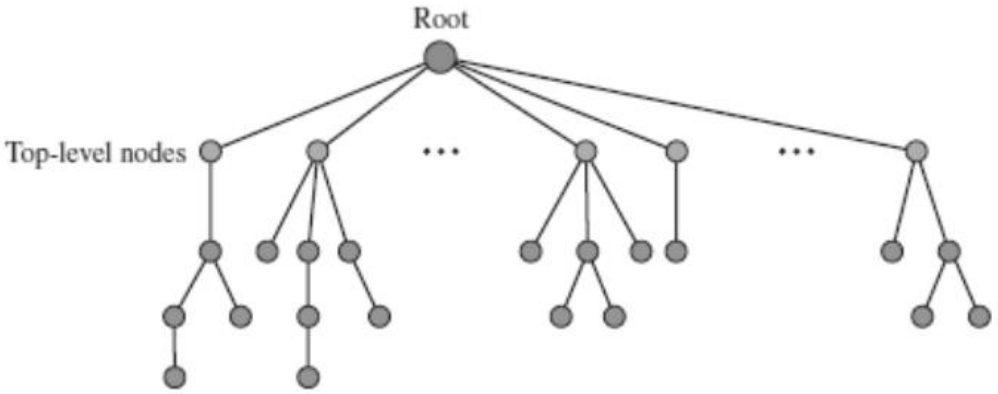
Hierarchical Name Space

- In a hierarchical name space, each name is made of several parts.
- The first part can define the organization, the second part can define the name, the third part can define departments, 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.
- The responsibility for the rest of the name can be given to the organization itself. Suffixes can be added to the name to define 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.
• The names are unique without the need to be assigned by a central authority.
• The central authority controls only part of the name, not the whole name.

DOMAIN NAME SPACE

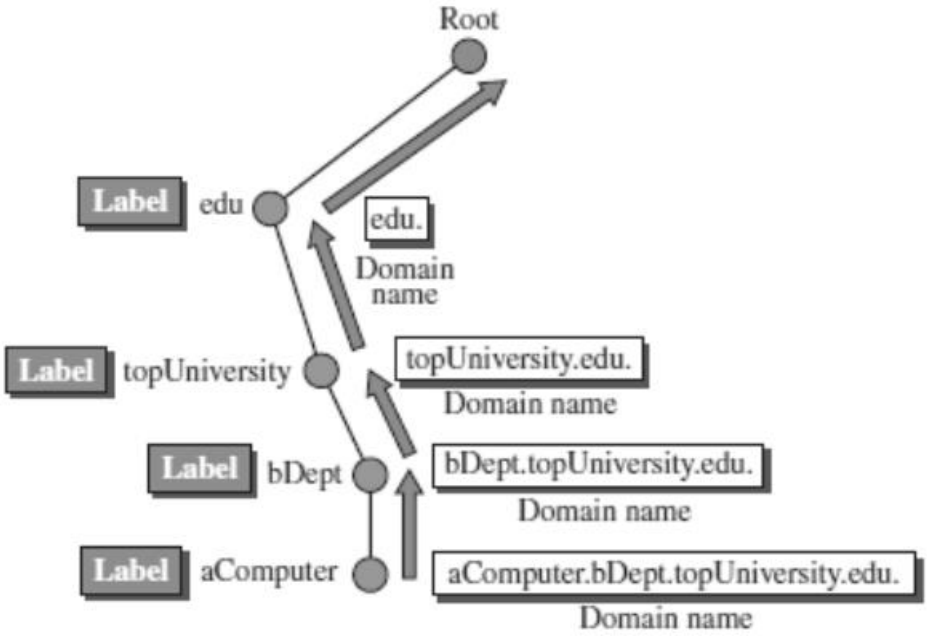
¬ To have a hierarchical name space, a domain name space was designed. In this design, the names are defined in an inverted-tree structure with the root at the top.
¬ Each node in the tree has a label, which is a string with a maximum of 63 characters.
¬ The root label is a null string.
¬ DNS requires that children of a node have different labels, which guarantees the uniqueness of the domain names.



¬ 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.

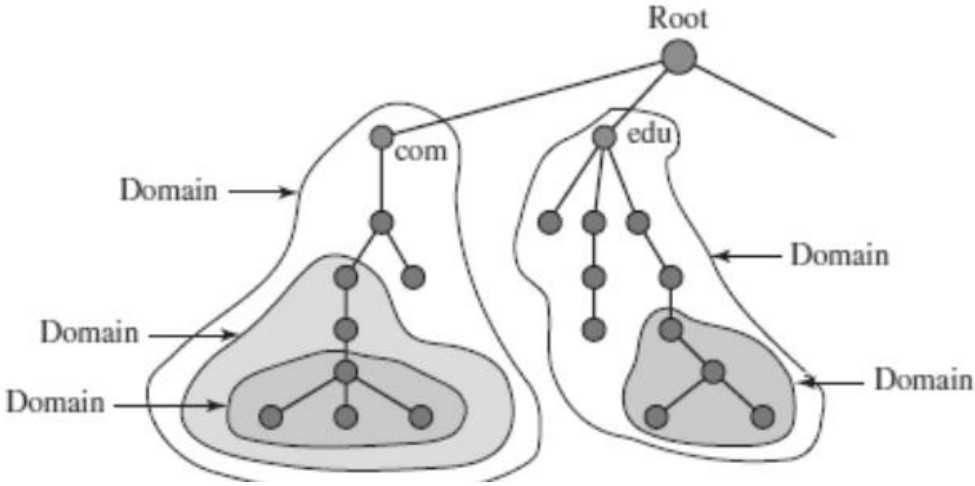
Domain Name

• Each node in the tree has a label called as 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).
• If a label is not terminated by a null string, it is called a partially qualified domain name (PQDN).



Domain

- A domain is a subtree of the domain name space.
- The name of the domain is the domain name of the node at the top of the sub- tree.
- A domain may itself be divided into domains.



DISTRIBUTION OF NAME SPACE

¬ The information contained in the domain name space must be stored.

¬ But it is very inefficient and also not reliable to have just one computer store such a huge amount of information.

¬ It is inefficient because responding to requests from all over the world, places a heavy load on the system.

¬ It is not reliable because any failure makes the data inaccessible.

¬ The solution to these problems is to distribute the information among many computers called DNS servers.

HIERARCHY OF NAME SERVERS

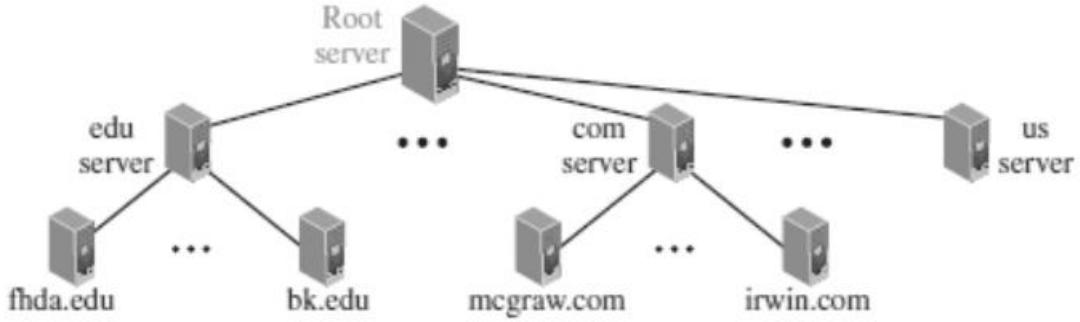
¬ The way to distribute information among DNS servers is to divide the whole space into many domains based on the first level.

¬ Let the root stand-alone and create as many domains 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.

¬ Thus we have a hierarchy of servers in the same way that we have a hierarchy of names.



ZONE

¬ What a server is responsible for, or has authority over, is called a zone.

¬ The server makes a database called a zone file and keeps all the information for every node under that domain.

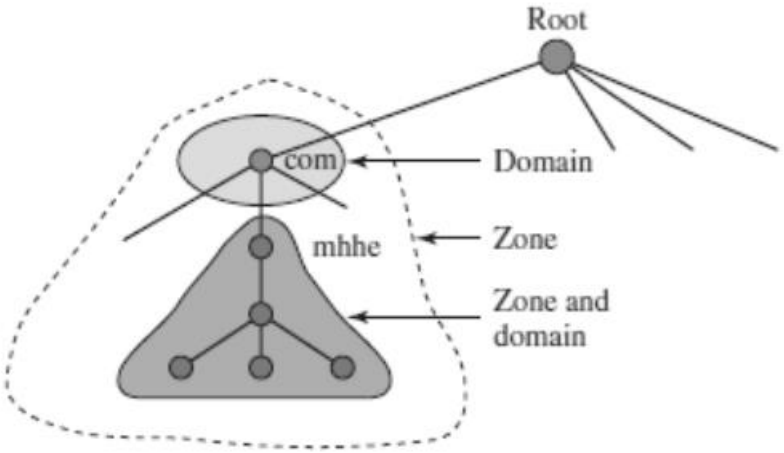
¬ If a server accepts responsibility for a domain and does not divide the domains into smaller domains, the domain and zone refer to the same thing.

¬ But if a server divides its domain into sub domains and delegates parts of its authority to other servers, domain and zone refer to different things.

¬ The information about the nodes in the sub domains is stored in the servers at the lower levels, with the original server keeping some sort of references to these lower level servers.

¬ But still, the original server does not free itself from responsibility totally.

¬ It still has a zone, but the detailed information is kept by the lower level servers.



ROOT SERVER

¬ A root sever 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.

¬ Currently there are more than 13 root servers, each covering the whole domain

name space.
¬ The servers are distributed all around the world.

PRIMARY AND SECONDARY SERVERS

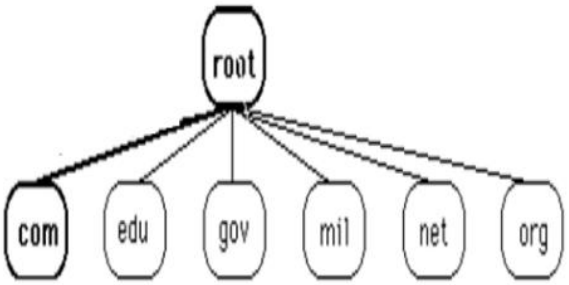
¬ DNS defines two types of servers: primary and secondary.
¬ A Primary Server is a server that stores a file about the zone for which it is an authority.
• Primary Servers are responsible for creating, maintaining, and updating the zone file.
• Primary Server stores the zone file on a local disc.
¬ A 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 disc.
¬ If updating is required, it must be done by the primary server, which sends the updated version to the secondary.
¬ A primary server loads all information from the disk file; the secondary server loads all information from the primary server.

DNS IN THE INTERNET

¬ DNS is a protocol that can be used in different platforms.
¬ In the Internet, the domain name space (tree) is divided into three different sections - Generic domains, Country domains, and Inverse domain.

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.
¬ The first level in the generic domains section allows seven possible three character levels.
¬ These levels describe the organization types as listed in following table.



.COM	Commercial Organizations
.EDU	Educational institutions
.GOV	Government institutions
.MIL	Military groups
.NET	Major network support centers
.INT	International Organizations
.ORG	Nonprofit Organizations

Country Domains

¬ The country domains section follows the same format as the generic domains but uses two characters for country abbreviations
¬ E.g.; in for India, us for United States etc) in place of the three character organizational abbreviation at the first level.
¬ Second level labels can be organizational, or they can be more specific, national designation.
¬ India for example, uses state abbreviations as a subdivision of the country domain us. (e.g., ca.in.)

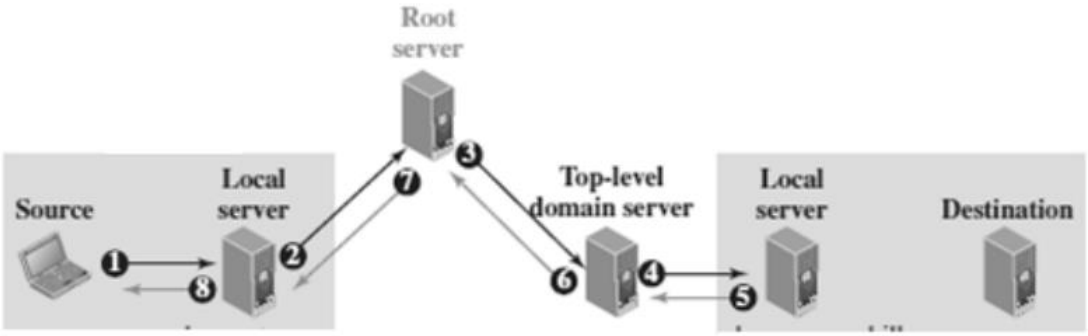
Inverse Domains

¬ Mapping an address to a name is called Inverse domain.
¬ The client can send an IP address to a server to be mapped to a domain name and it is called PTR(Pointer) query.
¬ To answer queries of this kind, DNS uses the inverse domain.

DNS RESOLUTION

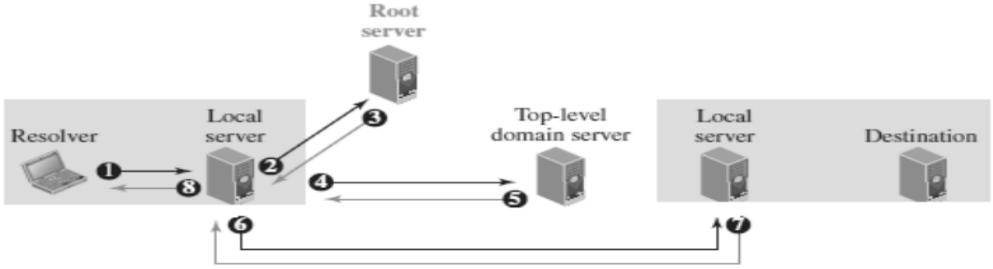
¬ Mapping a name to an address or an address to a name is called name address resolution.
¬ DNS is designed as a client server application.
¬ A host that needs to map an address to a name or a name to an address calls a DNS client named a Resolver.
¬ The Resolver accesses the closest DNS server with a mapping request.
¬ If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the information.
¬ After the resolver receives the mapping, it interprets the response to see if it is a real resolution or an error and finally delivers the result to the process that requested it.
¬ A resolution can be either recursive or iterative.

Recursive Resolution



• The application program on the source host calls the DNS resolver (client) to find the IP address of the destination host. The resolver, which does not know this address, sends the query to the local DNS server of the source (Event 1)
• The local server sends the query to a root DNS server (Event 2)
• The Root server sends the query to the top-level-DNS server(Event 3)
• The top-level DNS server knows only the IP address of the local DNS server at the destination. So it forwards the query to the local server, which knows the IP address of the destination host (Event 4)
• The IP address of the destination host is now sent back to the top-level DNS server(Event 5) then back to the root server (Event 6), then back to the source DNS server, which may cache it for the future queries (Event 7), and finally back to the source host (Event 8)

Iterative Resolution



• In iterative resolution, each server that does not know the mapping, sends the IP address of the next server back to the one that requested it.
• The iterative resolution takes place between two local servers.
• The original resolver gets the final answer from the destination local server.
• The messages shown by Events 2, 4, and 6 contain the same query.
• However, the message shown by Event 3 contains the IP address of the top- level domain server.
• The message shown by Event 5 contains the IP address of the destination local DNS server
• The message shown by Event 7 contains the IP address of the destination.
• When the Source local DNS server receives the IP address of the destination, it sends it to the resolver (Event 8).

DNS CACHING

¬ Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address.
¬ DNS handles this with a mechanism called caching.
¬ When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.
¬ If the same or another client asks for the same mapping, it can check its cache memory and resolve the problem.
¬ However, to inform the client that the response is coming from the cache memory and not from an authoritative source, the server marks the response as unauthoritative.
¬ Caching speeds up resolution. Reduction of this search time would increase efficiency, but it can also be problematic.
¬ If a server caches a mapping for a long time, it may send an outdated mapping to the client.
¬ To counter this, two techniques are used.
⎪ First, the authoritative server always adds information to the mapping called time to live (TTL). It defines the time in seconds that the receiving server can cache the information. After that time, the mapping is invalid and any query must be sent again to the authoritative server.
⎪ Second, DNS requires that each server keep a TTL counter for each mapping it caches. The cache memory must be searched periodically and those mappings with an expired TTL must be purged.

DNS RESOURCE RECORDS (RR)

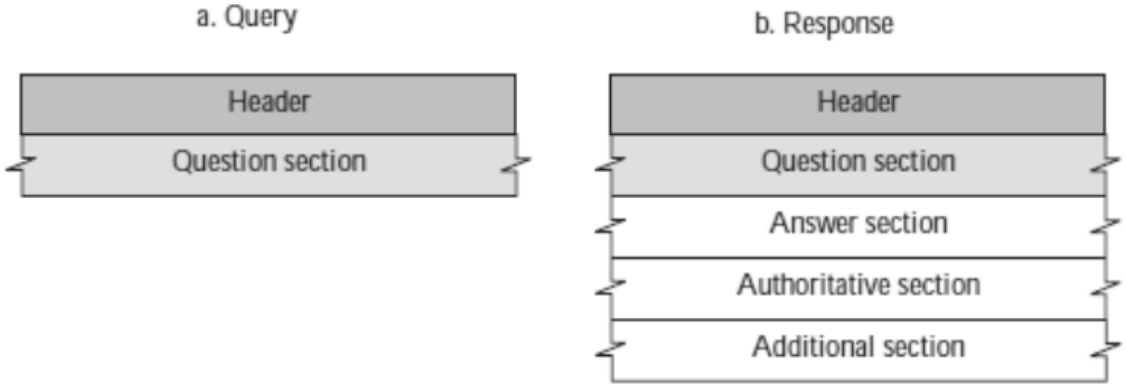
• The zone information associated with a server is implemented as a set of resource records.
• In other words, a name server stores a database of resource records.
• A resource record is a 5-tuple structure : (Domain Name, Type, Class, TTL, Value)
• The domain name identifies the resource record.
• The type defines how the value should be interpreted.
• The value defines the information kept about the domain name.
• The TTL defines the number of seconds for which the information is valid.
• The class defines the type of network.

Types of Resource Records

Type	*Interpretation of value*
A	A 32-bit IPv4 address
NS	Identifies the authoritative servers for a zone
CNAME	Defines an alias for the official name of a host
SOA	Marks the beginning of a zone
MX	Redirects mail to a mail server
AAAA	An IPv6 address

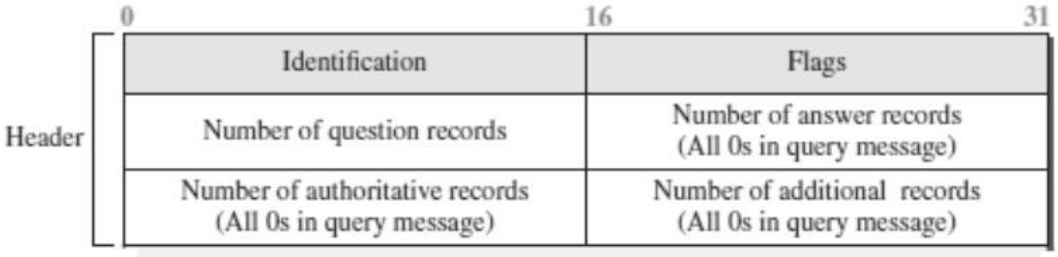
DNS MESSAGES

¬ DNS has two types of messages: query and response.
¬ Both types have the same format.
¬ The query message consists of a header and question section.
¬ The response message consists of a header, question section, answer section, authoritative section, and additional section .



¬ Header

• Both query and response messages have the same header format with some fields set to zero for the query messages.
• The header fields are as follows:



• The identification field is used by the client to match the response with the query.
• The flag field defines whether the message is a query or response. It also includes status of error.
• The next four fields in the header define the number of each record type in the message.

¬ Question Section

• The question section consists of one or more question records. It is present in both query and response messages.

¬ Answer Section

• The answer section consists of one or more resource records. It is present only in response

messages.

¬ ***Authoritative Section***

• The authoritative section gives information (domain name) about one or more authoritative servers for the query.

¬ ***Additional Information Section***

• The additional information section provides additional information that may help the resolver.

DNS CONNECTIONS

¬ DNS can use either UDP or TCP.
¬ In both cases the well-known port used by the server is port 53.
¬ UDP is used when the size of the response message is less than 512 bytes because most UDP packages have a 512-byte packet size limit.
¬ If the size of the response message is more than 512 bytes, a TCP connection is used.

DNS REGISTRARS

¬ New domains are added to DNS through a registrar. A fee is charged.
¬ A registrar first verifies that the requested domain name is unique and then enters it into the DNS database.
¬ Today, there are many registrars; their names and addresses can be found at http://www.intenic.net
¬ To register, the organization needs to give the name of its server and the IP address of the server.
¬ For example, a new commercial organization named wonderful with a server named ws and IP address 200.200.200.5, needs to give the following information to one of the registrars: Domain name: ws.wonderful.com IP address: 200.200.200.5.

DDNS (DYNAMIC DOMAIN NAME SYSTEM)

¬ In DNS, when there is a change, such as adding a new host, removing a host, or changing an IP address, the change must be made to the DNS master file.
¬ The DNS master file must be updated dynamically.
¬ The Dynamic Domain Name System (DDNS) is used for this purpose.
¬ In DDNS, when a binding between a name and an address is determined, the information is sent to a primary DNS server.
¬ The primary server updates the zone.
¬ The secondary servers are notified either actively or passively.
¬ In active notification, the primary server sends a message to the secondary servers about the change in the zone, whereas in passive notification, the secondary servers periodically check for any changes.
¬ In either case, after being notified about the change, the secondary server requests information about the entire zone (called the zone transfer).
¬ To provide security and prevent unauthorized changes in the DNS records, DDNS can use an authentication mechanism.

DNS SECURITY

¬ DNS is one of the most important systems in the Internet infrastructure; it provides crucial services to Internet users.

¬ Applications such as Web access or e-mail are heavily dependent on the proper operation of DNS.

¬ DNS can be attacked in several ways including:

- Attack on Confidentiality - The attacker may read the response of a DNS server to find the nature or names of sites the user mostly accesses. This type of information can be used to find the user's profile. To prevent this attack, DNS messages need to be confidential.
- Attack on authentication and integrity - The attacker may intercept the response of a DNS server and change it or create a totally new bogus response to direct the user to the site or domain the attacker wishes the user to access. This type of attack can be prevented using message origin authentication and message integrity.
- Attack on denial-of-service - The attacker may flood the DNS server to overwhelm it or eventually crash it. This type of attack can be prevented using the provision against denial-of-service attack.

¬ To protect DNS, IETF has devised a technology named DNS Security (DNSSEC) that provides message origin authentication and message integrity using a security service called digital signature.

¬ DNSSEC, however, does not provide confidentiality for the DNS messages.

¬ There is no specific protection against the denial-of-service attack in the specification of DNSSEC. However, the caching system protects the upper- level servers against this attack to some extent.

1.10 SNMP (SIMPLE NETWORK MANAGEMENT PROTOCOL)

¬ The Simple Network Management Protocol (SNMP) is a framework for managing devices in an internet using the TCP/IP protocol suite.

¬ SNMP is an application layer protocol that monitors and manages routers, distributed over a network.

¬ It provides a set of operations for monitoring and managing the internet.

¬ SNMP uses services of UDP on two well-known ports: 161 (Agent) and 162 (manager).

¬ SNMP uses the concept of manager and agent.

SNMP MANAGER

- A manager is a host that runs the SNMP client program
- The manager has access to the values in the database kept by the agent.

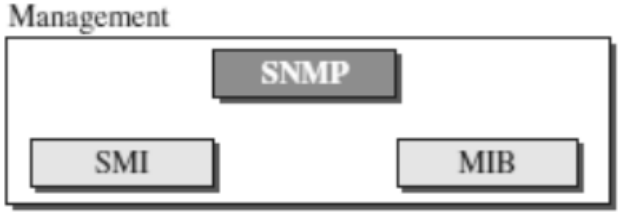
• A manager checks the agent by requesting the information that reflects the behavior of the agent.
• A manager also forces the agent to perform a certain function by resetting values in the agent database.
• For example, a router can store in appropriate variables the number of packets received and forwarded.
• The manager can fetch and compare the values of these two variables to see if the router is congested or not.

SNMP AGENT

• The agent is a router that runs the SNMP server program.
• The agent is used to keep the information in a database while the manager is used to access the values in the database.
• For example, a router can store the appropriate variables such as a number of packets received and forwarded while the manager can compare these variables to determine whether the router is congested or not.
• Agents can also contribute to the management process.
• A server program on the agent checks the environment, if something goes wrong, the agent sends a warning message to the manager.

SNMP MANAGEMENT COMPONENTS

• Management of the internet is achieved through simple interaction between a manager and agent.
• Management is achieved through the use of two protocols:
o Structure of Management Information (SMI)
o Management Information Base (MIB).



Structure of Management Information (SMI)

• To use SNMP, we need rules for naming objects.
• SMI is a protocol that defines these rules.
• SMI is a guideline for SNMP
• It emphasizes three attributes to handle an object: name, data type, and encoding method.
• Its functions are:
ϖ To name objects.
ϖ To define the type of data that can be stored in an object.
ϖ 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 tree structure starts with an unnamed root. Each object can be defined using a sequence

of integers separated by dots.
⎪ The tree structure can also define an object using a sequence of textual names separated by dots.

Type of data

⎪ The second attribute of an object is the type of data stored in it.
⎪ To define the data type, SMI uses Abstract Syntax Notation One (ASN.1) definitions.
⎪ SMI has two broad categories of data types: simple and structured.
⎪ The simple data types are atomic data types. Some of them are taken directly from ASN.1; some are added by SMI.
⎪ SMI defines two structured data types: sequence and sequence of.

- ♣ Sequence - A sequence data type is a combination of simple data types, not necessarily of the same type.
- ♣ Sequence of - A sequence of data type is a combination of simple data types all of the same type or a combination of sequence data types all of the same type.

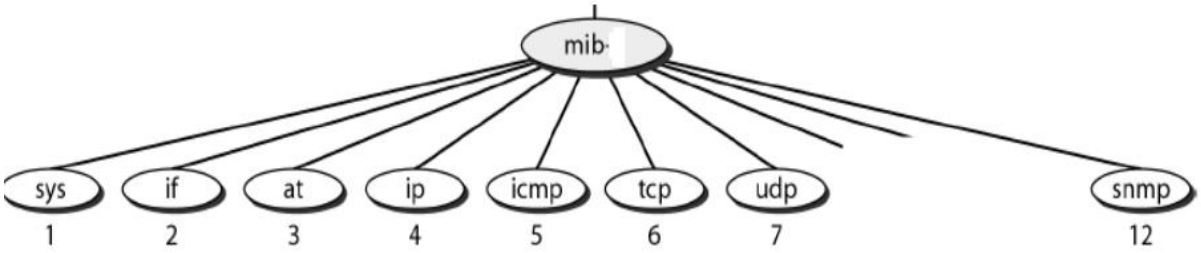
Encoding data

⎪ SMI uses another standard, Basic Encoding Rules (BER), to encode data to be transmitted over the network.
⎪ BER specifies that each piece of data be encoded in triplet format (TLV): tag, length, value

Management Information Base (MIB)

The Management Information Base (MIB) is the second component used in network management.

- Each agent has its own MIB, which is a collection of objects to be managed.
- MIB classifies objects under groups.



MIB Variables

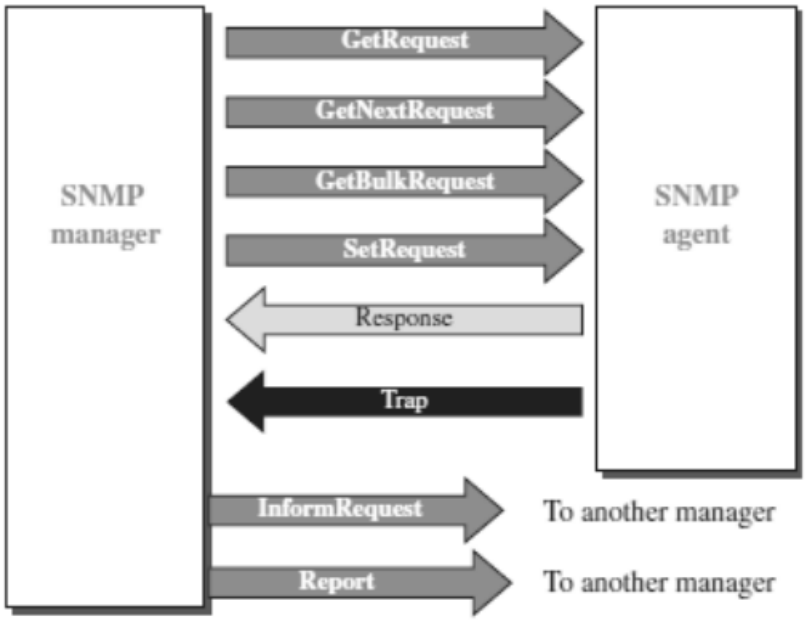
MIB variables are of two types namely simple and table.

- Simple variables are accessed using group-id followed by variable-id and 0
- Tables are ordered as column-row rules, i.e., column by column from top to bottom. Only leaf elements are accessible in a table type.

SNMP MESSAGES/PDU

SNMP is request/reply protocol that supports various operations using PDUs.
SNMP defines eight types of protocol data units (or PDUs):

GetRequest, GetNext-Request, GetBulkRequest, SetRequest, Response, Trap, InformRequest, and Report



GetRequest

♣ The GetRequest PDU is sent from the manager (client) to the agent (server) to retrieve the value of a variable or a set of variables.

GetNextRequest

♣ The GetNextRequest PDU is sent from the manager to the agent to retrieve the value of a variable.

GetBulkRequest

♣ The GetBulkRequest PDU is sent from the manager to the agent to retrieve a large amount of data. It can be used instead of multiple GetRequest and GetNextRequest PDUs.

SetRequest

♣ The SetRequest PDU is sent from the manager to the agent to set (store) a value in a variable.

Response

♣ The Response PDU is sent from an agent to a manager in response to

GetRequest or GetNextRequest. It contains the value(s) of the variable(s) requested by the manager.

Trap

♣ The Trap PDU is sent from the agent to the manager to report an event. For example, if the agent is rebooted, it informs the manager and reports the time of rebooting.

InformRequest

♣ The InformRequest PDU is sent from one manager to another remote manager to get the value of some variables from agents under the control of the remote manager. The remote manager responds with a Response PDU.

Report

♣ The Report PDU is designed to report some types of errors between managers.