Unit I

Chapter 1

Application Layer

Syllabus

Client Server Paradigm : Communication using TCP and UDP, Peer to peer paradigm, **Application Layer Protocols :** DNS, FTP, TFTP, HTTP, SMTP, POP, IMAP, MIME, DHCP, TELNET.

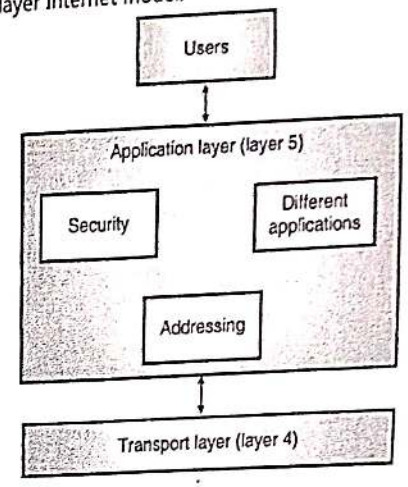
Chapter Contents

1.1 Introduction :

- Application layer is the topmost layer in the TCP/IP protocol suite.
- The hardware and software of the Internet was designed and developed for providing various types of services at the application layer.
- All the other layers (4 of them) make these services possible.
- We will discuss various services provided at the application layer first (in this chapter) and later on study the supporting role of the other layers, providing these services.
- Many application programs have been created and used during the lifetime of the Internet. Some of them could never become standards.
- Some others have become obsolete. Some have been modified, other have been replaced by new ones. But some applications have survived the test of time and have become standard applications.
- Everyday new application protocols are being added to Internet.
- The Internet can provide services via two types of applications :
  1. The traditional applications.
  2. The new applications.
- The traditional applications make use of the **client server** paradigm whereas the new applications are based on the **peer-to-peer** paradigm.
- The application layer provides communication with the help of a **logical connection** which is an **imaginary connection** between the application layers of the two communicating computers. This is not the physical connection.
- The actual communication however involves all the lower layer and different types of devices such as routers, switches etc.

1.1.1 Position of Application Layer :

- The application layer is the topmost (fifth layer) of the Internet model. This is layer where all the interesting applications are found.
- People can use the Internet due to the presence of application layer.
- The layers below the application layer provide reliable transport but they do not do any real work for the users.
- In other words, the other four layers are created so that people can use the various application programs. Fig. 1.1.1 shows the position of application layer in the 5-layer Internet model.



(G-629)**Fig. 1.1.1 : Position of application layer**

- The application layer provides services to the users. The users can be humans or software. It enables the user to access the network.
- The application layer receives services from the transport layer.

Support protocols :

- For the real applications in the application layer to function, there is a need of support protocols.
- The three areas or protocols required for such support are :
  1. Network security.
  2. Domain Name Service (DNS).
  3. Network management.
- Security is not a single protocol but it contains a large number of concepts and protocols used for providing privacy.
- DNS is used to handle naming or addressing within the Internet. The third support protocol is network management.
- In this chapter we are going to discuss some common client - server applications that are used in the Internet. Some of the important applications discussed in this chapter are : DNS, FTP, HTTP, SMTP and MIME.

1.2 Providing Services :

- All the communication networks which were designed to be used in the era prior to the Internet era were designed to provide a specific type of service.

- An example of such a service is the telephone service. The network for telephony was originally designed to provide only the voice service. Later on the same network was used to provide some other services such as the FAX.
- In a similar manner, the Internet also was designed for providing service to the users all over the world. But the Internet is more flexible than the other services such as postal service or telephone service, due to the layered architecture of TCP/IP suite.
- Application layer being the topmost layer in the TCP/IP suite, is slightly different from the other layers.
- The application layer protocols only take services from the other layer protocols but they do not provide any service to the protocols belonging to the other layers in TCP/IP suite.
- Therefore it is easily possible to add or remove protocols to/from the application layer. This layer is the only layer which can provide services to the Internet users.
- Due to the flexibility of the application layer, it is possible for us to add new application protocols to the Internet.

1.2.1 Standard and Non-standard Protocols :

- The protocols belonging to the first four layers of the TCP/IP suite have to be standardized and documented in order to ensure proper operation of the Internet.
- These protocols are generally included in the package along with an operating system such as windows or UNIX.
- However the application programs can be either standard or nonstandard, for ensuring flexibility.

1. Standard Protocols (Application Layer) :

- In our day to day life, we use several application layer programs for our interaction with the Internet. These programs are standardized and well documented by the Internet authorities.
- Each standard protocol is in the form of a pair of computer programs.
- These programs have been designed to interact with the user and the transport layer so as to provide a specific service to the user.

2. Nonstandard Protocols (Application Layer) :

- By writing two programs which can interact with a user and the transport layer to provide a specific service to the user, any programmer can create a nonstandard application layer program.
- The creation of a nonstandard protocol does not need any approval of the Internet authorities if it is used privately.
- The Internet has become so popular because of these nonstandard application layer protocols.

1.3 Application Layer Paradigms :

- During the life time of the Internet, three different paradigms have been developed.
- They are as follows :
  1. Client-server paradigm.
  2. Peer to peer paradigm.
  3. Mixed Paradigm.

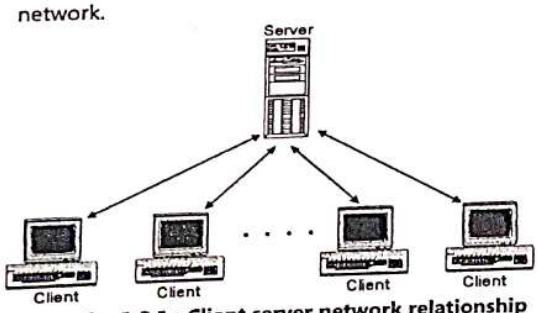
1.3.1 Traditional Paradigm : Client Server :

- The **client-server paradigm** is a traditional application layer paradigm which was the most popular paradigm until a few years ago.

Server and client processes :

- An application program called as the server process is basically the service provider in this paradigm.
- The server process runs continuously and waits for another application program called as the **client process** to make a connection through the Internet to ask for a service.
- Some server processes have been designed to provide some specific type of services. The server processes are supposed to run continuously but the client process does not run continuously.
- In fact it is started when the client needs some service from a server process. A server process can provide the same specific service to a number of client processes which request for that service.
- In computer networking the computers connected to the Internet are known as the **end systems.** The examples of end systems are as follows :
  1. Desktop computers
  2. PCs
  3. Workstations
  4. Household applications
  5. Web TVs and set top boxes
  6. Digital cameras etc.

- The end systems are also known as **hosts** because they run application programs such as Web browser program, or a Web server program etc.
- Hosts can be of two different categories as follows :
  1. Client 2. Server
- In client-server network relationships, some computers act as server and other act as clients.
- A **server** is a computer, which makes the network resources available to other computers when they request it.
- It also provides some services to them. A **client** is the computer running a program that requests the service from a server.
- Local Area Networking (LAN) uses the client-server network relationship for its operation.
- You can construct a client server network by using one or more powerful computers as a servers and the remaining computers as clients.
- Client-server network typically uses a directory service to store information about the network and its users.
- All available network resources such as files, directories, applications and shared devices, are centrally managed and hosted by the server and then are accessed by client in a client-server network.
- Fig. 1.3.1 shows client-server network relationship. The server provides security and administration of the network.



(G-41) **Fig. 1.3.1 : Client server network relationship**

- In client-server networks the processing tasks are divided between clients and servers.
- Clients request services such as file storage and printing and servers deliver them.

Applications :

- The following traditional services are still using the client server paradigm for their operation :
  1. WWW : World wide web
  2. HTTP : Hyper Text Transfer Protocol
  3. FTP : File Transfer Protocol
  4. email
- Some of these protocols have been discussed in this chapter.

1.3.2 New Paradigm : Peer-to-Peer (P2P) :

- In response to the needs of some new applications on the Internet, a new paradigm called as peer to peer paradigm has emerged in recent days.
- It is also known as the **P2P** paradigm. Here the continuously running server process is not needed. Instead the responsibility of the server process is shared by the **peers.**
- Most of the Internet applications available today operate on the client-server paradigm. But gradually the **peer-to-peer (P2P) paradigm** also has gained some importance.
- The principle of P2P paradigm is that two peers (laptops, desktops or mainframes) can exchange services by communicating directly with each other.
- If the file requested by a client to server is a large file such as a music or video file, then it puts a lot of load on the server machine.
- In such situations the **P2P** paradigm becomes attractive. The P2P paradigm is also attractive in a situation in which two peers want to exchange files without involving the server.
- However it should be noted that the **P2P** paradigm does not ignore the client-server paradigm completely. Instead the P2P allows some users to share the duty of the server.
- Instead of sharing of a big file using client-server connection, the P2P paradigm will let the server download a part of that file and then share it among themselves.
- Thus in P2P paradigm the same computer has to sometimes behave like a client and at some other time like a server.
- In other words, the same computer will be a client for some applications for certain amount of time and server at other times.
- However such applications are not a part of the Internet, but they are controlled commercially.
- In P2P paradigm any computer connected to the Internet can provide service as well as request for a service.
- That means it can work as a **server** at one time and as a **client** at some other time.

- One of the best examples of Internet application in which the P2P paradigm is used is **Internet.**

Telephony :

- Another situation in which the P2P paradigm can be more useful is when one Internet users wants to share something (a file for example) with another Internet user.

Advantages :

- The main advantages of P2P paradigm are as follows :
  1. It is easily scalable.
  2. It is cost effective because an expensive server need not be used.

Disadvantages :

- Along with the above stated advantages, there are some drawbacks of P2P paradigm :
  1. Providing a secured communication is difficult.
  2. This paradigm cannot be used by all the Internet applications.

Applications :

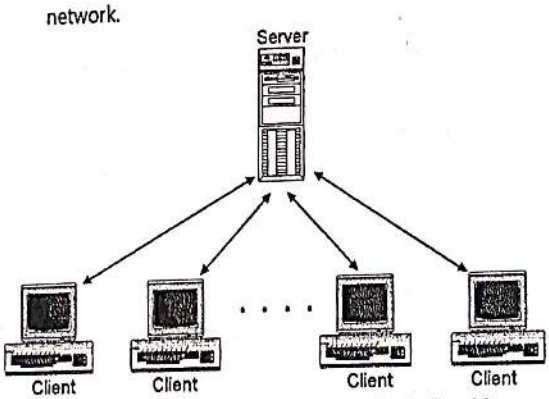
- The following Internet applications use the P2P paradigm :
  1. Skype
  2. Internet telephony
  3. IPTV.

1.3.3 Mixed Paradigm :

- In order to get the benefits of both the paradigms, some applications may try to use the mixture of the two paradigms.

1.4 Client Server Paradigm :

- In computer networking the computers connected to the Internet are known as the **end systems.**
- The examples of end systems are as follows :
  1. Desktop computers
  2. PCs
  3. Workstations
  4. Household applications
  5. Web TVs and set top boxes
  6. Digital cameras etc.
- The end systems are also known as **hosts** because they run application programs such as Web browser program, or a Web server program etc.
- Hosts can be of two different categories as follows :
  1. Client 2. Server
- In client-server network relationships, some computers act as server and other act as clients. A **server** is a computer, that makes the network resources available to other computers when they request it. It also provides some services to them. A **client** is the computer running a program that requests the service from a server.
- Local Area Networking (LAN) uses the client-server network relationship for its operation. You can construct a client server network by using one or more powerful computers as a servers and the remaining computers as clients. Client-server network typically uses a directory service to store information about the network and its users.
- All available network resources such as files, directories, applications and shared devices, are centrally managed and hosted by the server and then are accessed by client in a client-server network.
- Fig. 1.4.1 shows client-server network relationship. The server provides security and administration of the network.



(G-41) Fig. 1.4.1 : Client server network relationship

- In client-server networks the processing tasks are divided between clients and servers. Clients request services such as file storage and printing and servers deliver them.

Client :

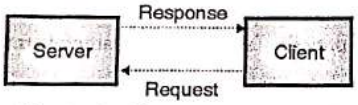
- The individual workstations in the network are called as the clients. A client can also be a mobile PC, PDA and so on.
- In short we can define a **client** as a program running on the local machine which requests some services from the server.
- It is said that the client program is a **finite** program. We have discussed it later on in this chapter.

Server :

- The central computer which is more powerful than the clients and which allows the clients to access its softwares and database is called as the server.
- Server computers typically are more powerful than client computers or are optimised to function as servers.
- No user can access the resources of the servers until he has been authenticated (permitted) by the server to do so.
- Generally we can define as **server** as a program which is running on the remote server computer to provide service to all the clients. It only initiates a service when requested by that client computer.
- The server program is called as an **infinite** program. We have discussed the reason for it later on in this chapter.

Communication in client-server configuration :

- Fig. 1.4.2 explains the principle of communication in the client server configuration.
- The client places a request on the server machine when he wants an access to the centralised resources.



(G-42) **Fig. 1.4.2 : Client/server communication**

- The server responds to this request and sends the signal accordingly to the client as shown in Fig. 1.4.2.
- The software run at the client computer is called as client program. This software configures that particular computer to act as a client.
- Similarly the software run on the server computer is called as server program. It configures that particular computer to act as a server.
- A server program when started, will run infinitely unless it faces some problem. Therefore it is called as an **infinite** program.
- A server program waits for incoming requests from clients. On receiving a request, it will respond to the request in one of the following way :
  1. Iteratively 2. Concurrently
- A client program will be started by the user and gets automatically terminated when the service is complete. Therefore it is called as the **finite** program.
- Generally the communication with the server is initiated by the client by using the IP address of the remote machine and the well known port address of the specific server program which running on that machine.
- The request respond process in client - server communication can get repeated multiple times. But still this process will eventually come to an end.
- Therefore it is called as the finite process.

1.4.1 Concurrency :

- We will discuss the concurrency in the client as well as server because both can operate in the concurrent mode.

1.4.1.1 Concurrency in Clients :

- It is possible to run the clients on a machine either in the iterative mode or in the concurrent mode.
- The operation of clients in the **iterative** mode involves running them (clients) one by one. That means one client must first complete the cycle of start, run and terminate after which the machine can start running some other client. But this method is too time consuming.
- Therefore the computers now a days make use of the **concurrent** clients where the machine can run two or more clients simultaneously. This would save time to a great extent.

1.4.1.2 Concurrency in Servers :

- The severs can be of two types : namely **iterative** servers or **concurrent** servers. The iterative server is capable of processing only one request at a time.
- That means the server should first complete the cycle of receiving a request, processing it and sending the response to the requesting client, after which it can handle another request.
- However the **concurrent** server can handle many requests simultaneously on the time sharing basis.

Protocols used :

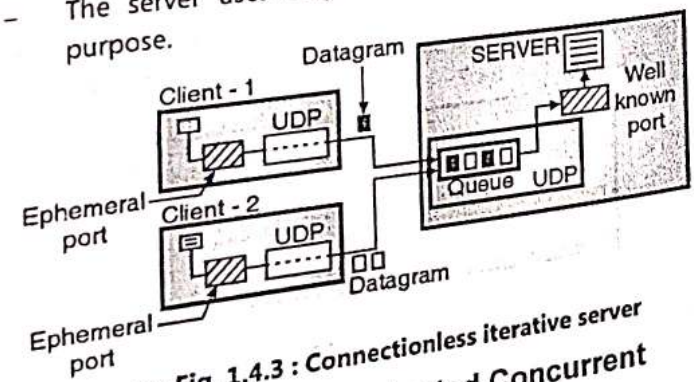
- The protocols that are commonly used by the servers are UDP, TCP or SCTP. Therefore the operation depends upon the following two factors :
  1. The transport layer protocol.
  2. The service method.

1.4.2 Types of Servers :

- We can classify the servers into following four types :
  1. Connectionless iterative (uses DDP).
  2. Connectionless concurrent.
  3. Connection - oriented iterative.
  4. Connection - oriented concurrent (Uses TCP/SCTP).
- Let us discuss them one by one.

1.4.2.1 Connectionless Iterative Server :

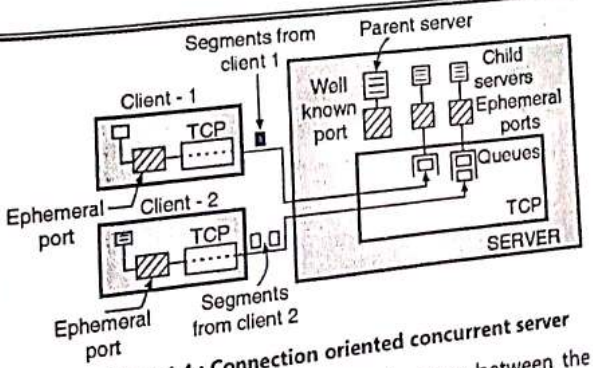
- This type of server processes one request at a time, and it uses UDP for communication.
- The principle of operation of this server is as shown in Fig. 1.4.3. As shown, a server receives a request through UDP and it sends its response back through UDP.
- The server pays attention to only one datagram at a time and ignores all others. These datagrams are stored in a queue as shown in Fig. 1.4.3 and wait for their turn to come. These datagrams could be from different clients or from the same client. But irrespective of that, the datagrams will be processed one by one.
- The server uses only one well known port for this purpose.



(G-1987) **Fig. 1.4.3 : Connectionless iterative server**

1.4.2.2 Connection - Oriented Concurrent Server :

- The concurrent connection - oriented servers generally use TCP or SCTP. This type of server can serve many clients simultaneously.
- The communication between the server and clients is connection oriented type. Therefore the first step will be to establish a connection between the server and each client.
- This connection will remain open as long as all the data stream is not processed. After processing the entire stream, the connection is terminated.
- In this type of communication, each connection between a client and server needs a separate port and many such connections are open simultaneously. Hence this type of server cannot use only one port. So many ports are required to be used. But a server can use only one well-known port.
- In order to solve this problem, they use **only one** well known port and multiple **ephemeral** ports as shown in Fig. 1.4.4. The server uses a will known port to accept the incoming requests.
- Once the connection is established with the requesting [illegible] server assigns a temporary port to this [illegible] the well known port free.



(G-1988) **Fig. 1.4.4 : Connection oriented concurrent server**

- Now the transfer of data can take place between the temporary port at a client and the temporary port at the server.
- Another client can now make a request at the well known port which has been freed.
- As shown in Fig. 1.4.4, the server creates **child processes** which are copies of the original **(parent)** process, in order to serve many clients simultaneously.
- Also observe in Fig. 1.4.4, that a separate queue has been assigned to each connection. The segments received from the client are stored in the appropriate queues.

1.4.3 Socket Interface :

- The question is how a client process communicate with a server process ?
- A computer does this by executing a program which is a set of sequentially written instructions.
- A computer program consists of different sets of instructions for mathematical operations, string manipulations, input / output access etc.
- If it is necessary for a program on one machine to communicate with another program running on some other machine, **then a set of new instructions should be written** in order to instruct the transport layer to establish a connection, complete the bidirectional data transfer and terminate the connection.
- Such a set of new instructions designed for interaction between two entities is called as **Interface.**

1.4.4 Types of Interface :

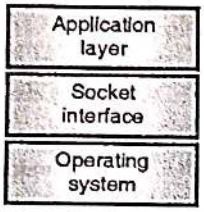
- Three different interfaces that have been designed for communication are :
  1. Socket interface.
  2. Transport layer interface.
  3. STREAM.
- Out of these we will discuss only the **socket interface.**

1.4.5 Socket Interface :

- We can understand the socket interface better if we learn more about the relationship between the operating system (Unix, Windows etc.) and the TCP/IP protocol suite.
- Refer Fig. 1.4.5 to understand the conceptual relationship between an operating system and TCP/IP suite.

Definition :

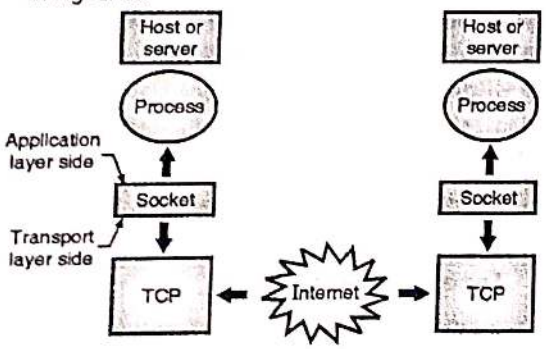
- We may define the **socket interface** as the set of instructions which helps an application access the services provided by the TCP/IP protocol suite. It is located between the application program and operating system.



(G-1989) **Fig. 1.4.5 : Relation between TCP/IP suite and operating system**

1.4.6 Socket :

- In most applications there exists a pair of communicating processes. They send messages to each other. These messages must travel the underlying network.
- The sending process sends messages into the network through its **socket** and receiving process receives messages from the network through its socket as shown in Fig. 1.4.6.



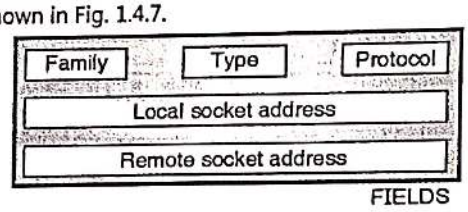
- Processes are controlled by application developers
- TCP is controlled by the operating system
- UDP can be used in place to TCP

(G-630) **Fig. 1.4.6 : Socket**

- Thus **socket** is defined as an interface between the application layer and the transport layer within a host.
- It is also called as the Application Programming Interface (API) between the application and the network.
- In Fig. 1.4.6 we have assumed that the transport protocol being used is TCP. But note that UDP can also be used. In the Internet, a socket is a software **data structure.**

Data structure :

- A socket is defined with the help of the format of data structure. This format of data structure is dependent on the language used by the processes.
- In C language the socket is defined as a five field structure. It is also called as **struct** or **record** and is as shown in Fig. 1.4.7.



(G-1990) **Fig. 1.4.7 : Socket data structure in C**

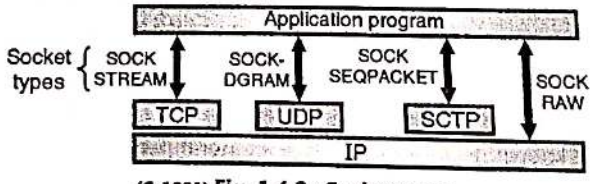
- It is important to understand that a programmer is not supposed to modify this structure, which is already defined as shown in Fig. 1.4.7.
- The programmer is supposed to only use the header file which contains the definition of the structure.
- The five fields defined in the data structure are as follows :

1. **Family :**

- This field in the data structure will define the protocol group eg. IPv4, IPv6, UNIX domain protocols etc.
- The family type used for TCP/IP can be defined by the constant IF_INET for IPv4 and IF_INET for IPv6 protocols.

2. **Type :**

- This field in the data structure will define four types of sockets as shown in Fig. 1.4.8.



(G-1991) **Fig. 1.4.8 : Socket types**

3. **Protocol :**

- This field in the data structure is used to define the protocol which uses the interface. For TCP/IP protocol the value set in the protocol field is 0.

4. Local socket address :

- This field in the data structure is used for defining the local socket address.
- A socket address is obtained by combining an IP address and a port number.

5. Remote socket address :

- This field is used for defining the remote socket address.

Functions :

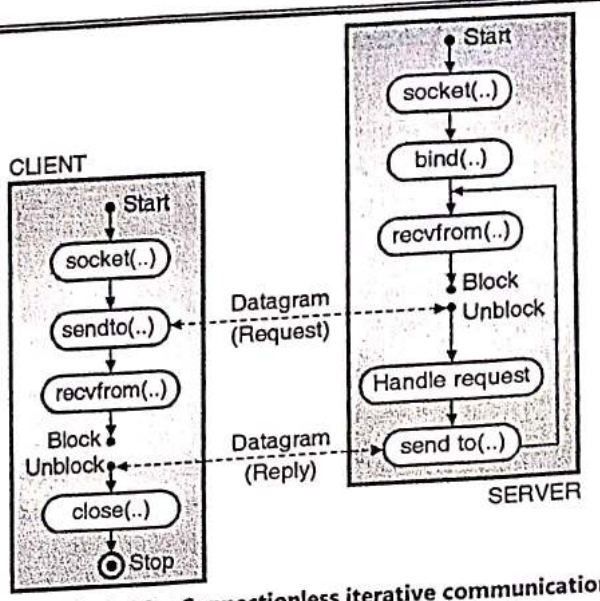
- A list of predefined functions is used to facilitate the interaction between a process and the operating system.
- These functions are combined to create processes.
- Various important functions are as follows :
  1. The socket function.
  2. The bind function.
  3. The connect function
  4. The listen function.
  5. The accept function.
  6. The fork function.
  7. The send and receive function.
  8. The sendto and recvfrom function.
  9. The close function.
  10. Byte ordering functions.
  11. Memory Management functions.
  12. Address conversion functions.

Header files :

- We need to use the header files in order to use the functions stated earlier. This header file is defined in a separate file named as **headerfiles.h**
- This file will then be included in the program so as to avoid inclusion of long lists of header files.
- All these header files may not be needed in all the programs but still they are recommended to be included.

1.4.7 Communication using UDP :

- A simplified flow diagram for the communication using UDP has been shown in Fig. 1.4.9.
- It diagrammatically illustrates the client - server communication with the help of UDP.



(G-1992) **Fig. 1.4.9 : Connectionless iterative communication using UDP**

- The connectionless iterative communication using UDP can be divided into two processes :
  1. Serve process 2. Client process.

1.4.8 Server Process :

- Refer Fig. 1.4.9. The server process will start first. The first step is that the server process calls the **socket** function for creating a socket.
- Then the server process calls the **bind** function which binds the socket to its well known port, and also to the IP address of computer which is running the server process.
- Next the **recvfrom** function is called by the server process which blocks the process until it receives the **request** datagram from the client. When the request datagram is received by the server process, the **recvfrom** function will unblock the server process, extracts the client socket address and address length from the request datagram, and retunes this information to the server process.
- The server process saves this information and calls a procedure or function in order to handle the request received from the client process.
- After readying the results it will call the function **sendto** and sends results to the requesting client process by making the use of the saved information.
- There is an infinite loop existing at the server process as the output of **sendto** block goes back to the input of **recvfrom** function as shown in Fig. 1.4.9. This infinite loop is used by the server to respond to the requests originated by the same or different clients.

1.4.9 Client Process :

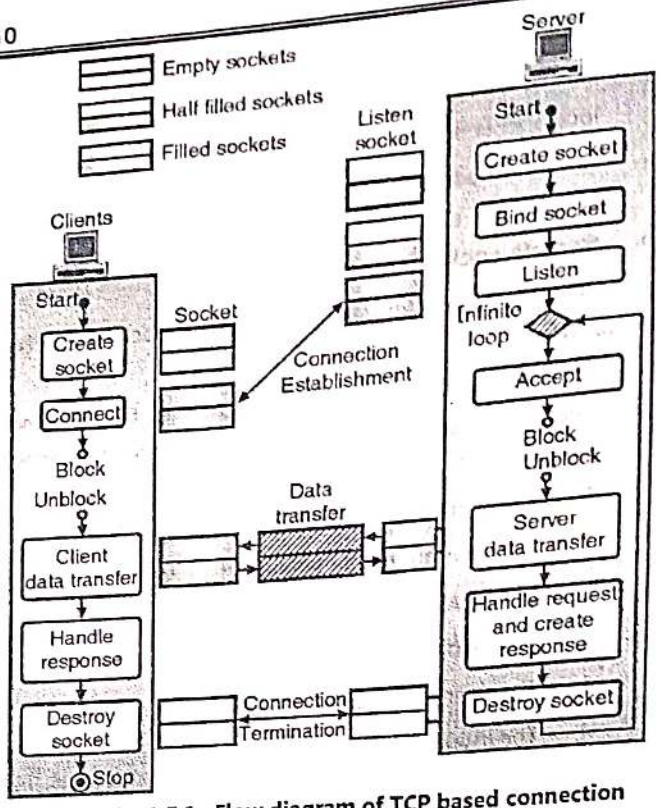
- Refer Fig. 1.4.9 to understand the sequence of events taking place at the client process which is much simpler as compared to the server process.
- First of all the client process calls the **socket** function for creating a socket. Next step is to call **sendto** function in order to pass the socket address of the server and the location of buffer.
- The UDP is supposed to take the data from this buffer and make the datagrams.
- As the next step, the client process calls the **recvfrom** function which blocks the client process until it receives the **response** message from the server process.
- As soon as the **response** from the server is received, the UDP delivers the received data to the client process.
- Due to this the **recv** function unblocks and delivers the received data to the client process.
- For all this discussion it was assumed that the client message is very small which can fit into one single datagram.
- But if the client message is long then we have to repeat the two functions **sendto** and **recvfrom.** But the server process is not aware of the multiple datagrams sent by the same client for the same communication. So it handles each request as an independent one.

1.5 Communication using TCP :

- After discussing the connectionless iterative communication using UDP, now let us discuss the connection oriented concurrent communication using TCP (the case of SCTP would be similar).
- The connection oriented concurrent communication using TCP can be divided into two processes :
  1. Server process 2. Client process.
- The flow diagram for TCP based communication is as shown in Fig. 1.5.1.

1.5.1 Server Process :

- Refer Fig. 1.5.1, in which it is the server process that starts first. The first step is that the server process calls the **socket** function for creating the socket. This socket is called as **listen socket** because it is going to be used only during the establishment of connection.
- As the next step, the server process calls the **bind** function, which will bind this connection to the socket address of the server computer.



(G-2203) **Fig. 1.5.1 : Flow diagram of TCP based connection oriented concurrent communication**

- The server function then calls the **accept** function which is basically a **blocking** function. It will block the server process until the TCP receives a connection request (SYN segment) from a client.
- On receiving the **request** from the client process, the **accept** function is unblocked and a new socket is created which is called as the **connect** socket, which carries the socket address of the requesting client.
- As soon as the **accept** function is unblocked the server understands that the requesting client needs some service.
- The server process or parent process now calls the **fork** function, in order to provide the concurrency.
- The **fork** function will create a **child process** which is a new process and it is an exact replica of the **parent process.**
- Thus after we call the **fork** function, two processes are running simultaneously (parent and child) i.e. concurrently, but each one is capable of handling different things.
- Each process now has two sockets namely **listen** and **connect** sockets. Now the parent process will handover the task of serving the existingly served client to the **child process** and calls the **accept** function again to

wait for the request from another client, because the newly created child process is now ready to serve the client who is already being served and make the parent process.

- The child process will first close the **listen socket** and then call the **recv** function so as to receive data from the client.
- The **recv** function is very similar to the **recvfrom** function is basically a blocking function. It is blocked upto the instant when a segment is received.
- As shown in Fig. 1.5.1, the child process uses a loop and keeps calling the **recv** function repeatedly as long as it does receive all the segments sent by the client.
- All this data is then given to a function called as **handle Request** by the child process for handling the request and send the results to the requesting client.
- The **send** function is then called for sending the results to the client.
- All the discussion till now was based upon certain assumption. They were as follows :
  1. We have used the simplest possible flow diagrams.
  2. The size of data sent to client is very small and we can send it in just one call of the **send** function.
- Actually if the data sent to the client is not small, the server may have to call the **send** function repeatedly.
  3. Third assumption was that the TCP is able to send the entire client data in one segment.
- In reality TCP may require several segments to send the client data.
- Therefore the client may not receive all the data it requested for in one segment.

1.5.2 The Client Process :

- Refer Fig. 1.5.1 to understand the client process. As per the flow graph given there, the first step the client takes is it calls the **socket** function for creating a socket.
- Next it (client process) calls the **connect** function for making a request to connect to the server.
- As we know, the **connect** function is a blocking function. It will remain blocked until the connection is established between client TCP and server TCP.
- On the return of **connect** function, the client calls the **send** function for sending data to the server.
- The server may need to call the **send** function only once or several times depending on the size of data. For calling the **send** function repeatedly, we have to use a loop in the flow diagram.
- Next the client calls the **recv** function. This function will remain blocked as long as the first segment of data does not arrive.
- The server may be able to send all the data by calling the **send** function only once but the TCP may not be able to send it using only one segment.
- Therefore the **recv** function may have to be called repeatedly by the client process to receive all the data.

1.5.3 Peer to Peer Paradigm :

- Most of the Internet applications available today operate on the client-server paradigm. But gradually the **peer-to-peer (P2P) paradigm** also has gained some importance.
- The principle of P2P paradigm is that two peers (laptops, desktops or mainframes) can exchange services by communicating directly with each other.
- If the file requested by a client to server is a large file such as a music or video file, then it puts a lot of load on the server machine.
- In such situations the **P2P** paradigm becomes attractive. The P2P paradigm is also attractive in a situation in which two peers want to exchange files without involving the server.
- However it should be noted that the **P2P** paradigm does not ignore the client-server paradigm completely. Instead the P2P allows some users to share the duty of the server.
- Instead of sharing of a big file using client-server connection, the P2P paradigm will let the server download a part of that file and then share it among themselves.
- Thus in P2P paradigm the same computer has to sometimes behave like a client and at some other time like a server.
- In other words, the same computer will be a client for some applications for certain amount of time and server at other times. However such applications are not a part of the Internet, but they are controlled commercially.

1.5.4 P2P File Sharing :

- P2P means process to process file sharing.
- The P2P file sharing is the most important Internet application because the highest amount of Internet traffic, corresponds to the P2P file sharing.
- Modern P2P file sharing system shares MP3 (3 to 8 M bytes), videos (10 to 1,000 M bytes), images, software documents etc.

- In this section, we will discuss the protocols and networking issues in P2P file sharing.
- Before going into details of P2P file sharing system, let us take an example. Suppose **Rahul** uses the P2P file sharing application for MP3 downloading. He runs the P2P file sharing software on his home PC (peer). He uses an ADSL connection to access the Internet. He shuts down his PC every night and does not have a hostname. So everytime he connects to the Internet the ISP will assign a new IP address to his PC.
- Suppose that Rahul is connected to the Internet and searching for the MP3 for a particular song of a particular artist.
- As soon as he goes into search, the P2P application displays a list of those peers who are currently connected to the Internet and have a copy of that song, for sharing.
- Each one of them is an ordinary PC owned by an ordinary Internet user like Rahul.
- Rahul then requests the required MP3 file from one of the peers say Preeti's PC. Then a direct TCP connection gets established between Rahul and Preeti's PC and the MP3 file is sent from Preeti's PC to Rahul's PC.
- If Preeti disconnects her PC from the Internet in the middle of this download, then Rahul's P2P file sharing software may attempt the remaining part of the MP3 file from the other peer.
- Also when the download from Preeti to Rahul is going on, some other user can download some other song from Rahul's PC.
- Thus the P2P file sharing allows direct sharing of information without any independent server getting involved. However P2P file sharing operates on the client server principle. The requesting person acts as a client and the requested user acts as the server. The file is sent using the File Transfer Protocol (FTP).
- In P2P file sharing system, typically a large number of users are connected to Internet and each user has objects such as MP3, videos, software and images for sharing.

1.6 Domain Name System (DNS) :

SPPU : Dec. 16, May 18, In Sem. March 19, Dec. 19

University Questions

Q. 1 What is domain name system ? Explain how a resolver looks up a remote name with suitable example. **(Dec. 16, 6 Marks)**

Q. 2 What is DNS ? What is server hierarchy ? Explain domain name resolution process. **(May 18, Dec. 19, 6 Marks)**

Q. 3 What is Domain Name system ? Explain how a resolver looks up a remote name with suitable example ? **(March 19, 6 Marks)**

Definition of DNS :

- Domain name system (DNS) is a system that translates domain names into IP addresses. The DNS servers are Internet's equivalent of a phone book. They maintain the directory of domain names and translate them into IP addresses

Addressing :

- For communication to take place successfully, the sender and receiver both should have addresses and they should be known to each other.
- The addressing in application program is different from that in the other layers.
- Each program will have its own address format. For example an e-mail address is like abc@vsnl.net whereas the address to access a web page is like http :// www.google.com/
- It is important to note that there is an alias name for the address of remote host. The application program uses an alias name instead of an IP address.
- This type of address is very convenient for the human beings to remember and use. But it is not suitable for the IP protocol.
- So the alias address has to be mapped to the IP address. For this an application program needs service of another entity.
- This entity is an application program called DNS. Note that DNS is not used directly by the user. It is used by another application programs for carrying out the mapping.

1.6.1 How does DNS Work ? :

SPPU : Dec. 08, May 09, Dec. 09, Dec. 12

University Questions

Q. 1 Explain how DNS service works. **(Dec. 08, May 09, Dec. 09, Dec. 12, 8 Marks)**

- To map a name onto an IP address, an application program calls a library procedure called the **resolver.** The name is passed on to the resolver as a parameter.
- The resolver sends a UDP packet to a local DNS server which looks up the name and returns the corresponding IP address to the resolver.

- The resolver then sends this address to the caller. Then the program can establish a TCP connection with the destination or sends in the UDP packets.

1.6.2 Name Space :

- The names assigned to machines should be selected carefully from the name space.
- There should be a complete control over the relation between the names and the IP addresses.
- The names and corresponding addresses are uniquely defined. A name space maps each address to a unique name.
- It can be arranged in two different ways :
  1. Flat name space.
  2. Hierarchical name space.

1.6.3 Flat Name Space :

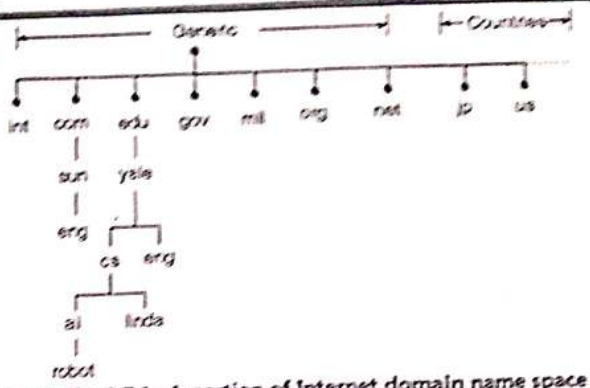
- In a flat name space, a name is assigned to every address. This type of name is simply the sequence of characters.
- That means it does not have any structure. The flat name space is not suitable for large systems like Internet, because there can be ambiguity and / or duplication.

1.6.4 Hierarchical Name Space :

- In the hierarchical name space, each name is made of many parts.
- The first part may correspond to the name of an institution, the second part may define the department and so on.
- The part that defines the nature of institution and name of institution is assigned by a central authority.
- The responsibility of deciding the rest of the name can be given to that institute itself.
- That institute can add suffix or prefix to the name for defining its host or resources.

1.7 Domain Name Space :

- Conceptually the Internet has been divided into hundreds of top level domains. Each domain covers many hosts.
- Each domain is divided into several sub domains and they are further partitioned and so on.
- These domains can be represented by a tree as shown in Fig. 1.7.1.

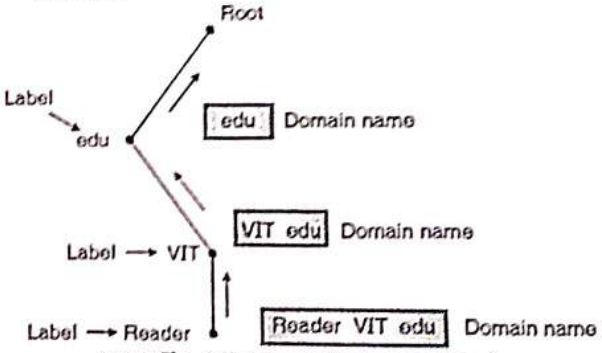


(G-631)**Fig. 1.7.1 : A portion of Internet domain name space**

- The top level domains are of two types namely generic and countries.

Generic domains :

- The generic domains are com (commercial), edu (educational institutions), gov (government), int (some international organizations), mil (military), net (network providers) and org (nonprofit organizations).
- The country domains include one entry for every country.
- Each domain is named by following an upward path. The components are separated by dots e.g. **eng.sun.com.** This is called hierarchical naming.
- Another example of hierarchical naming is shown in Fig. 1.7.2. The upward followed path has been shown by an arrow.



(G-632)**Fig. 1.7.2 : Domain names, labels and hierarchical naming**

Label :

- Each node in the tree has a label (or component) and it can be specified using up to 63 characters.
- If we had to remember the IP addresses of all of the Web sites we visit every day, we would all go nuts.
- Human beings just are not that good at remembering strings of numbers.
- We are good at remembering words, however, and that is where domain names come in.

- You probably have hundreds of domain names stored in your head. For example :
  - www.yahoo.com - the world's best-known name
  - www.mit.edu - a popular EDU name
  - encarta.msn.com - a Web server that does not start with www
  - www.bbc.co.uk - a name using four parts rather than three
  - ftp.microsoft.com - an FTP server rather than a Web server
- The COM, EDU and UK portions of these domain names are called the **top-level domain** or **first-level domain.**
- There are several hundred top-level domain names, including COM, EDU, GOV, MIL, NET, ORG and INT, as well as unique **two-letter combinations for every country.**
- Within every top-level domain there is a huge list of **second-level domains.** For example, in the COM first-level domain, you have got :
  - yahoo
  - msn
  - microsoft
  - plus millions of others.
- Every name in the COM top-level domain must be **unique,** but there can be duplication across domains.
- For example, **msn.com** and **msn.org** are completely different machines.
- In the case of bbc.co.uk, it is a third-level domain. Up to **127 levels** are possible, although more than four is rare.
- The left-most word, such as **www** or **encarta,** is the **host name.** It specifies the name of a specific machine (with a specific IP address) in a domain.
- A given domain can potentially contain millions of host names as long as they are all unique within that domain.

Absolute and relative domain names :

- Domain names can be of two types : absolute or relative.
- An absolute domain name always ends with a dot (or period as it was called). For example eng. sun. com. But the relative domain does not end with a dot.

Are domain names case sensitive ?

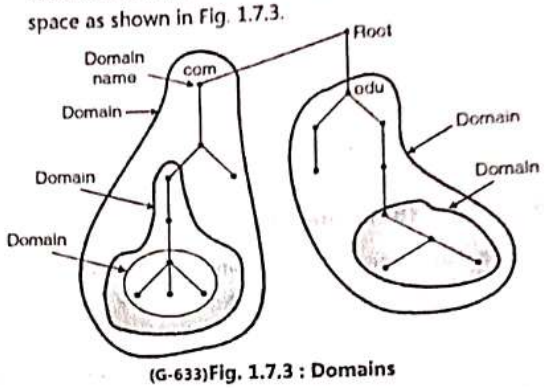
- No they are not case sensitive. So com and COM means the same thing.

How many characters ?

- Component names can have upto 63 characters and the full path name can at the most have 255 characters. Each domain controls how it allocates the domain under it.
- To create a new domain we have to take a permission of the domain in which it is to be included.

Domain :

- A domain can be defined as a subtree of the DNS name space as shown in Fig. 1.7.3.



(G-633)**Fig. 1.7.3 : Domains**

- The name of the domain is the domain name of the node at the top of the subtree as shown in Fig. 1.7.3. e.g. com or edu.
- A domain can be divided into subdomains as shown in Fig. 1.7.3. Note that the naming follows organizational boundaries, not physical networks.
- That means even if two different departments are located in the same building, they can have distinct domains.
- But the computers belonging to the same department kept in two different buildings will not have different domains.

1.8 Distribution of Name Space :

- The information contained in the domain name should be stored.
- But this is a huge information and if we store it on one computer then the system would be highly inefficient and unreliable.
- It will be an inefficient system because the system will be heavily loaded by the requests coming from all over the world.
- It will be unreliable because failure of one computer will make the data inaccessible. If we make a distributed name space then all these problems can be overcome.

1.8.1 Hierarchy of Name Servers :

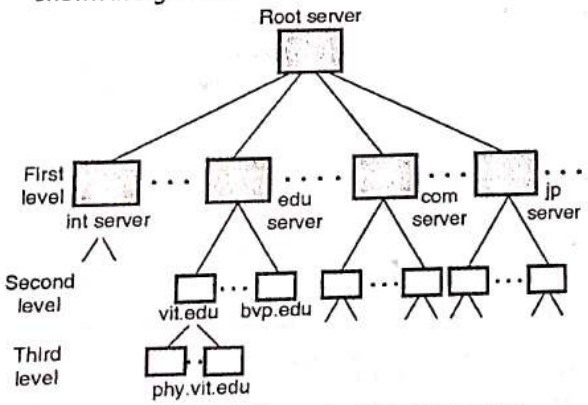
SPPU : May 18, May 19, Dec. 19

University Questions

Q. 1 What is DNS ? What is server hierarchy ? Explain domain name resolution process.
(May 18, Dec. 19, 6 Marks)

Q. 2 What is DNS server ? Explain lookup methods used by the DNS to resolve the remote names.
(May 19, 6 Marks)

- Name server contains the DNS database i.e. the various names and their corresponding IP addresses. Theoretically a single name server could contain the entire DNS database.
- But practically to store such a huge information at one place is inefficient and unreliable. Such a server will be soon overloaded and be useless and worst thing is if it ever goes down the entire Internet will go down.
- The solution to this problem is to distribute the information among many computers called **DNS servers.**
- Then we have to use a hierarchy of the Name servers as shown in Fig. 1.8.1.

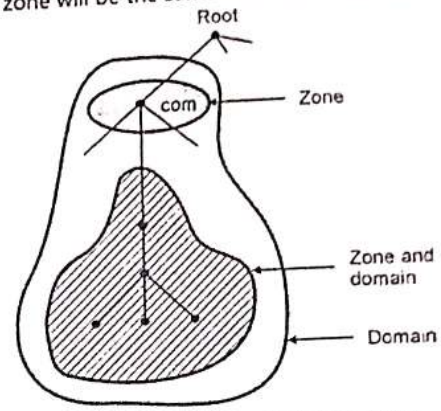


(G-634)**Fig. 1.8.1 : Hierarchy of name servers**

- First the whole space is divided into many first level domains. The root server stands alone and can create as many first level domains as required.
- The first level domains are further divided into smaller sub domains called second level domains. They can be further divided as shown in Fig. 1.8.1.
- Each server can be responsible (authoritative) to either a large or small domain. Note that the hierarchy of servers is similar to the hierarchy of names.
- The whole DNS name space is divided up into non overlapping **zones.** The concept of zones is as explained below.

Zones :

- With a number of DNS servers being used instead of a single one, we have to define the area over which each server has an authority. What a server is responsible for or has authority over is called as a zone.
- If a server is appointed for a domain and the domain is not further divided into sub-domains then the domain and zone will be the same as shown in Fig. 1.8.2.



(G-635)**Fig. 1.8.2 : Domains and zones**

- The server makes a database called a zone file. It keeps all information about every node under that zone.
- But if a server divides its domains into sub domains and delegates a part of its authority to other servers then domain and zone will be different from each other. This is shown in Fig. 1.8.2.
- The information about the nodes that belong to the sub domains is stored in the servers at the lower levels. The higher level and original server keeps some sort of reference of these lower level servers.

Root server :

- A root server is defined as a server whose zone consists of the whole DNS tree. It does not store any information about domains but delegates the authority to other servers.
- It only keeps the reference of these servers. There are more than 13 root servers and they are distributed all around the world.

Primary and secondary servers :

- DNS defines two types of servers namely the primary servers and the secondary servers.

Primary server :

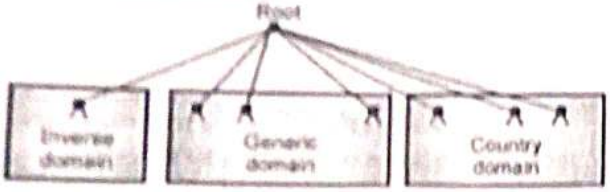
- It is a server which stores a file about its zone. It is authorized to create, maintain and update the zone file. It stores the zone file on a local disk.

Secondary server :

- This server transfers complete information about a zone from another server which may be primary or secondary server.
- The transferred information is saved on the disc storage of the secondary server. The secondary server is not authorized to create or update a zone file.
- If its zone file is to be updated, then it is to be done by the primary server.

1.9 DNS in the Internet :

- Let us now understand how DNS is used in Internet where the domain name space (tree) is divided into three different sections as shown in Fig. 1.9.1.
  1. Generic domain
  2. Country domain
  3. Inverse domain



(SPPU-434)**Fig. 1.9.1 : Use of DNS in Internet**

1.9.1 Generic Domains :

- The registered hosts are defined in the generic domains according to their generic behaviour e.g. com for commercial organizations.
- The first level in the generic domains section allows 14 possible lables. Some of them are given in Table 1.9.1.

Table 1.9.1 : Generic domain lables

Lable	Description
aero	Airline or aerospace related companies.
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
int	International organizations
mil	Military organization
net	Network support centers
org	Non-profit organizations

1.9.2 Country Domain :

- This domain section uses two character country abbreviations eg. US for united states. Second lable in this domain can specify organization or national designations.

1.9.3 Inverse Domain :

- The inverse domain is used for mapping an address to a name. This is exactly the opposite process discussed so far in which a name is mapped onto the address.

1.10 Name Address Resolution :

SPPU : May 16, Dec. 16, May 18, In Sem. March 19, May 19, Dec. 19

University Questions

Q. 1 State and explain name address resolution techniques in DNS. **(May 16, 6 Marks)**

Q. 2 What is domain name system ? Explain how a resolver looks up a remote name with suitable example. **(Dec. 16, 6 Marks)**

Q. 3 What is DNS ? What is server hierarchy ? Explain domain name resolution process. **(May 18, 6 Marks)**

Q. 4 What is Domain Name system ? Explain how a resolver looks up a remote name with suitable example ? **(March 19, 6 Marks)**

Q. 5 What is DNS server ? Explain lookup methods used by the DNS to resolve the remote names. **(May 19, 6 Marks)**

Q. 6 What is DNS ? What is service hierarchy ? Explain domain resolution process ? **(Dec. 19, 6 Marks)**

- The process of mapping a name to an address or vice versa is called as name address resolution.

Resolver :

- DNS application is based on the client server model. If a host wants to map a name to address or vice versa it calls a DNS client named as resolver.
- In other words, when the name ↔ address mapping is necessary a host calls a resolver. The resolver then sends a mapping request to the closest DNS server and accesses its storage.
- If this server has the requested information, it gives that information to the resolver but if it does not have the requested information, then it refers the resolver to other servers or asks other servers to provide the information.

- Thus the resolver receives the mapping from some source.
- It then checks for errors and if found error free delivers the mapping to the requesting process.

Mapping names to addresses :

- Generally the resolver gives a domain name to the server and requests for the corresponding IP address. The server checks the generic or country domains to get the corresponding address.
- If the domain name is from the generic domain section then the resolver receives a domain name such as,

  xxx.yyy.zzz.edu

- The query is sent to the local DNS server for resolution by the resolver. If the local server does not get the answer then, it will refer the resolver to other servers or asks them directly.
- The same procedure is followed for a name from country domain.

Mapping addresses to names :

- Here, a client sends an IP address to a server and requests for its name. This type of query is called as PTR query.
- To answer the PTR query, the DNS uses the inverse domain.
- If the IP address is 142.36.48.118 then the resolver first inverts the address and adds two labels "in_addr" and "arpa" to it. So the domain name sent is :

  118.48.36.142.in_addr.arpa.

- This is received by the local DNS and resolved.

1.10.1 Recursive Resolution :

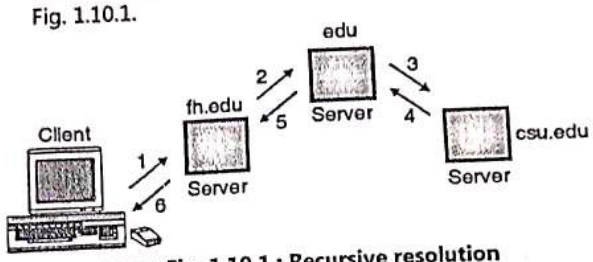
SPPU : May 16, May 19

University Questions

Q. 1 State and explain name address resolution techniques in DNS. **(May 16, 6 Marks)**

Q. 2 What is DNS server ? Explain lookup methods used by the DNS to resolve the remote names. **(May 19, 6 Marks)**

- Sometimes a client (resolver) requests for recursive or final answer from a name server. If this server is authorized for the domain name, it checks its database and sends a reply.
- But if this server is not authorized it diverts this request to another server (usually the parent server) and waits for the response.
- If the parent has the authority, then it sends the answer, otherwise it diverts the query to another server. When the query is solved, the response is returned back to the requesting client.
- Such a query is called as recursive query and the process is called recursive resolution. It is illustrated in Fig. 1.10.1.



(G-637)**Fig. 1.10.1 : Recursive resolution**

1.10.2 Iterative Resolution :

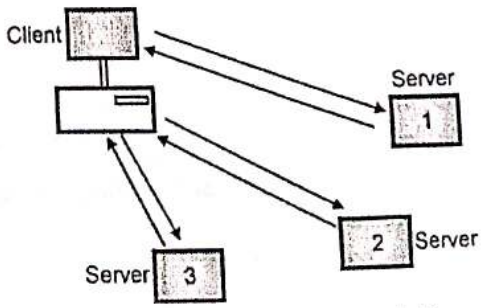
SPPU : May 16, May 19

University Questions

Q. 1 State and explain name address resolution techniques in DNS. **(May 16, 6 Marks)**

Q. 2 What is DNS server ? Explain lookup methods used by the DNS to resolve the remote names. **(May 19, 6 Marks)**

- This type of mapping can be done if the client does not ask for recursive answer. In iterative resolution, if the server has authority for the name it will send the answer.
- But if it does not have the authority then it returns to the client the IP address of the server that holds the answer to the query.
- The client has to repeat the query to this new server. If this server also cannot answer the query then it sends the IP address of another server to the client.
- Now the client should send the query to this third server.
- This process is called as iterative resolution because client sends the same query to different servers.
- Fig. 1.10.2 illustrates the iterative resolution.



(G-638)**Fig. 1.10.2 : Iterative resolution**

DNS examples :

- The DNS system is a **database**, and no other database on the planet gets this many requests.
- No other database on the planet has millions of people changing it every day, either. That is what makes the DNS system so unique!

For example :

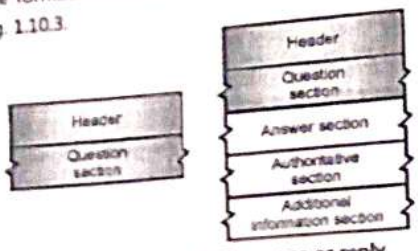
- www.yahoo.com - the world's best-known name
- www.mit.edu - a popular EDU name
- encarta.msn.com - a Web server that does not start with www
- www.bbc.co.uk - a name using four parts rather than three
- ftp.microsoft.com - an FTP server rather than a Web server
- www.spce.ac.in - Server in India 'in' domain.
- The COM, EDU and UK portions of these domain names are called the **top-level domain** or **first-level domain.**
- There are several hundred top-level domain names, including COM, EDU, GOV, MIL, NET, ORG and INT, as well as unique two-letter combinations for every country.

Comparison of Iterative Resolution and Recursive Resolution :

Sr. No.	Parameter	Iterative resolution	Recursive resolution
1.	Definition	Iteration refer to a situation where some statements are executed again and again using loops until some condition is true	Recursion referes to a situation where a function calls itself again and again until some base condition is not reached.
2.	Performance	Execution is faster because it does not use stack.	Comparatively slower because before each function call the current state of function is stored in stack. After the return statement the previous function state is again restored from stack.
3.	Memory	As it does not use stack memory usage is less.	Memory usage is more as stack is used to store the current information
4.	Code size	Bigger	Smaller

1.10.3 The DNS Message Format :

- DNS has two types of messages as follows and both of them have the same format.
  1. Query 2. Responses or reply
- The formats of the two DNS messages are as shown in Fig. 1.10.3.



(a) Query (b) Response or reply

(G-439) **Fig. 1.10.3**

- Both query and reply messages have the same header format with some fields set to zero for query messages.
- The header is 12 byte long. The header format for both the types of messages is shown by shaded portions in Fig. 1.10.3.

1.10.4 Caching :

- Every time a query is asked, the server has to spend time in searching the corresponding IP address.
- If this searching time is reduced then efficiency would go up. The searching time can be reduced by using a technique 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 other client request for the same mapping, it can check its cache memory and resolve the problem at its own level. This will certainly save a lot of time.
- But the problem with caching is that, if a server caches (stores) a mapping for a long time then the mapping may get outdated and the client will not get the latest mapping.
- This problem can be solved by adding the time to live information (TTL) to the mapping and each server is asked to keep a TTL counter for each mapping in its cache.

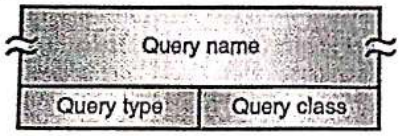
1.10.5 DNS Records :

- There are two types of records used in DNS as follows :
  1. Question records and
  2. Resource records

- The question records are used in the query and response messages whereas the resource records are used in the response messages.

Question Records :

- The client uses the question record to get the required information from the server.
- The format of question record has been shown in Fig. 1.10.4.



(G-1791) **Fig. 1.10.4 : Question record format**

- Various fields in the question record format are query types and query class.

Query name :

- This field has a variable length and it contains a domain name. The count field tells us how many characters are present in each section.

Query type :

- This field is 16-bit long and it defines the type of query.
- Some of the commonly used query types are A, NS, CNAME, SOA, ANY etc.

Query class :

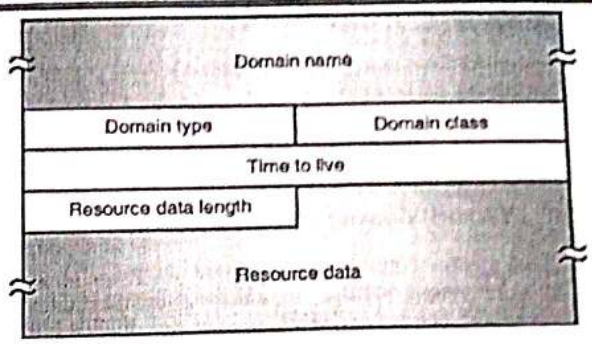
- This field also is 16-bit long. It defines the specific protocol using DNS.
- Table 1.10.1 has listed some of possible classes. However the most important class would be IN i.e. internet (class 1).

Table 1.10.1 : Query classes

Class	Mnemonic	Explanation
1.	IN	Internet
2.	CSNET	CSNET network (Not used now)
3.	CS	COAS network
4.	HS	The Hesiod server (MIT)

Resource Record :

- Each domain name i.e each node on the tree in DNS is associated with the resource record which is a part of the server database.
- Resource records are returned by the server to the client. The format of RR has been shown in Fig. 1.10.5.



(G-1792) **Fig. 1.10.5 : Format of resource record**

1. **Domain name :**
- This field contains the domain name and its length is not fixed. It has a variable length. The domain name in the question record is duplicated here.

2. **Domain type :**
- This field and the query type field in the question record are the same except the last two types i.e. AXFR and ANY are not allowed.

3. **Domain class :**
- This field is same as the query type field in the question record.

4. **Time to - live :**
- This field is 32 bit long and it defines the time for which the answer is valid (in seconds). If the contents of this field is zero, then it indicates that the resource record is used only in a single transaction.

5. **Resource data length :**
- This field is 16 bit long. It is used for defining the length of the resource data.

6. **Resource data :**
- As shown in Fig. 1.10.5 the resource data field is a variable length field. It contains the answer to the query or domain name or the additional information.
- The format and contents of this field depend on the value of the type field. It can be one of the following :
  1. A number
  2. A domain name
  3. An offset pointer
  4. A character string.

Encapsulation :

- DNS can use either TCP or UDP. It may choose any one of these protocol but in either case the server uses port 53.
- The UPD is preferred if the length of response message is upto 512 bytes whereas TCP is used if the message length is larger than 512 bytes.

Registrars :

- New domains are added to DNS through a registrar, which is a commercial entity.
- Whenever an organization applies for DNS domain name, a registrar first checks that the requested domain name is unique.

1.11 World Wide Web (WWW) :

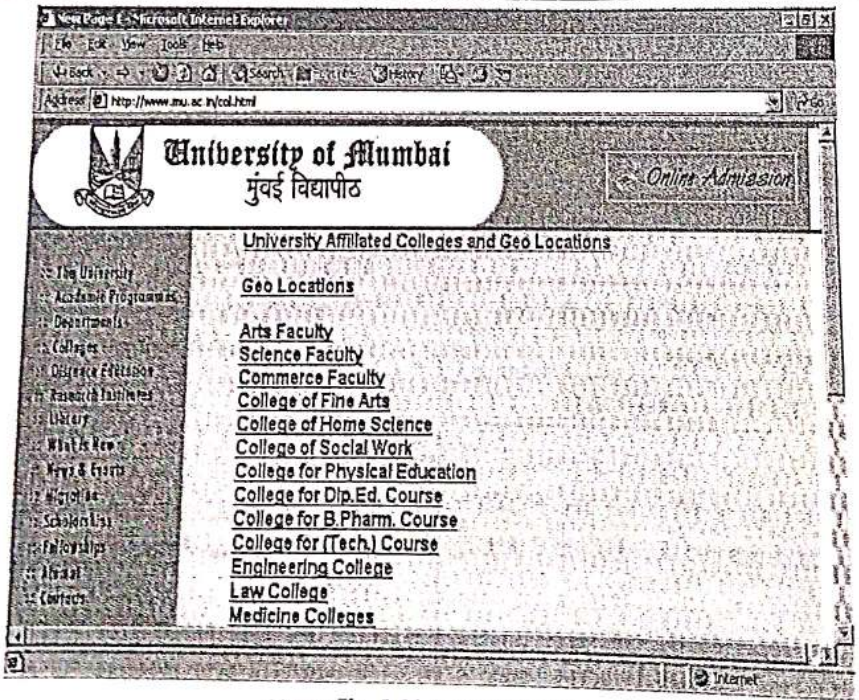
- People have become aware of the power of Internet through WWW. HTTP is a file transfer protocol which is specifically designed to facilitate access to the WWW.
- The World Wide Web is an architectural framework for accessing documents which are spread out over a number of machines over Internet.
- It has a colourful graphical interface which is easy for the beginners to use. It provides information on almost every subject. The web (also known as WWW) began in 1989 at CERN the European center for nuclear research.
- The web was designed basically to connect scientists stationed all over the world. The web is basically a client-server system.
- The web pages are written in the languages HTML and Java. The growth of the World-Wide Web (WWW or simply Web) today is simply phenomenal.
- Each day, thousands of more people join the Internet (above 100 million users at recent estimates).
- Easy retrieval of electronic information along with the multimedia capabilities of Web browsers (like Mosaic or Netscape) are the factors responsible for this explosion.
- This topic provides some basic information behind some of this technology used in accessing the World-Wide Web.

Difference between Web and Internet :

- The Web and the Internet are not the same thing. The Web is a collection of standard protocols or instructions, sent back and forth over the Internet to gain access to information.
- The Internet, on the other hand, is a "network of networks" -- a more physical entity.

1.11.1 Web from the Users Side :

- The user (client) looks at the web as a collection of vast worldwide collection of documents called **pages** in short.
- **Links or pointers :** Each page may contain links or pointers to it, related pages, anywhere in the world. A user can follow a link by clicking on it.
- This will take him to the pages pointed by the links. This process can be repeated indefinitely.
- **Hypertext :** Pages which point to the other pages are said to use hypertext.
- **Browser :** The program used for viewing pages is called as a browser.
- The job of a browser is to fetch the page requested by the user, interprets the text and formatting commands which it contains, display the page with proper format on the screen. An example of a web page is shown in Fig. 1.11.1.



(G-653) **Fig. 1.11.1 : A Web page**

- A web page starts with a title and contains the following :
  1. Some information
  2. Strings of text, linked to other pages
  3. E-mail address of the page's maintainer.

Hyperlinks :

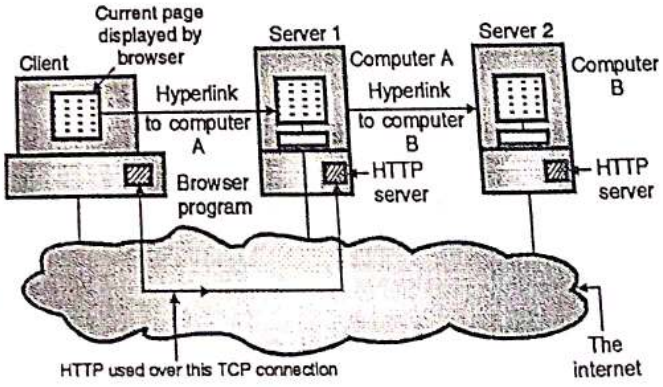
- Strings of text that are links to other pages are called hyperlinks. They are high lightened by underlining, using special colour or both.
- In order to follow a link, the user has to place the cursor on the high lightened area using the mouse or arrow keys and select it by clicking the mouse or pressing the ENTER key.
- The browsers can be of two types, namely the graphical browsers and non-graphical browsers. But the graphical browsers are more popular. Voice based browsers are also being developed.
- Most browsers have a large number of buttons and features which make the navigation on web easier. There can be a button to back to the previous page or a button for going forward to the next page.
- Some browsers can provide a facility of having a button or menu item to set a bookmark on a given page and another one to display the list of bookmarks.
- This makes it possible to revisit any of them with a single click on mouse.
- It is also possible to save pages or print them. Lot of options are available to control the screen layout and setting various preferences of the users.
- The web pages can also contain line drawings, icons, maps, photographs etc and they can be linked (if required) to another page.

Hypermedia :

- All pages may not be viewable in the conventional way because some pages may contain audio tracks, video clips or both.
- If the hypertext pages are mixed with other media, the result of such a mixing is called as hypermedia. Some browsers are capable of displaying all kinds of hypermedia but others cannot do so.
- Many web pages contain large images that take a long time to load. When the images are being loaded, the user does not have anything to see.
- To solve this problem, some browsers first fetch and display the text and then get the images. The user can read the text when images are getting loaded.
- Another strategy can be to provide an option to disable the automatic fetching and displaying of images.
- One more alternative opted by some page writers is to display the full image in a coarse resolution and then to fill up the details gradually.
- Some web pages display forms requesting the user to fill up information.
- This is meant for searching a database for a user supplied item or ordering a product etc.
- Some web pages contain maps which allow the users to click on them to get the zooming facility or get information about the clicked geographical area.
- For hosting a web browser a machine should be directly connected to the Internet or at least have a SLIP or PPP connection to a router or other machine which is directly connected to Internet.
- This is because of the manner in which the browser fetches a page. To fetch a page it has to establish a TCP connection to the machine from where the page is to be fetched.

1.11.2 Web from the Servers Side :

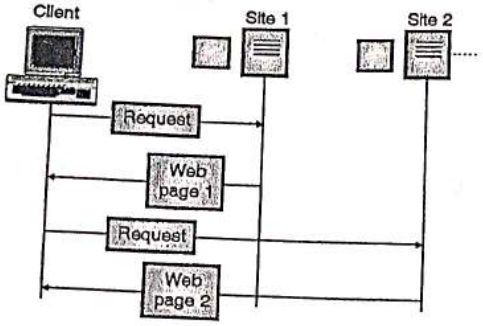
- Every website has a server process. It is listening to TCP port 80 on which incoming clients (browsers) are connected.
- Once a connection is established, the client sends a request and the server sends a reply for that. Then the connection is released.
- The protocol used for defining the legal request and replies is called HTTP. Fig. 1.11.2 shows various parts of the web model.



(G-654) **Fig. 1.11.2 : Web model**

1.11.3 WWW Architecture :

- The WWW is a distributed client/server service. A client (user) uses a browser to access a service using a server.
- But the service provided is distributed over a number of separate locations called as sites. Fig. 1.11.3 shows the architecture of WWW.

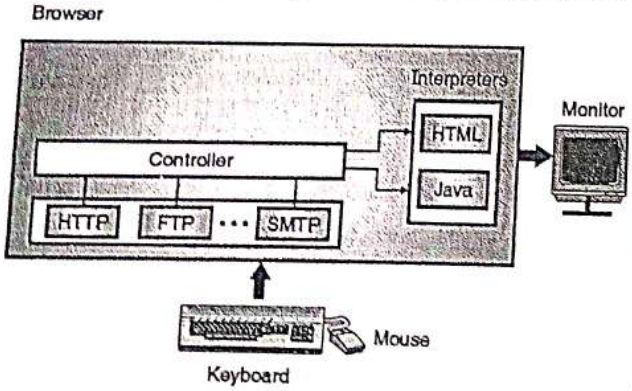


(G-655) **Fig. 1.11.3 : WWW architecture**

- As shown in Fig. 1.11.3, there are number of sites and each site holds a number of web pages. These pages can be retrieved and viewed by using browsers.
- The client sends a request through its browser to get a web document from a particular site. This request contains the site address and web page address (called URL) along with some other information.
- The server at the requested website finds the document and sends it to the client.

1.11.4 Browser (Web Client) :

- Even though a number of browsers are available around, the browser architecture is nearly the same for all of them.
- Each browser consists of the following parts :
  1. A controller
  2. Client programs
  3. Interpreters.
- Fig. 1.11.4 shows the general architecture of a browser.



(G-665) **Fig. 1.11.4 : Browser architecture**

- The controller receives input from the keyboard or mouse. It then uses the client programs like HTTP, FTP etc to access the document.
- After accessing the document, the controller makes use of an interpreter such as HTML or Java (depending on type of document) and displays the accessed document on the screen.

1.11.5 Server :

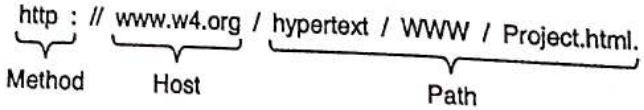
- All the information is stored in the form of web pages at the server. Whenever a client requests for one the corresponding document is sent to the client.

1.11.6 Uniform Resource Locator (URL) :

- The client accessing a web page needs an address.
- The HTTP uses the URL to facilitate the access of any document distributed over the world.
- The URL specifies any information on Internet by using four thing as shown in Fig. 1.11.5(a).
- They are as follows :
  1. Method or protocol
  2. Host computer
  3. Port
  4. Path.

(G-660) **Fig. 1.11.5(a) : URL**

- Method is the protocol used such as FTP, HTTP which helps retrieving the desired information. Host is the computer where the required information is located.
- The name of the computer begins with www but this is not mandatory. URL can optionally contain the server's port number.
- If the port is to be included then it should be inserted between host and path and it should be separated by a colon, as shown in Fig. 1.11.5(a).
- Path is the name of the file where the information is located. The port and path fields are separated from each other by a slash.
- **Version :** The latest version of HTTP is 1.1 but the versions 0.9 and 1 are also used.
- The example of URL is shown in Fig. 1.11.5(b). Note that the port is not included.



(G-1969) **Fig. 1.11.5(b) : Example of URL**

1.11.7 Cookies : User-Server Interaction :

- We know that the HTTP servers are stateless. The disadvantage of being stateless is that the server cannot identify the client.
- The meaning of statelessness is that the client server relationship gets over as soon as their communication terminates.
- But the advantage of statelessness is that the server design is simplified to a great extent and it permits the engineers to develop high performance web servers which can handle thousands of TCP connections at a time.
- But many a times it is necessary for a web site to identify users. In such cases HTTP uses **cookies.** Cookies are defined in RFC 2109 and they allow sites to keep track of users.
- Cookies are not used by all the sites but some of the prominent sites that use cookies are : Yahoo, Amazon etc.

Components of cookie technology :

- Following are the four components of the cookie technology :
  1. A cookie header line in HTTP response message.
  2. A cookie header line in the HTTP request message.
  3. A cookie file kept on the user's end system and managed by user's browser.
  4. A back end database at web site.

Operating principle :

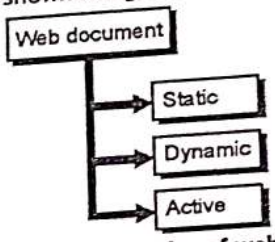
- If a new user X contacts a site (that uses cookies) for the first time, then that web site creates a unique identification number for this new user and then creates an entry in its back end data base.
- This entry is associated with the identification number of user X.
- The server will then respond to X's browser by including the header set-cookies : header, in the HTTP response number of user X.
- For example the header line can be :

  Set-cookie : 1 2 3 4 5 6 7

  Where, 1 2 3 4 5 6 7 is the identification number.
- When X's browser receives the HTTP response message, it reads the set cookie : header. The browser then appends a line to the special cookie file which is managed by the browser.
- This line will include the hostname of the server and the identification number 1 2 3 4 5 6 7. Next time when X visits this same site again, his browser will include the same identification number in each of his HTTP request.
- Thus it is now possible for the web site to track X's activities. It is then possible to know the areas of interest of X, which pages does he visit and at what time etc.
- Cookies simplify the internet shopping to a great extent but they remain highly controversial because they are thought as invasion in users privacy.
- It is possible to use cookies to gather personal information about X across a large number of websites.

1.12 Web Documents :

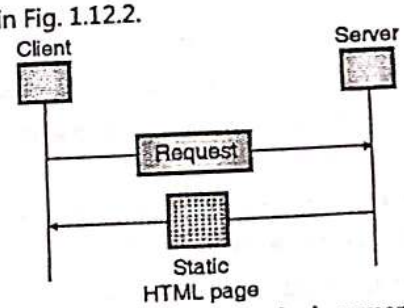
- The web documents can be classified into three categories as shown in Fig. 1.12.1.



(G-668) **Fig. 1.12.1 : Categories of web documents**

1.12.1 Static Documents :

- The contents of static documents are fixed. These contents are created and stored in a server. If required the client can get a copy of static document.
- The contents of the static document are determined when it is created.
- These contents cannot be changed when the static document is being used.
- It is possible to change the contents of static document at the server but the user cannot change them. The user can display the static document by using a browser as shown in Fig. 1.12.2.



(G-669) **Fig. 1.12.2 : Static document**

1.12.2 HTML (Hypertext Markup Language) :

- The web pages are created by using a language called HTML
- It uses certain marks to format the text. For example if a part of text is required to be "boldface" then we can use the beginning and ending bold face tags (marks) in the text as shown below :

  <B> –Beginning of boldface

  </B>–End of boldface.
- Here <B> and </B> are the instructions for the browser.
- The browser will make the part of the text between these tags bold. HTML lets the user to use only ASCII characters for the main text as well as for formatting instructions.
- So every computer can receive the whole document as an ASCII document. The formatting instructions are used by the browser to format the data.

Advantages of HTML :

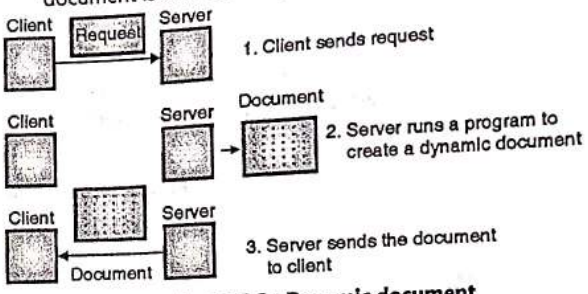
1. Any one can edit it.
2. It is easy to learn and use.
3. People located in different parts of world can work on the same document.
4. It widens the access to web publishing for non-technical users.
5. It is a very flexible tool which can be used for a number of applications.
6. It can be installed free of cost.
7. It is widely used and almost every browser supports it.
8. It is fast to download because the text is compressible.
9. It can be used to present almost any kind of data.

Disadvantages of HTML :

1. As any one can edit, this may be too open for some applications (for example confidential documents).
2. It is open to SPAM and vandalism.
3. Requires Internet connectivity to collaborate.
4. Due to flexibility of its structure, the structure can become disorganized.
5. It takes a long time to choose the colour scheme of page and to create tables, forms etc.
6. It can only create static and plain pages. It is not useful to create dynamic pages.
7. Security features of HTML are not good.
8. It is not centralized. So all the web pages must be edited separately.
9. It has very limited styling capabilities.

1.12.3 Dynamic Document :

- The dynamic documents are not present in a predefined format, like static documents. A dynamic document is created by a web server on the request for the document from a browser.
- Refer Fig. 1.12.3 to understand how a dynamic document is created and passed on to the client.



(G-670) **Fig. 1.12.3 : Dynamic document**

- First the client sends a request to the web server. After receiving this request, the web server will execute an application program to create a dynamic document.
- The server returns the dynamic document as a response of the request to the client. The contents of a dynamic document will be different corresponding to every request.
- A simple example of a dynamic document is to get time and data from the server. A server follows the steps given below to handle dynamic documents :
  1. The server checks the URL in order to find if it has defined a dynamic document.
  2. If the URL has defined the dynamic document, then the server executes the program.
  3. The output of this program is the dynamic document. It is returned back to the client.

1.12.4 Common Gateway Interface (CGI) :

- CGI is the name of a technology which creates the dynamic documents and handles them too.
- CGI is in fact a set of standards. It defines the way in which a dynamic document should be written, the way in which input data be supplied to the program and how the output result be used.

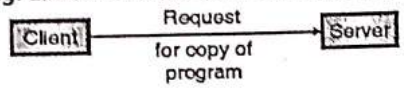
- Note that CGI is not a new language. It allows the user to use the existing languages such as C, C++, Perl etc. However CGI defines rules and terms which are to be followed by the programmers.
- The word **common** in CGI shows that this standard defines some rules which are commonly applicable to any language or platform.
- The word **gateway** indicates that a CGI program is gateway for accessing other resources such as databases and graphic packages.
- Lastly the word **interface** in CGI indicates the presence of a set of terms, calls and variables which can be used in any CGI program.

CGI Program :

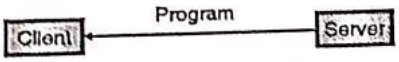
- It is a code which is written in one of the languages that supports CGI (such as C, C ++, etc.).

1.12.5 Active Documents :

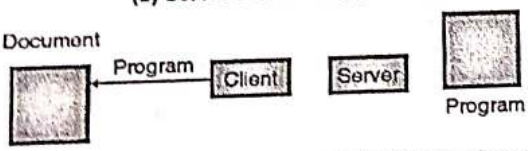
- Active document can be defined as the program, that is needed to be run at the client side.
- The examples of active documents are the programs creating animated graphics on the screen or the ones which help interaction with the user.
- Refer Fig. 1.12.4 to understand this concept.



(a) Client request for copy of program



(b) Server sends a copy of program



(c) Clients runs the program and creates the documents

(G-671) Fig. 1.12.4 : Active document

- It shows that whenever a browser requests for an active document, the server will send a copy of document in the form of byte code.
- The active document will then be run at the browser (client) site. The server stores the active document in the form of a binary code.
- The active document is stored on the server but it is not run on the server. The client receives the document and stores it, and can run it as many times as required without repeating the request.
- The server sends the active document to the client in the binary form. So it is possible to compress it at the server's site and then decompress it at the clients site.
- This will save the bandwidth as well as the transmission time.

Steps in creation of an active document :

- Refer Fig. 1.12.4 to understand the creation, compilation and execution of an active document.

1. At the server, a program is written in source code and stored in a file.
2. Then the program is compiled and binary code is creates and stored in a file at the server's site.
3. A client (browser) requests for a copy of program as shown in Fig. 1.12.4(a). This program is transported from the server to the client in the compressed form.
4. The client converts the received program from binary code into executable code using its own software.
5. The client runs the program to create the desired result which can include animation or interaction with the user.

1.13 Electronic Mail :

SPPU : Dec. 04, May 05, May 06

University Questions

Q. 1 What are the basic functions of E-mail systems ? Explain the importance of MIME in e-mail system. **(Dec. 04, 8 Marks)**

Q. 2 Explain various functions of E-mail system. Why MIME is used in E-mail system ? **(May 05, 8 Marks)**

Q. 3 Explain the function of E-mail system. **(May 06, 8 Marks)**

- One of the most popular network services is electronic mail (e-mail). Simple Mail Transfer Protocol (SMTP) is the standard mechanism for electronic mail in the internet.
- The first e-mail systems simply consisted by file transfer protocols. But some of the limitations of this system were as follows :

1. It is difficult to send a message to a group of people.
2. Message did not have any internal structure. So its computer processing was difficult.
3. The sender never used to know if a message arrived or not.

4. It was not easy to handover one's e-mails to someone else for the purpose of managing them when one is out of town or country for sometime.
5. The user interface with the transmission system is poorly integrated.
6. It was not possible to create and send messages containing a text, drawing, facsimile and voice together.

- So more elaborate e-mail systems were proposed. ARPANET e-mail proposals were published as RFC 821 (transmission protocol) and RFC 822 (message format). These are used in Internet.

1.13.1 E-mail Architecture and Services :

SPPU : Dec. 10, May 15

University Questions

Q. 1 Explain email architecture and its services. (Dec. 10, 8 Marks)

Q. 2 Explain the email architecture and its services. (May 15, 4 Marks)

- An e-mail system consists of two subsystems :
  1. User agents and 2. Message transfer agents.

  User agents : They enable users to read and send e-mail.

  Message transfer agents : They move the messages from the sender to the receiver.

Basic Functions :

- E-mail systems support five basic systems which are as follows :
  1. Composition
  2. Transfer
  3. Reporting
  4. Displaying and
  5. Disposition

1. Composition :

- The process of creating messages and to answer them is known as composition.
- The system can also provide assistance with addressing and a number of header fields attached to each message.

2. Transfer :

- It is the process of moving messages from the sender to the recipient.
- This includes establishment of a connection from sender to destination or some intermediate machine, transferring the message, and breaking the connection.

3. Reporting :

- The reporting system is designed to tell the sender about whether the message was delivered or rejected or lost.

4. Displaying :

- It is the process of displaying the incoming messages so that it can be read by the user. For this purpose simple conversions and formatting are required to be done.

5. Disposition :

- This is concerned with what the recipient does with the received message. Disposition is the final step in e-mail system.
- Some of the possibilities are as follows :
  1. Throw after reading
  2. Throw before reading
  3. Save messages
  4. Forward messages
  5. Process messages in some other way.

Advanced features of E-mail systems :

- Some of the advanced features included in addition to the basic functions are as follows :
  1. Forwarding an e-mail to a person away from his computer.
  2. Creating and destroying mailboxes to store incoming e-mail.
  3. Inspecting contents of mailbox, insert and delete messages from the mailboxes.
  4. Sending a message to a large group of people using the idea of mail list.
  5. To provide the facility of registered e-mail.
  6. Automatic notification of undelivered e-mails.
  7. Carbon copies
  8. High priority e-mail (setting the priority of e-mails)
  9. Secret (encrypted e-mail)
  10. Alternative recipient. This allows automatic forwarding of an e-mail to an alternate recipient if the main recipient is not available.

E-mail Envelope :

- In the modern e-mail systems, there is a distinction made between the e-mail and its contents. An e-mail envelope contains the message, destination address, priority, security level etc.

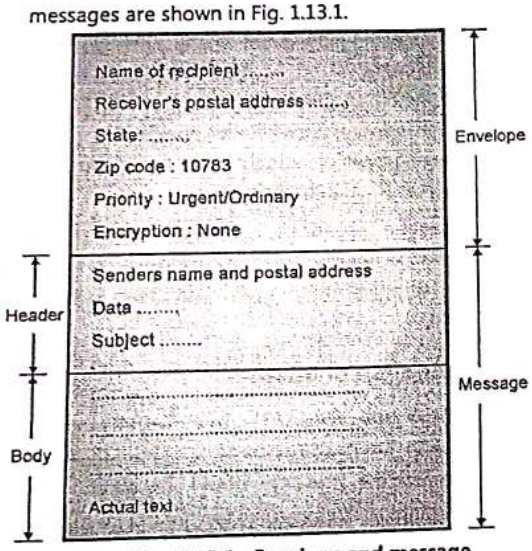
- The message transport agents such as SMTP use this envelope for routing.

Message :

- The actual message inside the envelope is made of two parts :

  1. Header and 2. Body

- Header carries the control information while body contains the message contents. Envelopes and messages are shown in Fig. 1.13.1.



(G-640) **Fig. 1.13.1 : Envelope and message**

1.13.2 Message Formats :

- Let us now discuss the e-mail message formats.

RFC 822 :

- All the e-mail messages consist of an envelope, a few header fields, a blank line and then the message body.
- Each header field logically consists of a single line of ASCII text which consists of the field name, a colon and a field.
- Normally the user agent builds a message and passes it to the message transfer agent which uses some header fields for construction of an envelope.
- Table 1.13.1 shows the principle header fields related to the message transport. Let us discuss them one by one.

Table 1.13.1 : RFC 822 header fields related to message transport

Header Name	Meaning
To :	E-mail address of primary recipients
Cc :	E-mail address of secondary recipients (Carbon copy)
Bcc :	E-mail address for blind carbon copies
From :	Originator of the message
Sender :	E-mail address of the person sending the message
Received :	Line added by each transfer agent along the route
Return – Path :	Can be used to identify the path back to the sender.

1. The To : field :

- This field gives the DNS address of the primary recipient. It is allowed to have multiple recipients.

2. The Cc : field :

- This field gives the addresses of any secondary recipients. Cc stands for carbon copy.
- Whatever message and attachments are sent to the primary recipient the same are sent to the secondary recipient as well.

3. The Bcc : field :

- The long form of Bcc is blind carbon copy. This field is like Cc field, except that this is deleted from all the copies sent to the primary and secondary recipients.
- Thus a sender can send copies to third parties without primary and secondary recipients knowing about it.

4. From : and Sender : fields :

- These fields tell about who wrote the message and who actually sent the message respectively because the person who creates the message and the person who sends it can be different.
- The From : Field is necessary but the Sender : field can be omitted, if it is same as the From : field.
- These fields are required when the message cannot be delivered and is to be returned to the sender.

5. Received : field :

- A line containing Received : is added by each message transfer agent along the way. This line carries the agent's identity, date and time at which the message was received.
- It also contains some other information that can be used to find bugs in the routing system.

6. The Return-Path : field :

- This field is added by the final message transfer agent and it is intended to tell how to get back to the sender.

- This information can be obtained from all the received : headers.

Other header fields :

- In addition to the fields of Table 1.13.2, RFC 822 messages may contain many other header fields.
- These are used by either the user agents or human recipients some of them are shown in Table 1.13.2.

Table 1.13.2 : Some fields in RFC 822 message header

Header	Meaning
Date :	The date and time of the message.
Reply-To	E-mail address to which the reply is to be sent
Message-Id :	Message identifying number
In-Reply-To :	Message-Id of the message to which this is a reply
References :	Other relevant message identifying numbers
Keywords :	Keywords chosen by user
Subject :	Summary of the message for the one line display.

- The RFC 822 allows the users to invent new headers for their own private use but it is essential that these headers start with the string X-. For example X-Event of the week.

Message Body :

- The message body comes after the header. The users can include anything that they want to send, in the message body.
- It is possible to terminate the messages with ASCII cartoons, quotations, political statements etc.

1.14 MIME – Multipurpose Internet Mail Extensions :

SPPU : May 08, May 09, Dec. 09, Dec. 10, Dec. 16, Dec. 18

University Questions

Q. 1 Write short notes on : MIME. (May 08, Dec. 09, Dec. 10, 5 Marks)

Q. 2 Where and why do we use MIME ? (May 09, 8 Marks)

Q. 3 What is MIME ? Discuss its role in SMTP. (Dec. 16, 4 Marks)

Q. 4 What is MIME ? Explain the MIME header with suitable example. (Dec. 18, 4 Marks)

- In the early days, the e-mail used to consist of only the text messages in English and expressed in ASCII.
- RS 822 was sufficient for this environment. But in the worldwide internet environment, this approach is not adequate.
- Some problems are encounter in sending and receiving the following types of messages.
  1. Messages in certain languages that have accents such as French or Germans.
  2. Messages which do not contain text e.g. audio and video.
  3. Messages in the languages which do not have alphabets (e.g. Chinese and Japanese).
  4. Messages which contain some non-Latin alphabets such as Russian or Hebrew.
- The solution to these problems was MIME i.e. Multipurpose Internet Mail Extensions. It was proposed in the standard RFC 1341 and then updated in RFC 1521.

1.14.1 Principle of MIME :

- MIME uses the same RFC 822 format but it adds structure to the message body (In RFC 822 there is no structure to the message body). In addition to this, MIME defines encoding rules for non ASCII messages. It is possible to send MIME messages using the existing mail programs and protocols.
- The sending and receiving programs need to be changed to achieve this, which users can do themselves.

New message headers :

- Five new message headers are defined for MIME. They are listed in Table 1.14.1.

Table 1.14.1 : New headers in MIME

Sr. No.	Header Name	Meaning
1.	MIME – Version :	Indicates the MIME version
2.	Content–Description :	Tells what is in the message
3.	Content – Id :	Identifier
4.	Content – Transfer – Encoding :	How is the body wrapped for transmission
5.	Content – Type :	Type of the message

1. MIME-Version :

- It tells the user agent that this message is a MIME message and it also specifies the version of MIME being used.

2. Content-Description :

- This header tells what the message is. It is in the form of ASCII string. This header is needed because the recipient will know whether it is worth decoding and reading the message.

3. Content-Id :

- This field identifies the contents. Its format is same as the format of standard Message- Id header.

4. Content-Transfer-Encoding :

- This field tells how the body is wrapped for transmission through a network. The network can object to most characters except for the letters, numbers and punctuation marks.
- Five schemes are provided for this purpose in addition to an escape to new schemes. These schemes are as follows :
  - (a) The first scheme simply uses ASCII text. Each ASCII character uses 7-bits and can be carried directly if each line contains at the most 1000 characters.
  - (b) The second scheme is similar to the first scheme except for one change. Instead of 7-bit characters, now 8-bit characters are used. The maximum line length remains unchanged.
  - (c) Third is the messages using binary encoding. The problems with these messages are that they do not always use all the 8 bits and do not stick to the 1000 line per character limit. There is no guarantee that these messages arrive error free.
  - (d) The fourth type is encoded binary messages using base 64 encoding which is also called as ASCII armour.
  - (e) The fifth type is the messages with a special type of encoding called **quoted-printable encoding.**

5. Content-Type :

- This is last header in Table 1.14.1 and it is used to specify the type of the message body. RFC 1521 defines seven types with each one having one or more subtypes.
- The type and subtype are separated by a slash such as,

  Content – Type : Video / mpeg.

- The subtype must be given in the header. Table 1.14.2 gives the initial list of types and subtypes specified in RFC 1521.

Table 1.14.2 : The MIME types and subtypes in RFC 1521

Type	Subtype	Description
Text	Plain	Text in the unformatted way
	Richtext	Text includes simple formatting commands
Image	Gif	Still pictures in GIF format
	Jpeg	Still pictures in JPEG format
Audio	Basic	Audio or sound content
Video	Mpeg	Movie (video) in MPEG format
Applications	Octet-stream	Byte sequence in uninterpreted form
	Post script	A printable document in Post script
Message	Rfc 822	A MIME RFC 822 message
	Partial	Split message for transmission
	External body	Message itself should be fetched over the net
Multipart	Mixed	There are independent parts in the specified order
	Alternative	Same message in different formats
	Parallel	Parts must be viewed simultaneously
	Digest	Each part is a complete RFC 822 message

- Note that many new types have been added to the basic list of Table 1.14.2 and the addition is still being made. Let us discuss the content types listed in Table 1.14.2.

(a) Text :

- The text type is for straight text. There are two subtypes namely plain and richtext. The text/plain combination represents the original messages without any encoding or further processing.
- The text/richtext allows simple formatting in the text. It allows the text with boldface, italics, small and large point sizes, indentation, subscripts, page layout etc.

(b) Image :

- This MIME type is used for transmitting still pictures. There are many formats used for storing and transmitting images with or without compression. The two subtypes are GIF and JPEG.

(c) Audio and Video :

- The audio type is for sound and video is for moving pictures. The video does not include any soundtrack. Only one video format defined is MPEG which is designed by the Moving Picture Experts Group (MPEG).

(d) Applications :

- This type is used for formats which require external processing and which is not covered by any other type.
- The octet stream is a sequence of uninterpreted bytes. When it is received a user agent should display it and suggest the user to copy it in a file for further processing.
- The other subtype is post script. It corresponds to the post script language produced by Adobe systems which is used for describing printed pages.

(e) Message :

- This type allows one message to be fully encapsulated inside the other message. This is useful in order to forward e-mails.
- The partial subtype allows to break an encapsulated message into pieces and send them separately. The external body subtype can be used for very long messages such as video films.

(f) Multipart :

- This is the last type of multipart. It allows a message to contain multiple parts in the same message. The beginning and end of each part is clearly demarked within a message.
- There are four subtypes. The mixed subtype allows each part to be different. In the alternative subtype each part should contain the same message expressed in a different medium or encoding.
- The alternative subtype can be used for multiple languages as well.
- The parallel subtype is used for viewing all parts simultaneously e.g. audio and video parts of a movie.
- The fourth subtype is digest. It is used when many messages are packed together to form a composite message.

1.15 Message Transfer Agent : SMTP :

SPPU : May 04, Dec. 06, Dec. 08, Dec. 11, May 15, Dec. 16

University Questions

Q. 1 Explain the role of SMTP and POP protocols in e-mail transfer. **(May 04, 8 Marks)**

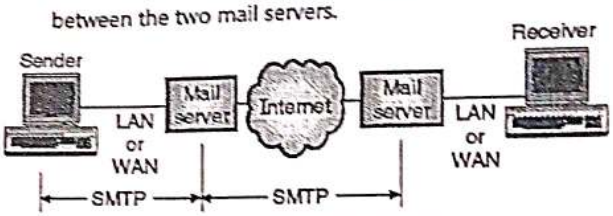
Q. 2 In SMTP, if we send a one line message between two users, how many lines of commands and responses are exchanged ? Give the example. **(Dec. 06, 12 Marks, Dec. 08, 10 Marks)**

Q. 3 Write a short notes on SMTP. **(Dec. 11, 4 Marks)**

Q. 4 Describe SMTP header format. **(May 15, 4 Marks)**

Q. 5 What is MIME ? Discuss its role in SMTP. **(Dec. 16, 4 Marks)**

- The actual mail transfer is carried out through the message transfer agent.
- A system should have the client MTA in order to send a mail and it should have a server MTA in order to receive one.
- SMTP is the protocol which defines MTA client and server in the Internet.
- As shown in Fig. 1.15.1, the SMTP is used twice, once between the sender and sender's mail server and then between the two mail servers.

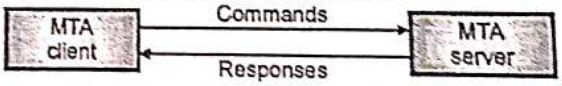


(G-641)**Fig. 1.15.1 : SMTP range**

- The job of SMTP is simply to define how commands and responses be sent back and forth. Each network can choose its software package for implementation.

1.15.1 Commands and Responses :

- As shown in Fig. 1.15.2, SMTP the transfer of messages between MTA client and MTA server takes place using the command and response principle.



(G-642)**Fig. 1.15.2 : Commands and responses in HTTP**

- Each command or response is terminated by a two character end of line token. The two characters used are carriage return and line feed.

1.15.2 SMTP (Simple Mail Transfer Protocol) :

SPPU : May 18, Dec. 19, In Sem. March 20

University Questions

Q. 1 Explain in brief SMTP protocol. **(May 18, Dec. 19, March 20, 4 Marks)**

- In internet the source machine establishes a connection to port 25 of the destination machine so as to deliver an e-mail.

- An e-mail daemon which speaks SMTP is listening to this port.
- This daemon is supposed to perform the following tasks :
  1. Accept the incoming connections, and copy messages from them into appropriate mailboxes.
  2. Return an error message to the sender, if a message is not delivered.
- SMTP is a simple ASCII protocol. Once a TCP connection between a sender and port 25 of the receiver is established, the sending machine operates as a client and the receiving machine acts as a server.
- The client then waits for the server to take initiative in communication. The server sends a line of text which declares its identity and announces its willingness/ unwillingness to receive mail.
- If the server is not prepared, the client will release the connection, wait for some time and try again later. But if the server is willing to accept e-mail, then the client announces the sender of e-mail and its recipient.
- If such a recipient exists at the destination, then the server tells the client to send the message. The client, then sends the message and the server sends back its acknowledgement.
- No checksums are generally required because TCP provides a reliable byte stream. If there are any more e-mail, then they can be sent now.
- After exchanging all the e-mail, the connection is released. SMTP uses numerical codes. The lines sent by the client are marked C : ; and those sent by the server are marked S : ;
- Some of the commands, useful for communication are :

  HELO, RCTP, DATA, QUIT etc.
- RCTP represents recipient. If only one command is used then the message is being sent to only one recipient. If the command is used many times, then it indicates that the message is sent to more than one recipients.
- In such a case each message is individually acknowledged or rejected.
- The syntax of four character commands for the clients are rigidly specified but the syntax for the replies are not that rigid.
- The SMTP protocol is well defined by RFC 821 but some problems are still present.

Problems in SMTP :

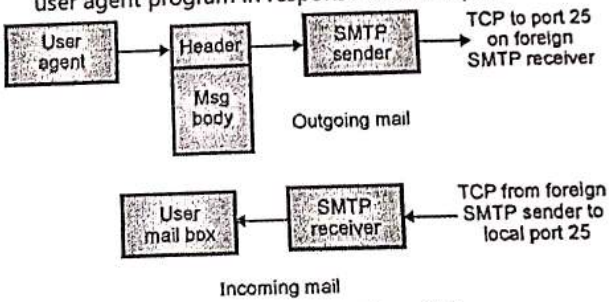
- Some of the problems in SMTP are as follows :
  1. Some older versions of SMTP are not capable of handling messages longer than 64 kB.
  2. If client and server have different time-outs, then one of them may give up when the other is still busy. This will terminate the connection unnecessarily.
  3. In rate situations, infinite mailstorms can be triggered.

Extended SMTP (ESMTP) :

- Some of these problems can be solved by using the extended SMTP (ESMTP) which is defined in RFC 1425.

1.15.3 Components of E-mail System :

- The three main components of internet mail system are :
  1. User Agent (UA)
  2. SMTP sender
  3. SMTP receiver
- They are shown in Fig. 1.15.3. The mail is created by a user agent program in response to user input.



(G-643) **Fig. 1.15.3 : SMTP mail flow**

- Each created message consists of a header which includes the recipient's E-mail address and other information and the message body containing the message to be sent.
- These messages are lined up to form a queue and provided as input to an SMTP sender program.
- The SMTP sender takes messages from the queue and transmits them to the proper destination host via SMTP connection over one or more TCP connections to port 25.
- The SMTP protocol is used to transfer a message from the SMTP sender to SMTP receiver and it uses TCP connection for the same.
- The SMTP receiver accepts each arriving message and stores it in the user mail box.

- If the mail is to be forwarded then the SMTP receiver copies it to the outgoing mail queue.

1.15.4 SMTP Commands :

- The operation of SMTP consists of a series of commands and responses exchanged between the SMTP sender and receiver.
- The SMTP sender establishes the TCP connection to the receiver. After establishing the connection, the SMTP sender sends commands over the connections to the receiver.
- The SMTP receiver generates exactly one reply from the SMTP receiver. Table 1.15.1 shows the SMTP commands.
- Each command consists of a single line of text which begins with a four letter command code followed in some cases by an argument field.
- Most replies are a single line. However multiline replies also are possible.

Table 1.15.1 : SMTP commands

Name	Description
HELO	Send identification of the sender.
MAIL	Identifies originator of mail.
RCPT	Identifies recipient of mail.
DATA	Transfer message text.
RSET	Abort the current mail transaction.
NOOP	No operation.
QUIT	Close TCP connection.
SEND	Send mail to terminal.
SOML	Send mail to the terminal if possible, otherwise to mailbox.
SAML	Send mail to terminal and mail box.
VRFY	Confirm user name.
EXPN	Return membership of mailing list.
HELP	Send system-specific documentation.
TURN	Reverse role of sender and receiver.

1.15.5 SMTP Operation :

- The basic SMTP operation occurs in three phases :
  1. Connection setup
  2. Exchange of one or more command-response pairs
  3. Connection termination

1. Connection setup :

- The sender opens (i.e. creates) a TCP connection with the receiver. Once the connection is established, the receiver identifies itself with "220 Service Ready".
- The sender identifies itself with HELO command. The receiver accepts the sender's identification with "250 OK".

2. Mail transfer :

- Once the connection has been established, the SMTP sender may send one or more messages to SMTP receiver. There are three logical phases to transfer a message :
  1. A MAIL command identifies the originator of message.
  2. One or more RCPT commands identify the recipient for this message.
  3. A DATA command transfers the message text.

3. Connection closing :

- The SMTP sender closes the connection in two steps. First the sender sends a QUIT command and waits for a reply.
- 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.

1.15.6 Comparison of HTTP and SMTP :

Table 1.15.2 : Comparison of HTTP and SMTP

Sr. No.	SMTP	HTTP
1.	Message is transferred from client to server.	Message transfer is from client to server or the other way round.
2.	Uses TCP.	Uses TCP.
3.	Uses port 25 for transmission.	Uses port 80 for transmission.
4.	SMTP messages are to be read by humans.	HTTP messages are to be read and understood by the HTTP servers and HTTP clients.
5.	These messages are first stored and then forwarded.	These messages are immediately delivered.

1.16 Message Access Agent : POP and IMAP :

- The SMTP is used in the first and second stages of mail delivery. But SMTP is not used in the third stage, because SMTP is a push protocol which is meant for pushing the message from client to server.
- The third stage needs a **pull** protocol because the client has to pull messages from the server. The bulk data gets transferred from the server to client.
- Therefore third stage uses a message access agent which is a pull protocol.

- The two message access agents available are :
  1. Post Office Protocol, version 3 (POP 3).
  2. Internet Mail Access Protocol (IMAP 4).

1.16.1 POP 3 : SPPU : May 04, May 08, May 19

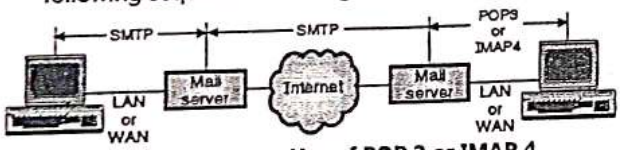
University Questions

Q. 1 Explain the role of SMTP and POP protocols in e-mail transfer. **(May 04, 8 Marks)**

Q. 2 What is the difference between IMAP and POP 3 protocols ? Explain when and where they are used ? **(May 08, 8 Marks)**

Q. 3 Write short note on POP3 and IMAP. **(May 19, 6 Marks)**

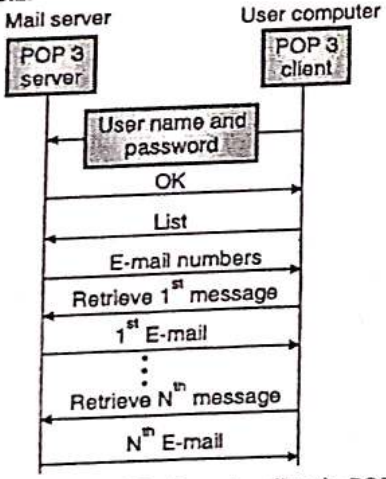
- The POP3 consists of client POP3 software and server POP3 software.
- Out of these, the client POP3 software is installed on the receiving computer whereas the mail server gets the server POP3 software installed on it.
- When the user wants to download email from the mailbox on the email server, the events take place in the following sequence. Refer Fig. 1.16.1.



(G-645) **Fig. 1.16.1 : Use of POP 3 or IMAP 4**

1. The client (user) establishes a connection with the server on TCP port 110.
2. The client then sends its user name and password to the server in order to access the mailbox.
3. The user is then allowed to list and get the mail messages one by one.

- This is called as downloading. It is illustrated in Fig. 1.16.2.



(G-647) **Fig. 1.16.2 : Downloading in POP3**

Modes of POP 3 :

- POP3 has two modes of operation :
  1. Delete mode and 2. Keep mode.
- **Delete mode :** In this mode the mail is deleted from the mailbox after each retrieval.
- This mode is used when the user is working on his permanent computer because it is then possible for him to save and rearrange the received mail after reading it.
- **Keep mode :** If operated in this mode, the mail remains in the mailbox after retrieval.
- This mode is used when the user accesses mail away from the primary computer. The read mail can be organized later.

Disadvantages of POP3 :

1. POP3 does not allow organization of email on the server.
2. The user can not create different folders on the server. It can create them only on his own computer.
3. The user can not partially check the contents of E-mail before down loading.

1.16.2 IMAP4 : SPPU : May 08, May 19

University Questions

Q. 1 What is the difference between IMAP and POP 3 protocols ? Explain when and where they are used ? **(May 08, 8 Marks)**

Q. 2 Write short note on POP3 and IMAP. **(May 19, 6 Marks)**

- Internet Mail Access Protocol Version 4 (IMAP4) is another mail access protocol which is very similar to POP3 but has more features.
- This makes IMAP4 more powerful but more complex as compared to POP3. IMAP is more sophisticated than POP3 and it is defined in RFC 1064.
- IMAP is ideal for a user having multiple computers such as a laptop on the road, PC at home and a workstation in office.
- IMAP maintains a central repository which can be accessed from any machine. So IMAP does not copy e-mail to the user's personal machine.
- An important feature of IMAP is its ability to address mail not by arrival number but by using attributes. That means the mailbox is like a relational database system than a linear sequence of messages.

Features of IMAP4 :

1. It is possible for the user to check the header before down loading.
2. It is possible for the user to search for the contents of E mail before downloading.
3. It is possible to partially download E mail.
4. It is possible for the user to create, rename or delete mailboxes on the mail server.
5. It is possible for the user to create a hierarchy of mailboxes in a folder for storing e-mails.

1.16.3 Comparison of IMAP and POP 3 :

SPPU : May 07, May 08, May 09, Dec. 09, Dec. 11, Dec. 12, Dec. 13, In Sem. March 20

University Questions

Q. 1 List the similarities and differences between POP3 and IMAP. From ISP point of view which protocol would be better and why ? **(May 07, 8 Marks, May 09, 10 Marks)**

Q. 2 What is the difference between IMAP and POP 3 protocols ? Explain when and where they are used ? **(May 08, 8 Marks)**

Q. 3 Differentiate between POP3 and IMAP. **(Dec. 09, 6 Marks, Dec. 11, Dec. 13, 8 Marks)**

Q. 4 List the similarities and differences between POP3 and IMAP. **(Dec. 12, 8 Marks, March 20, 4 Marks)**

Table 1.16.1 : Comparison of IMAP and POP 3

Sr. No.	Parameter	POP 3	IMAP
1.	Protocol is defined at	RFC 1939	RFC 2060
2.	TCP port used	110	143
3.	e-mail is stored at	User's PC	Server
4.	e-mail is read	Off line	On line
5.	Time required to connect	Small	Long
6.	Use of server resources	Minimal	Extensive
7.	Multiple mail boxes	Not possible	Possible
8.	Who backs up mailboxes	User	ISP
9.	For mobile users	Not good	Good
10.	User control over download	Little	Great
11.	Partial message downloads	No	Yes
12.	Simplicity in implementation	Yes	No
13.	Support	Wide spread	Increasing

1.17 File Transfer Protocol (FTP) :

SPPU : May 11, Dec. 12, Dec. 13, May 15, Dec. 18, In Sem. March 19, March 20

University Questions

Q. 1 What is FTP ? Where and when it is used ? Why does it require 2 ports ? Explain atleast 5 user commands used in FTP ? **(May 11, 10 Marks)**

Q. 2 Where and when FTP is used ? Explain the importance of two parts in FTP ? **(Dec. 12, 8 Marks)**

Q. 3 What is FTP ? Why it requires two ports ? Explain at least five user commands used in FTP. **(Dec. 13, 8 Marks)**

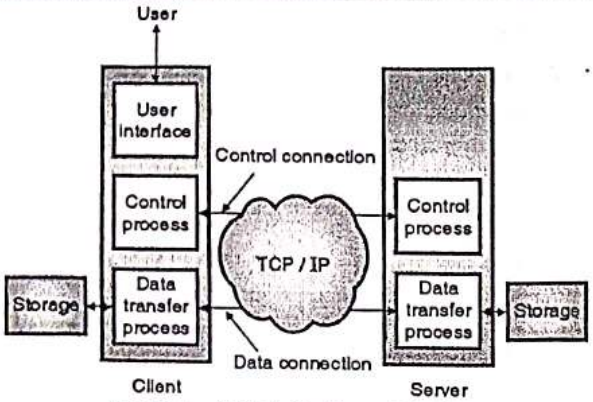
Q. 4 What is FTP ? Where and when is it used ? Why does it require two ports. **(May 15, 6 Marks)**

Q. 5 What is FTP ? Which ports does it use and for what purpose ? Explain any 4 commands in FTP. **(Dec. 18, 6 Marks)**

Q. 6 What is FTP ? Explain any four commands used in FTP. **(March 19, 6 Marks)**

Q. 7 What is FTP ? Where and when is it used ? Why does it require two ports ? List and explain any four FTP commands. **(March 20, 6 Marks)**

- A standard mechanism provided by the Internet which helps in copying a file from one host to the other is known as the File Transfer Program (FTP).
- Some of the problems in transferring files from one system to the other are as follows :
  1. Two systems may use different file name conventions.
  2. Two systems may represent text and data in different ways.
  3. The directory structures of the two systems may be different.
- FTP provides a simple solution to all these problems. The basic model of FTP is shown in Fig. 1.17.1.
- FTP establishes two types of connections between the client and server. One of them is used for data transfer and the other is for the control information.



(G-648)**Fig. 1.17.1 : Basic model of FTP**

- The fact that FTP separates control and data makes it very efficient. The control connection uses simple rules of communication.
- Only one line of command or a line of response is transferred at a time. But the data connection uses more complex rules due to the variety of data types being transferred.
- FTP uses port 21 for the control connection and port 20 for the data connection. Both these are well known TCP ports.
- As shown in Fig. 1.17.1 the client is made of three blocks namely :
  1. User interface
  2. Control process and
  3. Data transfer process.
- The server has two blocks : the control process and data transfer process. The control connection connects the control processes while data connection connects the data transfer processes as shown in Fig. 1.17.1.
- The control connection is kept alive during the entire interactive FTP session.
- The data connection is first opened, file is transferred and data connection is closed. This is done for transferring each file.

Control connection :

- This connection is created in the same way as the other application programs described earlier. Control connection remains alive during the entire process.
- The IP uses minimize delay type service because this is an interactive connection between a user and a server.

Data connection :

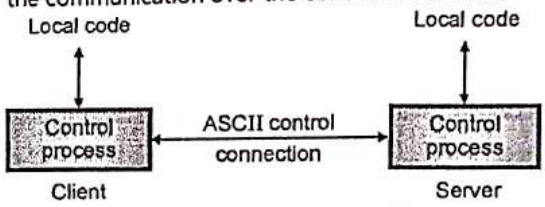
- Data connection uses the port 20 at the server site. This connection is opened when data to be transferred is ready and it is closed when transfer of data is over.
- The data connection does not remain open continuously like control connection. It is opened and closed many times as per requirement.

1.17.1 Communication in FTP :

- FTP operates in client – server environment. The two computers involved in communication may be different in terms of the operating systems, character sets, file structures and file formats etc.
- FTP can make them compatible. The approaches for communication over control connection and data connection are different from each other.

1. Communication over control connection :

- Refer Fig. 1.17.2 to understand the FTP's approach for the communication over the control connection.

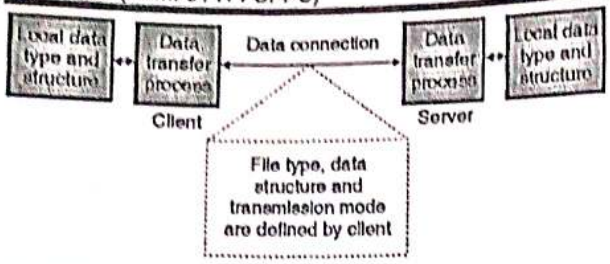


(G-649)**Fig. 1.17.2 : Communication over control connection**

- Similar to SMTP, FTP uses a set of ASCII characters to communicate across the control connection. Communication is achieved through a process of commands and response. One command is sent at a time.
- Each command or response is only of one short line. So it is not necessary to think about file format or file structure.
- Each line is ended with a two character token. The two characters used in the token are carriage return and line feed.

2. Communication over data connection :

- The purpose of implementing a data connection is to transfer a file. For this the client has to define the following :
  1. Type of file being transferred.
  2. Structure of data in the file
  3. Mode of transmission.
- Before the transmission over data connection, the communication over control connection is performed. Refer Fig. 1.17.3 to understand communication over data connection.



(G-650)**Fig. 1.17.3 : Communication over the data connection**

- The problem of heterogeneity is solved by defining three attributes of communication : file type, data structure and transmission mode.

1.17.2 File Types :

- FTP can use one of the following file types for transfer of data over the data connection :
  1. ASCII file 2. EBCDIC file 3. Image file.
- ASCII file is a text file, EBCDIC file can be transferred if both ends use EBCDIC encoding.
- Image file is the default format for transferring the binary files. With ASCII or EBCDIC files one more attribute must be added for defining the printability of the file.
- This attribute is nonprint or TELNET.

1.17.3 Data Structure :

- FTP can use one of the following data structures :
  1. File structure (default)
  2. Record structure and
  3. Page structure.
- File has no structure. It is simply a continuous stream of bytes. In the record structure the file is divided into records.
- This data structure is suitable only for the text files. In page structure, a file is divided into pages which can be stored or accessed randomly or sequentially.

1.17.4 Transmission Mode :

- FTP uses one of the following modes to transfer a file :
  1. Stream mode
  2. Block mode and
  3. Compressed mode.

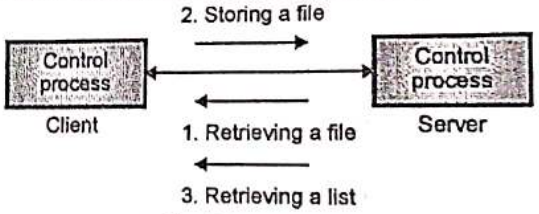
1. **Stream mode :**
- In this mode the data is delivered from FTP to TCP in the form of continuous stream of bytes. TCP chops this data into segments of appropriate size.
- Stream mode is the default mode of transmission.

2. **Block mode :**
- In this mode, data delivery from FTP to TCP takes place in the form of data blocks. Each such block is preceded by a 3 byte header.

3. **Compressed mode :**
- For big files the data can be compressed. Generally a run length encoding is used for compression.

1.17.5 File Transfer :

- File transfer takes place over the data connection and the commands are sent over the control connection. The commands supervise the data transfer.
- But file transfer in FTP means one of the following :
  1. **Retrieving a file :** Server copies a file onto a client.
  2. **Storing of a file :** A file can be copied from client to the server.
  3. A server sends a list of directory or file names to the client. FTP treats such a list of directory also as a file.
- The file transfer has been illustrated in Fig. 1.17.4.



(G-651)**Fig. 1.17.4 : File transfer**

1.17.6 FTP Commands :

SPPU : May 11, May 12, Dec. 13, Dec. 16, In Sem. March 20

University Questions

Q. 1 What is FTP ? Where and when it is used ? Why does it require 2 ports ? Explain atleast 5 user commands used in FTP ? **(May 11, 10 Marks)**

Q. 2 Explain atleast 8 important commands used in FTP. **(May 12, 4 Marks)**

Q. 3 What is FTP ? Why it requires two ports ? Explain at least five user commands used in FTP. **(Dec. 13, 8 Marks)**

Q. 4 State and explain six commands in FTP. **(Dec. 16, 6 Marks)**

Q. 5 What is FTP ? Where and when is it used ? Why does it require two ports ? List and explain any four FTP commands. **(March 20, 6 Marks)**

- The following commands are used for copying files using FTP.

Table 1.17.1 : FTP commands to transfer files

Command	Explanation
Get	Copy a file from remote host to local host
M get	Copy multiple files from the remote host to local host
Put	Copy a file from local host to remote host
M put	Copy multiple files from the local host to remote host

- FTP commands used to connect to a remote host are as shown in Table 1.17.2.

Table 1.17.2 : FTP commands to connect to a remote host

Command	Explanation
Open	Select the remote host and initiate login session
User	Identify the remote user ID
Pass	Authenticate the user
Site	Send the information to the remote host.

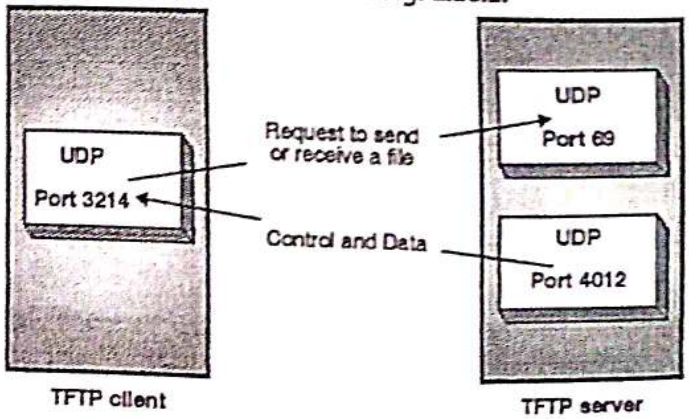
- FTP commands used to end an FTP session are as shown in Table 1.17.3.

Table 1.17.3 : FTP command to terminate session

Command	Explanation
Quit	Disconnect from the remote host and terminate FTP.
Close	Disconnect from the remote host but leave FTP client running.

1.18 TFTP :

- The **Trivial File Transfer Protocol (TFTP)** is a minimal protocol for transferring files without authentication and without any separation of control information and data as in FTP.
- In certain situations, the user needs to just copy a file and does not need all the features provided by the FTP protocol.
- Take the example of booting of a discless work station or a rooter.
- For booting them, it is only necessary to download the bootstrap and configuration files without using any sophistication of FTP. We use TFTP in such situations.
- TFTP is frequently used by devices without permanent storage for copying an initial memory image (bootstrap) from a remote server when the devices are powered on.
- Due to the lack of any security features, the use of TFTP is generally restricted.
- TFTP uses the unreliable transport protocol UDP for the transportation of data.
- TFTP is an extremely simple protocol. For a diskless work station the TFTP can be stored in a ROM because it needs to use only the basic IP and UDP.
- TFTP can perform only two functions – either read a file or write a file.
- In the **file reading** operation, a file is copied from the server site to the client site whereas in the **file writing** operation, a file is copied from a client site onto a server site.
- TFTP does not provide any security. TFTP uses the UDP services on the well known port 69.
- Each TFTP message is carried in a separate UDP datagram.
- The first two bytes of a TFTP message specify the type of message, which can be a request to download a file, request to upload a file, a data message, or an acknowledgement or error message.
- At the beginning of a TFTP session a TFTP client sends a request to upload or download a file from a UDP port to the (well-known) UDP port 69 of an TFTP server.
- When the request is received the TFTP server picks a UDP port of its own and uses this port to communicate with the TFTP client.
- Thus, both client and server communicate using ephemeral ports as shown in Fig. 1.18.1.



(G-652) Fig. 1.18.1 : TFTP

- Since UDP does not recover lost or corrupted data, TFTP is supposed to maintain the integrity of the data exchange.
- TFTP transfers data in blocks of 512 bytes. A 2 byte long sequence number is assigned to each block and is transmitted in a separate UDP datagram.
- A block must be acknowledged before the next block can be sent.
- When an acknowledgment is not received before a timer expires, the block is retransmitted.
- When the receiver receives a block that is less than 512 bytes long, it assumes that the end of file has been reached.

1.18.1 Applications of TFTP :

1. TFTP is used for basic file transfer application.
2. Used to initialize bridges and routers.
3. It is used in conjunction with DHCP to obtain the contents of the configuration file.

1.18.2 Advantages :

1. It is a very simple protocol.
2. It needs a very small storage space and can be stored on a ROM.

1.18.3 Disadvantages :

1. TFTP does not have flow control.
2. There is no error control mechanism.
3. There is no provision for security.

1.18.4 Comparison of FTP and TFTP :

SPPU : May 06, Dec. 07, Dec. 08, May 09, Dec. 09, May 12, Dec. 13, Dec. 14

University Questions

Q. 1 Differentiate FTP and TFTP. (May 06, 8 Marks)

Q. 2 Compare between FTP and TFTP. (Dec. 07, Dec. 08, May 09, Dec. 09, 8 Marks)

Q. 3 Differentiate between FTP and TFTP. (May 12, Dec. 13, 6 Marks)

Q. 4 Differentiate FTP and TFTP. (Dec. 14, 4 Marks)

Table 1.18.1 : Comparison of FTP and TFTP

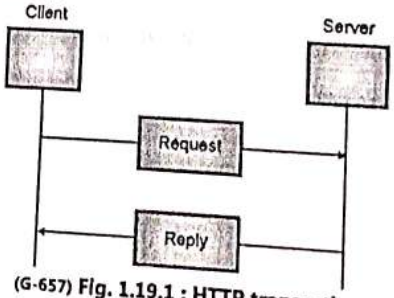
Sr. No.	Parameter	FTP	TFTP
1.	Operation	Transferring files	Transferring files
2.	Authentication	Yes	NO
3.	Control and data	Separated	Not separated
4.	Protocol	TCP	UDP
5.	Ports	21 – control, 20 – data	Port 3214, 69, 4012
6.	Data transfer	Reliable	Unreliable

1.19 HTTP (Hypertext Transfer Protocol) :

- The main function of HTTP is to access data on WWW. This protocol can access the data in various forms such as plaintext, hypertext, audio, video etc.,
- The function of HTTP is equivalent to a combination of FTP and SMTP. It uses services of TCP. It uses only one TCP connection (port 80).
- There is no separate control connection like the one in FTP. Only the data transfer takes place between the client and server so there is only one connection and it is the data connection.
- The data transfer in HTTP is similar to SMTP. The format of the messages is controlled by MIME like headers.

1.19.1 Principle of HTTP Operation :

- The principle of HTTP is simple. A client sends a request. The server sends a response. The request and response messages carry data in the form of a letter with a MIME like format.
- Fig. 1.19.1 shows the HTTP transactions between client and server.



(G-657) Fig. 1.19.1 : HTTP transaction

- The client initializes the transaction by sending a request message and the server responds by sending a response.

1.19.2 The Web and HTTP :

- HTTP is the Web's application layer protocol. It is the heart of the Web.
- It has been defined in [RFC 1945] and [RFC 2616].
- HTTP is implemented in two programs :
  1. A client's program 2. A server's program.

- These programs are executed on different and systems and talk to each other by exchanging HTTP messages.
- HTTP defines how Web clients such as browsers request Web pages from Web servers and how servers transfer Web pages to clients.
- HTTP uses TCP as its under lying transport protocol (rather than using UDP). The HTTP client first initiates a TCP connection with the server.
- After establishing a connection, the browser and the server processes access TCP through their socket interface. TCP provides a reliable data transfer service to HTTP.
- That means each HTTP request message, transmitted by a client will eventually arrive intact at the server. Similarly each HTTP response message transmitted by the server will eventually arrive intact at the client, due to the reliable TCP connection.
- Due to this kind of layered architecture HTTP need not have to worry about the lost data or about the details of how TCP deals with the loss and retransmission of data. It is managed by TCP.

Statelessness :

- In HTTP, the server sends the files requested to the client without storing any state information about the client.
- So it may happen that the same client may ask the same information repeatedly to the server and the server would not even understand it. So it will keep resending those files.
- As the HTTP servers does not maintain any information about the state of client it is called as a stateless protocol.

1.19.3 Non-persistent and Persistent Connection :

SPPU : May 08, May 11, May 12, May 16

University Questions

Q. 1 Differentiate between persistent and non-persistent HTTP connection. **(May 08, May 11, 8 Marks)**

Q. 2 Explain persistent and non-persistent HTTP connection. **(May 12, 10 Marks)**

Q. 3 Explain persistent and non persistent HTTP. **(May 16, 4 Marks)**

- HTTP is capable of using both non-persistent and persistent connections. HTTP uses persistent connection in its default mode.
- But HTTP clients and servers can be configured to use the non-persistent connection as well.

1. Non-persistent connections :

- Let us discuss the step-by-step procedure followed for transferring a web page from server to client for a non-persistent connection.
- Imagine that the web page consists of a base HTML file and many JPEG images and that all these objects reside on the same server.
- Let the URL for the base HTML file be as follows :

  http://www.vit.edu/itdept/home.index
- Then the sequence of events is as follows :

  1. The HTTP client process initiates a TCP connection to the server www.vit.edu on port number 80, which is the default port number for HTTP.
  2. The HTTP client, sends an HTTP request message to the server via its socket associated with the TCP connection. This request message is of the following format :

     Path name/itdept/home.index.
  3. The HTTP server process receives the request message via its socket associated with the connection. It then retrieves the object.

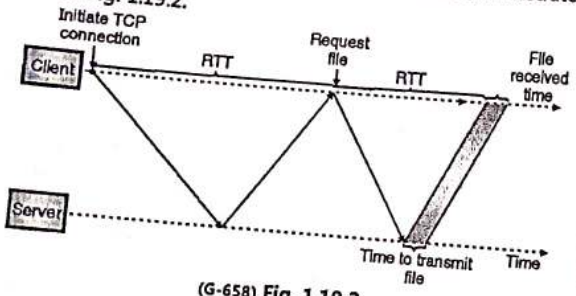
     /itdept/home.index from its storage.
- It then encapsulates this retrieved object in an HTTP response message and sends the response message to the client via its socket.

  4. The HTTP server process tells TCP to close the TCP connection.
  5. As soon as the HTTP client receives the response message, the TCP connection is terminated.
  6. The response message indicates that the encapsulated object is an HTML file. The client takes out the file from the response message and examines the HTML file. The client will find references to all the JPEG objects.
  7. The client follows the first four steps for each JPEG object.
- As the browser receives the web page, it displays the page. Different browsers can display the same web page differently.
- However HTTP is not concerned about this. Its specifications define only the communication between the HTTP client program and HTTP server program.

- The steps discussed earlier were for the **non-persistent** connection where each TCP connection is closed after the server sends the object.
- That means the TCP connection does not persist for other objects. Each TCP connection transports one request message and one response message.

Round-Trip Time (RTT) :

- The RTT is defined as the time taken by a small packet to travel from client to server and then back to the client.
- The components of RTT are :
  1. Packet propagation delays
  2. Packet queuing delays
  3. Packet processing delays.
- Now consider the sequence of events taking place when a user clicks on a hyperlink. These events are illustrated in Fig. 1.19.2.



(G-658) **Fig. 1.19.2**

1. The browser initiates a TCP connection between the browser and web server. This process makes use of a **three way handshake.**

- In the three way handshake, the client sends a small TCP segment to the web server. The server acknowledges and responds with another small TCP segment. Finally the client acknowledges back to the server.

2. After completing the first two parts of the three way handshake the client sends the HTTP request message to the server.
3. In response the server sends the HTML file to the client. The total response time as shown in Fig. 1.19.2 is equal to 2RTT plus the time taken by the server to transmit the file.

Disadvantages of non-persistent connections :

1. It is necessary to establish and maintain a new connection for each requested object.
2. For each connection TCP buffers need to be allocated and TCP variables need to be kept in both the client and server.
3. There is a delay of 2RTTs associated with the transfer of each object.

2. Persistent connection :

- The disadvantages of non-persistent connections can be overcome if persistent connection is used. With the persistent connection, the server leaves the TCP connection open after sending a response.
- All the requests and responses between the same client and server can be sent over the same connection. Hence the entire web page can be sent over a single persistent connection.
- It is also possible to send the multiple web pages residing on the same server to the same client over a single persistent TCP connection. The TCP connection is closed only after the time out interval by the HTTP server.

Types of persistent connections :

- The two versions of persistent connections are as follows :

  1. Without pipelining 2. With pipelining.

1. Without pipelining :

- For this version, the client has to issue a new request only when it receives the previous response. The delay of only one RTT is experienced by the client in order to request and receive each object.
- This is an improvement over the non-persistent connection which experiences a delay of 2RTT. This delay can be reduced by using pipelining.
- Another disadvantage of no pipelining is that the TCP connection becomes idle i.e. does nothing while it waits for another request after the server had sent an object.

2. With pipelining :

- This mode reduces the delay further. The default mode of HTTP uses persistent connection. With pipelining the HTTP client will issue a request as soon as it encounters a reference.
- This allows the HTTP to make back to back requests. It can make a new request before receiving the response.
- When the server receives back to back requests, it sends the objects back to back.
- With pipelining only one RTT will be expended for all the referenced objects. Another advantage is that the pipelined TCP connection remains idle for a very short time.

1.19.4 HTTP Messages : SPPU : Dec. 13

University Questions

Q. 1 Explain two types of messages used in HTTP. (Dec. 13, 8 Marks)

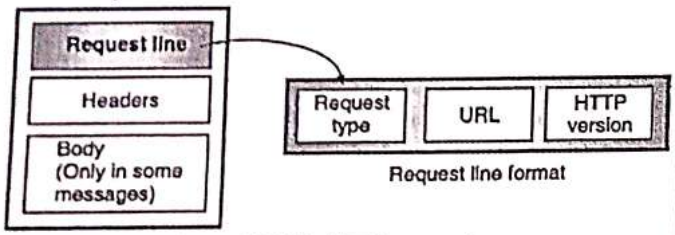
- The HTTP messages are of two types :
  1. Request message
  2. Response message.
- The format of both these messages is almost the same.

1.19.5 Request Message : SPPU : Dec. 13

University Questions

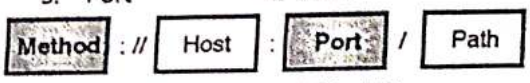
Q. 1 Explain two types of messages used in HTTP. (Dec. 13, 8 Marks)

- Fig. 1.19.3(a) shows the format of the request message. It consists of a request line, headers and sometimes a body.



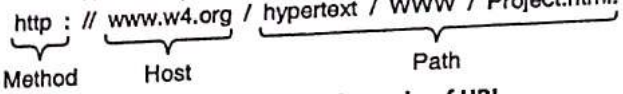
(G-659) **Fig. 1.19.3(a) : HTTP request message**

1. **Request line :**
   - The request line is used for defining the request type, resource (URL) and HTTP version as shown in Fig. 1.19.3(a).
   - **Request type :** Several request types are defined.
   - **Uniform Resource Locator (URL) :** The client accessing a web page needs an address. The HTTP uses the URL to facilitate the access of any document distributed over the world. The URL defines four thing as shown in Fig. 1.19.3(b). They are as follows :
     1. Method 2. Host computer
     3. Port 4. Path.



(G-660) **Fig. 1.19.3(b) : URL**

- Method is the protocol used such as FTP, HTTP. Host is the computer where the required information is located. The name of the computer begins with www but this is not mandatory.
- URL can optionally contain the server's port number. If the port is included then it should be inserted between host and path and it should be separated by a colon.
- Path is the name of the file where the information is located.
- **Version :** The latest version of HTTP is 1.1 but the versions 0.9 and 1 are also used.
- The example of URL is shown in Fig. 1.19.3(c).



(G-1969) **Fig. 1.19.3(c) : Example of URL**

1.19.6 Methods (Request Type) :

- This is one of the fields in the request line format. It defines different types of messages referred to a request types or methods.
- The request method is a command or request issued by the client to the server.
- Following are some of the important methods (request types).

1. **GET :**

   The client uses this method for retrieving a document from the server. The address from where this document is to be obtained is defined in the URL.

2. **HEAD :**

   The client uses this method in order to obtain some information about a document but not the document itself.

3. **POST :**

   This method is used when the client wants to provide some information to the server.

4. **PUT :**

   This is used by the client for providing a new or replacement document to be stored on the server.

5. **PATCH :**

   This method is similar to PUT. But there is one change. The patch request contains a list of differences which should be implemented in the existing file.

6. **COPY :**

   This method is used to copy a file to another location.

7. **MOVE :**

   This method is used for moving a file to another location.

8. **DELETE :**

   It is used for removing a document on the server.

9. **LINK :**

It is used for creating a link or a link from a document to another location. The location of the file is specified in the URL request line and the location of destination is specified in the entity header.

10. **UNLINK :**

It is used for deleting the links created by the LINK method.

11. **OPTION :**

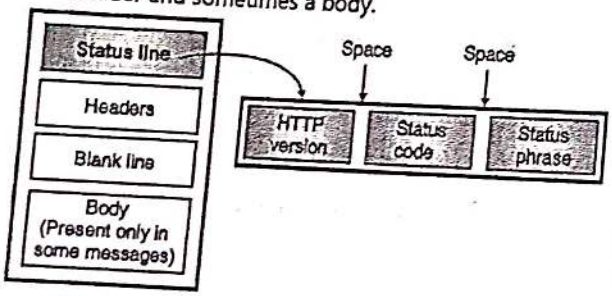
It is used by the client to ask the server about various options that are available.

1.19.7 Response Message : SPPU : Dec. 13

University Questions

Q. 1 Explain two types of messages used in HTTP. **(Dec. 13, 8 Marks)**

- Fig. 1.19.4(a) shows the format of the response message. A response message is made of a status line, a header and sometimes a body.



(a) Response message **(b) Status line format**

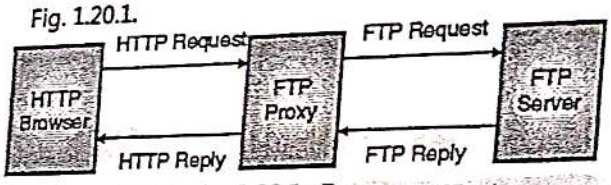
(G-662) Fig. 1.19.4

Status line :

- The status line is used for defining the status of the response message. As shown in Fig. 1.19.4(b) it consists of HTTP version, status code and status phrases with spaces in between.
- **HTTP Version :** This field indicates the version of HTTP being used. This field is same as the HTTP version field used in the request line.
- **Status Code :** It is a three digit field which is similar to those in FTP and SMTP protocols.
- **Status Phrase :** It is used for explaining the status code in the text form.

1.20 Proxy Server :

- All the servers cannot speak HTTP some of them use the FTP, Gopher or some other protocols. A large information is available on FTP and Gopher servers so it should be made available to web users.
- To do so, one solution can be to have a browser which can use the HTTP as well as FTP, Gopher and other protocols.
- But this makes the browser unnecessarily large. The other solution to this problem is proxy server, shown in Fig. 1.20.1.



(G-656) Fig. 1.20.1 : Proxy server

- Proxy server is basically a gateway which communicates using HTTP to the browser FTP, Gopher or some other protocol for communicating to the server.
- It receives HTTP requests from a browser, converts them in FTP or Gopher requests and sends them to the FTP/Gopher server as shown in Fig. 1.20.1.
- Proxy server can be a program running on the same machine working as a browser or it can be a separate machine.
- The users can configure their browsers with proxies for those protocols which the browser does not use for communication.
- The other important feature of a proxy server is caching. A caching proxy server collects and stores all the pages which pass through it.
- When a user asks for a page, the proxy server will first see if it has the page stored with it. If the page is there then it will see if the page is up to date.
- If the page is updated then, it passes the page to the user otherwise it will fetch a new copy of the page. A proxy server can be put inside a firewall. The user can access the web but he is not allowed the full Internet access.
- In such situation the user talks to the proxy server and the server communicates with obtains different sites and obtains pages on behalf of the user.

1.20.1 HTTP Security :

- As such HTTP does not provide any security. But it can be run over the secured socket layer (SSL). If so, the HTTP is called as HTTPS.
- The security features of HTTPS include confidentially, authentication of client and server and data integrity.

1.21 Remote Login : TELNET and SSH :

- The Internet and TCP/IP suite have been designed primarily to provide service to its users.
- The requirements of different users will be of different types and with increase in the number of users, the number of diversified demands will also be very large.
- It is practically impossible to write a specific client - server program for each demand.
- Therefore a general purpose client - server program should be developed which will help a user to access any application on a remote computer.
- That means a user will be allowed to log into a remote computer. Two of such general purpose client - server programs which allow remote login are : TELNET and SSH.

1.21.1 TELNET :

- The long form of TELNET is TErminal NETwork. It was proposed by ISO as a standard TCP/IP protocol for a virtual terminal service.
- TELNET enables a user to establish a connection to a remote system.

Concepts related to TELNET :

- Some of the important concepts related to TELNET are as follows :
  1. Time sharing environment.
  2. Login : Local or Remote.
  3. Network Virtual Terminal.

Time Sharing Environment :

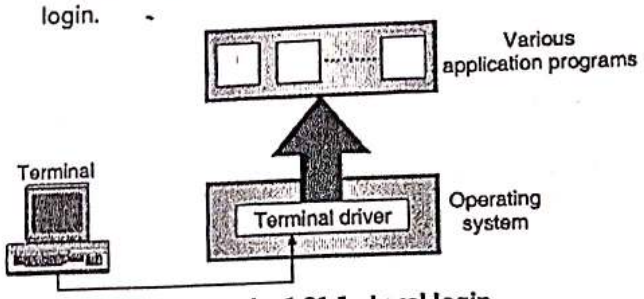
- TELNET was designed during those days when almost all the operating systems were operating on the time - sharing principle.
- In the time sharing environment there is a large central computer which supports all the users. All the processing is done by the central computer, and each user feels that it is a dedicated computer.
- The users can access all the common system resources, use all the programs or switch from one program to the other.

Login :

- In a system based on time sharing, every user must have an identification and a password for his authentication.
- Whenever a user wants to access the system he will log into the system with his user id and password.
- The system will check the password to allow only the authorized users to access the resources.
- The logic can be one of the following two types :
  1. Local login. 2. Remote login.

1. Local login :

- The user login into a local time sharing system is called as local login. Fig. 1.21.1 illustrates the principle of local login.

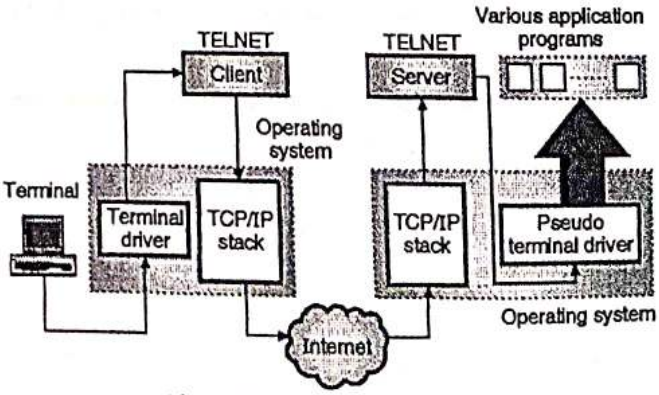


(G-1793) **Fig. 1.21.1 : Local login**

- The local login takes place in a step - by - step manner as follows :
  1. The user types at the keyboard of a terminal.
  2. The terminal driver accepts these keystrokes.
  3. It converts the keystrokes to characters.
  4. It passes the characters to operating system.
  5. The O.S. understands the combination of characters.
  6. It allows access of intended application to the user.

2. Remote Login :

- The user will have to go for the remote login process when he wants to access an application program residing on a remote computer.
- He can do it using the TELNET client and server programs. Fig. 1.21.2 illustrates the principle of remote login.



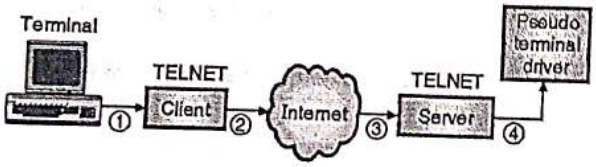
(G-1794) **Fig. 1.21.2 : Principle of remote login**

- Remote login takes place in a step-by-step manner as follows :
  1. The user types at the keyboard of a terminal.

2. The terminal driver at local O.S. accepts the characters but sends them to TELNET client without interpreting them.
3. TELNET client converts them into NVT characters. NVT is Network Virtual Terminal. This is a universal character set.
4. NVT characters are delivered to TCP/IP stack (local).
5. The NVT characters travel on the Internet and reach the TCP/IP stack of the remote machine.
6. The NVT characters are applied to the TELNET server which converts them appropriately so that the remote computer can understand them.
7. These characters are applied to a software called pseudo terminal driver.
8. The O.S. at the remote machine then passes the character to the intended application.

1.21.2 Network Virtual Terminal (NVT) :

- NVT character set is a universal interface defined by TELNET in order to ensure that a user can access any remote computer in this world. Fig. 1.21.3 illustrates the concept of NVT.



① - Local computer character set
② and ③ - NVT character set
④ - Remote computer character set

(G-1795) **Fig. 1.21.3 : Concept of NVT**

- The local computer character set is used for the communication between the user terminal and TELNET client.
- Then between the TELNET client and TELNET server the communication takes place using the NVT character set.
- And finally the remote computer character set is used for the communication between the TELNET server and the pseudo terminal driver as shown in Fig. 1.21.3.
- NVT has two sets of characters. One set is for the data and the other set is for control. Both have 8 bit characters.

1.21.3 Security Problems of TELNET :

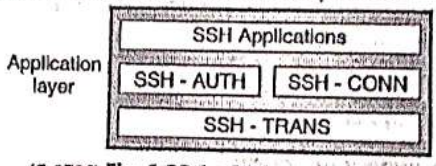
- TELNET is not a very secured system. It needs username and password for logging in. But it is not enough.
- A snooper software would be enough to capture the login name and password even if they are encrypted.

1.22 Secure Shell (SSH) :

- Secure Shell or SSH is another popular remote login application program.
- The underlaying transport program for SSH is TCP. This is similar to TELNET.
- However SSH has two advantages over TELNET :
  1. It is more secured than TELNET.
  2. It provides more services.
- There are two versions of SSH namely SSH_1 and SSH_2, out of which SSH_2 is being used. We will discuss SSH_2 in this section. Note that these two versions are not compatible to each other.

SSH Components :

- This is a proposed application layer protocol and as shown in Fig. 1.22.1, it has four components.



(G-1796) **Fig. 1.22.1 : SSH components**

- The four SSH components are :

1. SSH – TRANS.	3. SSH – CONN.
2. SSH – AUTH.	4. SSH – Applications.

1. SSH – TRANS :

- The long form is SSH – Transport Layer Protocol. TCP is not a secured protocol, therefore SSH makes use of a protocol which creates a secured channel on top of TCP.
- This new secured channel is an independent protocol called SSH – TRANS.
- When SSH is used, the client and server will first establish an unsecured TCP connection and then develop a secured layer over this by exchanging various security parameters.
- The SSH – TRANS protocol provides the following services :
  1. Confidentiality of the messages.
  2. Data integrity of the exchanged messages.
  3. Authentication of the server.
  4. Message compression.

2. **SSH – AUTH :**

- The second component of SSH is the SSH – AUTH i.e. SSH – Authentication protocol.
- This protocol is used to authenticate the client for the server after establishing a secure channel between client and the server.

3. **SSH – CONN :**

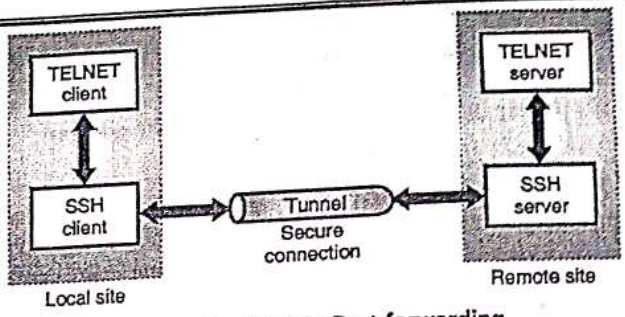
- The third component of SSH is SSH – CONN. i.e. SSH connection protocol.
- This piece of software is called for by the SSH once a secure connections has been established and authentication done.
- SSH – CONN performs the multiplexing as one of its services.
- It allows the client to create multiple logical channel over the secure channel established between the client and the server.

4. **SSH – Applications :**

- As soon as the connection establishment, authentication etc. is complete, the SSH connection can be used by multiple applications.
- Each application can create its own logical channel and make use of secure SSH connection. In addition to the remote login, the other applications that make use of SSH are : file transfer application. That is called as secure file transfer.

1.22.1 Port Forwarding :

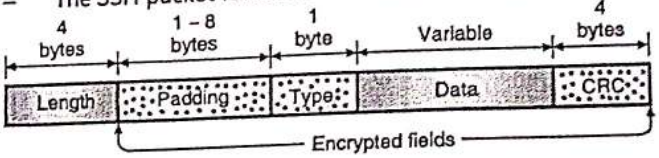
- Port forwarding is one of the services provided by the SSH – protocol.
- The port forwarding mechanism can be used to access application programs which do not provide any security. e.g. TELNET or SMTP.
- Such application programs can use the secure channel created by SSH to create a tunnel to carry the messages as shown in Fig. 1.22.2. Therefore this mechanism is also called as SSH Tunneling.
- We can apply the port forwarding concept to change the insecure connection between TELNET client and TELNET server into a secure connection, as shown in Fig. 1.22.2.



(G-1797) **Fig. 1.22.2 : Port forwarding**

1.22.2 SSH Packet Format :

- The SSH packet format is as shown in Fig. 1.22.3.



(G-1798) **Fig. 1.22.3 : SSH packet format**

- Description of various fields are as follows :

1. **Length :**

   This is a 4-byte long field which defines the length of the SSH packet which includes the type, the data and the CRC fields but does not include the length and the padding fields.

2. **Padding :**

   This is a variable length field. Its length can vary from 1-byte to 8-bytes. Padding field will make the attack on security more difficult.

3. **Type :**

   This is a 1-byte field which is used to specify the type of packet used by the SSH protocol.

4. **Data :**

   This is a variable length field. We can obtain the length of the data field by deducting the 5-bytes from the value of the length field.

5. **CRC :**

   This 4-bytes long field is used for error detection purpose.

1.22.3 Comparison of TELNET and SSH :

Table 1.22.1 : Comparison of TELNET and SSH

Sr. No.	Parameter	TELNET	SSH
1.	Port number	Uses TCP port number 23.	Uses TCP port number 22.

Sr. No.	Parameter	TELNET	SSH
2.	Security	Less secured than SSH.	Highly secured
3.	Data format	Telnet sends the data in plain text.	SSH sends all the data in encrypted format. SSH uses secure channel to transfer data over the network.
4.	Authentication	No authentication mechanisms.	SSH uses public key encryption in order to authenticate the remote users.
5.	Use of bandwidth	Low	High
6.	Suitable for	Private networks	Public networks
7.	Operating system	Used in linux and windows operating system	All popular operating system

1.23 Host Configuration : DHCP :

- DHCP (Dynamic host configuration protocol) is the first client server application program that is used after a host is booted.
- Thus it works as a bootstrap when the host is booted and is to be connected to the Internet, but does not know its IP address.
- A computer that makes use of the TCP/IP suite must know its IP address.
- Along with its IP address it must also know the following information :
  1. Subnet mask of the computer
  2. IP address of the router, so that it can communicate with other networks.
  3. IP address of the name server so that it can use the names instead of addresses.
- All this information can be saved in a configuration file and accessed by computer when booting takes place. This is known as host configuration process.
- But what will happen if the workstation is diskless or the computer is with a disc but it is being booted for the first time.
- If a computer is diskless, then it is possible to store the operating system and networking software in the ROM.
- But this information is not known to the manufacture and therefore cannot be stored in ROM.
- This information is dependent on the configuration of individual machine and it defines which network the machine is connected to.

1.23.1 Previously used Protocols :

- Now a days DHCP has become the formal protocol for host configuration. But the two protocols which were used earlier for the same purpose were RARP and BOOTP.
- RARP is Reverse Address Resolution Protocol and BOOTP stands for Bootstrap protocol.

1.23.2 DHCP : SPPU : In Sem, March 20

University Questions

Q. 1 What is DHCP ? What are it's advantages ? Explain various messages used in DHCP ? (March 20, 4 Marks)

- The Dynamic Host Configuration Protocol (DHCP) was devices by IETF in order to make the configuration automatic.
- Thus DHCP does not require an administrator to add an entry for each computer, to the database that a server uses.
- Instead, in DHCP a mechanism is provided for any computer to join a new network and obtain an IP address automatically with no manual intervention. This is known a plug and play networking.
- Thus DHCP allows the use of computers that, run server software as well as computers that run client software.
- When a computer that runs client software is shifted to a new network, it can use DHCP to obtain configuration information automatically.
- DHCP assigns a permanent address to a nonmobile computer that run server software.
- This address will not change when the computer **reboots.**
- To accommodate both type of computers, DHCP makes use of a client server approach.
- When a computer boots, it will broadcasts a DHCP Request. In response a server sends a DHCP Replay. An administrator can configure a DHCP server to have two types of addresses.

- First is the permanent address that are assigned to server computers, and second type is a pool of addresses which can be assigned on the basis of demand, when a computer boots and sends a request to DHCP.
- The DHCP find the configuration information by accessing its database If the database contains a specific entry for the computer then the server returns the information from the entry.
- However if there is no such entry exists for the computer, then the server chooses the next IP address from the pool and assigns it to the computer.

What is DHCP :

- DHCP, as the name suggests, is a protocol used for dynamically configuring the hosts on a network, such as workstations, personal computers and printers.
- DHCP can help in assigning various types of information such as routing information, directory-services information and default web server and mail servers.
- However, the most important and commonly used information for which DHCP is used is the IP address and subnet mask information.
- DHCP was primarily designed for managing the network and the clients automatically. With DHCP, it is not necessary to configure the network and client information manually for individual hosts.
- In addition, DHCP can coexist with statically configured hosts with fixed IP addresses. DHCP can also carry out the allocation of certain configuration information to a host on a permanent basis.
- This protocol provides a four point information (IP address, subnet mask, IP address of router, IP address of name server) to a diskless computer or to a computer which is booted for the first time.
- It is a client / server protocol which is backward compatible to the BOOTP.

1.23.3 Advantages of DHCP :

SPPU : In Sem. March 20

University Questions

Q. 1 What is DHCP ? What are it's advantages ? Explain various messages used in DHCP ? (March 20, 4 Marks)

- The use of DHCP on a network offers the following advantages :
  1. It sets free the network administrator from the duties of setting up the configuration information, such as the IP address, the subnet mask, and the routing tables, manually. The DHCP simplifies network administration by doing these tasks automatically.
  2. Avoids this and the sometimes the same IP address is assigned to two different hosts. The DHCP avoids this and the consequent malfunctioning of both the hosts from happening.
  3. If the DHCP was not used, then the movement of computers from one network to another requires must be reconfigured. With DHCP, you can move the computers to different subnets or networks without the need to reconfigure them. In such situations, DHCP takes care of IP address assignment and other configuration details.
  4. Mobile computers, such as laptops and palmtops, can easily get connected to different networks. They don't require reconfiguration any more as they get their configuration information from the DHCP server.
  5. DHCP allocates IP addresses from a pool of IP addresses. In addition, when a computer gets disconnected, its released IP address is returned to the resource pool. Therefore, the possibility of having unused IP addresses are minimized.

1.23.4 Components of DHCP :

- The use of DHCP on a network requires the following three components :

1. **DHCP server :**

- It assigns the IP address and other information to the clients when they request for the information.

2. **DHCP client :**

- It communicates with the DHCP server to get the desired information regarding its configuration. This communication can take place when the computer starts.
- The user of the DHCP client can also initiate a DHCP client request to the DHCP server to renew its information.

3. **DHCP relay agent :**

- It is used to relay (forward) client requests to the DHCP server.

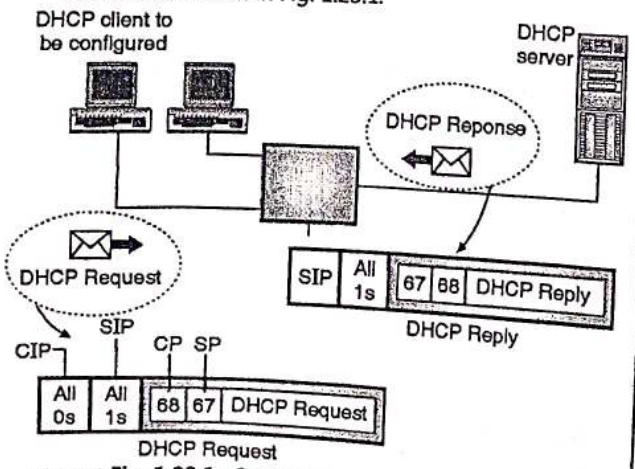
- This is required when the DHCP server is yet to assign the client an IP address.
- Without an IP address, a client cannot use IP routing on its own.
- A DHCP relay agent helps the client to communicate with the DHCP server when the client does not have an IP address.
- When a client starts, it has an IP address of 0.0.0.0. It sends a broadcast message containing its MAC address and the computer name.
- In response the DHCP server sends an offer message that contains the MAC address of the client, the IP address offered to that client, the lease period for which the IP address will remain valid and its own IP address.
- The lease period is the time duration for which a client can use the IP address that has been assigned to it by the DHCP server.
- You can configure a DHCP server to set the lease time.
- When the client receives the IP address, it accepts the offer and then broadcasts the message that it has accepted the offer.

1.23.5 DHCP Operation :

- We will discuss the DHCP operation under two different operating conditions :
  1. DHCP client and server on the same network.
  2. DHCP client and server on different networks.

Operation on the same network :

- This situation is not a very common one. But sometimes the DHCP client and server happen to be on the same network as shown in Fig. 1.23.1.

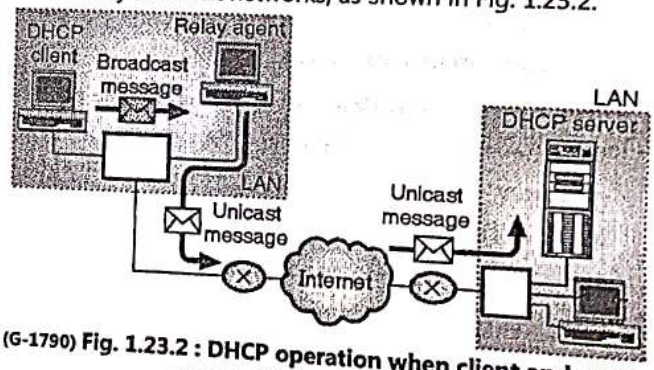


(G-1789) **Fig. 1.23.1 : Operation of DHCP when client and server are on the same network**

- The operation takes place as follows :
  1. The DHCP server sends a passive open command on port 67 of UDP and waits for clients response.
  2. The DHCP client sends an active open command on port 68 of UDP. This message is encapsulated in the UDP datagram with port 67 as destination port and port 68 as the source port. The UDP datagram is then encapsulated in an IP datagram. Note that the client at this time does not know its own IP address (i.e. the source address) and the server's IP address (destination address). Therefore the client uses an all zero address as source address and an all one address as destination address.
  3. The server responds to this message by sending either a broadcast or a unicast message using port 67. It uses port 68 as the destination port. Broadcast address is used only for those system which do not allow the bypassing of ARP.

1.23.6 DHCP Operation on Different Networks :

- In this situation the DHCP client and server are on two entirely different networks, as shown in Fig. 1.23.2.



(G-1790) **Fig. 1.23.2 : DHCP operation when client and server are on different networks**

- In this situation a problem arises due to the broadcast nature of DHCP request. The client does not know the IP address of the server.
- Hence the DHCP request is a broadcast type (all 1s IP address). Any server does not allow the broadcast request to pass through it. So this request cannot reach the DHCP server.
- In order to solve this problem we can configure one of the hosts or router to operate as a relay agent as shown in Fig. 1.23.2.
- The relay agent knows the unicast address of the DHCP server. The relay will look for the broadcast request on port 67.

- As soon as it receives the broadcast request message, it encapsulates this message in a unicast datagram and sends it to the DHCP server.
- Such a unicast message is allowed to pass through by any router. Thus the request message reaches the DHCP server.
- The DHCP server sends its reply to the relay agent which in turn sends it to the DHCP client.

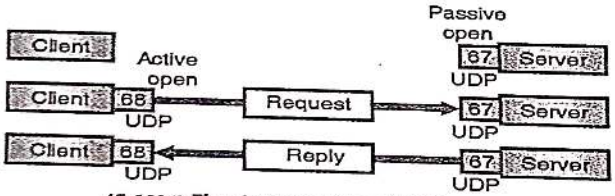
Note : In Fig. 1.23.2 only the message between the relay agent and client is it broadcast type. All the other messages are unicast types.

1.23.7 UDP Ports : SPPU - In Sem. March 20

University Questions

Q. 1 What is DHCP ? What are it's advantages ? Explain various messages used in DHCP ? **(March 20, 4 Marks)**

- The interaction between a client and DHCP server has been shown in Fig. 1.23.3.



(G-1994) Fig. 1.23.3 : Use of UDP ports

- The well known port 67 is used by the server, which is normal.
- But the client uses the well known port 68, which is not normal. It is unusual. Why does a client choose the well known port 68 rather than an ephemeral port ?
- The answer is for prevention of a problem when the reply from the server to client is of **broadcast** type. In order to understand the exact nature of the problem, let us assume that an **ephemeral port** is used instead of the well known port 68 and study its effect.
- Suppose host A on a network is using a DHCP client. It is using the ephemeral port say 2017 which we have chosen randomly.
- On the same network, there is another host B, which is using a DAYTIME client on ephemeral port 2017 which is accidentally the same.
- In this situation, the DHCP server sends a broadcast reply message with the destination port number 2017 and broadcast IP address $FFFFFFFF_{16}$.
- Every host has to open a packet which carries this destination IP address. Host A would find a message from an application program on ephemeral port 2017.
- Thus the DHCP client receives a **correct message** but the DAYTIME client receives an **incorrect message.**
- This confusion takes place due to the process of demultiplexing which is based on the **socket address.**
- Remember that a socket address is the combination of IP address and port number and both are same in this case.
- If a well known port (less than 1024) is used then the use of same two destination port numbers would be prevented.
- It would not be possible for host B to select port 68 as the ephemeral port due to the fact that ephemeral port numbers are greater than 1023.
- The final question is what happens if host B is also running the DHCP client ? The answer is that because of the same socket address, both the clients will receive the message.
- In order to handle such a situation, **the third identification number** is used to differentiate the clients.
- In DHCP this another number is called as the **transactional ID** and for each DHCP connection, it is chosen randomly. It is almost impossible that both the host choose the same transactional ID.

1.23.8 Using TFTP :

- Note that all the information needed by a client for booting purpose is not sent by the server.
- The server, in its reply message will define the **pathname** of a file in which all the booting information needed for the client is sent.
- The client can then use a TFTP message that is encapsulated in a UDP user datagram, to obtain the remaining necessary information.

1.23.9 Error Control :

- DHCP can use either UDP (as discussed) or TFTP. Note that UDP does not provide any error control.
- Then what should be done if a request is lost or damaged ? OR if the reply is damaged ?
- As UDP does not provide any error control, the DHCP should provide it.

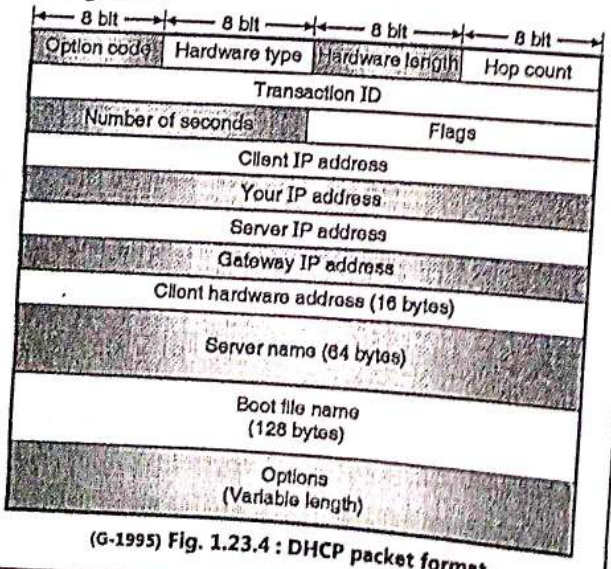
- Two strategies could be used to achieve the goal of error control :
  1. Ask UDP to use checksum. The UDP has an option of using the checksum.
  2. Ask DHCP client to use timers alongwith the retransmission policy if DHCP request or reply gets damaged or lost.

1.23.10 Optimizations in DHCP :

- The DHCP protocol has following steps :
- The first step is that a computer **broadcasts** a DHCP discover message in order to find DHCP server, and the other step is that the computer selects one of the available DHCP servers that responds to its message and sends a request to that server.
- To avoid a situation in which a computer follows both steps each time its boots or each time it needs to extend the lease, DHCP uses caching.
- When a computer discovers a DHCP server, the computer saves the address of that server in a cache on **permanent** storage (e.g. a disk file).
- Similarly, once an IP address has been allotted to it the computer saves the IP address in a cache. When a computer reboots, it uses the cached information to revalidate its former address.
- Doing so saves time and reduce network traffic.

1.23.11 Packet Format :

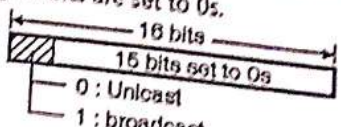
- The format of a DHCP packet has been shown in Fig. 1.23.4.



(G-1995) Fig. 1.23.4 : DHCP packet format

- Let us describe each field in the DHCP packet.

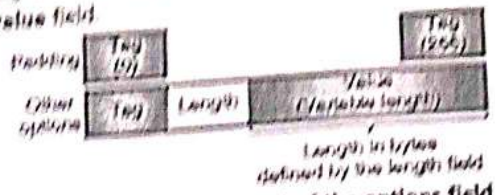
1. **Operation code :**
   - This is an 8 bit field which is used to define the type of DHCP packet. If this field contains (1) then the packet is **request type** and if this field contains (2) then the packet is **reply type.**
2. **Hardware type :**
   - This 8-bit field is used to define the type of physical network. An integer has been assigned to each type of network e.g. the value of this field is 1 for Ethernet.
3. **Hardware length :**
   - This is an 8-bit field which is used for defining the length of the physical address in bytes. The value of this field is 6 for Ethernet because the physical address of Ethernet is 6 byte long.
4. **Hop count :**
   - This is an 8-bit field which is used for define the maximum number of hops a packet can travel.
5. **Transaction ID :**
   - This is a 32-bit or 4-byte long field which carries an integer in it.
   - The contents of this field are known as transaction **identification** and it is set by the client. This field is used for matching a reply with the request.
   - The same value is returned by the server in its reply packet.
6. **Number of seconds :**
   - This is a 16-bit field which is used to indicate the amount of time (in seconds) elapsed from the instant at which the client started to boot.
7. **Flag :**
   - This is a 16-bit long field, as shown in Fig. 1.23.5. Out of these 16 bits, only the leftmost bit is used and the remaining 15 bits are set to 0s.



(G-1996) Fig. 1.23.5 : Format of the flagfield

   - The leftmost bit is used to specify a forced broadcast reply (instead of unicast) from the server.
8. **Client IP address :**
   - This 4-byte long field is used to carry the client IP address. A "0" in this field indicates that the client does not have this information.

5. **Your IP address :**
   - This is also a 4-byte long field which is used to carry the clients IP address. This address is requested by the client and filled by the server in the reply message.
10. **Server IP address :**
   - This is also a 4-byte long field which contains the IP address of the server. This address is sent by the server in the reply message.
11. **Gateway IP address :**
   - This is a 4-byte or 32 bit long field that contains the IP address of a router which is filled in the reply message by the server.
12. **Client hardware address :**
   - This is a 16-byte field which contains the physical address of the client.
13. **Server name :**
   - This is a 64 byte long field which is filled on the optional basis by the server in a reply packet.
   - This field consists of a null terminated string containing the domain name of the server.
   - If no information about the server name is to be given, then the server should fill up this field with all zeros.
14. **Boot filename :**
   - This is a 128-byte field which contains a null terminated string consisting of full pathname of the boot file.
   - This path can be used by the client in order to obtain additional information about booting. This field is filled by the server in the reply message on the optional basis.
   - If the server does not want to fill data in this field, then the entire field should be filled up with 0s.
15. **Options :**
   - This is a 64-byte field which can be used for a dual purpose as follows :
     1. It is used to carry some additional information such as default router address or network mask.
     2. Or it is used to carry some specific information about the vendor.
   - It is important to note that, the **options** field is used only in the **reply message**. The server makes use of a number called **magic cookie.**
   - After finishing reading of the message the client searches for the magic cookie.
   - If it is present, then the next 60 bytes data will correspond to **options**. Fig. 1.23.6 shows the format of the option. It consists of three fields as follows : a 1-byte **tag** field, a 1-byte **length** field and a variable length **value** field



(5-1997) **Fig. 1.23.6 : Format of the options field**

   - The function of the length field is to specify the length of the variable length **value** field and not of the whole option.

1.24 Configuration of DHCP :

- DHCP is capable of providing static and dynamic address allocation.

1.24.1 Static Address Allocation :

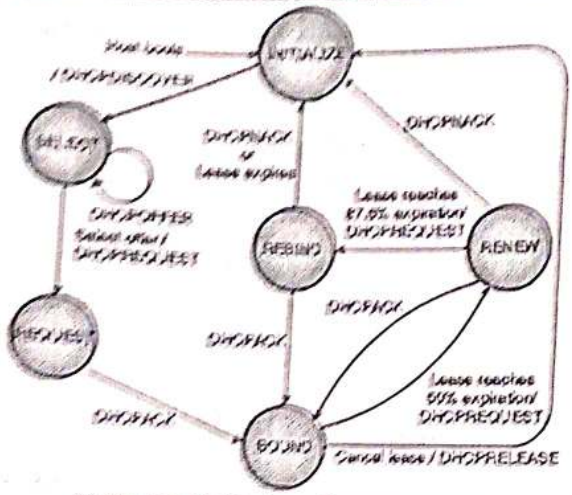
- In the static address allocation capacity, the database of DHCP server would bind the physical addresses to the IP addresses.

1.24.2 Dynamic Address Allocation :

- DHCP has another database in which a pool of available IP addresses is present. This database is used by DHCP when a client requests for a temporary IP address.
- In response to such a request the DHCP allots an unused IP address from the pool to the requesting client for a negotiable period of time.
- The dynamic IP address allocation is required when a host moves from one network to the other or when the host is frequently getting connected or disconnected to the same network.
- The dynamic IP address is assigned temporary for a short duration.
- The DHCP server issues a **lease** for a specific period of time. The client has to renew the lease as soon as it expires or stop using the assigned IP address.

1.24.3 Transition States :

- In the dynamic address allocation mode, the DHCP client acts as a state machine. Depending on the received message it will make transitions from one state to the other.
- The transition diagram with important states has been shown in Fig. 1.24.1.



(C-152) Fig. 1.24.1 : State diagram of DHCP

1.24.3.1 Address Acquisition States :

- When the DHCP is being used to obtain an IP address, a client is in one of six states. The state transition diagram in Fig. 1.24.1 shows events and messages that force a client to change state.
- When a client first boots, it enters the INITIALIZE state. In order to get an IP address, the client first contacts all DHCP servers in the local net.
- To do so, the client broadcasts a DHCPDISCOVER message and enters into the SELECT state.
- Because DHCP is an extension of BOOTP, the client sends the DHCPDISCOVER message in a UDP datagram with the destination port set to the BOOTP port (i.e. port 67). All DHCP servers which are connected to the local net receive the message, and those servers that have been programmed to respond to that particular client send a DHCPOFFER message. Thus, a client may receive zero response or multiple responses.
- When the client is in state SELECT, it collects DHCPOFFER responses from DHCP servers.
- Each offer contains configuration information for the client along with an IP address that the server is offering to lease to the client.
- The client must choose one of the responses (e.g., the first to arrive), and negotiate with the server for a lease. To do so, the client sends a DHCP REQUEST message to the server, and enters the REQUEST state.
- To acknowledge the request has been received and to start the lease, the server responds by sending a DHCPACK.
- When of the acknowledgement is received the client to moves to the BOUND state, where the client proceeds to use the address.
- To use DHCP, a host becomes a client by broadcasting a message to all servers on the local network. The host then collects lease offers from servers, selects one of the offers, and acknowledges the acceptance with the server.

1.24.3.2 Early Lease Termination :

- We think of the BOUND state as the normal state of operation; a client generally remains in the BOUND state when it is using the IP address allotted to it.
- If a client has secondary storage (e.g., a local disk), the client can store the IP address on it and request for the same address when it restarts again.
- In some cases, however, a client in the BOUND state may discover that it does not need an IP address anymore. For example, suppose a user attaches a portable computer to a network, uses DHCP to acquire an IP address and then uses TCP/IP to read electronic mail.
- The user may not know how long reading mail will require, or the portable computer may allow the server to choose a lease period.
- In any case, the minimum lease period specified by DHCP is of one hour. If after obtaining an IP address, the user discovers that no e-mail messages are waiting to be read, the user may choose to shutdown the portable computer and move to another location.
- If the clients does not needs a lease anymore, DHCP allows a client to terminate a lease and does not for the lease to expire.
- Early termination is especially important if only a small number of IP addresses are available at the server as compared to the number of computers that attach to the network. If each client terminates its lease as soon as the IP address is no longer needed, the server will be able to assign the address to another client.
- To terminate a lease early, a client sends a DHCPRELEASE message to the server. once an address is released the client is not allowed to use that address further.
- Thus, after transmitting the release message, the client must not send any other datagrams that use the address.

- As seen in the state transition diagram of Fig. 1.24.1 a host that sends a DHCPRELEASE leaves the BOUND state, and must start at the INITIALIZE a host state again before it can use IP.

1.24.3.3 Lease Renewal States :

- We have seen that when a client acquires an address, it moves to the BOUND state. After entering the BOUND state, the client sets three timers that control lease renewal, rebinding and expiration.
- A DHCP server can specify precise values for the timers when it allocates an address to the client; if the server does not specify timer values, the client uses the default values.
- The default value for the first timer is one-half of the total lease time. When the first timer expires, the client must attempt to renew its lease. To request a renewal, the client sends a DHCPREQUEST message to the server from which it had obtained the lease.
- The client then moves to the RENEW state and waits for a response. The DHCPREQUEST contains the IP address which the client is currently using, and asks the server to extend the lease time to use the same address.
- Similar to the initial lease negotiation, a client can request its preferred period for the extension, but the actual lease time allotment is controlled entirely by the server.
- A server can respond to a client's renewal request in one of two ways: it can instruct the client to stop using the address or it can allow the client to continued use.
- If it allows the client to continue then, the server sends a DHCPACK, which causes the client to return to the BOUND state and continue using the same IP address.
- The DHCPACK can also contain new values for the client's timers. If a server does not allow the client to continue using the same address then, the server sends a DHCPNACK (negative acknowledgement), which causes the client to stop using the address immediately and return to the INITIALIZE state.
- After sending a DHCPREQUEST message that requests an extension on its lease, a client remains in state RENEW and waits for a response from the server.
- If it does not receive any response then the server that granted the lease is either down or unreachable. To handle this situation, DHCP relies on a second timer, which was set when the client entered the BOUND state.
- The second timer expires after 87.5% of the lease period, and makes the client to move from state RENEW to state REBIND. When making the transition, the client assumes the old DHCP server is not available anymore and starts broadcasting a DHCPREQUEST message to any server on the local net.
- Any server configured to provide service to the client can respond positively (i.e. to extend the lease), or negatively (i.e. to deny further use of the same IP address). If it receives a positive response, the client returns to the BOUND state, and resets the two timers.
- If it receives a negative response, the client must move to the INITIALIZE state, must immediately stop using the IP address, and must acquire a new IP address before it can continue to use IP.
- After moving to the REBIND state, a client should have asked the original server and all servers on the local net for a lease extension. Sometimes a client does not receive any response from any server before its third timer, expires, the lease expires.
- The client must stop using the IP address, must move back to the INITIALIZE state, and begin acquiring a new address.

Review Questions

Q. 1 Explain in brief about the application layer.

Q. 2 Write a short note on providing services.

Q. 3 Explain about the standard and nonstandard protocols at the application layer.

Q. 4 Explain in brief client-server paradigm.

Q. 5 State the problems and applications of client-server paradigm.

Q. 6 Explain the P2P paradigm.

Q. 7 State the merits, demerits and applications of P2P paradigm.

Q. 8 Explain the term API and state its types.

Q. 9 Define a socket and state its role.

Q. 10 Draw and explain the structure of www.

Q. 11 Explain the non-persistent and persistent connections in HTTP.

Q. 12 Write a note on : HTTP messages.

Q. 13 What is FTP ? Explain the communication in FTP.

Q. 14 Write a note on E-mail.

Q. 15 Compare SMTP and HTTP.

Q. 16 Write a note on message access agents.