Module VI

Transport layer & Application layer.

The transport layer is responsible for process-to-process delivery of the entire message. A process is an application program running on a host.

Transport layer ensures that the whole message arrives and provide error control and flow control at the source to destination.

* Transport layer is responsible for the delivery of a message from one process to another

Transport layer provides service to the application layer and takes service from Network layer.

Note:- Node to Node delivery – Data Link layer.
host to host delivery – Network layer
process to process delivery – Transport layer

A transport layer protocol can be either connectionless or connection-oriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine.

A connection oriented transport layer make a connection with transport layer at the destination machine first before delivering the packets. After all the data is transferred, the connection is terminated.

* The transport level protocol will require an additional address, known as port Number, to select a particular process among multiple processes running on the destination host.

At the transport level, communication can take place between process or application programs by using port address.

* Transport layer address is specified with the help of a 16 bit port number in the range of '0' to 65535

Internet Assigned Number Authority (IANA) has divided the address in three Ranges. 1) well-known ports.
2) Registered ports. 3) Dynamic ports.

* Well-known ports :- The ports in range from '0' to '1023' are assigned and controlled by IANA. These port numbers are commonly used as universal port number in the server for the convenience of many clients.

* Registered ports :

Registered ports in the range from 1024 to 49151 are not assigned or controlled by IANA. However, they can only be registered with IANA to avoid duplication.

* Dynamic ports :- (Ephemeral port numbers)

Dynamic ports (49152 – 65535) are neither controlled by IANA nor need to be registered. They can be defined at the client site and chosen randomly by the transport layer software.

Well-known ports (UDP) Examples.

Port		protocol
7	–	Echo.
9	–	Discard.
11	–	user
13	–	~~Dynamic~~ Daytime
53	–	DNS.
67	–	BOOTP server.
68	–	BOOTP client
69	–	TFTP
111	–	RPC.
123	–	NTP
161	–	SNMP

Note : Socket Address

Process-to-process delivery need two identifiers, IP address and port number at each end to make connection.

The combination of an IP address and port number is called Socket address. The client socket address defines the client process uniquely and server socket address defines the server process uniquely.

TFTP → Trivial File transfer Protocol

RPC → Remote procedure Ca

NTP → Network Time protoco

SNMP → Simple Network Mangen Protocol

Note :-

In the internet, the transport layer address are called port in ATM networks, they are called AAL-SAPs.

Generic term is Transport Service Access point (TSAP), then Netwo layer address is called NSAPs (Network service Access point). IP addresses are example of NSAPs.

* Duties of transport layer

1) Packetizing.
2) Connection control
3) Addressing.
4) Providing reliablity.

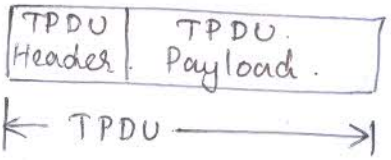
Packetizing :- The transport layer create packet out of the messag received from the application layer. Packetizing is the process of dividing a long message into smaller one. These packets a then encapsulated into the data field of the transport layer packet and headers are added.

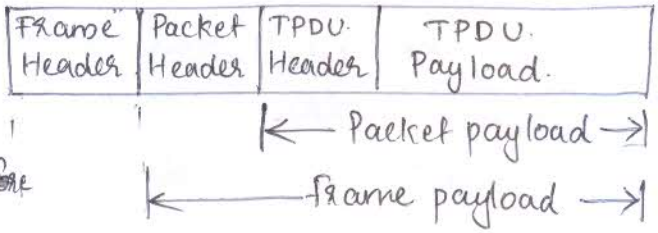
- TPDU (Transport protocol Data unit)

Connection control

Transport layer protocol may be divided into following two categories.

1) connection oriented delivery
2) Connection less delivery.





Addressing

Computer often run several programs at the same time ~~therefore~~ therefore transport layer need to addressing different programs using port number.

Providing reliablity

flow control and error control should be incoporated. Transport layer must provide a reliable packet delivery for process to process.

* Typical QoS parameters for transport layer.

1) Connection establishment delay :- The time difference between the instant at which transport connection is requested and the instant at which it is confirmed is called as connection establishment delay. The shorter the delay the better the service

2) Connection establishment failure probablity :- It is the probabli that connection is not established even after the maximum connec establishment delay. This can be due to network congestion or some other problem.

3) Throughput : It measures the number of bytes of user data transferred per second, measured over some time interva It is measured seperately for each direction.

4) Transit delay :- It is the time for a message being sent by the transport user on the source machine and its being received by the transport user on the destination machine.

5) Residual error ratio :- It means measures the number of lost messages as a function of total message sent. Ideal the value of this ratio should be zero and practically it should be small as possible.

6) Protection :- This parameter provides a way to protect the transmitted data from being read or modified by some unauthorized parties.

7) Priority :- This parameter provides a way for the user to show that some of its connection (or process) are more important than the other one.

8) Resilence :- Due to internal problem or congestion, the transport layer spontaneously terminates a connection. the resilence parameter gives the probability of such a termination

* Transport Service primitives.

The transport service primitives allow the transport user such as application programs to access the transport service. Each transport service has its own service primitives

Primitive	TDPU Sent	Meaning
LISTEN	-(none)-	Block until some process tries to connect
CONNECT	Connection request	Actively attempt to establish a connection
SEND	Data	send data
RECEIVE	(none)	Block until a data TPDU arrives.
DISCONNECT	Disconnection request	Replace the connection

* In order to implement the transport layer services between two two transport entities, ~~the~~ we have to use a transport protocol

-the Transport protocols have to deal with the following tasks. 1) Error control 2) Sequencing 3) Flow control.

Important
* Elements of transport protocols.

1) Addressing - TSAP (Transport Service Access point)
- Socket addressing.

2) Establishing a Connection. - Connection established between Source and destination (Process to process)

3) Releasing a connection – Proper connection release fro both side (close the connection)

4) Flow control & Buffering – for flow control, a sliding window is required on each connection to keep a fast transmitter from overrunning a slow receiv
– the sender should buffer outgoing TPUDs until they are acknowledged
– Receiver accept TPUDs only when a free buffer is availa at receiver side.

5) Multiplexing & Demultiplexing – addressing mechanism allows multiplexing & demultiplexing by t transport layer.

6) Crash recovery – The crash can be recovered by re-tran smitting the lost one.

* The Internet transport protocols (TCP & UDP)

The internet has two main protocols in the transpo layer. One of them is connection oriented and other one support connection less service.

TCP (Transmission control protocol) is a connection oriented protocol and UDP (User's datagram protocol) is a connectionless protocol

* TCP (Transmission Control protocol)

TCP provides a connection-oriented, full duplex, reliable streamed delivery service using IP to transport messages between two processes.

Reliablity is ensured by
* Connection Oriented Service.
* Flow control using sliding window protocol
* Error detection using Checksum
* Error control using go-back-N ARQ technique.
* Congestion avoidance algorithms.

For Obtaining the TCP service, it is necessary for both sender and receiver to create end point called sockets. Each socket has a socket number or socket address.

The Socket address of two parts
1) IP address.
2) port Numbers.

In order to obtain TCP service, it is necessary to establish a connection between the sockets on the sending and receiving machines.

Socket call		Meaning
Socket	—	Create a new end point (socket)
BIND	—	Give a local address to a socket
LISTEN	—	Show willingness to accept connection
ACCEPT	—	Block the caller until a connection attempt arrives.
CONNECT	—	Attempt to make a connection.
SEND	—	Send the data over the connection.
RECEIVE	—	Receiver data over the connection.
CLOSE	—	Release the connection.

* the same Socket can be used for establishing more than one connection at a time. connections are identified by the socket identifiers at both ends.
* TCP does not support multicasting or broadcasting.
* TCP connection is a byte stream and not a message stream.
* when a application passes data to TCP, the TCP may send it immediately or may collect the data for some time and send at once. (which is called buffering)
* If an application wants the data to be sent immediately, the it can use PUSH flag which will force the TCP to send data without any delay.
* If the sending application puts some control information in the data stream and gives it to TCP along with the URGENT flag then the TCP will stop accumulating data and transmit everything it has for that connection immediately.
ie, URGENT flag is always indicate the urgent data.

TCP services

Services offered by TCP to the processes at the application layer

1) process-to-process communication.
- communication using Port numbers.

2) Stream Delivery Service.

TCP allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.

3) Sending & Receiving Buffer.

The sending and receiving process may not read data at the same speed. TCP needs buffers for storage there are two buffers, the sending buffer and the receiving buffer. One for each direction.

4) Segments.

The Network layer as a service provider for TCP needs to send data in packets. not as stream of bytes. At the transport layer, TCP groups a number of bytes together into a packets called a segments, TCP adds a header to each segment *the segments are encapsulated in IP datagrams.

5) Full duplex Communication

data can flow in both directions at the same time, Each TCP then has a sending and receiving buffe

6) Connection-Oriented Service.

a) the two TCPs establish a connection between them
b) Data are exchanged in both direction.
c) The connection is terminated.

7) Reliable Service

It use an acknowledgment mechanism to check the safe and sound arrival of data.

TCP Features.

1) Numbering System
   - Byte number
   - Sequence number.
   - Acknowledgment Number.

Byte number : TCP number all data bytes that are transmitted in the numbering starts with a randomly generated number bytes are numbered from 1057 to 7056

Sequence number:- After the bytes have been numbered, TCP assign a sequence number to each segment that have being sent. The sequence number for each segment is the number of the first byte carried in the segment.

Acknowledgement Number : The value of the acknowledgement field in a segment defines the number of the next byte a receiver expects to receive,
The acknowledgement number is cumulative, means that if a receiver uses 5643 as an acknowledgment number, it has received all bytes from the beginning up to 5642.

2) Flow control
3) Error control
4) Congestion control

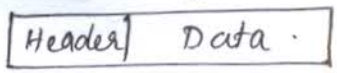
* TCP header format

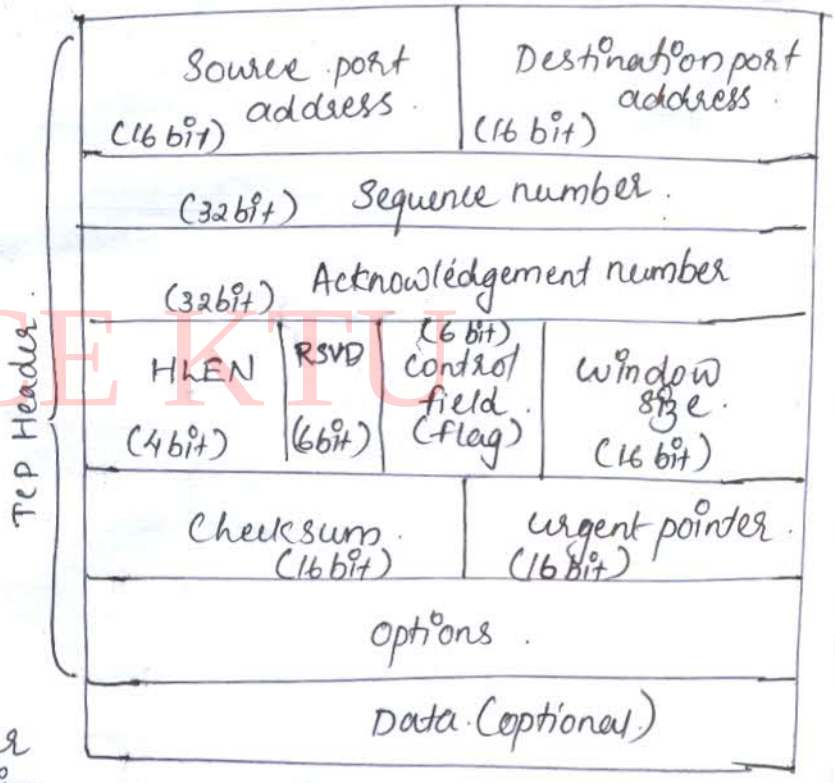
- The TCP segment consist of a 20-60 byte header.
- The header is 20 bytes if there is no options and up to 60 bytes if it contain options.

Source port address
- 16 bit field.
- It defines the port number of the application program in the host of the sender.

Destination port address
- 16 bit field.
- It defines the port number of the application program in the host of the receiver.

-TCP Segment





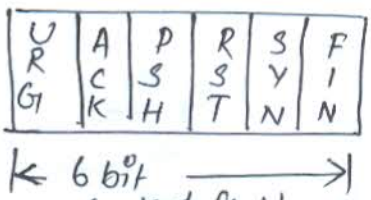
Sequence number
- 32 bit field, defines the number assigned to the first byte of data contained in this segment.
- During the connection establishment, each party uses a random ~~number~~ number generator to create an Initial sequence number (ISN), which is usually different in each direction.

Acknowledgment number.
- 32 bit field, defines the byte number that the receiver of the segment is expecting to receive from the other party.

Header length (HLEN)
- 4 bit field, indicate the total header length (length between 20-60byte) ie, value of field can be between 5 (5x4=20) and 15 (15x4=60).

Reserved (RSVD)
- 6 bit field reserved for future use.

Control (Flag)
- 6 diffrent control bits or flags, one or more of these bits can be set at a time.



Flag	Description
URG	– The value of urgent pointer field is valid.
ACK	– The value of the acknowledgment field is valid.
PSH	– push the data
RST	– Reset the connection
SYN	– Synchronize sequence numbers during connection
FIN	– Terminate the connection.

Window size
- 16 bit field.
- define the size of window, in bytes that the other party must maintain.
- maximum size of the window is 65,535 bytes.
- normaly the value referred to as the receiving window.

Checksum.
- 16 bit field, contains the checksum of Header (error detection)

Urgent pointer.
- 16 bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data.

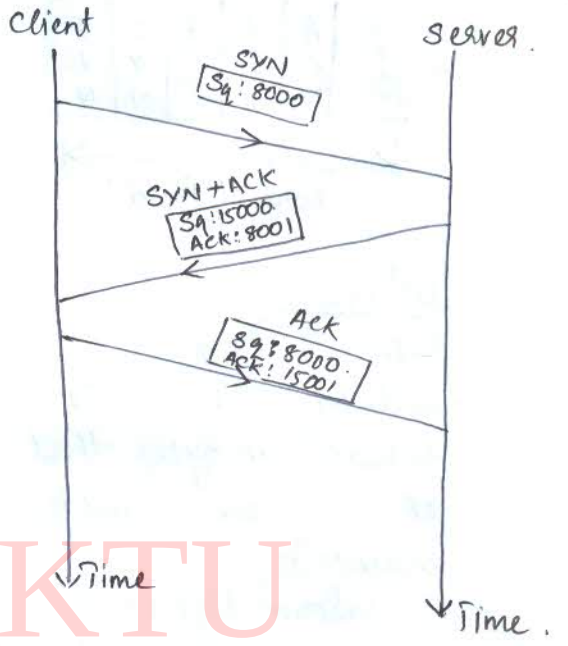
Options.
- Optional 40 bytes of informations.

* TCP connection
TCP is connection-Oriented, A TCP protocol establishes a virtual path between the source and destination. All the segment belonging to a message are then sent over this virtual path.
- Acknowledgment & re-transmission of damaged or lost frame.
- if a segments arrives out of order, the TCP holds it until the missing segment arrive.

TCP connection-oriented transmission require three phases.
1) Connection establishment
2) data transfer.
3) Connection termination.

Connection establishment
- TCP transmit data in full duplex mode.
- Each party must initialize communication and get approval from the other party before any data are transferred.

* The connection establishment in TCP is called Three-way Handshaking



Step① The client sends the first segment, a 'SYN' segment, in which only the SYN flag is set. This segment is for synchronization of sequence number, it consume one sequence number. When data transfer starts, the sequence number is incremented by 1
- A SYN segment cannot carry data, but it consume one sequence number.

Step② The server sends the second segment, a SYN+ACK segment, with 2 flag bit set. SYN & ACK. This segment has a dual purpose.
- SYN for communication in the other direction.
- Ack for the acknowledgment of client SYN segment.
- A SYN+ACK segment cannot carry data, but does consume one sequence number of server and acknowledgement number.

Step③ The client sent the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment (or first segment sent by server)
- An ACK segment, if carry no data, consume no sequence number.

* This 3 steps show how TCP connection establishment ~~takeplace~~ occure.

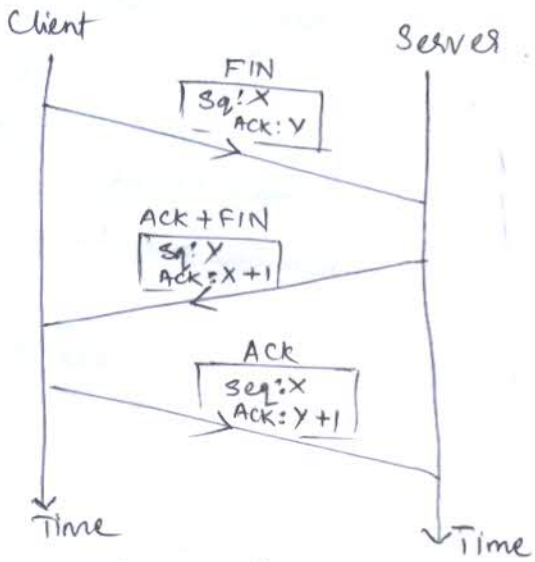
Data transfer
After connection establishment, bidirectional data transfer can take place the client and server can send data and acknowledgments.
- The data segments sent by the client have the PSH (Push) flag set so that the server TCP knows to deliver data to the server process as soon as they received.

Connection Termination

Any of the two parties involved in exchanging data can close the connection. (usually initiated by the client)

* Two options for connection termination.
  1) Three-way Handshaking
  2) Four-way Handshaking with a half-close option.

Three-way Handshaking



Step(1) : The client TCP, after receiving a close command from the client process, send FIN segment in which the FIN flag set

- FIN segment consumes one sequence number if it does not carry data.

Step(2) The server TCP, after receiving the FIN segment, inform the process and sends the second segment FIN + ACK segment to confirm the receipt of FIN segment and at the same time to announce the closing of the connection in other direction.

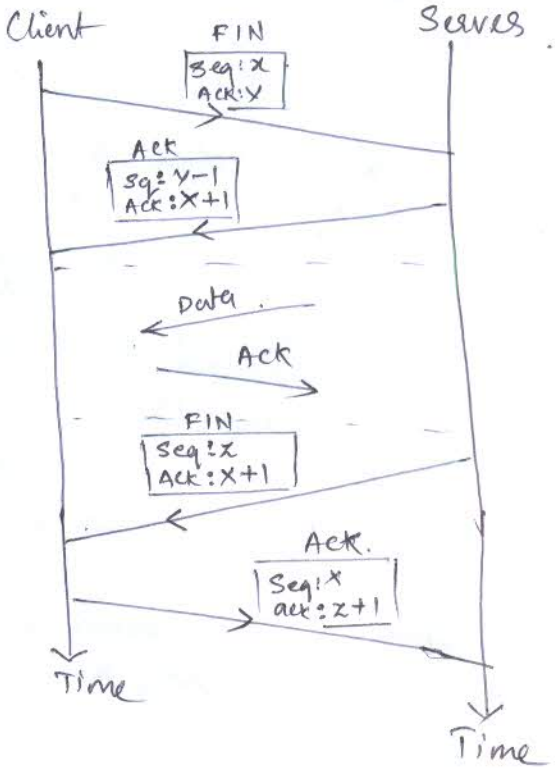
- The FIN + ACK segment consumes one sequence number if it does not carry data.

Step(3) The client TCP sends the last segment, an Ack segment to confirm the receipt of FIN segment from the TCP server. This segment contains the acknowledgement number, which is 1 plus the sequence number received in the FIN segment from the server.

* Half close

- In TCP, one end can stop sending data while still receiving data.
- when a client sends data to the server to be ~~stored~~ sorted, the server to receive all the data before sorting can start. Client, after sending all data, can close the connection in the outbound direction. However, the inbound ~~data still needs time for sorting~~ direction must remain open to receive the sorted data.
- The client half-closes the connection by sending FIN segment. The server accepts the half close by sending ACK segment. The data transfer from the client to server stops. The server can still send data. when the server sent all the processed data. It send a FIN segment and acknowledged by client.

(Half Close)



Flow control

A sliding window is used to make transmission more efficient as well as to control the flow of data, so that the destination does not become overwhelmed with data.
- TCP sliding window are byte-oriented.

* The size of the window is the lesser of receiver window (rwnd) and congestion window (cwnd)

window size = rwnd or cwnd (small value)

* Error detection and correction in TCP is achieved through the use of three simple tools
  1) checksum
  2) Acknowledgment
  3) Time-out

* Each segment includes a checksum field which is used to check for a corrupted segment. If the segment is corrupted, it is discarded by the destination TCP and considered as lost.
- TCP use a 16-bit checksum that is mandatory to every segment

* Acknowledgment: TCP use acknowledgement to confirm the receipt of the data segment
Ack segment do not consume sequence number and are not acknowledged

* Retransmission: a retransmission occurs if the retransmission timer expires or three duplicate Ack segments have arrived.
- No retransmission timer is set for an Ack segment.

* Data may arrive out of order and be temporarily stored by the receiving TCP, but TCP guarantees that no out-of-order segment is delivered to the process.

The state used in TCP connection

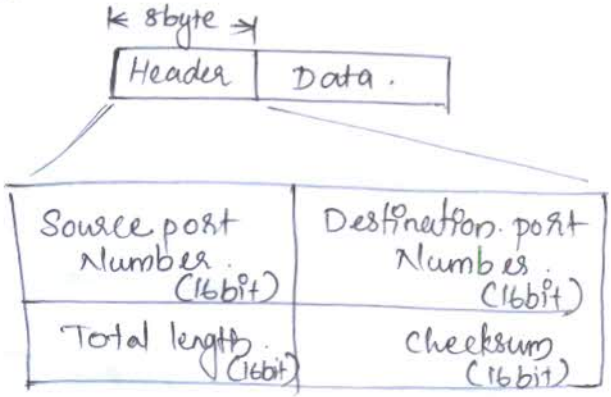
1) CLOSED
2) LISTEN
3) SYN RCVD
4) SYN SENT
5) ESTABLISHED
6) FIN WAIT 1
7) FIN WAIT 2
8) TIMED WAIT
9) CLOSING
10) CLOSE WAIT
11) LAST ACK

USER DATAGRAM PROTOCOL (UDP)

* The user datagram protocol is called a connectionless, unreliable transport protocol. It does not add anything to the service of IP except to provide process-to-process communication instead of host-to host communication.
* It performs very limited error checking.
* UDP is very simple protocol using minimum of overhead.
* does not care much about reliability.

User Datagram

* UDP packets, called user datagrams.
* have fixed-size header of 8 bytes.



Source port number (16 bit)
- Indicate the port of sending process.

Destination port (16 bit)
the port number used by the process running on the destination host

Total length (16 bit)
total length of the user datagram (data + Header)

UDP Length = IP length − IP header's length

Checksum (16 bit)
used to detect errors over the entire user datagram (header + data)

UDP operation.

* Connectionless Services.

UDP provides connectionless services, This means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagram even if they are coming from the same source process and going to same process.

* user datagrams are not numbered.
* there is no connection establishment and no connection termination.
* each user datagram can travel on a different path.

Flow control and Error control

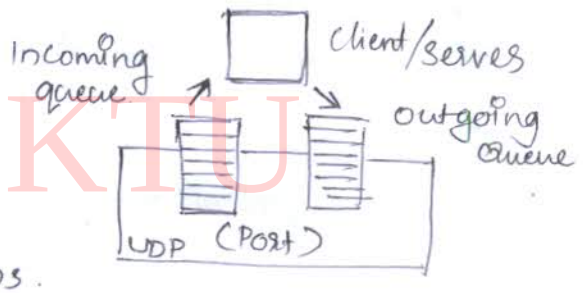
* UDP is a very simple, unreliable transport protocol. There is no flow control and hence no window mechanism.
* The receiver may overflow with incoming messages.
* There is no error control mechanism in UDP except for the checksum. This means that the sender does not know if a messages has been lost or duplicated.
* When the receiver detect an error through checksum the user datagram is silently discarded.

The lack of flow control and error control means that the process using ~~UDP~~ UDP should provide.

1) Encapsulation and Decapsulation.
2) Queuing.

Encapsulation and Decapsulation :- To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages in an IP datagram.

Queuing :- Queues are associated with ports.



At the client site, when a process starts, it request a port number from the operating system. Some implementations create both an incoming and outgoing queue associated with each process.

The client process can send messages to outgoing queue by using the source port number specified in the request. UDP removes the message one by one and, after adding the UDP header, delivers them to IP

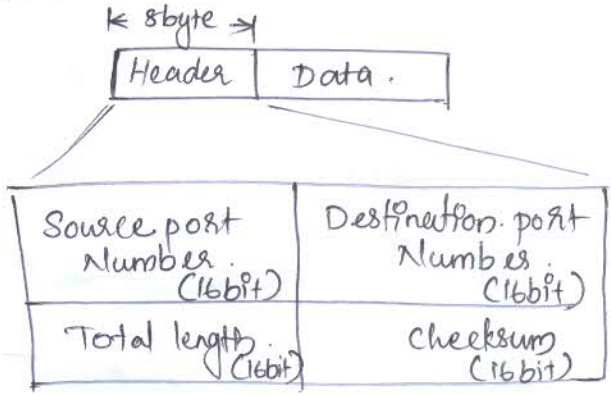
When a message arrives for client, UDP checks to see if an incoming queue has been created for the port number specified in the destination port number field of the user datagram. If there is such a queue. UDP sends the received user datagram to the end of the queue. If there is no such queue UDP discard the user datagram and ask ICMP protocol to send a port unreachable message to server.

USER DATAGRAM PROTOCOL (UDP)

* The user datagram protocol is called a connectionless, unreliable transport protocol. It does not add anything to the service of IP except to provide process-to-process communication instead of host-to host communication.
* It performs very limited error checking.
* UDP is very simple protocol using minimum of overhead.
* does not care much about reliability.

User Datagram.

* UDP packets, called user datagrams.
* have fixed-size header of 8 bytes.



Source port number (16 bit)
- Indicate the port of sending process.

Destination port (16 bit)
the port number used by the process running on the destination host

Total length (16 bit)
total length of the user datagram (data + Header)

UDP length = IP length − IP header's length

Checksum (16bit)
used to detect errors over the entire user datagram. (header + data)

UDP operation.

* **Connectionless Services.**
UDP provides connectionless services, This means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagram even if they are coming from the same source process and going to same process.
* user datagrams are not numbered.
* there is no connection establishment and no connection termination.
* each user datagram can travel on a different path.

Use of UDP

* UDP is suitable for a process that requires simple request-response communication with little concern for flow control and error control.
  - It is not usually used for a process such as FTP that need to send bulk data.
* UDP is suitable for a process with internal flow and error control Mechanisms.
  Eg:- Trivial file Transfer protocol (TFTP)
* UDP is suitable transport protocol for multicasting, Multicasting capability is embedded in the UDP software but not in the Tcp software.
* UDP is used for ~~Management~~ management processes such as SNMP
* UDP is used for some route updating protocols such as Routing Information protocol (RIP)

* Application layer.

The application layer enables the user, whether human or software, to access the network.

* The application layer is responsible for providing service to the user.
* The application layer receives services from transport layer.
* The application layer programs are based upon the concept of client and server
* For communication between client & server, addressing is needed. when a client request a service from the server, it has to include the server address as destination address and its own address as source address. when a server responds, it reverses the address.
* Important application.
  1) Electronic mail
  2) World wide web.
  3) Multimedia
  4) Remote file transfer and access.

The most common service provided is SMTP or electronic mail. It allows the user to send a message to another user in Internet

file transfer :- user can transfer a file from its compter to the server or transfer a file from a server to its computer.

—This application is called FTP.

* The client/server programs can be divided ~~into two~~ into two categories.

1) those that can be directly used by the user
eg: email.

2) those that support other application programs.

- DNS (Domain name system) is a supporting program that is used by other program such as email.

File Transfer protocol (FTP)

Transfering file from one computer to another is one of the most common task expected from a networking or internetworking

* Popular protocol involved in transferring file is, File transfer protocol (FTP)

* FTP is the standard mechanism provided by TCP/IP for copying a file from one host to another.

- Some of the problems in transferring files from one system to the other are,

1) Two systems may use diffrent file name conventions
2) Two system may represent text and data in diffrent type
3) directory structures of the two systems may be different

FTP provides a simple solutions to all problems.

* FTP differs from other client/server application in that it establishes two connections between the host (make more efficient)

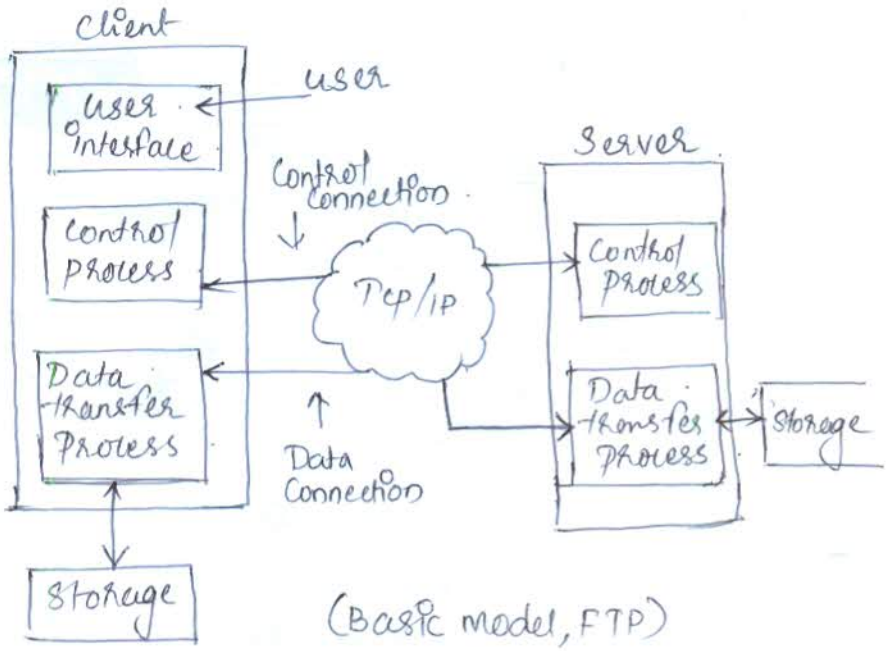
1) one connection is used for data transfer.
2) next is for control informations (Commands & Responses)

- The control connection use very simple rule of communication.
  - transfer only one line of command or response at a time.
- Data connection uses more complex rules due to the variety of data types being transferred, complexity is at the FTP level. not TCP, For TCP both connections are treated the same.

* FTP uses the service of TCP. It need two TCP connections. The well-known port 21 is used for the control connection and the well-known port 20 for the data connection.

* Client has three components.
  1) User Interface
  2) Control process
  3) Data transfer process.

* Server has two components.
  1) Server control process
  2) Server Data transfer process.



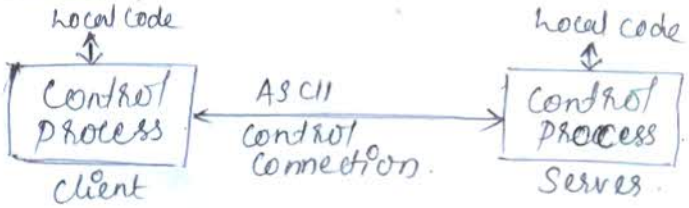
(Basic model, FTP)

- The control connection is made between the control processes. The data connection is made between the data transfer processes.
- The control connection remain connected during the entire interactive FTP session. The data connection is opened and the closed for each file transferred.

Note: when a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times. If several files are transferred.

Communication over control connection

FTP uses a set of ASCII character to communicate across the control connection. Communication is achieved through commands and response.

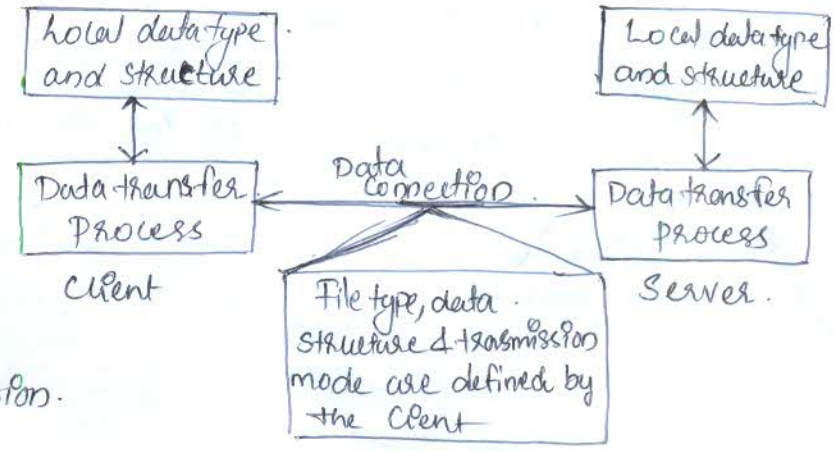


- One command is sent at a time. Each command or response is only of one short line. Therefore, it is not necessary to think about file structure.
- Each line is terminated with two characters end of line tokens.

Communication over Data Connection

The purpose of implementing a data connection is to transfer file, For this client has to define the following
1) Type of file being transferred.
2) Structure of data.
3) Transmission mode.

* File transfer occurs over the data connection under the control of the commands sent over the control connection.



* File transfer in FTP means.

1) A file is to copied from the server to the client. This is called retrieving a file. It is done under the supervision of the RETR command

2) A file is to be copied from the client to the server. This is called storing file, it is done under the supervision of the STOR command

3) A list of directory or file name is to be sent from the server to client. This is done under the supervision of the LIST command (FTP treats a list of directory or file name as a file)

* The client must define the type of file to be transferred, the structure of the data, and the transmission mode. Before sending the file through the data connection, Prepare a transmission through the control connection (RETR, STOR, LIST etc.)

* File types
  1) ASCII file
  2) EBCDIC file
  3) Image file.

ASCII file is the default format for transferring text file, EBCDIC file can be transferred if both end use EBCDIC encoding. Image file is the default format for transferring binary files. The file is sent as continuous stream of bits without any interpretation or encoding.

Note:- In ASCII file, Each character is encoded using 7-bit ASCII. The sender transforms the file into ASCII character and the receiver transform the ASCII characters to its own representation.

* Data Structure
  1) File structure (default)
  2) Record structure.
  3) Page structure.

File structure has no structure. It is simply a continuous stream of bytes. In the record structure, the file is divided into records. This can be used only with text files. In the page structure, the file divided into pages, which can be stored randomly or sequentially.

* Transmission mode
  1) Stream mode
  2) Block mode
  3) Compressed mode

Stream 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.

Block mode :- data can be delivered from FTP to TCP in blocks. Each block is preceded by a 3 byte header

Compressed mode :- the data can be compressed, Generally, a run length encoding is used for compression.

* File Transfer
  1) Retreving a file. (RETR command)
  2) Storing of a file (STOR command)
  3) Retreving a list (LIST command)

* FTP Commands

	Commands	Meaning
FTP Commands to transfer files	Get	– copy a file from remote host to local host
	M. Get	– copy multiple file from remote host to local host
	put	– copy a file from local host to remote host
	M.put	– copy multiple file from local host to remote host
FTP Commands to Connect to a remote host	open	– Select the remote host and initiate high session.
	user	– Identify the remote user ID
	pass	– Authenticate the user.
	site	– Send the information to the remote host
FTP Commands to terminate session.	Quit	– Disconnect from the remote host & terminate FTP
	close	– Disconnect from the remote host but leave FTP client running.

Note :- Anonymous FTP

To use FTP, a user need an account and a password on the remote server. Some sites have a set of files available for public access, to enable anonymous FTP. To access this files, a user does not need to have an account or password.

user can use Anonymous as username and guest as password.

Domain Name System (DNS)

IP addresses are convenient and compact way for identifying machines and are fundamental in TC/IP. It is unsuitable for human user. Meaningful hig-level symbolic names are more convenient for humans. Application software permit users to use symbolic names, but the underlaying network protocol require IP addresses.

ie, application layer need a address, which is high-level symbolic names (Each program will have its own address format)
(alias name)

* Application layer use names with proper syntax with efficient translation mechanism.
  - Domain name system (DNS) was invented for this purpose.

DNS – address mapped, alias name to IP address.

DNS is not used directly by the user. It is used by another application programs for carrying out the mapping.

DNS working

To map a name onto a 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 local DNS server which look 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.

Name space
- Flat Name space
- Hierarchical Name space

A name space that maps each address to unique name.

* Flat name space :- A name in this space is a sequence of characters without structure. The name may or may not have a common section; if they do, it has no meaning.
  - the main disadvantage of a flat name space is that it cannot be used in a large system such as Internet because it must be centrally controlled to avoid ambiguity and duplicate.

Hierarchical Name space :- name is made of several parts. the fist part can define the nature of the organization,

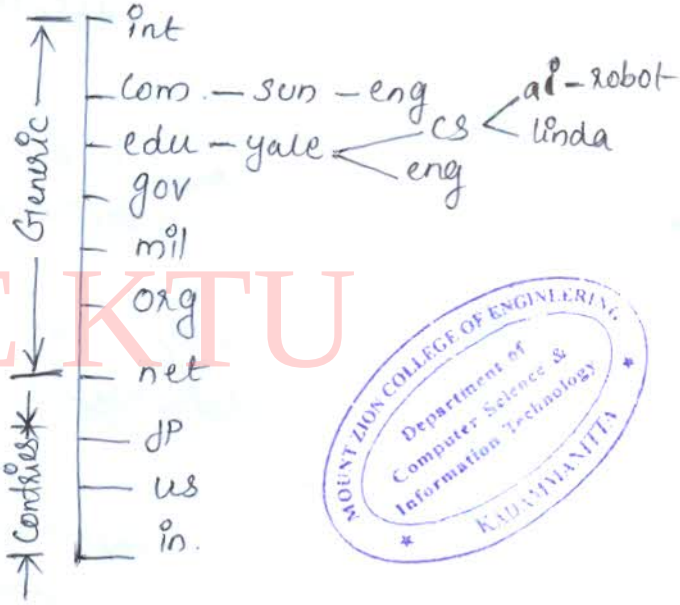
The second part can define the name of organization. the third part can define departments in the organization. and so on.
- the authority to assign and control the name spaces can be decentralized. A central authority can assign the part of the name that define the nature of organization and the name of organization. The responsibility of rest of name can be given to the organization itself.

* DNS Name space

Domain name space was designed from hierarchial name space.

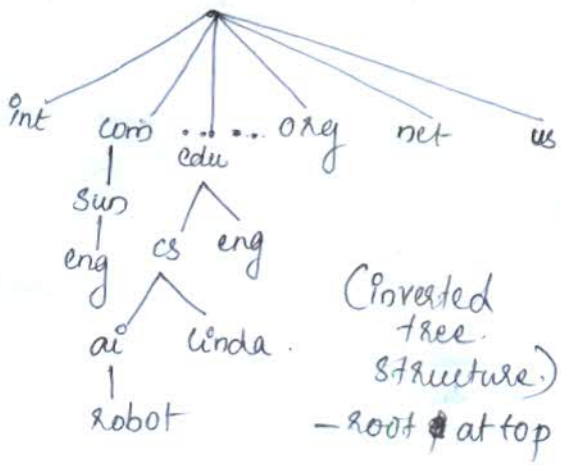
Conceptually, the Internet has been divided into hundreds of top-level domains. Each domain covers many hosts. Each domain is divided into several subdomain and they are further divided and so on.

top-level domain < Generic, Countries



The Generic domain are com (commercial), edu (education), gov (government), int (International) mil (military), net (network provider) and org (non-profit organization).
The ~~Cont~~ country domain include for every ~~Cont~~ country.

* The components are separated by dots Eg:- eng.sun.com. this called hierarchical naming.
* Domain can be represented as tree (inverted) the tree can have only 128 levels, level 0 (root) to level (127)



(Inverted tree structure)
- root at top

Label

Each node in the tree has a label, which is a string with a maximum of 63 characters
The root label is null label (zero string)

DNS requires that branches from the same node have different labels, which guarantees the uniqueness of domain name.

Domain Name : Each node in the tree has a domain name. a full Domain Name is a sequence of labels separated by dots. the domain always read from the node up to the root. the last label is the labe of the root (null).

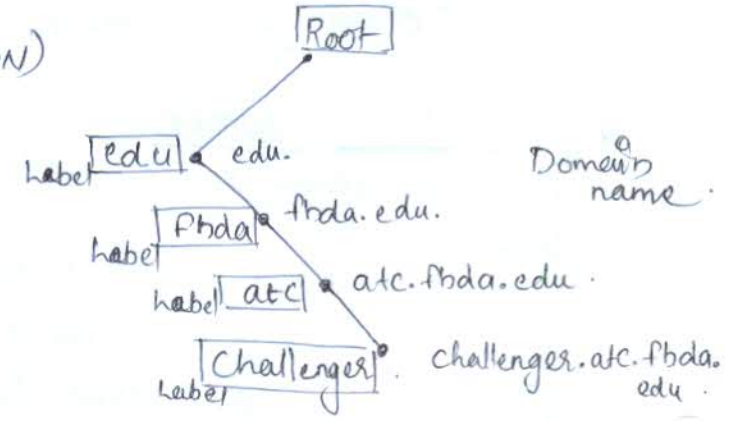
ie, last character is a dot. because the null string is nothing.

Fully Qualified Domain Name (FQDN)

If a label is terminated by a null string, it is called a fully Qualified domain name.

* FQDN is domain name that contains the full name of host. (It contain all labels)

Eg:- Challenger.atc.fbda.edu.



Partially Qualified Domain Name

If a label is not terminated by a null string, it is called a partially Qualified domain name (PQDN). A PQDN start from a node, but does not reach the root.

- Resolver can supply the missing part, called the Suffix to create FQDN

Eg:- user define pustial name., Challenger.

the DNS client adds the Suffix atc.fbda.edu before passing the address to DNS Server.

* The DNS client normally holds list of Suffixes.

Name Servers (Hierarchy of Name Servers)

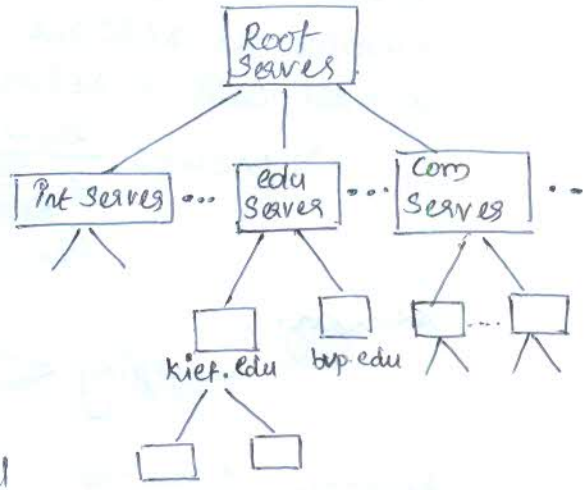
* Name Server consist of DNS database, ie, the various name and their corresponding IP addresses. Theoretically, a single name Server could contain the entire DNS database. But practically to store such a huge information at one place is inefficient and unreliable

- ∴ distribute the information among many computers called DNS servers

In DNS server Hierarchy the whole Name space is divided into may first level domain, the first level domain are further divided into Smaller Subdomains. called Second level domain They can.

further divided and go on.

* The Root Server stand ~~along~~ alone.
* Each Server can be responsible to either large or small domain.
* The whole DNS Server / DNS name space is divided up into non-overlapping zones.
* A Server is responsible for or has authority over is called zone.



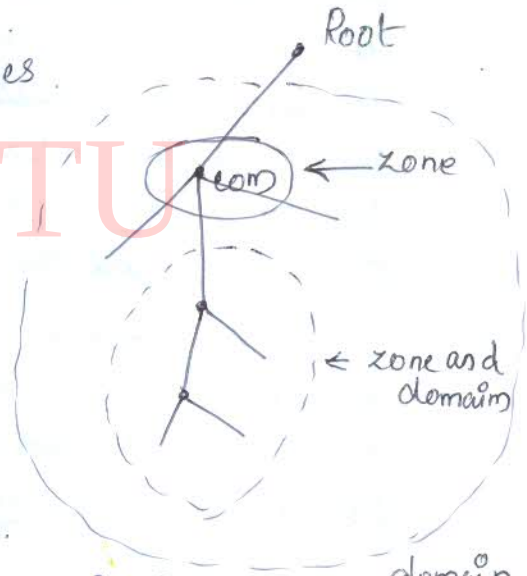
(Hierarchy of Name Server)

- If Server is appointed for a domain as zone and the domain is not further divided into subdomain then the domain and zone will be same.

* the Server make a data base ~~of~~ called a zone file

Root Server :- A root Server is a server whose zone consist of the whole DNS tree. It does not store any information about domains but delegates the authority to other servers. It keep the reference of these servers.

* There are more than 13 root servers and they are distributed all around the world.



Primary Server
It is a server which store a file about its zone. It is authorized to create, maintain and update the zone file. It store the zone file into local disk

Secondary Server.
this server ~~transform~~ transfers complete information about a zone from another server which may be primary or secondary.. the secondary server not authorized to create or update zone file.

* Secondary server load information from the primary servers.

Resolution
The process of mapping a name to an address or an address to a name is called as name address resolution.

Resolver :- DNS is the client server application. A host which wants to map a name to address or vise versa calls a DNS client named as resolver. When the name address mapping is necessary a host calls a resolver.

process → Request → Resolver → Request → Server
process ← Reply (Error-free) ← Resolver ← Reply ← Server

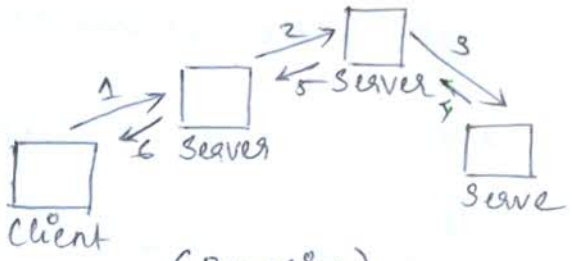
Mapping
- Name to address
- Address to Name

Recursive Resolution

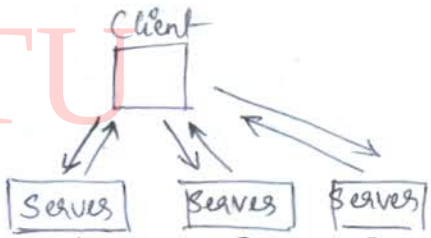
Sometimes, a client (resolver) request is recursive for a final resolution.

- If a server is authorised for a domain name, it checks it database and send reply. But if this server has not authorized it divest the request to another server at wait for response. If the parent is an authority, it respond, other wise. Send the query to another server.
- When the query is solved, the response is return back to requesting client this query is called recursive query and process is called recursive resolution

Resolution
- Recursive Resolution
- Iterative Resolution.



(Recursive)



(Iterative)

Iterative Resolution

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, IP address of the server that it thinks has the answer for the query. The client has to repeat the query to this server. If server has also cannot answer it return the IP address of New server to client.

- this process is called Iterative Resolution.

Client sends the same query to different servers.

ELECTRONIC MAIL

One of the most popular network service is electronic mail (e-mail). Simple Mail protocol (SMTP) is the standard mechanism for electronic mail in the internet.

* The first email system simply consisted by file transfer protocols. But some of the limitations of this system are.

1) Sending a message to a group of people was inconvenient
2) Message did not have any internal structure (Therefore, 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 email to someone else.
5) It was not possible to create and send messages containing a text, drawing, fasmile and voice together.

Therefore, more elaborate e-mail system were proposed, ARPANET e-mail were published as RFC 821 (transmission protocol) and RFC822 (message format). These are used in Internet.

E-mail Architecture and services.

An e-mail system consists of two sub systems.

1) User agents. 2) Message transfer agents.

User agents :- They allow the people to read and send e-mail

Message transfer agents :- They move the message from the source to the destination.

Basic functions :- E-mail systems support five basic systems ~~others~~.

1) composition. 2) Transfer. 3) Reporting
4) Displaying 5) Disposition.

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 field attached to each message.

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 outputting the message, and releasing the connection.

Reporting :- This is to tell the sender about whether the message was delivered or rejected or lost

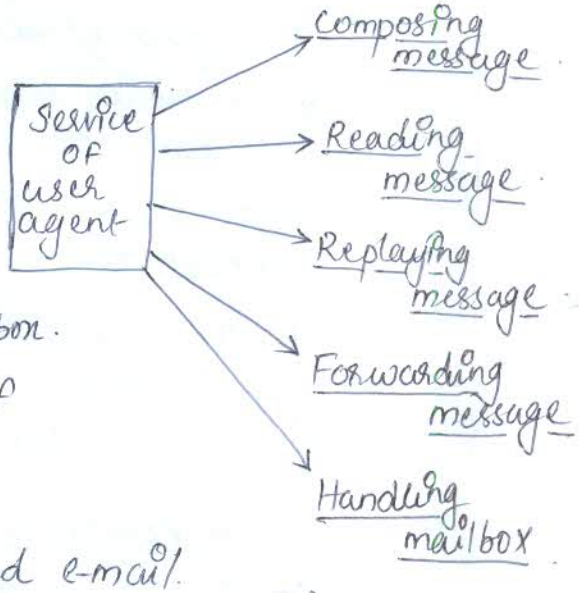
Displaying:- It is the process of displaying the incomming messages. For this purpose, simple conversion and formating are required to be done.

Disposition:- This is concerned with what recipient does with the message after receiving it. Some of the possibilities are.

1) Throw after reading. 2) Throw before reading.
3) Save message 4) Forward message 5) Process message in some other way.

Advanced Features of E-mail System.

1) Forwarding an e-mail to a person away from his computer.
2) Creating and destroying mailbones to store incoming e-mail.
3) Inspecting contents of mailbox, insert and delete message from the mailbox.
4) Sending a message to a large group of people using the idea of mail list
5) To provide registered e-mail.
6) Automatic notification of undelivered e-mail.
7) Carbon copies. 9) Secret (encrypted e-mail)
8) High priority e-mail 10) Alternative recipient.



E-mail Envelope.

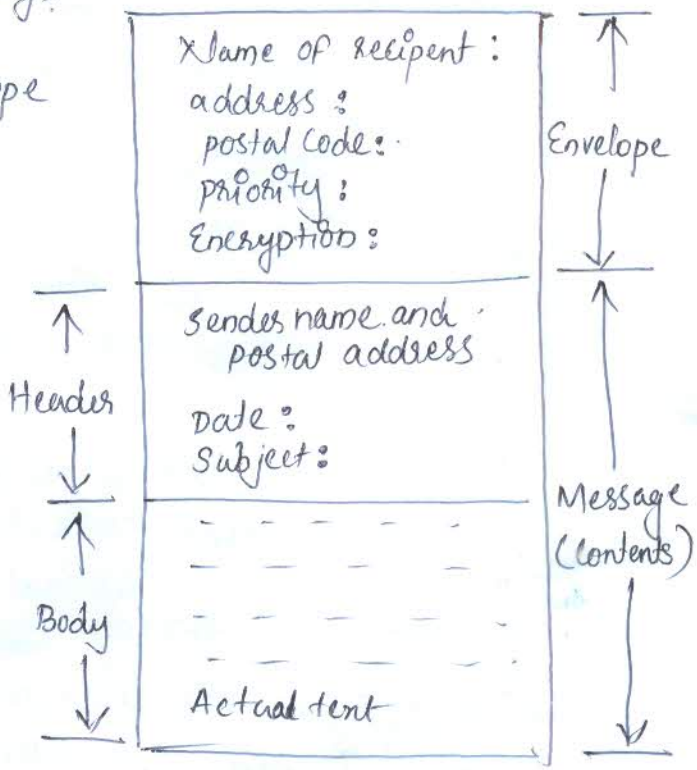
In modern e-mail system, there is distinction made between the e-mail envelope and contents. An e-mail envelope contains the message, destination address, priority, security level etc. The message transport agents use this envelope for routing.

Message.

The actual message inside the envelope is made of two parts.

1) Header 2) Body.

Header ~~contai~~ carries the control information while body contains the message contents.



Message format – RFC 822.

Messages consist of a primitive envelope, some number of header field, a blank line and then the message body.

Header field logically consists of a single line of ASCII text which contains the field name, a colon and a field.

(RFC 822)

Header Name	Meaning
TO:	– E-mail address of primary recipients.
CC:	– E-mail address of secondary recipients.
BCC:	– E-mail address for blind Carbon copies.
From:	– Person who create the message.
Sender:	– E-mail address of the actual Sender.
Received:	– Line address by each transfer agent along the route.
Return-path:	– Can be used to identify the path back to the Sender.
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
Reference:	– Other relevant message identification numbers.
Keywords:	– Keywords chosen by user.
Subject:	– Summary of the message for the one line display.

* Normally, the user agent builds a message and passes it to the message transfer agent which uses some header field for construct of envelope.

Message body

The message body comes after the header. The user can put whatever they want to send, in the message body. It is possible to terminate the message with ASCII cartoons, quotations etc.

MULTIPURPOSE INTERNET MAIL EXTENSIONS (MIME)

RS822 email used to consist of only the text messages in English and expressed in ASCII. But in the world wide Internet environment, this approach is not adequate.

* Some problems are encountered in sending and receiving the following types of messages.
  1) Messages in the languages having accents such as French or Germans.
  2) Messages which do not contain eg: audio and video.
  3) Message in the languages which do not have alphabets. (eg: Chinese and japanese)
  4) Message is non latin alphabets such as Russian or Hebrew

The solution to these problem was MIME, (Multipurpose Internet Mail Extension) it was proposed in RFC 1341 and then updated in RFC 1521

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 message.

* MIME messages can be sent using the existing mail programs and protocols. The user can change sending and receiving programs themselves.

New-Message Headers.

Five new message Headers are defined for MIME.

1) MIME-version 2) Content-Type 3) Content-Transfer-Encoding
4) Content-Id 5) Content-Description

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

Content-Type :- It is used to specify the type of message body. RFC 1521 defines seven types with each one having one or more subtypes are separated by a slash.

Eg:- Video/mpeg

* The subtype must be given in the header.

Type	subtype	Description
Text	Plain	Text with unformatted Text
	Enriched	Text including simple formating Commands.
Image	Gif	Still pictures in GIF format
	Jpeg	Still picture in Jpeg format
Audio	Basic	Audible sound.
Video	MPeg	Movie in MPEG format
Application	Octet-stream	Byte sequence in unInterpreted form.
	post script	A printable document in post script.
Message	RFC822	A MIME RFC 822 message.
	partial	Split message for transmission.
	External body	Message itself should be fetched over the net.
Multipart	Mined	Independent part in specified Order.
	Alternative	Same message in different formats.
	Parallel	parts must be viewed simultaneously.
	Digist	Each part is complete RFC 822 message.

Content-Transfer-Encoding:- This header defines the method used to encode the messages into 0s and 1s for transport.

	Type	Description
1)	7-bit	NVT ASCII character and short line.
2)	8-bit	non ASCII character and short line. Lines.
3)	Binary	non ASCII character with unlimited length.
4)	Base-64	6-bit blocks of data encoded into 8-bit ASCII character.
5)	Quoted-printable	non ASCII chereacters encoded as an equal sign followed by an ASCII code.

Content-ID :- This field identifies the contents. This header uniquely identifies the whole message in a multiple-message environment. Its format is same as the format of standard Message-ID header.

Content-Description :- This field tells the message is. It is in the form of ASCII string. This header is required because the recipient will know whether it is worth decoding and reading message.

(This header define whether the body is image, audio or video also)

SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)

Simple network Management protocol (SNMP) may be defined as the framework for managing devices in Internet using the TCP/IP protocol suite.

* It provides a set of fundamental operations for monitoring and managing an Internet.
* SNMP uses the concept of manger and agent. a manager usually a host, control and monitors a set of agents, usually routers.
* SNMP is an application-level protocol in which a few manager stations control a set of agents.

* SNMP frees management task from both physical characteristics of the managed device and the underlying ~~ted~~ networking technology.

Concept of ~~Manag~~ Managers and Agents

A management station, called a manager, is a host that run the SNMP client program. A managed station, called Agent is a router (or a host) that run the SNMP server program Mangement is achieved through Simple interaction between manager and Agent.

* The agent keeps performance information in a database. The manager has access to the values in the database.

Eg:- The ~~mang~~ manager can fetch and compare the numbes of packets received and forwarded to see if the router is congested or not.

* The server program running on the agent can check the environment and, if it notices something unusual, it can sent a warning message (called a trap) to manager.

- Management with SNMP is based on three basic ideas.

1) A manager checks an agent by requesting information that reflects the behavior of the agent

2) A manager forces an agent to perform a task by resetting values in the agent database.

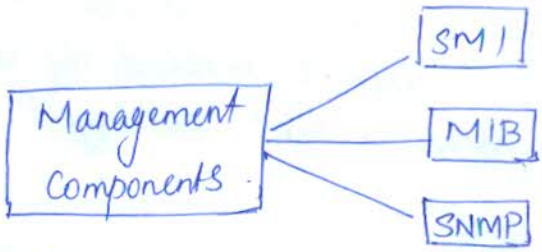
3) An agent contributes to the management process by warning the manger of an unusual situation.

Internet Management Components.

Management in the internet is achieved not only through the SNMP Protocol but also by using other protocols that cooperate with SNMP.

* At top-level, management is accomplished with two other protocols.
  1) structure of Management information (SMI)
  2) Management information base (MIB)

* SNMP uses the services provided by the SMI & MIB protocols to do its job.
* SNMP, MIB & SMI use other protocols such as abstract syntax notation1 (ASN1) and basic encoding rules (BER)



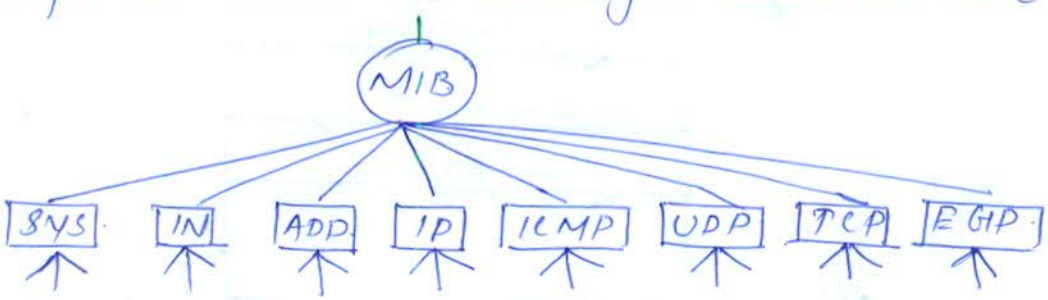
SMI :- The SMI is a component used in network management. Its functions are to name objects; to define the type of data that can be stored in an object, and to show how to encode data for transmission over the network.

MIB :- The management information base (MIB) is the second component used in network management. Each agent has its own MIB, which is a collection of all objects that the manager can manage

The objects in the MIB are categorized under eight different groups

1) System 2) interface 3) address translation 4) IP
5) ICMP 6) UDP 7) TCP 8) EGP

These groups are under the MIB object identifier tree.



SNMP : SNMP defines five messages.

1) Get Request 2) Get Next Request 3) Set Request
4) Get Response. 5) Trap.

Get Request :- The GetRequest message is sent from the manager (Client) to agent (server) to retrieve the value of a variable.

GetNext Request : The GetNext Request message is sent from the manager to agent to retrieve the value of varible. The retrieved value is the value of the object following the defined object in the message.

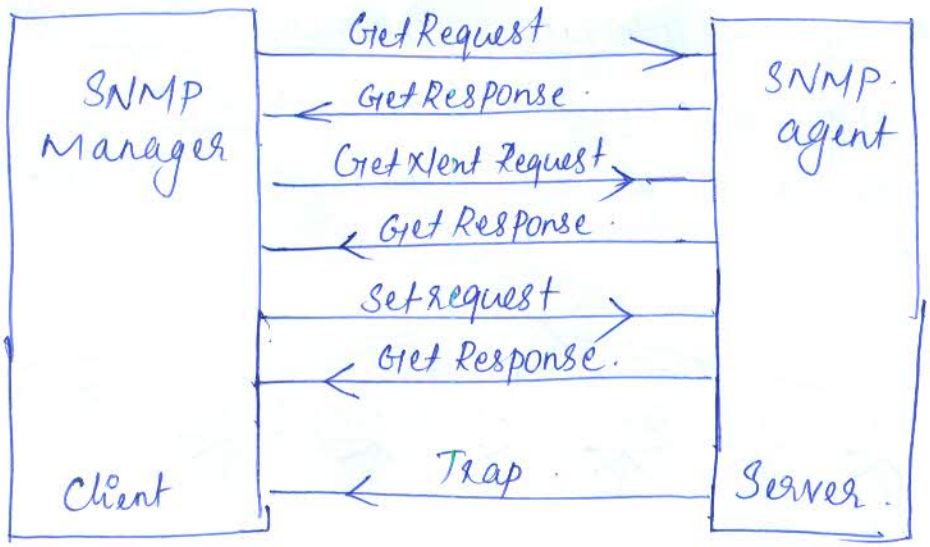
- It mostly used to retrieve the values of the entries in a table.
- If the manager does not know the indexes of entries, it cannot retrieve the values, However it can use GetNext Request and define the object.

Get Response :- The GetResponse message is sent from an agent to a manager in response to GetNext Request.
It contains the values of variables requested by the manager.

Set Request :- The Set Request message is sent from the manager to the agent to Set (store) a value in a variable.

Trap :- The trap message is sent from the agent to the manager to report an event.

Eg :- If the agent is rebooted, it informs the manager and reports the time of rebooting.



CN

World Wide Web

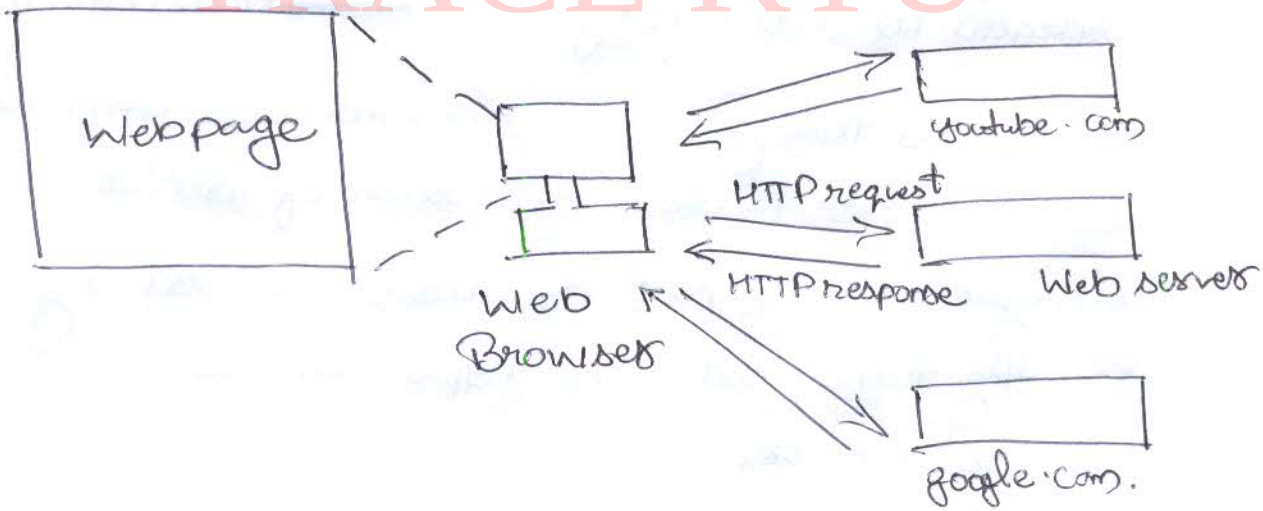
- It is an architectural framework for accessing linked content spread out over millions of machines all over the internet.
- Easy for begenness to use & provide access with a rich graphical interface to an enormous wealth of information on almost every conceivable subject.
- Web began in 1989, first prototype operated ~~three~~ 18 months later. (Tim Berners Lee).
- First graphical browser called Mosaic was released in Feb 1993 by Marc ~~Ander~~ Andreessen.

- The period through 2000, when many web companies became worth hundreds of millions of dollars overnight only to go bust practically the next day when they turned out to be hype, even has a name, dot com era.

- W3C → World Wide Web Consortium.
  homepage → www.w3.org.

Architectural Overview

- Web consists of vast worldwide collection of content in the form of webpages.
- The idea of having one page point to another, now called hypertext was invented by Vannevar Bush in 1945. (before internet was invented).
- Pages are viewed in a program called a browsers.
- Browsers fetches the page requested, interprets the content and displays the page properly formatted on the screen.
- A piece of text, icon, image and so on associated with links to other pages is called a hyperlink.



- Page fetching is done by the browser without any help from the users.
- Each page is fetched by sending a request to one or more servers which respond with the contents of the page.

- The request-response protocol for fetching pages is a simple text based protocol that runs over TCP. It is called HTTP. (Hypertext Transfer Protocol).

- The content may simply be a document read off a disk, or result of database query & program execution.

- The page is a static page if it is a document that is the same everytime it is displayed.

- In contrast, if it was generated on demand by a program or set of programs it is a dynamic page.

- Dynamic page may present itself differently each time it is displayed. (eg: bookstore website).

- Cookie stores the information about which user likes what & prefers what to be bought or watched on web.

Client Side

- Each page on the web is assigned an URL (Uniform Resource Locator) that effectively serves as the page's worldwide name.

- URL has 3 parts:
  - → protocol ~~also~~ also known as schema
  - → DNS name of the machine on which the page is located.
  - → path uniquely indicating the specific page.

eg:

http://www.cs.washington.edu/index.html

http: protocol; www.cs.washington.edu: DNS name of the host; index.html: path name

When an user clicks on a hyperlink, the browser carries out a series of steps in order to fetch the page pointed to.

Steps:

1. The browser determines the URL.
2. The browser asks DNS for the IP address of the server www.cs.washington.edu.
3. DNS replies with 128.208.3.88.
4. The browser makes a TCP connection to 128.208.3.88 on port 80 the well known port of the HTTP protocol.
5. It sends over an HTTP request asking for page /index.html
6. The www.cs.washington.edu server sends the page as an HTTP response, for example, by sending the file /index.html.
7. If the pages include URLs that are needed for display, the browser fetches the other URLs using the same process.
8. The browser displays the page.
9. The TCP connections are released if there are no other requests to the same servers for a short period.

- The http protocol is the web's native language, the one spoken by web servers.
- FTP protocol is used to access files by FTP, file transfer protocol. Web makes it easy to obtain files placed on numerous FTP servers throughout the world by providing a simple, clickable interface instead of command line interface.
- The Mailto protocol does not fetch webpages but allows users to send email from a web browser.
- RTSP and SIP protocols are for establishing streaming media sessions & audio & video calls.
- URLs are generalized into URIs (Uniform Resource Identifiers). Some URIs tell how to identify or locate a resource while others tell the name of the resource but not where to find out. Such URIs are called URNs (Uniform Resource Names).

- To be able to display a webpage, the browser has to understand its format. Webpages are written in a standardized language called HTML.
- Browser consults the table of MIME types to recognize the type of files used in the webpage.

- A plug in or helper applications can be used for this purpose. A plugin is a third-party code module that is installed as an extension to the browsers. Common examples are PDF, Flash etc.

- Helpers application is a complete program running as a separate process. It usually just accepts the name of a scratchfile where the content file has been stored, opens the file & displays the content.

SERVER SIDE

The server is given the name of a file to lookup and return via the network. (In both cases) the steps that the server performs in its main loop are:

1. Accept a TCP connection from the client.
2. Get the path to the page which is the name of file requested.
3. Get the file.
4. Send the contents of the file to the client.
5. Release the TCP connection.

- Web servers are implemented with a different design to serve many requests per second. One problem with the simple design is that accessing files is often the bottleneck.

Issues:

1. Disk reads are very slow compared to execution.
2. Same files may be read repeatedly using operating system calls.
3. Only one request is processed at a time.

- Make the servers multithreaded to tackle the problem of serving a single request at a time.

> Steps that occur after the TCP connection if any secure transport mechanism have been established.

1. Resolve the name of the webpage requested.
2. Perform access control on the webpage.
3. Check the cache.
4. Fetch the requested page from disk or run a program to build it.
5. Determine the rest of the resource. ← (eg: MIME)
6. Return the ~~setu~~ response to the client.
7. Make an entry in the server log.

STATIC WEBPAGE

The basis of the web transferring web pages from server to client. Web pages are static. That is, they are just files sitting some server that present themselves in the same way each time they are fetched and viewed. A page containing a video can be static webpage.

HTML

- HTML was introduced with web. It allows users to produce webpages that include text, graphics, video, pointers to other webpages and more.
- HTML is a markup language or a language for describing how the documents are formatted.

- A webpage consists of head & body each enclosed by tags. although most browsers do not complain if the tags are missing.

CSS

- Cascading Style sheets introduced style sheets to the web with HTML 4.0.
- CSS defines a language for describing rules that control the appearance of tagged content.

Dynamic Webpages and Web Applications

- Web is being used for applications and services such as buying products on e commerce sites, exploring maps etc. The twist ~~in~~ is that these applications run inside the browsers with user data stored on servers in internet data centers.

  They use web protocols to access information via the internet and the browser to display a user interface.

- The advantage is that users do not need to install separate applications and user data can be ~~accessed~~ accessed from different computers and backed up by the service providers. This model is a prevalent form of cloud computing.

Server Side.

Standard API's have been developed for webservers to invoke programs. The existence of these interfaces make it easier for developers to extend different ~~webpage~~ servers with web applications.

- First API method for handling dynamic page requests which was available since beginning is called CGI or Common Gateway Interface.

- CGI provides an interface to allow webservers to talk to back-end programs & scripts that can accept i/p & generate HTML pages in response.

- A popular language for writing these scripts is PHP (Hypertext preprocessor). To use it, the server has to understand PHP, just as a browser has to understand CSS to ~~insts~~ interpret webpages with style sheets.

- JSP (JavaServer Page) is similar to PHP except that the dynamic part is written in the java programming language instead of PHP.

- ASP.NET (Active Server Page.Net) is Microsoft's version of PHP & javaserver page. It uses programs written in Microsoft's proprietary .NET networked application framework for generating the dynamic content.

Client Side

- CGI & PHP scripts solve the problem of handling i/p & interactions with databases on the server. but none of them can interact with users directly. The technologies used to produce interactive webpages are broadly referred to as Dynamic HTML.

- Most popular scripting language for the client side is JavaScript. It is a high level language.

- An alternative to javascript on windows is VBScript which is based on visual basic.

- Another popular method is use of applets for a virtual computer called JVM.

- ActiveX controls used by Microsoft instead of java applets.