# Introduction to computer networks

**Definition :**

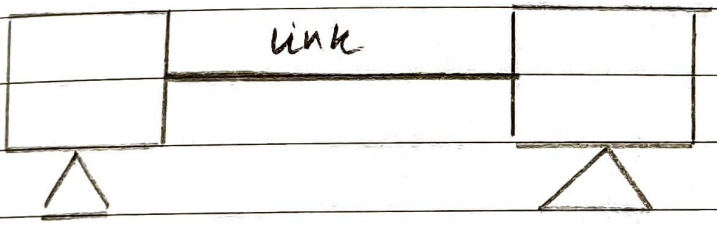
When 2 or more computers communicate with each other it is called as a computer network

**Types of connection :**

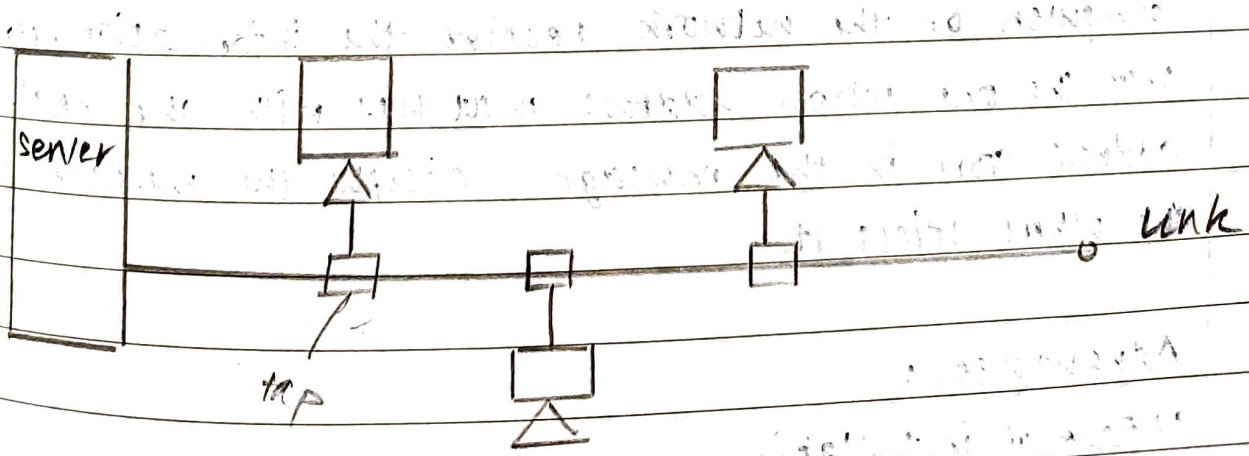
There are 2 possible types of connection

1) Point-to-point / (P2P)
2) multipoint

Point-to-Point connection provides a dedicated link between 2 devices. The entire capacity of the link is reserved for transmission between these 2 devices




multipoint connection is the one in which more than 2 devices share a single link. The capacity of the channel is shared between all the connecting devices



Topology

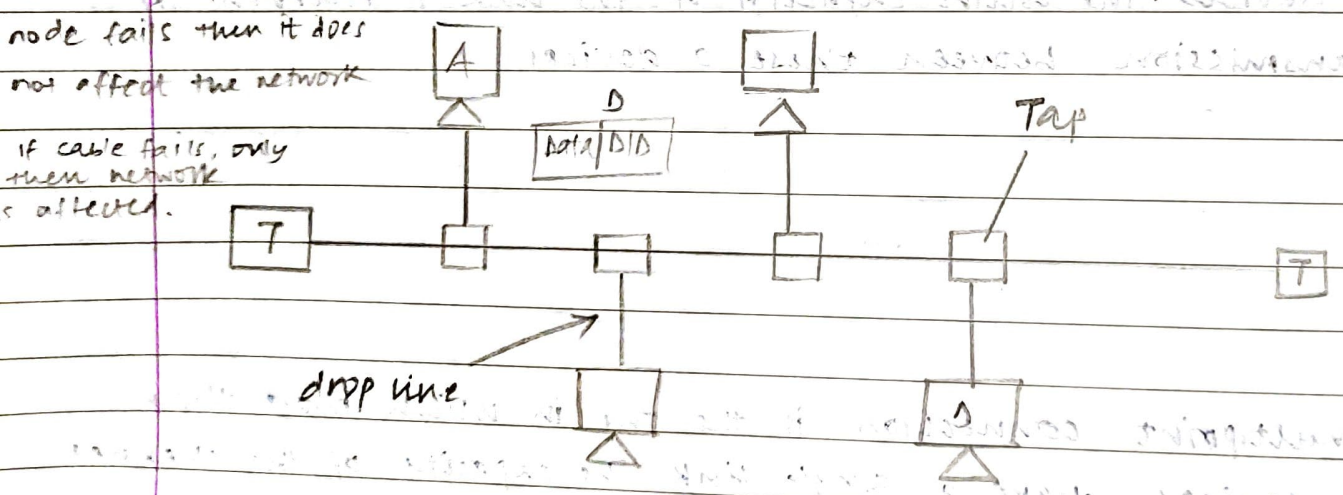
— The term topology refers to the way in which a network is laid out physically.

- 2 or more devices connect to a link and 2 or more links form a topology

1) Bus topology

It uses a multipoint connection; one long cable acts as the backbone to the link all the devices in the network

All the nodes are connected to a bus cable by drop line and taps




when a computer sends a signal on the cable, all the computers on the network receive the information. However only the one whose address matches with the destination address store in the message; accepts the message while all the others reject it

Advantages:

1) Ease of installation
2) less cabling need
3) cost efficient

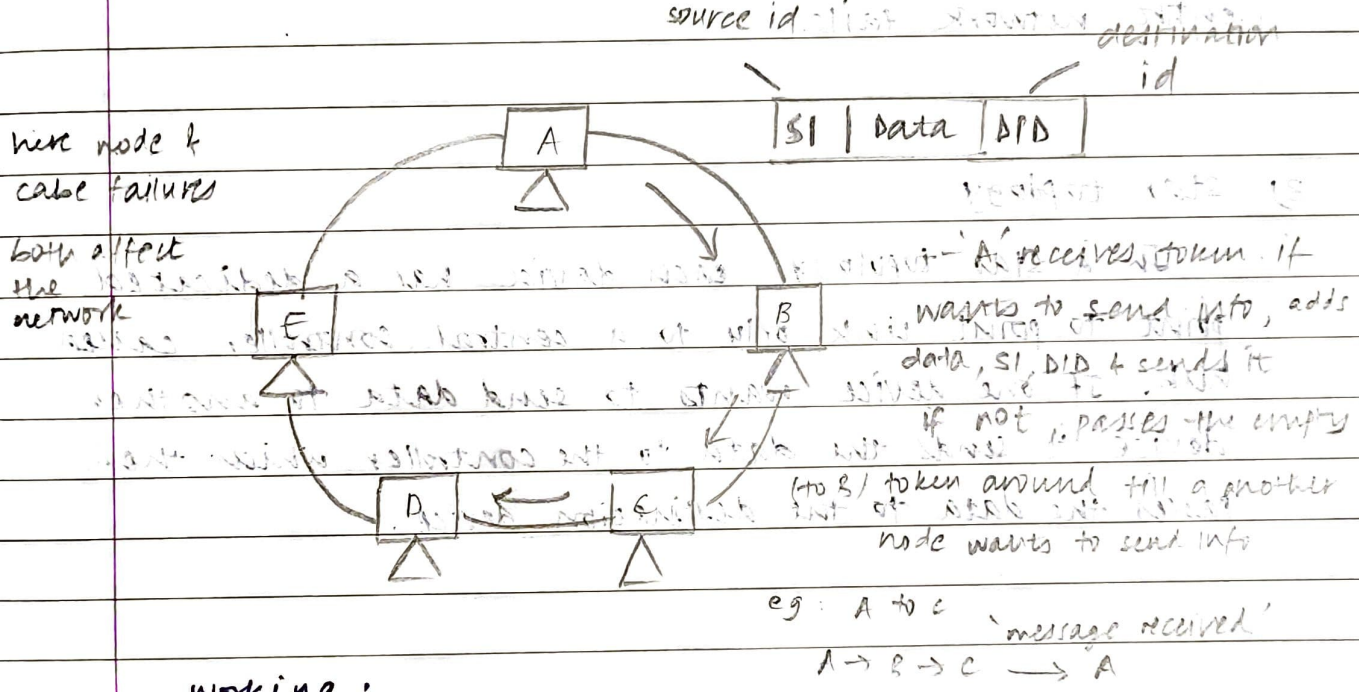
Page No.:
Date:

Disadvantages

1) difficult to add new connections
2) a break in the backbone cable stops the entire transmission

2) Ring topology / Token ring topology

Each device has a dedicated point-to-point connection with only 2 devices on their either side
The nodes are connected in a ring and data travels in one direction using a control signal called 'token'




working :

(1) A short message called token is passed around a ring until a computer wishes to send information
(2) The computer modifies the token, adds an address and data and sends it around the ring
(3) Each computer receives the token and info passed & passes to the next until the address matches
(4) The receiver computer returns a message to the sender that the message is received.
(5) The sending computer then creates another token & places it on the network

Page No.:
Date:

Advantages :

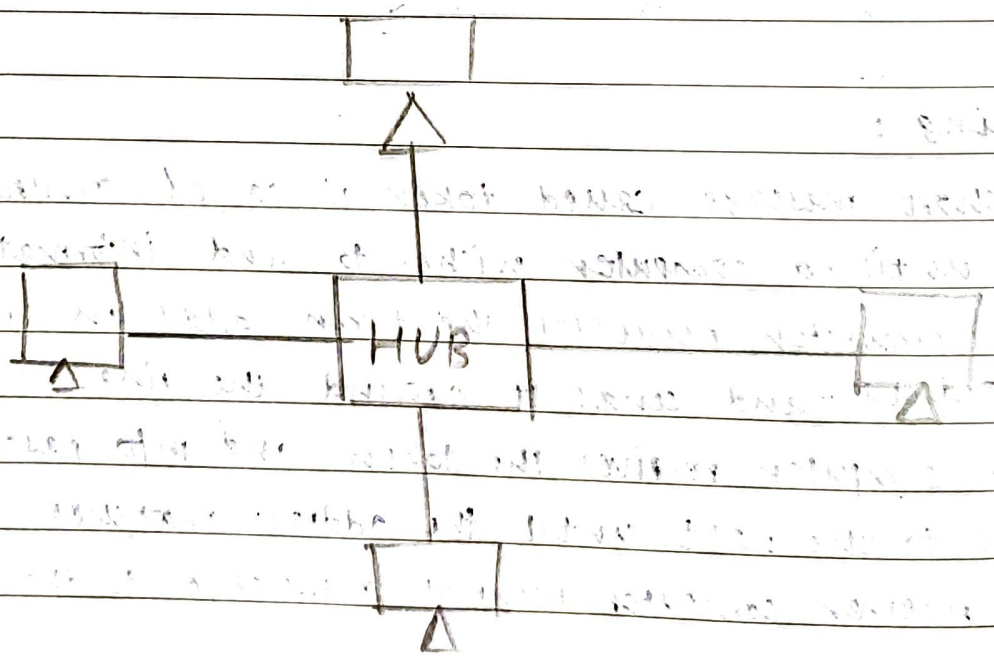
1) Easy to install
2) Adding or removing devices requires changing only 2 connections
3) Fault isolation / detection is simplified

Disadvantages :

1) A break in the ring can disable the entire network
2) If any node fails the token cannot be passed so the entire network fails.

3) Star topology

In a star topology, each device has a dedicated point to point link only to a central controller called hub. If one device wants to send data to another device it sends the data to the controller which then passes the data to the destination device.



Page No.
Date:
YOUVA

Advantages :

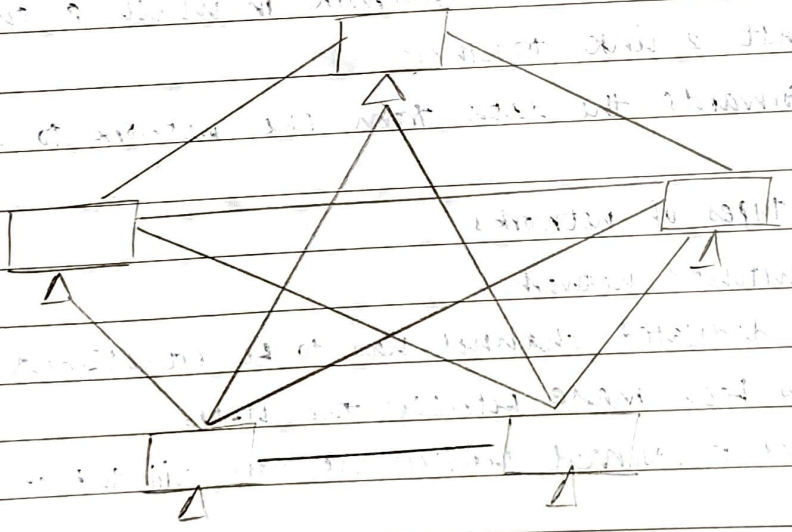
1) It does not allow direct traffic between the devices
2) less expensive than mesh topology
3) requires only one link and one i/o port per device
4) easy fault identification & fault isolation

Disadvantages :

1) star topology is dependent on one single point - the HUB. If the HUB goes down, the entire network is dead

4) mesh topology

In mesh topology every device has a dedicated point to point link with every other device in the network. The link carries traffic only between the 2 devices it connects.



no of links required = $\frac{n(n-1)}{2}$

Advantages :

1) Each connection can carry its own data rate, thus eliminating the traffic problem
2) It is robust i.e if one link becomes unstable, it doesnot affect the entire system
3) Privacy and security is maintained
4) Fault identification & Isolation is easy

Disadvantages :

1) Amount of cabling and no of i/o port requirements are huge
2) It is expensive
3) Adding and removing of devices is difficult.

→ switching

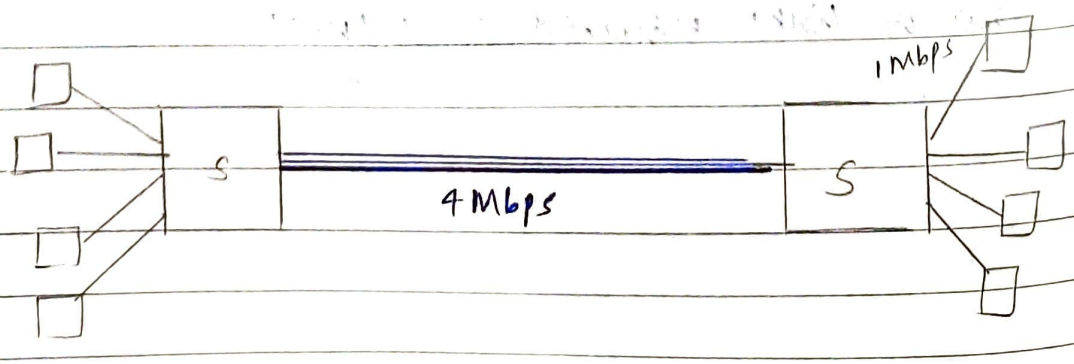
An internet is a switched network in which a switch connects atleast 2 links together

A switch forwards the data from one network to another network.

There are 2 types of networks

1) circuit switched network

* 1. In this dedicated channel has to be established before the call has been made between the users
2. This channel is reserved between the users till the connection is active.




— high capacity line — data line

3. Phone : 4 telephones at each side are connected to a switch. The line connecting the 2 switches is a high capacity that can handle 4 voice communication lines at the same time (since there are 4 users)

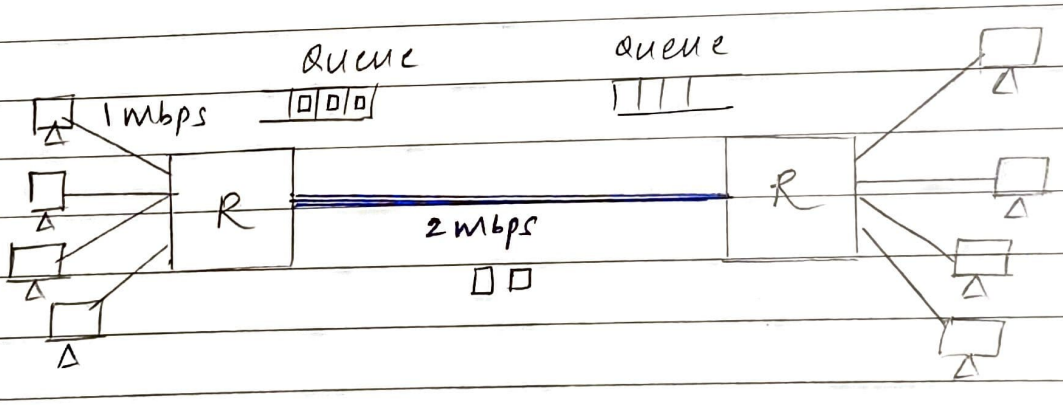
4. The switches can only forward the data.

5. If all the telephone sets are busy, the capacity of the thick line is completely used.

6. If only one telephone set is connected, only 1/4 th of the capacity is being utilized. That means the circuit switch network is efficient only when it is used at its full capacity

2) Packet switch network

1. In this it is not required to establish the connection initially. The connection is available to use by many users.
2. The communication between the 2 ends is in blocks of data called packets.
3. Instead of continuous communication, exchange of individual data packets between the nodes take place.
4. The switch functions for both storing and forwarding the data packets. This switch is called 'router'




5. A router in a packet switch network has a queue that can store and forward data packets.
6. Assume the capacity of the thick line = 2 x capacity of data line (2 mbps for given eg)

Page No.:
Date:

case (1) : If only 2 comps need to communicate with each other, there is no waiting for the packets.

case (2) : If packets arrived at one router when the thick line, working at its full capacity, the packets should be (is already) stored and forwarded in the order they arrive. Hence the packet switch is more efficient than a circuit switch network but packet delay may occur.

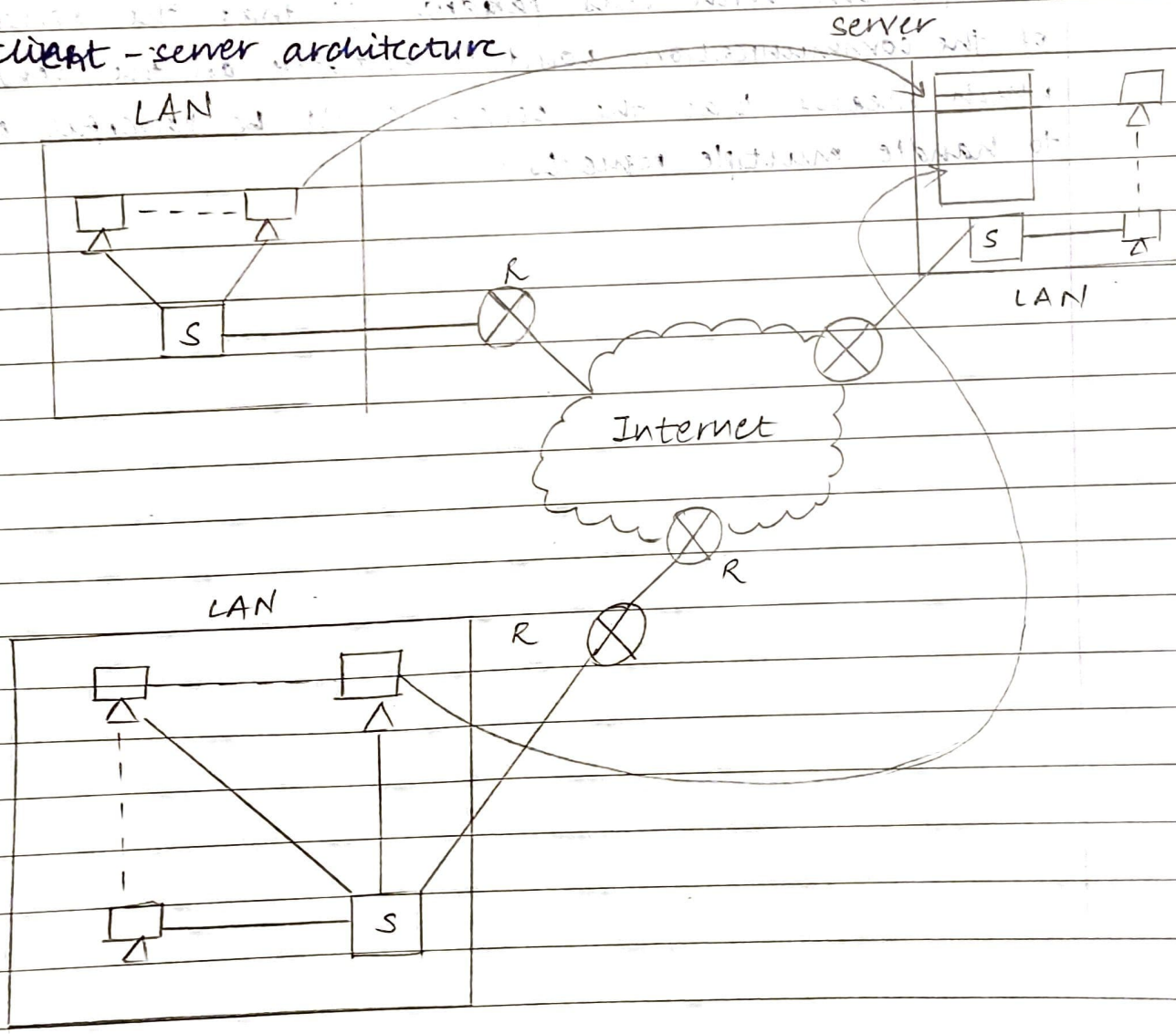
# Application layer

An application layer provides services to the user. communication is provided using a logical connection which means that the 2 application layers assume that there an imaginary direct connection through they can send and receive the data

Application layer paradigms :

The 2 paradigms that have been developed during the lifetime of internet are client - server paradigm and peer - peer (P2P) paradigm

client - server architecture.

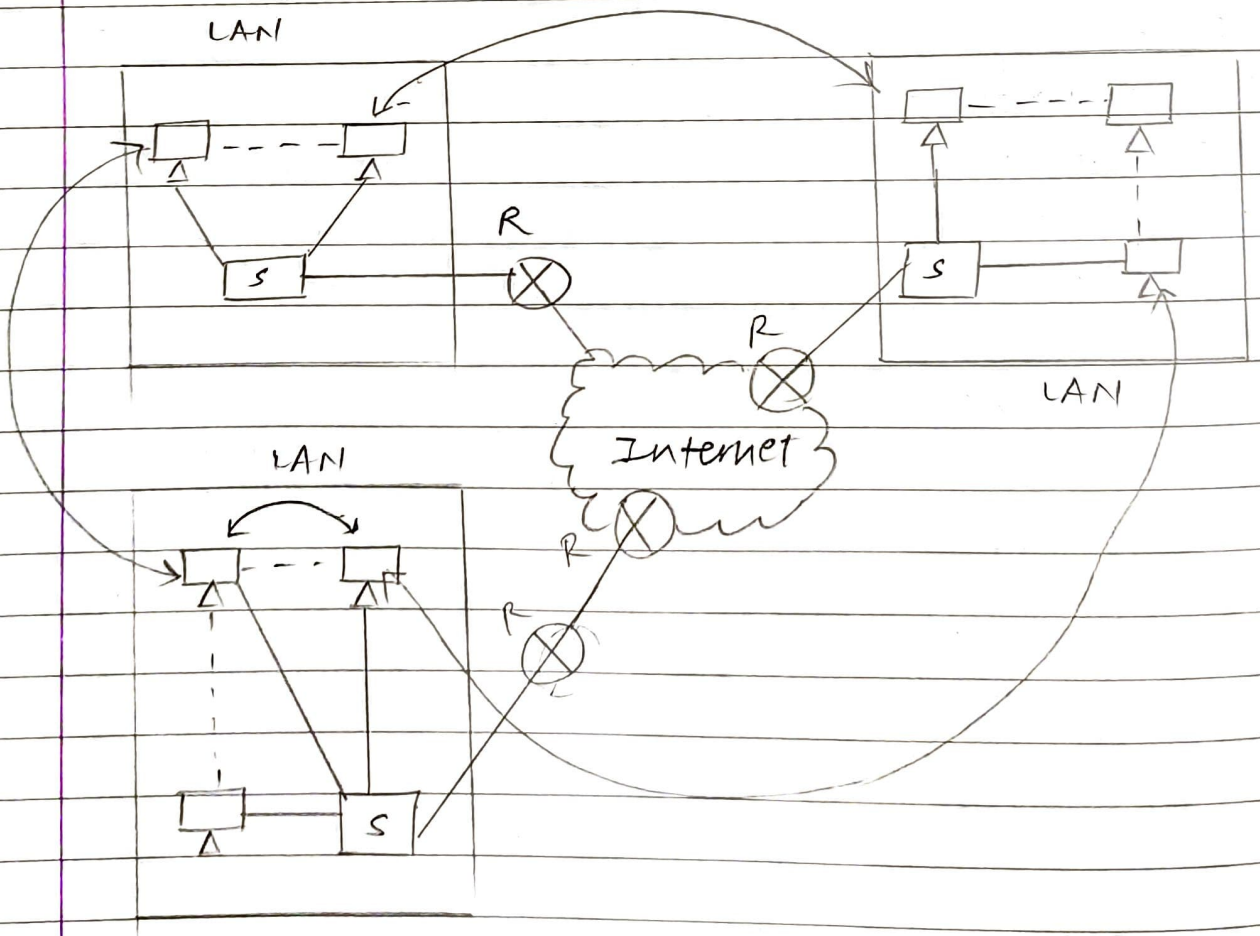


1) In the client-server, the service provider is an application program called a server process. It runs continuously waiting for another program called the client process to make a connection through the internet and ask for service.

2) The server process must be running all the time and the client process should be started when the client needs to receive a service.

3) The role of the client and the server program is totally different i.e we cannot run a client program as a server and vice-versa.

4) One problem with this paradigm is that the concentration of the communication load is entirely on the server which means that the server should be powerful enough to handle multiple requests.



1) In P2P network, there is no need for server process to be running all the time and waiting for the client process to connect.

2) In this paradigm the responsibility is shared between the peers.

3) A computer connected to the internet can provide service at one time & receive service at another time

Advantages of P2P

1) cost efficient:

/ no need of expensive servers to be running & maintained at all times

2) traffic distribution

Disadvantages

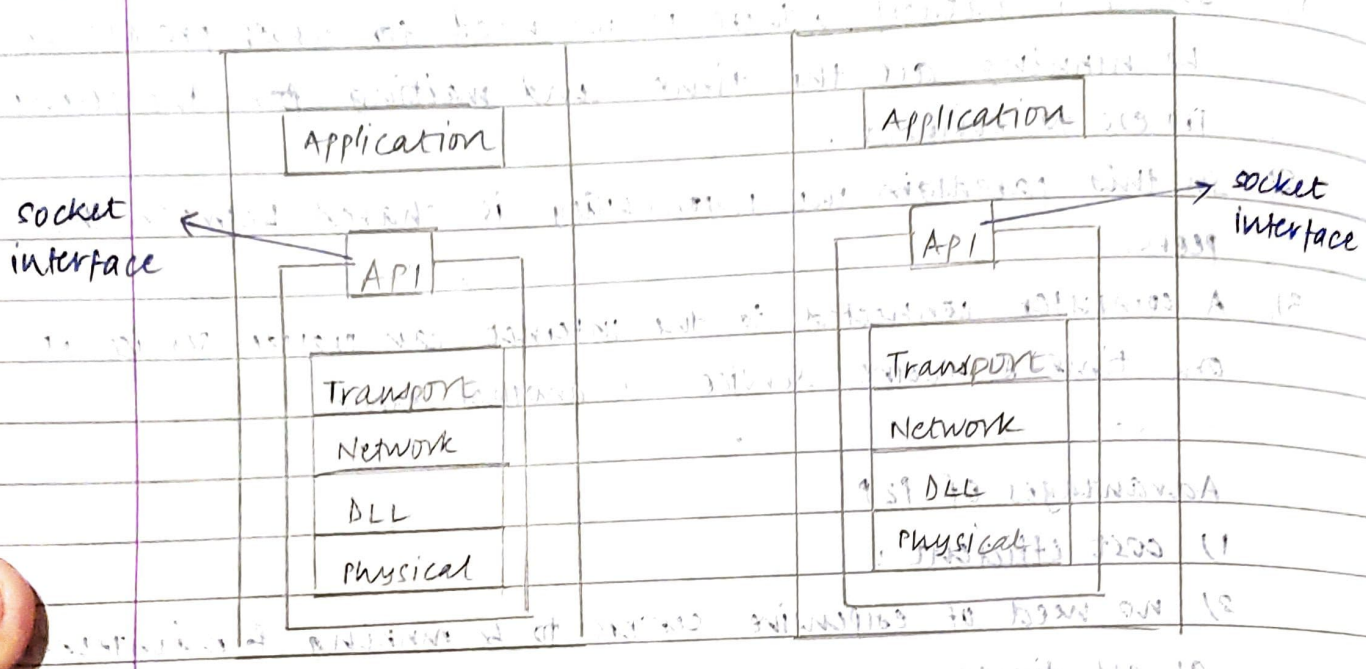
1) security is a major disadvantage as it is difficult to create secure communication between the distributed services

Applications: torrents, skype, etc

→ API - Application programming interface

A computer program is written with a predefined set of instructions that tells the computer what to do. If we need a process to communicate with another process, we need a set of instructions to tell the lowest 4 layers of the TCP/IP to open the connection, send and receive data from the other process and close the connection. These instructions are referred to an API

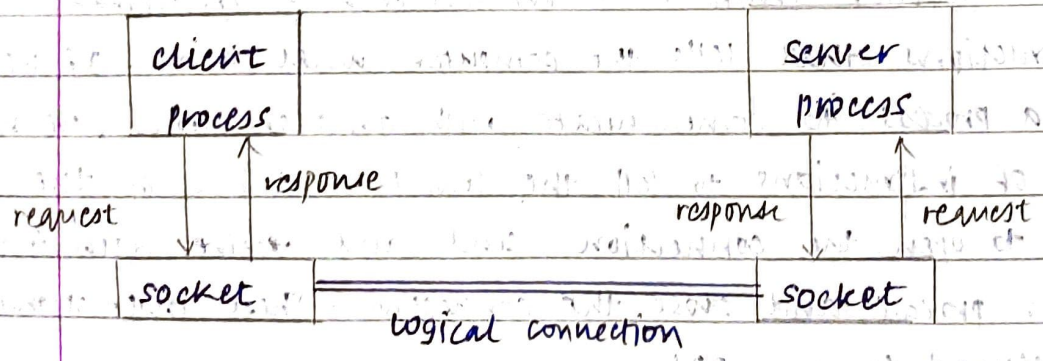
An interface is the set of instructions between 2 entities i.e the application layer and the operating systems that encapsulates the lowest 4 layers




client side server side

- socket

1. A data structure that is created and used by the application program
2. communication between a client process and a server process is actually a communication between 2 sockets created at both the ends




- The application program creates 2 sockets one at each end and defines the source and destination address to send and receive data to and from the server

Socket address - 1. The interaction between a client and server is a 2 way communication, hence we need a pair of addresses ~~to send~~ local and remote

2. A socket address should first define a computer on which the client or server is running. This is achieved with the help of IP addresses.

3. However, several clients or server processes may be running at the same time, hence we need another identifier to define a specific client or server process, in the communication. This is achieved by port number

| IP address | Port number |
|---|---|
| 32 bits | 16 bits |

socket address

Standard client - server applications :

There are 6 standard application programs :

1) HTTP and world wide web
2) File transfer
3) Electronic mail application
4) Remote login using telnet
5) Remote login using SSH (secure shell)
6) Domain name systems (DNS)

1) HTTP & world wide web

1. The web is a depository of information in which the documents called web pages are distributed all over the world, and related documents are linked together.
2. Linking allows one web page to refer to another webpage stored in another server somewhere else in the world.

3. The linking of webpages was achieved using the concept of hypertext

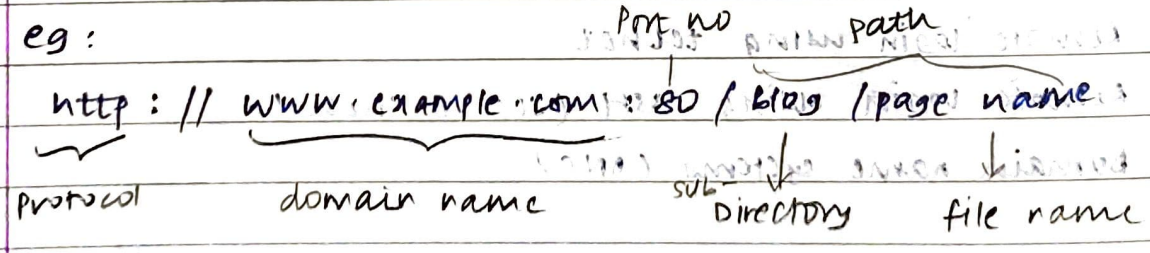
World wide web architecture

1. The world wide web is a distributed client-server service in which a client calls using a browser can access a service using a server.
2. The service provided is distributed over many locations called sites.
3. Each site holds one or more documents called webpages
4. Each webpage can contain link to another webpage in the same or other site

URL - uniform resource locator

1. A webpage needs to have a unique identifier to distinguish it from other webpages.
2. To define a webpage we need 3 identifiers i.e host, port no, and path
3. Before defining the webpage we also need to tell the browser what is the type of client-server application we want to use which is specified using protocol

eg:




Protocol - The first identifier is the abbreviation for the client server program that we need in order to access the webpage

most of the times, the protocol is http. we can make use of other protocols such as FTP (file transfer protocol)

Page No.:
Date:

ip add or
(domain name)

Host :- The host identifier can be the ip address of the server or the unique domain name given to the server

Port - The port is a 16 bit integer and is predefined for the client-server application.
eg: if http protocol is used for accessing the webpage port no 80 will be used

Path - The path identifies the location and name of the file in the OS

Web documents :
The documents in the world wide web can be grouped into 3 categories :

1) Static documents
1. These are fixed content documents that are created and stored in a server.
2. The client can only get a copy of the documents
3. The contents of the static docs can be changed at the server level.
4. The client can only use the document to view via a web browser
5. static docs are created using languages such as html, xml, (extensible markup language), html + css

2) Dynamic documents
1. It is created by a web server when a browser requests for the document
2. When a request arrives, the web server runs an application program or a script as a response to the browser that requested the document
3. Because a fresh document is created for each request the contents of a dynamic doc may vary from one request to another

4. The languages which are used to create dynamic docs are asp, jsp, php, etc
(active server pages) (java server pages)

3) Active documents

1. For many applications we need a program or a script to run at the client side. These are called active documents
2. An active webpage or an active doc, is a page where the browser performs the logic instead of a server
3. For eg: When you have a page which is showing the share prices, then you want it to update after every 5 seconds
4. To make this happen, the solution will be to use AJAX with java script
5. With active webpage everything is happening inside the browser without the need to reload the page everytime you want new information
6. eg: java script, java applets

→ Hypertext transfer protocol (http)

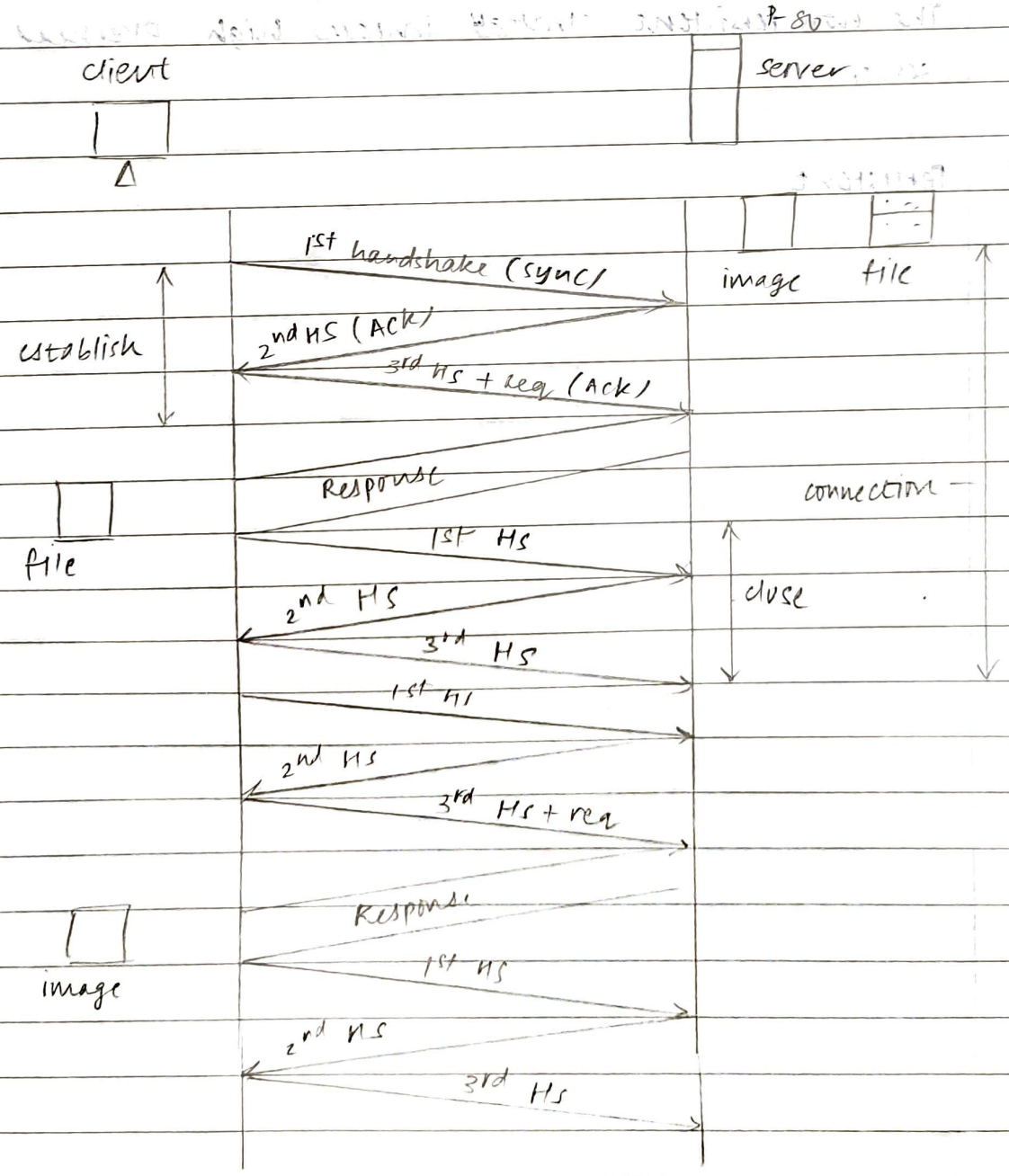
1. It is used to define how the client server programs can be written to retreive web pages from the web.
2. An http client sends a request and http server returns a response.
3. The server uses port number 80 whereas the client uses a temporary port for connection / communication
4. Http uses the services of TCP which is a connection oriented and reliable protocol.
5. This means that before any transaction between client and server can take place, connection needs to be established between them

Page No.:
Date:

Non-persistent & persistent connection

1. The hypertext concept embedded in web pages requires several requests and response.
2. If the objects to be retrieved are located on the same server we have 2 choices:
   a) To retreive each object using a new TCP connection - non persistent connection
   b) To make a TCP connection and retrieve all the objects together - persistent connection

Non-persistent



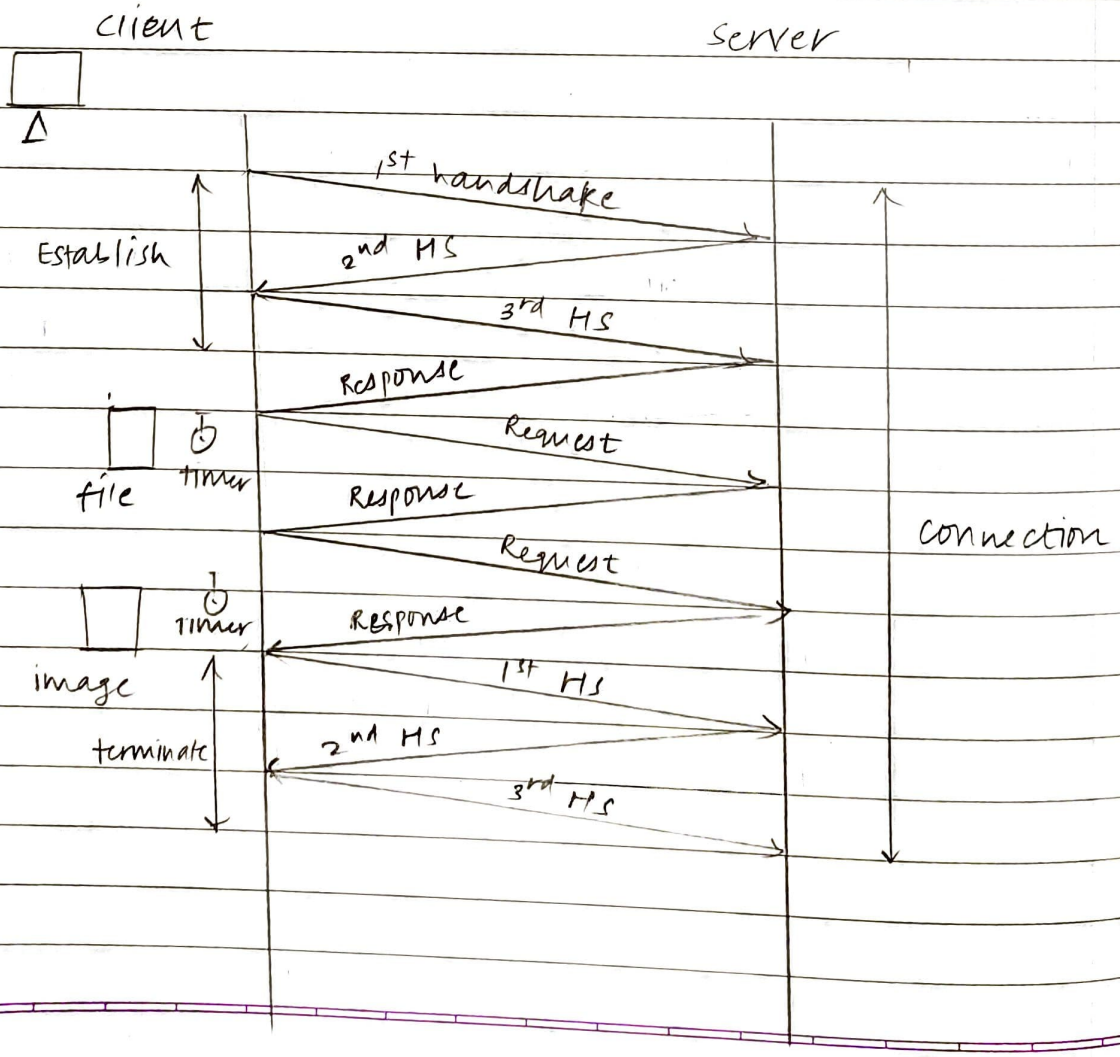
In a non persistent connection the TCP connection is made for each request or response. The following are the steps :

1. The client opens a TCP connection and sends a request
2. The server sends the response and closes the connection
3. The client reads the data until it encounters end of file and then closes the connection

If a file contains links to n different files, all located on the same server, the connections must be opened and closed n+1 times

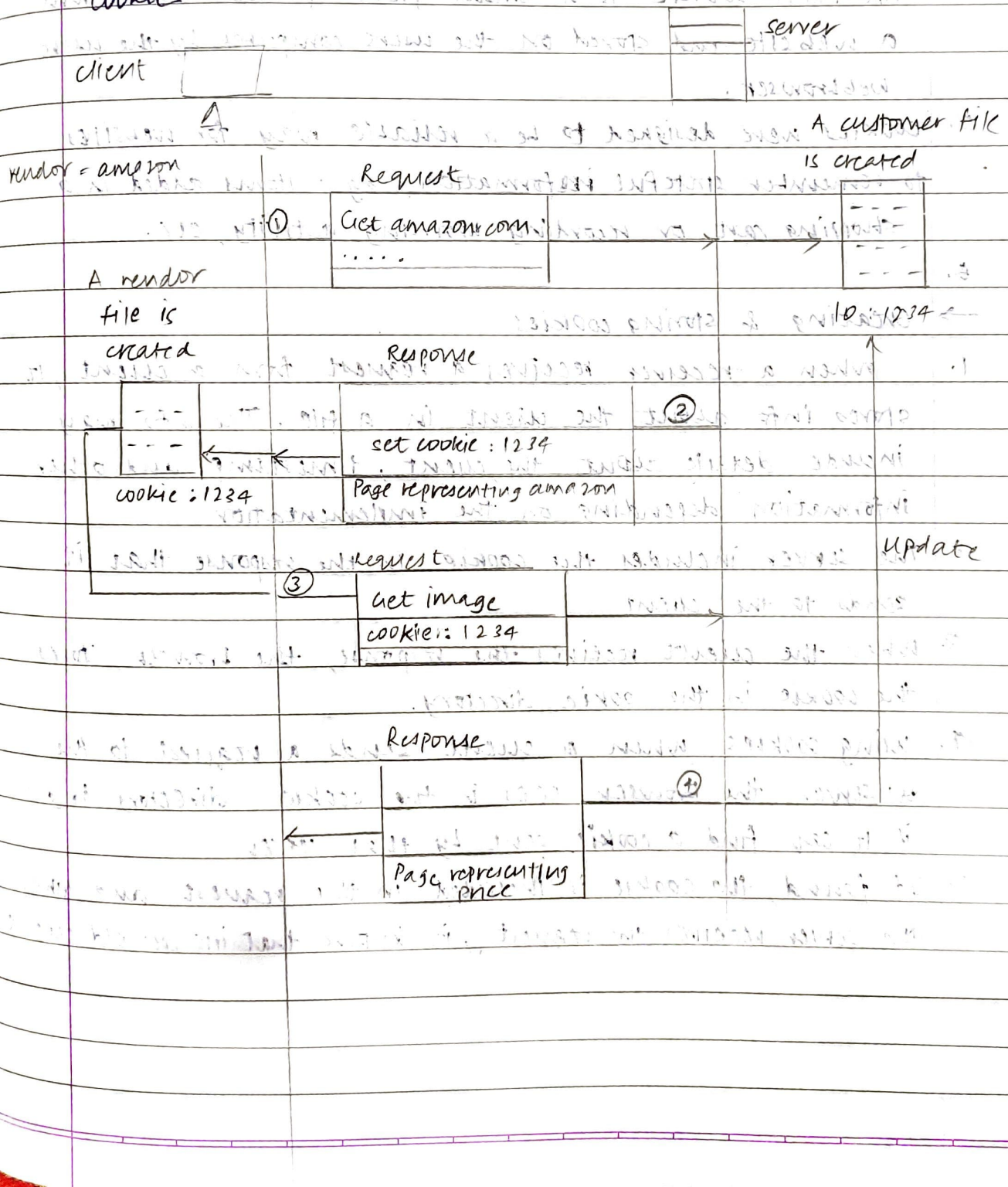
The non persistent strategy imposes high overhead on the server

2. Persistent



1. In a persistent connection the server leaves the connection open for more requests after sending the response.
2. The server can close the connection at the request of a client or if a timeout has been reached.
3. Time and resources are saved using persistent connection
4. The round trip time for connection establishment and connection termination is saved.

- cookies :



1. World wide web or www was designed as a stateless entity i.e a client sends a request, a server responds, and the relationship is over.
2. The original purpose of the web retrieving available documents exactly suited this design
3. Today, the web has other functions that needs to remember some information about the client. This need is fulfiled by the concept of cookies
4. An HTTP cookie is a small piece of data sent from a website and stored on the users computer by the users webbrowser.
5. Cookies were designed to be a reliable way for websites to remember stateful information, eg: items added in a shopping cart or recording browsing activity, etc.

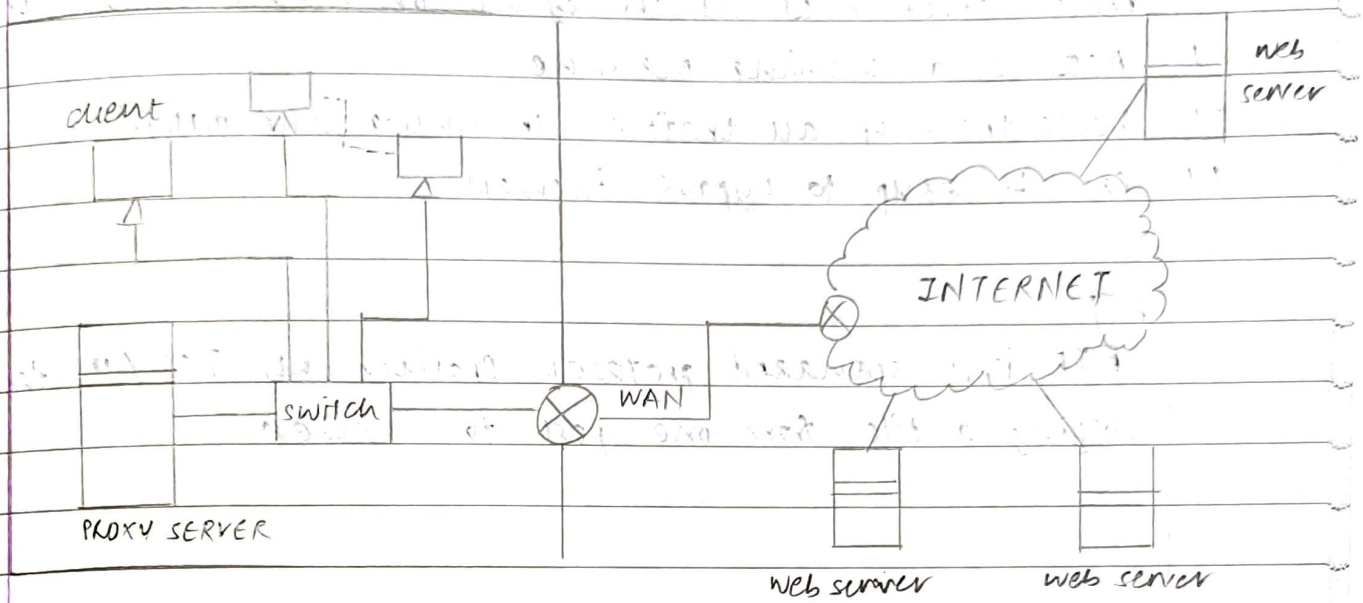
6.

→ Creating & storing cookies

1. When a receiver receives a request from a client, it stores info about the client in a file. The info may include details about the client, timestamp and other information depending on the implementation
2. The server includes the cookie in the response that it sends to the client
3. When the client receives the response, the browser stores the cookie in the cookie directory.
4. Using cookies, when a client sends a request to the server, the browser sees in the cookie directory to see if it can find a cookie sent by that server
5. If found, the cookie is included in the request and when the server receives the request, it knows that it is an old client.

Page No.
Date:
YOUVA

Proxy server :



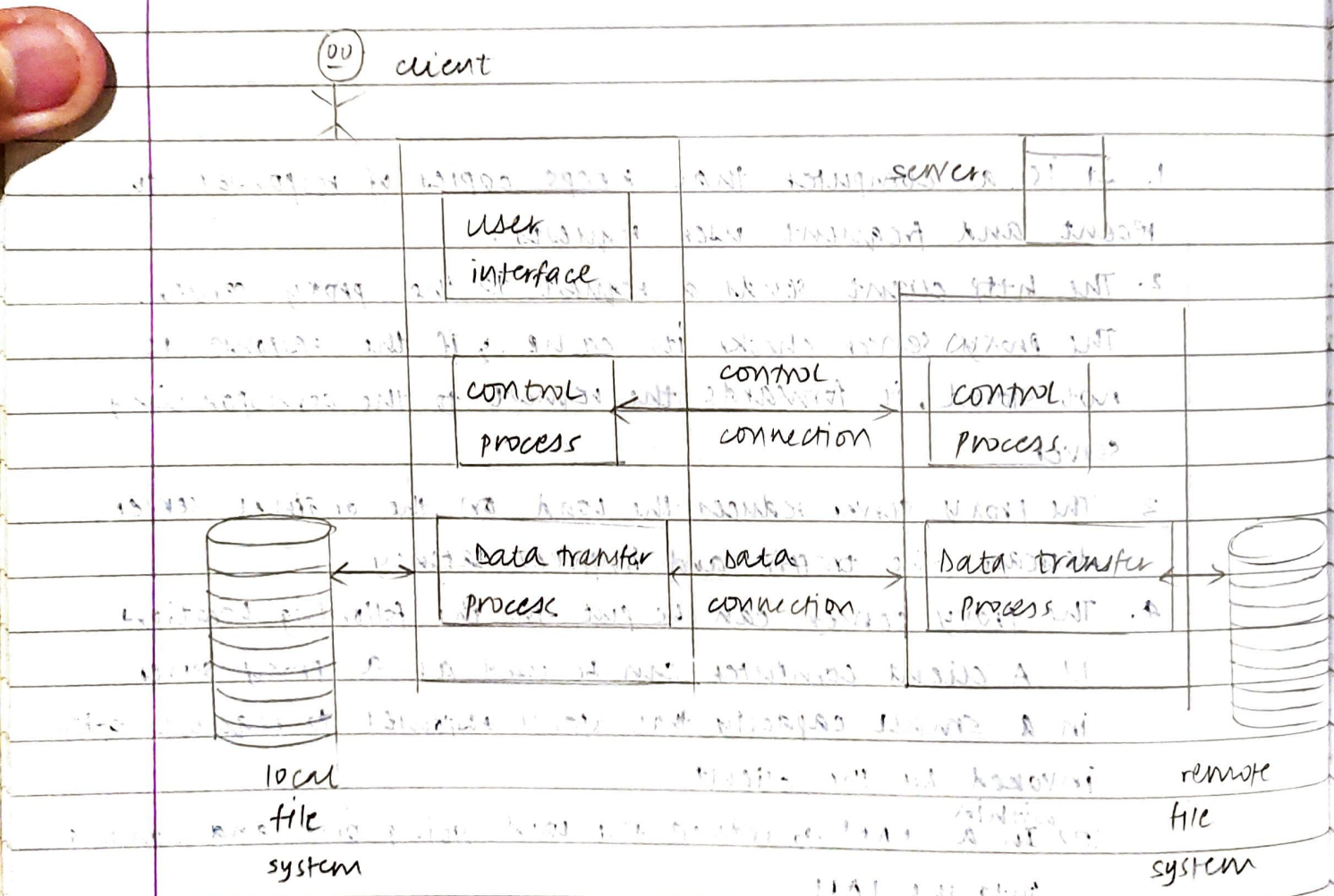

1. It is a computer that keeps copies of responses to recent and frequent user requests.
2. The http client sends a request to the proxy server. The proxy server checks its cache. If the response is not stored, it forwards the request to the corresponding server
3. The proxy server reduces the load on the original server, decreases the traffic and improves latency
4. The proxy server can be put in the following locations:
   (1) A client computer can be used as a proxy server in a small capacity that stores responses to request often invoked by the client
   (2) within a LAN to reduce the load going out and coming into the LAN
   (3) An ISP (internet service provider) with many customers can install a proxy server to reduce the load going out & coming into the ISP network

Functions of proxy server (P.S)

1) Proxy server acts as an agent on behalf of its client
2) Acts as a caching machine
3) keeps track of all traffic - incoming and outgoing
4) Can be setup to bypass firewall

FTP

FTP is a standard protocol provided by TCP/IP for copying a file from one post to another.

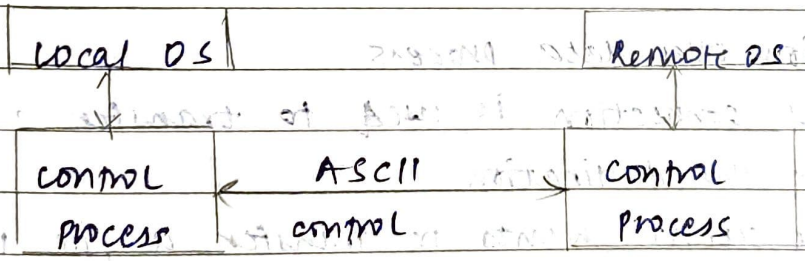


1. The client has 3 components - user interface, client control process and client data transfer process
2. The server has 2 components - server control, and server data transfer process
3. There are 2 connections which are established - control connection & data connection
4. The suppression of connection i.e commands and data transfer makes FTP more efficient
5. The control connection remains connected during the entire interactive FTP session, the data connection is opened & closed for each file transfer activity
* 6. FTP uses 2 well known TCP ports - port 21 is used for control connection & port 20 is used for data connection

→ 7. FTP communication

i) communication over control connection

For control communication FTP uses a set of ASCII characters. communication is achieved through commands and responses




commands are sent from the client to the server. Some FTP commands are :

- USER username
- PASS password
- MKD make directory
- RMD remove directory
- RETR retrieve the file
- QUIT logout of the system
- TYPE file type

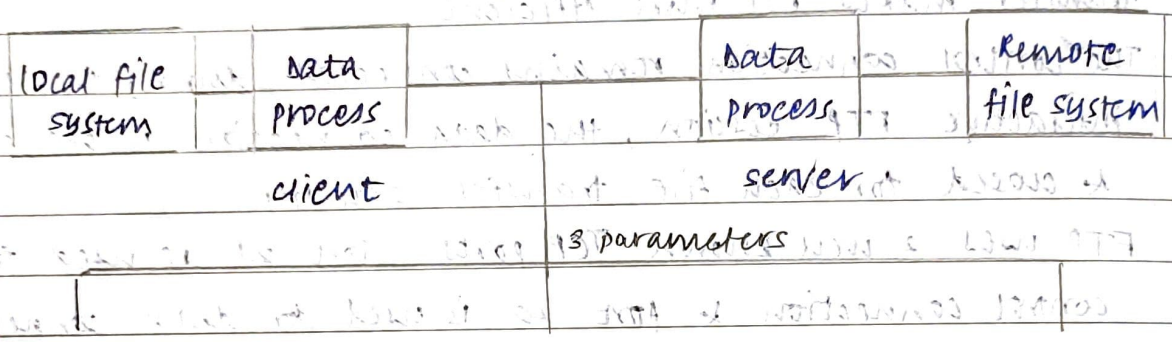
Every FTP command generates atleast 1 response. Following are a few examples of responses in FTP:

(server to client):

125 - Data connection open

230 - user login okay

ii) communication over data connection




1) File type
- ASCII
- EBCDIC
- Image

(extended binary coded decimal interchangeable code)

2) Data structure
- File structure (default)
- Record structure
- Page structure

3) Transmission mode
- Stream mode (default)
- Block mode
- compressed mode

communication over data process

The data connection is used to transfer a file from the source to the destination

1. If the client wants to transfer a file through the data connection, he must mention/define the type of the file to be transferred, the structure of the data & the transmission mode

(1) Data structure - FTP can transfer a file in one of the following structures of the data:

a. File structure - It is used by default & has no structure. It is just a continuous stream of bytes

b. Record structure - The file is divided into records. This can only be used with text files

c. Page structure - The file is divided into pages with each page having a page no and a page header. The pages can be stored and accessed randomly or sequentially

(2) File type - FTP can transfer one of the following file types :

a) ASCII file

b) EBCDIC file (extended binary coded decimal interchange code)

c) Image file

(3) Transmission mode - FTP can transfer file using one of the following 3 transmission modes :

a) Stream mode - This is the default mode, data is delivered from FTP to TCP as a continuous stream of bytes

b) Block mode - data can be delivered from FTP to TCP in a ... blocks where each block is preceded by a header

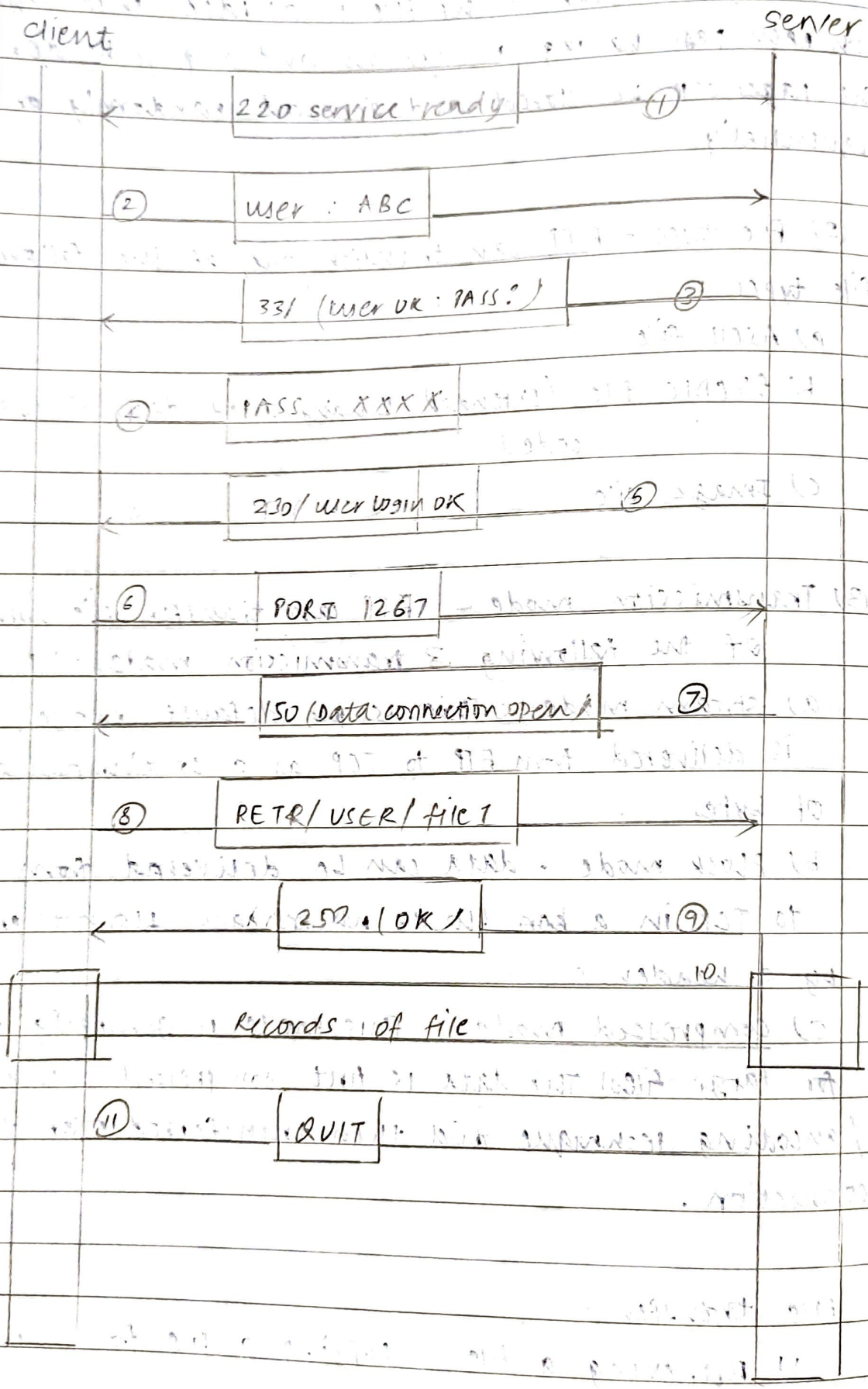
c) Compressed mode - This mode is generally used for large files. The data is first compressed using an compression/encoding technique and then transferred over the connection.

2. File transfer

1) Retrieving a file - copying a file from server to client i.e download

2) Storing a file - copying a file from client to server i.e uploading

3) Retrieving list - It gives a list of files or drives from server to client

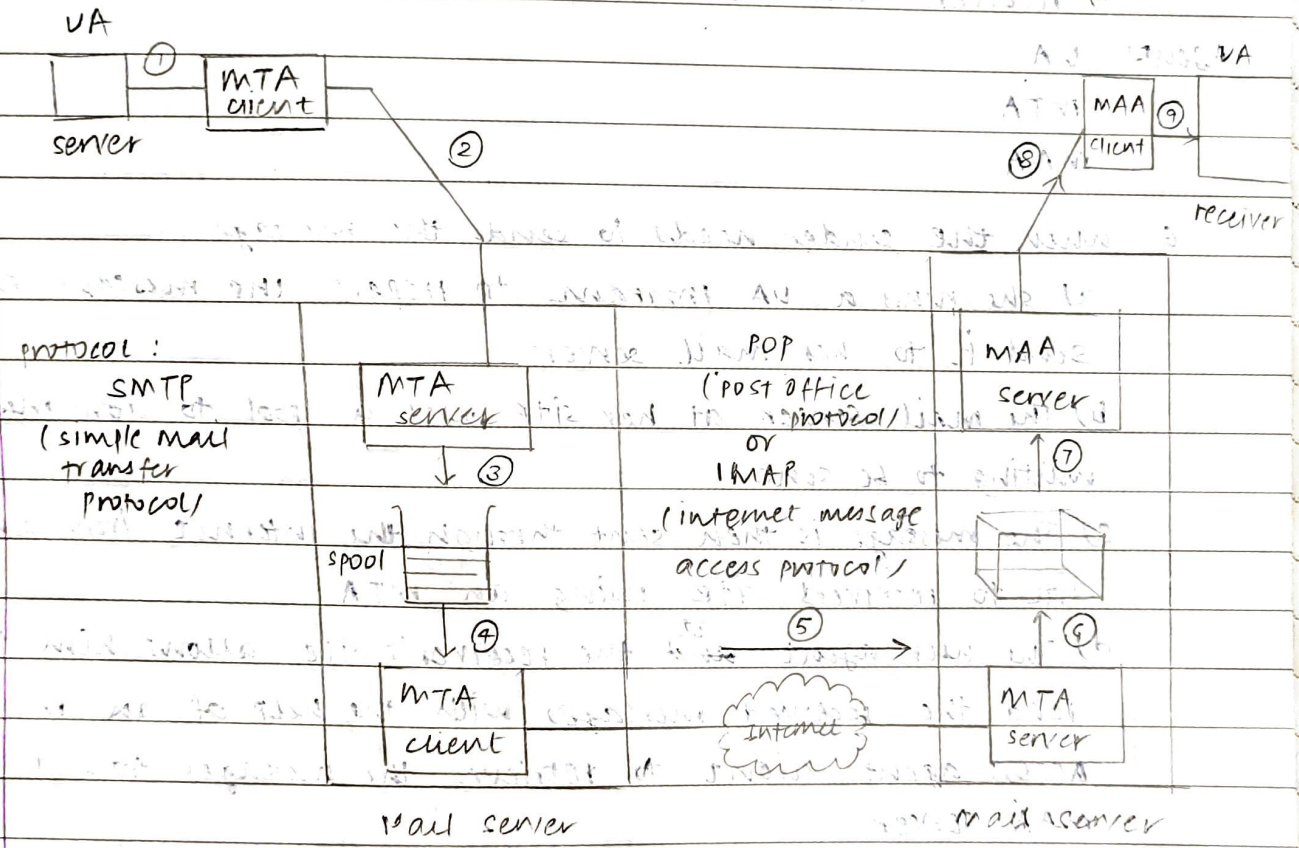

client
server
220 service ready
1
2
user : ABC
331 (user OK · PASS.)
3
4
PASS : XXXX
230 / user login OK
5
6
PORT 1267
150 / Data connection open /
7
8
RETR / USER / file 1
250 / OK /
9
10
Records of file
11
QUIT

Page No.:
Date:

# Electronic - mail

1. E-mail allows users to exchange messages. Email is considered as a one way transaction where the sender sends an email but he will may not expect a response.
2. The ideas of client - server programming is implemented using some intermediate servers.
3. The users run only client progs when they want & the intermediate servers apply the client-server paradigm.

## Architecture :




UA - user agent

MTA - message transfer agent

MAA - message access agent

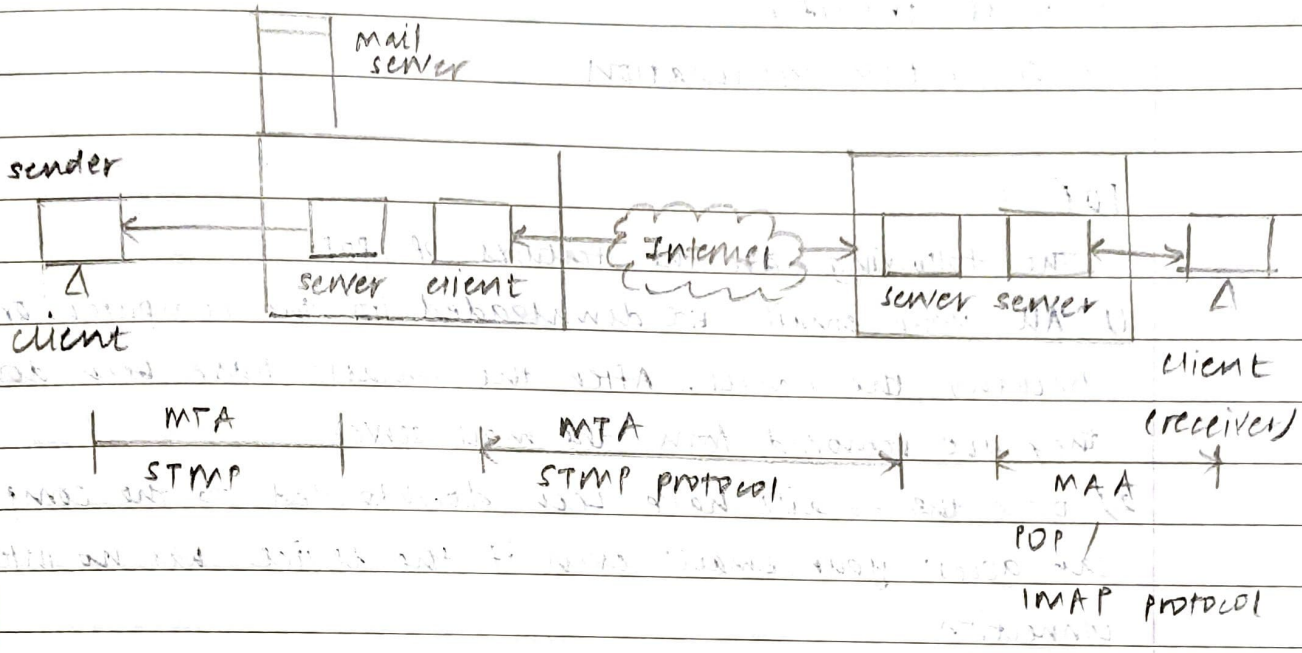
1. The sender and receiver of the email are connected to 2 mail servers.
2. The administrator has created one mailbox for each user, where the received messages are stored.
3. A mailbox is a part of the server hard drive. A special file with permission restrictions.
4. The administrator has also created a queue / spool to store messages waiting to be sent
5. A simple email takes 9 diff steps to go from the sender to receiver and uses 3 diff agents

Agents : UA
MTA
MAA

6. When the sender needs to send the message :
   a) she runs a UA program to prepare the message and send it to her mail server
   b) The mail server at her site uses a spool to store messages waiting to be sent
   c) The message is then sent through the internet from sender's site to receiver's site using an MTA
   d) The user agent at the receiver's site allows him to read the received messages with the help of an message Access agent client to retrieve the messages from a MAA server.
7. : The email system needs two user agents, 2 pairs of MTAs (client & server) and one pair of MAA (client & server)

Page No.:
Date:

Protocols used in Email :




1. most of the internet systems use simple mail transfer protocol (SMTP), has a method to transfer mail from one user to another.
2. SMTP is a push protocol and is used to send a mail whereas POP / IMAP are used to retrieve these mails at receiver's site.

SMTP:

1. It is an application layer protocol. The client who wants to send a mail opens a TCP connection and then sends the mail across the connection.
2. The SMTP is always on listening mode (server is on). As soon as it listens for a TCP connection from any client, the SMTP protocol initiates a connection on port 25
   SMTP uses port 25
3. After successfully establishing TCP connection, the client process sends the mail instantly
4. SMTP uses commands & responses to transfer messages between MTA client & MTA server
5. The process of transfering a mail message occurs in 3 phases:

1) connection establishment
2) mail transfer
3) connection termination

POP-3

The following are the features of POP-3 :

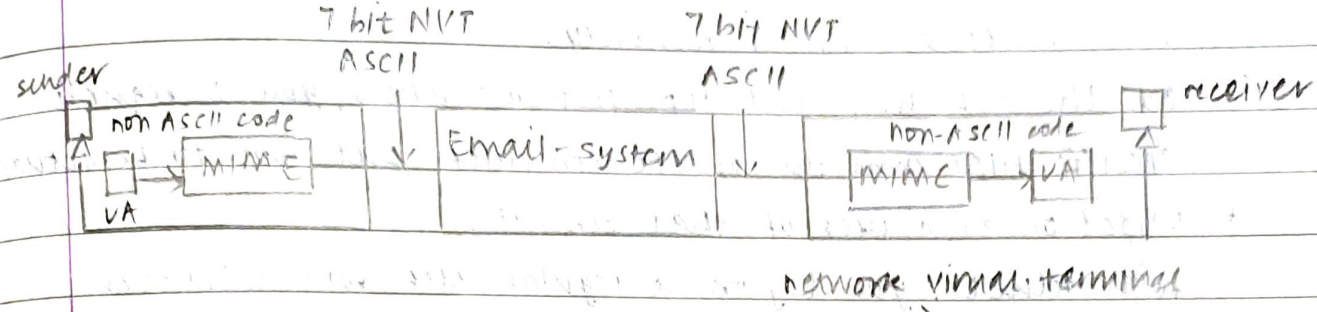
1) All your emails are downloaded to the computer or device checking the email. After the emails have been downloaded they are removed from the mail server
2) once the emails have been downloaded to the comp, you can access your emails even if the device has no internet connection
3) Data such as emails and folders are not synchronized between different devices. This means if you setup your email on your mobile device with POP-3, the emails will be downloaded completely on your mobile device POP-3 removes emails from mail server. It will no longer be possible for you to read your emails using the web because they are already downloaded on the mobile device.
4) POP-3 has 2 modes, keep mode & delete mode. In the delete mode, the mail is deleted from the mailbox after each retrieval; and in keep mode, the mail remains in the mailbox after retrieving. The delete mode is normally used when the user is working on a permanent computer

IMAP

1. Email remains on the email server allowing you to setup the account on multiple computers and devices to access your email
2. Folders can be created on the server to better manage the messages. The folders then sync across all devices used to check your email.
3. Sent messages are also saved in the sent folder allowing you to view sent emails from any device.
4. The server saves the status of an email. This allows you (read, unread, replied, forwarded etc)

to see the status from any computer or device.

MIME - multipurpose internet mail extension

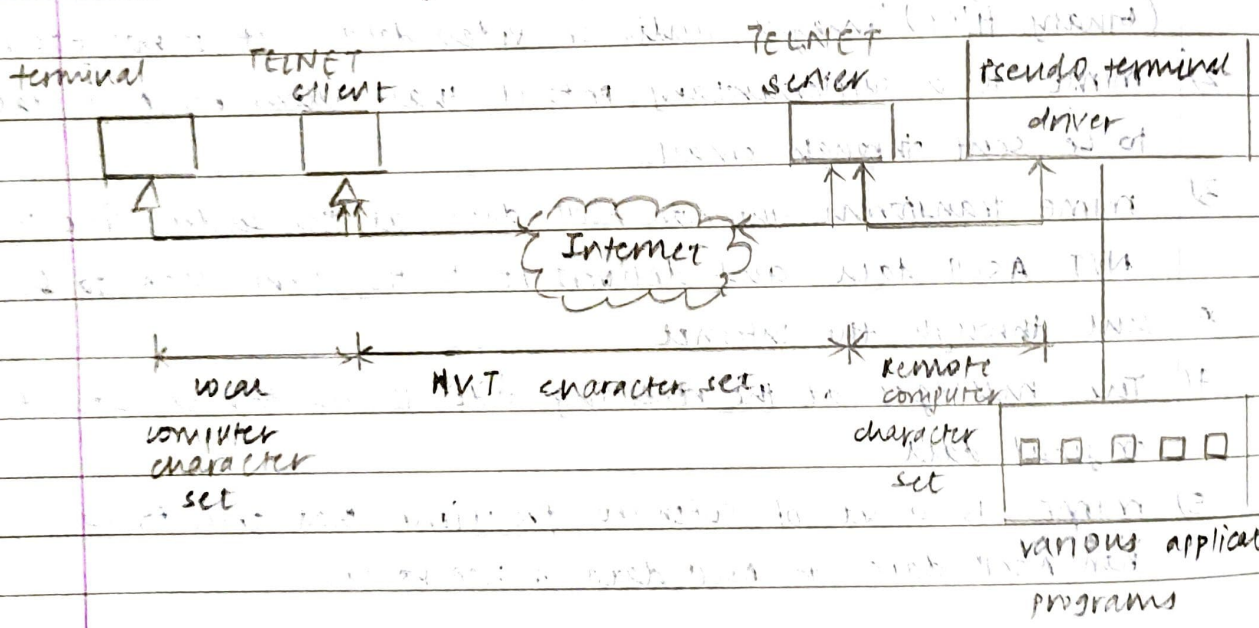



1) Electronic mail can send messages only in NVT 7 bit ASCII format. In other words, if a user wants to send email in any other language than English or he wants to send (binary files) images, audio or video data; it is not possible
2) MIME is a supplementary protocol that allows non ASCII data to be sent through email
3) MIME transforms the non ASCII data at the sender side to NVT ASCII data and delivers it to the client MTA to be sent through the internet
4) The messages at the receiving side is transformed back to original data
5) MIME is a set of software functions that transforms non ASCII data to ASCII data & vice versa.

| MIME Header |
| --- |
| mime-version : 1.1 |
| content-Type : type/subtype |
| content-Transfer-Encoding : Encoding Type |
| content-ID : message ID |
| content-Description : textual explanation |

TELNET :

1. Telnet is a protocol used for accessing remote computers. Through telnet, an administrator or another user can access someone else's computer remotely.
2. On the web http and ftp protocols allow you to request specific files from the remote computers but not to be actually logged on as a user of that computer.
3. With telnet, you log on a regular user with whatever permissions or priviledges you may have been granted.
4. It is mostly used by network administrators to remotely access and manage devices.




Process for remote login using TELNET

The following steps take place for remote login :

1) The user types at the keyboard of the terminal
2) The terminal driver at the local operating system accepts the characters, but sends them to the telnet client without interpreting them.
3) Telnet client converts them into NVT characters (universal character set)
4) The NVT characters travel on the internet to reach the remote machine

5) The NVT characters are applied to the telnet server which converts them appropriately so that the remote computer can understand them.

6) These characters are applied to a software called pseudo terminal driver which then passes the characters to the intended application

Disadvantages

1) TELNET is not a very secure system although it uses username & password for login.

2) A sniffing software would be enough to capture the login details of the user.

→ Secure shell (SSH) - Also known as secure socket shell

It is a network protocol that provides administrators with a secure way to access a remote computer. It provides strong authentication and secure encrypted data communication between 2 computers connecting over an insecure network such as the internet.

Whenever data is sent via computer to the network SSH automatically encrypts it, and when the data reaches its receiver, SSH automatically decrypts it.

SSH uses modern secure encryption algorithms and is effective enough to be used within mission critical applications at major populations. little loss of data affects company

SSH protocol covers authentication, encryption, and provides integrity of data transmitted over the network.

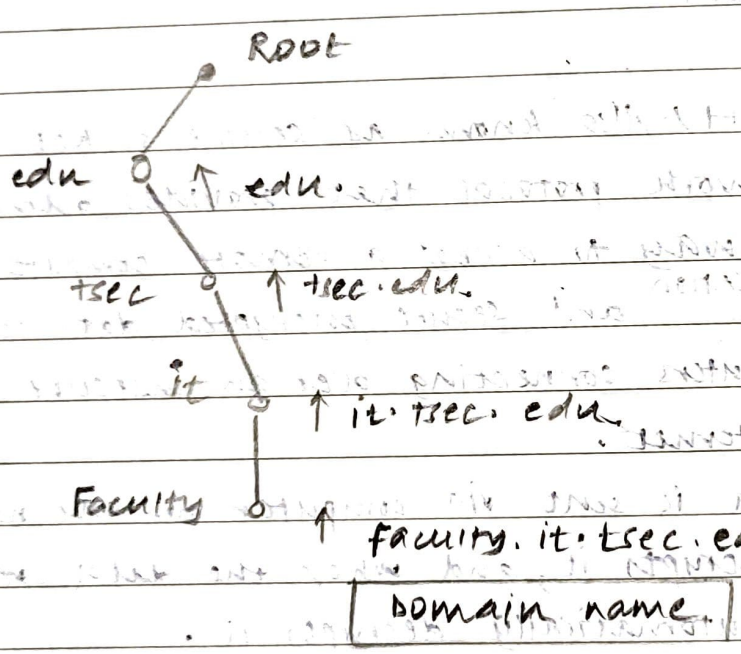
Page No.:
Date:

## Domain Name System (DNS)

To identify an entity over the internet, we use ip addresses, which uniquely identifies the host on the internet. ~~How~~.

However, people prefer to use names instead of numeric addresses. Therefore, the internet needs to have a directory system that can map a name to an address. This is achieved by DNS.

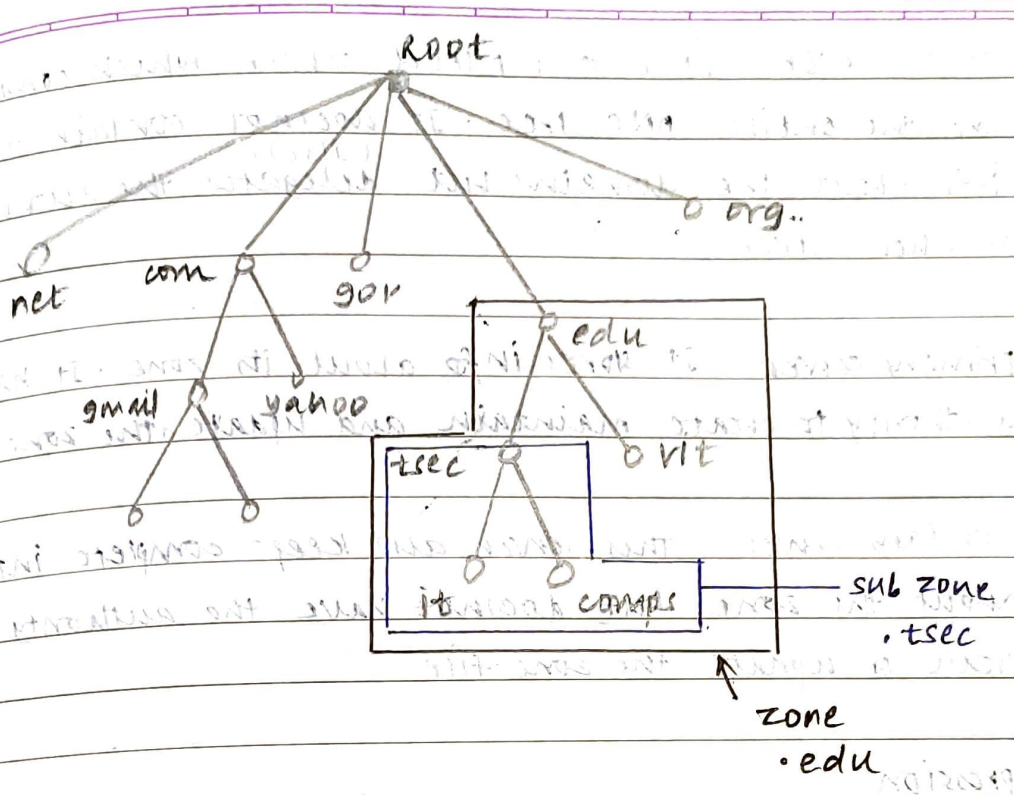
DNS helps to resolve host name to an ip address. It uses a hierarchial naming scheme and distributed ip addresses and associated names.




DNS architecture :

DNS comprises of domain names, spaces and name servers.

b)




1) Domain Name

It is a symbolic string associated with an ip address

It is divided into 2 categories:

a) Generic : example - .com, .edu, .gov, .org, .net

b) country : example - .in, .au, .uk, .us

2) Domain name spaces

It refers to a hierarchy in the internet naming structure

3) Name server

It contains the DNS database. This database comprises of various names and their corresponding ip addresses

Since it is not possible for a single server to maintain the entire DNS database, the info is distributed amongst many DNS servers.

There are 3 categories of name servers that manages the entire DNS

a) root server - It is a top level server which consists of the entire DNS tree. It does not contain any info about the domains but (gives) delegates the authority to other servers

b) Primary server - It stores info about its zone. It has the authority to create, maintain and update the zone file

c) Secondary server - This server also keeps complete info about the zone but does not have the authority to create or update the zone file

→ Compression

It is a reduction of no of bits needed to represent data. compressing data can save storage capacity, speed up file transfer and decrease the cost for storage, hardware and network bandwidth

There are 2 techniques for compression:

1) Lossless
2) Lossy

data unaltered
integrity maintained

Lossless compression - In this compression the integrity of data is preserved because the compression and decompression algorithms are exact inverse of each other i.e no part of the data is lost in the process.

It is normally used when we cannot afford to lose any data

eg: text file

Page No.:
Date:

# lossless compression Algo

(RLE) Run length encoding | Dictionary coding | Huffman coding | Arithmetic coding

1) Run length encoding

original Data :

A A A B B B B C D D D D D D E E E

3 4 1 6 3

compressed Data : 3A4B1C6D3E

Run length : It is the simplest method of removing redundancy. It can be used to compress data, made up of any combination of symbols.

This method replaces a repeated sequence known as run of the same symbol, using 2 entities

1) count 2) symbol itself

eg :

compression ratio = $\frac{17}{10} = 1.7$ ( $\frac{\text{total chars in original data}}{\text{total chars in compressed data}}$ )

original data : 000000000000 1 000 11 00000000

12 3 8

compressed data : 12 3 0 8

2 consecutive one's so '0' to represent it

Assume 1 digit = 4 units (bits)

4 digits = 16 units (bits)

compression ratio = $\frac{16}{23}$

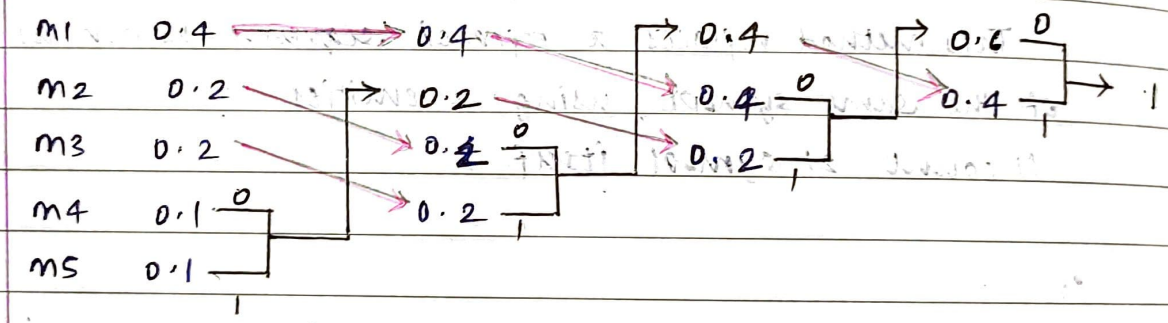
A modified version of RLE can be used if there are only 2 symbols in the data such as a binary pattern made up of 0's and 1's. In this case we count one of the symbols that occur between each occurance of the other symbol.

eg : previous page

2) Hoffman coding / encode

i)

| message | m1 | m2 | m3 | m4 | m5 |
|---|---|---|---|---|---|
| probability | 0.4 | 0.2 | 0.2 | 0.1 | 0.1 |



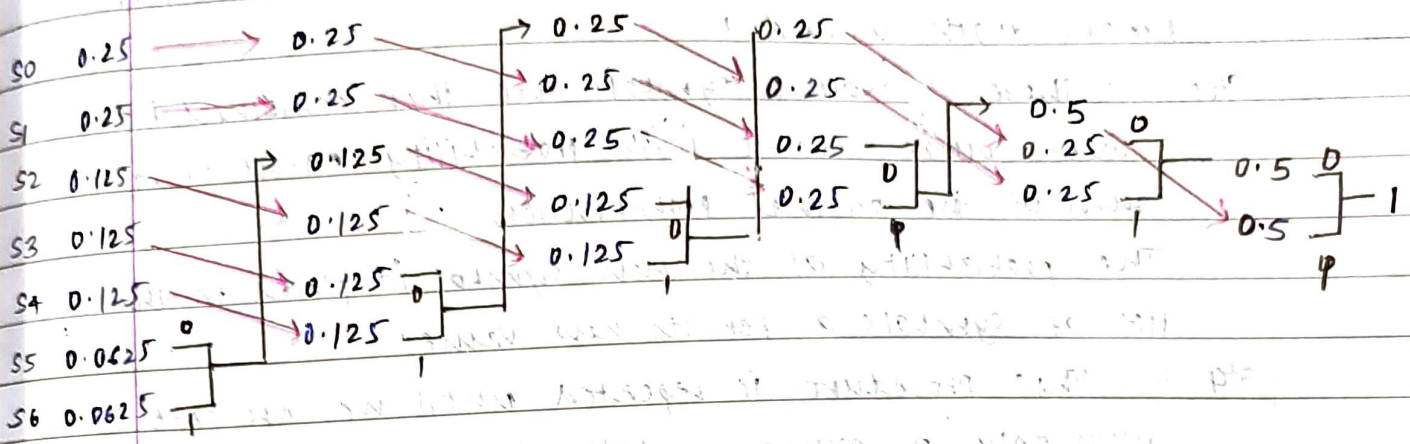

| message | m1 | m2 | m3 | m4 | m5 |
|---|---|---|---|---|---|
| probability | 0.4 | 0.2 | 0.2 | 0.1 | 0.1 |
| code | 00 | 10 | 11 | 010 | 111 |

Average codeword length = (0.4 x 2) + (0.2 x 2) + (0.2 x 2) + (0.1 x 3) + (0.1 x 3)
= 2.2 bits / symbol

sum of probability into bits

iii)

| msg | s0 | s1 | s2 | s3 | s4 | s5 | s6 |
|---|---|---|---|---|---|---|---|
| probability | 0.25 | 0.25 | 0.125 | 0.125 | 0.125 | 0.0625 | 0.0625 |




| msg | s0 | s1 | s2 | s3 | s4 | s5 | s6 |
|---|---|---|---|---|---|---|---|
| probability | 0.25 | 0.25 | 0.125 | 0.125 | 0.125 | 0.0625 | 0.0625 |
| code | 10 | 11 | 001 | 010 | 011 | 0000 | 0001 |

Average codeword length = (0.25 x 2) + (0.25 x 2) + (0.125 x 3) + (0.125 x 3) + (0.125 x 3) + (0.0625 x 4) + (0.0625 x 4)

= 2.625 bits/symbol

→ 1. It is a lossless data compression algorithm
2. The idea is to assign variable length codes to i/p characters.
3. Lengths of the assigned codes are based on the frequencies of the corresponding characters
4. The most frequent character gets the smallest code & the least frequent gets the largest code.

Page No.
Date.

Algorithm :

Step 1 : The source messages are arranged in the order of decreasing probability. The 2 source symbols having the lowest probability are assigned with binary digits 0 and 1

Step 2 : These 2 source symbols are then combined into a new source symbol with probability equal to the sum of the original probabilities.
The probability of the new symbol is placed in the list of symbols as per its new value.

Step 3 : This procedure is repeated until we are left with only 2 source symbols, for which a zero and one are assigned.

Step 4 : The code of each original source symbol is obtained by working backwards and tracing the sequence of 0's and 1's assigned to that symbol.

Dictionary based encoding (LZW) [lempel-ziv-welch]

eg : b c d e f g h b c m f b c

Initial dictionary (this is present with both sender and receiver)

| Index | Entry |
|---|---|
| 1 | b |
| 2 | c |
| 3 | d |
| 4 | e |
| 5 | f |
| 6 | g |
| 7 | h |
| 8 | m |
| . | . |

b c d e f g h b c m f b c

Encoding i/p 1 char, but index is that char + next char

| i/p | Entry | Index | o/p |
|---|---|---|---|
| b | bc | 9 | 1 |
| c | cd | 10 | 2 |
| d | de | 11 | 3 |
| e | ef | 12 | 4 |
| f | fg | 13 | 5 |
| g | gh | 14 | 6 |
| h | hb | 15 | 7 |
| bc | bcm | 16 | 9 |
| m | mf | 17 | 8 |
| f | fb | 18 | 5 |
| bc | – | – | 9 |

output sent to receiver = 1 2 3 4 5 6 7 9 8 5 9

Decoding concat with previous character

| i/p | Decoded symbol |
|---|---|
| 1 | b |
| 2 | c |
| 3 | d |
| 4 | e |
| 5 | f |
| 6 | g |
| 7 | h |
| 9 | bc |
| 8 | m |
| 5 | f |
| 9 | bc |

Dictionary

| i/p | entry |
|---|---|
| – | – |
| bc | 9 |
| cd | 10 |
| de | 11 |
| ef | 12 |
| fg | 13 |
| gh | 14 |
| hb | 15 |
| bcm | 16 |
| mf | 17 |
| fb | 18 |

entire previous string + 1st char of current string

2) BAABA BBBAA BBBBAA

Initial dictionary

| Index | Entry |
|---|---|
| 1 | B |
| 2 | A |

Encoding

| I/P | entry | index | O/P |
|---|---|---|---|
| B | BA | 3 | 1 |
| A | AA | 4 | 2 |
| A | AB | 5 | 2 |
| BA | BAB | 6 | 3 |
| B | BB | 7 | 1 |
| BB | BBA | 8 | 7 |
| AA | AAB | 9 | 4 |
| BB | BBB | 10 | 7 |
| BBA | BBAA | 11 | 8 |
| A | - | - | 2 |

output sent to receiver = 1 2 2 3 1 7 4 7 8 2

Page No.:
Date:

## Decoding

| i/p | Decoded symbol |
|---|---|
| 1 | B |
| 2 | A |
| 2 | A |
| 3 | BA |
| 1 | B |
| 7 | BB |
| 4 | AA |
| 7 | BB |
| 8 | BBA |
| 2 | A |

no entry of 7 so take previous decoded symbol (full) + 1st char of that symbol only

Dictionary

| i/p | entry |
|---|---|
| — | — |
| BA | 3 |
| AA | 4 |
| AB | 5 |
| BAB | 6 |
| BB | 7 |
| BBA | 8 |
| AAB | 9 |
| BBB | 10 |
| BBAA | 11 |

## Arithmetic coding

msg : bcab

A = { a, b, c, d }

P = { 0.4, 0.3, 0.2, 0.1 }

| Symbol | Probability | Interval |
|---|---|---|
| a | 0.4 | 0 - 0.4 |
| b | 0.3 | 0.4 - 0.7 |
| c | 0.2 | 0.7 - 0.9 |
| d | 0.1 | 0.9 - 1 |

of i/p symbol

from 1st table

* value = low + Range x high value of symbol
of previous table

step 1 : i/p = b (0.4 - 0.7)

| symbol | value | Interval |
|---|---|---|
| a | 0.4 + (0.3 X 0.4) = 0.52 | 0.4 - 0.52 |
| b | 0.4 + (0.3 X 0.7) = 0.61 | 0.52 - 0.61 |
| c | 0.4 + (0.3 X 0.9) = 0.67 | 0.61 - 0.67 |
| d | 0.4 + (0.3 X 1) = 0.7 | 0.67 - 0.7 |

step 2 : i/p = c (0.61 - 0.67)

| symbol | value | Interval |
|---|---|---|
| a | 0.61 + (0.06 X 0.4) = 0.634 | 0.61 - 0.634 |
| b | 0.61 + (0.06 X 0.7) = 0.652 | 0.634 - 0.652 |
| c | 0.61 + (0.06 X 0.9) = 0.664 | 0.652 - 0.664 |
| d | 0.61 + (0.06 X 1) = 0.676 | 0.664 - 0.676 |

step 3 : i/p = a (0.61 - 0.634)

| symbol | value | Interval |
|---|---|---|
| a | 0.61 + (0.024 X 0.4) = 0.6196 | 0.61 - 0.6196 |
| b | 0.61 + (0.024 X 0.7) = 0.6268 | 0.6196 - 0.6268 |
| c | 0.61 + (0.024 X 0.9) = 0.6316 | 0.6268 - 0.6316 |
| d | 0.61 + (0.024 X 1) = 0.634 | 0.6316 - 0.634 |

initial table & tag value is given to the receiver

Step 4 : b ( 0.6196 - 0.6268 )

| | symbol | value | interval |
|---|---|---|---|
| no need to find values in this table i.e for last symbol. This is because no i/p comes after this, so we donot need interval values | a | 0.6196 + (0.0072 × 0.4) = 0.62248 | |
| | b | 0.6196 + (0.0072 × 0.7) = 0.62464 | |
| | c | 0.6196 + (0.0072 × 0.9) = 0.62608 | |
| | d | 0.6196 + (0.0072 × 1) = 0.6268 | |

When i/p is equal to b, the final sequence interval is 0.6196 to 0.6268

The final tag value is $\frac{0.6196 + 0.6268}{2} = 0.6232$

Receiver's side [Decoding]

modified value = $\frac{\text{new value} - \text{low value of symbol}}{\text{range}}$

| | value | o/p symbol | low | high | range | modified value |
|---|---|---|---|---|---|---|
| tag value | 0.6232 | b | 0.4 | 0.7 | 0.3 | $\frac{0.6232 - 0.4}{0.3} = 0.744$ |
| | 0.744 | c | 0.7 | 0.9 | 0.2 | $\frac{0.744 - 0.7}{0.2} = 0.22$ |
| | 0.22 | a | 0.0 | 0.4 | 0.4 | $\frac{0.22 - 0.0}{0.4} = 0.55$ |
| | 0.55 | b | 0.7 | 0.9 | 0.3 | $\frac{0.55 - 0.4}{0.3} = 0.5$ |

Page No.:
Date:

2) msg = { BBAB * }
A = { A, B, * }
P = { 0.4, 0.5, 0.1 }

JPEG joint photographic experts group

0 - 255 (grayscale)
black white

one pixel value sudden change next pixel value
high frequency pixel

low frequency - same range

human eye cannot see if high frequency pixels are missing

Step 1) : Divide matrix into 8 rows & 8 columns

DCT - reverse DCT too

average of all values in matrix ⇒ gives $a^{th}$ row & $b^{th}$ col element of matrix

2) low frequency, high frequency values separated in DCT matrix

3) quantization (rounding off) - has quantization tables

Q100 table no
high quality with low compression

Q1 table no
poor quality with high compression

JPEG - lossy compression technique

DCT is designed to work only on values ranging to -128 to 127 matrix

so from 0 to 255 ⟹ convert to -128 to 127

so to do that, subtract 128 from each pixel entry

$D = TMT'$

T is matrix with values between 0 to 255

M —— " —— -128 to 127

T' is transpose of T

quantization - we lose pixels

$C = \text{round}\left(\frac{D}{Q}\right)$ — D: matrix after transformation; Q: quantization matrix

lose pixels while rounding off

C = [ low frequency data | high frequency data ]

Zigzag coding

pick up all low frequency data 1st & then high frequency data

apply any encoding technique and get final compressed data

JPEG - joint photos experts group

Page No.:
Date:

Decoding :

Fill values (arrange values) back using zigzag

↑

R = this matrix x Quantization matrix

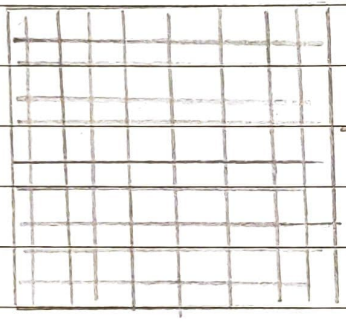
reverse DCT matrix transformation

$D' = T' M T$ $D' = N = T' R T$ given by sender

↓

Get this matrix back in range 0 - 255 by adding 128

8 x 8 matrix



Image

T to T'

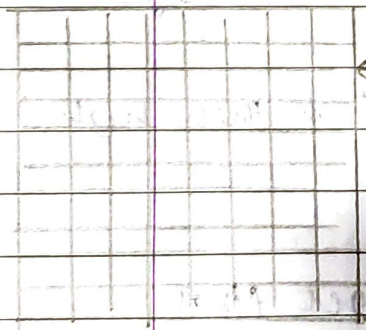
Transformer (lossless) → Quantizer (lossy) → Encoder (lossless)

Quantization table

| | |
|---|---|
| $Q_1$ | [ ] |
| $Q_2$ | |
| $Q_3$ | [ ] |

1100101....1

8 x 8



Image

Inverse transform $T'RT$ ← Dequantizer ← Decoder

Quantization table

video compression:

motion pictures experts group (MPEG)

I , P , B

I — intra coded frame (independent)

B — bidirectional

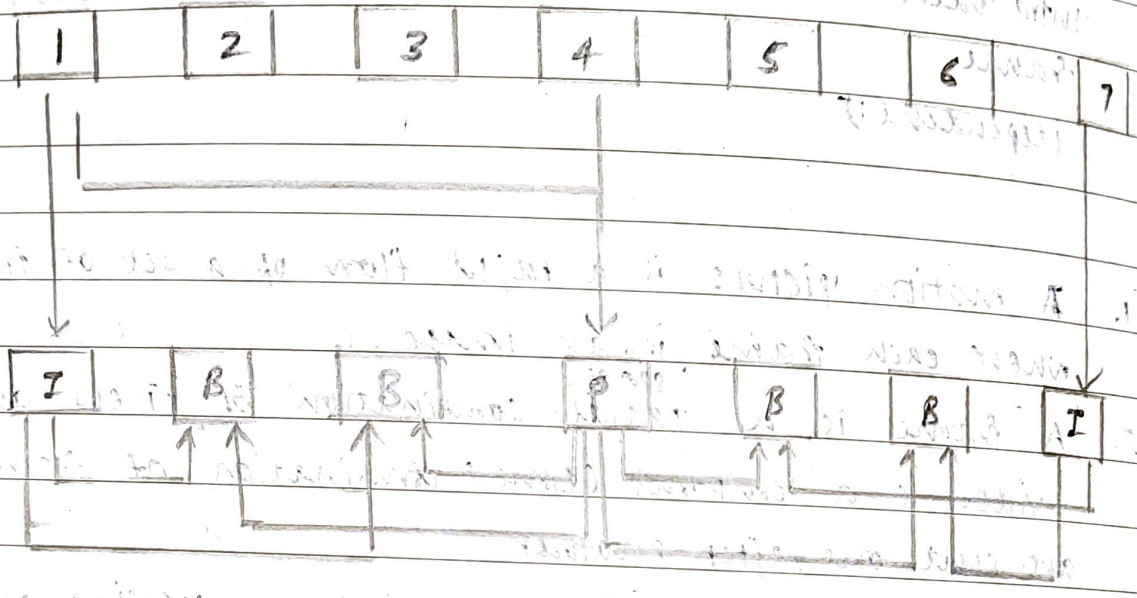
1. A motion picture is a rapid flow of a set of frames, where each frame is an image.
2. A frame is a spatial (space) combination of pixels and a video is a temporal (time) combination of frames, that are sent one after another.
3. compressing video means spatially compressing each frame and temporally compressing a set of frames.
4. The spatial compression of each frame is done with JPEG
5. In temporal compression, redundant frames are removed

eg: when we watch TV, we receive 50 frames/sec However most of the consecutive frames are almost same. To temporarily compress the data, MPEG first divides the set of frames into 3 categories:
I frame, P frame and B frame

i) I frame (intra coded frame): It is an independent frame that is not related to any other frame. They are present at regular intervals. They cannot be constructed from any other frame.

ii) P frame (Predicted frame): It is related to the preceeding I frame or P frame.
It contains only the changes from the preceeding frames. P frames carry very less information.

iii) B frames (Bidirectional frame) : It is related to the preceding and following I frame or P frame. B frame is never related to another B frame.



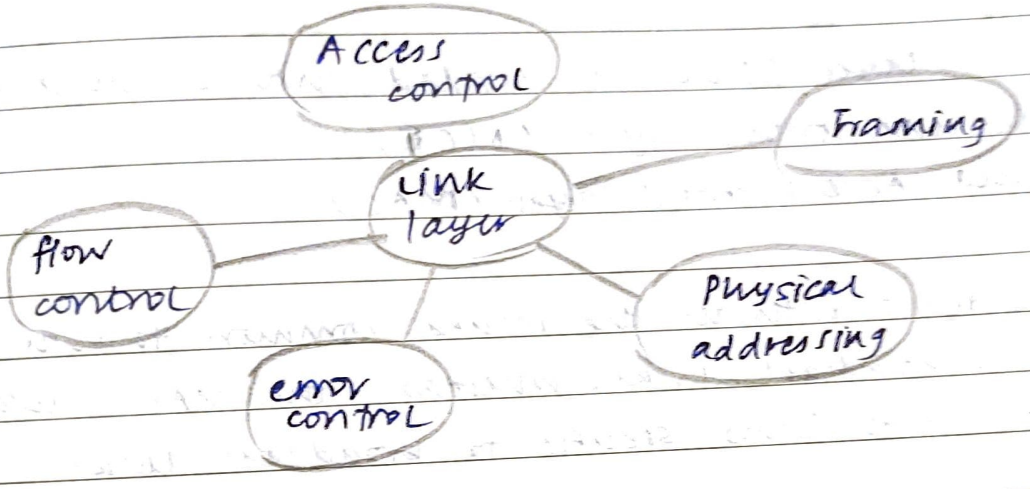

Transmit : I P B B I B B

Display : I B B P B B I

# Data link layer

It is the 2nd layer of the OSI reference model. communication at application, transport & network layer is end to end, whereas, communication at the data link layer is node to node.

Functionalities of data link layer:




Protocol data unit: | Header | Packet | Trailer |

Framing:

1. DLL receives data from the network layer & divides it into manageable units called frames
2. The format of the frame is: It is called as protocol data unit

Physical addressing

1. It provides the addressing into by adding header to each frame. Thus physical addresses of the source & destination machine are added to each frame.

Flow control

It provides flow control mechanism to ensure that sender is not sending the data at the receiver can not process

Error control

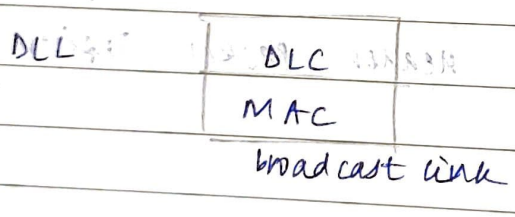
It provides error control mechanism to detect & retransmit damaged, duplicate or lost frames

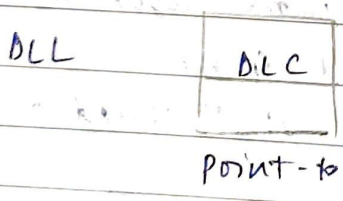
Access control

It provides access control when 2 or more devices are attached to the same link; DLL determines which device has control over the link at any given point.

→ Data link layer (DLL) is divided into 2 sublayers
1) Data link control layer (DLC)
2) media Access control layer (MAC)

The DLC deals with all the issues common to both point to point & broadcast links, whereas, the MAC sublayer deals only with issues specific to broadcast links.




DLC sublink: DLC functions include framing, flow & error control and error detection & correction.

Page No.:
Date:

# Flow and error control

## Error detection techniques:

### 1) cyclic redundancy check (CRC)

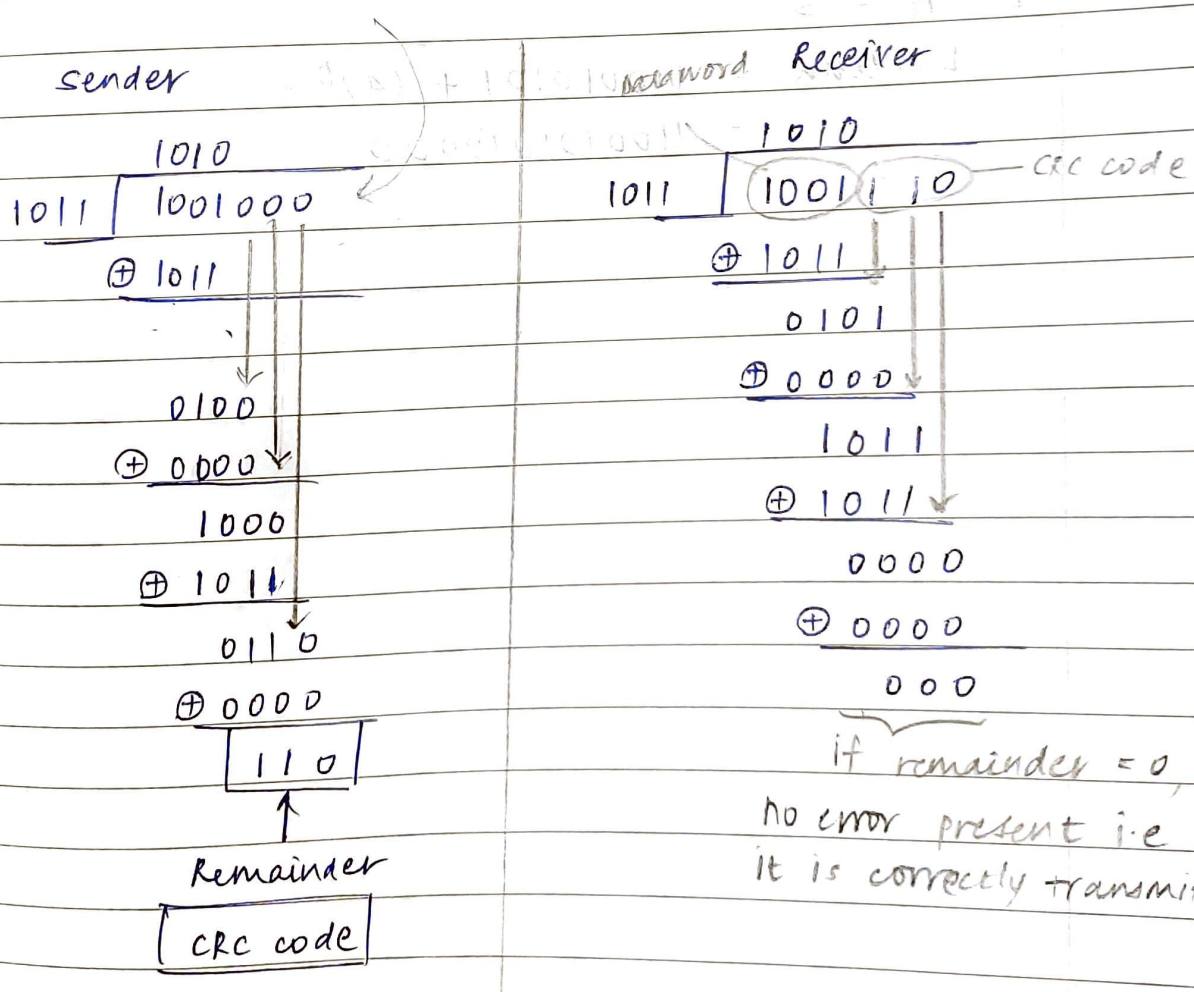
Q. Generate CRC code for the dataword 1001 and divisor 1011

Solution:

1) no of bits in divisor ie $n = 4$

Dividend = dataword + (n-1) 0's

= 1001 000



code word generation

In CRC the required code word is obtained by writing dataword followed by the remainder i.e 1001110

If the remainder at the receiver end is zero, then the received ~~text~~ code is error free, and is accepted by the receiver.

A non zero remainder indicates presence of error and hence corresponding code word is rejected by the receiver.

2) Dataword : 110010101
Divisor : 10101

i) n = 5
Dividend = 110010101 + (4)0's
= 110010101 0000

Page No.
Date

sender

```
          11111011
10101 | 11001010100000
       ⊕ 10101
         11000
       ⊕ 10101
         11011
       ⊕ 10101
         11100
       ⊕ 10101
         10011
       ⊕ 10101
         01100
       ⊕ 00000
         11000
       ⊕ 10101
         11010
       ⊕ 10101
         11110
       ⊕ 10101
         1011
```

remainder

CRC code

Receiver

```
          11111011
10101 | 11001010111011
       ⊕ 10101
         11000
       ⊕ 10101
         11011
       ⊕ 10101
         11100
       ⊕ 10101
         10011
       ⊕ 10101
         01101
       ⊕ 00000
         11010
       ⊕ 10101
         11111
       ⊕ 10101
         10101
       ⊕ 10101
         00000
```

if Q ⇒ $x^4 + x + 1$

$1x^4 + 0x^3 + 0x^2 + 1x + 1$

10011

Framing :

1. The DLL needs to pack bits into frame so that each frame is distinguishable from another.
2. Framing separates a message from one source to a destination by adding a sender address and a destination address
3. Frames can be of 2 types
   i) Fixed sized
   ii) variable sized

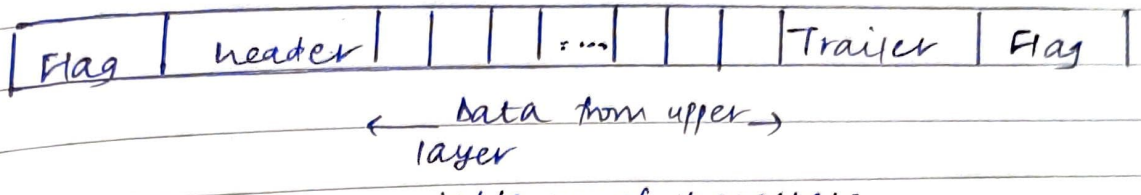
i) In Fixed sized framing there is no need for defining the boundaries of the frame, the size itself can be used as the delimited

ii) variable sized framing - In this we need to define a way that indicates the end of frame and the start of the next.

2 approaches are used for this purpose :
i) character oriented approach (Byte oriented)
ii) bit oriented approach

i) character oriented framing
1. In this framing, data to be carried are 8 bit ASCII characters.
2. The header normally carries the source & destination addresses and other control information; and the trailer carries error detection bits
3. To separate one frame from the next, 8 bit / 1 byte flag is added at the beginning & end of the frame.
4. The flag is composed of a special character that

signal the start & end of the frame.




Problem with the approach:

1) In sending info such as graphs, audio, video, any pattern used for flag could also be a part of the information.

2) If this ~~is the~~ case, if the receiver encounters these patterns in the middle of the program it will think it has reached the end of the data

Solutions:

Byte stuffing strategy

In byte stuffing, a special byte is added to the data section of the frame when there is a character with the same pattern as the flag.

This byte is usually called as an escape character 'ESC' and has a predefined bit pattern

Whenever the receiver encounters this escape char, it removes it from the data section & treats the next char as data

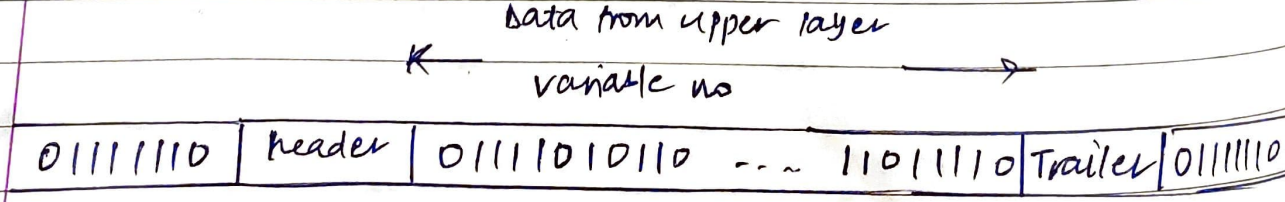
⇒ flag fixed.

Page No.:
Date:
YOUVA

bit ~~st~~ oriented framing

In bit oriented framing the data section of a frame is a sequence of bits to be interpreted by the upper layer.

However in addition to the header, we need a delimiter to separate one frame from the other. most protocols use a special 8 bit pattern flag as the delimiter to define the beginning and end of the frame (01111110) zero, six 1's, zero



Problem

If the flag pattern appears in the data we need to inform the receiver that it is not the end of the frame

Solution :

Bit stuffing

To avoid the above problem we stuff one single bit to prevent the ~~problem from~~ pattern from looking like a flag

In bit stuffing if a zero and 5 consecutive 1's are encountered, an extra zero is added.

This extra stuffed bit is eventually removed from the data by the receiver.