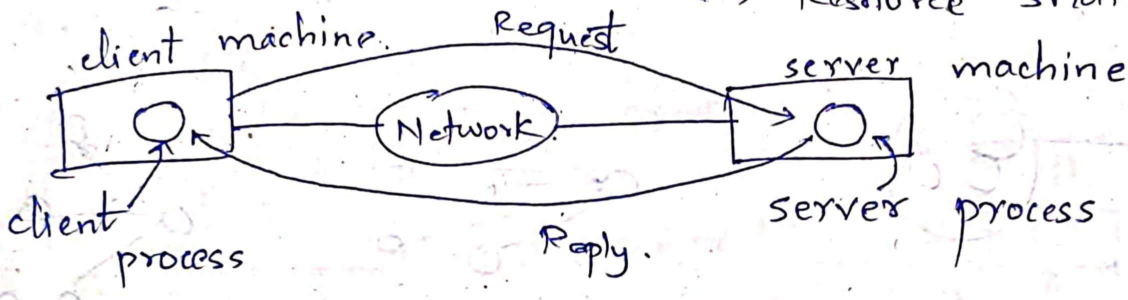
What is $\boxed{\text{network?}}$ → interconnection of devices.

Connected to network → $\boxed{\text{end systems (or) hosts.}}$

Digital data → parts / fragments
↓
$\boxed{\text{packets}}$

Uses of CNN : → Business Applications
→ Reshorce sharing, etc.




Network Essentials:

→ Modem (Modulator + Demodulator) :
Hardware device that converts data into a suitable format for transfer between two computers.

→ Ethernet: system for connecting computers to form a LAN.

→ Router: A device which forwards data packets / $\boxed{\text{fragments}}$ along networks.

- Connected to atleast 2 networks, comonly 2 LAN's

→ Repeaters: Regenerate, replicate a signal

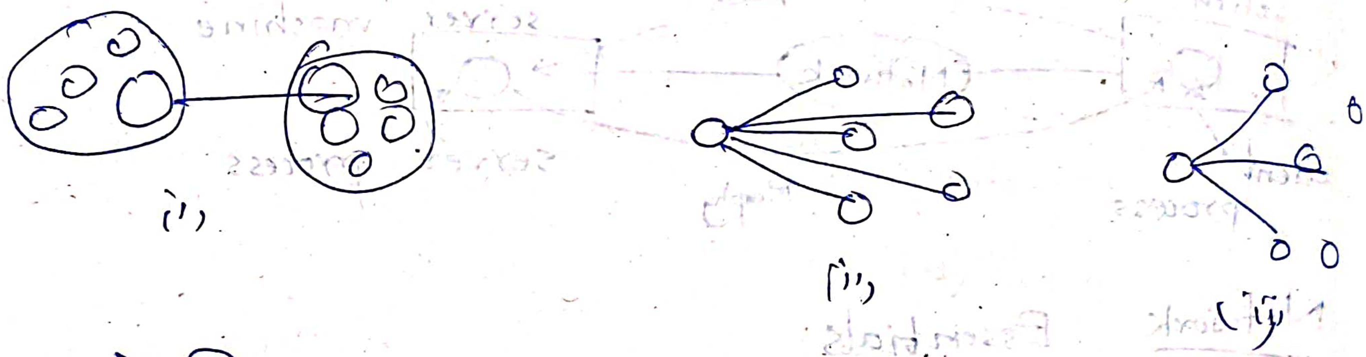
# Classification of Networks:

① Transmission Technology:

(i) Unicasting: Exactly one sender & one receiver

(ii) Broadcasting. To all hosts

(iii) Multicasting: To subset of hosts.




→ Personal Area Networks: (PAN's)

→ Local Area Network: (LAN's)

→ Metropolitan area network: (MAN's)

→ Wide Area Network: (WAN's) (WWW)

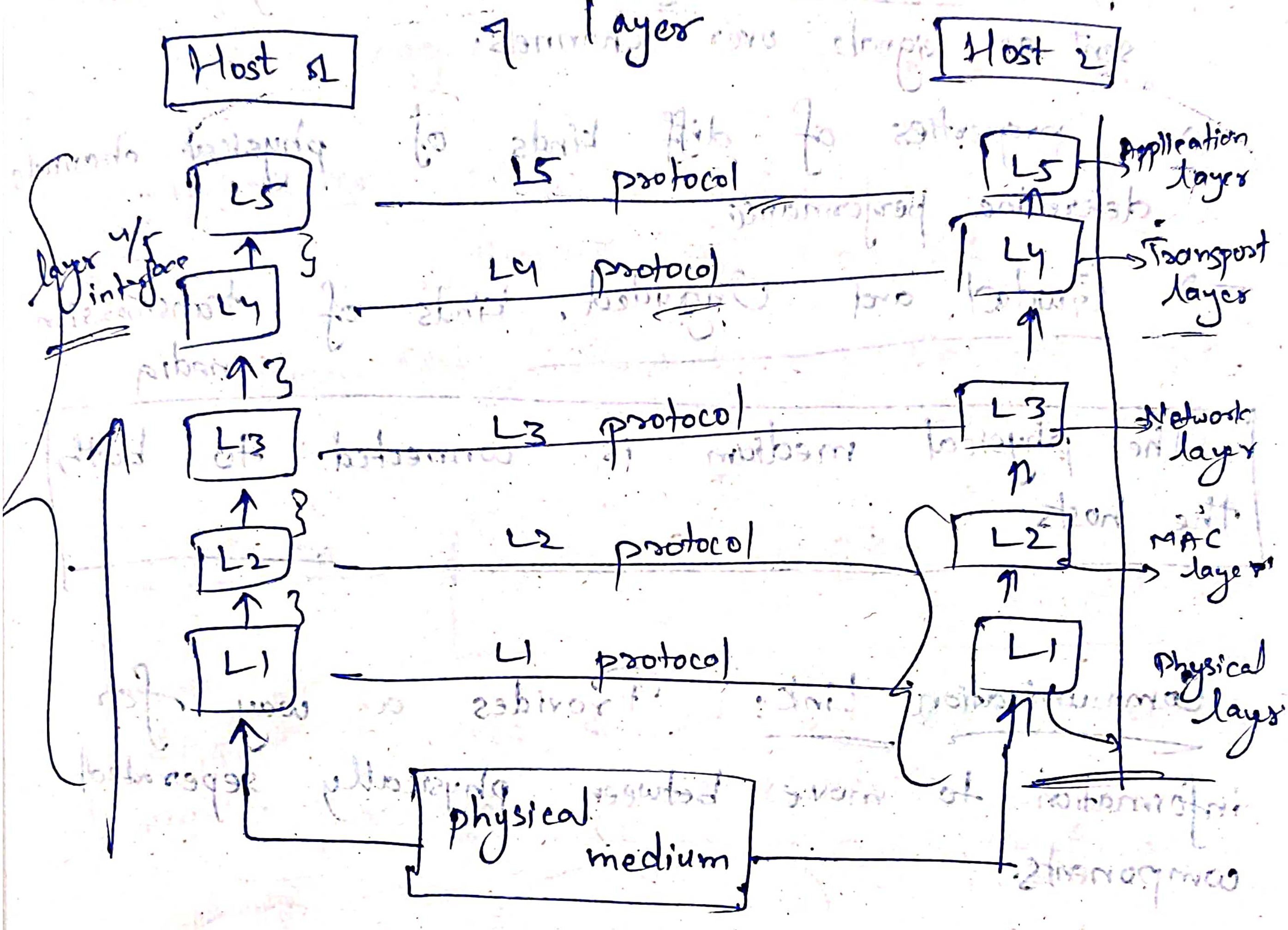
order

Network Software: (Brain behind internet)

Protocol: Agreement between communicating parties on how communication is proceeded.

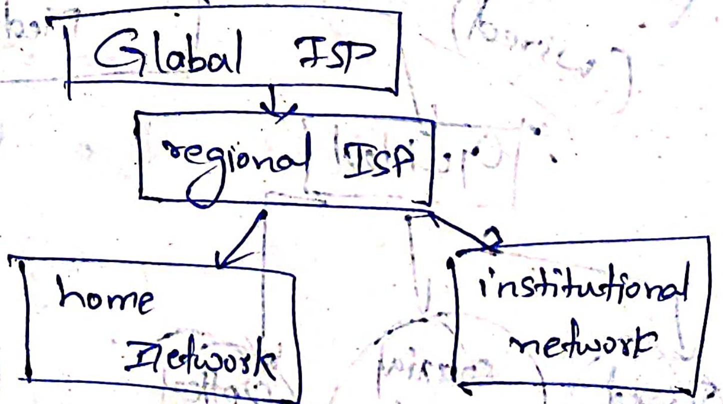
→ Violation makes it difficult

To organise functionality, it is stacked as 7 layers.




ISP - Internet Service Provider

→ Heirarchial order:



12-01-2022

Multiplexing : Technique which combines multiple signal into one signal. → Transmission over a communication channel

↳ large Bandwidth is saved, cost reduced, circuit complexity reduced and multiple signals can be sent over a single channel.

Analog : f division & wavelength division multiplexing. multiplexing
↓
continuous

Digital : Time division multiplexing
↓
discontinuous

FDM (frequency division multiplexing):
↳ TV and Radio.

→ Bandwidth of communication channel should be greater than combined bandwidth of individual signals.

→ Divides the bandwidth → logical sub channels and each is seperated by an unused bandwidth called guard band.

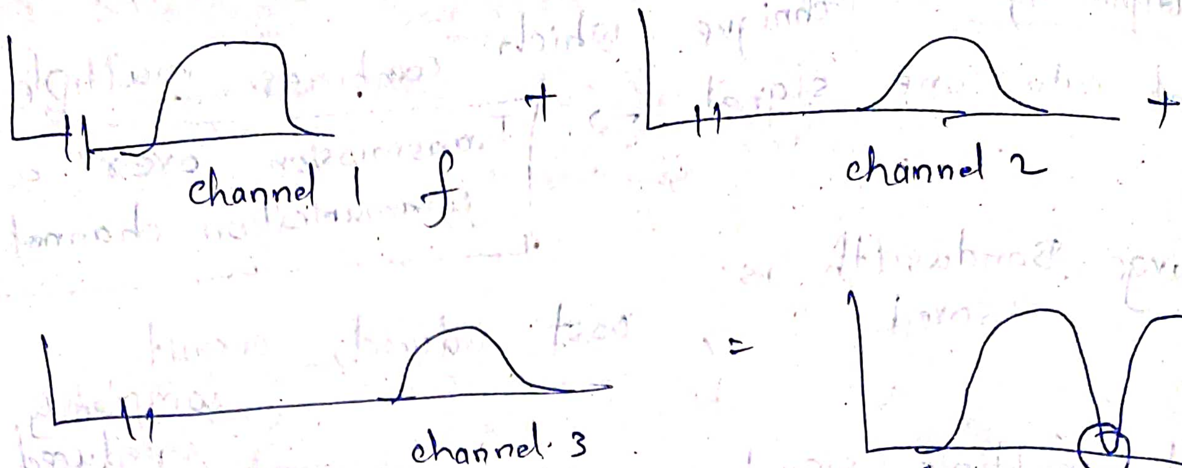
↳ A narrow frequency range which seperates two signal frequencies

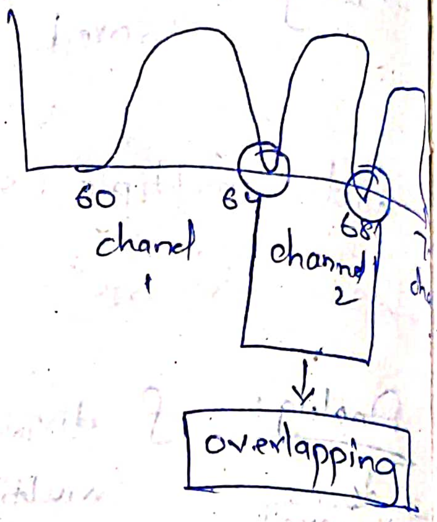
↳ Used to seperate 2 users and prevent overlapping

Ex: 4 GHz for 100 people

One = $\frac{4 \times 10^8}{100}$ = 40 Hz

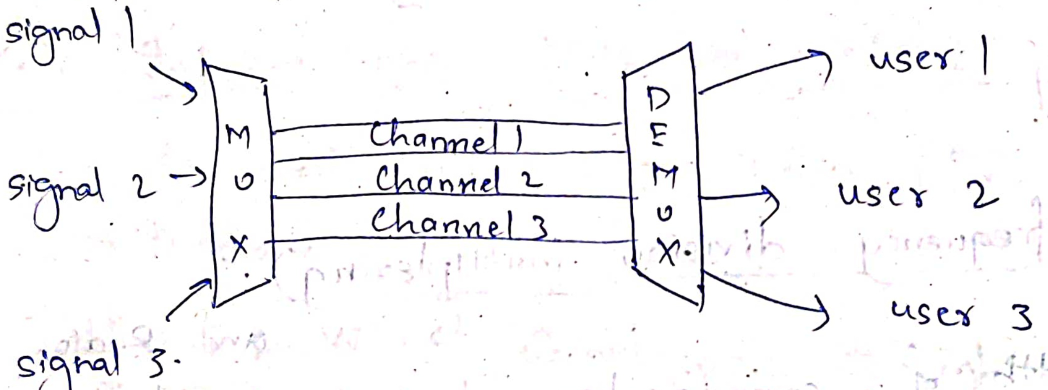
but only (0-35 Hz) (35 Hz - 40 Hz) guard band.




without guard band, overlapping happens.

Schematic Diagram:




Advantages:
- → transmit multiple signals
- → FDM, demodulation process is easy.
- → Doesn't need synchronisation.

Disadvantages: Needs a large bandwidth communication channel.

Applications:
- → FM, AM.
- → used in first gen cellular phone.
- → TV Broadcasting

## WDM (Wavelength Division Multiplexing)

→ A technique that increases the bandwidth of a communication channel (optical fiber) by simultaneously allowing multiple optical signals.

→ Working principle of WDM is similar to FDM but here, we use optical signals instead of electric signals.

Schematic Diagram (similar to FDM)

## 2 kinds of WDM Techniques: (WDM)

→ Dense WDM (longer distance)

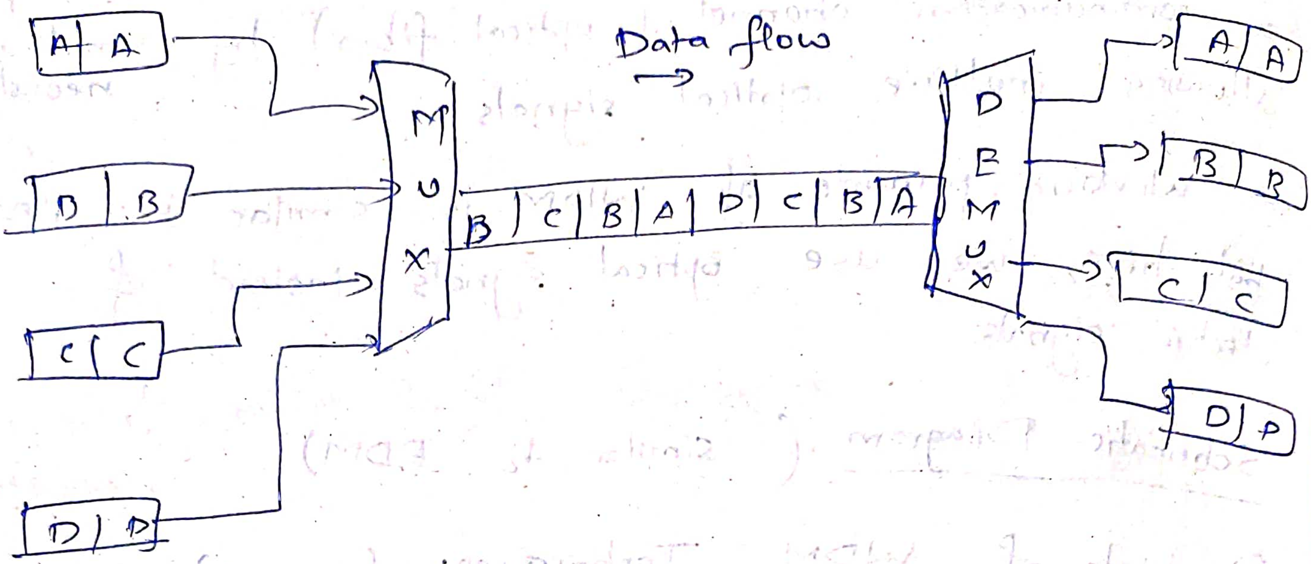
## Time Division multiplexing: (TDM)

→ Multiple signals are combined and transmitted one after other on same channel

↳ Assigning a time slot strictly for one user

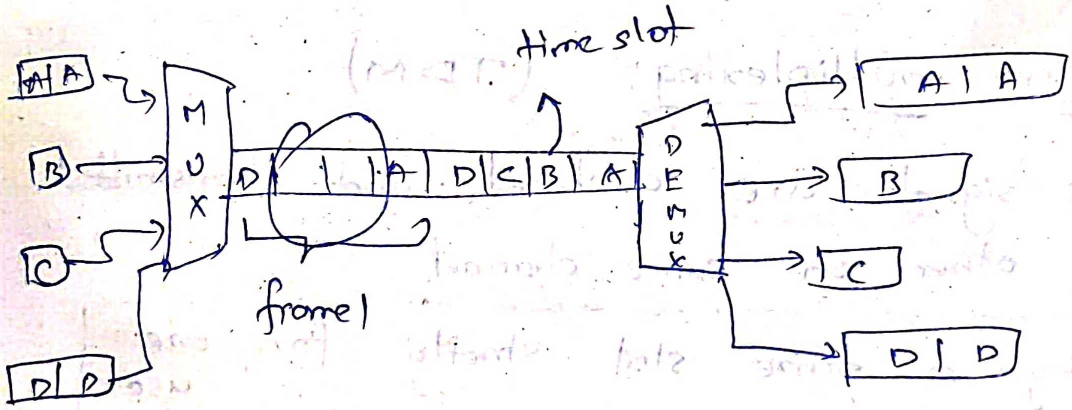
→ All signals operate with same frequency but are transmitted at different times.

## Schematic Diagram




The users needs to synchronised (so that one can be transmitted and then another one)

2 types:
- Synchronous TDM (fixed time slot)
- Asynchronous TDM (Not fixed slot time)

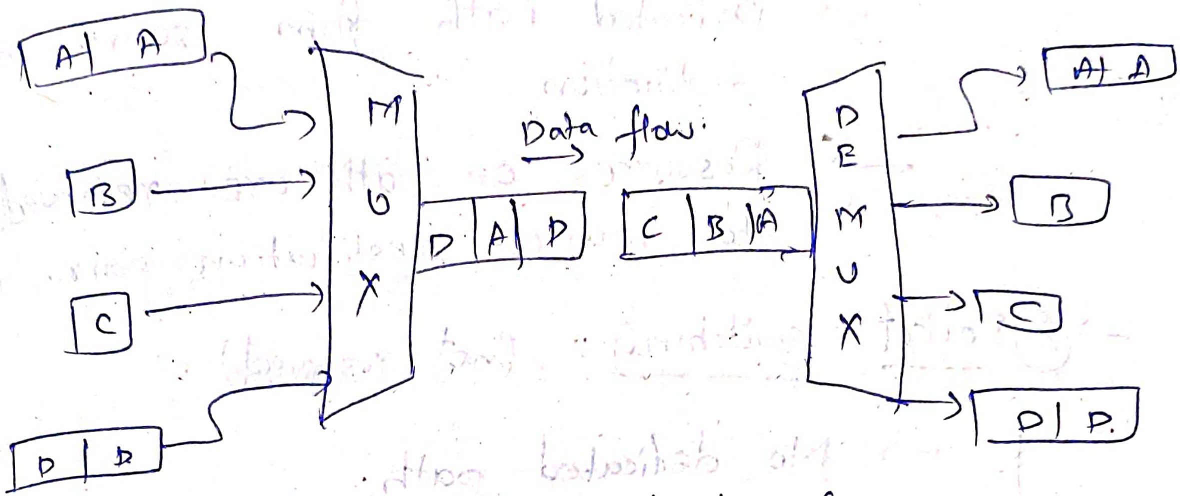



(synchronous TDM)

$\boxed{\text{number of time slot = number of transmitters.}}$

## Asynchronous

$\boxed{\text{time slots} \neq \text{number of transmitter}}$




**Advantages:**
→ full bandwidth for one user at one time.

→ TDM is flexible than FDM

→ crosstalk problem is less.

**Disadvantages:** → synchronisation required.

$\boxed{\text{CDMA : Code Division Multiple Access}}$

↳ Our information given → We recieve
Others → Dot product is zero. } orthoganal codes.

13-01-2022

End systems connected by :

→ ① Circuit switching : (reserved)

→ Dedicated path from source to destination

→ Resources on path are reserved to source - destination pair.

→ ② Packet switching : (not reserved)

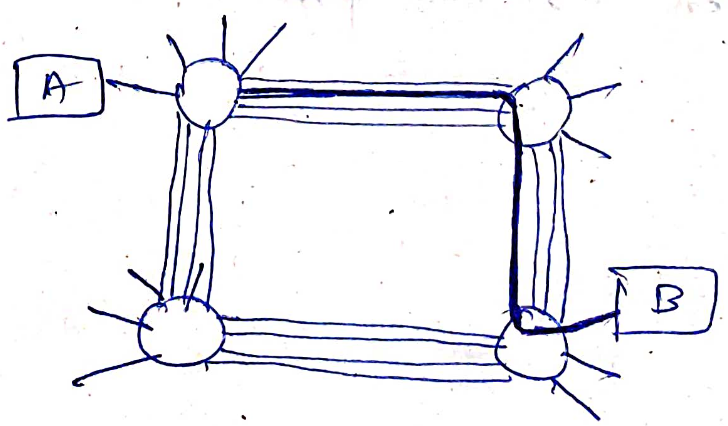
→ No dedicated path.

→ A switch / router forwards packets to another router / destination on the path.

→ Any message can take any path from the mesh of network

→ Gives the best of network path available. but not guaranteed.

① Circuit switch network




4 circuits / paths available.

The connection between source and destination is called as circuit.

→ Resources such as bandwidth, buffers are blocked for communication on that circuit.

→ Links are finite, hence very less users at once (Disadvantage).

Ex: Older Telephone Networks.
↳ Gives the best network (less delay) possible. (Advantage)

Ex: A circuit switch network transfer 640,000 bits between post A and post B. There are other links. It is following TDM which has 24 slots. Maximum supported speed is 1.536 MBPS. and time to establish a network is 500 ms. Calculate the total time taken

A) Total time taken = Establishment time + Transfer time.

Total slots = 24

1 slot speed = $\frac{1.536}{24}$ = $\boxed{\text{1 circuit speed}}$

= 64 KBps (approx)

Total time = $\frac{\overset{10}{64\cancel{0000}}}{64\cancel{000}}$ = 10 s.

$\boxed{\text{Total time} = 10.5\text{ s}}$

② Packet switch network :

We won't share partial amount of packet.
There is a store and ~~share~~ forward kind of system

→ Before receiving the entire amount of packet, it is stored in buffer and queue.

→ Only after receiving entire packet, it forwards

Each packet is having "L" bits and link capacity is r bits per second. How much time will it take

Ⓐ $\boxed{\frac{L}{r} \text{ seconds}}$ to store completely in buffer

and $\frac{L}{r}$ from buffer to destination.

$\boxed{\text{Total } \frac{2L}{r} \text{ from source to destination}}$

first packet : $\frac{2L}{r}$ and subsequent ones follow kind of pipelining

Why can't bit by bit ?

Ⓐ Because when transferred bit by bit, the router or any other intermediate network component doesn't know the receiver nor the sender.

The reference to what destination the packet is sending to is. IP Address.

That IP address should be embedded into packet

The input queue will be filled faster.
output queue is a lot slower (6 times less than input queue)

So there is a delay called queuing delay.

Note: Internet is a packet switching network.

17-01-2022

## Stastical Multiplexing:

Ex! Users = 1 Mbps link.

→ user is active for 10% of time & user generate 100 kbps

- Circuit switching means 100kbps reserved for each user all the time → can support 10 users simultaneously

$$\boxed{P(x \geq 35) = \sum_{i=11}^{35} {}^{35}C_i \, (0.1)^i \, (0.9)^{35-i}}$$

↳ probability that 11 or more users are active simultaneously in packet switching ⇒ $\boxed{0.0004}$

⇒ We can support 35 users with $\boxed{P = 0.0004}$

→ Circuit switching is used for time critical applications.

↳ Videos, audio calls.

→ packet switching problems → Queing delays
→ packet loss
↳ not suitable for delay constrained applications.

## Layered Network Architecture:

Network organisation is a complicated task.

↳ Divide and conquer → academic.
→ finance
→ administration.

- A layer provides a service (set of actions) to the immediate higher level.
- Each layer can be analysed and tested independently and newer technologies can be adopted easily.

## Open System inter connection (OSI) model:

7 layer:

Application layer
presentation layer
session layer
Transport layer
Network layer
Data link layer
Physical layer

heirarchy

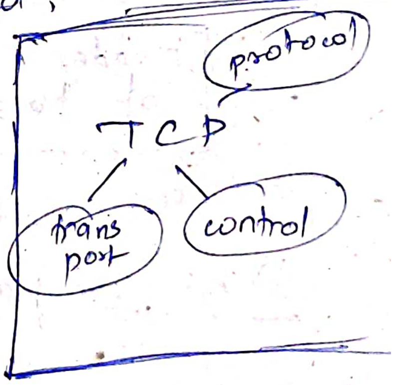
Most import

1. Application Layers:
   - user programs that work in hand

Ex: FTP, SMTP are protocols
(file transfer) (simple mail)

2. presentation layer:
   - syntax and information transmitted
   - translation
   - Encoding data : Compression / convertion; encryption and decryption.

3. Session layer:
   - Allows to establish a link.
   - Dialogue control : Bidirectional or, unidirectional
   - Token management : Both sides do not attempt same operation. It gives token to perform such actions
   - Synchronisation : Pausing and resuming a download.




4. Transport Layer:
   - Connection oriented services
     - $\boxed{\text{flow control}}$
     - guaranteed delivery.
   - Ensures data delivery is
     - error free
     - in sequence
     - no loss, duplication and corruption of packets

(5) Network layer:

- Interface between host and network
- $\boxed{\text{Routing}}$
- Congestion and Deadlock
- Internetworking.

(6) Data-link and physical layer:

(MAC)

Medium Access control.

→ Takes packet from network and moves to next router

→ error free delivery
error correction

physical layer: Electric signals with created
→ signal modulation.

18/1

Internet protocol stack:

| Header of transport layer + payload = segment |
|---|

Delays in packet switching network

→ processing, queueing, transmission, propogation. delay.

| Nodal delay = sum of all delays |
|---|

(or)

$\boxed{\text{end to end delay}}$ → delay from source to destination:

processing delay : Time required to examine the packet header. (unwrap)

↳ Determines where to direct the packet.

↳ Check for errors.

↳ In microseconds.

Queuing delay : If a router is busy in processing and transmitting a packet it will wait in queue.

No queing delay if router is idle.

Order is microseconds to milliseconds

generally a random variable

Transmission delay: delay in pushing ~~a~~ packet into the link.

length of packet = $l$ bits. and transmission rate = $r$

$$\boxed{\text{Transmission delay} = \frac{l}{R}}$$ → microseconds to milliseconds

propogation delay: Time required to propogate from one end of the link to the other end.

↳ depends on. physical link between the routers.

$2\times10^8 - 3\times10^8$ m/s. (range)

→ depends on distance

$$\boxed{\text{delay} = \frac{d}{s}}$$

## Traffic intensity:

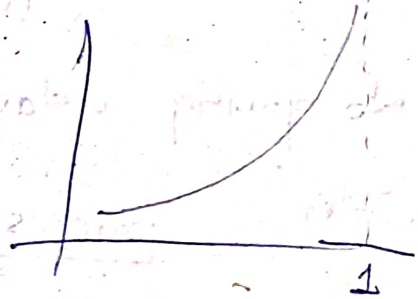
→ Queing delays are random
→ indication of queuing delay

Let "a" be the average number of packets arriving at a queue.

Each packet length = L , transmission rate = R

$$\boxed{\text{Traffic intensity} = \frac{aL}{R}}$$

If TI > 1 , queue length increases to ∞




If traffic intensity close to 1, there will be significant queue delay.

**Throughput**: Rate of successful message delivers

Instaneous throughput is at rate at which Host B is receiving data.

Suppose it takes T seconds to transfer F bits

Minimum of available limit is called throughput

$$\min(R_1, \ldots R_n) = \text{Throughput}$$

19/1 : ① N packets simultaneously transmitted (or) ② queued. packet length = $L$ Transmission rate = $R$.

What is average queuing delay for N packets.

A) $1^{st}$ packet delay = 0

$2^{nd}$ packet delay = $\frac{l}{r}$.

$\frac{2l}{r}$

⋮

$\therefore$ Average delay = $\frac{\text{sum of all packet delay}}{\text{no of packets}}$

$$= \frac{\frac{\cancel{N}(N-1)L}{2R}}{\cancel{N}} \Rightarrow \left(\frac{N-1}{\cancel{}}\right)\frac{L}{2R}.$$

② (i) path has 3 links $R_1 = 500$ kbps

$R_2 = 2$ Mbps.

$R_3 = 1$ Mbps.

Throughput = ?

A. $\boxed{\text{Throughput} = 500 \text{ kbps}}$

(ii) File size = 4 ~~billion~~ million bytes,

$$\text{Time taken} = \frac{\text{File size}}{\text{Throughput}} = \frac{4 \times 10^{\cancel{6}}, \cancel{00}, \cancel{000} \cancel{\times}}{\cancel{5}\cancel{00} \times 10\cancel{00}}$$

$$= 8\text{ s}$$

(3) packet length = 1000 bytes.

propogation speed = $2.5 \times 10^8$ m/s

length of link = 2500 km.

Transmission rate = 2 Mbps.

propogation delay = $\frac{\text{distance}}{\text{speed.}}$ = $\frac{\cancel{2500 \times 10^3}}{\cancel{2.5 \times 10^8}}$

$10^3$

$\cancel{25} \times 10^2$

= 0.01 s

Note : propogation delay = $d_p = \frac{m}{s}$

transmission delay = $d_t = \frac{L}{R}$

$$\boxed{\text{end to end delay} = \frac{L}{R} + \frac{m}{s} = d_{e2e}}$$

If $t=0$ at start, then at $t = d_{trans}$, the last bit just ends A, just out of A placed on link

at $t = d_{trans}$ if $d_{prop} > d_{trans}$, the first bit is on the link, if $d_{prop} < d_{trans}$, the first bit will outside of A, closer to B.

Ex) $s = 2.5 \times 10^8$ m/s $L = 120$ bits $R = 56$ kbps.

distance = ? $d_{prop} = d_{transmit}$ m = ?

$\frac{m}{s} = \frac{L}{R}$ $m = \frac{\cancel{120}\ 60}{\cancel{56} \times \cancel{10^3}} \times 25 \times \cancel{10^7}$ 36 15 $10^4$

28 14 7

$= \frac{1.1\ \cancel{15} \times 25 \times 10^7}{7}$

Ex: packet length = 1500 bytes. Transmission link = 2 Mbps.

no of packets in queue = 4. What is queuing delay of the packet which is about to enter?

(i) Transfer 4.5 packets needs to be done

$\Rightarrow$ $4 \times 1500 + 750$ bytes needs to be transmitted.

$\Rightarrow$ Queuing delay $= \dfrac{[4 \times 1500 + 750]}{2 \times 10^6}$

General expression : $\dfrac{[(L-x) + nL]}{R}$.

Traceroute :

→ runs in any internet host :

sends a packet to router 1 and router 1 sends back or transmit back packet and then calculate end to end delay.

21/1

Networks under attack

1. malicious stuff - malware

↳ deleting our files

↳ spyware : detects. security id, numbers, etc.

↳ compromised host can be part of organisation called as botnet.

self replicating : clicking a link , entering details

| | |
|---|---|
| virus : | requires user interaction |
| worms : | requires no user interaction |

## Denial of Service Attacks ( DoS attacks) :

↳ bring down a service
↳ renders a service unusable by legitimate users.

1. Vulnerability attack

bombarding with unnecessary packet like giving an ununderstable packet which can't be decoded

→ can stop / crash the host

2. Bandwidth flooding

sends a packets overflow and floods, so that actual users can't connect

3. Connection Flooding

using $\boxed{\text{TCP}}$ which is connection based communication (1st con & 2nd comm)

Affecting the host so that it will not accept the legitimate connection users.

## Distributed DoS (DDoS) :

leveteraging botnets with 1000's of hosts and that then does attack on single host

↳ Much harder to stop and attack.

## packet sniffers :

↳ can be wired (or) wireless

↳ an imposter passive reciever which can obtain a copy of every packet that is transmitted.

Sniffed packets can be analysed offline for sensitive information (personal private information, trade secrets, passwords)

WireShark is an example of packet sniffer.
↳ analyze layer by layer protocol and give us a report (open source)

→ packet sniffers are passive

do not inject packets into the channel ← passive → Difficult to test

→ One way to defense against it is cryptography

---

## Application Layer

Network Application management: Writing programs that run on different end systems will communicate with each other over the network.

(Example in slides)

Network core devices do not function at the application layer

Applications use the service of network (Transport layer)

Architecture of Application:

→ Client - Server architecture

→ peer-to-peer architecture

Application developers decides either of these 2

## Client - server Architecture :

**Server:** An end system that serves the requests from various hosts.
↳ always on.

**Client:** An end system that requests a server for content.
↳ can be ON-OFF or always ON.

**Examples:** e-mail, file transfer, etc.

## P2P Architecture :

→ End system has direct communication.
↳ peers.

**Example:** Skype, Torrents, etc.

| Advantages | Disadvantages |
| --- | --- |
| → Full distribution | → ISP friendly : assymmetric data traffic. |
| → self scalable. | → Less security |
| → Cost effective. | → Incentives : peers should share bandwidth |

2/1

Frame : $\boxed{\text{Header + payload}}$ in MAC Layer
(MAC)

Datagram : $\boxed{\text{Network Header + payload}}$ in Network Layer

Example problem :

① Optical fibre, core has $\mu_{core} = 1.7$ and $\mu_{cladding} = 1.2$

$\mu_{co}$ $\mu_{cl}$.

What is the speed of light inside the core?

$$\mu_{core} = \frac{\text{speed of light}}{\text{speed inside core}}$$

$$1.7 = \frac{3 \times 10^8}{s} \Rightarrow s = \frac{3 \times 10^8}{1.7}$$

$$\boxed{s_{core} = 1.76 \times 10^8} \qquad \boxed{s_{cladding} = 2.5 \times 10^8}$$

What is core-cladding interface angle? ⇒ critical angle.
so that total internal reflection happens.

$$\sin\theta = \frac{1.76}{2.5} = 0.704$$

$$\theta_c = \sin^{-1}(0.704)$$

but the interface angle $= 90 - \theta_c = \theta$.

(2) 10 sig with 4000H Multiplex on single channel

What is minimum bandwidth required for multiplex channel.

a) with guard band

b) without guardband

(A) without guardband = 10 sig × 4000 Hz

= 40000 Hz = 40 kHz

with guardband : no of bands required = 9

each guardband = 10% of 4000 Hz = 400 Hz

Total bandwidth = 40 KHz + 3600 Hz

= 40.36 kHz

---

(3) 4 channels — 2 channels = 200 kbps each

— 2 channels = 150 kbps each

(no synchronisation bits)

Assume 4 bits from 200 kbps & 3 bits from 150 kbps channel. 4 bits from 200

3 bits from 150 kbps.

Questions:

a) What is size of frame in bits? 14 bits in frame (sum of all bits)

b) What is frame rate? 50000 frames / second = f

c) What is duration of frame? $\frac{1}{50000}$ = 20 us

d) What is datarate of entire multiplexing?

50000 × 14 = 700,000

= 700 kbps

process communicating:

A program running within the end system.

Client process → running on a client
server process → running on a server.

actual communicators

→ sends and receives messages to and from transport layer through software interface ⇒ socket.

also called as Application programming Interface

through

called as API

* Client & server process are not in the same pc

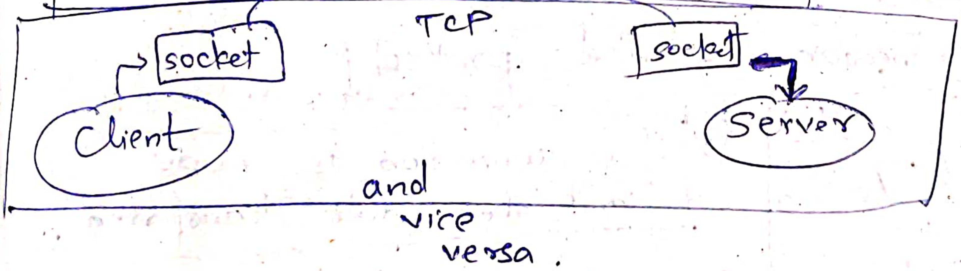
unlike OS process

exchange messages and communication

through

Hands to socket through transport layer

Transport layer




Application Layer

Application processes, sockets and underlying transport layer

Services of Transport layer (w.r.t Application layer)

→ Reliable data Transfer : Guaranteed Delivery.
(Application layer's role ends after giving it to socket)

→ Throughput : ensures that minimum throughput is maintained.

Ex: Telephony → minimum throughput is required for Transmission.

Video application service.

→ Timing : Maintaining maximum delay the application can tolerate.

Transport layer guarantees the packet be delivered in the given time.

→ Security : End point authentication, encryption and decryption

These are indirectly the requirements, conditions and criteria of application layer.

Transport protocols (protocols used in transport layer)

1. Transmission control protocol (TCP)

↳ connection oriented protocol

- handshaking
- full duplex connection
  - ↳ client to server and vice versa
- ↳ first connection is made and then data transferred
- rearrangement of packets in the desired way
  - ↳ orderlyness of packets

→ Reliable data transfer : no data corruption.

→ Maintains congestion control

- lowers the higher
- highers the lower

2. User Datagram protocol (UDP)

↳ connectionless protocol → no congestion control

no handshaking

unreliable data transfer

↓

no acknowledgement from the destination.

- TCP is complex to implement, time consuming and less data loss
- UDP is easier to implement, easier to connect, faster data transfer and can tolerate minute data losses.

Ex: Skype is using video service

↳ uses UDP → faster connection and data transfer

↓

can tolerate glitching (data loss)

Youtube uses TCP → slower than UDP

→ cannot tolerate data loss.

Addressing process: Helps you identify what process is directed to what destination. process for communication.

Identify Host by IP Address

Identify processes by port numbers

Ex: Email: 25, web server: 80

# Application Layer – Introduction

defines :

- types of messages exchanged : request and response based.
- syntax of various message types : fields in messages, how fields are delineated.
- semantics of the field : meaning of information in the field.
- Rules for determining : When and how a process sends and receive messages.

3/1

Web and HTTP → web's application layer protocol

uses client-server arch. with TCP ⇐ Hypertext transfer protocol

wb page : Document and contains of objects.

images | vidoes | applets.

Web server

- Client and server communicate by exchanging HTTP messages

in Web : Browser – Opera, Chrome, etc.

Uniform Resource Locator : URL

URL → object be addressed by URL

URL → contains object with path & hostname → hostname/path of object

## HTTP :

- HTTP client initiates a connection with HTTP server → Handshaking

↓

Establishing a firm connection. (using TCP)

- Once connection is established, client and server communicate through socket interface

through ↗ ↖ through

Client sends request and receives messages.

Server recieves request and send messages.

- client / server need not worry about packets. (does not have any control) after sending through socket.

**Stateless protocol:** Server sends requested file without storing state information of client

Ex: Requesting for a file two times and server still sends 2 of them.

## HTTP connection :

Let's say web page : 1 HTML and 10 pages.

2 ways :

- persistent : One TCP connection and send all files. Overhead : minimal.
- non persistent : Open TCP connection, → take the file, → give to client → close. Overhead : maximal

→ Opens 10 times connection and closes

## Non-persistent connection:

For each file. → This is the workflow/working mechanism. → ① HTTP client initiates a connection with server on port number 80

↓

can be serial or parallel.

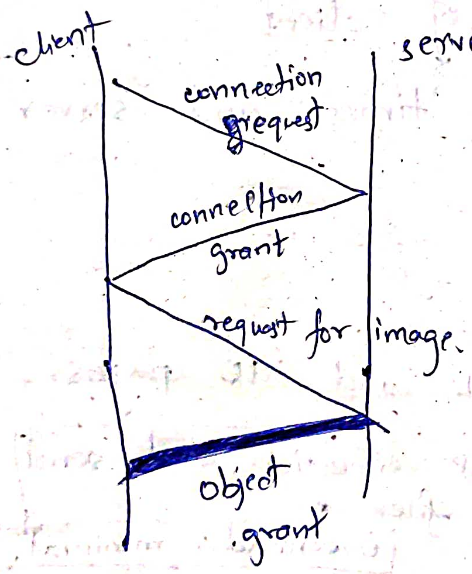
② Client sends a HTTP request and server receives request and send response → via socket interface

↓

③ HTTP server tells TCP to close the connection

## Round Trip Time: (RTT)

* (image in slides) *




Total time for 1 object receival =

1 RTT (for connection) +

1 RTT (for object)

$\boxed{\therefore \text{Total time for 1 object} = 2 \text{ RTT}}$

$\boxed{10 \text{ objects} = 20 \text{ RTT}}$

serial non-persistent connection.

**Note:** Default mode HTTP : persistent connection + pipelining persistent connection: → TCP closes after some idle period.

→ server leaves the connection after sending the HTTP response.

pipelining happens of all TCP's.

HTTP Request format:

- HTTP Request Message :

```
GET /somedir/page.html HTTP/1.1
     path                version
Host: www.iiits.ac.in
Connection: close.
User-agent: Mozilla/4.0
Accept-language: En.
```

- Methods: GET, PUT, POST, DELETE, etc.

* (diagram in slides) *.

1/2

HTTP Response:

```
HTTP/1.1 200 ok.
Connection: close.
Date: Sat, 07 Jul 2007 12:00:15 GMT
Server: Apache/1.3.0 (unix)
```

* diagram in slides *

Cookies :

Ex: visiting ebay (typically for not the first time)

first time : backend database store for some time and does backend processing.

- Required to not to give a response only, rather to remember/identify the user (Ex: Amazon) to enhance the recommendations.

- There will be given an unique id / number for amazon server for further purposes :

Cookies → identify a person

Cookies → show user relevant information.

- The unique number is shared with both Backend server and also client side.

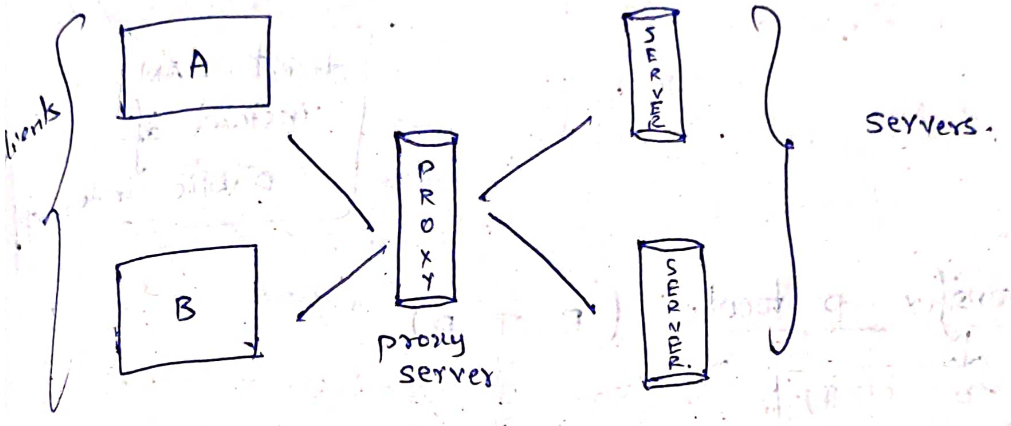
- Downside : Selling of user data and misuse

Cookies → Happens on HTTP Message only

Web Caching (cache) : concept of storing frequently used client memory.

↳ Optimising the request - response (of server) for 1000 of clients.

↳ also called as proxy servers.




Initiates a TCP connection (handshaking) with the client and proxy server and if we get a hit, response is given immediately, but if it's a miss, it then opens a TCP connection with the proxy server and the server (analogous to main memory)

Ex: Given avg object size = 1 Mbps

Link = 15 Mbps avg request rate = 15 objects per sec.

avg response time = 2 sec.

Traffic intensity = 1 (very huge delay)

$\boxed{\therefore \text{ Hence, proxy server is needed}}$

Simple fix : Increase link to 100 Mbps or so.

Traffic intensity reduces.

Another solution is web server / proxy

offloads original server's loads

saves time when hit

saves bandwidth

direct LAN instead of public internet

File Transfer protocol: (F T P)

similar to HTTP

client server architecture

TCP

2 TCP connections

TCP control connection

TCP data connection

what is control information?

user identification.

Change remote directory

commands to put and get files

FTP is said to control information. out-of-band while HTTP is inband

inband: sending in same TCP link

out-of-band: sending in different TCP links

* Commands and Replies in slides * (not important)

## Electronic mail (E-mail):

Asynchronous communication medium.

doesn't expect other person to be attentive.

Major comp:
- User agent
- mail server
- HTTP

tool used to communicate. Ex: Outlook

could be client

user uses this tool to push the message onto mail server

- Once handshake is completed between 2 user agents, then the mail goes to the mailbox of other user

No information yet!

- When he's online, he receiver from mail server 2

* Diagram in slides *

User $\xrightarrow{\text{sent}}$ Mailbox 1 $\xrightarrow[\text{TCP connection}]{\text{after}}$ Mailbox 2 $\xrightarrow{\text{retrieval}}$ user retrive

receives by mail Server

through user agent

## Sample Mail Transfer protocol (SMTP)

transfers messages from user's mail server to receipent's mail server

serves as both server and client

sender as client

receives as server

indicating client server architecture

Vice versa - true

SMTP is asynchronous coz:

- If receipt's mail server down, client reattempts to send n
- If n iterations later, if not received, it notifies the sender and dispose the mail

2/2

SMTP $\boxed{\text{asynchronous}}$ Sequence of Operations:

↳ push, rather than pull like HTTP

↳ End to End TCP connection

problems / Disadvantages.

- restricts the body to 7-bit ASCII
- only text messages
- Messages sometimes doesn't get placed in immediate mail server

Message Format: Header: To, From

Optional Header: subject

| SMTP | HTTP |
|---|---|
| • push | • pull |
| • 7 bit ASCII rule | • No restriction. |
| • Only Text | • can encapsulate each object in own HTTP message |

From B's mail server to mail box, we can't use push. (We need to pull the information)

Here comes mail access protocols (MAP)

- post office protocol V3 (POP3)
- Internet MAP IMAP
- HTTP

## POP3 :

POP3 3 Major phases

- Authorisation
  - user <username> and pass <password>
- Transaction
  - user agent sends commands & server responds with +OK or -ERR.
- Update

Download and delete

Download and keep

IMAP

Associates messages with a folder

Has commands for folder creation and pushing of messages

search for message command

has information about state

components of message can be retrieved.

What is a Domain Name? — .com, .org, .edu

www.iitm.in

Domain: in

www.iitm.ac.in

Domain: in
subDomain: ac

DNS (Domain Name Service): A mapping service which maps from Domain Name to IP Address.

ICANN: Internet Corporation for Assigned Names and Numbers

translation

Domain Names

Absolute (Ends with)

Relative (Interpreted based on content)

Application Layer protocol used by other applications

→ Client server architecture, uses UDP at transport layer

DNS adds an additional delay to the internet applications that use it

services of DNS :

- internet directory
- Host Aliasing
- Mail Server Aliasing
- Load distribution

Ex. IRCTC

problems :

- A single point of failure
- Huge Traffic Volume
- Distant centralized database → $\boxed{\text{huge delay}}$
- Maintananee.

3/2 :

$\boxed{\text{DNS}}$ functionality

- internet's directory service
- Host aliasing
- Mail server aliasing
- Local Distribution. (IRCTC)

$\boxed{\text{DNS work by : Recursive and Iterative Query}}$ → $\boxed{\text{Root; Top and authorative}}$

$\boxed{\text{Local DNS server : Gives recent search items}}$ ~ $\boxed{\text{to web caching}}$

↳ If located, we need not go to top level and root DNS and authorative DNS.

* Diagram in slides A

Request to root, root gives IP Address to next heirarchy order,

$\boxed{\text{LOCAL DNS}}$ have IP address of above heirarchial order DNS's.

→ Local ISP can have local DNS

→ Host will query the local DNS and that takes to above order

→ Cache DNS replies.

**DNS Resource Records:**

- store resource records.
- tuple format {Name, value, Type, TTL}
- Time to live.

**Types in RR:**

**Type A.**
Name: HostName
Value: IP Address.

**Type NS.**
Name: Domain.
value: Hostname of authorative server.

**Type CNAME**

**Type MX**

Applications of P2P Architecture → no server

- File Distribution → one to many
- Database distribution → one to many
- Internet Telephony → Ex: Skype

## File Distribution:

- Each peer can redistribute any portion of file to any other peer.

  Ex: BitTorrent

- Scalability: Many peers can be there at one time.

  → In. Server client arch., one might not able to support 1M to 2M users, but in P2P arch., if the algorithm is right, one can support without limit

→ N peers, distribution time

→ Time taken to distribute a file to all users

**Distribution time for client server architecture:**

file size = F bits  users = N

upload rate = $u_s$ bits/sec

$$\text{Time taken (Minimum)} = \frac{NF}{u_s}\ s$$

If download limit is there, then $d_{min} = \min\{d_1, d_2, \ldots\}$

$$\therefore \text{Minimum distribution time} = \frac{F}{d_{min}}$$

$$\therefore \boxed{D_{cs} \geq \max\left\{\frac{NF}{u_s}, \frac{F}{d_{min}}\right\}}.$$

**Distribution time for P2P Architecture :**

File shared altleast once $\boxed{= \frac{F}{u_s} \text{ seconds.}}$

peer with least download rate takes time

$$\boxed{t < \frac{F}{d_{min}} \text{ seconds.}}$$

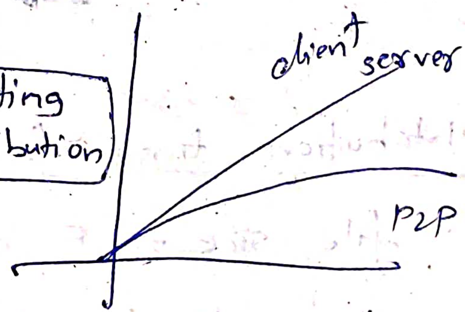
Total upload rate = $u_1 + u_2 + \dots + u_N$.

system delivers F bits to each N. Min dist time is

$$\boxed{t = \frac{NF}{u_{total}}}$$

$$\boxed{\therefore \text{Min time taken} = \max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_{total}}\right\}}$$

* Graph is in slides *




**Bit Torrent :**

↳ Collection of $\boxed{\text{peers}}$ $\boxed{\text{nodes participating in distribution}}$ in distribution of file called torrent.

→ peers in torrent download equals chunks of file ( typically 256 kbps )

→ peers accumulates more and more chunks over time

→ once completed, they may leave or continue

→ peers may leave with subset of files.

Each torrent has a node called tracker

When peers join the torrent, registers with tracker.

Each peer periodically updates tracker about it's presence.

A new peer requests for chunks that doesnot current there with her.

Order: Rarest first (Requesting policy)

Tit for Tat and give back to them in a higher rate

↳ identify top 4 peers who granted the peer chunks.

incentivising

unchoked (granting policy)

same for all peers.

New peer: optimistically unchoked

4/6 Distributed Hash Tables:

↳ Huge Database to stored among number of peers in a distributed way.

key-value pairs.

(Name, number)

(PAN, Aadhar)

Query by key and DB replies matching pairs

How ~~Transfer~~ occurs? : Theoe on both sides
Distribution

- Assign identifier to each node
  → integer in $[0, 2^{n}-1]$ for some fixed n.
- (key, value) pairs are also identified by integers using hash function

available for all users

converts strings to numbers i.e index

always successor

Assign the node based on index closed number
identifier = hashed value

challenge:
- if no proper organisation.

Circular DHT:
→ Each peer only aware of predecessor and successor.

- Dependency greatly reduced. instead of remembering all nodes.

Reduces number of queries.
→ N messages, at most

peer churn:
→ can come and go without warning.
→ keep track of immediate predecessor and successor
→ When neighbour leaves, peer updates it's neigbour.

live messages

7/2

# TRANSPORT LAYER

→ Reliable communication method : (without corruption or data loss)

controlling the transmission rate of transport layer entities in order to avoid or recover from, congestion within the network. (flow rate / congestion control)

## Services

- logical connection
  - no direct talking
- logical connection between processes of hosts
- constrained by service model
- Reliable data transfer
  - TCP protocol is safer
- can use encryption.
- segments.

## TL in internet :

- IP
  - → model → best effort delivery service model.
  - logical connection between hosts
    - Network layer
  - no gaurantee segment delivery
  - no ordely delivery
  - no gaurantee about integrity of data

UDP → process to process data delivery and error checking.

TCP → reliable data transfer

TCP → correct and inorder

TCP → congestion control

Transport layer works at it's best when underlying network layer does it's work

TL doesn't share data directly, instead:

→ to an intermediate socket → then to host

If multiple sockets, then each socket → unique identifier → port number

multiplexing: client side

demultiplexing: server side

* read slides for diagram *

8/1

Identified by 4 tuple

TCP socket

↓

host transport the socket

process

thread

task

Connectionless (no handshaking) UDP:

used in real time applications

Finer application service.

no connection establishment and no connection state

DNS

Small packet header overhead

stateless

buffers

congestion control parameters

Example

TCP: 20 bytes

UDP: 8 bytes

multiplexing, error checking.

no sequencing

principles of Reliable Data Transfer:

safe

Assumptions

No packet loss

No bit errors

RTD 1.0: