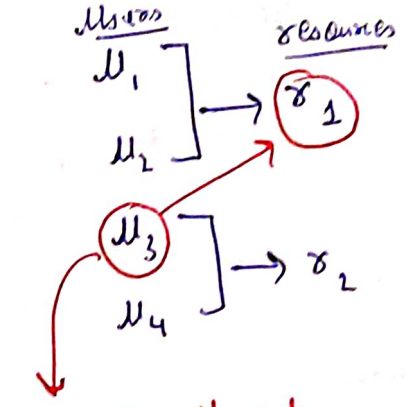
Protection & Security

#) Protection:

- It deals with the threats that are internal.
- It provides a mechanism for controlling the accen to prog., procen, users to the resource of a computer system.



#) Goals of protection:

(1) To prevent the mischievous, intentional violation of an accen restrited by user.

(2) Provides ~~dictas~~ distinguision b/w authorized and unauthorized usage.

③ It prevents breaching of confidentiality, integrity and availability

④ It prevents unauthorized reading of data, unauthorized modification of data & unauthorized destruction of data.

⑤ It prevents DOS (denial of service) and unauthorized use of resource.

↓

prevention of legitimate user.

#> Principles of Protection:

- Protection follows 'principle of least previlege' i.e. programs, users & proces should only be given enough privileges to perform their task.
- It limits the damage if any kind of protection is breached.

Domain of protection:

- Domain is a set of access right, each of which is an ordered pair < object name, rights set >.
  - May be a process or user.

- (previous unauthorized prevention vale points)

- D_1
  - $< O_3, \{ read, write \} >$
  - $< O_1, \{ read, write \} >$
  - $< O_2, \{ execute \} >$

$< O_2, \{ write \} >$ $< O_4 \{ print \} >$ $< O_1, \{ execute \} >$ $< O_3, \{ read \} >$

→ Domain Implementation with Unix:

- In Unix OS, domain is associated with the user.

  (Domain = User-id)

- → Switching the domain corresponds to changing the user identification temporarily.

⇝ This change is accomplish using file s/s.

→ each file has associated with it a domain bit (setupid bit).

when file is executed and setupid = On, then user-id is set to owner of the file being executed.

when execution completes, user-id is reset.

⇝ This change is accomplish using passwords.

→ 'su' command temporarily switches to another user's domain when other domain's password provided.

⇝ This change is accomplish using commands like 'sudo'.

Accen Matrix:

- It is a model of protection that can be viewed as a matrix.

  Rows represents domains
  columns represents objects

- Accen (i, j) is the set of operations that a prozen executing in domain D_i can invoke on objet O_j.

files →

domain \ objet	F_1	F_2	F_3	Printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

#) Implementation of accen matrix:

① Global table:

(same as simple accen matrix)

- each cell has a domain, objet & right set.
- If the table is large, the computation will get complex.

dis-adv.

- eg:- (previous table)

② Accen list for objet:

- each column implemented as an accen list for an objet

eg:- objet F_1 → read, write

objet F_2 → read

objet F_3 → read, execute, write

③ Capability list of domain:

- Here each row implemented as the capability of a domain.
- eg:- domain D_1 → read
  " D_2 → print
  " D_3 → read, execute
  " D_4 → read, write

④ Lock-key mechanism:

- It is a compromise b/w access list & capability list and each object has a list of unique bit patterns called locks.
  Each domain has a list of unique bit patterns called keys.

#) ~~de~~ Protection v/s Security

- Def.
- internal...
- it outlines which users are permitted to access a resource
- Simple queries are handled

- Def.
- external...
- it specifies whether or not a specific user is allowed to access the S/s.
- complex queries are handled.

#) Security:

- It deals with the threats to information both or external.
- Security helps o/s from being attack, threats & intruders.
  - attack → attempt to breach security.
  - threats → Potential security violation
- Can be done mostly using firewall.

#) Various security violation:

- CIA vala point.
- DOS vala point.
- Masquerading: pretending to be an authorized user

#) Levels of Security:

(1) Physical (2) Human (3) OS (4) network

① **Physical:**

The sites (places) that have computer s/s, machine rooms, terminals or workstations must be secured against armed or [illegible] intruders.

② **Human:**

Authorization must be done carefully to ensure that only appropriate and legitimate user can have accen to specified comp. s/s.

③ **Operating s/s:**

The s/s must protect itself from accidental or purposefull security breach.

④ **Network:**

Much computer data in modern s/s travels over shared lines like the Internet, wireless connections or dial-up lines. Intercepting these data could be just as harmful as breaking into a computer.

#) Hacker v/s Cracker

Hacker	Cracker
→ They look for flaws in comp. & internet security and their sole aim is to rectify these flaws and improve the security of the content.	→ The purpose is to break / breach the security of the computers & networks.
→ Hacker build things	→ Cracker break things
→ Have advance knowledge of the comp. security	→ Usually not very skillful except for some.
→ They help in catching the cracker	→ They tries to cover up their tracks.

#) Program threats:

- Proces, along with the kernal, are the only means to work on a computer. Therfore, crackers common goal is to breach the security & alter the proces to perform some malicius unwanted task.

1. Trojan Horse:

- It is a malicious program that may give full control of an infected PC to another PC.
  The code segment that tries to misuse its own environment.
- These code segments hidden as an attachment in an email or a free-to-download file, then transfer onto the user's device.
- Once downloaded, the malicious code will execute the task like → spy on user's online activity
  → steal sensetive data.

2. Trap Door or Back Door:

- A trap door is a kind of a secret entry point into a program that allows anyone to gain accen to any system without going through the usual security accen procedures.
- Trap door is a method of bypassing normal authentication methods.
  These are difficult to detect.
- Programmers uses trap door legally to debug & test programs.

3. Logic Bomb:

- It is a piece of code intentially inserted into a software system / computer system that will do a malicious function when specified conditions are met.
- It is hard to detect.

4. Stack and Buffer Overflow:

- It is a most common way for an attacker to attack from outside the system.
- The attacker use a network or dial-up connection to gain unauthorized access to the target s/s.
- The attack creates a bug in a program which is a simple case of poor programming.

  The attacker sends more data than the program was excepting.

- By using trial and error, or by examining the source code of the attacked program, attacker determines the vulnerabilities and writes a program to do the following:

  (1) Overflow an input field, command-line argument, or input buffer.

  (2) Overwrite the current return address on the stack with the address of the exploit code ~~that~~ loaded.

  (3) Write a ~~same~~ simple set of code for the next space in the stack, that includes the commands that the attacker wishes to execute.

5. Virus:

- Fragment of code embedded in a normal program.
- 'Self-replicating' and are designed to 'infect' other programs.
- 'modify' or 'destroy' the s/s files and cause s/s to crash & program to malfunctions.
- Generally copied via e-mail, or downloaded from infected internet file.

#) S/s and network Threats:

- It involves abuse of service and network connections.
  ↓
  provided by O.S
- OS resources & user files are misused.
- These threats includes : worms, port-scanning & DOS attacks

→ Masquerading & replay attacks are also commonly launched over networks b/w s/s.

→ In masquerading, one participant in a communication pretends to be someone else.

By masquerading, attackers breach authentication, security, integrity, availability, confidentiality, and they gain accen that they would not normally be allowed.

① Worms:

- A process that uses the spawn mechanism to duplicate itself.
- It spawn copies of itself, using up s/s resource and perhaps locking out all other processes.
- This further result in shutting down the entire network & s/s.

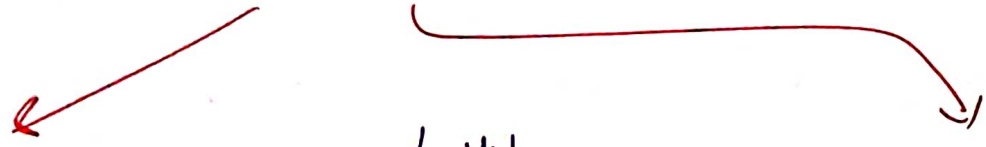
② Port Scanning:

- It is not an attack but rather a mean for cracker to detect a s/s's vulnerabilities to attack.
- port scaning is automated, involving a tool that attempts to create a TCP/IP connection to a specific port or a range of ports.
- If the connection was successful, the cracker (or tool) could attempt to communicate with the answering service, to determine if the service was indeed Send mail and, if so, if it was the version with the bug.

~~eavesdropping~~

③ Denial of service :

- DOS are not aimed to gain info. or steal resources, but rather at disturbing genuine use of a system or facility.
- network based.
- 2 categories :



use so many facility resource such that no useful work can be done.

disrupting the network of the facility

DDOS : (Distributed Denial of Service)

- An attack in which multiple compromised computer s/s attack a target such as server, website and cause a dos.
- The flood of incoming messages, connection req. to the target s/s forces it to slow down or even crash & shut down, thereby denying service to legitimate user.

→ How DOS attacks works:

- (racker / hacker / attacker → through e-mails / website
  ↓
  malware send to diff. comp. s/s
  ↓
  gain accen to the infected s/s through that malware.
  ↓
  groups of large no. of such infected s/s is called botnet, which is being (reated by the attacker
  →
  Through this botnet, the attacker sends a large no. of junk data, connection req., messages, etc. to the target s/s & forces it to slow down or even crash & shut down.

① Traffic attacks: huge vol. of TCP, IPCM packets are sent to the taget which result in lost of legitimate req.

② Bandwith attacks: overload the target with massive amount of junk data. Result in complete DOS.

③ Application attacks: Remove resources from application layer, resulting in unavailability of s/s service.

→ DDOS prevention:

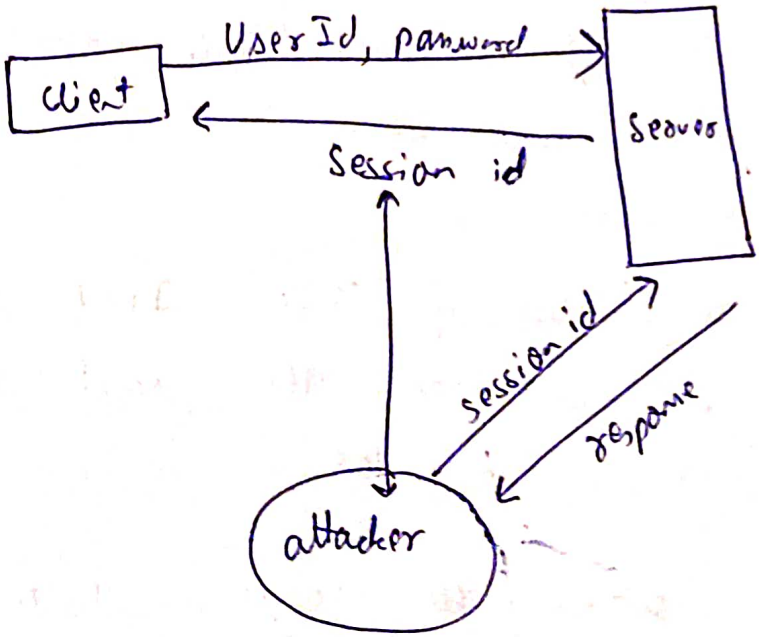
① Have a DDos plan

② Protect your network → tools like antivirus, firewalls, network monitoring s/w.

③ Call a DDos specialist

④ Overprovision bandwidth.

#> Session hijacking (TCP hijacking)



- User connects with server through his Id & password. The server than creates a session & provide a session Id to the user so that he can give multiple req. to server without using Id, pass every time & using session id.

But attacker steal this session Id & pretend like an authentic user to the server & then sends req. to server for gain personal data of the user.

~~Attacker~~

#) Loosely Coupled S/s v/s Tightly Coupled S/s

- S/s in which there is distributed mem.
- data rate is low
- H/w & s/w are not dependent on each other
- issue in one segment will still keep the S/s running

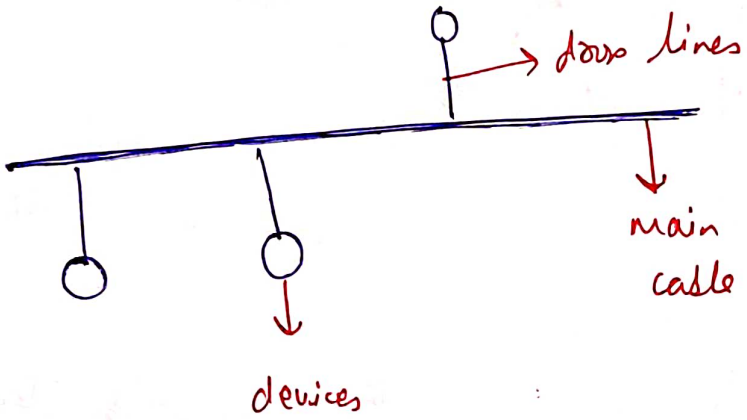
- S/s in which there is shared mem.
- data rate is high
- H/w & s/w depends upon each other
- issue in one segment will bring the entire S/s down.

#) Network topologies:

- Arrangement with which computer s/s or network device are connected to each other.

① Bus Topology:

Every computer & network device is connected to a single thick cable.



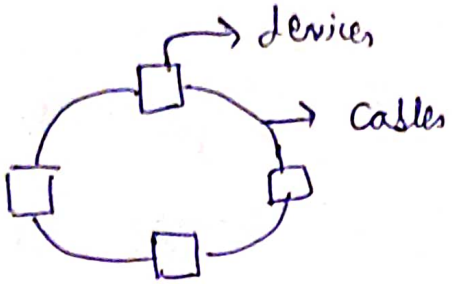
adv:

① cost effective
② Easy to understand
③ Cable req. is least.

dis:

① single point of failure (main cable)
② Cannot share multiple data at same time by diff. devices.

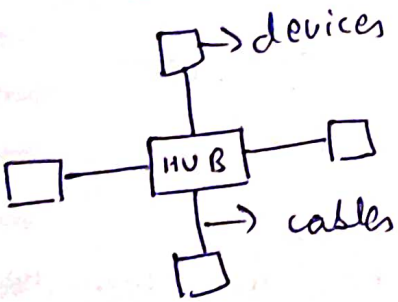
(2) Ring topology:



- Computers or devices are connected to each other in such a manner & comp. is only connected to it's previous & next comp., thus forming a ring.
- adv: (1) cost effective (2) easy to understand (3) less cable req. (4) NO single point of failure (5) can share multiple data at same time.

dis-adv: (1) Difficult to add extra devices (2) Troubleshooting is difficult.

(3) Star topology:

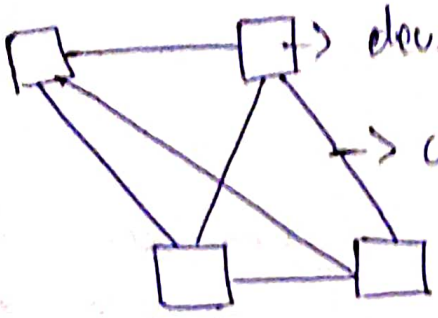


- all comp. or devices are connected to hub for sharing their resources.

- adv: (1) Easy to setup (2) Easy to troubleshoot (3) fast performance

dis-adv: (1) Expensive (2) single point of failure (HUB)

(4) MESH topology :



- point-to-point connection
- All devices are connected to each other.
- no. of cables = $^nC_2$ = $\frac{n(n-1)}{2}$
- no. of ports in each device = $n-1$

adv:
1. Provides high seq. & privacy
2. Easy to connect
3. no single point of failure.
4. Fast comm.
5. Reliable

dis-adv:
1. more cables to handle (Bulk of wires)
2. ~~cost~~ Expensive

(a) Full-Mesh :

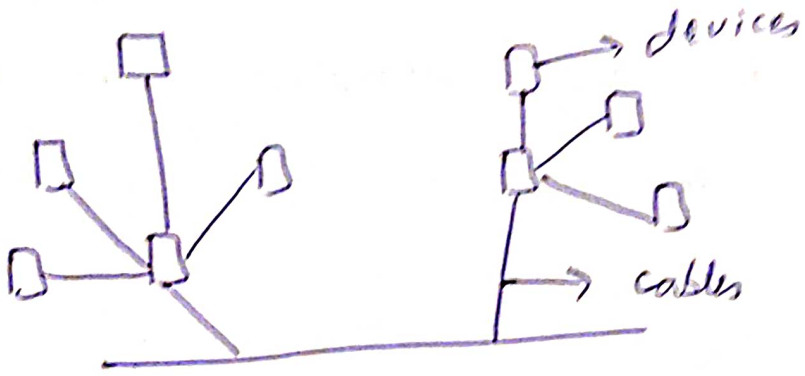
all comp. are connected to each other

(b) Partial-Mesh :

not all but certain computers are connected to those with which they communicate frequently.

⑤ Tree Topology:

- Combines the characteristics of bus & star topology.
- all comp. are connected with each other in heirarchal fashion.

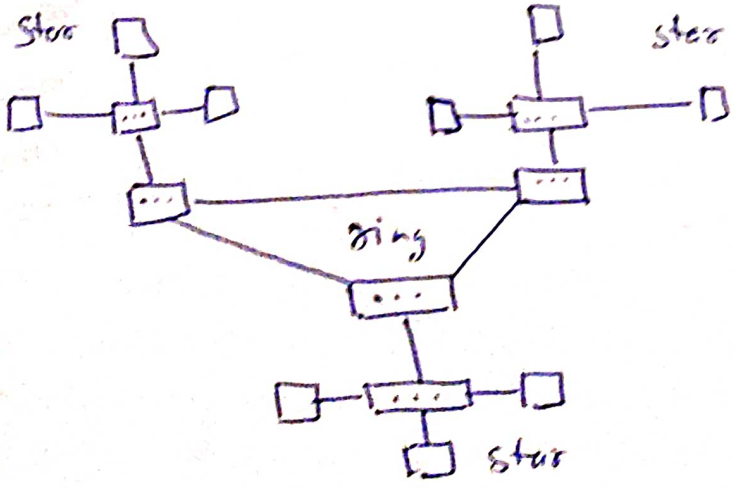


adv: ① no single point of failure
② Can be further extended

dis adv: ① Heavily cabled ② Costly
③ difficult to maintain

⑥ Hybrid Topology:

- It is a combination of various different topologies.



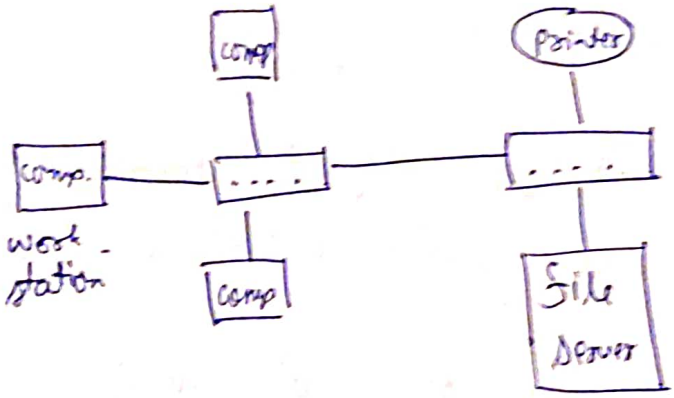
adv: ① Flexible
② Can be further extended

dis: ① Heavily cabled
② Costly
③ difficult to maintain.

Network Types:

① LAN (Local Area Network):

- Group of computer connected to each other in small area such as building or office.
- Personal computers and workstations can share data easily and at a very high speed.

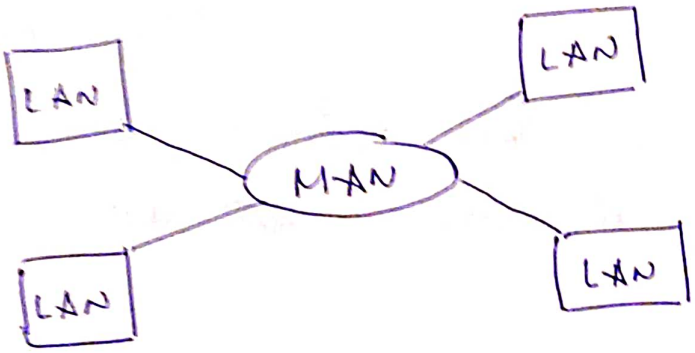


adv:
- Can communicate with each other without internet.
- Cost effective
- High security
- data transfer rate ↑

~~dis adv~~

② MAN (metropolitan area network):

- Covers larger area than LAN and smaller than WAN.
- It covers a large geographic area by interconnecting different LAN.
- Govt. agencies uses MAN to connect to the citizen & private industries.



- adv:
  - used in communication b/w banks in a city
  - —"— —"— colleges
  - —"— —"— .
  - —"— —"— military
  - —"— —"— airlines.

③ WAN: (Wide Area Network)

- Network that extends over a large geographical area such as state or countries.
- Spans through telephone lines, satellite links & fibre optical cable.
- eg:- Internet.
- 2 types :- ① wired -> telephone lines, ISC/OFC
  ② wireless -> satellite links

adv:
① Interconnection b/w company's branch through WAN.
② Fast transmission of messages/data

dis:
① Security issue
② Needs firewall & anti-virus
③ High set-up cost

④ PAN (Personal Area Network):

- The network is restrained to a single person, i.e., communication b/w comp. & comp. devices is centered only to an individual's work space.
- Typically, the network spans to 10 mtrs.

#> Switching techniques:

- In a large network, there can be multiple paths from sender to receiver.
  The switching techniques will decide the best route for data sharing.

① Circuit Switching:

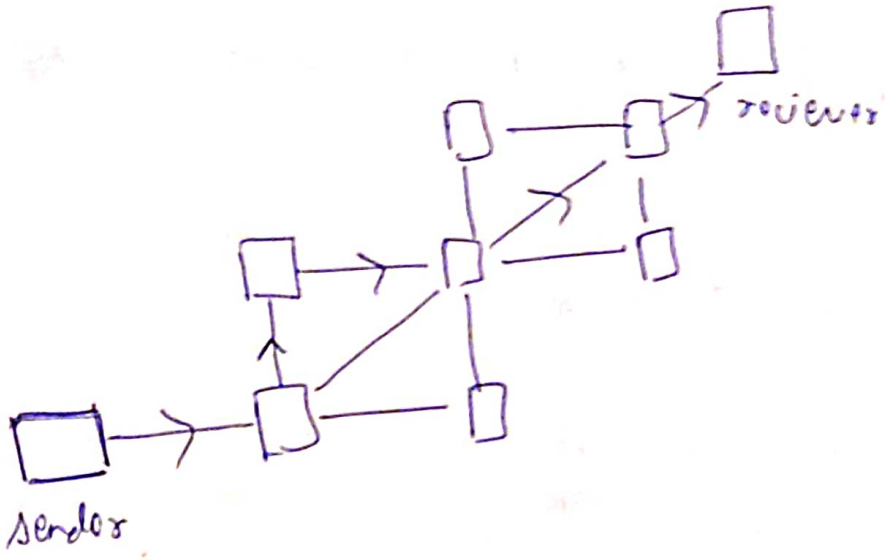
- Technique that establish a dedicated path b/w sender & receiver.
- Once connection/path is established, it will remain exist until the connection is terminated.

→ When user wants to send the data, a req. signal is sent to reciever & reciever sends back the acknowledgement to ensure availability of dedicated path.

→ 3 phase:
① Circuit establishment
② Data transfer
③ Circuit disconnect

→ Eg:- ~~Telephone line~~ Telephone calling (modern vali)

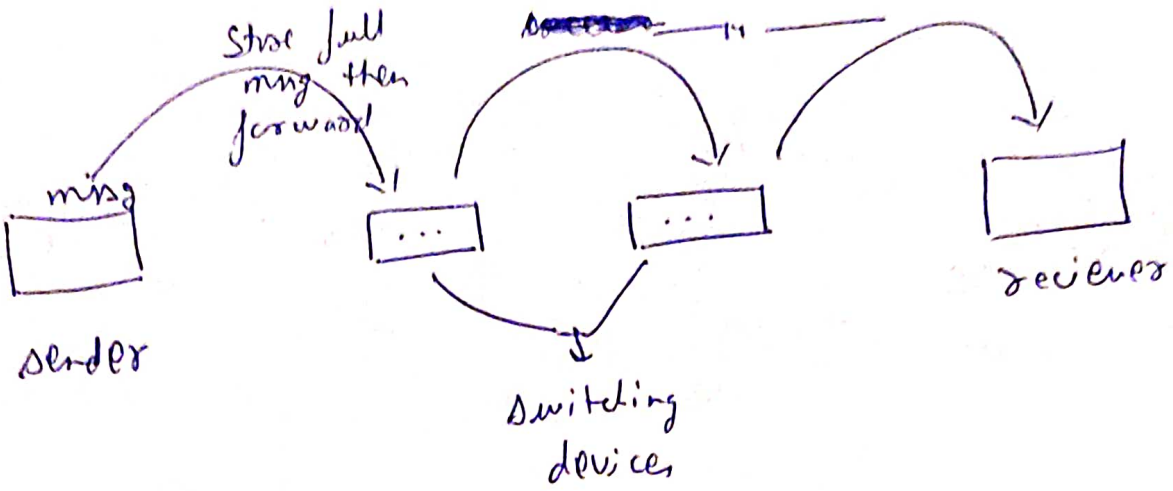
→



② Message Switching:

- no establishment of dedicated path.
- somewhere b/w CS & PS.
- The whole message is treated as a data unit and is transfered / switched entirely to the destination.
- Switch working on mesg switching first recieves the whole msg & transfer it to the next switch. Entire message is hopped further until it reaches destination.

→



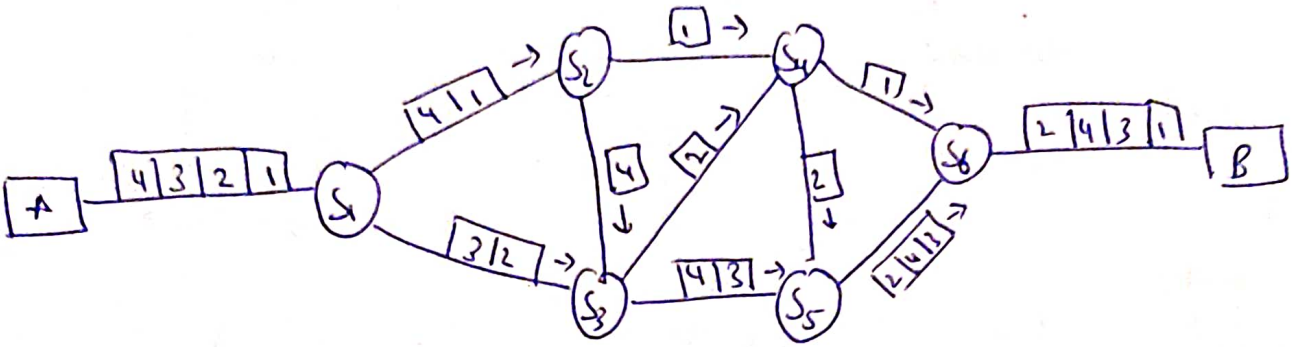
→ characteristics:

① Store and forward

② Message delivery: wrapping entire info. in a single msg & transfer it to source destination.

③ Packet Switching:

- message is sent in one go, but is divided into smaller peices & they are sent individually
- smalled peices of msg is called packets & packets are given a unique number to identify their order at the recieving end.
- All packets are re-assembled at the recieving end in correct order.



→ Approaches :-

① Datagram Packet Switching (Connectionless):

Each packet contains the info. about destination & switch use this info to forward ~~packet~~ packet to correct destination.

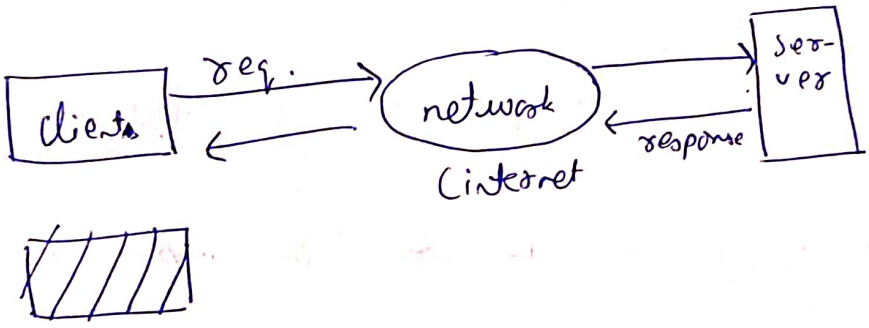
② Virtual Circuit Switching (connection-oriented):

a pre-planned path is established before msg is sent.

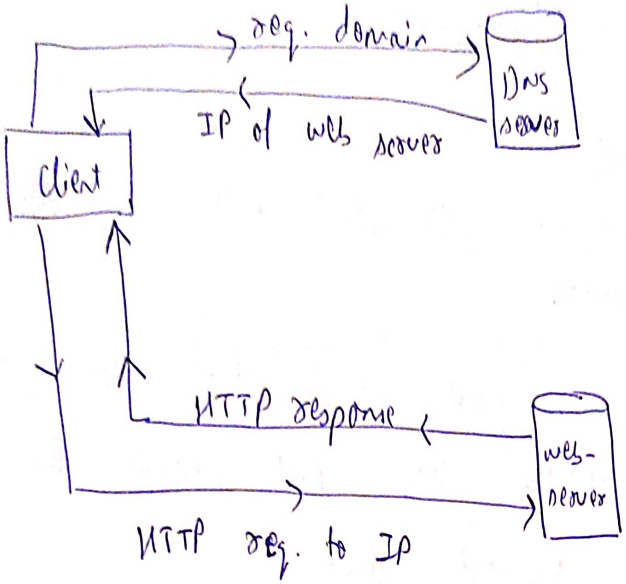
Client Server architecture :

- Client → host is a comp. s/s that sends a ~~request~~ request to the server.
- Server is a remote comp. which responds to the user/ client's req. and provide the client with the required data.

→ Basically, client requests for something & the server fullfill that req. of user/ client.



① User enters his req. in browser.
② Browser than req. DNS (Domain Name s/s) server.
③ DNS server lookup for the address of the web server
④ DNS server responds with the IP addres of the web server.
⑤ Browser sends ~~user~~ an HTTP/HTTPs req. to web server's IP (provided by DNS)
⑥ Server sends necessary files of the website & browser renders that ~~website~~ files & website is displayed.



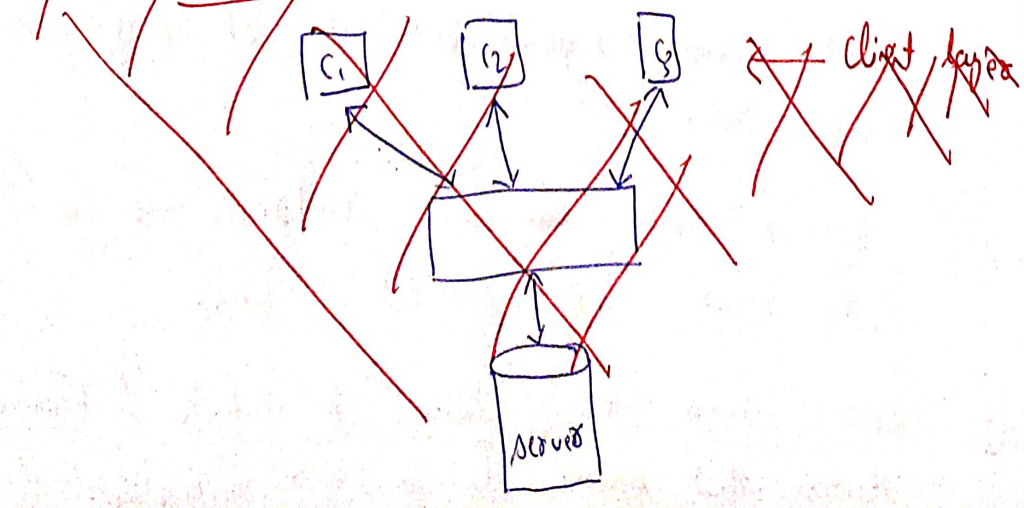
=> 2 tier CSA:

- Consist of two layers i.e. client layer & server layer (previous jo hamne padha).

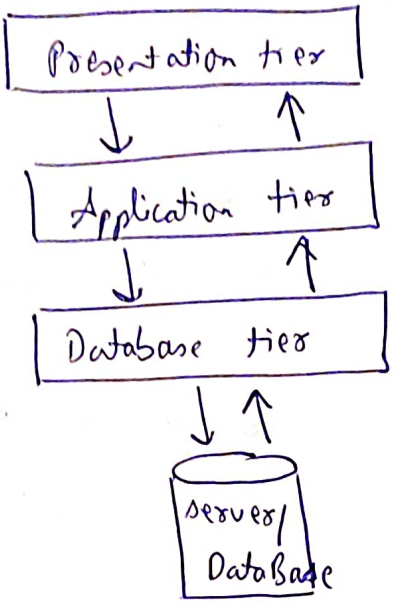
-) easy to maintain & implement
-) cost effective
-) poor security
-) not very scalable.

] this gets fixed in 3-tier.

=> 3 tier CSA:



⇒ 3-tier CSA:



- **Presentation tier:**
  - Take req. from the client & displays info. to client. It communicates with app. tier through web browser & display info through web browser.
  - developed using HTML, CSS, JS.
- **Application tier:**
  - Information gathered from PT is procesed in here. detail
  - developed using Java, PHP.
- **Database tier:**
  - It is used to store the requested/ procesed info. so that it can be retrieved later when required.
  - developed using ~~MSE~~ MySQL, ~~Oracle~~ Oracle.

#) RPC (Remote Procedure Call):

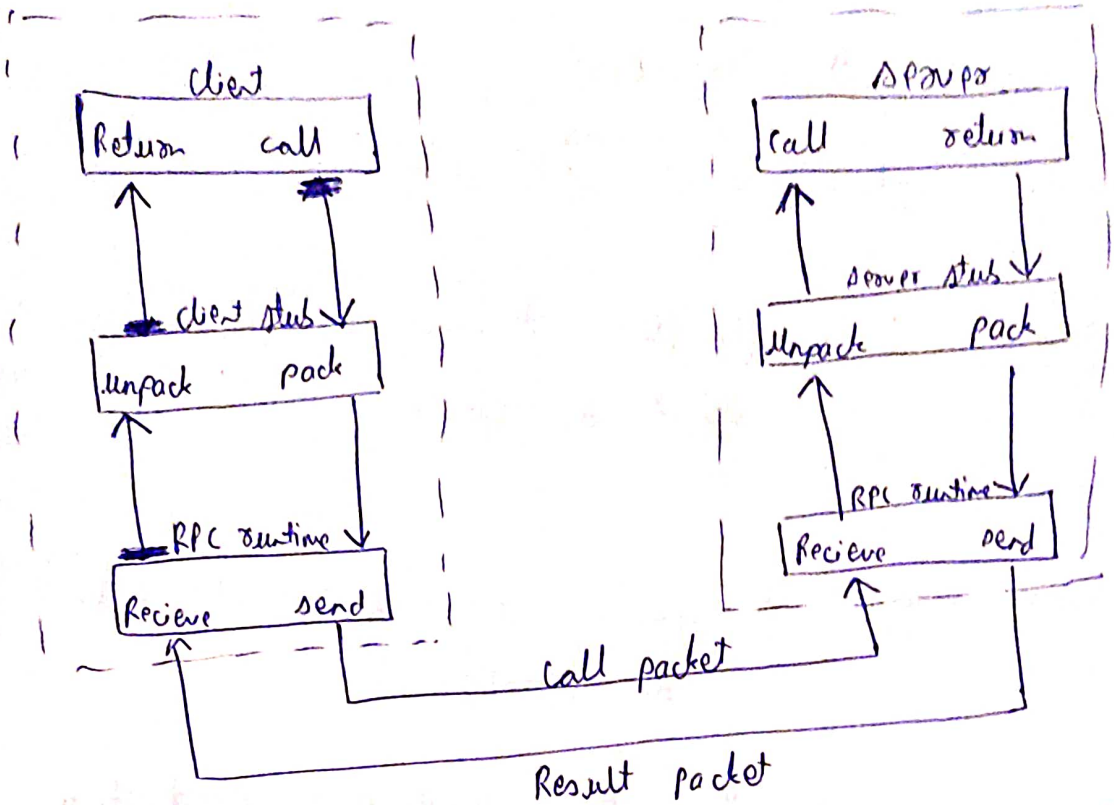
- It is a protocol that one program can use to request a service from a program located in another computer on a network without having to understand the network details.
- also called as ~~procedural~~ funct^n call or subroutine call.
- RPC uses CS A.

How it works:

→ It includes mainly 5 elements:

1. Client
2. client stub → Piece of code used for converting the parameters
3. RPC Runtime → RPC communication package
4. Server Stub
5. Server

@



- → client will send / call for a req. to client stub.
- → client stub will pack the data / req.s into a packet & will further send the packet to RPC runtime of client side.
- → RPC runtime of client side will send that call packet to " " " server "
- → " " " " " will further send that packet to server stub & server stub will unpack the packet & sends the req. to server.
- → .....

→ client: • Initiates RPC
• Invokes client stub

→ RPC runtime: • Handles transmission of message b/w client & server.

→ The server stub: • invoke the appropriate procedure in server.

→ Server: • execute the appropriate procedure.

→ RPC provides abstraction.