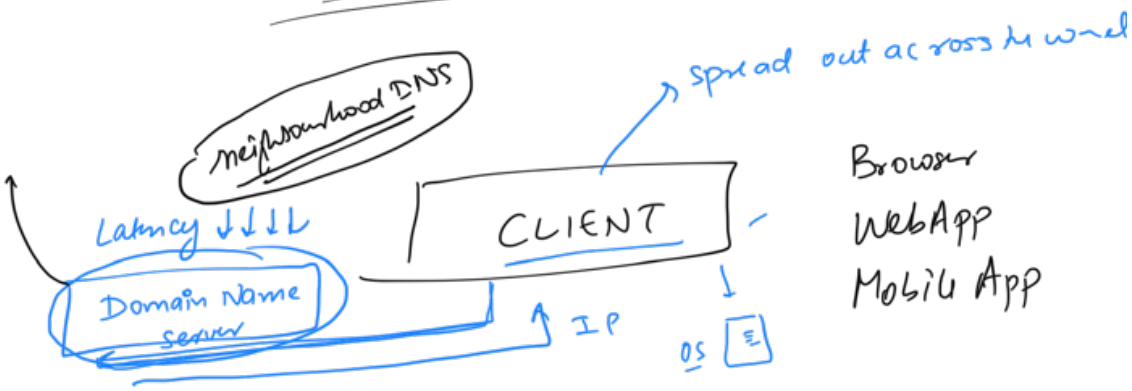
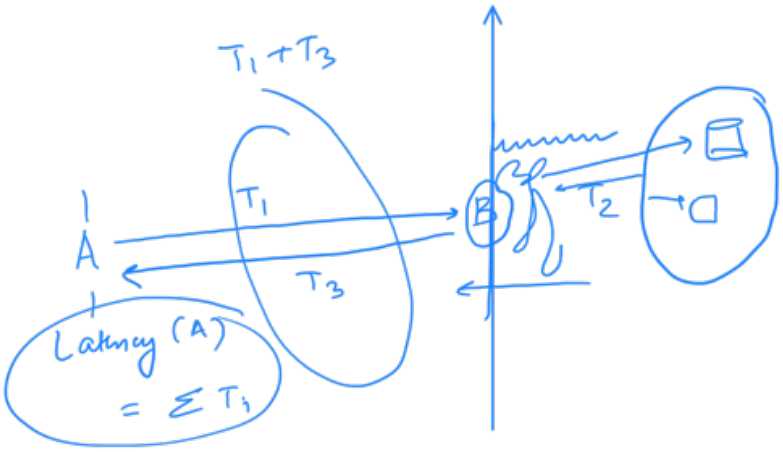
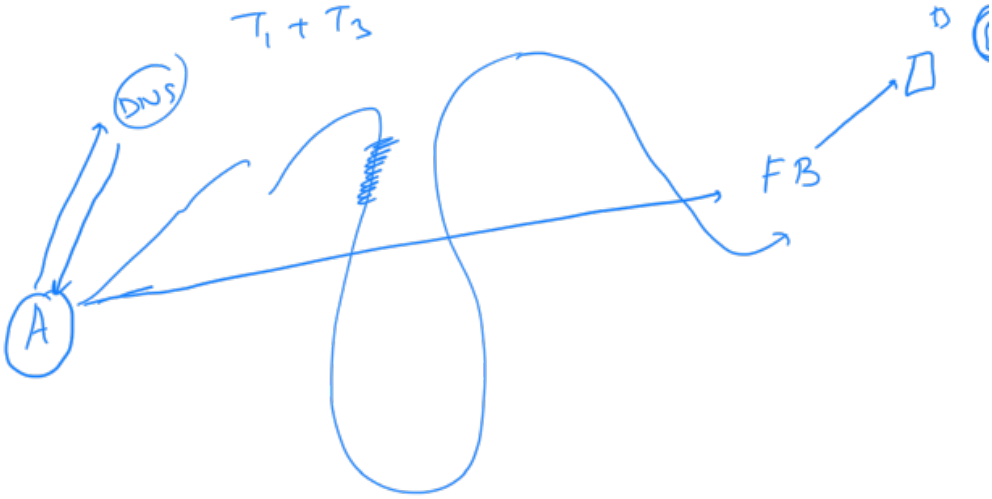
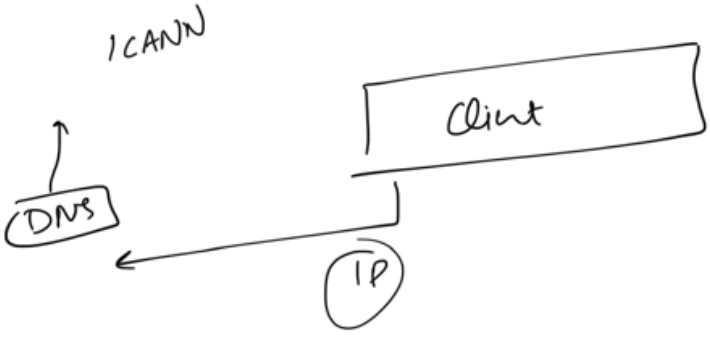
High Level Design

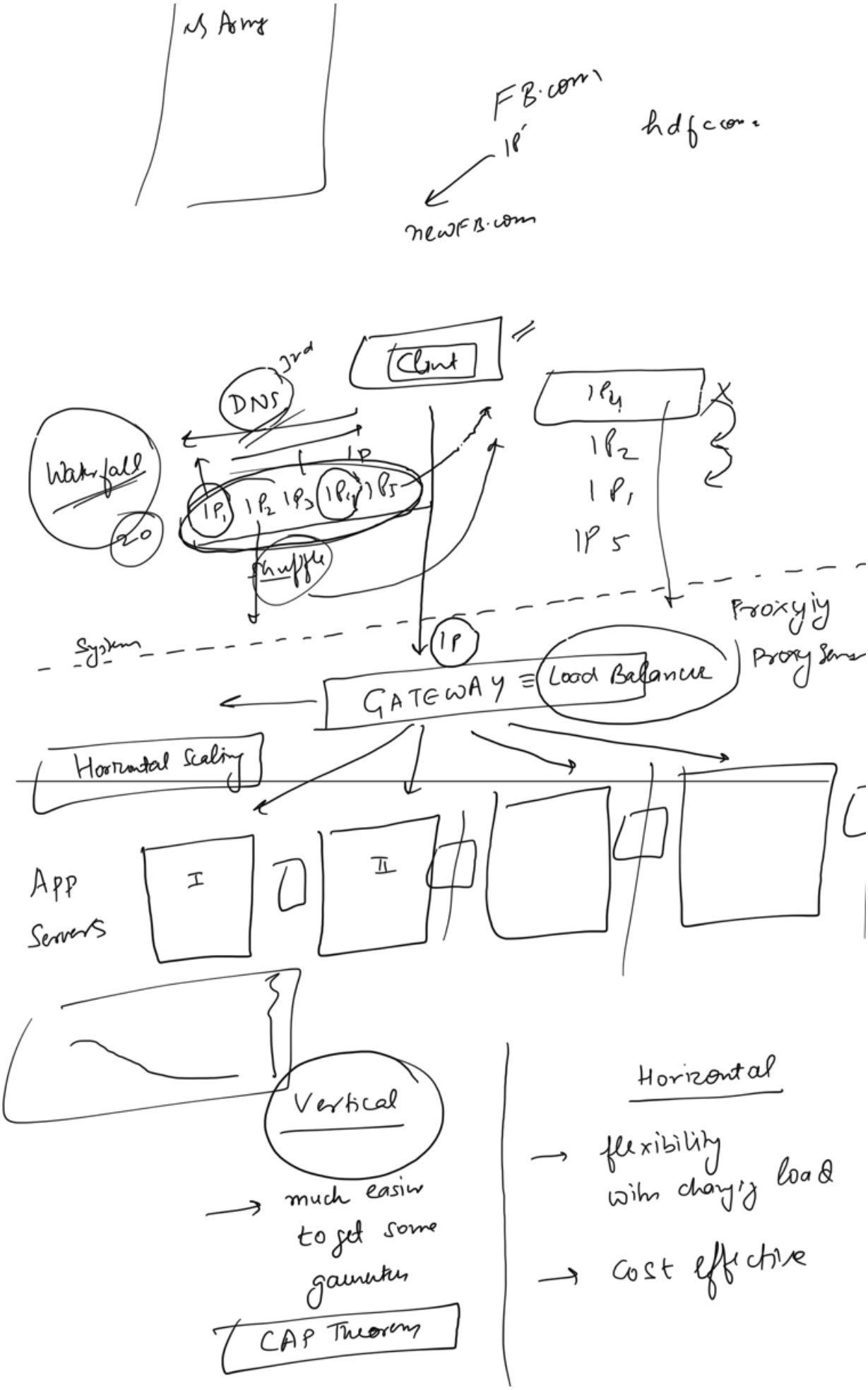


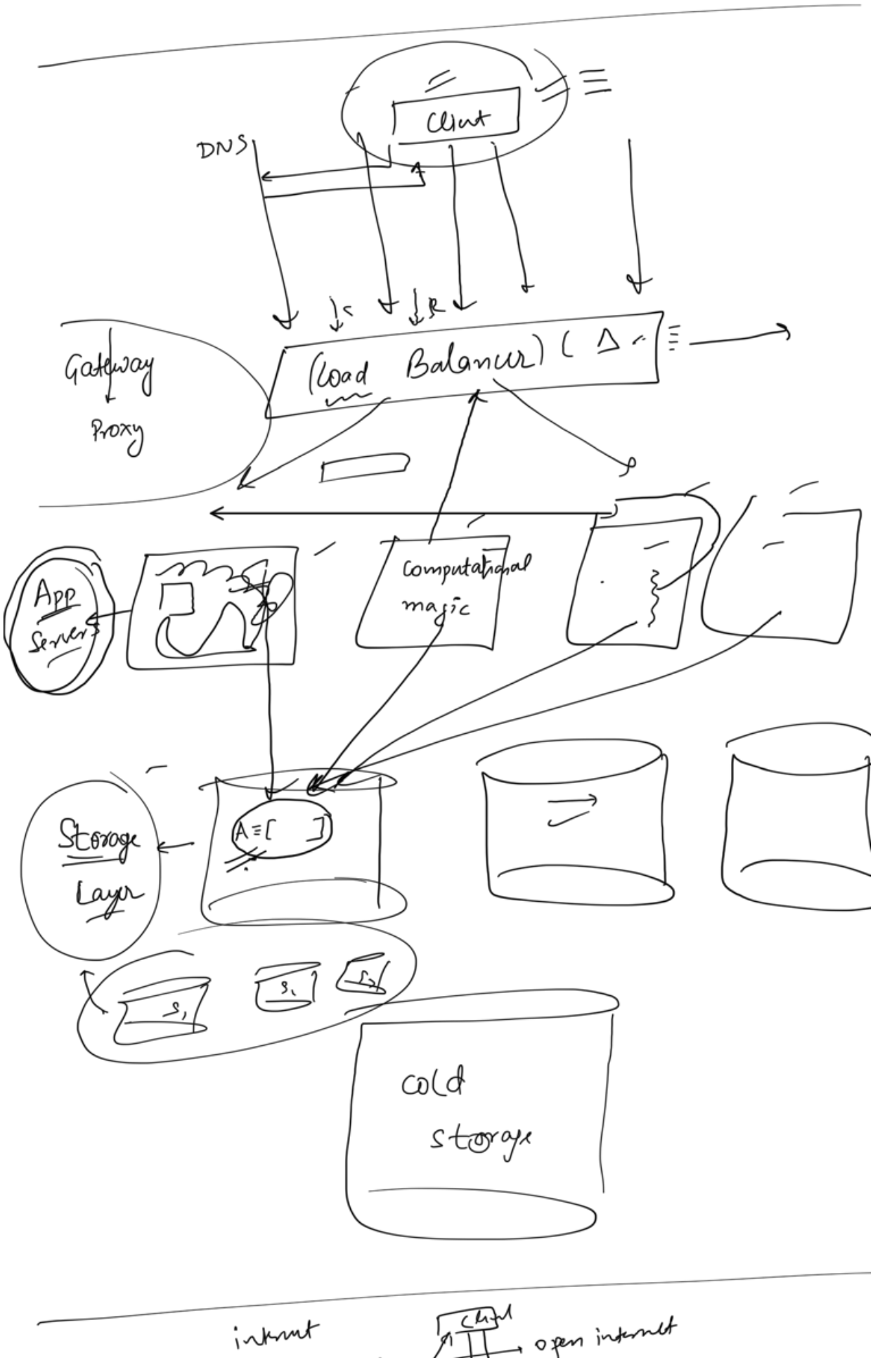


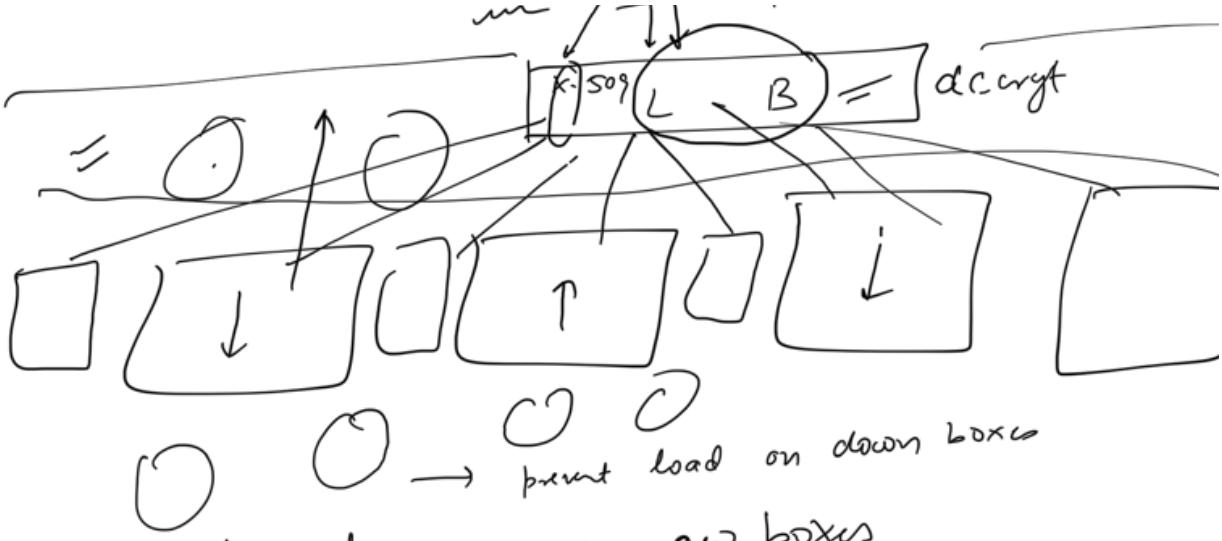




ISPs
Google
FB


Army
FB.com
IP
newFB.com
hdfc.com
Client
3rd
DNS
IP
Waterfall
20
IP1 IP2 IP3 IP4 IP5
Shuffle
IP4
IP2
IP1
IP5
System
Proxying
Proxy Server
IP
GATEWAY = Load Balancer
Horizontal Scaling
App Servers
I
II
Vertical
much easier to get some guarantee
CAP Theorem
Horizontal
flexibility with changing load
Cost effective


Client
DNS
Gateway
Proxy
Load Balancer
App Server
Computational magic
Storage Layer
A=[ ]
cold storage
internet
open internet



→ prevent load on down boxes

introduced → rgist new boxes

→ balance out

— X.509 —

Layer 4 ≡ Transport Layer

addition security

~~security~~ Source IP
Source Port
Dest IP
Dest Port

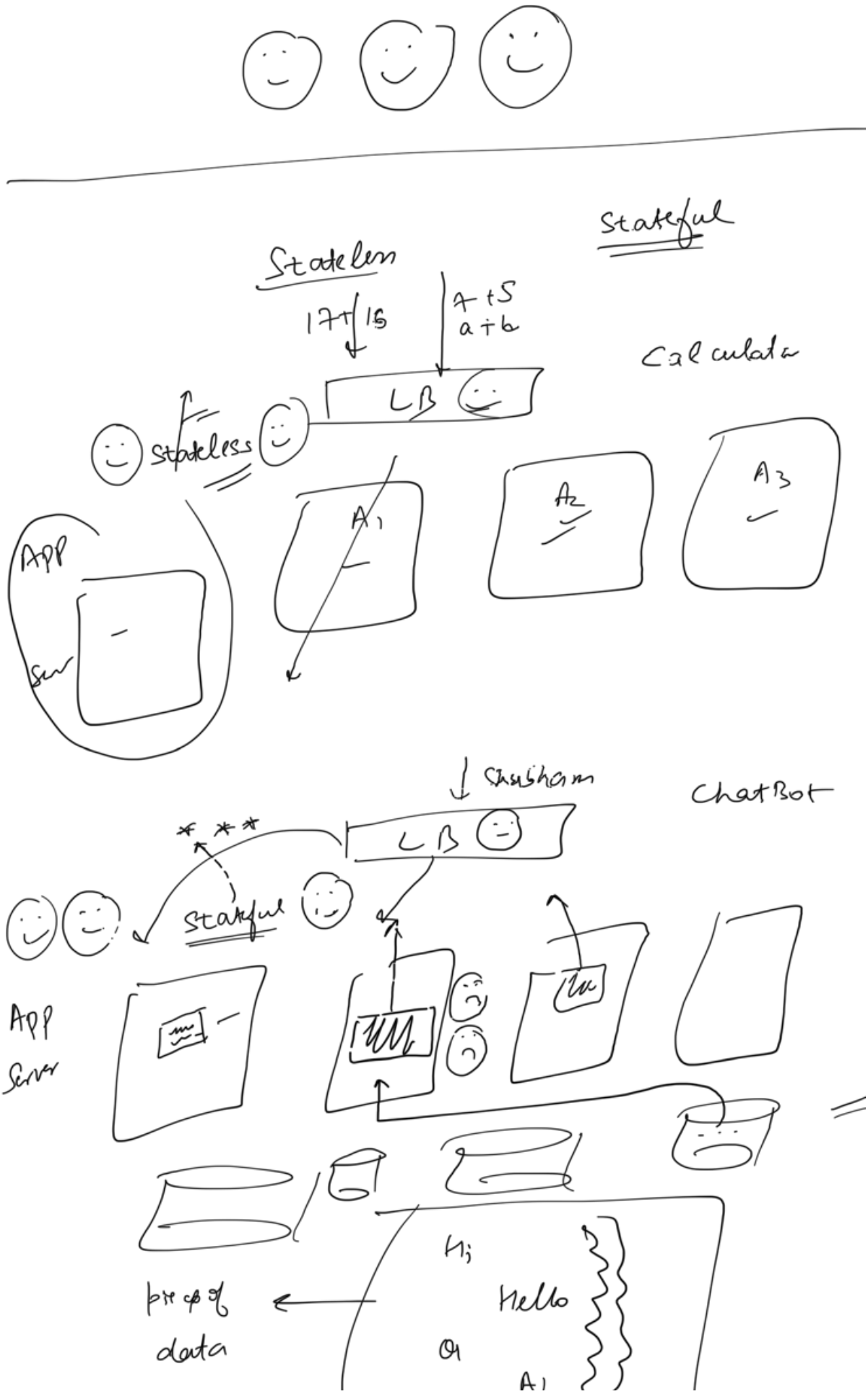
fast
simple

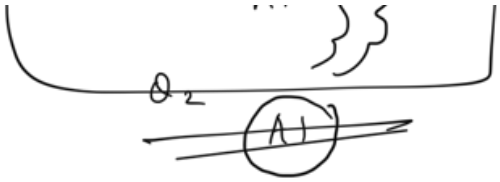
Layer 7 LB ≡ Application

user id ≡ 707
uid = 107

slowest

much better flexibility

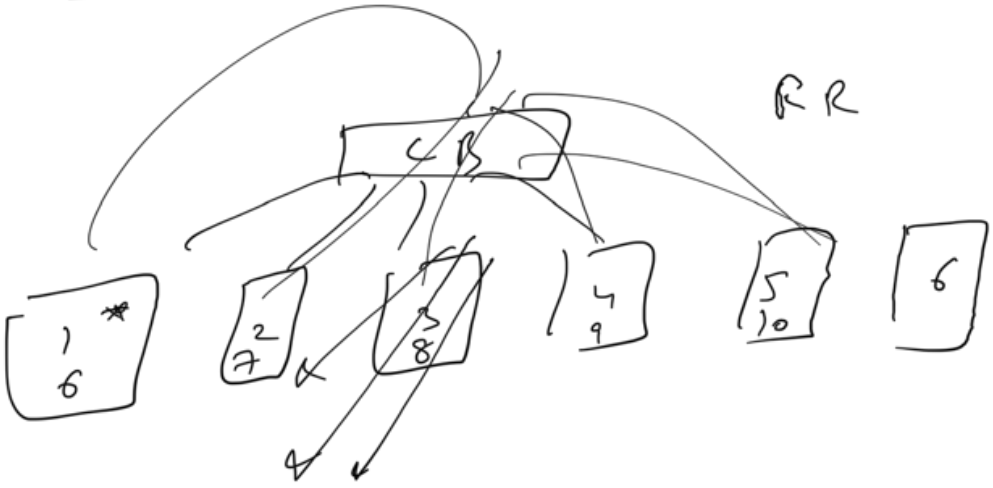

Stateful
Stateless
17+15
4+5
a+b
Calculate
LB
stateless
App
Ser
A1
A2
A3
Shubham
ChatBot
LB
stateful
App
Server
Hi;
Hello
Q1
A1
prof
data

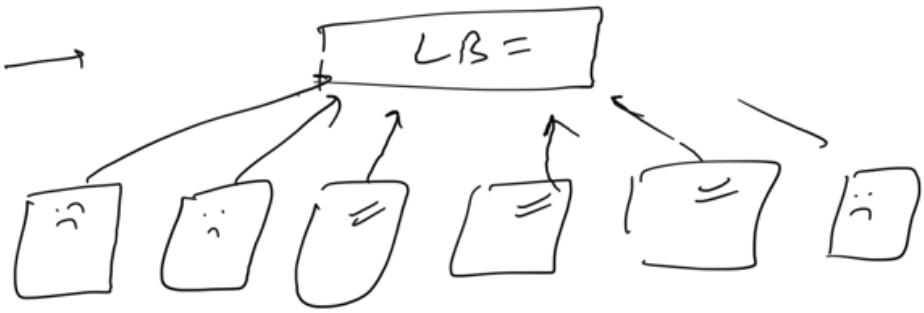


Load Balancing

Statelen

→ Round Robin

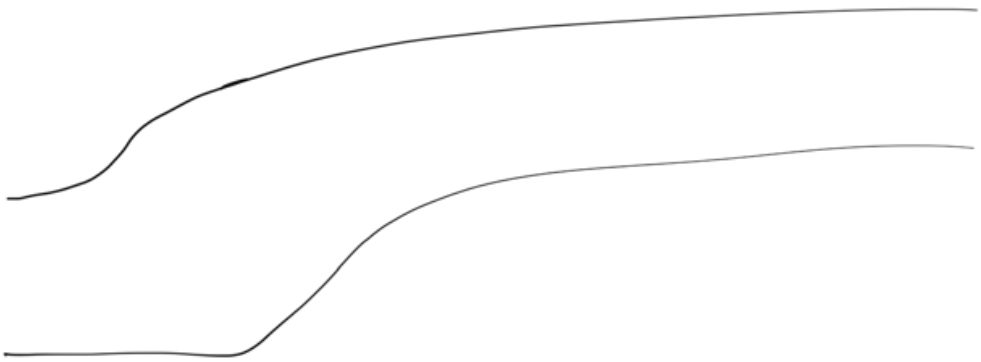


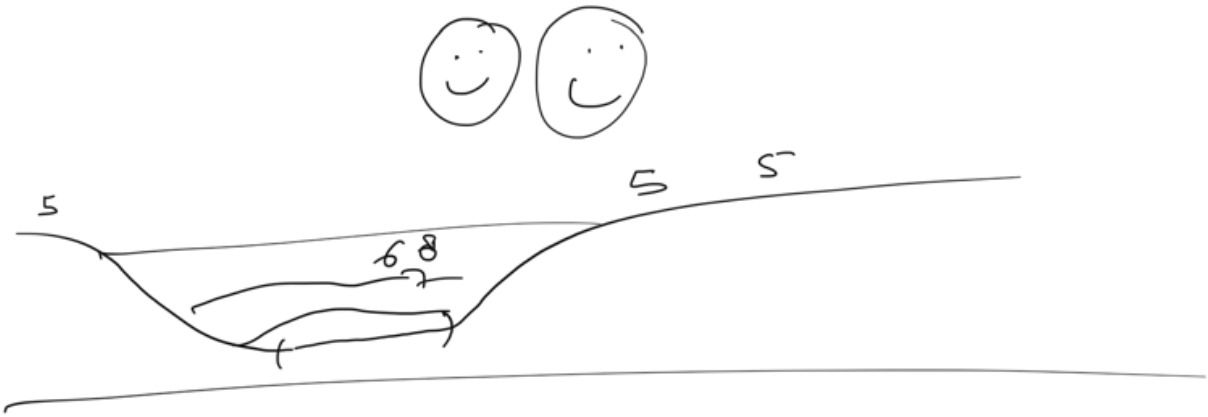


WRR

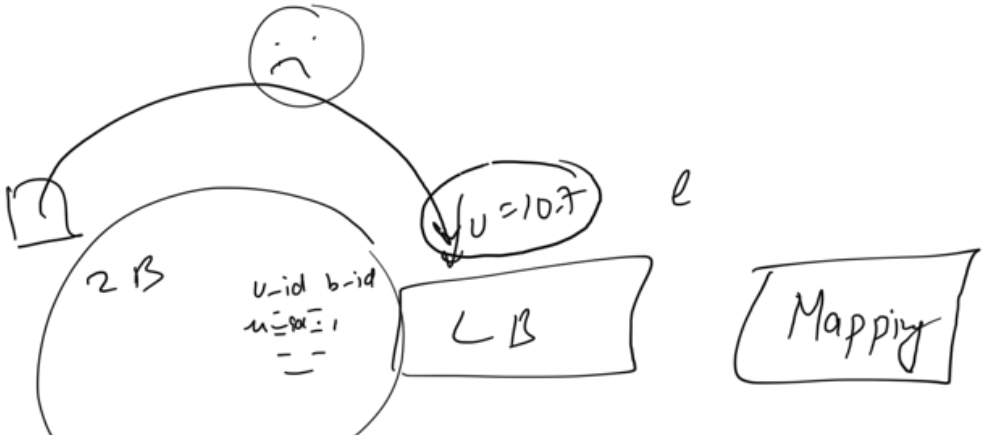
→ (Least Connection first)

→ WRR ARR





Load Balancing in Statfull



Sol (1) Maintain Map

BAD

Load Balancing func will slow down

map-size = order of key-size

Sol (2) Hash

BAD!!!

10 boxes ≡ 10 App Server

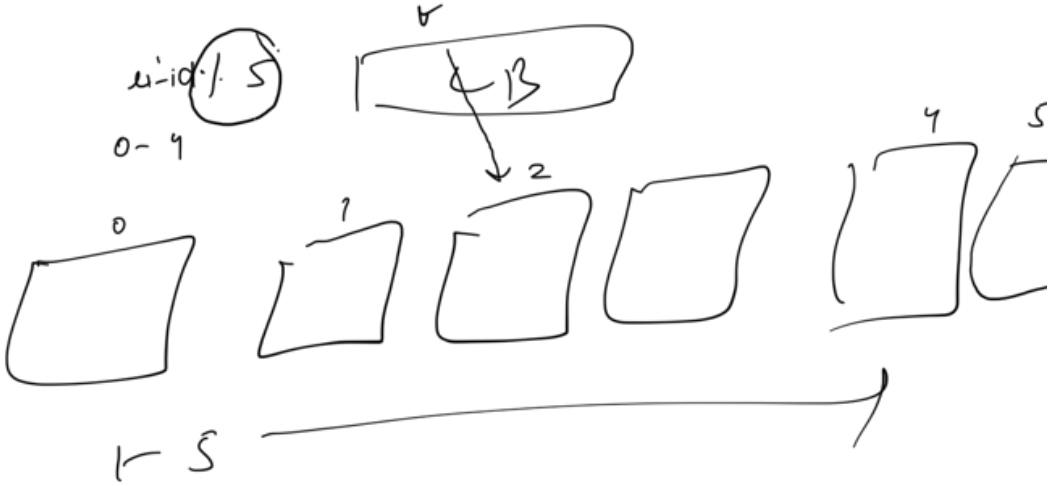
0-9

uid %. 10 = □

Agush ≡ 107

uid % 5

0-4



→ gaurav of stickiness

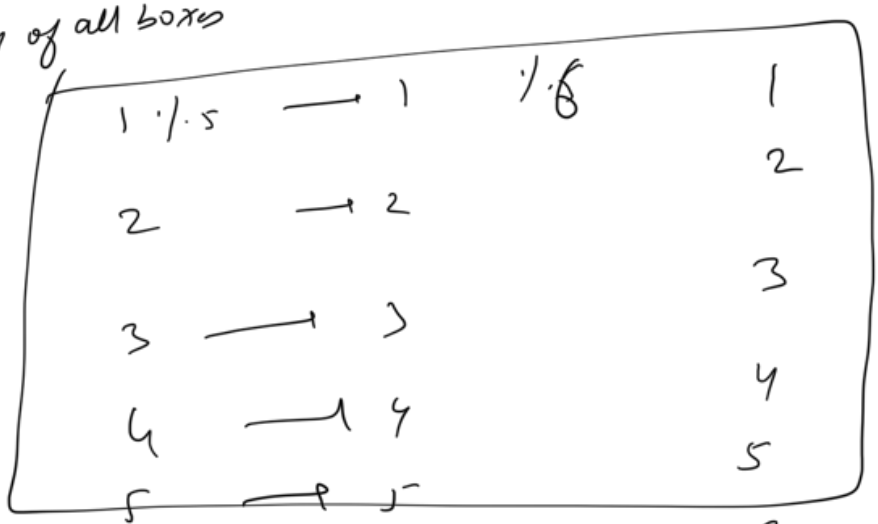
→ LB

Downside

On server inc/dec

you will have %6

to move state of all boxes internally



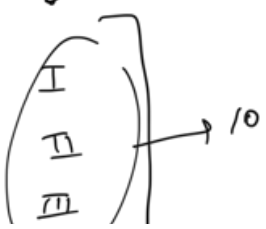
in LB

③ Ranges

1 - 1000

1001 - 5000

1 - 10,000

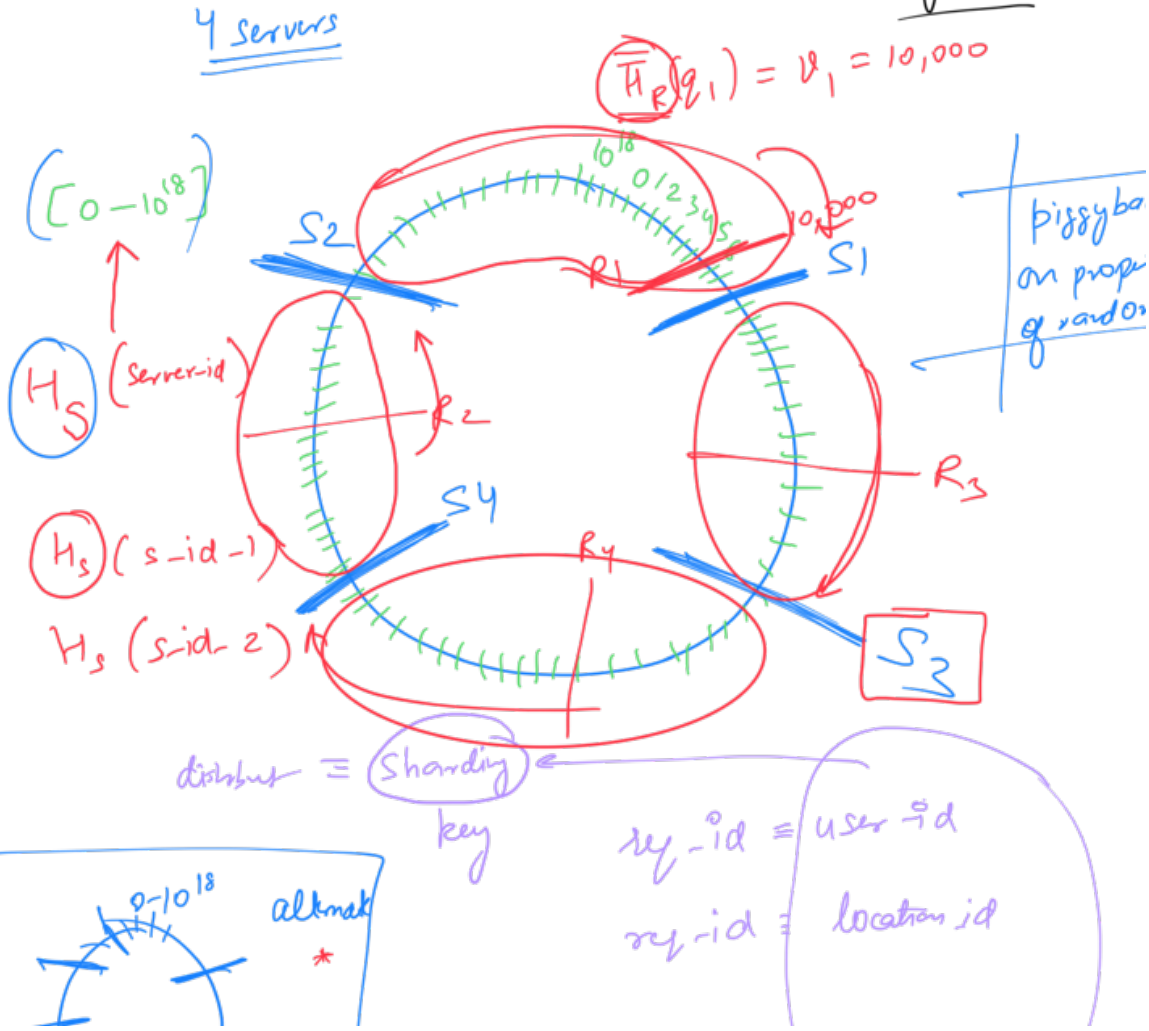


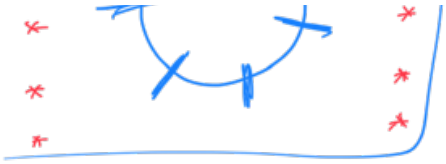
Bottleneck: Add / Removal of servers is inefficient

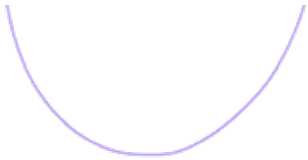
4 Consistent Hashing

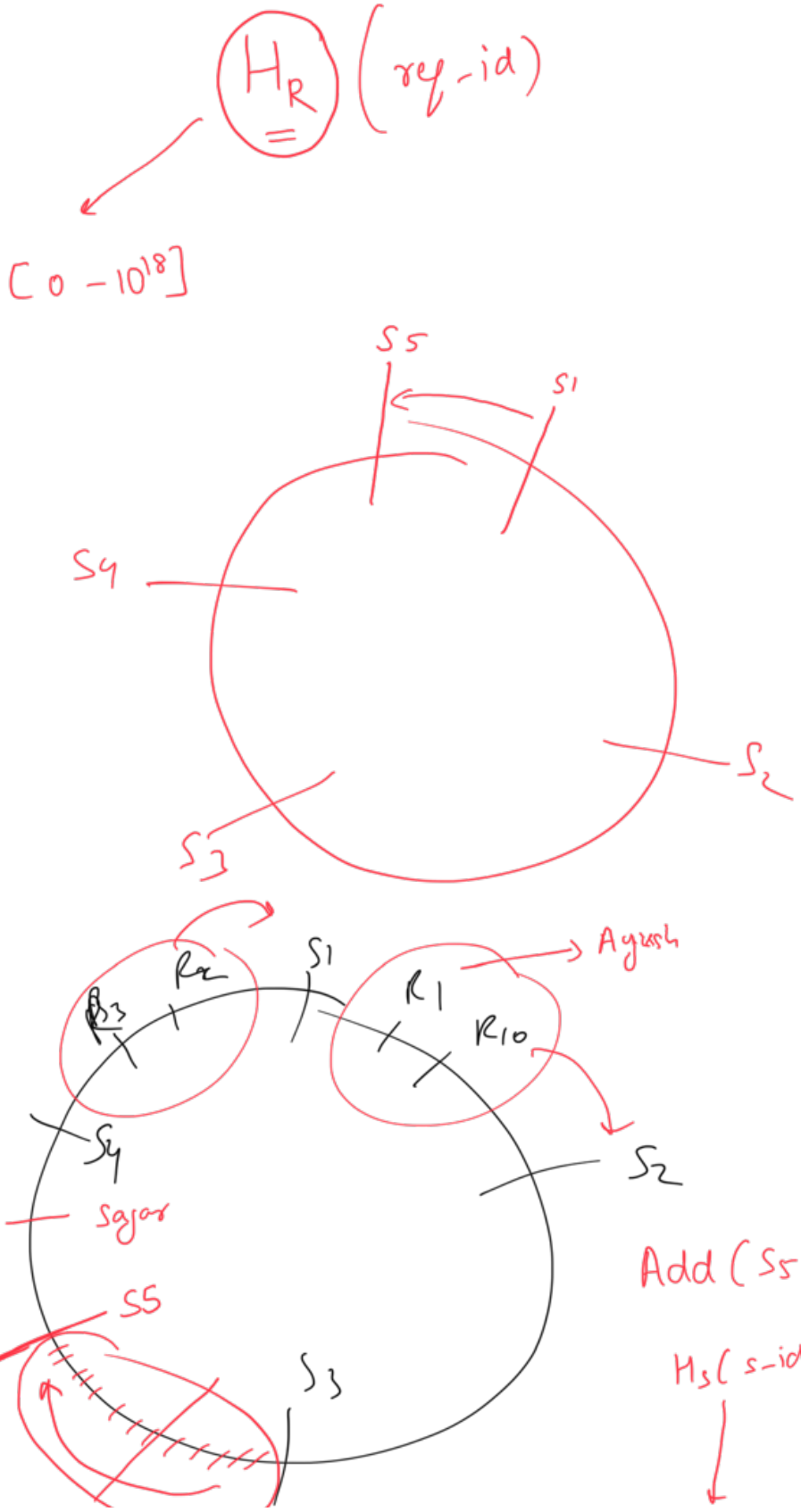
LB for Stateful Applications

const addition/removal of servers

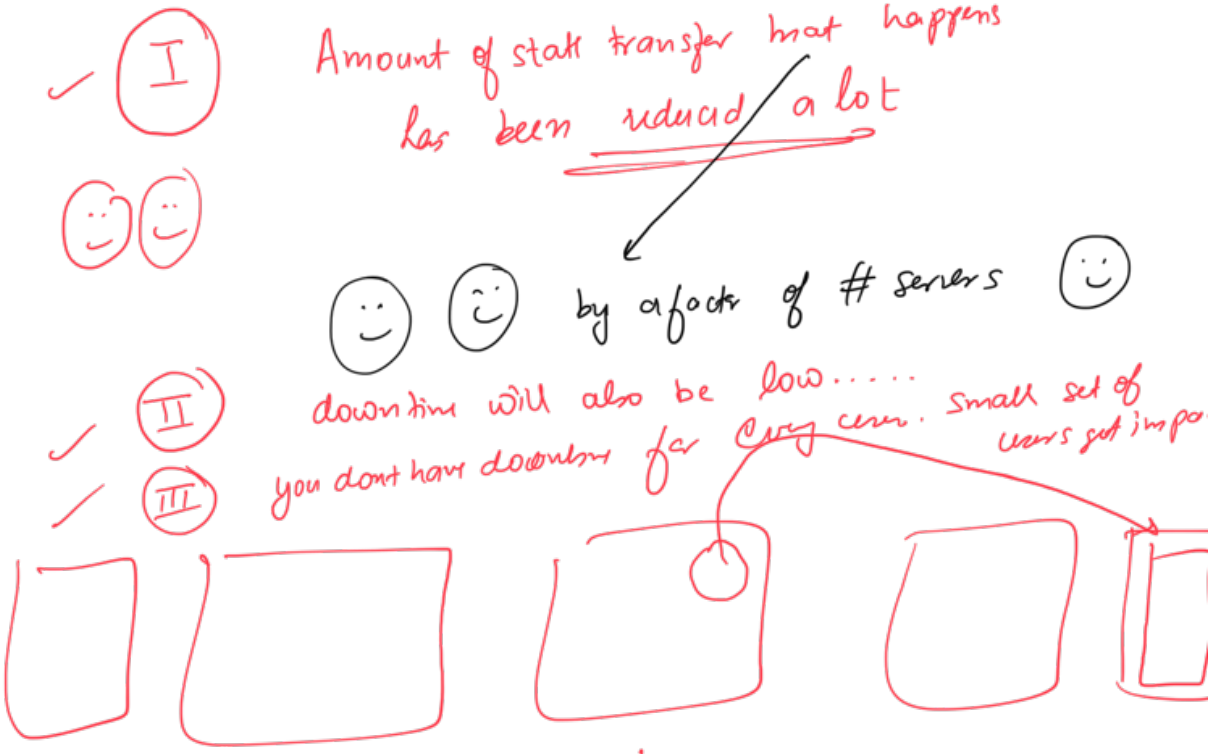


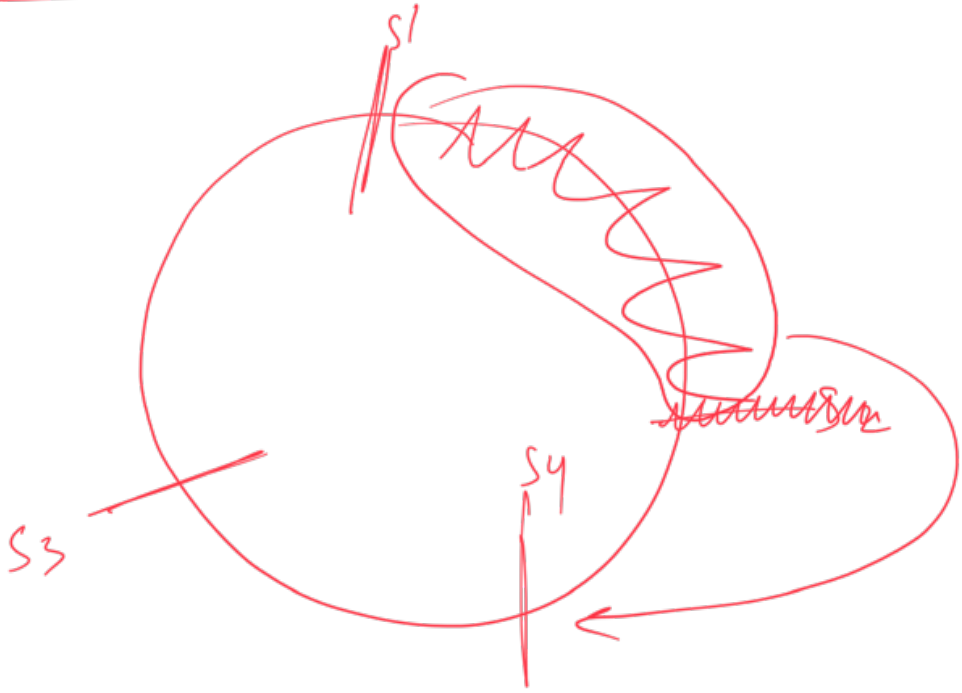




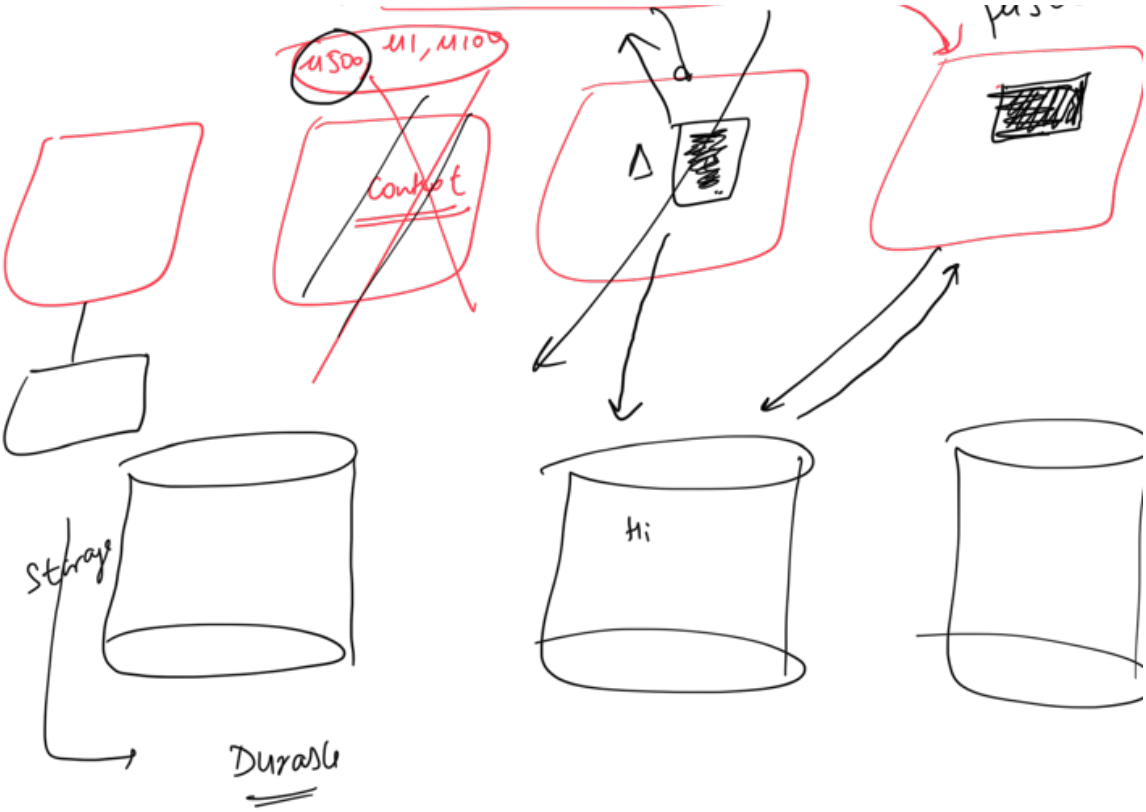

H_R (req_id)
[0 - 10^{18}]
S5
S1
S4
S2
S3
R2
S1
R3
R1
Ayush
R10
S4
S2
Sagar
S5
S3
Add (S5)
H_S (s_id, 5)

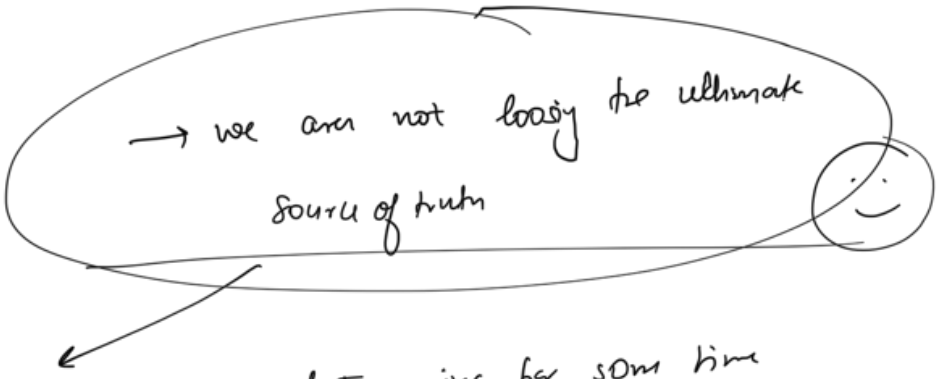
Lokesh


I
Amount of state transfer that happens has been reduced a lot
by a factor of # servers
II
III
downtime will also be low.....
you dont have downtime for every user. small set of users get impa


S1
S3
S4


LB





(A) cold start ≡ request's latency inc for some time

Hi

Hello

Q1

A1

Q2

A2

Online Learning Algo's ≡ model

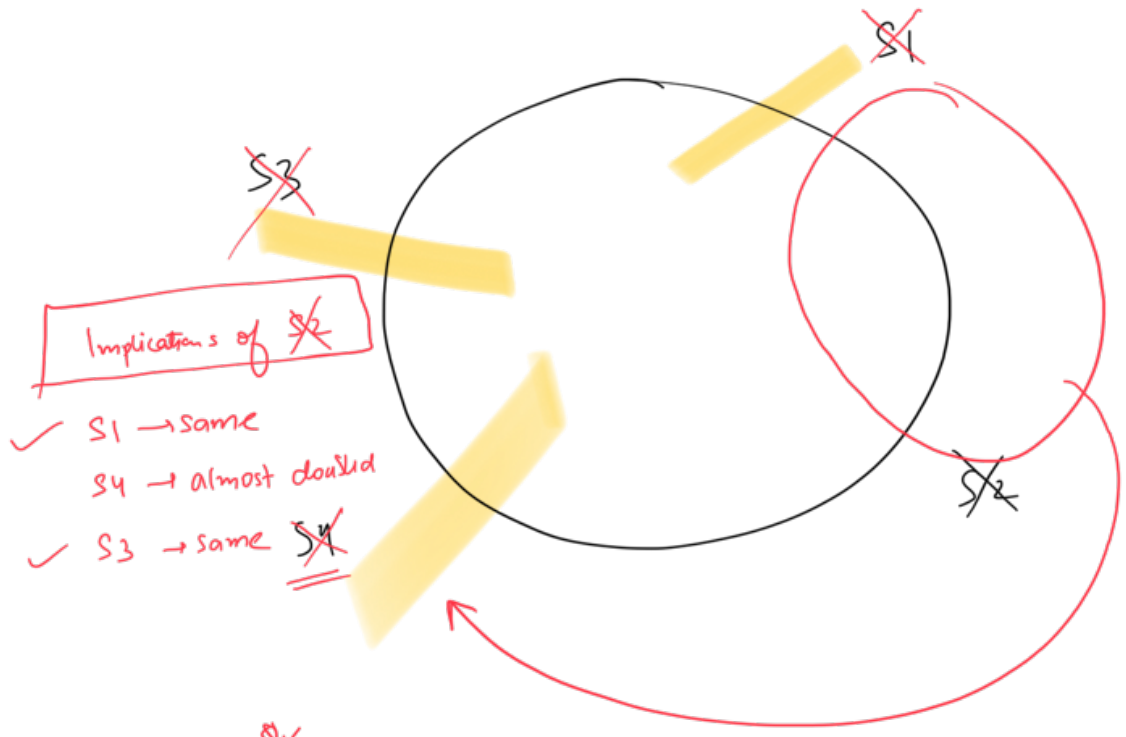
(B) pre-emptive copying

(I) ← Storage layer → new App server

(II) ← Mslave

Solved problem

'impact of server addition or removal is minimized



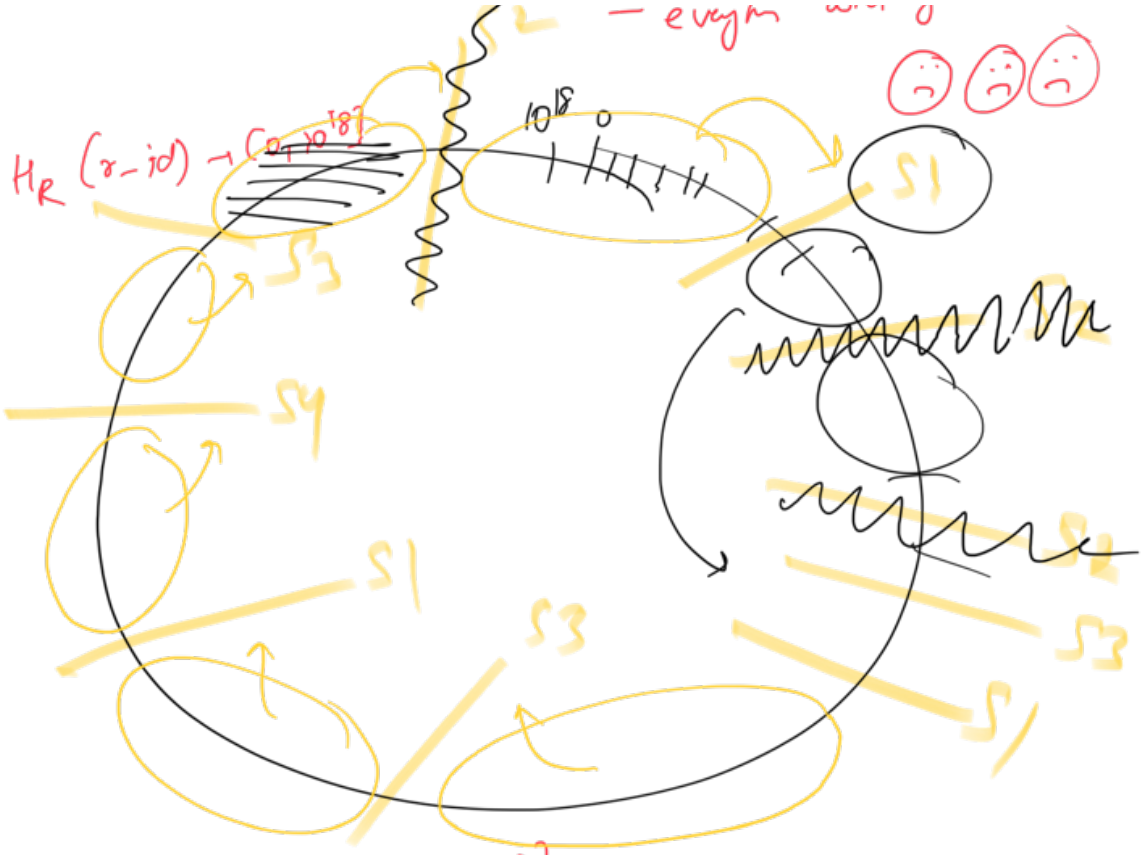
Implications of S2

- ✓ S1 → same
- S4 → almost doubled
- ✓ S3 → same S4

S1 ≡ same

S3 ≡ 3X

"Cascadiy Failures"

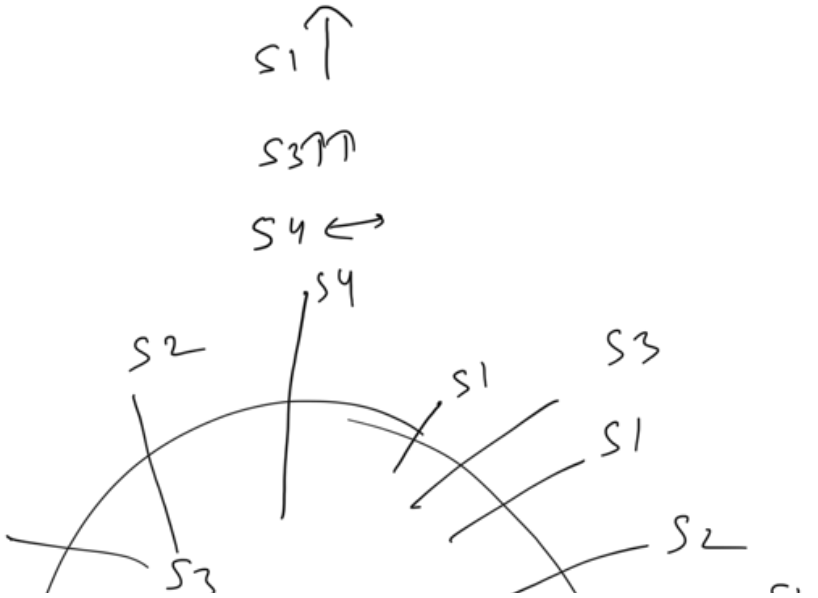


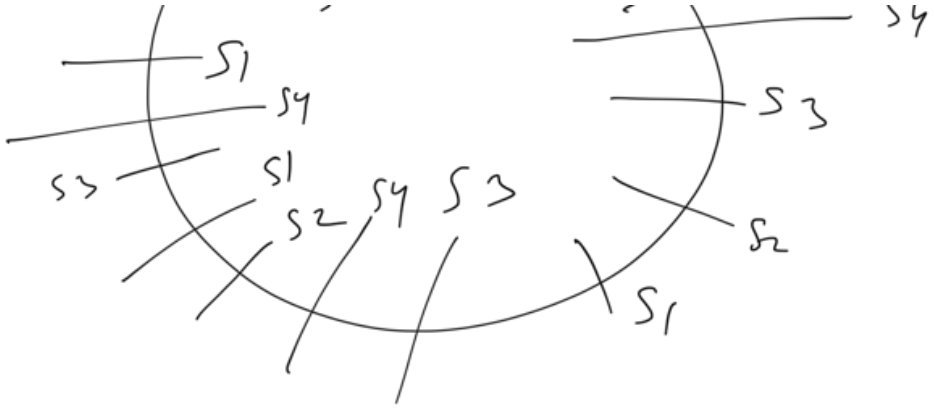
$H_S^1 (s\text{-}id) \to [0, 10^{18}]$

$H_S^2 (s\text{-}id)$

$H_S^3 (s\text{-}id)$

$|servers| \times |H_S\ func|$

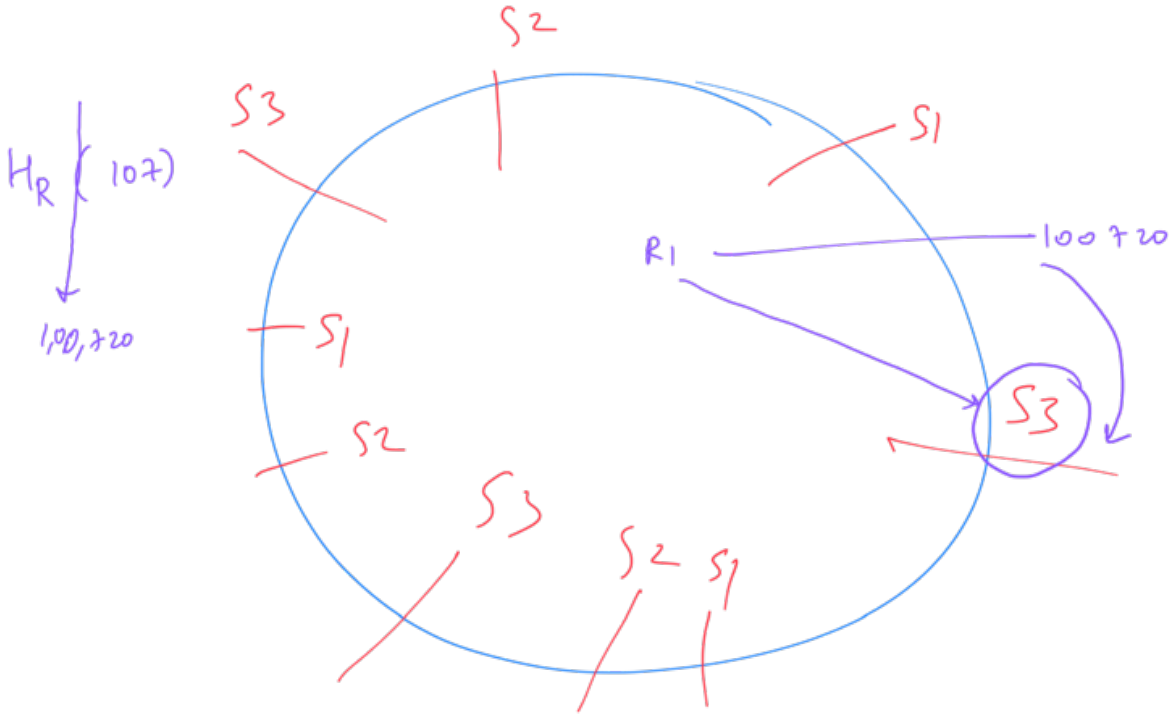




solved

Cascading Failures

[Now, the load will get divided almost equally.



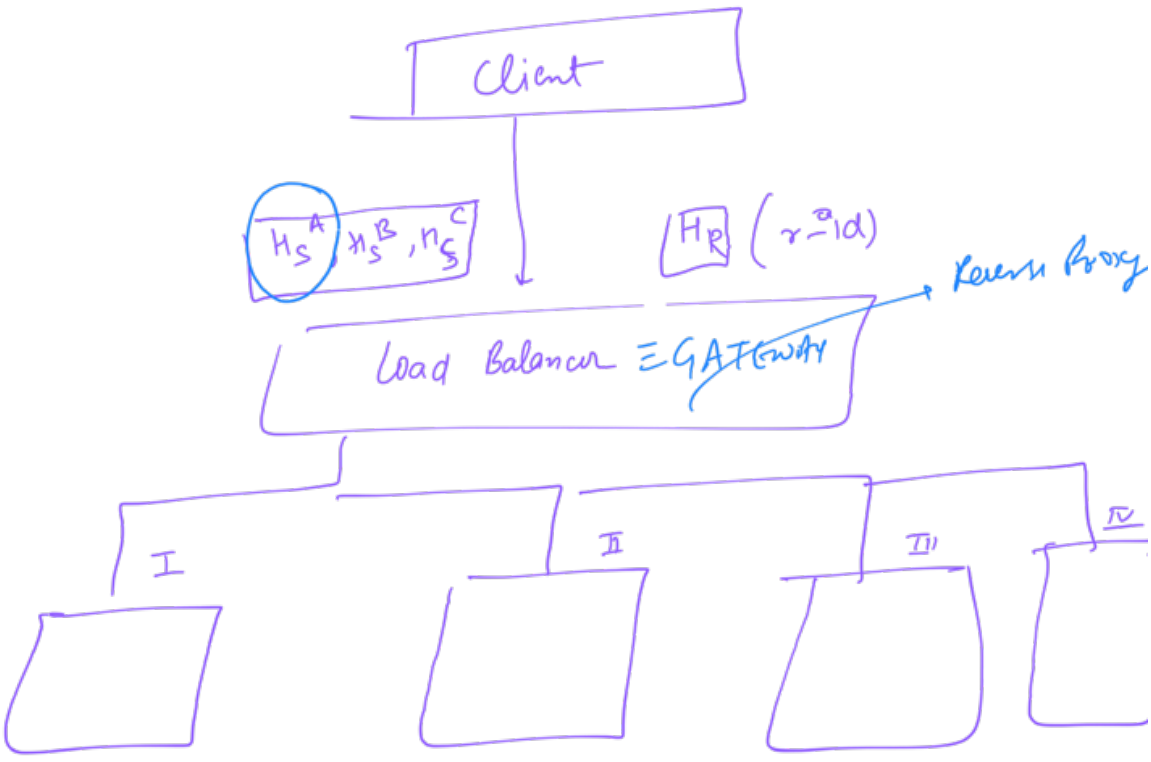
$|H_S|$ ↑↑ ≡ Randomness ↑↑

↓

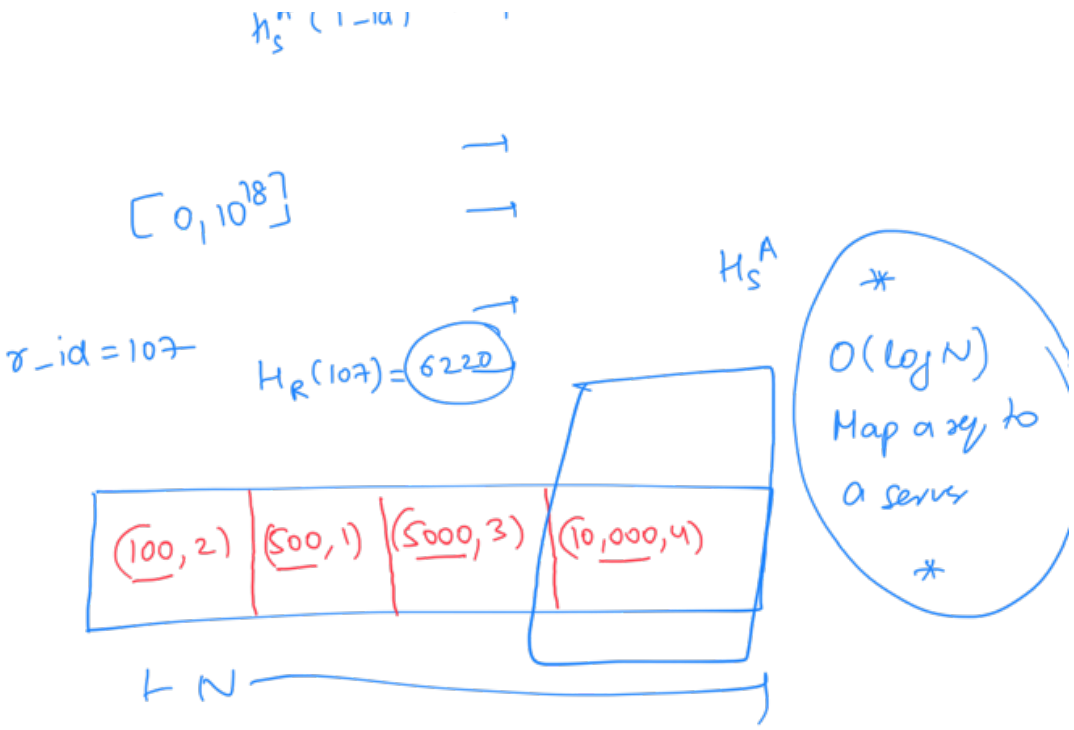
Equitable Distribution ↑↑

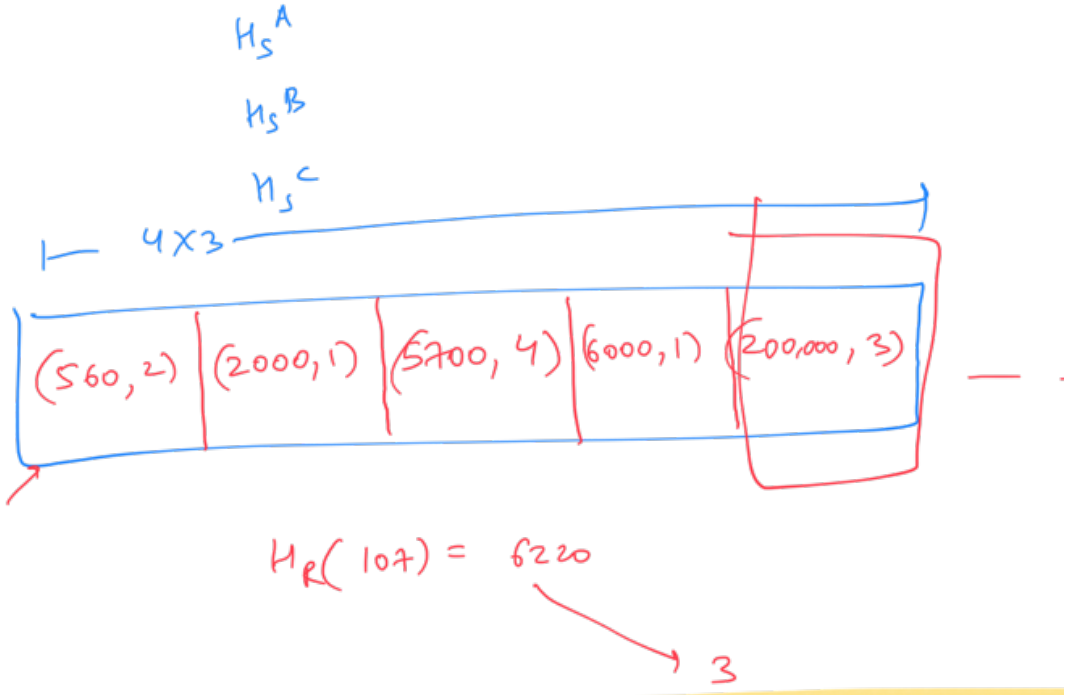
(H_S) TTT ≡ Time for Resolution / TTT

ASG



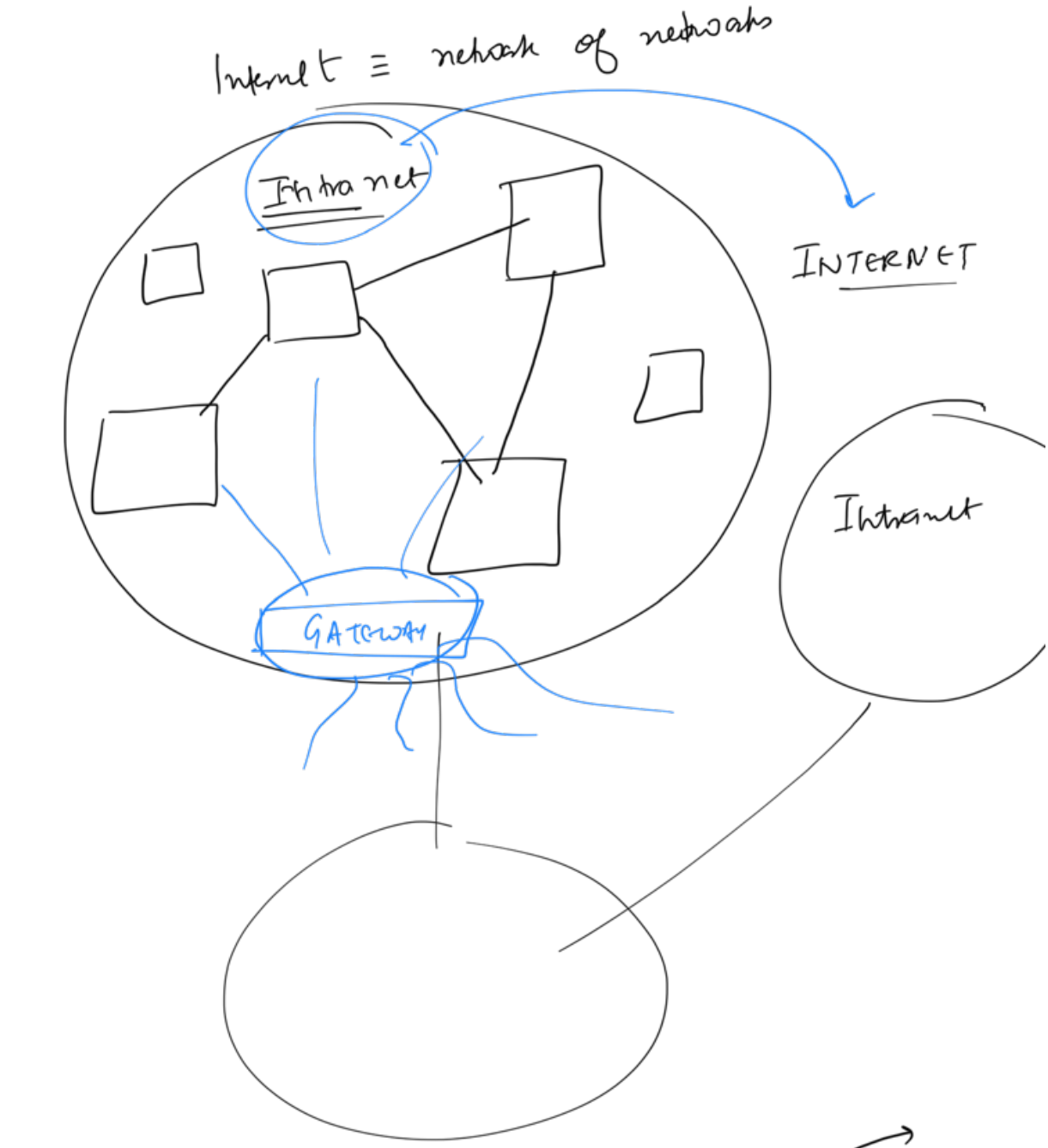
10 servers

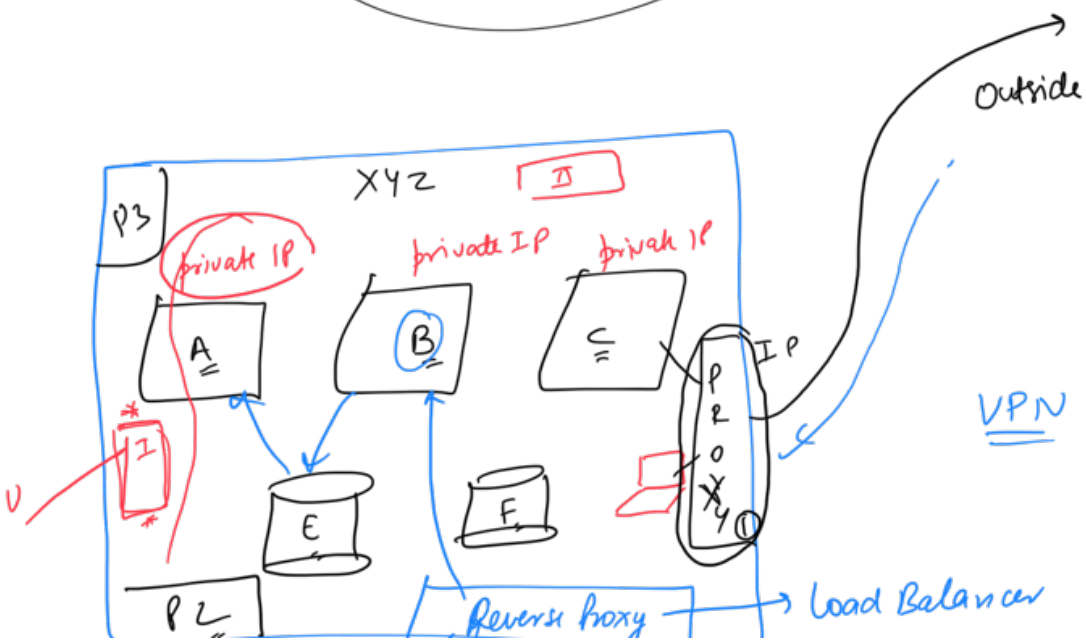




High Level Design

OSI Layers

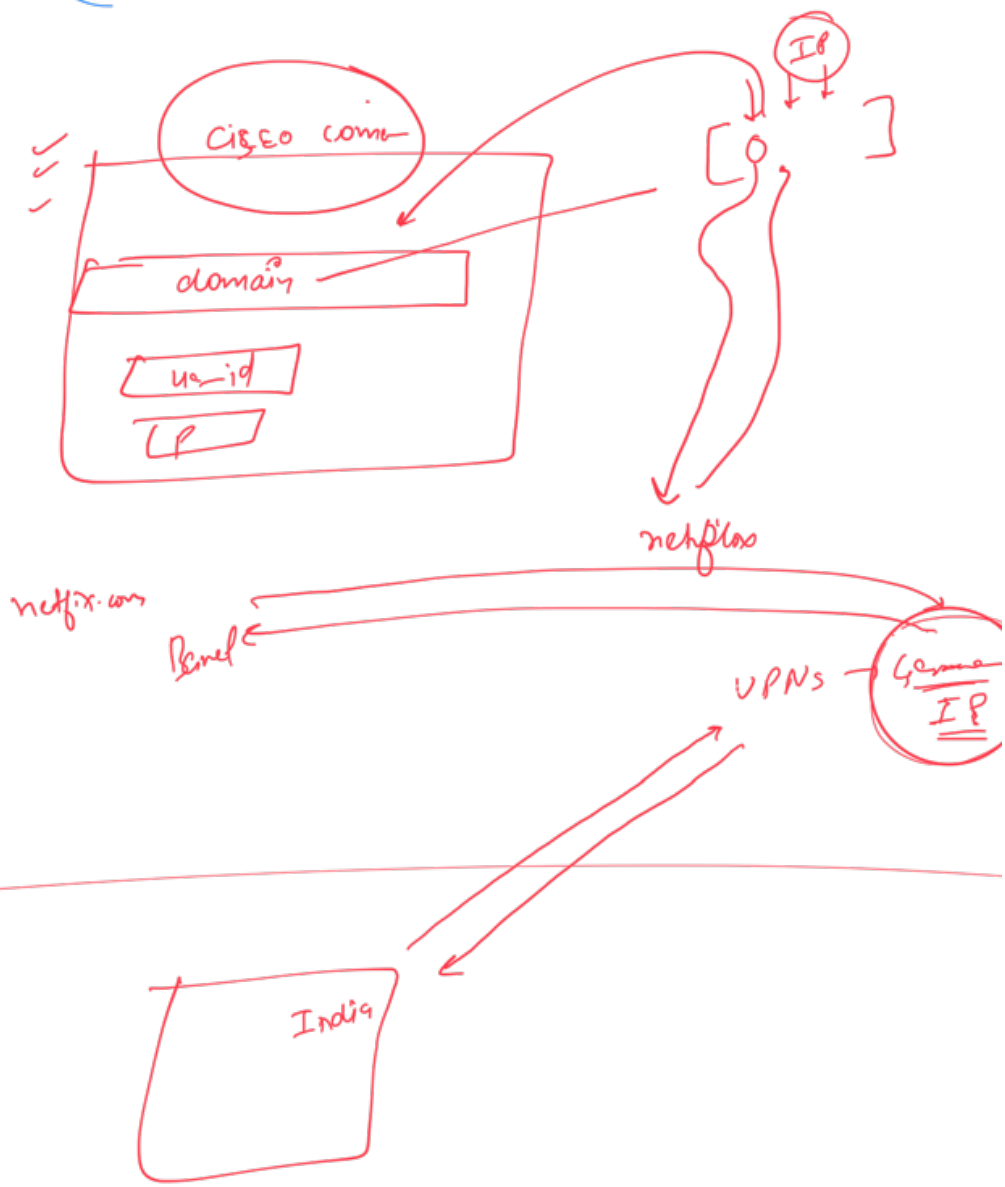

Internet ≡ network of networks
Intranet
INTERNET
Intranet
GATEWAY


Outside
XYZ
P3
private IP
private IP
private IP
A
B
C
IP
Proxy
VPN
U
I
E
F
P2
Reverse Proxy
Load Balancer

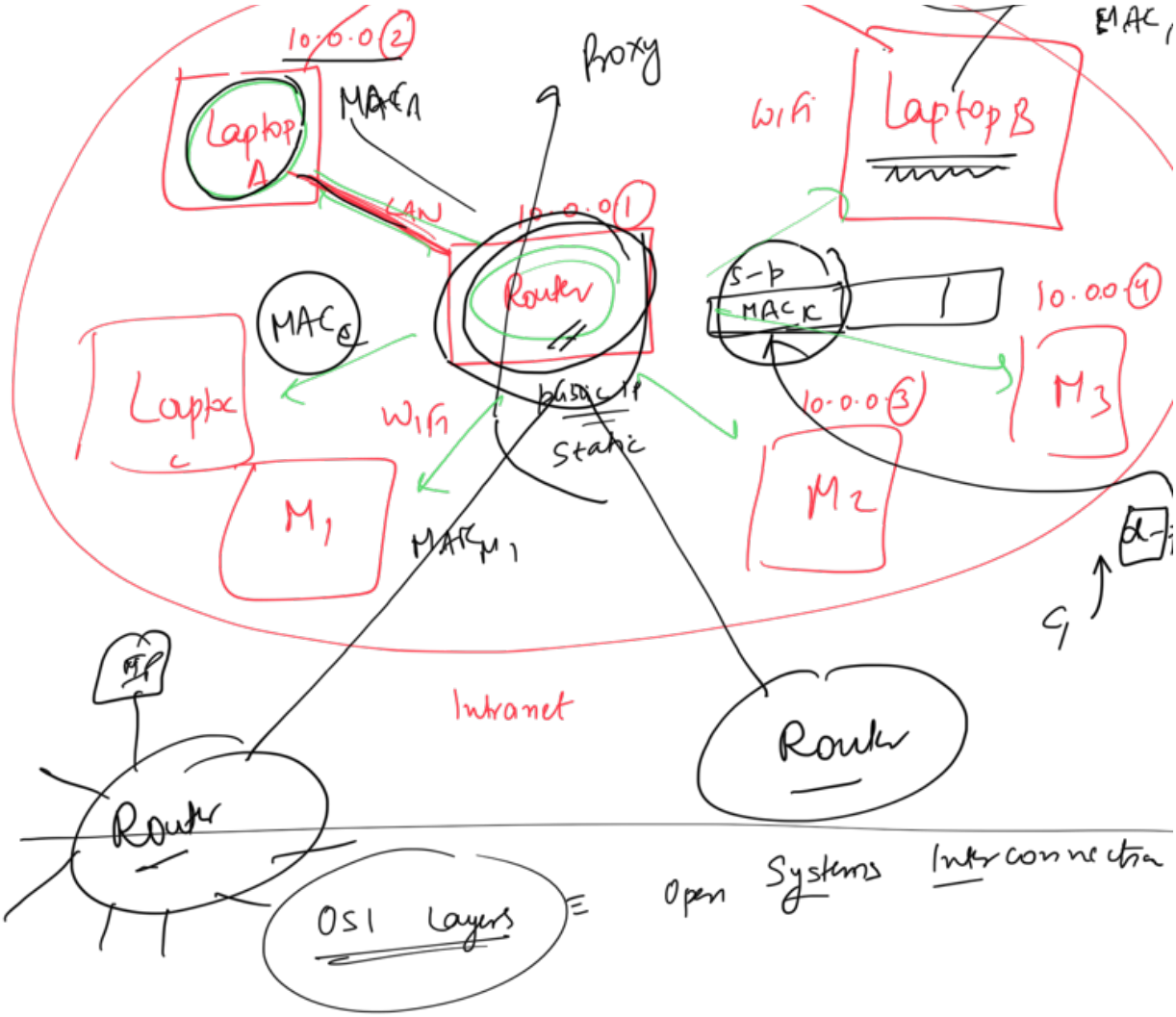
Gateway

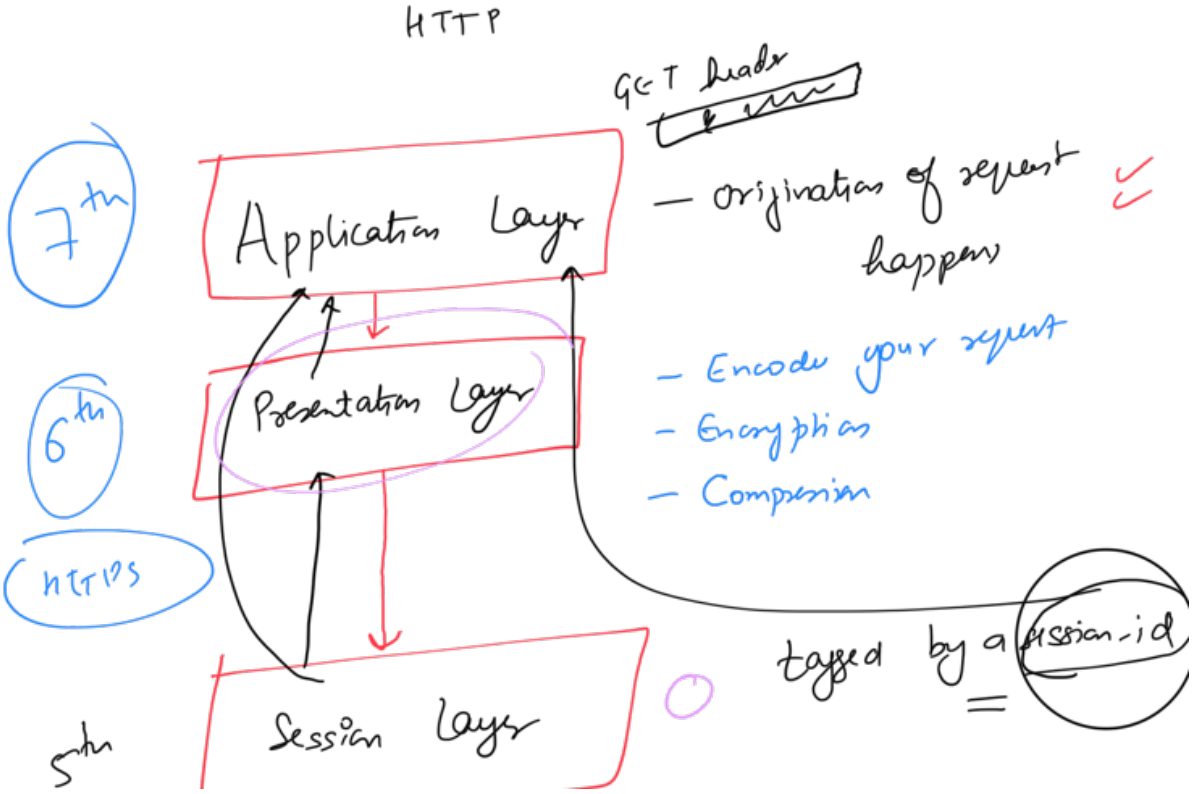
Proxy ≡ Gateway to the outside world

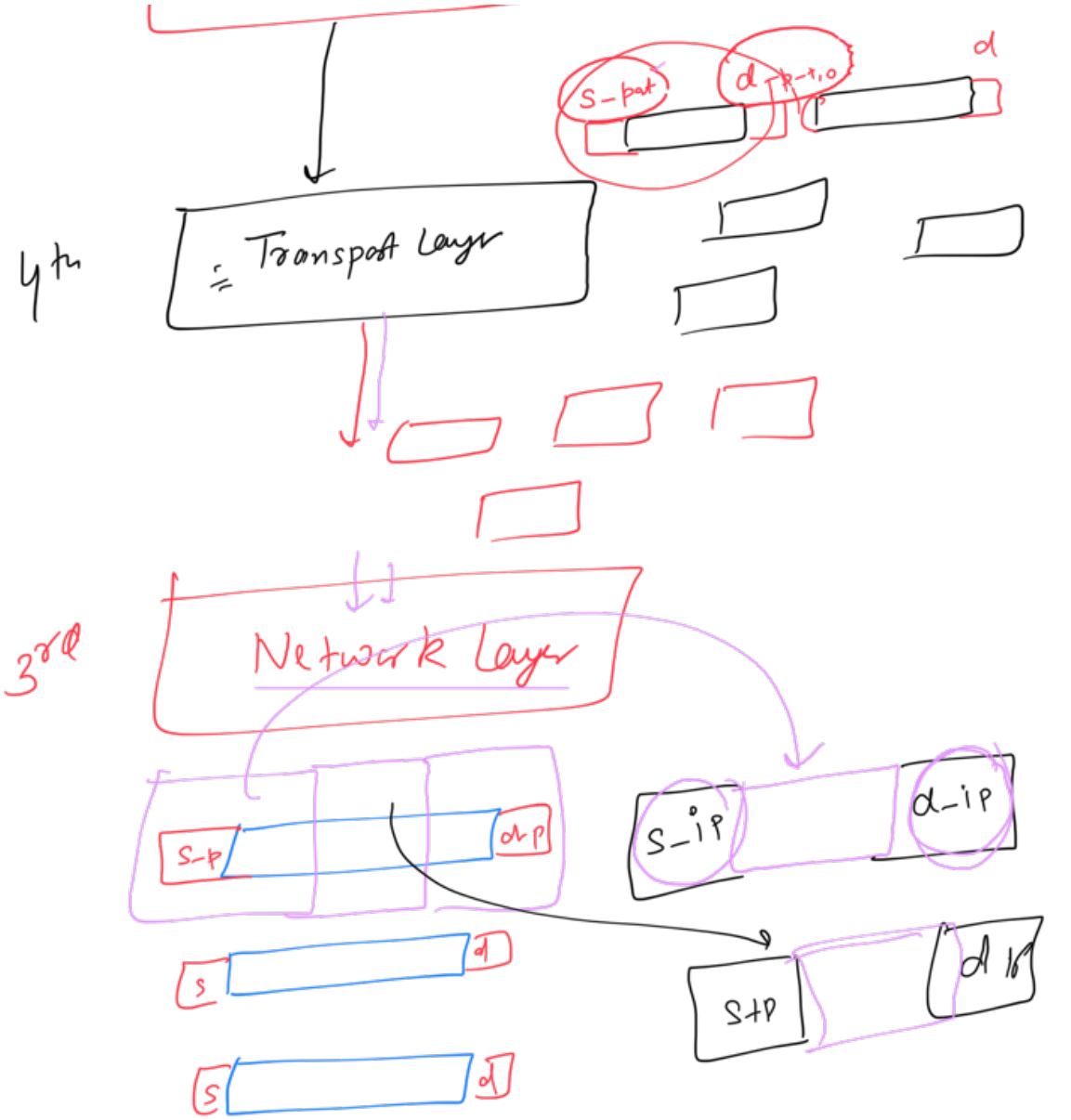
Reverse Proxy ≡ Gateway to the inside world

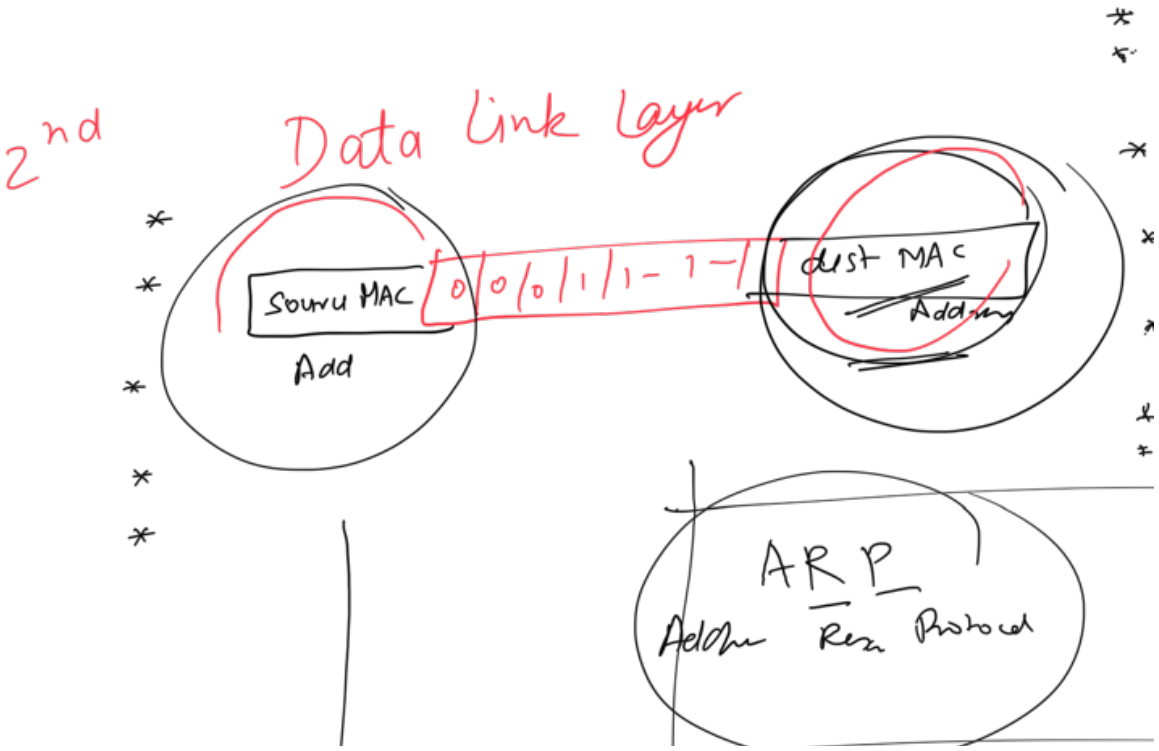


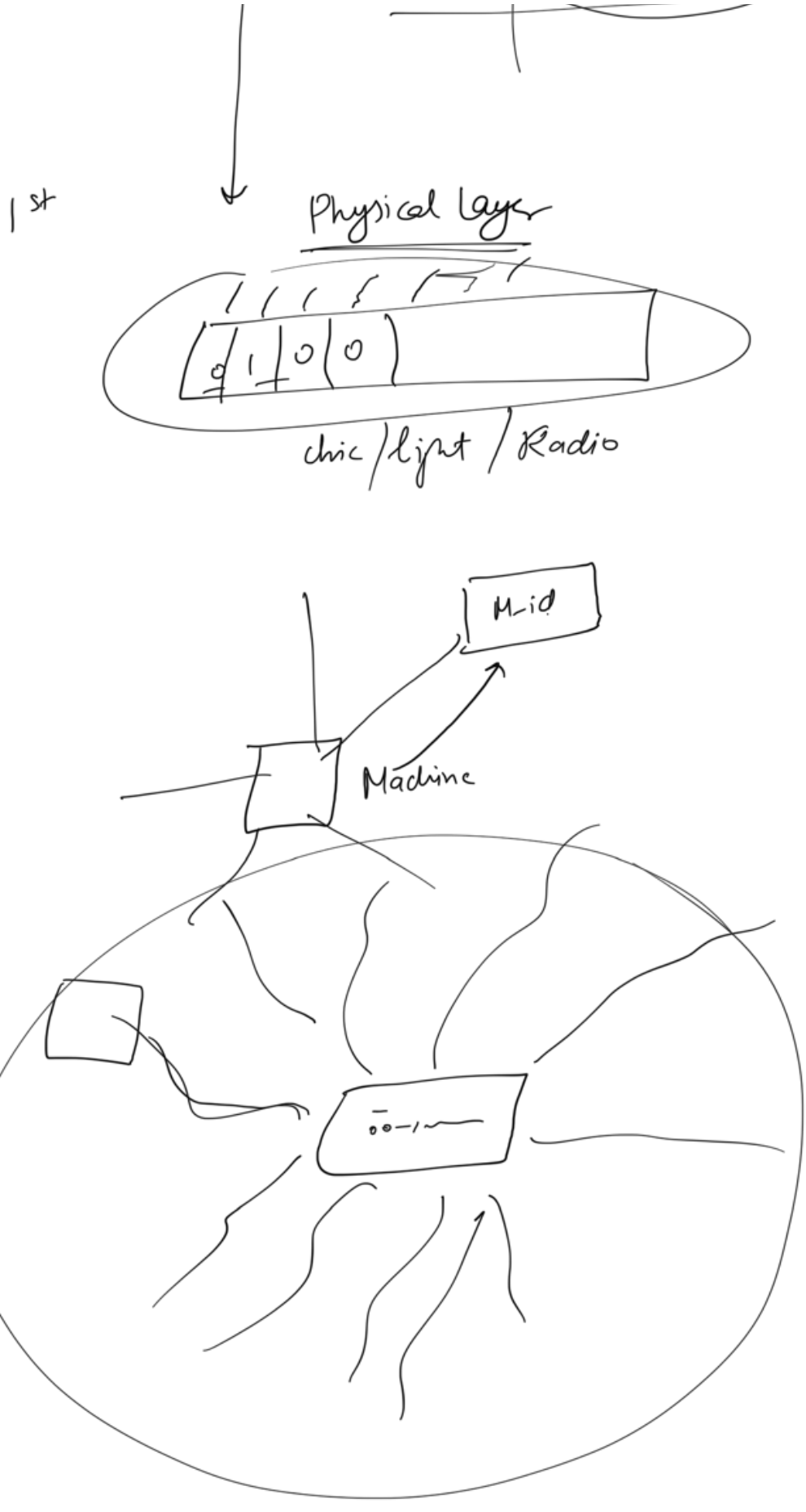
host

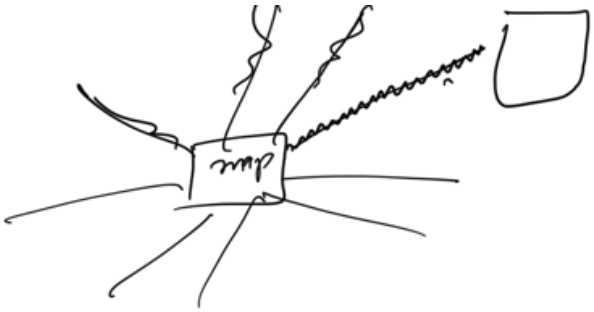

10.0.0.2
Laptop A
MAC A
Proxy
LAN
10.0.0.1
Router
MAC B
Laptop C
M1
Wifi
MAC M1
Static
Wifi
Laptop B
S-P
MAC C
10.0.0.4
10.0.0.3
M3
M2
Intranet
Router
Router
OSI Layers
Open Systems Interconnection

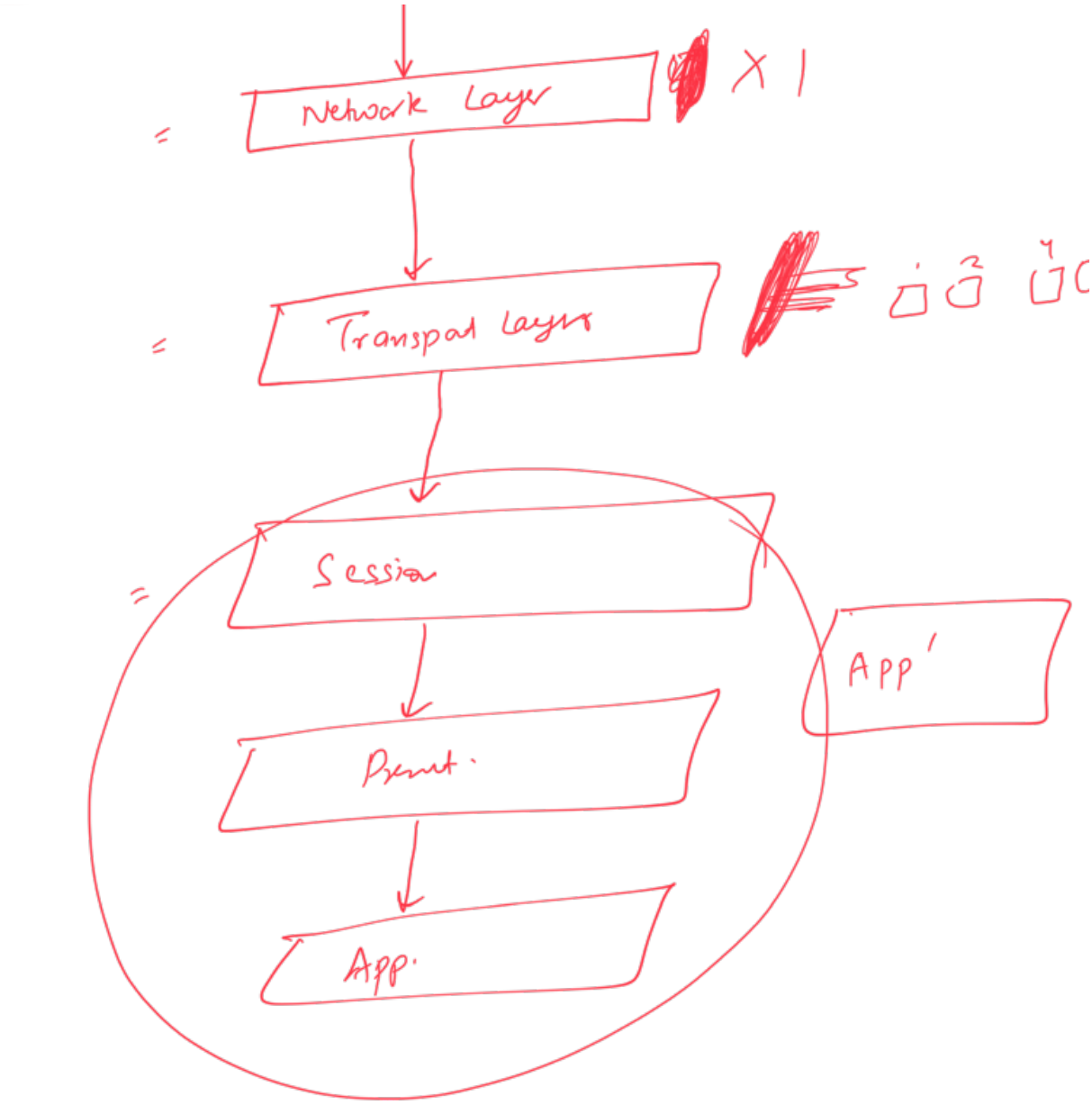

HTTP
GET header
7th
Application Layer
- origination of request happens
6th
Presentation Layer
- Encode your request
- Encryption
- Compression
HTTPS
5th
Session Layer
tagged by a session-id

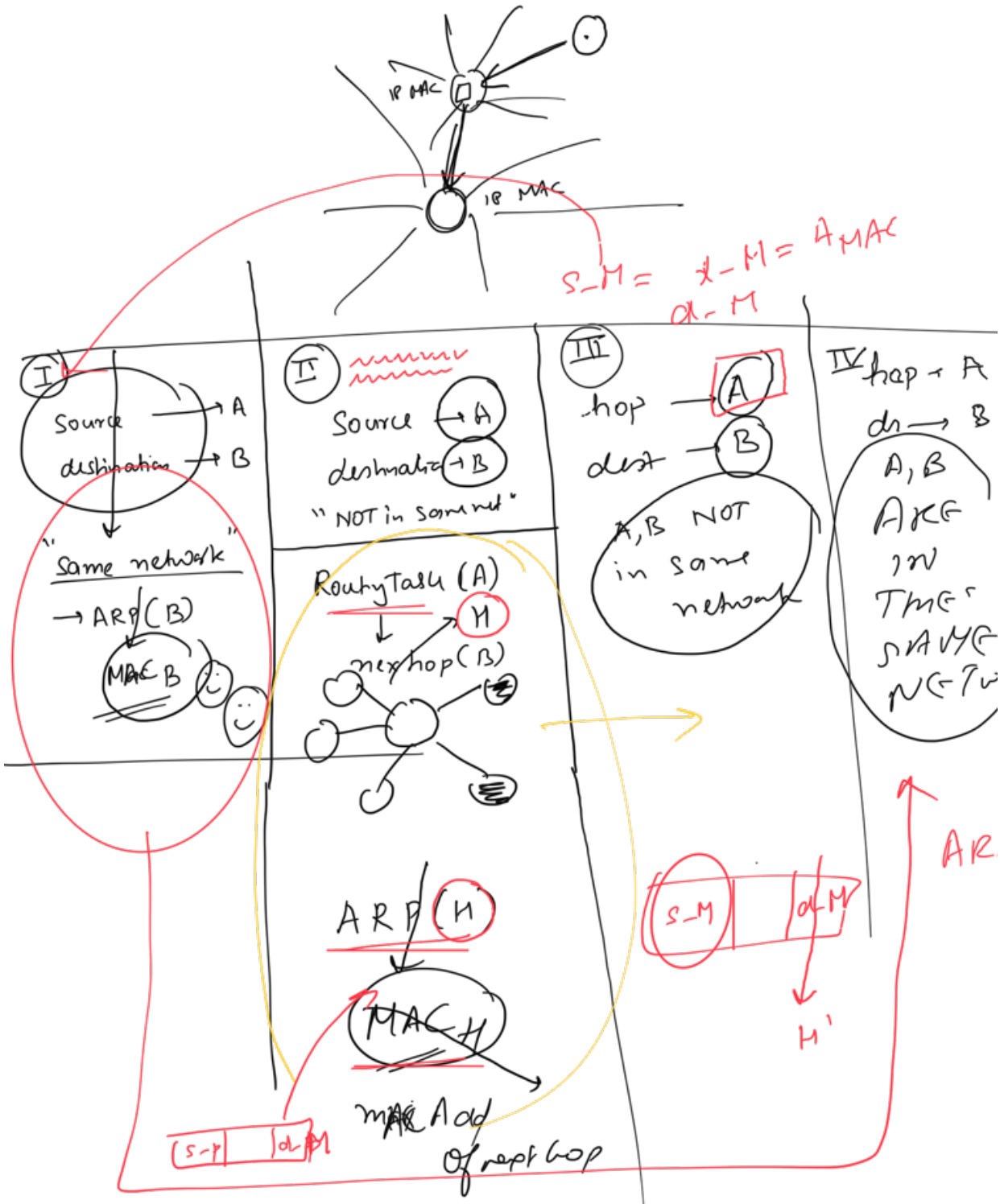

4th
Transpost Layer
S-port
d
3rd
Network Layer
S-p
d-p
S_ip
d_ip
S
d
S
d
S+p

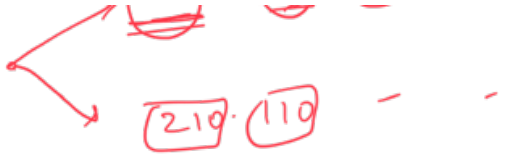

2nd
Data Link Layer
Sourv MAC
Add
dust MAC
ARP


1st
Physical layer
0 1 0 0
chic / light / Radio
M_id
Madine


Application
Presentation
Session
Transport
Network
Data Link
Physical Layer
Client
7
6
5
4
3
2
1
devices
Physical Layer
Data Link Layer
X N
X N
Dest MAC No
=
=
Your MAC No


Network Layer
Transport Layer
Session
Pres.
App.
App'


IP MAC
IP MAC
S_M = * - M = A MAC
d - M
I
Source
destination
A
B
"Same network"
ARP (B)
MAC B
II
Source
A
destination
B
"NOT in same net"
RoutyTask (A)
H
nexthop (B)
ARP (H)
MAC H
MAC Add
of nexthop
S-M
d-M
III
hop
A
dest
B
A, B NOT
in same
network
S_M
d_M
H'
IV hop - A
dr → B
A, B
ARE
IN
THE
SAME
NETW
AR


210
110