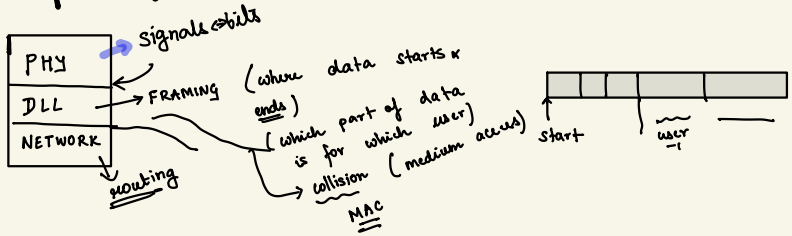
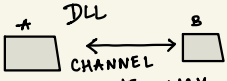
Class Notes

29/07/24 – Lec 1

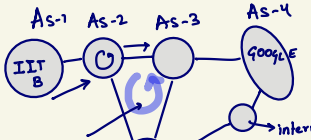
REF. TEXT : PETERSON & DAVIE
COMP NETS , SYSTEM :

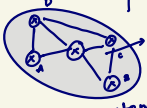
30/07/24 – Lec 2



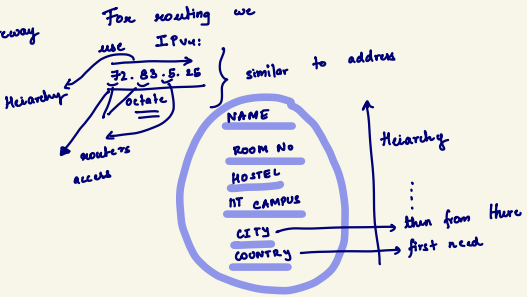


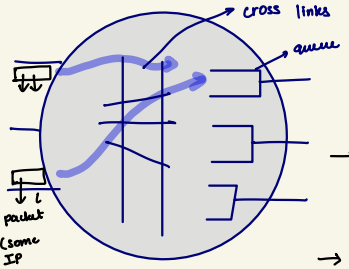
SIMPLEX: ONE WAY COMMUNICATION (eg radio, TV)
FULL-DUPLEX – (communicate in both directions simultaneously) → (a can send to b & b can " " a)
HALF-DUPLEX (com.. " " " but not ") (only one signal a → b or b → a a one time)
↳ WIFI (read why)





outside these autonom... systems border gateway protocol → BGP
tells other AS (autonomous system) which AS it can reach





Incoming packets mapped to same O/P link:-
so we need queues !
So we can use FIFO
→ But if some client pays more we give priority to his message
like prq :-

→ If queues is filled and extra packets come :
These packets are dropped (data drop)
(CONGESTION CONTROL) :-
↓
So ARPANET DID they will only handle this issue at end nodes not intermediate nodes and only for L4 (messages)

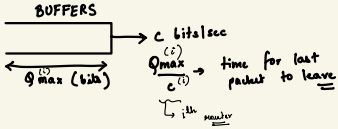
MESSAGES :
L2 : FRAME (1 unit of DATA)
L1 : SYMBOLS (Like Napolean).
L3 : Packet
L4 : (TCP) → segment
(UDP) → data gramc

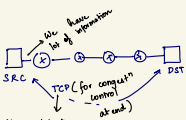
End to end delay of pkt (SRC to DST)

$\leq \sum_{i=1}^{n} \frac{Q_{max}^{(i)}}{c^{(i)}} + \sum_{i=1}^{n} d_i$ + transmition delay

d_i : speed of light delay

BUFFERS

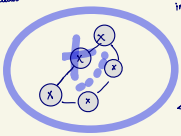




this detects congestions (pkt drops)
& slows down . (So rate of incoming PKts falls down)

So we need to use end to end delays because delays could cause problems

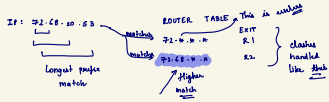
For some companies :
They do not send data through only path, because what if this path fails:



interior gateway protocol

distance way to handle congestion is keep updating weights of edges so to some destination it takes path with less congestion :-

↓ dijkstra



L4 : Transport layer

TCP : Transmission Control Protocol → Congestion control
→ Reliability
↓ data transfer
} this checks for reliability

UDP → Does nothing

LAYER-5 : Application layer
↳ web (HTTP)
Email (SMTP)
VOIP (voice over protocol)
(UDP over TCP)
Live do not have time for delay, sometimes need low delay)
TEXT MESSAGES
P2P...

Layer 4 :

Tcp .. UDP - - -

L3 : IP (internet protocol)

L2 : WIFI , 4G, ethernet

L1 : Wireless , optic fibre, ---

Design Protocols In Modules

each subproblem handled by some protocol
We discussed above OSI-5 → APPL, TRANS, NETL, DLL, PL

not protocols

ADVANTAGES OF LAYERING/MODULARITY
(1) EASE of development (easy to debug certain sets of problems)
(2) Debugging / Maintenance
(3) Many applications & Technology (at physical layer) working together :-
IMAGINE IF WHATSAPP FIXED TO WIFI ONLY !!
(4) If we want make changes at one layer we do not affect other layer.
(5) case of modification - only change 1 layer to address a problem.

Protocol Layering

PHY: wifi PHL, cach has its own PHL
DLL: ← wifi, ← Bluetooth, ← 4G, ← Ethernet
NET: IP (5-OSI) — Glue of the internet
TRANS: TCP, UDP
APPL: email, whatsapp, Youtube, :

Because of IP we can have many time PHL + DLL.

TCP → RFC (req for comments)
→ This is how initially TCP was called. So while making a TCP we have to satisfy old RFC and then we can add our own protocols !!

For wifi the protocol is IEEE 802.1|6|7|n|ac.....
→ standards as standards increase rate of data increase

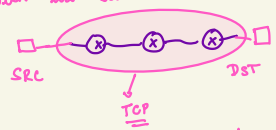
Advatages Of Layering:-

(1) Developement & Debugging (EASE)
(2) Easy to modify one layer without breaking the system. (we can work with one layer and not worry about the others).
(3) Can have different choices at the different layers :- (compatibility)

Disadvantages

(1) DEV 1 — DEV 2

Higher layer do not communicate with low levels :-

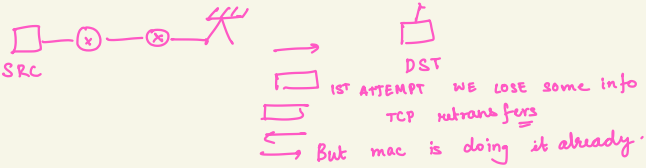


Let us say a pck is droped
It assumes that some routers in between must have had a queue full and this could not be reason, it just makes intelligent guess as these routers cannot communicate much with TCP. This drop could be due to some interference but TCP makes a wrong guess!

(2) Disadv

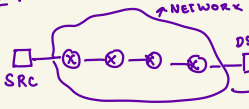
Redundancy of Tasks

(1) TCP : Handles retransmissions (Reliable) ↑ (Because we TCP make sure = no pckt drop)
But on the side mac retransmission is also going on



(3) SUBOPTIMALITY:-

Ex: VOIP (lowest time delay)



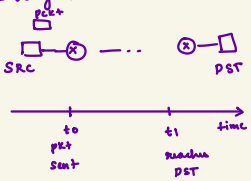
We do not have choice of choosing our path. We get what network gives ! No guarantee on QoS (Quality of Service : Amt of pkt drops, Latency (Delays)

we could have some other network by which both could have less latency. But this do not know!

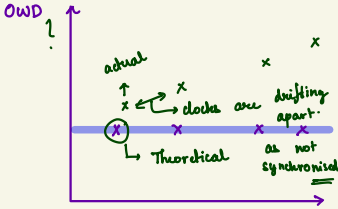
↑ known as BEST - EFFORT.

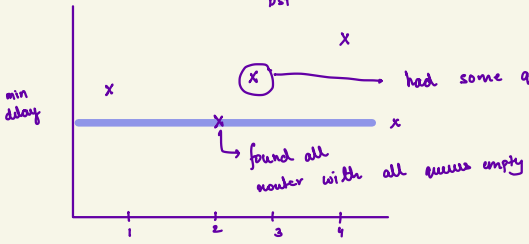
Latency Metrics:-

① One-Way Delay:

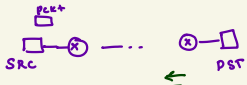


first issue:-
The clocks are not synchronised as they are not atomic clocks

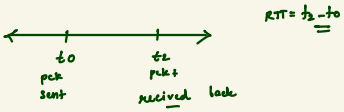




② Round Trip Time



pck sent (acknowledgement)



In VOIP we want ~few 100 ms at most

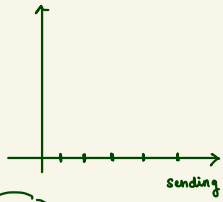
→ JITTER : variability in OWD latencies:-

Pkts 1, 2, ...

OWD $d_1, d_2, \ldots$

$e_k = |d_{k+1} - d_k|$ - jitter

Avg jitter = $\frac{1}{n-1} \sum_{i=1}^{n-1} e_i$



I want the spacing between packets to be same!

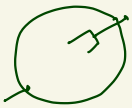
Due to delays problem ho jati hai
→ One way communication mein problem hai
Jitters ke karan quality ki maa bhen hajati hai
video quality is bad because some information should come after it we less spacing now has more.
Example in tennis (Remember Cholapur)

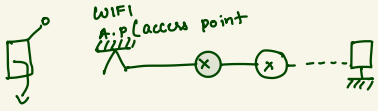
Telephonic Network

↳ Voice → This was the single application. We had very low delays, no loss, low jitter, low RTT, low OCD

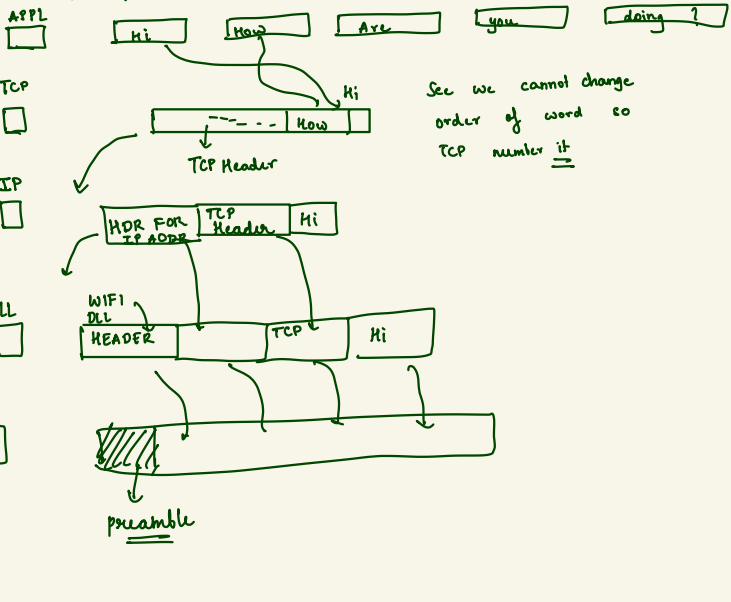
The wire telephones ?

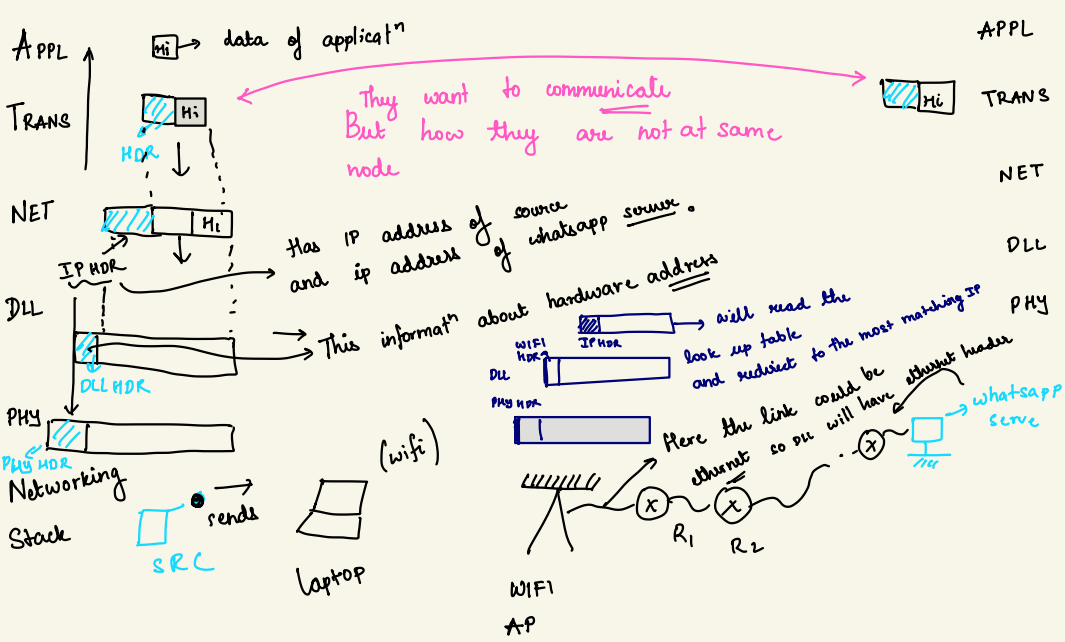
We had to make sure data entering every router should exit (without loss





whatsapp





* Both laptop and wifi are in hearing range of this channel.
So PHY layer of both take this message
But DLL of laptop rejects this message due to Hardware address in DLL not ment for Laptop.

The DLL header depends on technology used
- Here we have WIFI Header

Remember this encapsulation and decapsulatn happening till it reaches final IP, then completely de encapsulates.

- So if we want to send secret info, the wifi server should only have encryption key and you should not just "Hi" send encrypted version.

QoS: Quality of Service
Pkt loss (Drop) rate :-
Latency (OWD, RTT, Jitter)
Throughput

SLA service level aggrement
- it gives conformation about its own networks servia
- 99.9% ↓ Non-drop rate
- 30ms ↓ latency across AS
- See understand this network finds n sends message to the exit node in 30ms

Throughput: Data rate from a sender to a receiver per unit time

We care because sometimes we need to push a large data

Youtube — 10 Mbps — Video quality 4K

decrease of quality ↓ — Throughput

1 Mbps

We do not care for OWD as in start we can have a delay, but then things work smoothly. But we care about jitter because we want to avoid buffering.

VOIP: Throughput ?

↓

Claude Shannon

↳ Information Theory

Entropy: minimum number of bits required to encode the data

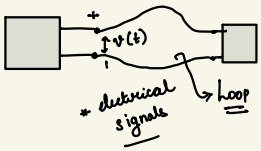
For voice we do not need much high throughput and if pkt drops then TCP backs off so we lose time

RTT ~ few 100 ms

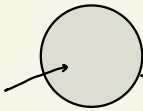
Loss ?

Physical layer

Wired Media



* electrical signals



$\oint \frac{dB}{dt}$ = induced voltage

* We want to minimise the area to reduce the induced voltage

So we use twisted pairs: -

area minimizes

Ethernet Cable

many twisted pair cables

There are different types of twisted pairs: - Supports

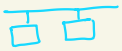
Cat 3: 10 Mbps upto 100m

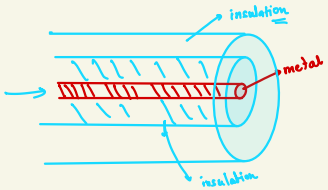
Cat 5: 100 Mbps " "

Cat 6: 1000 Mbps " "

Coaxial Cable: -

We can connect many PC's into it: -

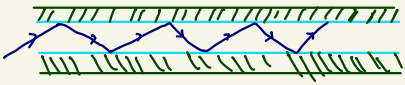




Thin Net Coax: 100 Mbps, 200m → 0.2 inch in dia

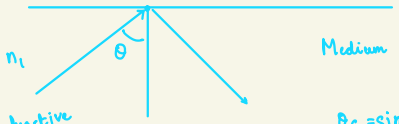
ThickNet Coax: 100 Mbps, 500m ↓ 0.4 inch in dia

Optic fibre



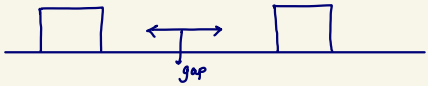
n_2

Medium 2



Medium 1

$\theta_c = \sin^{-1}\left(\frac{n_2}{n_1}\right)$



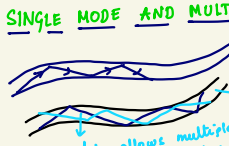
Single mode

↓

If I turn this laser on.

Only a single layer passes through :-

SINGLE MODE AND MULTI MODE

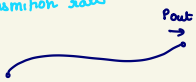


→ in single mode single ray can get through.

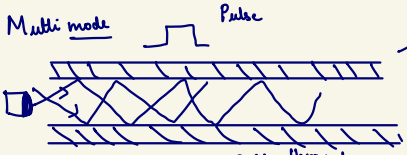
→ multi rays can get through

→ This allows multiple rays to get in.

OPTIC fibre have high distance transmition rate

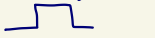


Multi mode

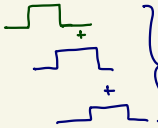


multiple layer can pass through

Single ray



(only a single pulse comes same as previous)



pulse of different layer get added

figuring out which is which is diff

So we can send much that through through single mode

Attenuation

$\frac{P_{in}}{P_{out}}$

We want to see the ratio :-

→ Tower emitting the power

Very little is reaching to phone so P_{in}/P_{out} very high

∴ we look at $10 \log_{10} \frac{P_{in}}{P_{out}}$ in dB

So A small diff ↑ in this high change in ratios

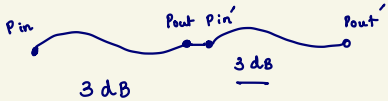
Ex $P_{out} = \frac{P_{in}}{2}$

Att $= 10 \log_{10} 2$ dB $\sim$ 3 dB

A 2 decrease in Pow is 3 decrease in Attenuation

Ex $P_{out} = \frac{P_{in}}{10}$

Att $= 10$ dB



3 dB

$\frac{P_{out}'}{P_{in}} = \frac{1}{4}$

Att 6dB

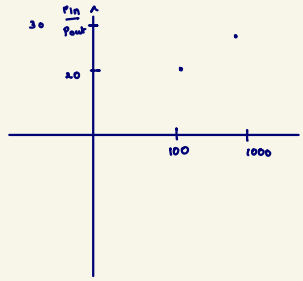
$10 \log_{10} \frac{P_{in}}{P_1} = 3$ -①

$10 \log_{10} \frac{P_{out}}{P_{in}} = 3$ -②

① + ②

$10 \log_{10} \frac{P_{in}}{P_1} \cdot \frac{P_{out}}{P_{in}} = 10 \log \frac{P_{out}}{P_1} = 1$

Pin / Pout	dB
2	3
4	6
10	10
!	!
100	20



Atten can also be thought in terms of Amplitude²

Power $\propto (\text{Ampl})^2$ → Ain, Aout

Atten → Ain, Aout

$L_s = 10 \log_{10} \left(\frac{A_{in}}{A_{out}}\right)^2$

$= 20 \log_{10} \frac{A_{in}}{A_{out}}$ dB

Absolute Power in decibel Scale

1mW as a reference

Power P (watts) in dBm

$10 \log_{10} \frac{P}{1mW}$ → $10 \log_{10} \frac{P}{10^{-3}}$

It sounds misleading but it means it emits 10^{-3} mW Power

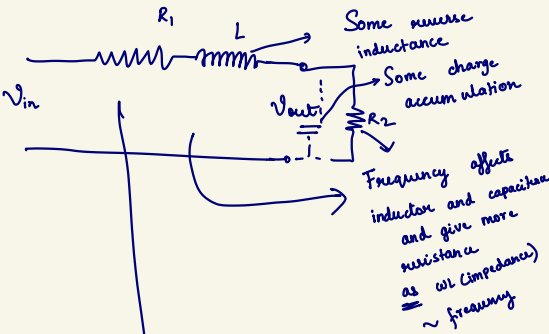
Ex

P = 1mW = 10^{-3} W

$10 \log_{10} \frac{10^{-3}}{10^{-3}} = 0$

2) P = 2mW

$10 \log_{10} \frac{2 \times 10^{-3}}{10^{-3}} = 3$ dB



So Government sets max transmition limit.

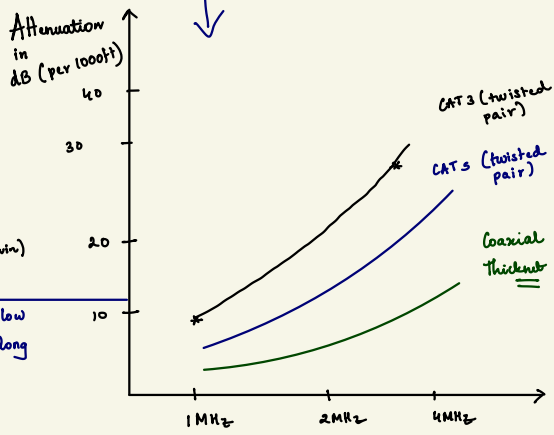
Also to make sure health hazards and other transmission

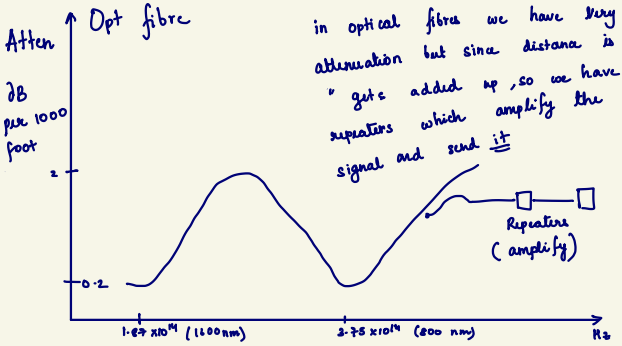
Rx Pow

P = -67 dBm } absolute power → With very small power we can still get wifi signals.

Rx Power / Noise Power } → power + noise

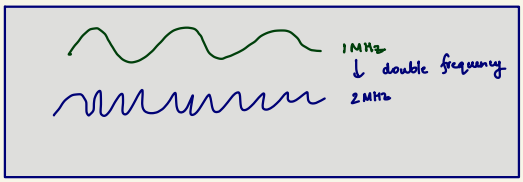
Also noise can be present in circuit as not at 0 K (Kelvin)





in optical fibres we have very low attenuation but since distance is long " gets added up, so we have repeaters which amplify the signal and send it

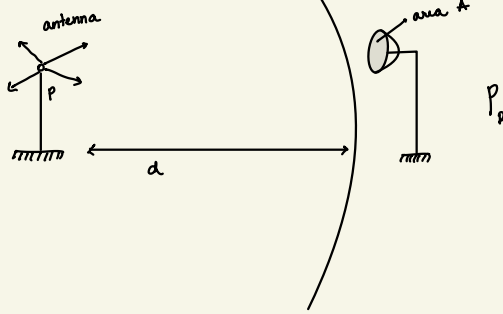
Repeaters (amplify)



The more frequency ranges I have I can send more signal

Shannon — Entropy
Shannon — Capacity } max how many bits we can push through
↓
Max data rate for communication over a channel

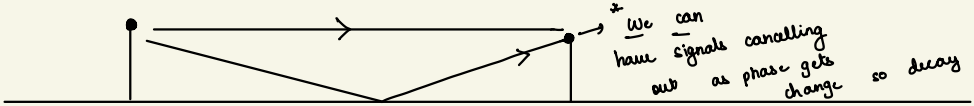
WIRELESS



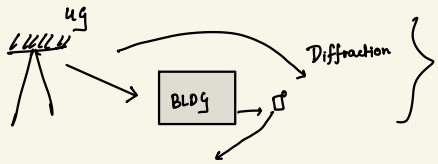
we only receive a tiny part

$$P_{RX} = \frac{P \cdot A}{4\pi d^2}$$

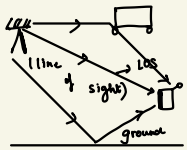
as $d \uparrow$ $P \propto \frac{1}{d^2}$



* We can have signals cancelling out as phase gets change so decay worse than d^2

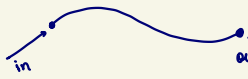


MULTI PATH



due to multipath we can have destructive interferrence.
We can also have construct-ive by moving a little bit

WIRELESS CHANNELS



In wireless signals the attenuation is much higher than wired

two people using the same wireless channel

wifi uses unlicenced band

signals from/for them infefer due to collision.

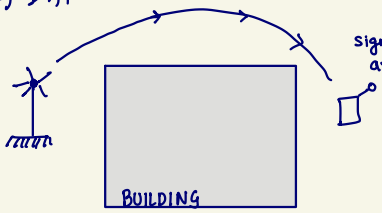
this happens in wifi

How to avoid collisions

in 4g we can avoid this because the tower tells the phone when to communicate

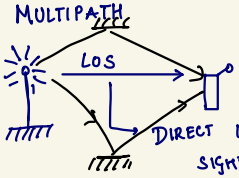
in these band we can transmit without paying government and cost is low
But since unlicenced many people use it and there could be a lot of interferrence

3) Diffraction :-



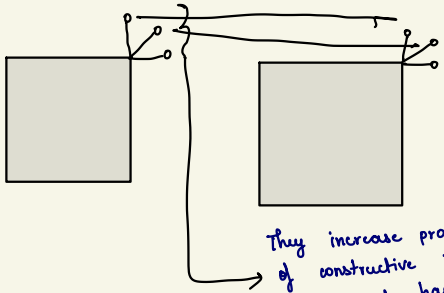
signal goes around the object (so diff to figure where tower is)

4) MULTIPATH



(signals bouncing up, signals get added up and we have distructive interference or even constructive

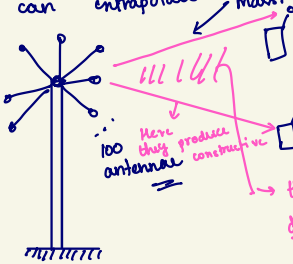
5) MIMO



many aerials

They increase probability of constructive interference because by having 3 we can have atleast one of them having constructive interference

So we can extrapolate massive Mimo

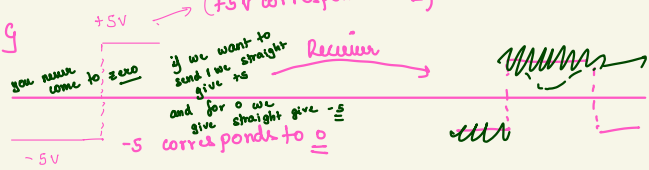


Two getting same freq signal.

Here these antennas produce distructive interference

SIGNALING

WIRED
Non-Return to zero



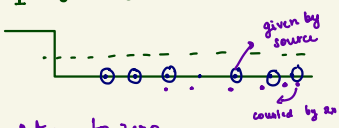
green is noise making it difficult to analyse signals

→ in return to zero we send signal zero by sending 0V then sending -5V.

ISSUES WITH NRZ

i) Bits

1 0 0 0 . . . 0



He has count how zeros he has received.

The signals are coming x Hz the receiver has a fast clock with y Hz

He will count more bits

Because signal of zero was there for a long time. So he counts more bits as he does not have synchronised clocks

In Return to zero we know that zero comes because of a 0V gap.

No need to synchronised clock.

How does a Receiver know when he starts getting message

PREAMBLE

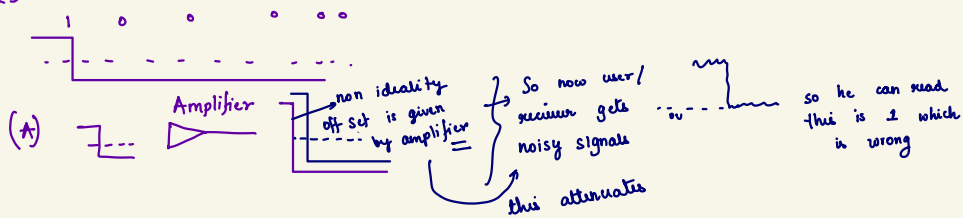
First he gets set bits which signify he is going to get message

eg 10101010 ...

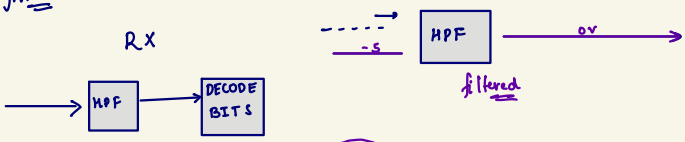
---

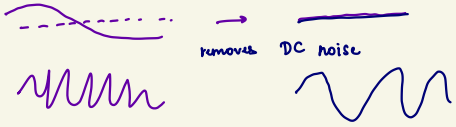
(2) Baseline Wander

1 0 0 0 0 0

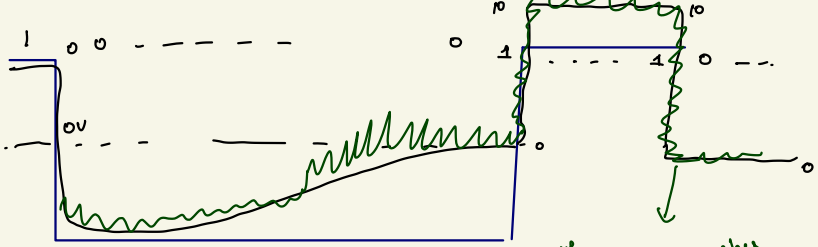


To avoid this we use a high pass filter → removes DC (offset) and low freq signals





in phones we have band pass filter which removes low freq and keeps frequency from high domain depending on "SIM CARD".



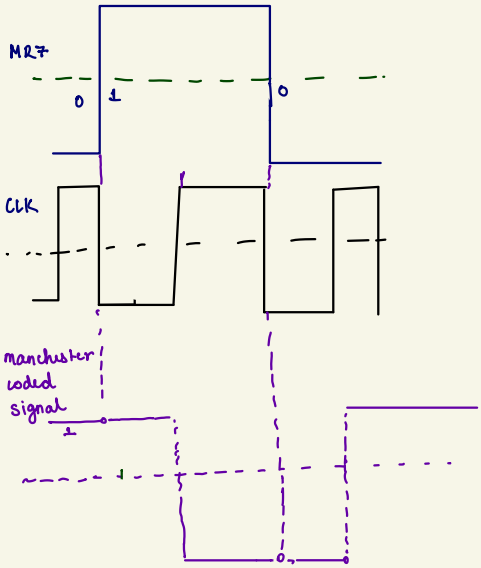
The noise pushes signal above 0V

Due HPF we create more problems

Black is after HIGH pass filter blue is ideal situation

High filter thinks continuous stream of 0's DC so removes it but when 1 comes there is a jump of 10V so a little noise could make bit errors

MANCHESTER CODING (used in Ethernet)

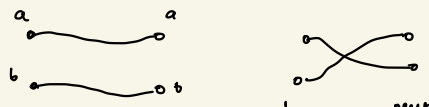


XOR	
0 0	0
1 0	1
0 1	1
1 1	0

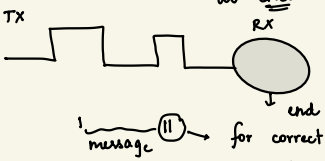
ⓐ Signal transition at every bit period

ⓑ Avg signal per bit period is 0

If polarity switches then message also flips



So we keep a preamble at end which tells what polarity we are supposed to use.

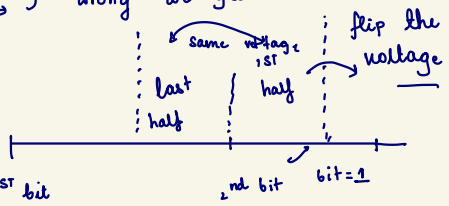


for correct polarity if he see's 00 he should flip

Differential Manchester Coding

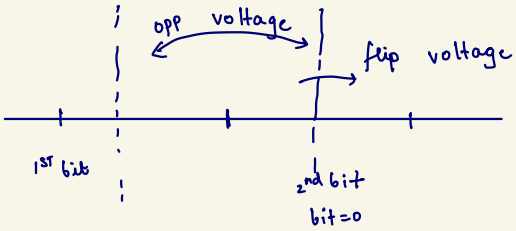
Rules

bits = 1 → even if polarity is wrong we get bits correct

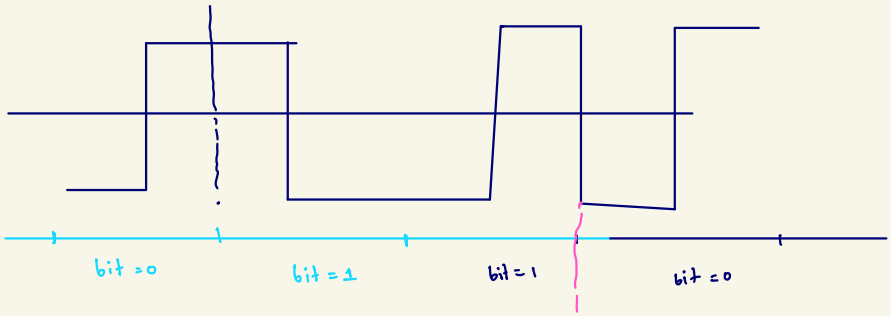


voltage in first half of bit period should be same as voltage in last half of previous period

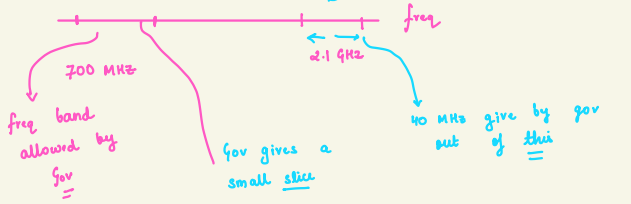
if bit=0 → voltage in 1st half of bit is opp of that in last half of prev bit period



eg)



Modulation :-
in wireless channels

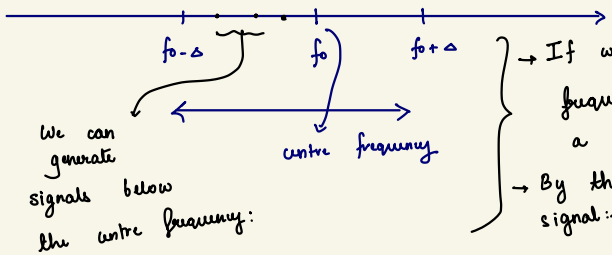


→ If we say Jio uses this band, we have to take fourier transform of the signal and it should lie in this band.

So there are international standards for signals.
So we can design signals which can help all people

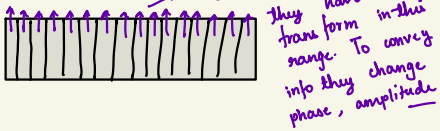
How to confine signal to a band

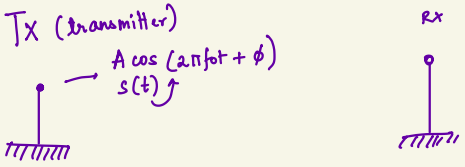
Default signal $A \cos(2\pi f t + \phi)$
A → amplitude, f → frequency, ϕ → phase

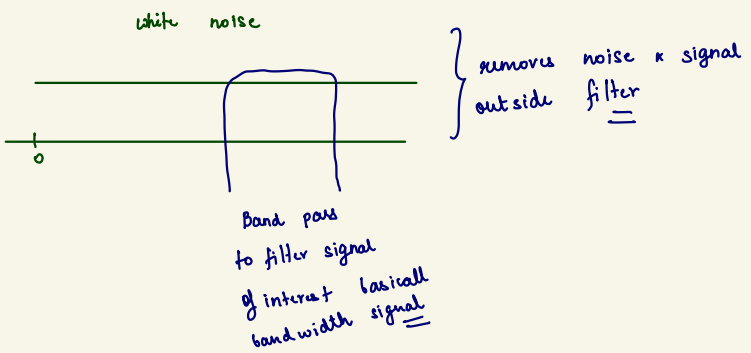


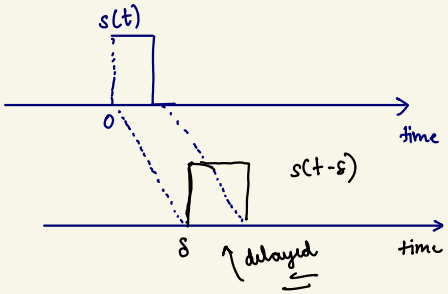
→ If we generate many frequency and use a band pass filter
→ By this distorts the signal :.
→ We want to directly be able to generate frequency of signal within this range

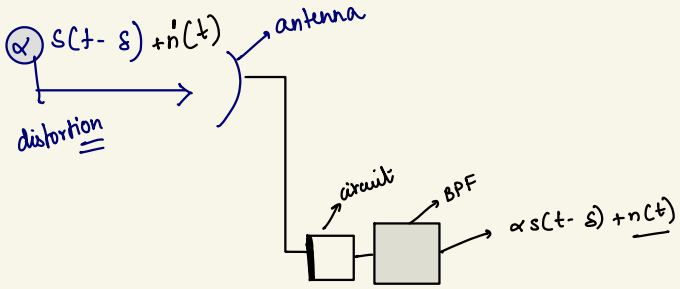
→ What sir suggested we can generate many signals in the same band by spliting a larger band into smaller bands





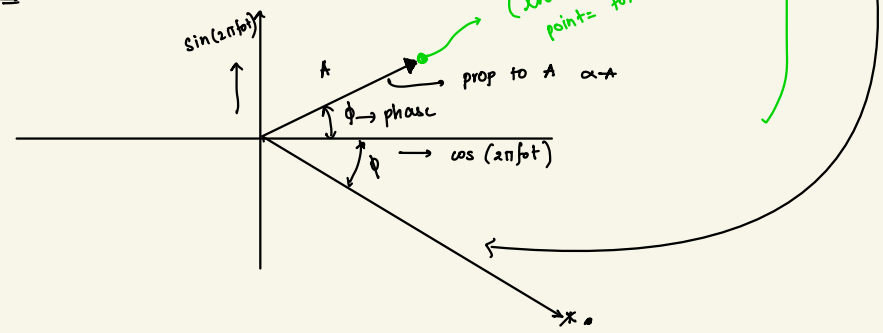






$A\cos(2\pi f_0 t - \phi)$ → Vector Space (?)

We are converting sine wave signals into vector space form



In this vector space

$a(t) = \dots$ $\quad 0 \le t \le T$

$b(t) = \dots$ $\quad$ " $\quad$ "

$$\langle a(t), b(t) \rangle = \int_0^T a(t)\, b^*(t)\, dt$$

3 dim space

$a = a_x \hat{e}_x + a_y \hat{e}_y + a_z \hat{e}_z$

$b = b_x \hat{e}_x + b_y \hat{e}_y + b_z \hat{e}_z$

$$\frac{a \cdot b}{\|a\| \|b\|} = \cos\theta$$

unit vectors

$\sqrt{\frac{2}{T}} \cos(2\pi f_0 t)$ $\quad T = 1/f_0$

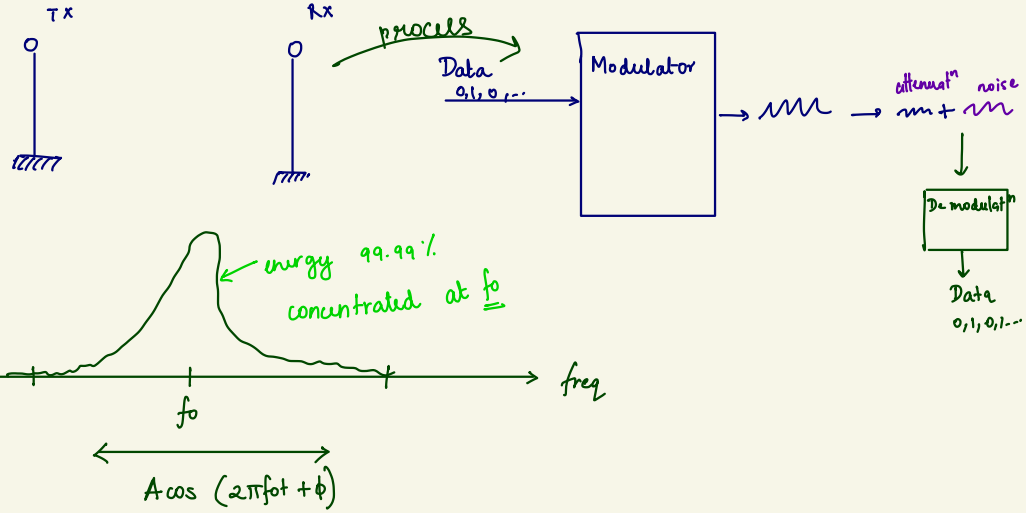
$\sqrt{\frac{2}{T}} \sin(2\pi f_0 t)$

Q) inner prod of these = 0

$$\langle s_1(t), s_2(t) \rangle = \frac{2}{T} \int_0^T \sin 2\pi ft \cos 2\pi ft \, dt$$

$$= \frac{2}{T} \times \frac{1}{2} \int_0^T \sin 4\pi ft \, dt$$

$$= \frac{1}{T} \left[-\cos 4\pi ft \right]_0^T = 0$$



* This is not good as this cannot convey any information

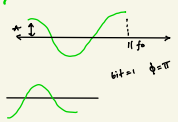
We need change signal to send message

modulate the carrier signal

we can change the amplitude, phase, frequency

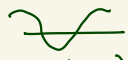
$A \cos(2\pi f_0 t + \phi)$

Ex) $\phi = 0$; bit 0



these signals get attenuated } so we have to analysis to understand what is the

Vector Space



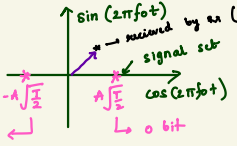
maps to a point

$A\cos(2\pi f_0 t)$

$= A\sqrt{\frac{T}{2}}\, e_x = s_0(t)$

$-A\cos(2\pi f_0 t)$

$= -A\sqrt{\frac{T}{2}}\, e_x \rightarrow s_1(t)$



$\sqrt{\frac{2}{T}}\cos 2\pi f_0 t$ → e_x

$\sqrt{\frac{2}{T}}\sin 2\pi f_0 t$ → e_y

constellation diagram

$r(t) = \alpha s_1(t-\delta) + n(t)$

will be rotated

let us assume he corrects it:-

delay rotates the way?

What will noise do?

Noise : white, gaussian, Noise

Additive (AWGN)

$\alpha s_1(t-\delta) + n(t)$

$\alpha s_1(t-\delta)$ — my signal

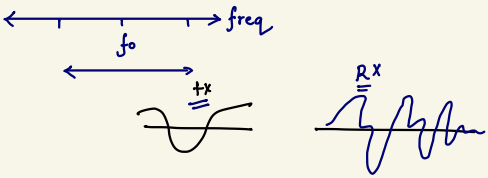
$n(t)$ — noise added thus additive

energy of noise over range

expected value

flat line

noise power / energy



We can model it as gaussian noise

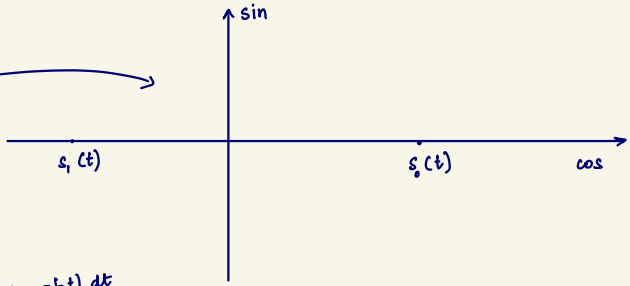
$r(t)$ → received

$s_1(t)$? $s_0(t)$ — how to figure out which signal

$r(t) \overset{?}{=} \alpha s_1(t) + n(t)$

$= \alpha s_0(t) + n(t)$

map



$\langle a(t), b(t) \rangle = \int_0^T a(t)\, b^*(t)\, dt$

$s_1(t) = -A\cos 2\pi f_0 t$

$s_{1,x} = \langle s_1(t), e_x \rangle = \int_0^T (A\cos 2\pi f_0 t)\left(\sqrt{\frac{2}{T}}\cos 2\pi f_0 t\right) dt$

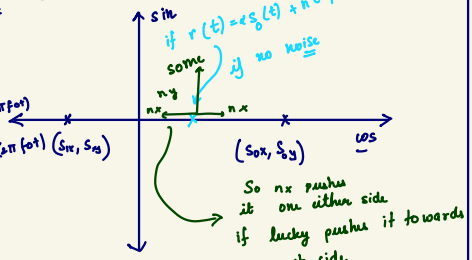
$= -A\sqrt{\frac{T}{2}}$

$s_{1,y} = \langle s_1(t), e_y \rangle = 0$ ← zero

$r(t) = r_x, r_y$?

$r_x(t) = \langle r(t), e_x \rangle$ ↳ $\cos(2\pi f_0 t)$

$r_y(t) = \langle r(t), e_y \rangle$ ↳ $\sin(2\pi f_0 t)$ (s_{1x}, s_{0y})



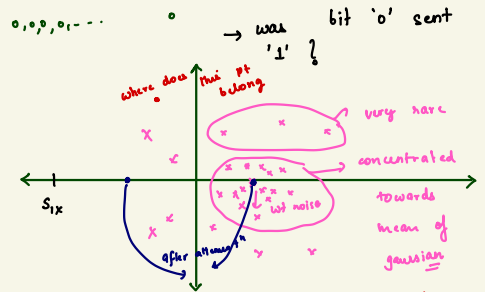
So nx pushes it one either side

if lucky pushes it towards correct side

$n_x = \langle n(t), e_x \rangle$

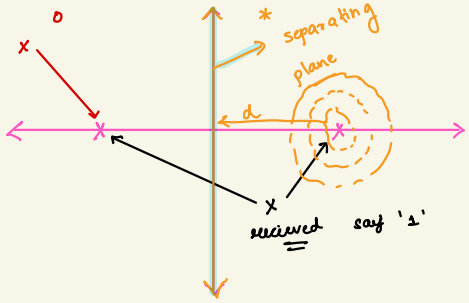
$n_y = \langle n(t), e_y \rangle$

} each is identical independent Gaussian RV



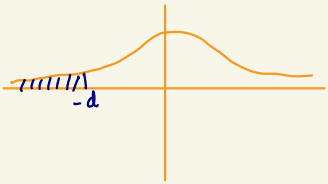
to check see closer to which pt of mean after attenuation.

Suppose we know constellation after attenuation and without noise?

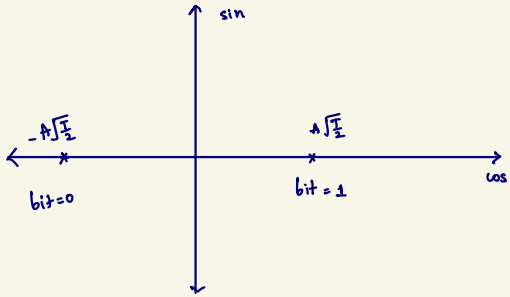


$P_r(n_x < -d)$?? → straight forward calculation

↓ variance, known

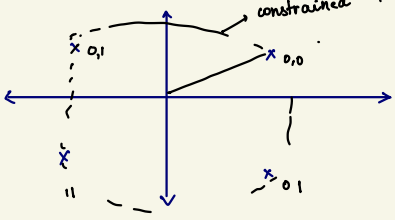


Binary phase Shift Key (shift)

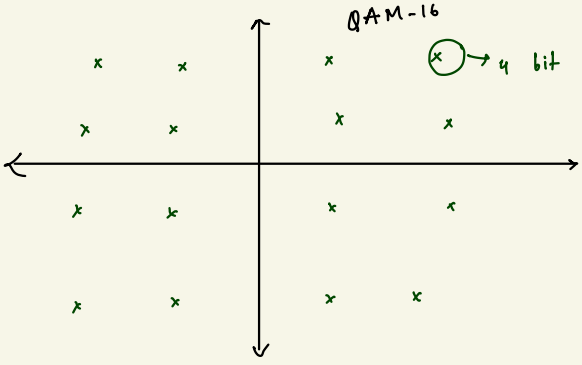


why can we only send 1 bit info
why not 10

Quadrature Phase shift key

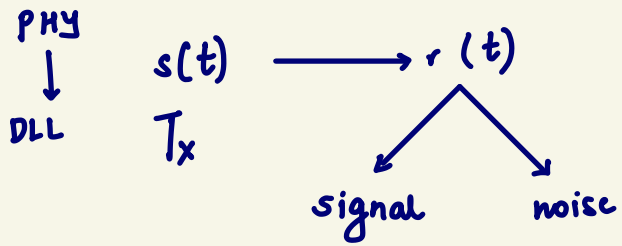


this bits in adjacent co-ordinate should have only 1 bit flip because reduces loss of info in demodulation due to wrong grouping

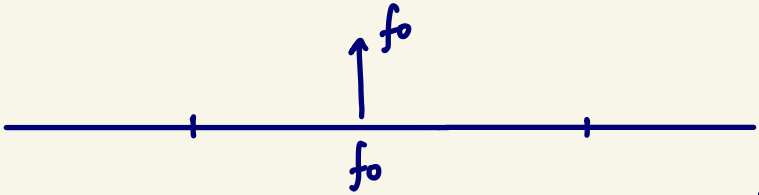


⊗ → 4 bit per-symbol

Disadvantage
→ if we keep power same
then ⊗ symbols get closure
chance of error increase

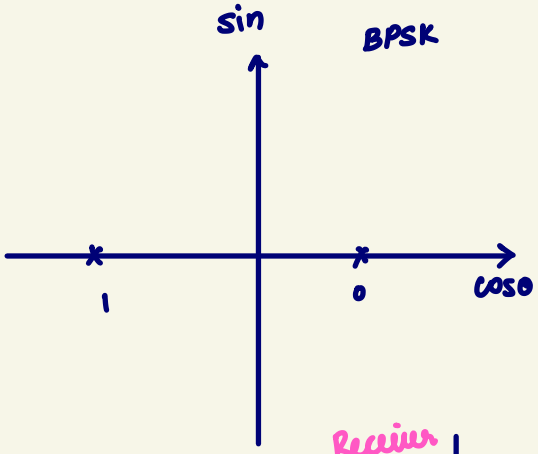


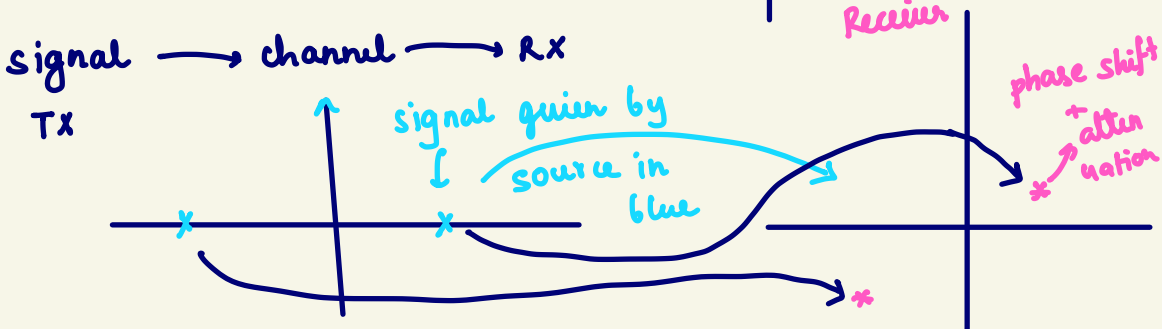
we want keep signal in a particular frequency band

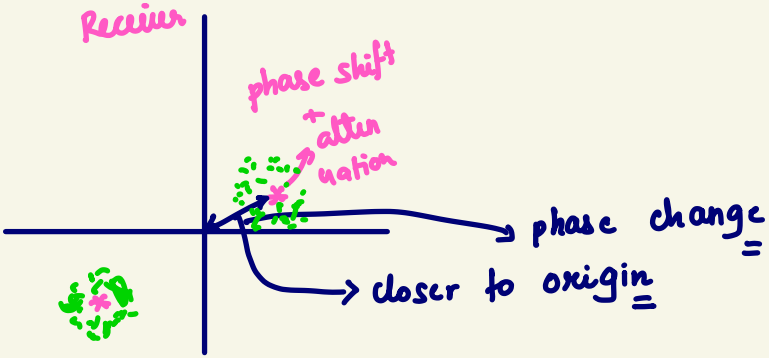


* to convey information we need to modulate or change freq / amplitude

$A\cos(2\pi f_o t + \theta)$







→ Our constellation gets rotated. Also we will see signals around these points

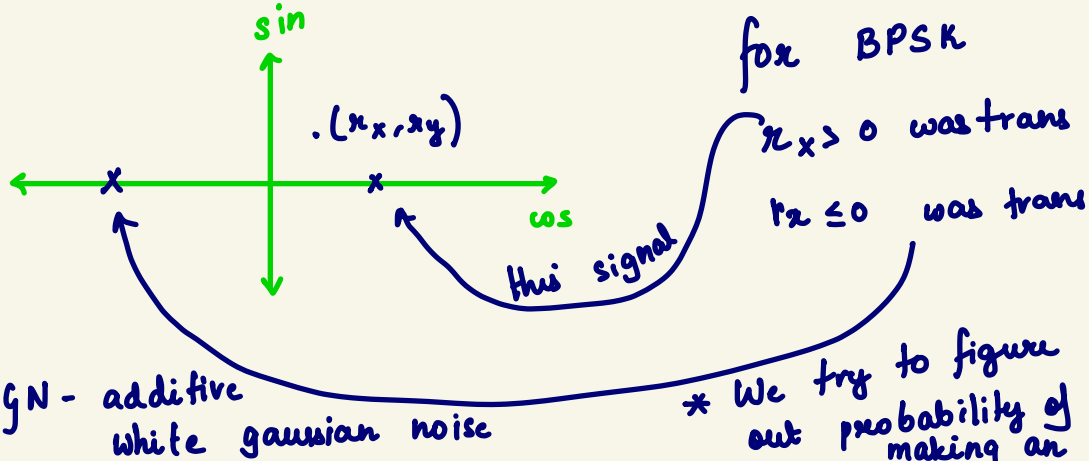
→ so receiver first rotates constellation back to correct angle with some error

→ We do not know the signal yet (1 or 0)

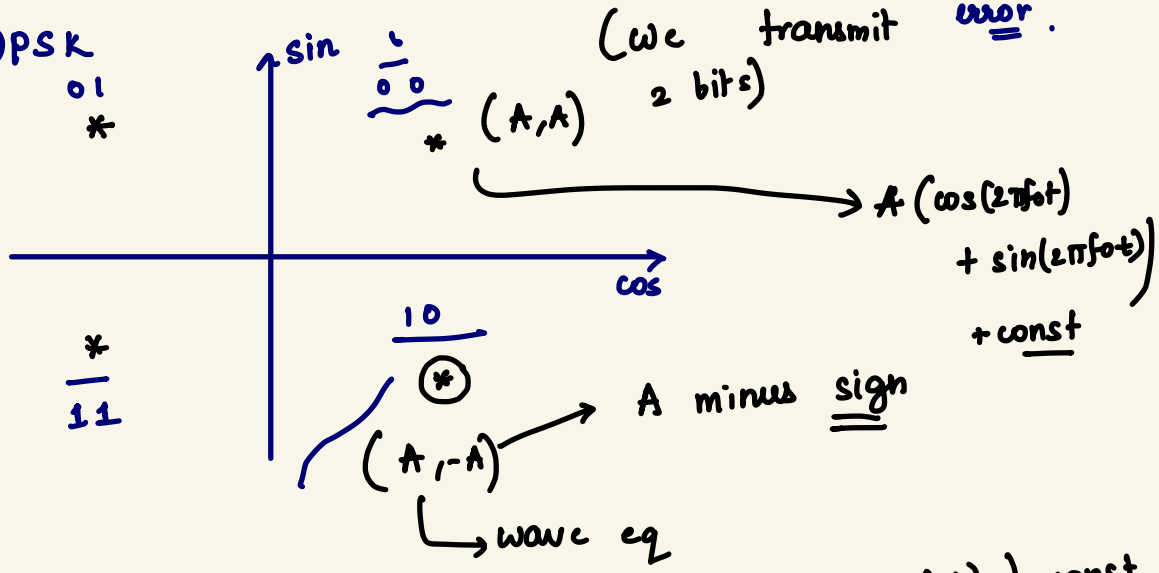
So we take inner product of sent signal with

e_x & e_y .

$r_x = \langle r(t), e_x(t) \rangle$ $\quad r_y = \langle r(t), e_y(t) \rangle$



QPSK



(we transmit error. 2 bits)

$A(\cos(2\pi f_0 t) + \sin(2\pi f_0 t)) + \text{const}$

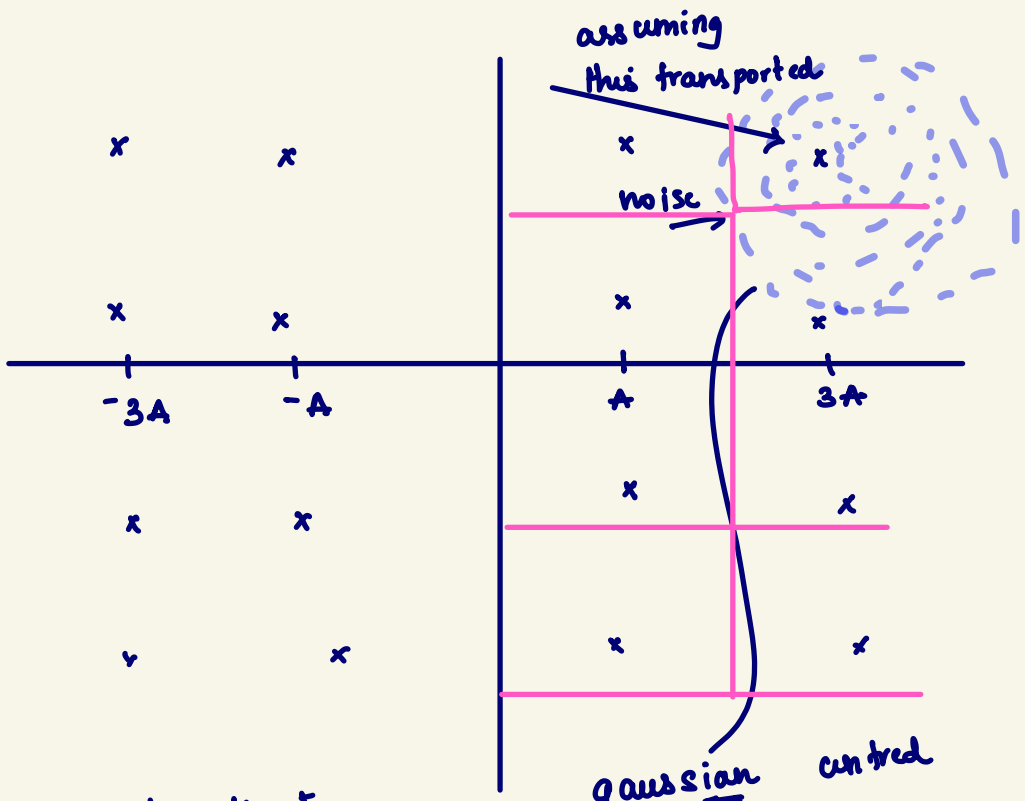
$A(\cos(2\pi f_0 t) + \sin(2\pi f_0 t)) + \text{const}$

We have given convention and this type of bit assignment reduces error.

* if (A,-A) was 11 instead of 10 then if due to error the signal of (A,A) was calculated to be (A,-A) then we would 11 both bits are wrong. If 10 then we got one bit correct.

QAM-16 — (sending 16 bits)

We have 16 constellat^n pts, each pt can have 4 bits.



If we do not estimate channel correctly.

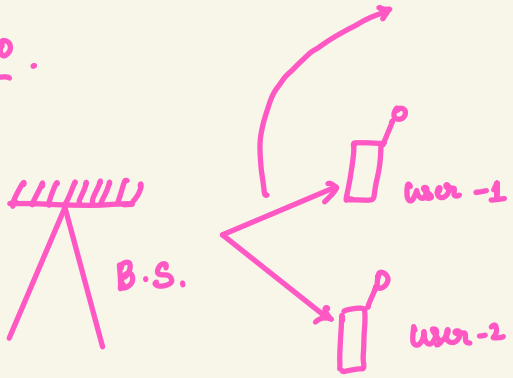
we get a rotated + shifted one then we can map regions like we did in BPSK

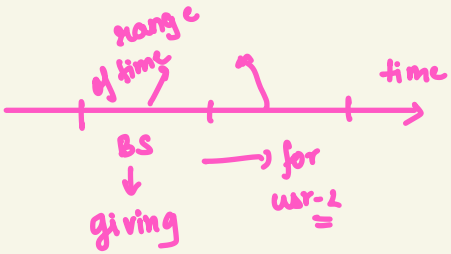
But since not alligned the rotation (phase) we can get lot of bit errors. So our phone has to keep doing this.

→ We could have errors and how to choose which constellatn pt.

→ Choose the one with the least distance from centres

So our phone has to have so info.

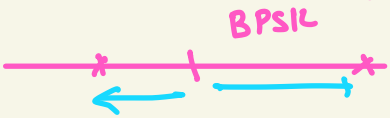


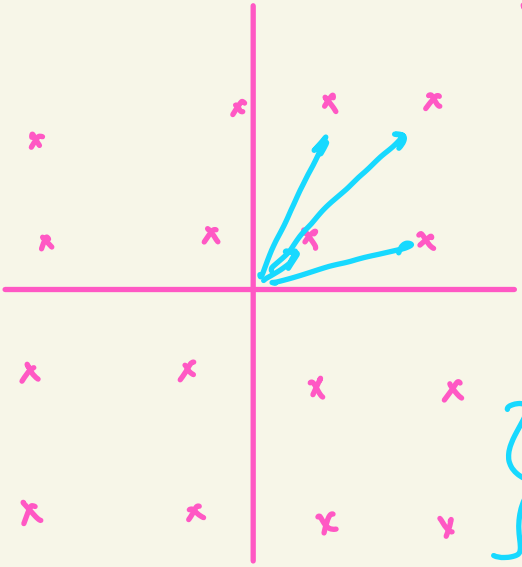


giving signal

So B.S has to decide what type signal (BPSK, QPSK ..) to user-1

So we need to know what level of attenuation, phase errors I can get.





The energy for each type is different for signal in QPSK for a quadrant. In BPSK they have the same energy. But we can lot of bit errors in QPSK.

So we will be give some probability then we should select given attenuatn what errors we can make:-

DLL → FRAMING → when does frame begin when does it end

→ Error detection → CRC (cyclic redundancy check)

→ medium access → (multiple user trying to use the channel at the same time)

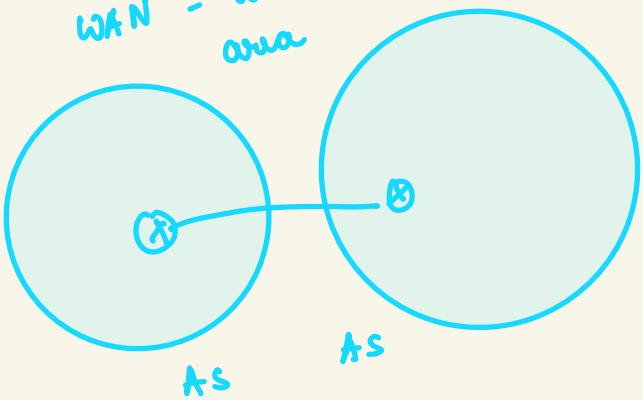
PHY → bits → c(t) → T(x) → r(t) → R(x) → guess bits

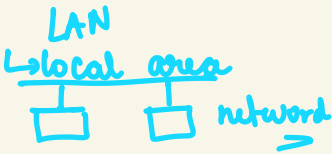
data 1011 01011 011 010 → which sub array of bits is IP address.

* How do we choose which frame has what data.

HDLC → High level data link control

Ref : Peterson & Davie

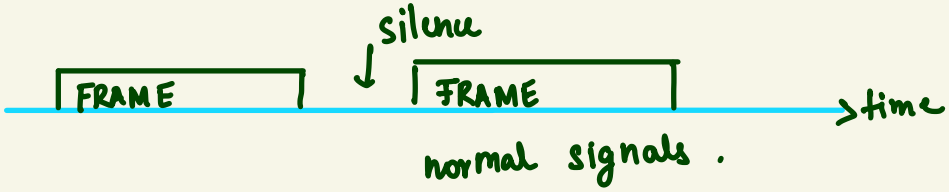




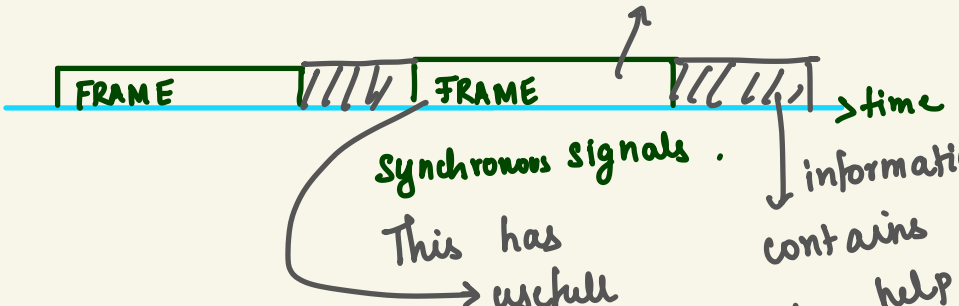
Sychronous Mode (HDLC)

always transmitting

} There is always some signal on line



normal signals.



Synchronous signals.

This has useful info how.

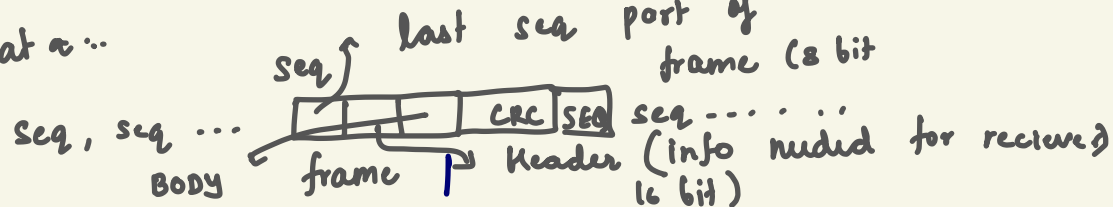
information contains that help use Synchronise clocks for channel estimat^{n}

* How should the user when important info (FRAME) is there and when no useful informat^{n} is being sent.

So HDLC has a DEFAULT SEQ : 0111 1110 ....

So we will have this sequence to synchronise

· One problem if this sequence is in the data ...

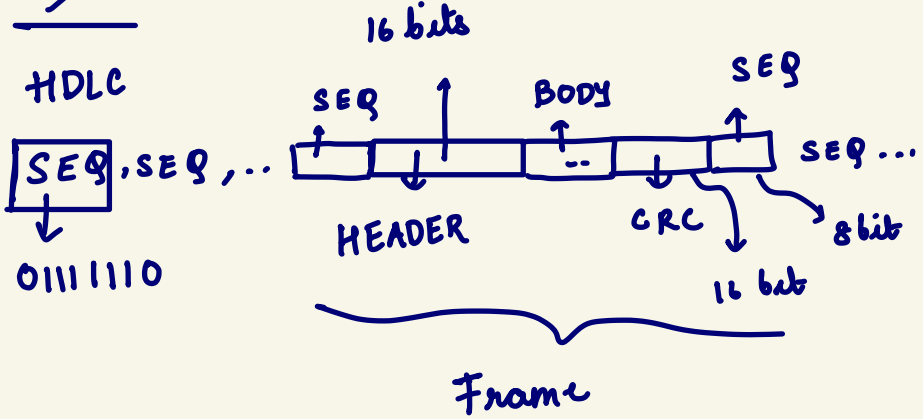


this particular sequence could come in Header to CRC then what to do

If our message/frame 0111 1110 so put some bit in our message to distort this.

DLL

HDLC



what is seq comes in body. Then what to do?

→ One way put length in header → does not hold chicken & egg?

→ So we can see seq has 6 consecutive ones. So where ever 5 ones in message insert a redundant zero. Whenever 5 ones put zero after it. Breaks the sequence:-

AT Transmitter

actual seq

01111111 01010101111110101011111111010 01111110

Ex ↑ this is header, body, crc

whenever 5 consec ones add a zero Rx stops reading here

01111110101010101111100101011111010 11101001111110

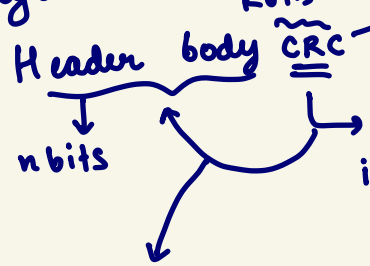
greens are added zero.

Now reciever has these bits. They know there is bit stuffing. Whenever he see 5 ones & one zero ignores '0'

If he sees 6 ones then it belongs to seq

→ If he sees 6 ones continuous and one one. Then there is an error and we discard the frame.

cyclic redundancy check



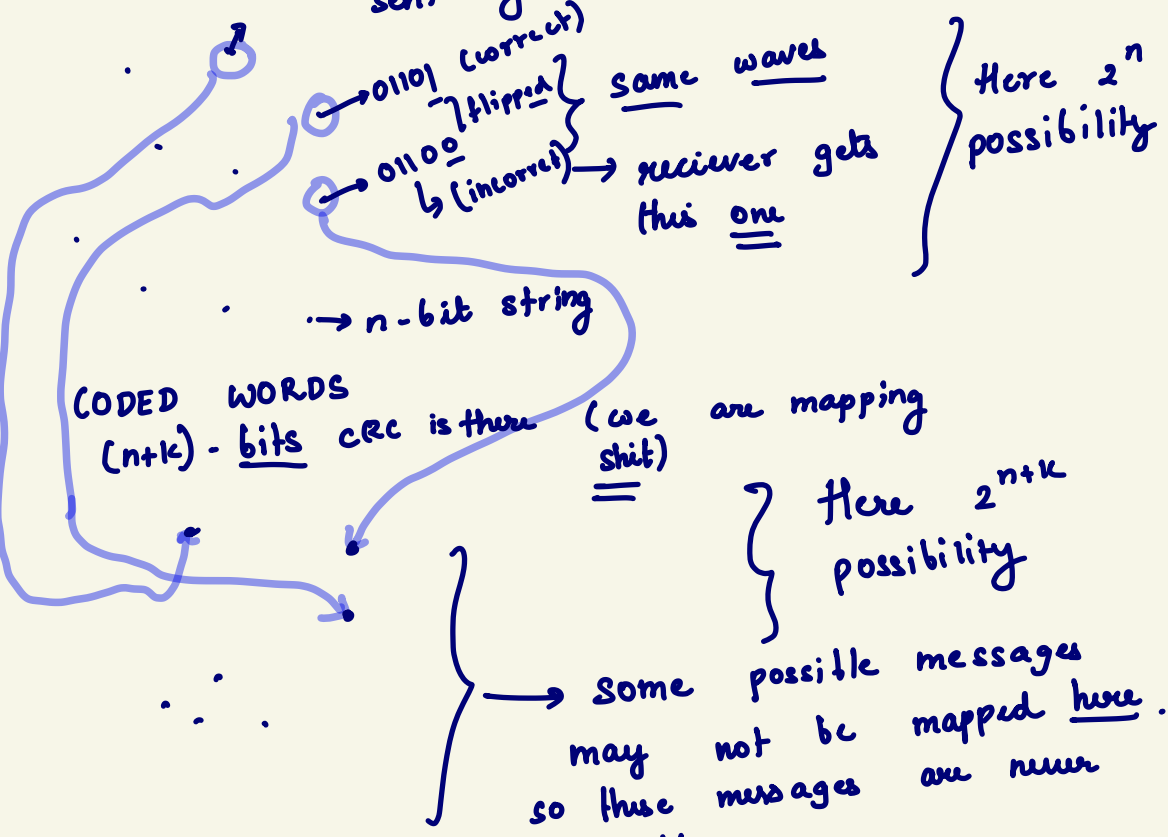
function of data of header & body

1) want to be able to easily detect a large type of bit errors. (single bit, burst of consec. errors etc)

2) creation of CRC & verification of CRC should be computable efficient: -

3) for given K crc shld work for any n.
→ shld be computable for any n :-

DATA WORDS
(n-bits)



HAMMING DISTANCE

given 2 code words : $a_1\ a_2 \dots a_m$
$b_1\ b_2 \dots b_m$

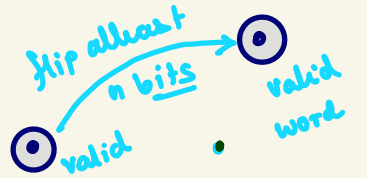
The number of bit in which they differ is called the hamming distance :-

MIN HAMMING DISTANCE of code : Minimum of H·D between all pairs of code-words

We want design code words with maximum H·D :

error detection : If min H·D is N (of a coding scheme) then we can always detect atleast n-1 bit errors

Because possibilites are 2^n we can go other possibility by flipping n bits or more because min possible distance is n

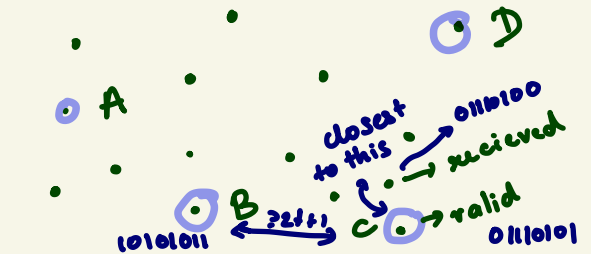


} many invalid word

→ If we flip less that n bits the we go from valid to invalid

- If we want to error then it should fall on invalid code words.
for that less than n bits

* error - correction

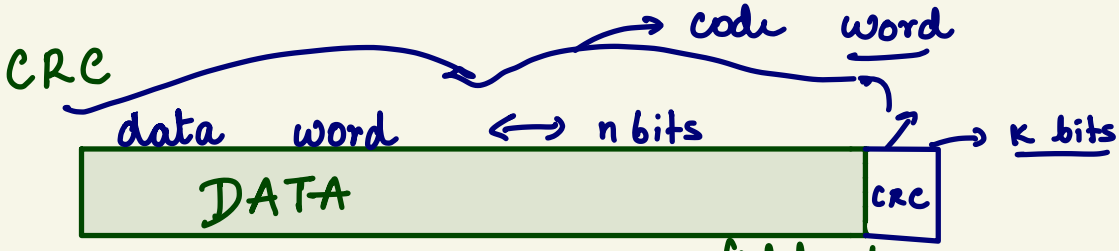


} How to choose correct one. Atmax n-1 errors can be caught

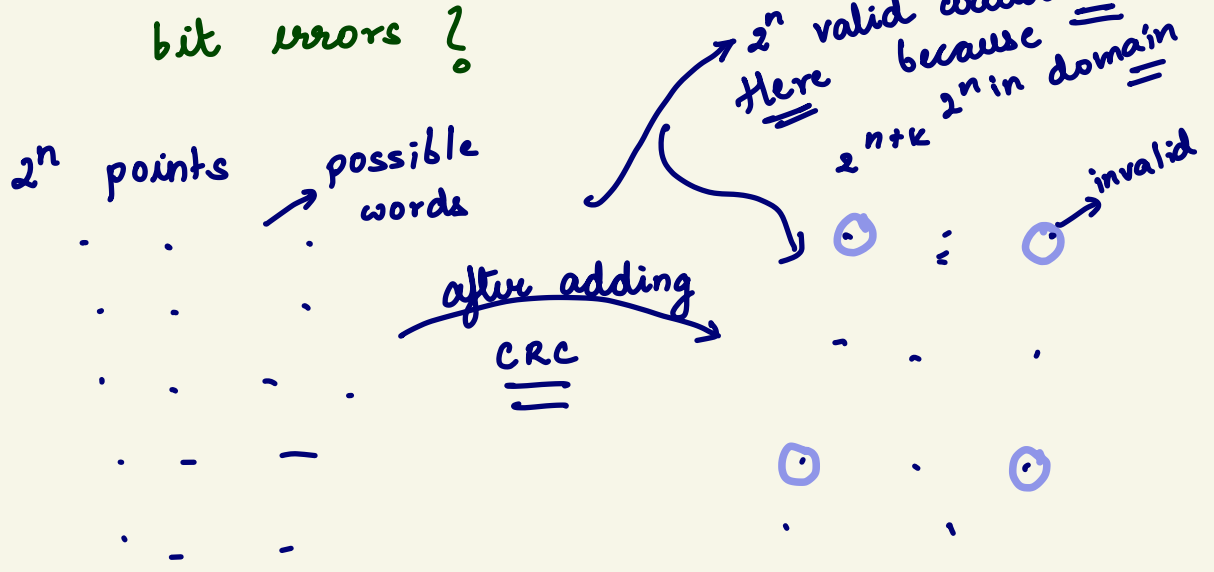
So choose this because we know we have 1 bit errors.

Suppose min H.D $(2t+1)$ and # bit errors is 't' or less then by mapping recieved code to the nearest + valid code word correct all errors

Galois Fields (To generate CRC)



we have added all fields to data, we apply CRC to detect bit errors ?



Minimum hamming should be large to detect as many errors as possible.

No matter how large data word code word should have a CRC.

Galois Fields

for addition use XOR

BIT MUL 0,1 and

X	0	1
0	0	0
1	0	1

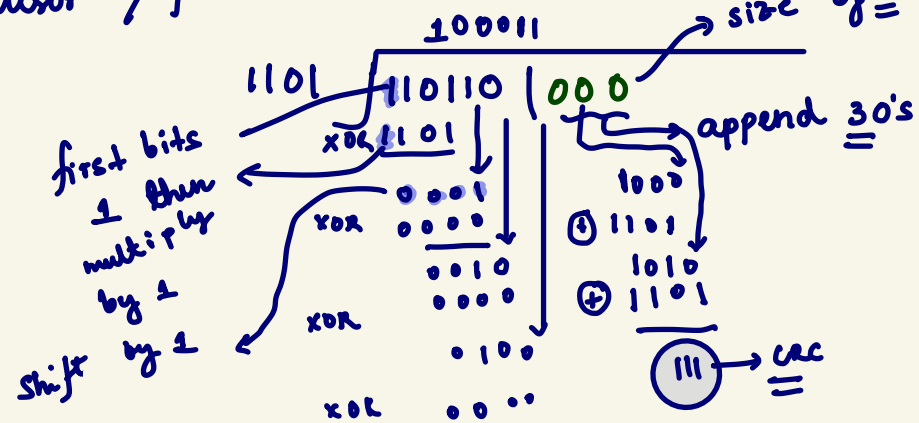
+	0	1
0	0	1
1	1	0

=

How generate crc ?

Ex) Data : 110110 K=3

Divisor / Generator : 1101 (K+1 bits size)

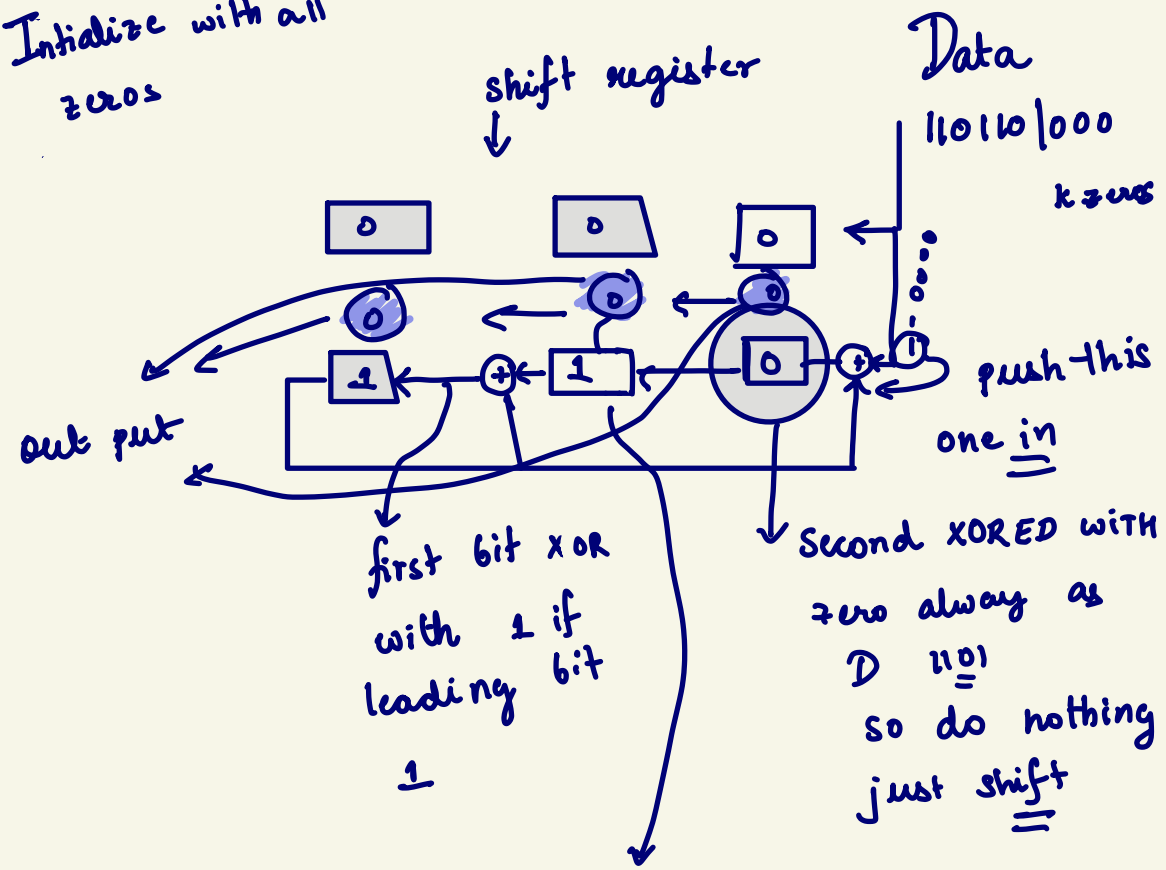


So code word

110 110 111

How to build a circuit

Initialize with all zeros



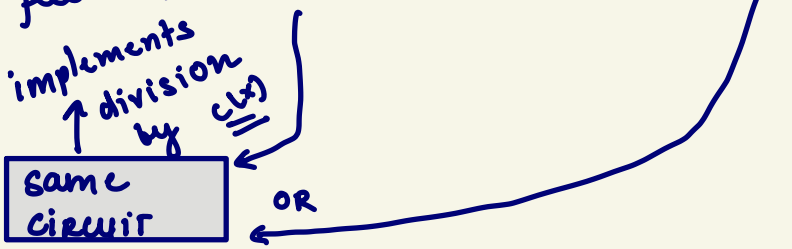
We have code word at receiver

We get codeword as

DATA' | CRC'

At receiver we can apply CRC process on DATA' and see whether CRC' = CRC''

Or we can feed DATA'+CRC' and see if remainder = 0



How to prove this method works

polynomial representation of bit strings

1101 → $1 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot 1 = x^3 + x^2 + 1$

x^3 x^2 x 1

→ divisor poly C(x)

C(x) (1+x) → 1101 → $(1+x)(x^3+x^2+1)$ → XOR

11

$(x^4+x^3+x+x^3+x^2+1)$

$(x^4+2x^3+x^2+x+1)$

$= x^4 + x^2 + x + 1$ → 11101

11 01

⊕ 1 1 0 10

1 0 1 1'

Errors ? error bits

$110\,110\,111 \rightarrow P(x)$ → codeword

$+\ 000001001 \rightarrow E(x)$ → error

same same

we recieve $\Rightarrow P(x) \oplus E(x) = 0$?

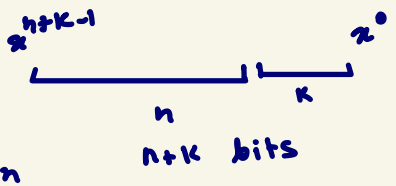
$C(x)$

if zero we are not detect error

if non-zero we caught the error.

Types of errors

1) SINGLE BIT ERROR



$E(x) = x^i$ → error function

→ for some $0 \le i \le (n+k-1)$

$\frac{P(x)}{C(x)} + \frac{E(x)}{C(x)} = ?$ $P(x)/C(x) = 0$

If $C(x) = x^K + \underbrace{\cdots}_{\text{anything}} + 1$

$C(x) * (x^m + \cdots + x^2)$

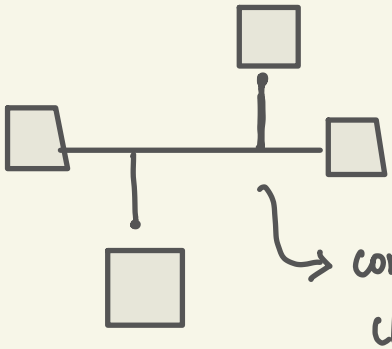
x^{K+m} something x^2

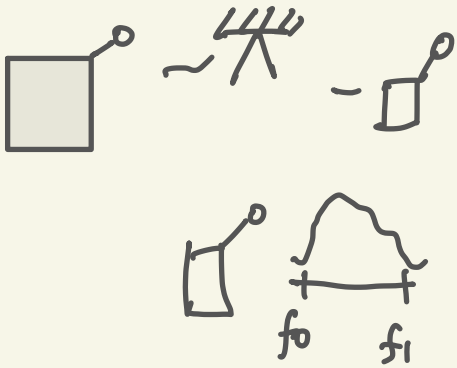
we can never get just x^i

we can always detect one error

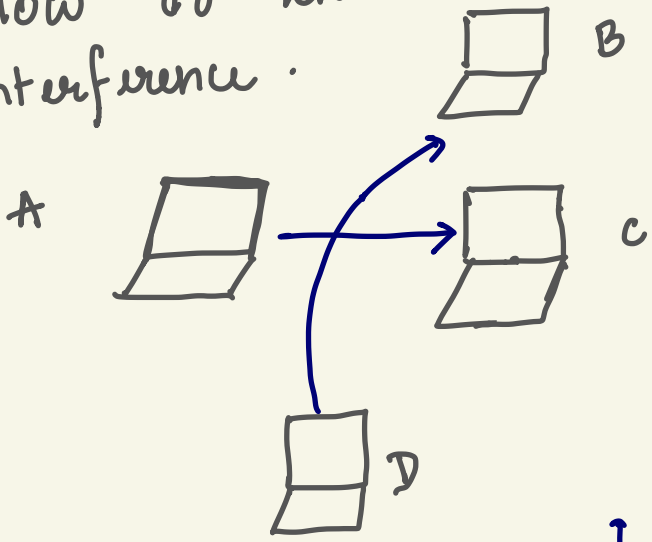
MAC layer

many devices share the same channel





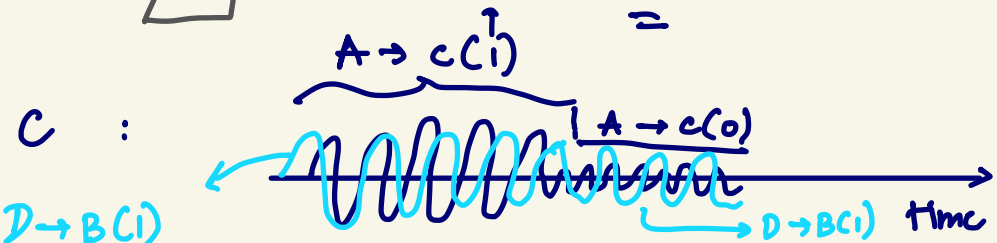
How do these communicate without interference.



A wants to send message to C

D want to send message B

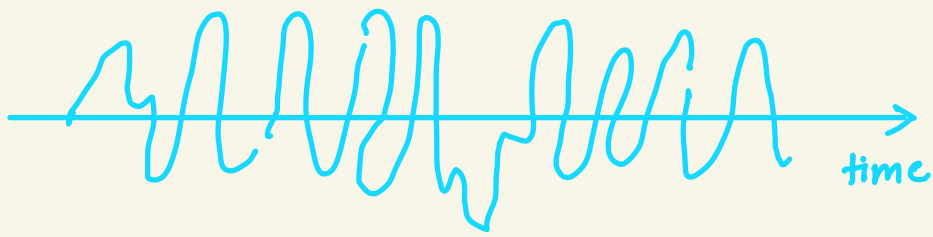
At C :



Now using same tech as above D starts communicating with B.

Now C will also hear it.

How to make sense of this. So net signal received by C



Q) What is being sent, by whom for whom?

Q) We need some sort of identification so we know message is for whom.

Identifiers?

D

A

B

f_0 f_1

band

→ Divide the frequency band into 4 parts and each signifies who is sending.

→ What if we put a new device E !

→ Then what no way to make them a part of our network.

C

This reduced bandwidth and throughput is reduced. Dividing band creates a performance issue

→ Performance: Throughput of A is 1/4 compared to if A used full band.

Addressing

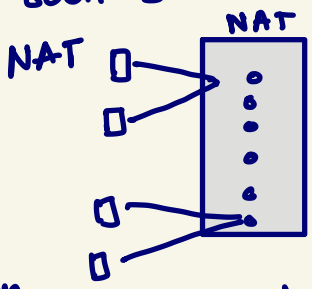
Assign bits to each node

Y2K → problem (lack of assigning bits)

IPv4: 32-bits (layer-3)

↓

{$2^{32} \sim 4$ billion} → We ran out these soon. So we use hacks



many machine could be using same IP for outer world

↓ Solution increase number of bits!

IPv6 : 128 bits

So for identification at MAC?

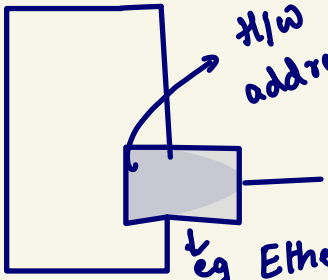
Q) How to assign address?

{manually, automated}

Some people could be entering the pool, some could be leaving.

We are at a low level, when phone boots ups we want it communicate without manually given address

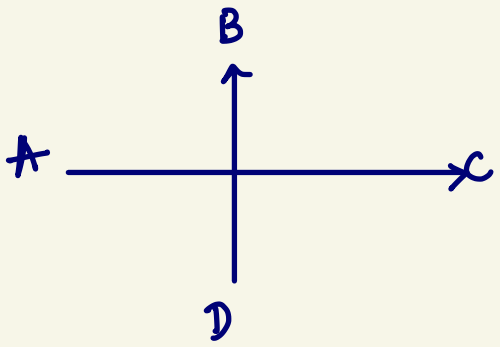
MAC layer (DLL) → use hardware address



→ So we have 2^{48} addresses while creating computers they should be given a unique mac address. So companies making these make sure every device has diff MAC address

How to deal with interference

→ Signals interfering with each other

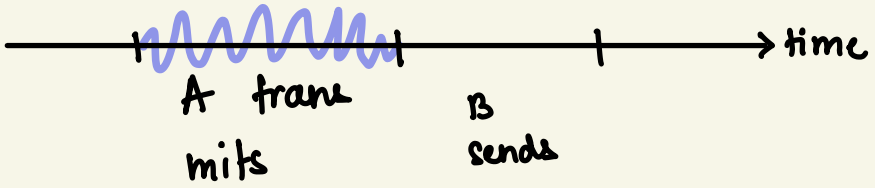


→ One Solution

diff freq band ?

Who use what band ?

One Solution :- only one device sends at a time



→ many problems: Let us say 'E' was out and E did not hear A and E also sends message.

→ Now when A is done who next send message.

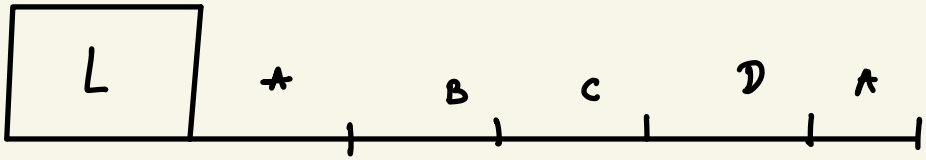
→ How to decide who is allowed to send and when!

→ One thing we can do is to have a leader!

Suppose leader is A.

Leader could decide who is allowed to send message.

Leader could decide who is allowed to send message. He could decide a temp schedule & and this to everyone. And we could keep revising when a new node enters!



Leader sends schedule

So slots should not go waste.

Also how should a new node communicate to leader to join.

If we want to send request then I should have a different band.
So we should be able to request leader for slots. We should schedule request rather than devices!

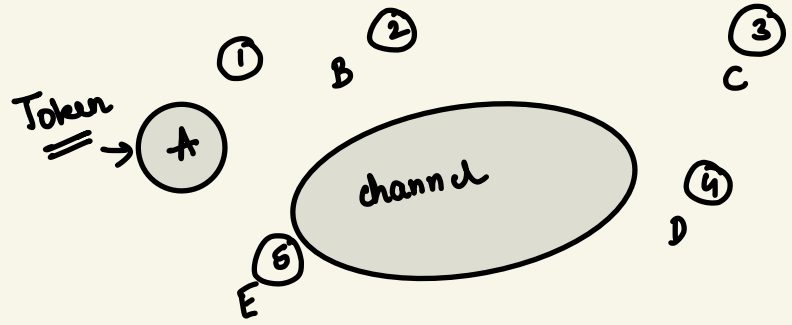
We should ask base station (leader). We will need multiple channels so the leader can get all possible information.

Who should be allowed to be send to leader. Here we face collisions.

How to avoid collisions here while sending to leader.

Token Based → method

Whoever has token is leader



A is transmitting, when A is done he will release a special message I am willing to release the token :-

Now let us say B has highest prio next, so A will give token to B and should communicate this over channel

So we should be able order devices and token should not be with one device for a long time.

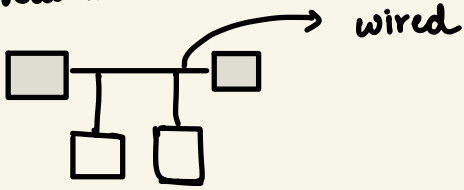
What if a particular node fails ?

Some issues ↑

No leader, No mechanism →

Wifi → allow
ethernet → collisions
} CSMA protocols

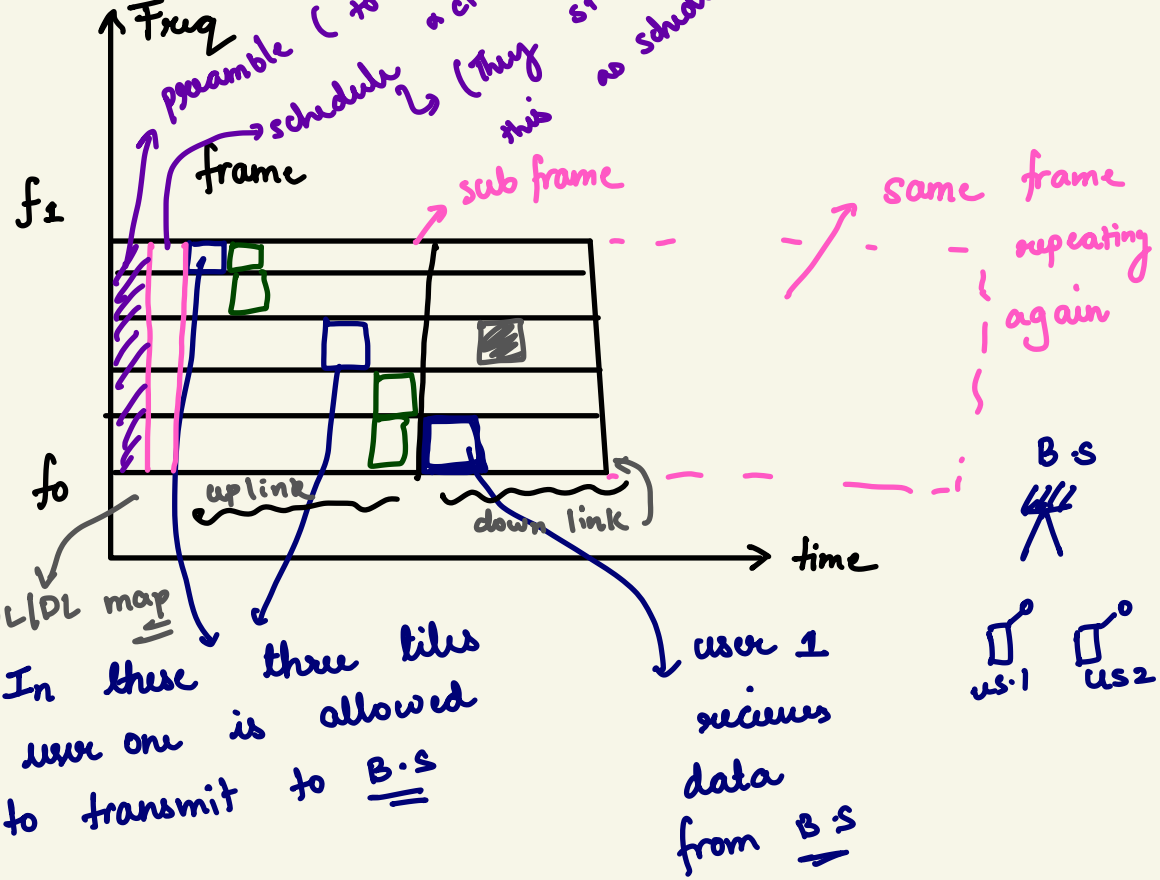
medium access :-



options : 1) Central - coordinator

: 2) No Central

The central co-ordinator decides when which devices can transmit and which devices can receive.



In these three tiles user one is allowed to transmit to B.S

user 1 receives data from B.S

So green is for us2

So how do users know which tile is theirs. A schedule is sent out before hand so each user knows which section of time & frequency belongs to them.

Why not something simple by giving all freq at a particular time instead of giving tiles! Why?

User - Sending Data - UP-LINK

" Receiving Data - DOWN LINK

Every user sees different kind of attenuation for the same tile. So U1 can see more attenuation for grey shaded tile and U2 has less. So he can push more data to u_1 at this tile.

*So we can optimize.

So users actually can tell B.S what tile is good.

Here in same frame we have up link and down link in the same frame.

In ethernet only one frame can be from either side. In 4G both devices are transmitting.

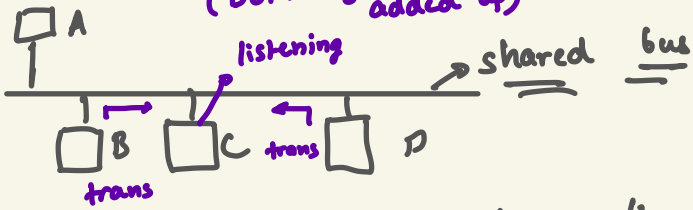
The up link (US → BS) and down-link (BS → US) is in the same frame.

The measure from time preamble for their slot

Issues with central co-ordinator

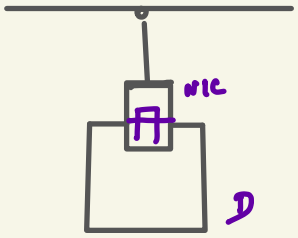
- Single pt of Failure
- May not be possible in certain situations

No central co-ordinator (Both signals get added up)



* If message not for them they discard. We want to connect and remove devices and use them for our advantage. We want play and pause. We have a single channel, no multiple channels.

Want
1) Plug n play
- connect
n
disconnect
2) No central co-ordinator



1) We could use token system

2) We could use a buffer.

3) We could have collisions because many people are transmitting

Idea: Suppose most of time 1 node transmitting only rarely we have collisions.

Collisions: Multiple nodes transmit simultaneously Signals add up at receiver

↳ Neither signal can be deciphered.

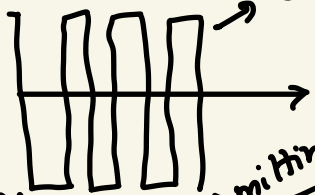
(On wire not much attenuation

Normally

↳

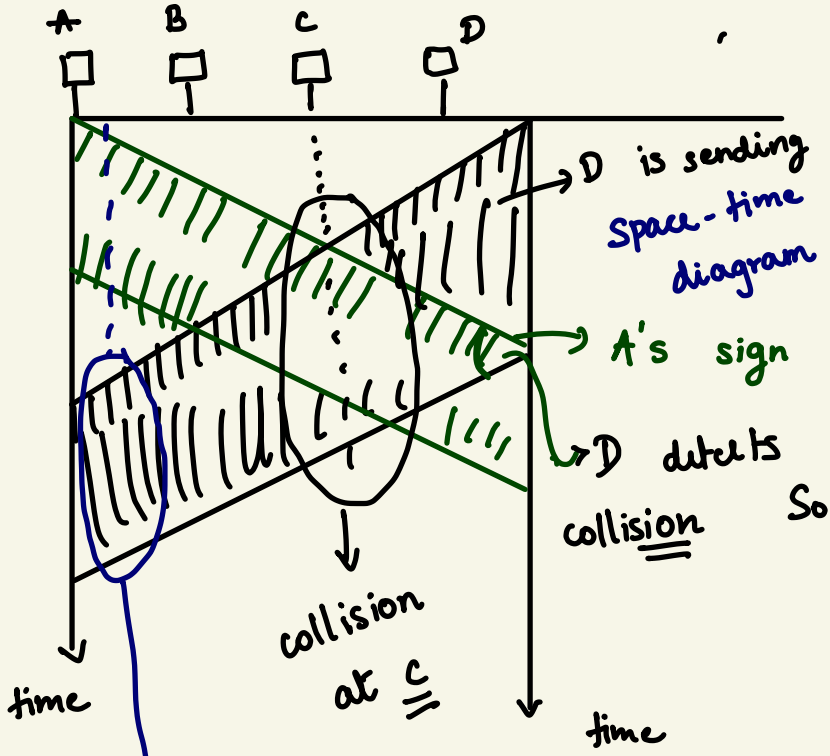
During collision amplitude increases and we can use energy levels to stop transmitting.

During collision



Once we see collision we should stop transmitting

If collisions occurs → stop transmitting



A hears every thing fine so how does it know c had a collision.

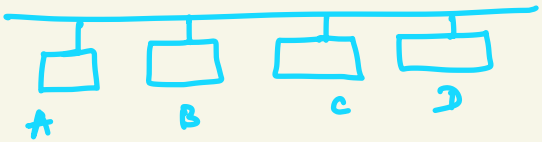
- How to convey to A collision had happened at C

The ethernet does not allow to send small frames. There is a minimum size of frame. So this make sures every one hears the collision

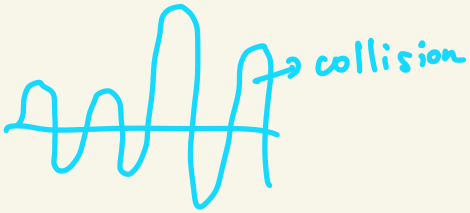
Ethernet

CSMA-CD ⟶ collision detection

carrier sense → CD: check energy levels.



→ if higher energy seen than expected then collision



Ethernet is created by

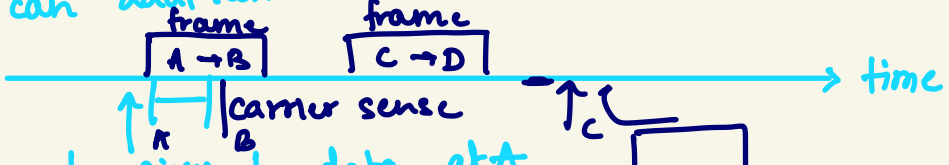
IEEE : 802.3

Wifi is a sister protocol

WIFI: 802.11

→No centralized co-ordinator

→We can add/remove nodes easily



L_3 gives L_2 data at A

→ L_3 layer of A

When I am transmiting no other should transmit

We use carrier sense to detect energy if some body is transmitting then carrier detects it and gives a "no" signal.

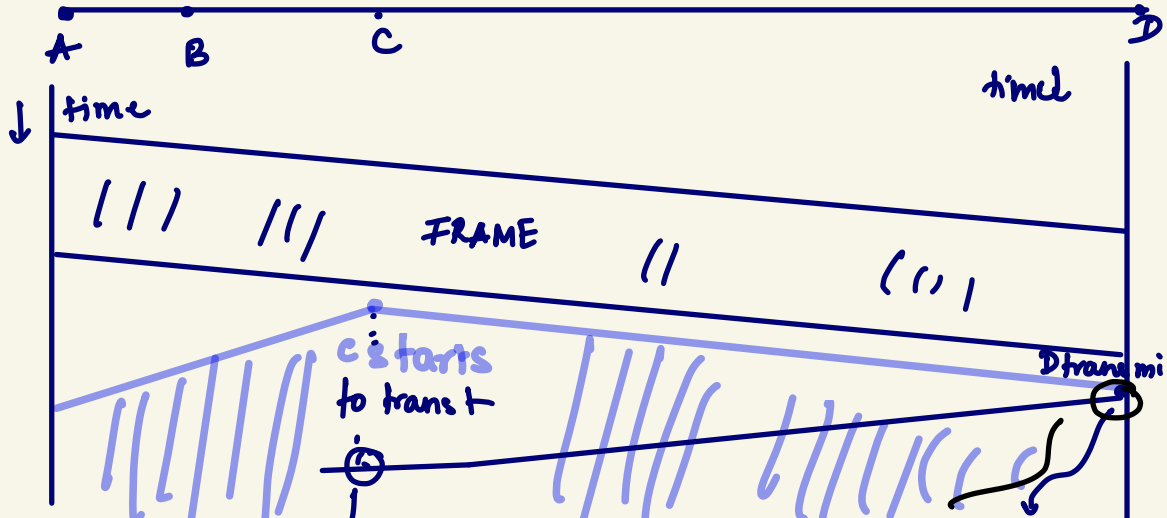
Ethernet uses manchaster encoding

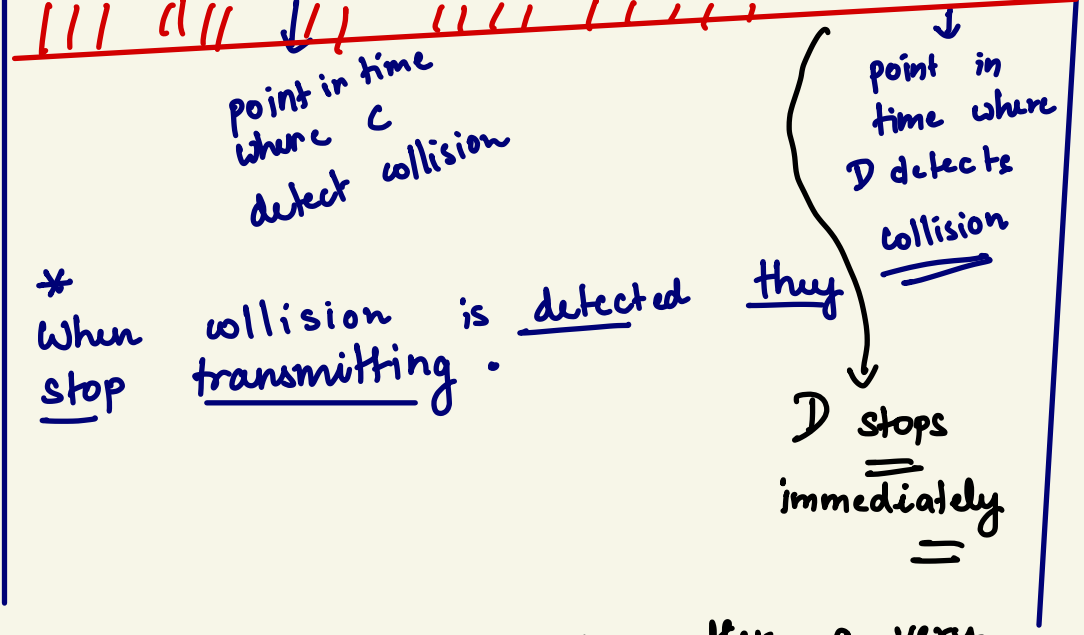
A is sending to B

→ If no collision then assumed the other party got data perfectly

→ We C want to pass data, it checks when there is silence on the channel.

space-time





* When collision is detected they stop transmitting.

If D detected and stop after a very small time and C misses this small signal sent by D then C has messed up.
So D sends out jamming signal.
The send one. So all hear collision.
So all who detect collision we send a jamming signal.
Jamming signal has to be long enough!
No how to decide who gets to transmit next!
Because collision can happen with n senders where $n \in \mathbb{N}$.
So we use random wait time

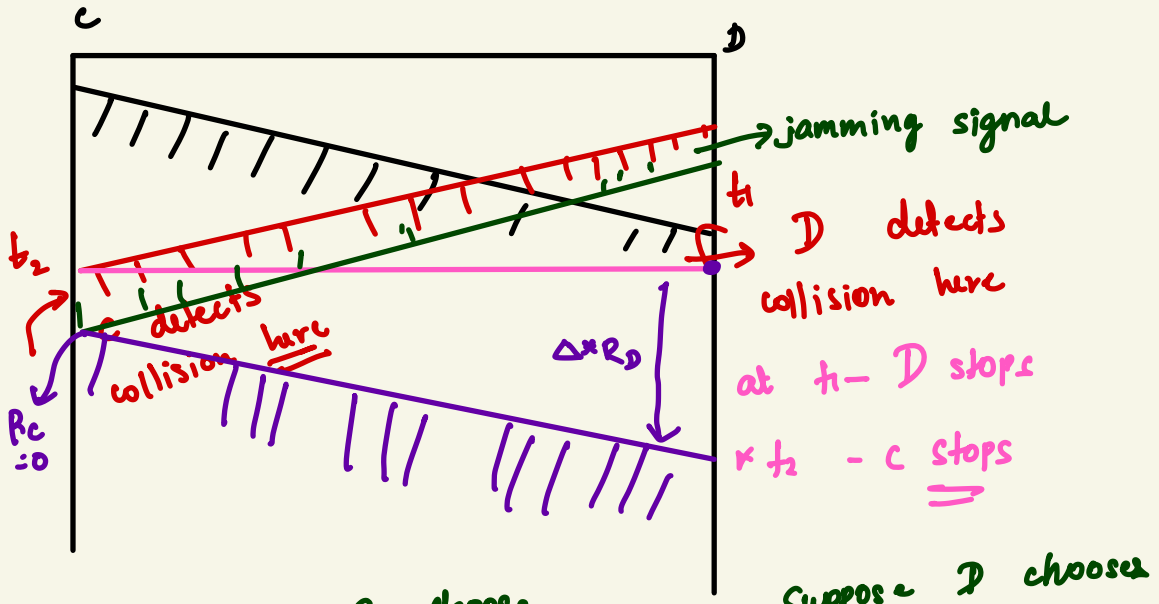
Each time node throws Random Number R, unif from some range wait for $R * \Delta$ before trying to send again :-

The one with unig min will win

Idea:

$R_C < R_D$ then C will transmit next wo wolliding with D.

We could again have collision if $R_C = R_D$ and if $R_C < R_D$ we want C to win so what Δ be :



Suppose C choose $R_C = 0$

Suppose D chooses $R_D = 1$

we want C to transmit without collision

We D stops hearing signal he will wait $R_D \times \Delta$

so $R_D \propto \Delta$ should put D in place where C starts transmitting and C's signal should reach D.

Δ should be large enough for D to hear C before transmitting.

So what should this Δ be.

If $\Delta >$ Round trip time then we are safe

Can show, if $\Delta >$ RTT then D will not transmit before C

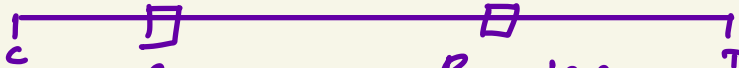
RTT = 2x (time for signal to go from one end of network to another)

Max cable length = 2500m

speed of signal = 2×10^8 m/s

OWD = $\frac{2500}{2 \times 10^8}$ = 12.5us

↳ These repeaters are also there so we delay due to them



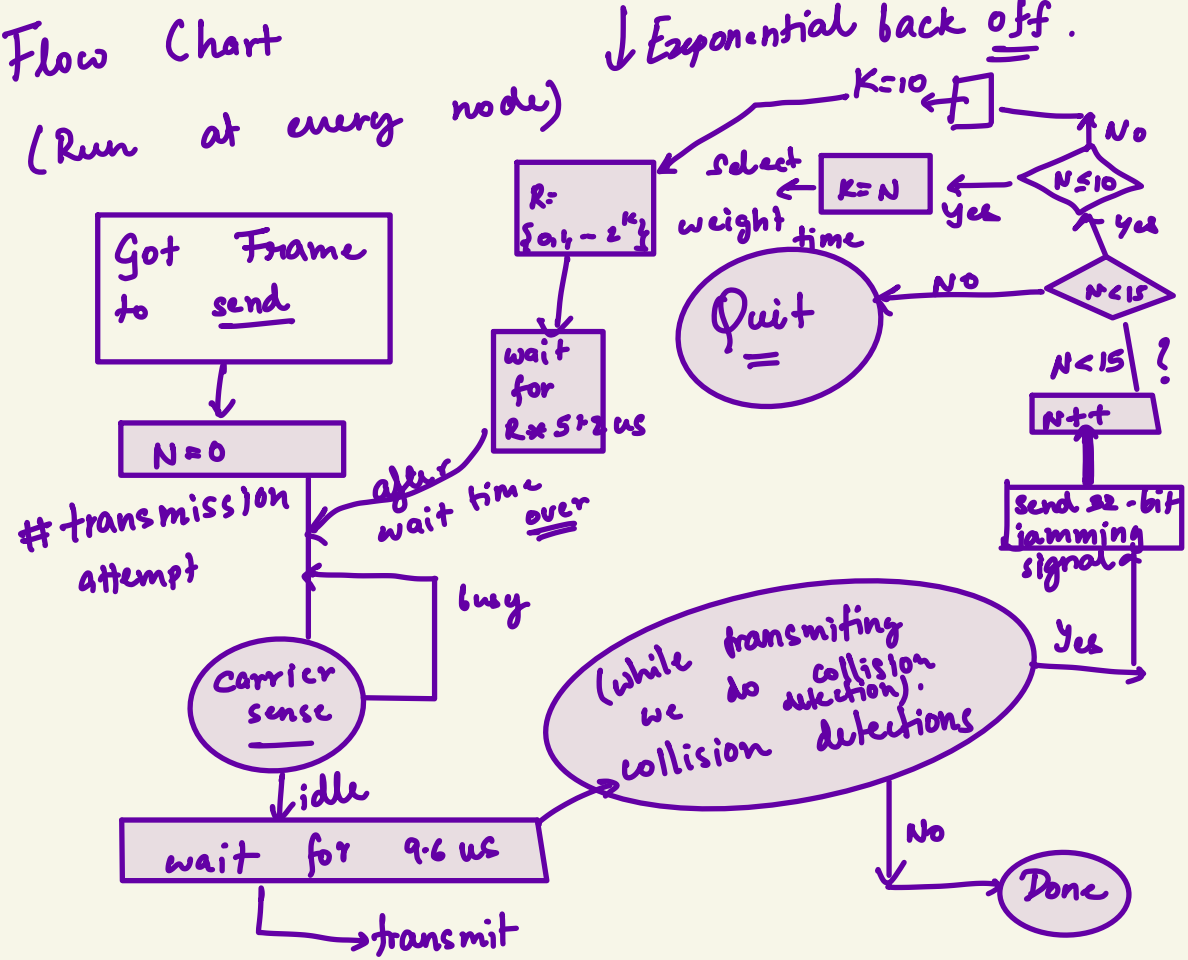
max (4) delay due to repeaters ← ideal 2×12.5 us

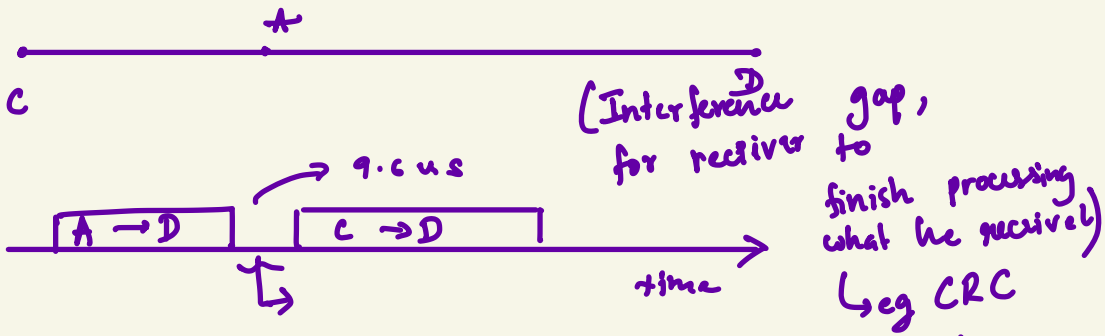
otherwise 51.2us

Flow Chart

(Run at every node)

↓ Exponential back off.





↳ eg CRC check

Exponential Back Off

- Don't know how many people colliding

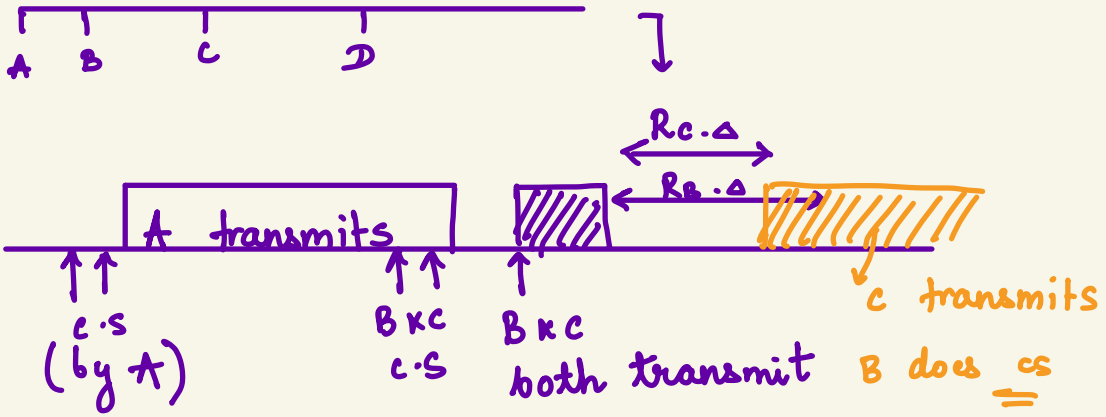
Suppose M colliding

this α should depend on M.

There will some optimal α we want

go over it so we keep increasing 2^K.

CSMA - CD - ethernet



$R \in \{0, 1, 2, \ldots, 2^K\}$

(We do this with hope that ∃ a unique min

K increments upto 10 for every collision

(for same frame)

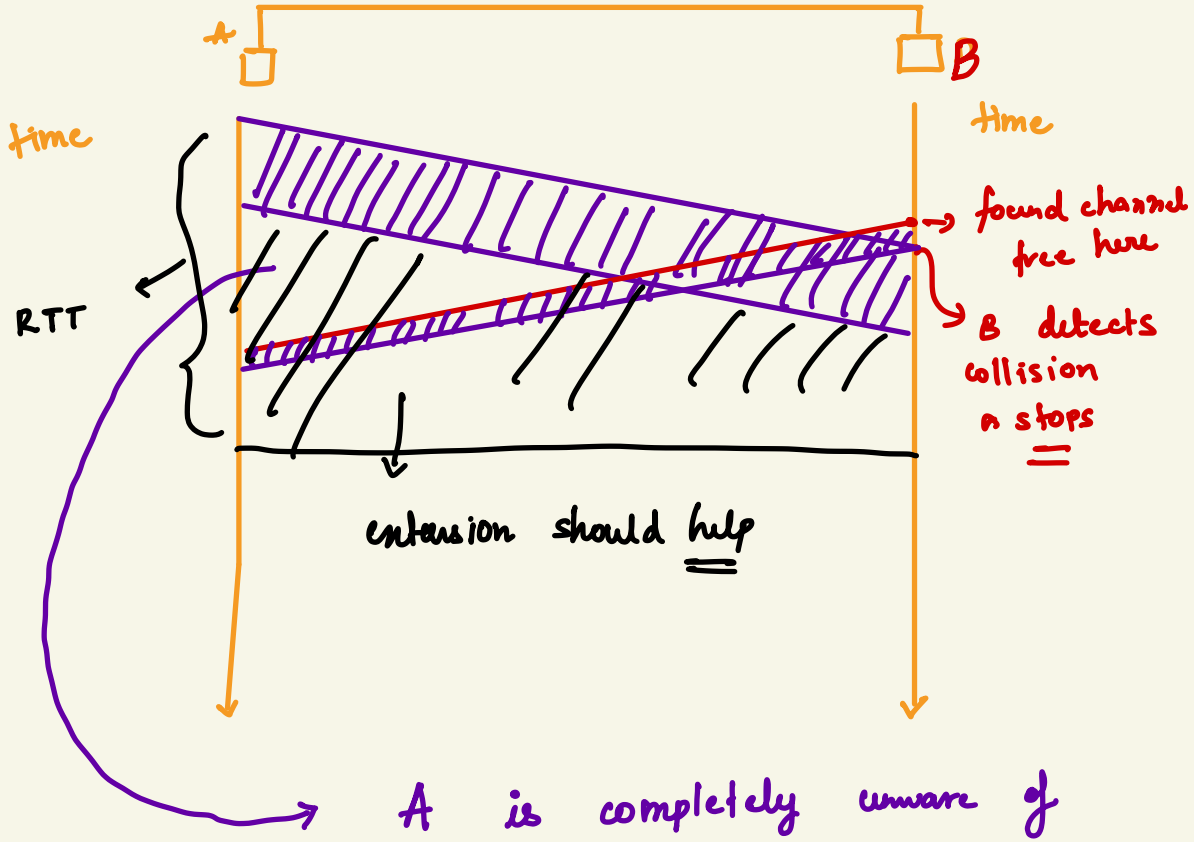
Simple, Robust ↑

↳ just connect machines to the bus

↳ limits on frame size (ethernet) 802.3

---

1) min frame size



A is completely unware of it because B's signal reached to it when A finished transmitting so all is fine for him

So we can do one thing we can make frame of large enough to detect collisions while detecting:-

Δ = worst case RTT

$\Delta = 51.2$ us

Original standard 10 Mbps

Frame size 64 bytes = 512 bits

Time to transmit @ 10 Mbps = $\frac{512}{10^7}$ = 51.2 us

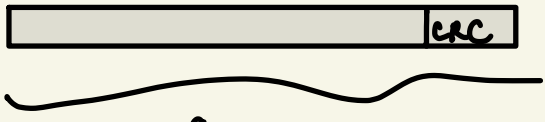
If Mbps ↑ we reduced length by that factor so RTT ↓ as

@ 100 Mbps

5.12 RTT

2) Maximum frame size

(a) Bit errors



~ 10^9 bits

1 bit error in 10^6 bits

so almost 1000 errors

∴ High prob of at least 1 bit error

b) time of transport increases

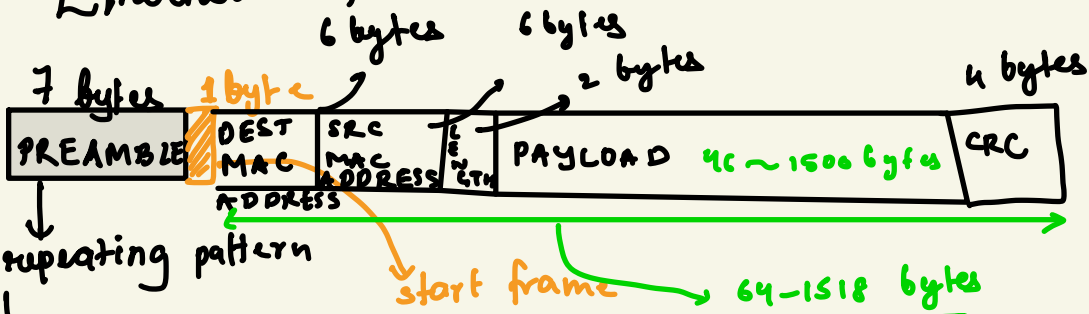
= if 10 Mbps then 100 seconds for above message!

So while encapsulation we should keep in mind because max length

b) Don't want single user to monopolise the channel for very long.

Max Frame size = 1518 bytes

Ethernet frame format



1 byte ≈ 10101010

manchester coding

[we need to know polarity

10101011

↓

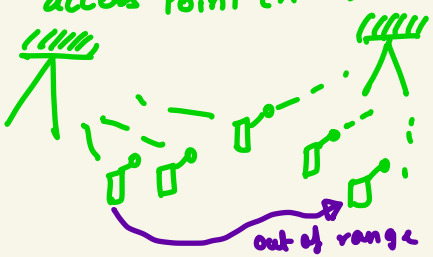
if it flipped

we know polarity flipped

WIFI (IEEE 802.11)

802.11 b/a/g/n/ac → variations of 802.11 standard

access point (A.P 1) (A.P 2)



→ Users have multiple access pts It can choose which one it wants.

If one transmits many can hear Some out of range cannot hear

↓ Ethernet topology in wifi

will carrier sense for CSMA-CD work?

No because out of range issues!

eg)

So A is transmitting to A.P 2 and B is transmitting already to A.P 2 but out of range of A. So collision.

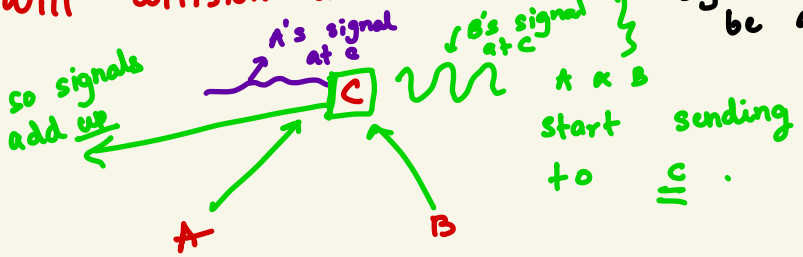
A cannot sense B's (signal)

Hidden terminal problem

Hidden.

C.S may not work

Will collision detect work



* signals can be attenuated (making deciphering tough to add up)

so signals add up

A & B start sending to C.

A's signal

B's signal

B's signal is like noise at A due attenuation (signal decay with distance $d \sim \frac{1}{d}$)

When one person is transmitting he cannot hear anybody else.

Everybody is whispering compared to us.

While transmitting A cannot hear anyone else.

Collision detection does not work!!.

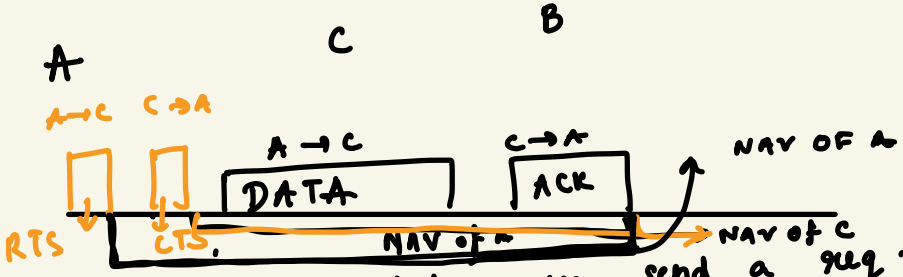
Here we cannot have same type of assumption as ethernet "if we hear no collision there is no collision".

Fix:
Acknowledgement.

Receivers sends ACK

If ACK collided we send A → C again!!

How to handle Hidden Terminal!



Before sending data we send a req to send to C & C will send a clear send. This to avoid the case where A sends while B is already transmitting to C ∴ B will not hear RTF but will get CTS. This makes sure all potential sender to C get CTS meant for one. CTS & RTS have a NAV (Network allocation vector. (Time during.

Rule: All hearing RTS & CTS (other than A & C must remain silent for NAV duration)

VIRTUAL CARRIER SENSE TO B DUE TO CTS OF C.

All senders to A & C are frozen during NAV.

- What if A send RTS but does not gets CTS?

- A will assume collision and retrasmits!

- what if RTS, CTS, DATA are sent but no acknowledgement to A?

same over here

we again send RTS!

WIFI MAC (IEEE 802.11)

LAN

In wireless, it is so not straightforward

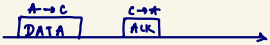
→ Cannot collision detect when we are transmitting our self.

→ Also carrier sense is difficult due to hidden terminal nod

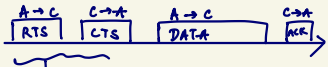
New-protocol

CSMA-CA

↳ collision avoidance



virtual carrier sensing

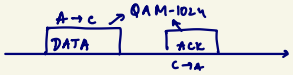


↳ optional, can be disabled

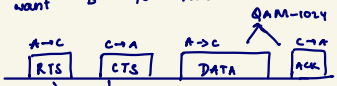
Why would one switch off RTS & CTS? This is overhead and it takes a lot of time ∴

Different modulation schemes

BPSK, QPSK, QAM-16, 64, 256, 1024



I want B to hear RTS & CTS of A, C



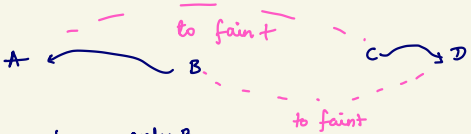
10's of bytes

→ We use BPSK for RTS & CTS

→ Because we want to avoid bit errors

→ So we use less modulation because we detect with low probability of other.

exposed terminal problem



A can hear only B

B hears A, C

D only hears C

B → sends to A

C → sends to D

B → A, C → D possible together in theory

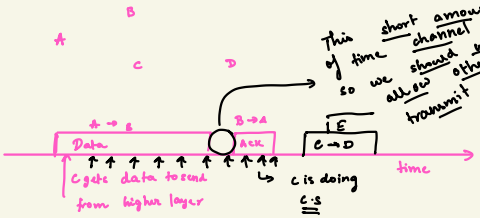
Case -1)

RTS/CTS enabled

If B starts first C remains silent (due to RTS).

When we do RTS/CTS we block C from sending data to D because C does C.S. But it could have sent to D because B's signal is too faint at D. So it would not affect C→D transmition.

So due to WIFI MAC protocol could not allow such theoretical transmission.



This short amount of time channel empty so we should let others to allow transmit before B send ACK. B needs time to process data

should C transmit immediately

→ in ethernet we allowed also immediate transmission because we could almost immediately detect collisions, Here we cannot unless we get an ack, so collision here are expense because if two simultaneously transmit we have to wait for ack

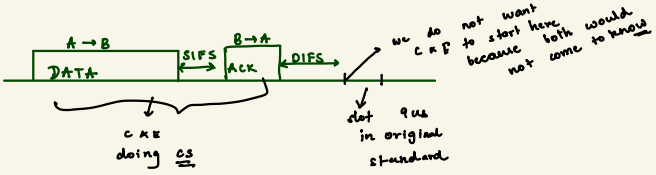
C going in Rx. mode now B has to switch to transmission

B has to receive DATA frame check for errors send frames

Rule : C has to wait at least DIFS

DIFS = (SIFS + 2 * slot duration) for channel to be free before trying to transmit

Assuming DIFS ends



so C & E choose contention window $r_C \in \{0,1,\dots CW_{max}\}$

$r_E \in \{0,1,\dots CW_{max}\}$

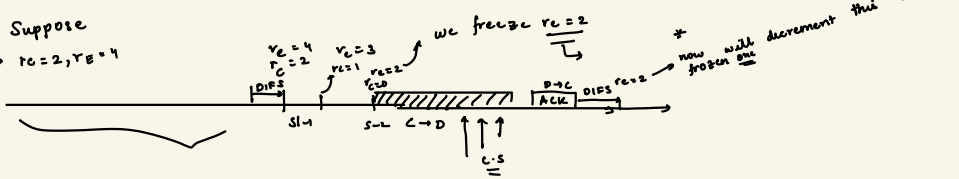
Init $CW_{max} = 15$ → upper limit 1023

For collision

$CW_{max} = 2 * CW_{max} + 1$

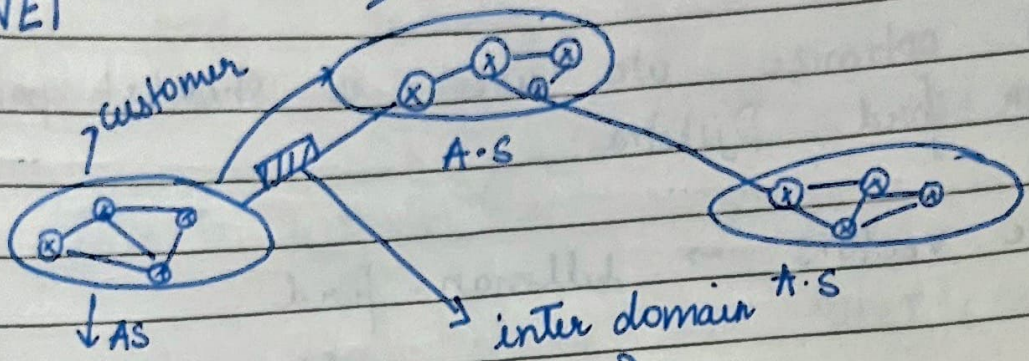
Suppose

→ $r_C = 2, r_E = 4$



NET

ISP → internet service provider



↓ AS

Intra-domain (RIP, OSPF) ISIS (inside AS it can decide its protocol)

inter domain (BGP) → This is between AS's

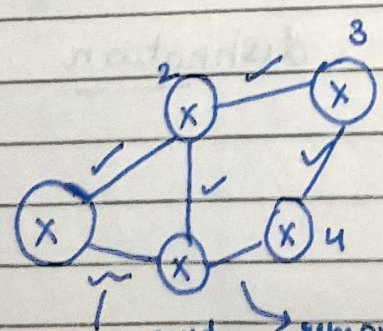
Customer → IDS

Service-level agreement (SLA)

Ex
- 100 Mbps
- > 99.9% UP-TIME
- 30 ms latency within own AS
- Packet-drop rate < 1%

} TCP is designed to create congestion

intra-domain Routing (interior-gateway protocol) (IGP)



↓ spanning tree protocol disadvantage

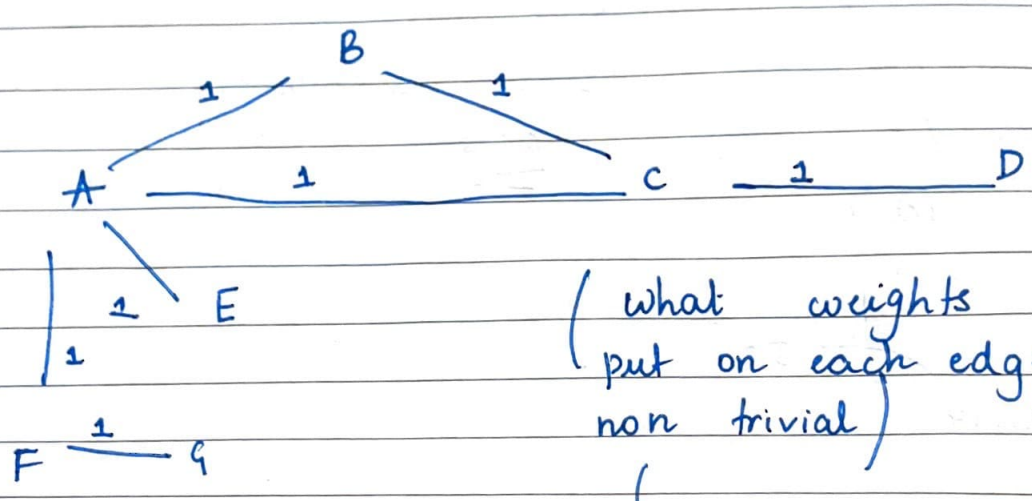
if 1 wants to send to 4 he has to via 1→2 2→3 3→4

if we use shortest

So to optimize we use a shortest path algo

→ Bellman-ford, Djikstra.

Distance vectors → bellman-ford



(what weights to put on each edge is non trivial)

At A (init):

DEST	COST	NEXT-HOP
A	0	-
B	1	B
C	1	C
E	1	E
F	.	F

* min-cost known to various destination

At C (Init)

DEST	COST	NEXT-HOP
C	0	—
A	1	A
B	1	B
D	1	D

→ Each node initially knows about its immediate neighbour

→ But we want whole network to be known to each network :-

→ So each nodes shares its DEST | COST columns of table

This is known as distination vector.

So update table after A gets info from c.

DEST	COST	NEXT-HOP
A	0	-
B	1	B
C	1	C
E	1	E
F	1	F
D	2	(C) ↓ info from c

if A edge B edge had 5 value, after getting info from (C) A → B path can have value 2

DEST	COST	NEXT
(A)	0	-
B	2	(C) → Because going from A→C reduces load.

Finally at (A)

→ Periodic exchange of information
→ Convergence of tables :-

At A

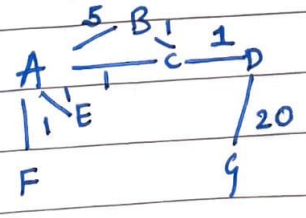
DEST	COST	NEXT-HOP
A	0	
B	2	C
E	1	E
F	1	E
D	2	C
G	2	F

D.V exchange periodically with neighbours
After few iterations, tables converge

what F → G fails (This entry is no loger valid so F should tell all its neighbours it can reach G →
So F says my distance to G → ∞

So A should also send info to its neighbours it can reach G
Everyone updates their table :-
So

new graph

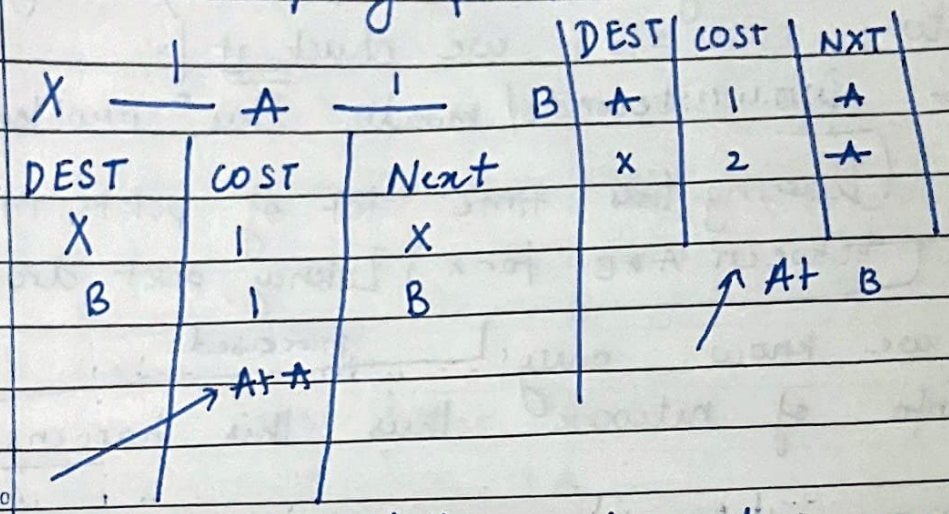


At A

DEST	COST	NEXT-HOP
G	∞	✗
	—	

Page :
Date :

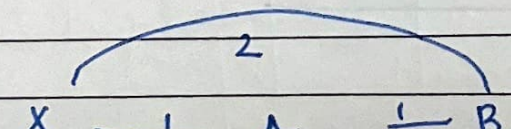
count-to-infinity problem



Suppose $A \to X$ fails A sends message to B (X, ∞) But B sends his table $\begin{pmatrix} X,2 \\ A,1 \end{pmatrix}$

When A was sending B sends this

so A can think the topology to B is like



(But this is not the case)

Now A makes table

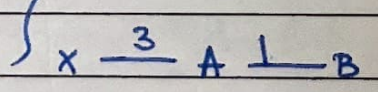
DEST	COST	NXT
X	3	B
B	1	B

Then sends to B $(X,3)$

B chucked out (X) But gets $(X,3)$ from A

DEST	COST	NEXT
~~B~~ X	4	A

Assumed network topology



Now B sends this to A

A makes $X = 4$ then B makes $X = 5$

this keeps happening till ∞

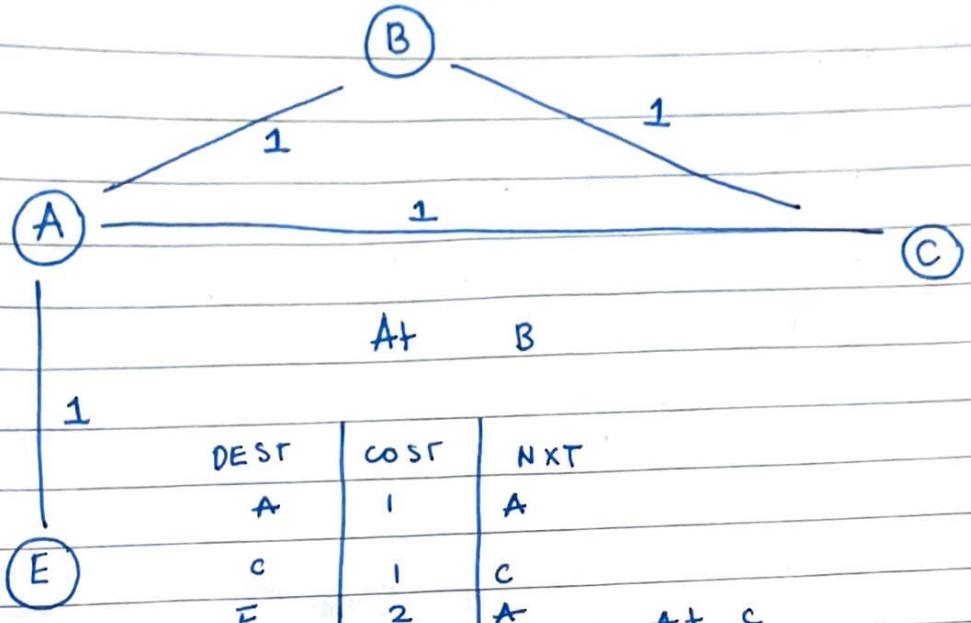
→ So we have loop
→ So RIP (routing information protocol) say ∞ = 16
→ So after COST = 16 we chuck it!
So after few seconds/ minutes this problem is caught! (During this time lot of pckts in between A & B for x) Latency, pckt drop increased
→ This is because we know only partial info of network thus this happens

SPLIT-HORIZON : DON'T share DV. Entry with a neighbour, if that neighbour is next hop to concerned destination.
Ex : B does not tell A send (x,2) to A since A is NXT-HOP for x
↗ Here we hide information

POISON REVERSE
TELL NEXT HOP NEIGHBOUR, my distance is dist. (cost) to dest is ∞
Ex: B tells A (x, ∞) instead of saying (x, 2)

↗ They are not perfect } * Some failures here too

FAILURE OF SPLIT HORIZON



At B

DEST	COST	NXT
A	1	A
C	1	C
E	2	A

At A

DEST	COST	NXT
B	1	B
C	1	C
E	1	E

At C

DEST	COST	NXT
A	1	A
B	1	B
E	2	A

Suppose A-E fails ! 0

A → B (E, ∞)
A → C (E, ∞)

→ A sends message to neighbours

⇒ Suppose A → C (E, ∞) gets lost

→ So ~~sends~~ C → A no entry for (E)

" B → A " " " "

Now table of B & C

DEST	COST	NXT
A	1	A
C	1	C
E	∞	+

DEST	COST	NXT
A	1	A
B	1	B
E	2	A

* So B sends this to C | (E, ∞) , C sends (E, 2) to C

Now B gets entry for E → (E, 3)

→ Because B think it can reach (E) by C

Now B sends to (A) . since E not NXT HOP for E

→ So A adds entry for (E, 4) B

→ Now " sends to (C) | NOT B E update E 5 A

→ E sends (E, 5) to B, B makes (E, 6), sends to A (E, 7) | B for entry of A . . . till ∞

ADV. OF DV

very simple

DIS ADV OF DV

creates count to ∞ problems

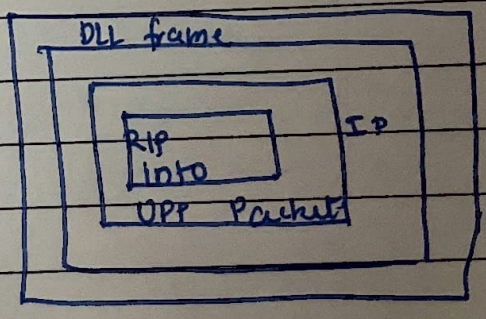
LINK STATE ROUTING (OSPF, ISIS) → DIJKSTRA

↓

open shortest path first

RIP (sends distance vector as OSPF a UDP packet) which has RIP info

transport ← UDP ↓ NET (IP) ← DLL ← PHY



→ This will be allowed to modify Routing tables → These modify local routing table

INTRA-DOMAIN ROUTING

D.V - BELLMAN-FORD

each node tells neighbours : distance to all others

+ simple

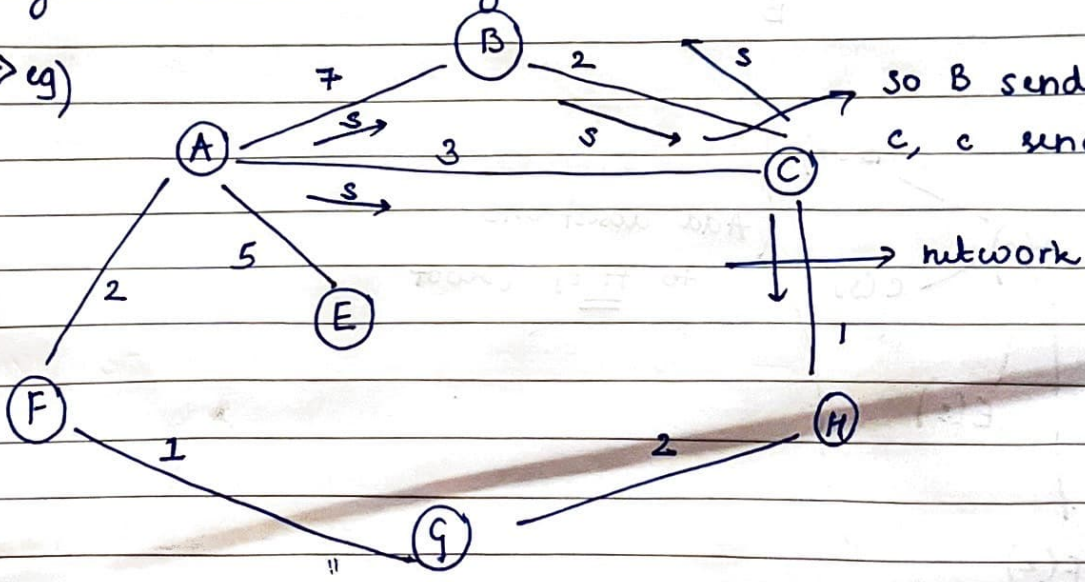
- count to ∞ } → We can use this for small
- convergence time high } network (10-20 nodes)

→ To avoid count to ∞ we do RIP (where ∞ = 16)

S.R: Link state routing → djikstra :-

↳ each node tells all others in the network :
My distance to neighbours ↳ BROADCAST

→ eg)



A broadcast message to everyone that

A: A → B
A → C
A → E
A → F

→ A tells to everyone that is connected to A
→ everyone and forwarded throughout network, everyone gets the message

Forward

A → C → D → G

→ So there are loops, we can one node recieving the message twice, so it should not forward it again more also

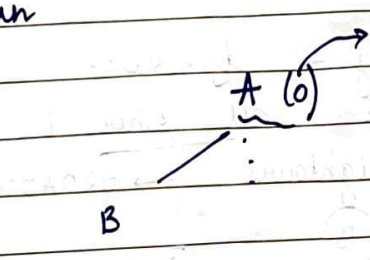
→ We want to sent to all neighbours who have probably not got the message

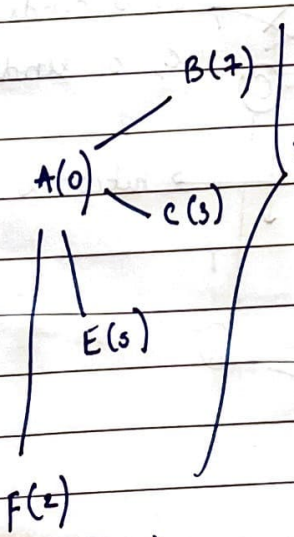
→ So ⊕ of this everyone has information of the full graph, so they can run Djikstra

Dry Run

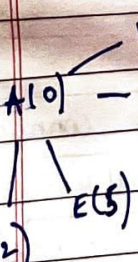
* shortest path tree from A to all others

Every node should the link that one node has for shortest

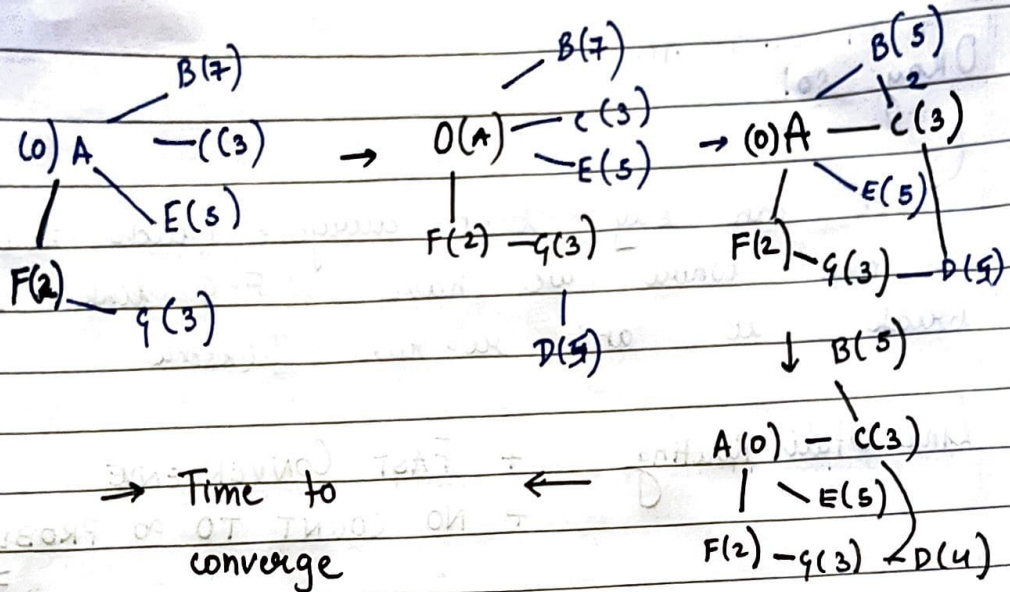




Add closest one to A(0) (root



→ now we have add F & its neighbours



→ Time to converge to this graph

Time for broadcast + Algorithm time complexity
↳ O(diameter of network)

Dijkstra

DEST	COST	NEXT
A	0	—
B	5	C
C	3	C
D	4	C
E	5	E
F	2	F
G	3	F

→ This algorithm does not (not) run on one node this keeps during broadcast

→ Fast as compared to distance vector

Suppose F → G fail

→ So F broadcast I cannot reach G } All re-run
→ So G " I cannot reach F } Dijkstra

→

→ Faster than D.V ↳ May have loops during this time

Okay so!

↳ We can say that everyone (each router has a tree where we have F-G link, we break it and re-run - Dijkstra

Link State Routing + FAST CONVERGENVE
+ NO COUNT TO ∞ PROBLEM

- we need to have more information to run Dijkstra
- need full topology info
- Full Broad cast for any change
- Djikstra algo must be run

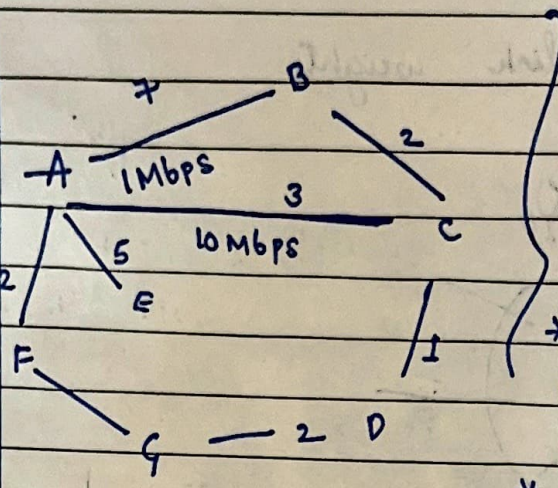
→ Count to ∞ avoided as here we have full topology

→

Q: what should link weights be?

→ Links with extreme high weights would not be used!

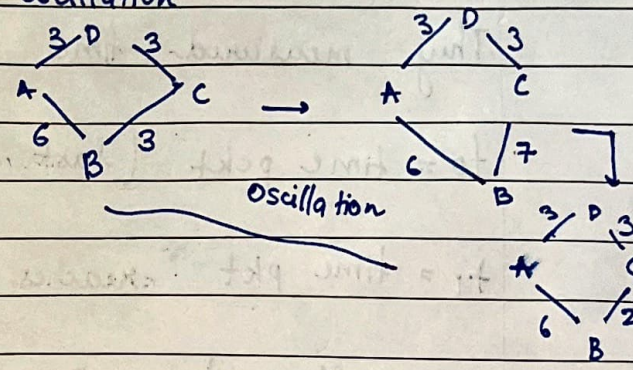
→ So carefully choose that we can include!



* So what if A-C gets full then in that case actual speed < 10 Mbps

* if we update weights it should broadcast that
* then all re-run Dijkstra
* continuous changing diverts traffic * these can cause oscillation

*

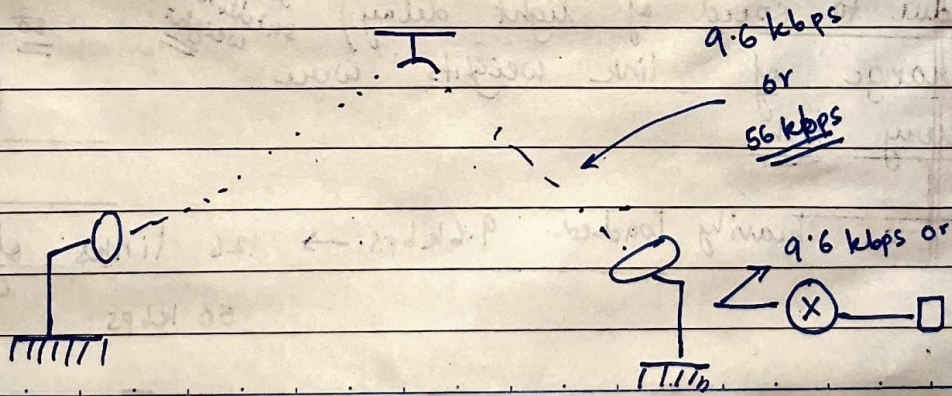


∴ RIP: default weight = 1 for every link (D.V)

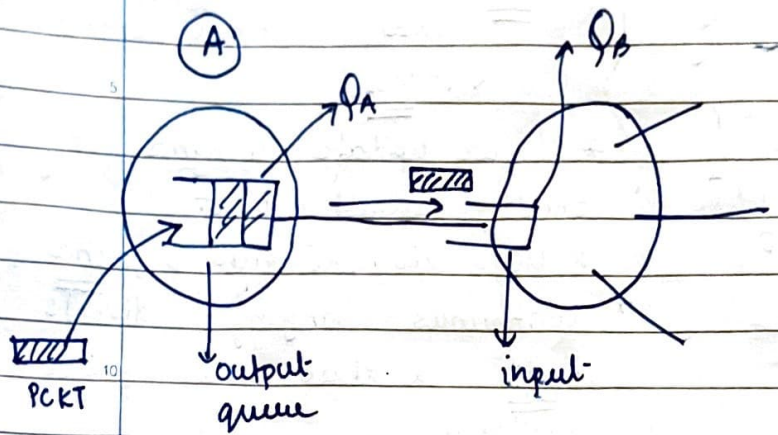
OSPF (LSR): weight $= \max\left(\frac{10^8}{\text{Link speed in bits/sec}}, 1\right)$

Original Arpanet :- (1969 -)

They had low speed satelite link



So what should link weights



They measured time to reach other queue

t_0 = time pckt (last bit) enters $Q_{(A)}$

t_1 = time pkt reaches Q_B (last bit)

measure $(t_1 - t_0)$ = Queuing delay (at Q_A) + speed of light delay + transmission delay (→ time empty bits of pckt on wire (bits pieces) so $\frac{P}{C}$ time)

AVG these delays over some time period } How they gave weights

Set link to this avg of delay

Issues

→ 1) OSCILLATIONS

2) satellite links were penalized a lot (due to speed of light delay) → 9.6 kbps terrestrial links had less weight than 56 kbps satellite weight

3) range of link weights were very high

↳ heavily loaded 9.6 kbps → 126 links of 56 kbps

Date : _____

Count to infinity problem :-

D-V

Routing Table

Dest	Next	cost

X —1— A —1— B

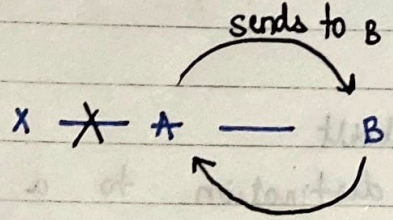
Dest	Next	Hop
A	A	1
B	A	2

↑ at X

Dest	Next	Hop
X	X	1
B	B	1

Dest	Next	Hop
X	A	2
A	A	1

Suppose X → A fails

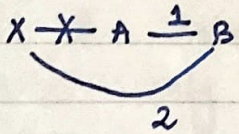


This is happening simultaneously!

((B → A) (A,1), (X,2))

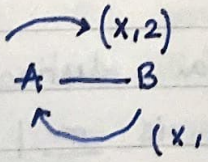
info !

B tells my distance to X is 2 then assume there is a path form B → X



(But this not true !)

B changes its table and makes (X, ∞). Now again both send messages



So A makes update

(X, ∞)

(B,1)

Date :

→B makes

X ✗ A — B → update X,3 vai

→ B thinks there is a path from A to X with weight 3 then make an entry

Table		
X	A	4

→ This keeps happening till ∞

→ This is called count to ∞

So if A had packet for X it would it to B, B would send it back A.

→ So RIP allows a max distance of 16

cost = 16 = ∞ = implies we cannot reach it

Solutions to Count ∞

Split Horizon :- information about

Do not advertise to a destination to a neighbour if the neighbour is the next hop to destination

X —1— A —1— B

Now X — A fails

Now A adv (X, ∞), simultaneously B sends (A, ∞) or does not send about (X) so, all it sends (A,1)

B TELLS NOTHING" ABOUT X

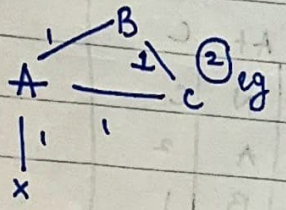
Split Horizon with poison reverse!

A node tells its next hop to a destination that its distance to the destination is ∞!

Date :

So " B sends Routing ADV (X, ∞) to A "

Another Example



At B

Dest	Next	Cost
A	A	1
C	C	1
X	A	2

At A

DEST	NXT	HOP(cost)
X	X	1
B	B	1
C	C	1

At C

DEST	NXT	COST
A	A	1
B	B	1
X	A	2

At X

Dest	NXT	COST
A	A	1
B	A	2
C	A	2

→ A to X link fails and A send to both (X, ∞) B, C that X is unreachable

→ But the message to is C is lost!

→ So B makes (X, ∞) in its table and sends this to C only not to A as we are using split horizon :-

→ But C can tell abouts its distance X to B as A is next hop not B.

→ So C sends $(X, 2)$ (periodic update)

→ Does not tell this to A as using split horizon.

So Now current state of table at B

D	N	C
A	A	1
X	C	3

Date :

So now B thinks there is path from c of dist 3 to X

→ Now B sends (X,3), (C,1) to A as A no more next hop for X.

→ Now finally table Status

At A

D	N	C
X	B	4
B	B	1
C	C	1

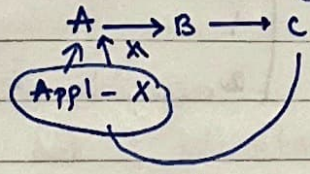
At B

D	N	C
X	C	3
B	B	0
A	A	1
C	C	1

At C

D	N	C
X	A	2
B	B	1
A	A	1

Now suppose any of them gets an message for X it will go in a loop for

eg 

So after four round ttl travel = 16 ∴ not reachable

→ Do RIP stops it.

RIP: Routing information protocol :- (D.V based)

Cost of all links = 1

max cost allowed to a distination is 16 (16 = ∞)

Advantages of DV

→ Simple & easy to implement (As no complicated algo, no BFS, ---)

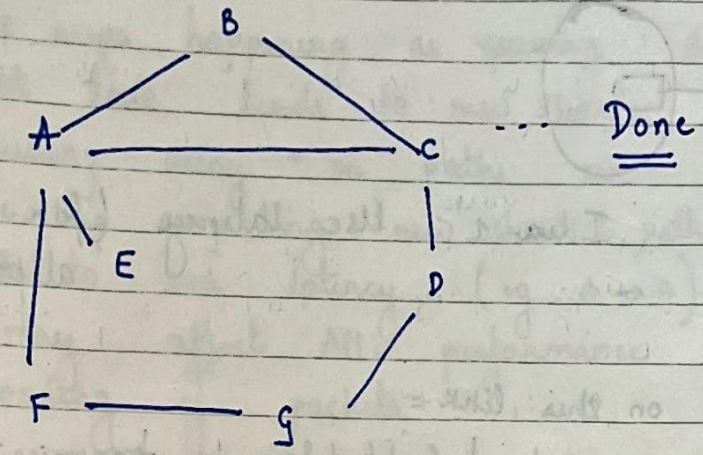
→ Disadvantages

- count to ∞ & routing loops :-
- converge of routing table

→ So due this we do LSR

Date : _____

Link State Routing !!



--- Done

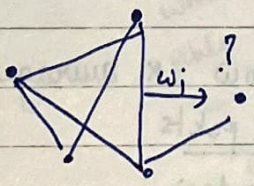
ADV:

No routing loops, no count to ∞ problems, convergence is faster of routing tables.

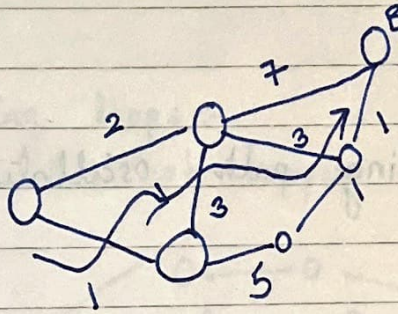
→ Disadv: Algo is more complex than D.V

How to choose LINK WEIGHTS

DV, LSR
↓ ↳ OSPF, ISIS →
RIP



eg)

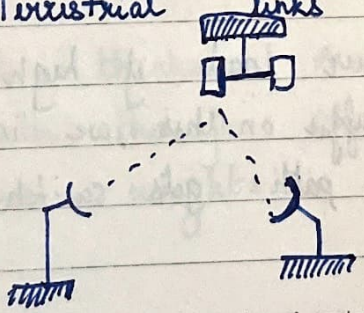


So how most packets will be sent through

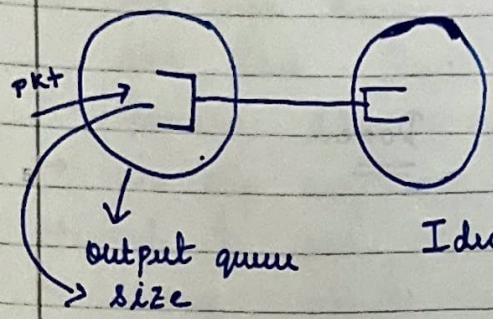
Arpanet :- two types of links 9.6 kbps, 56 Kbps

↳ U.S.A

Terrestrial links, Satellite links



Date : _____



Idea 1 : Use latency (for weights allocation)

latency of pkt on this link =
queue delaying + speed of light delay + transmission delay)
(when 1 first leaves) ↑ pkt

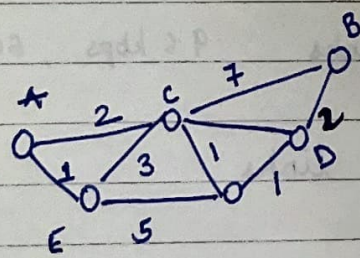
→ this same as time to put bits of wire

latency of packets keep changing ! (due to queuing delay & transmission delay).

Take time window & average out latency.
= latency of all pckts / time window

ISSUES :-

① under heavy load, routing path oscillation



→ We are using path A-E-C-D-B

But since our load if high lot of traffic on this, we increase weight, so path gets switched to A-C-B.

Date : _____

→ Then again since traffic shifts again we take another path :..

→ This keeps happening as queuing delays keep changing and this leads to new shortest paths.

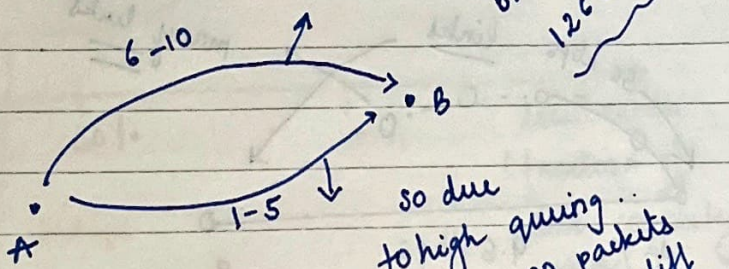
→ Queuing delay ↑ on paths use

→ We keep computing new shortest paths

→ end to end latency (eg. A → B) keep varying

→ ↳ may affect Appl performance

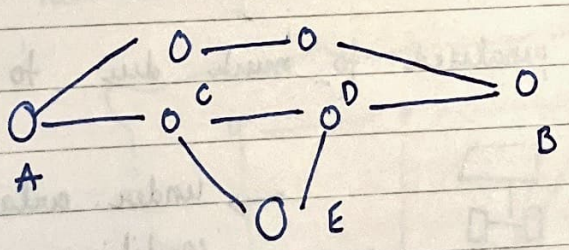
→ reordering of packets :-



so due to high queuing.. delays so packets when sent ah diff pckts could come earlier than expected ones.

Routing loops :-

→ This does not happen when weights are fixed.

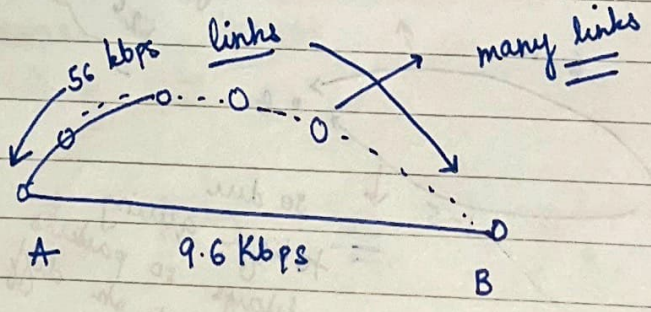


A thinks A-C-D-B is shortest path, c thinks it is A-C-E-D-B, even though what thinks, C does not have updated link weights.

Date : _____

→ So C sends it E, E sends it back to some other node or back to C

(2) Range of link weights were very large Because of this some links where penalized (due to very high weights).



9.6 Kbps ≈ 126.5 Kbps in some cases!

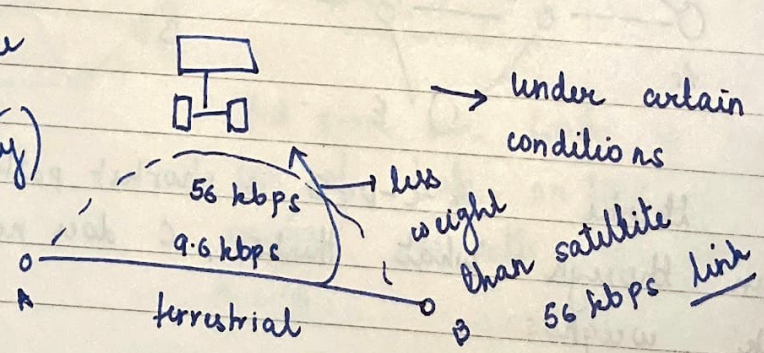
Link speed → queuing delay
→ transmission delay

transmission delay also affects queuing delay.

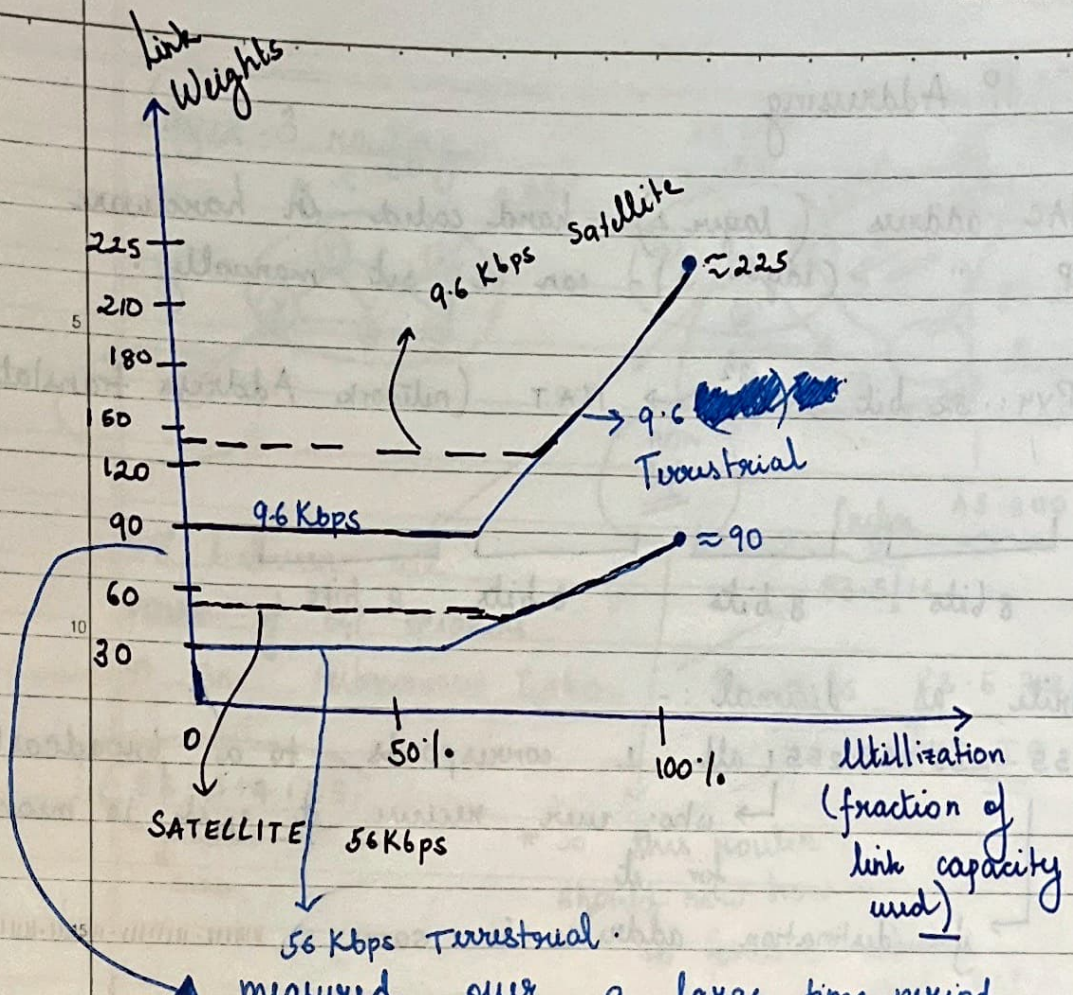
Is this logical?

3) Satellite links penalized to much due to transmission delay.

(No this due to speed of light delay)



Date : _____

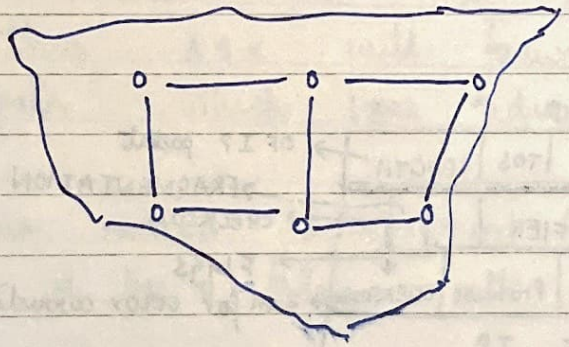


measured over a large time period

Today

OSPF :- wt of link = $\max\left(\frac{10^8}{\text{link speed (bps)}}, 1\right)$

NOC : network operation Centre (AT&T)



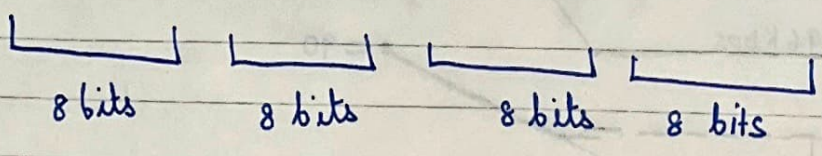
(They could see utilization using color coding)

Date : _____

IP Addressing

MAC address (layer 2) - hard coded in hardware
IP " (layer 3) - can be set manually.

IPv4 : 32 bit $\sim 2^{32}$ ⇝ NAT (network Address translation)



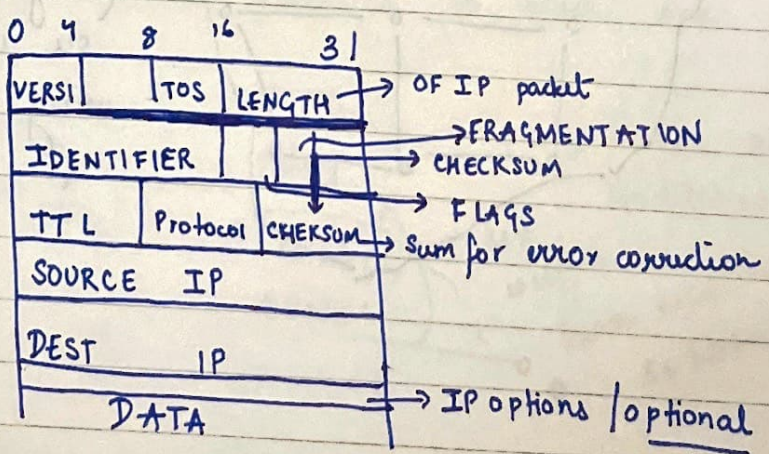
write as decimal :-

Ex 255 . 255 . 255 . 255 (all 1 corresponds to a broadcast)
↳ who ever recieve it, it is meant for it
↳ if destination address is same

10 . * . * . * } Private IP
192 . * . * . * } Reusable } can be used in their own originations

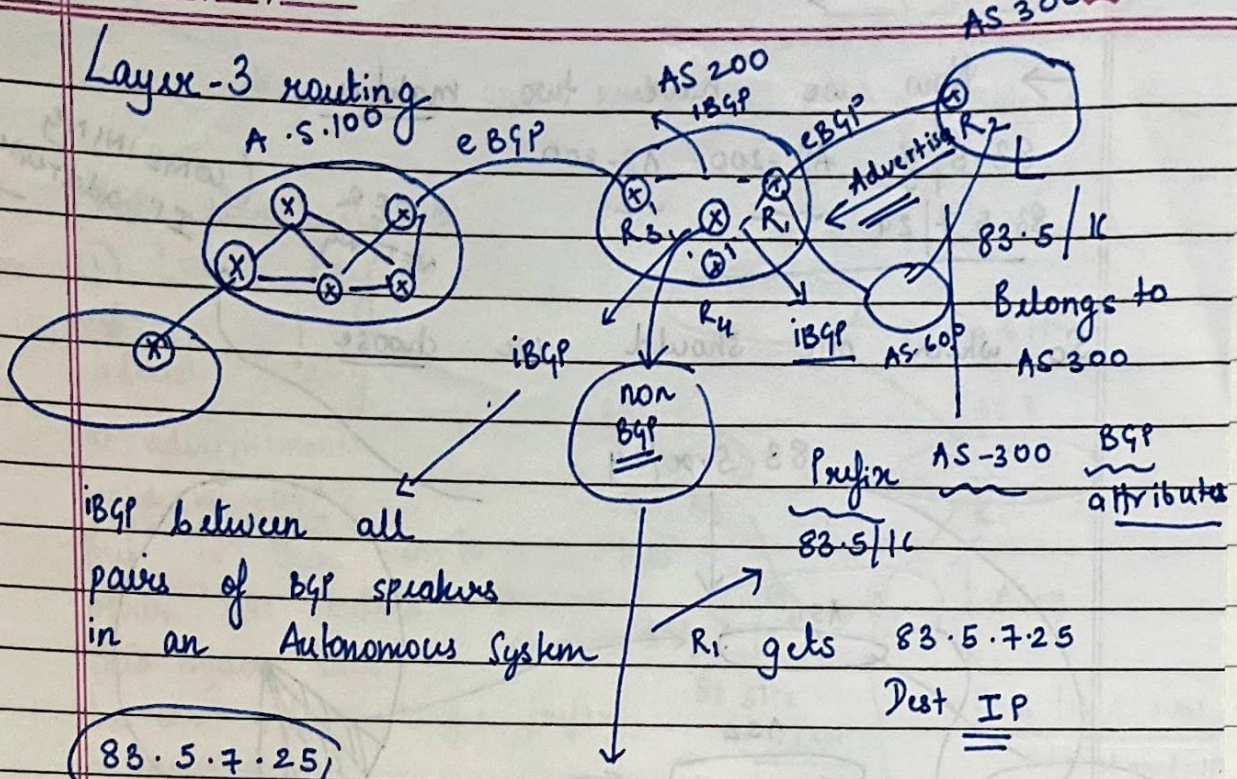
public IP ⟶ should be unique in the internet
↳ only one host must use that IP address.

IP Header



Date __/__/____

Layer-3 routing



AS200 → AS100 : 83.5/16 AS200-AS300

→ Golden rule : If an A.S (x) sends an advertisement for <prefix> x-y-z... to a neighbour, and the neighbours (sends a packet whose A.S x a packet) sends this BGP speaker a pkt with dest IP matching the prefix, then A.S.x will forward it along A.S path which was advertised.

→ So if we receive advertisement from node to BGP then it has to follow a path

→ Exceptions 83.5/16 | AS200 - AS-300

AS200 adv AS100 → shorter prefix what if AS-200 → AS-100

Longer prefix ← ~~83.5/16~~ | AS200 - AS600 - AS300

Given 83.5.7.32

Date __/__/__

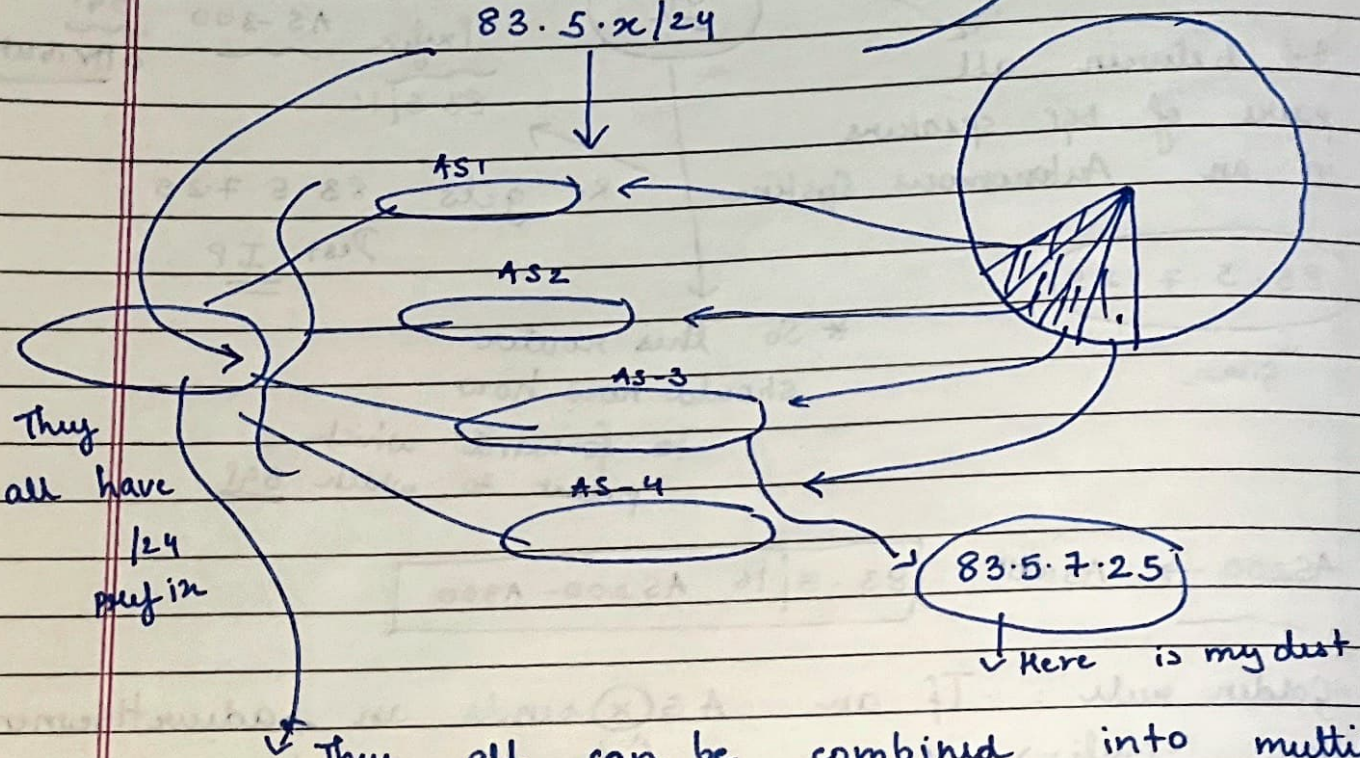
→ Now we have two matches

83.5/16 AS-200 - AS-300

83.5.7/24

SUPER NETTING (COMBINING IP address)

So which one should we choose !



They all can be combined into multiple IP address 83.5./16
x.y

Rule says follow longest prefix match :-

So AS-600 path will be used !

At destination we have the highest match So if we try to take a one with shorter prefix match we can have a shorter path but we may flow away from the actual destination :-

Date __/__/____

One - entry for one prefix in routing table.

BGP - attributes

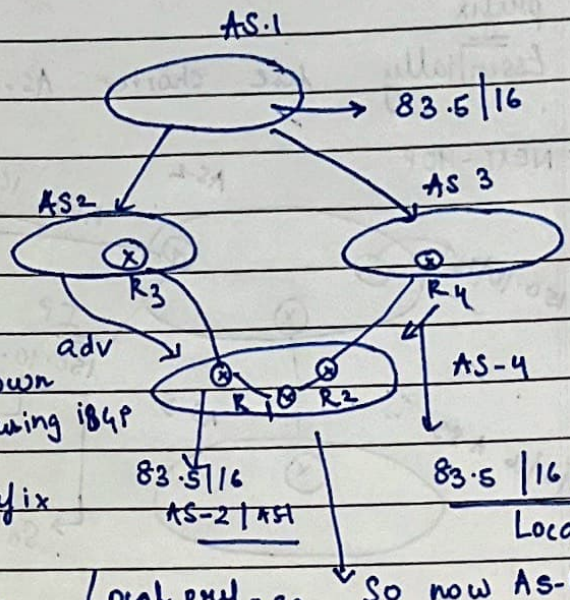
(1) LOCAL - Pref

↓

added locally to advertisements heard via eBGP.

This is then sent to other BGP routers in its own AS using iBGP

This makes sure the one with larger prefix is preferred :-

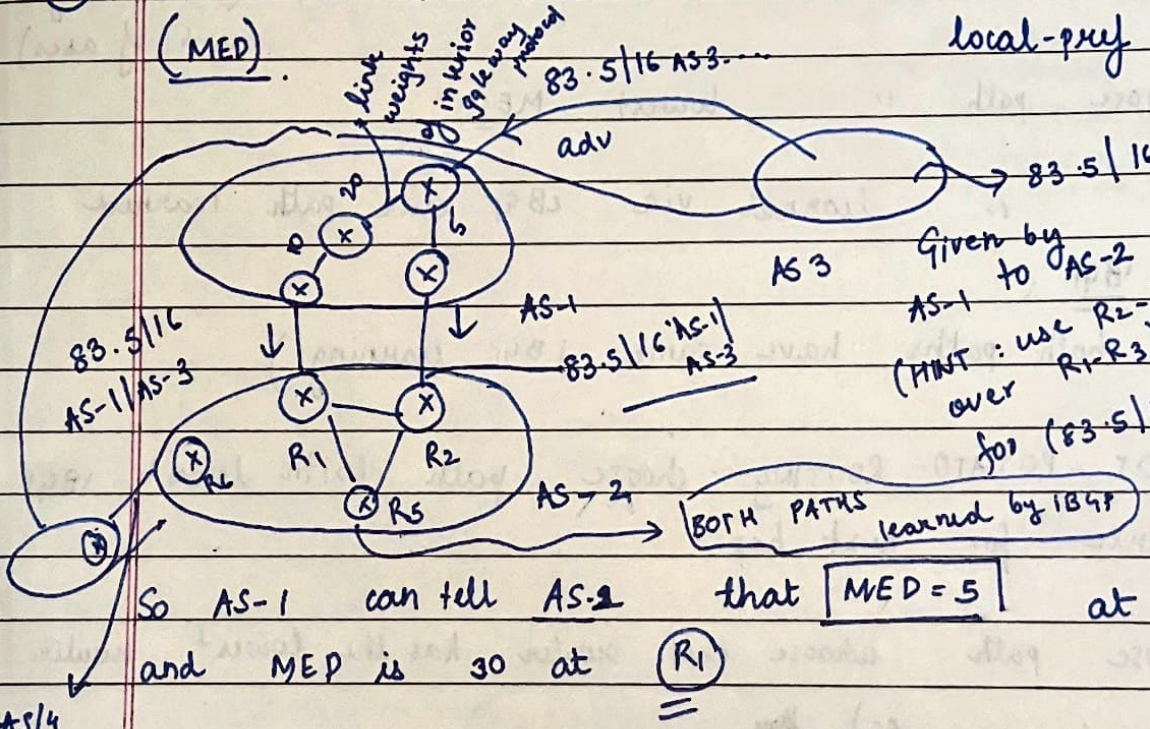


We do not forward local pref attributes, So any advertisement via eBGP we do not put our local pref, the reciever put his local pref

So now AS-4 can decide which path it wants to choose

* R_1, R_2 etch in AS-4 can add local-pref attributes

(2) Multi - Exit Discriminator (MED).



So AS-1 can tell AS-2 that MED = 5 at R_2=5 and MED is 30 at (R_1)

* Rule lower MED values are preferred (for same prefix)

Page No.

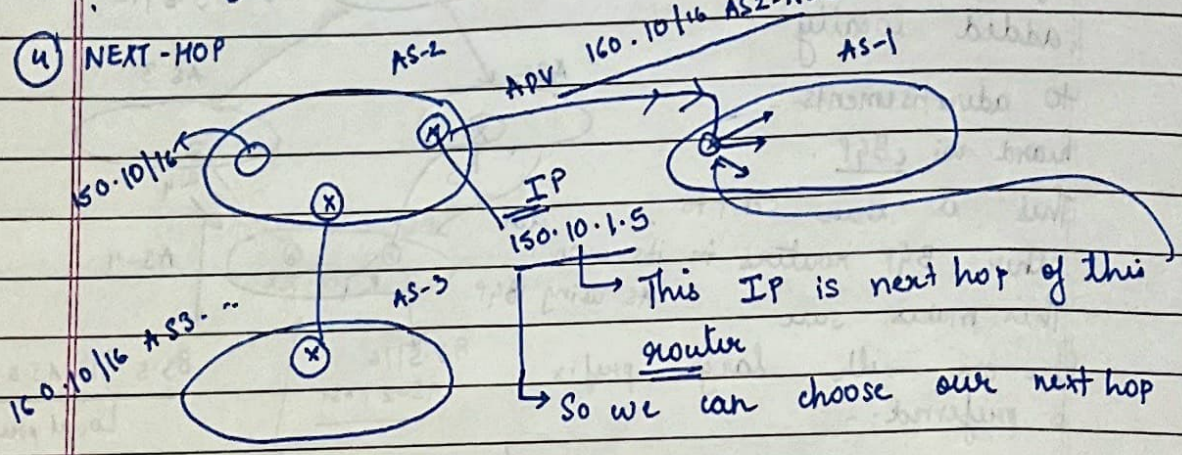
→ It may be logical to compare MED of only 1 Autonomous system Because others should/could be different

Date __/__/____

(3) AS-PATH : list of ASes to destinat^n as which has the prefix

Essentially use shorter AS-paths is preferred.

(4) NEXT-HOP



Rules for choosing paths (used at each BGP speaker) (for each unique prefix)

(a) Choose route with largest local pref

(b) " " " shortest AS-Path (in terms of num of ases)

(c) choose path " lowest MED

(d) " " learned via eBGP over path learned by iBGP :

(if both paths have same iBGP learning)

(e) HOT-POTATO-ROUTING : choose path with lowest iBGP metric for next hop:-

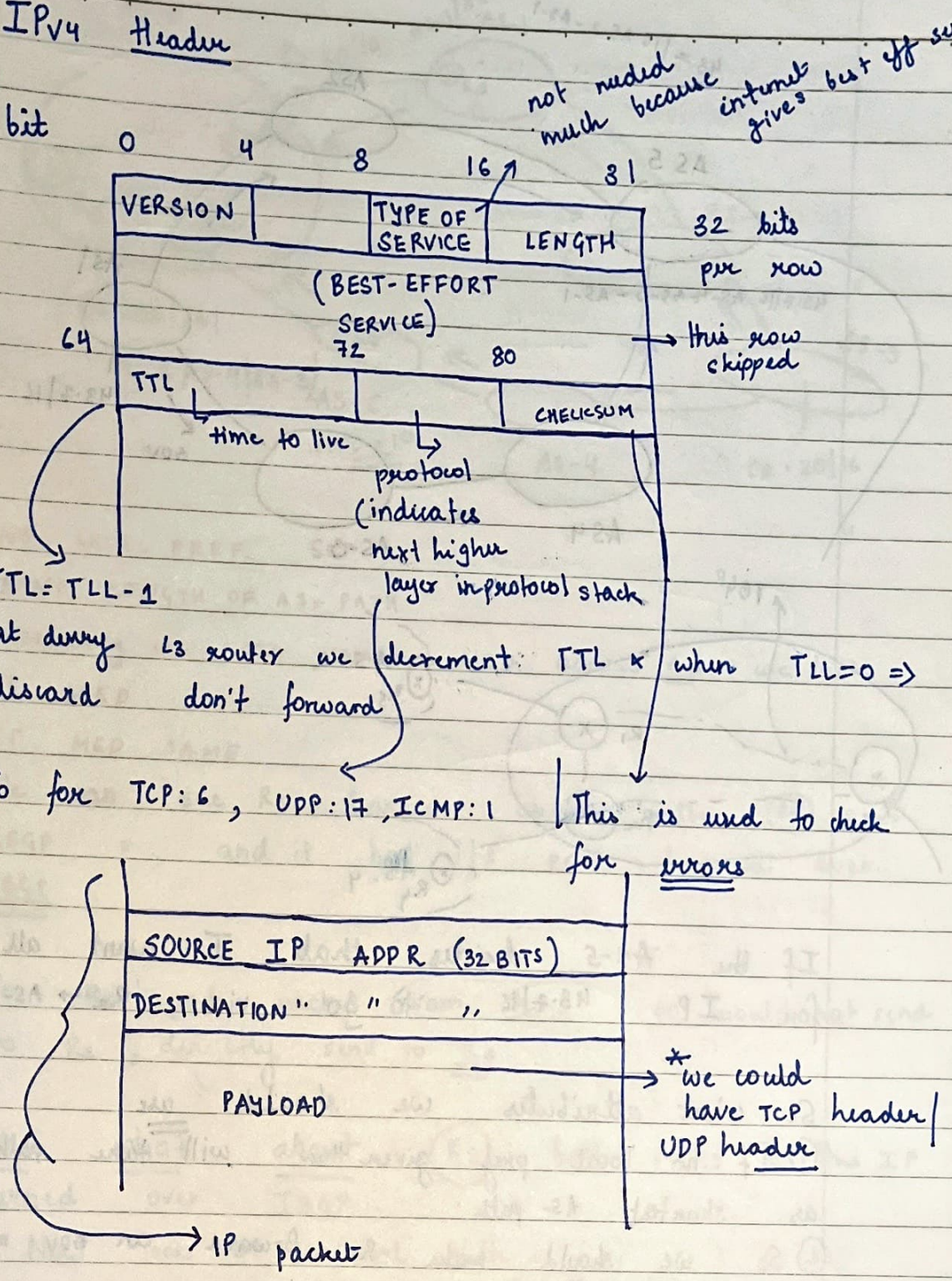
(f) choose path whose exit router has the lowest router ID (in same AS)

↳ (= highest ip address on all interfaces of router

Page No. ☐

Date :

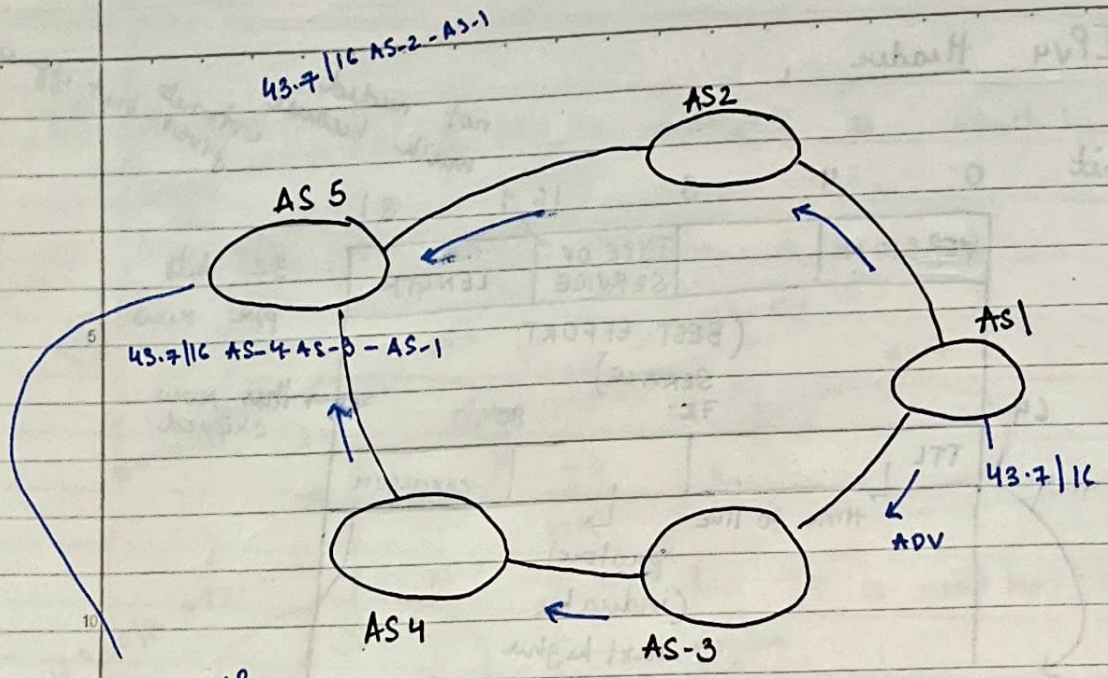
IPv4 Header

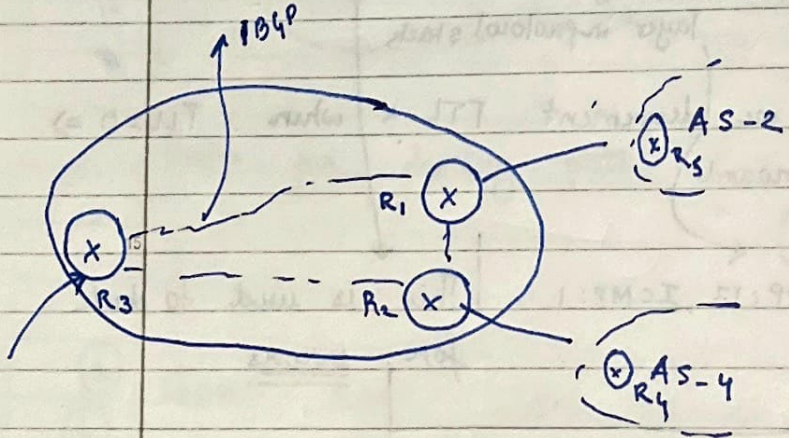


BGP forwarding rules :-
run at every BGP speaker (for each prefix)

1) LOCAL-PREF → larger
2) AS-PATH length - shorter
3) MED → smaller
4) eBGP - over - iBGP
5) HOT-POTATO Router. (shortest IGP to next hop)
6) lowest router ID :-

Date :





If the AS-5 decides that I want all traffic for IP 43.7/16 to go via $R_2 \rightarrow$ AS-4

So what attributes we should use

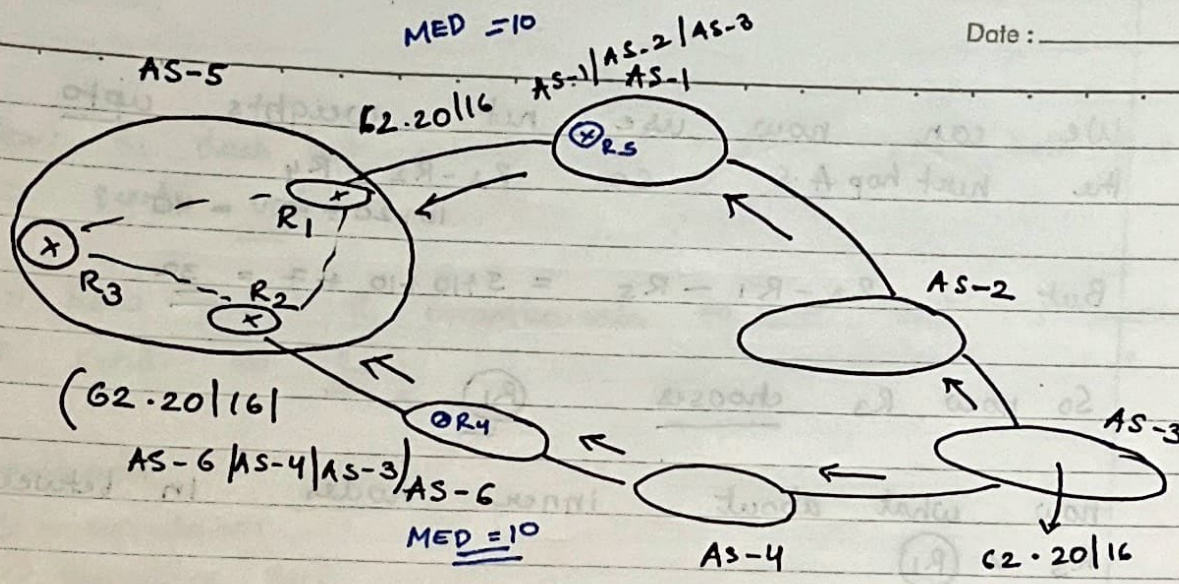
① If no local pref given then will take path from R_1 as shortest AS-path.

② So we should make R_2 forward an ADV over IBGP with higher local pref than R_1 so traffic goes via R_2

→ R_2 finds 43.7/16 AS-4-AS-3-AS-1 . LOCAL-PREF = 10

R_1 " " AS-5-AS-1 " " = 5

Date : _____



→ NO LOCAL PREF SET

→ SAME LENGTH OF AS-PATH

→ Assuming we can compare MED across A.S we look at MED

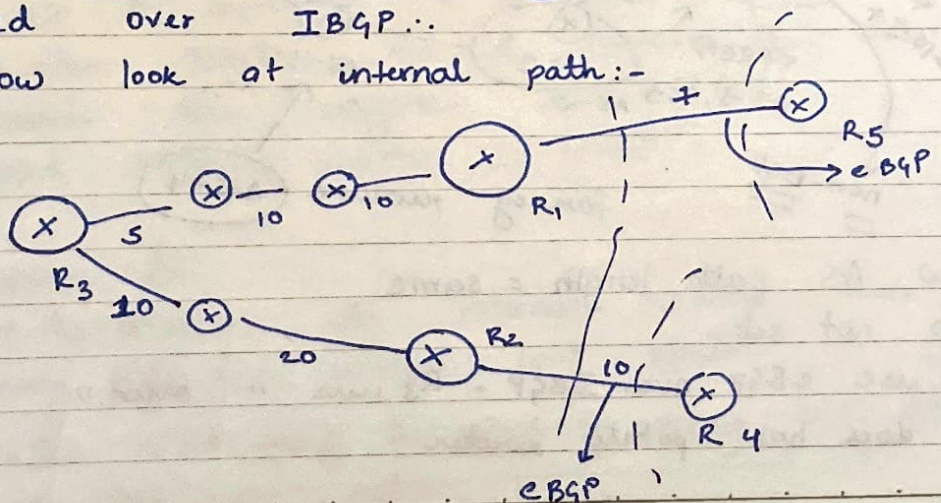
→ IF MED SAME

→ We can see R_1 learned about path (R2) over IBGP * and it has its path learned over EBGP

→ So R_1 got packet some how so would not send to R_2, directly send to R_5

→ Now what about (R3) both paths for this IP learned over IBGP..

↳ Now look at internal path:-



Date : _____

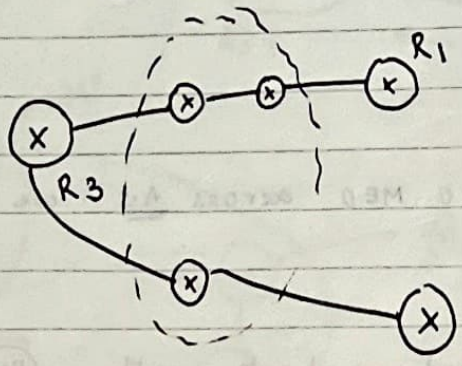
We can now use net weights upto the next hop A.S so $R_3 - R_2 - R_4$

$10 + 20 + 10 = 40$

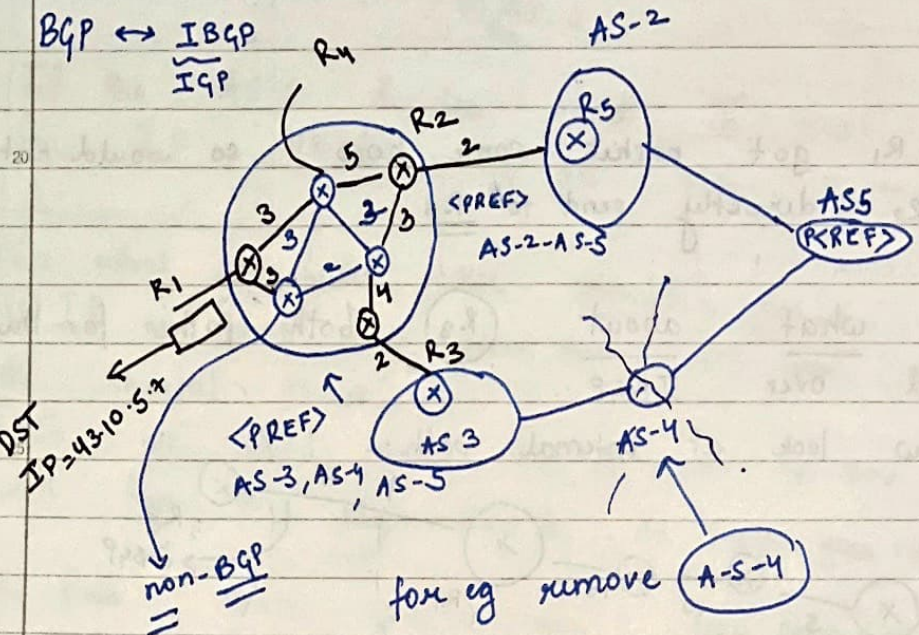
But $R_3 - R_1 - R_5 = 5 + 10 + 10 + 7 = 32$

So now R_3 chooses R_1

now what about inner nodes in between R_3 & R_1



BGP ↔ IBGP
IGP



→ now AS path length = same

→ MED not set

→ R_2 use eBGP over IBGP & R_3 uses "over"

→ R_1 does hot potato router

Date : _____

→ Now R_1 does hot potato as neither eBGP both iBGP so sends via

So how does R_1 communicate to R-4 that it wants to send to R_2 !

Methods

① encapsulation

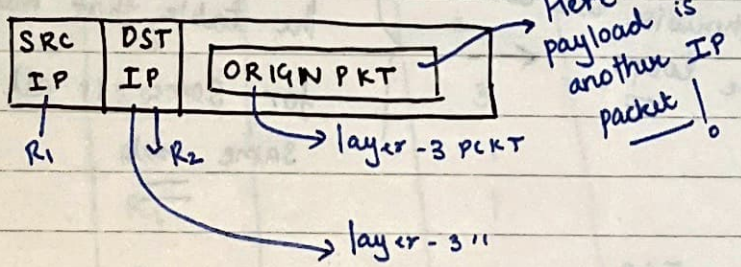
2) Pervasive BGP

3) Tagged IGP

① encapsulat^n (Suppose R_1 gets a packet for AS-5)

(<PREF> IP_AS-5)

So it encapsulates this by putting extra bits & making DST IP – R2



encapsulation of layer-3 into layer-3

→ So now R_1 sends to R_4 as runs Dijkstra

Now when reaches R_2 it removes this outer layer & forwards the original pckt to R-5

→ Payload itself is another IP packet :-

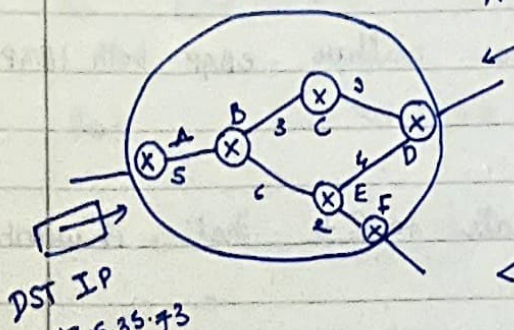
② Pervasive BGP

↳ Assumption : all internal routers run BGP

2) Every prefix has unique exit router in A.S

↳ This could easily be solved by setting local pref for each node for each <pref>

Date : _____



Adv 47.5/16

at A we have a BGP table

BGP table

DST	EXIT
<PREF>	D

IGP table

DST	NEXT HOP
D	B

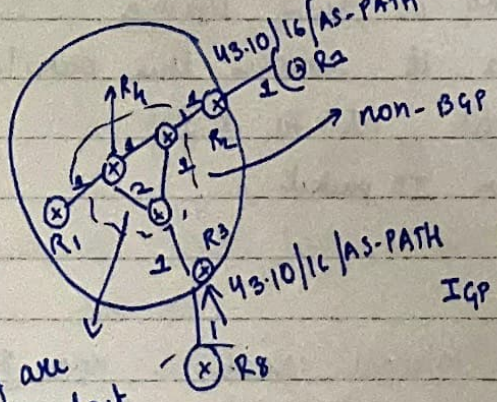
via Dijkstra or BELLMAN

All have the same the table that means for some IP all have same exit

* otherwise we could have loops

→ TAGGED IGP

BGP speaker (router) inserts some tagged information into IGP about prefixes learned via BGP



at R4

IGP table

DST	NEXT	TAG
R1	R1	
R2	R5	
R3	R6	
R5	R5	
R6	R6	

They are useless about these prefixes

Date : _____

R_2 can send information 43.10|16 TAG = R_2
And suppose network (AS) runs OSPF (LSR - Dijkstra)

So R_2 puts 43.10|16 (R_2) → TAG so in table for prefix we get a (entry) so now we can treat this as a node in table & create an Dijkstra table

Now table at (R_4)

DST	NEXT	TAG	COST
"	"	-	∞
43.10|16		R_7	
43.10|16		R_8	

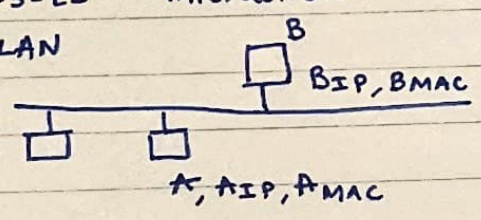
↳ no lost for those

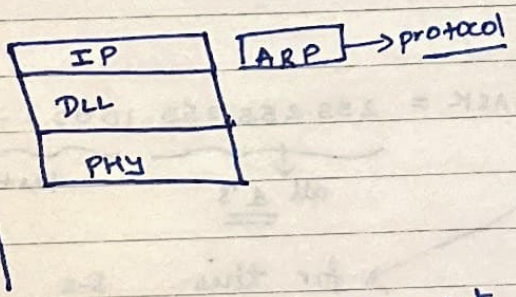
tagged info

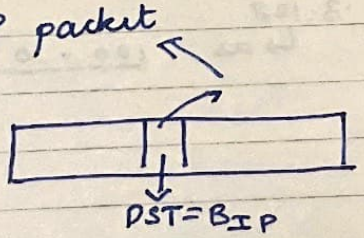
R_1	R_1	1
R_2	R_5	2
R_3	R_C	3
R_5	R_5	1
R_C	R_C	1
R_7	R_5	3
R_8	R_C	3

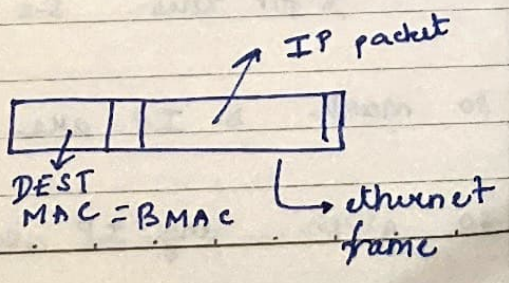
So now we have match for 2 tags so we

L3 - L2 interaction









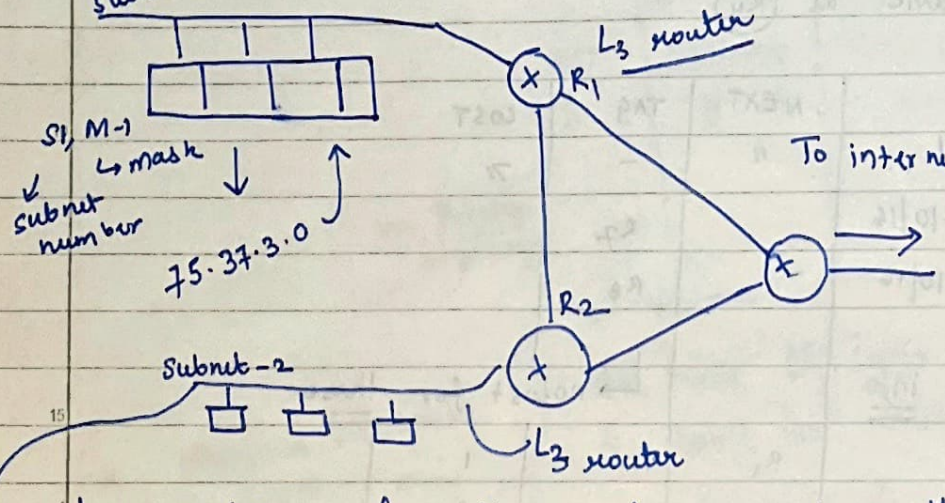
Date :

→ If we do not have MAC address of B what to put DEST MAC -ADDR

→ We use ARP protocol at layer -2 & make broadcast & ask who has BIP reply.

→ So before this SUBNETTING

NETWORK 75.37.3|24 prefix



Last 8 bits for all can be set by setting last 8 bits at L-3 router :-

→ So for all on S-1 put 0000 0000 → last 8 bits

→ " " " S-2 " 10 - - - → " "

Subnet num, Subnet MASK → identifies which bits of IP addr to consider

MASK = 255.255.255.1000 - - -

all 1's — last 8 bits

for this S-2 75.37.3.128

↳ as 100 - 0 → 128

So mask & IP are stored.

So given any IP, say x.

Date : _____

If $(X \& M_1) == S_1$ then X is in subnet-1

" $(X \& M_1) == S_2$ " " " " -2

Table at Subnet Num

R_1

	MASK	NEXT
S_1	M_1	—
S_2	M_1	R_2
0 thus		R_3

Date :

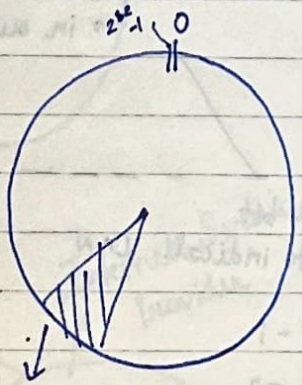
Protocol in IP header is next protocol :-
6 : TCP, 17 : UDP, 1 : ICMP } TCP is above IP
TCP : time to live :-
decremented at each IP router :-
0 => drop packet :

Routing Table :-

Destination	Next hop

→ small in number

IPv4 $\approx 2^{32}$ we need to give some kind of encoded IP addresses



this slice given to IIT B

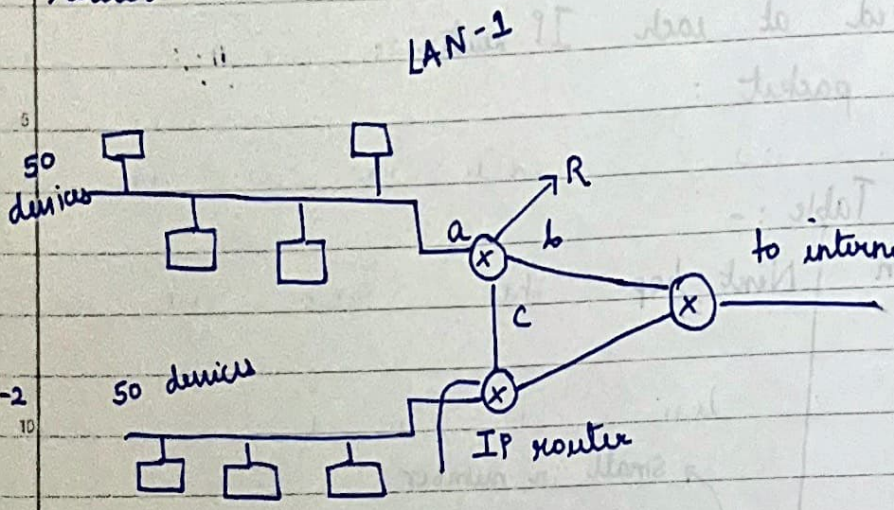
a.b.c.d
↳ octet

730.52.30.* → address space (2^8 choices)
↳ common prefix

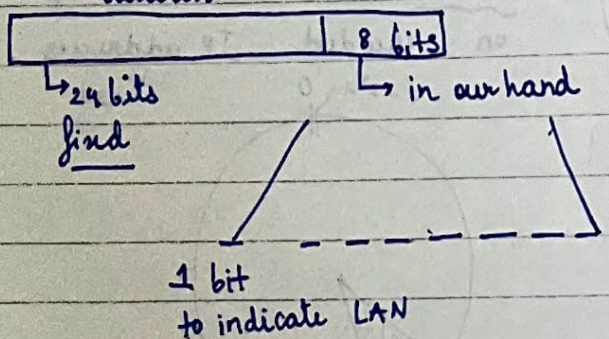
class A : 8 bits Network address (common prefix) 24 bits hosts → 2^{24} hosts
" B : 16 N/w 16 bits host
" C : 8+16 N/w 8 " "

Date :

Subnetting : Given a slice of IP addresses how to divide among LAN's, setup / config, internal router router :-



class C : address



0 — means LAN-1
1 - " " -2

Subnet mask : says which bits in IP address to use to decide which LAN to route.

eg) 25th bit is subnet mask :-

So how to chose mask

EX 11 - - 11 00 - - 0
(25 bits, 7 bits)

Subnet address

S_1 for LAN-1

S_2 " " -2

Router

Date : _____

Suppose R gets a packet and dest IP is 'D'

Is (D and M_1) == S_1 ? [S_1 = 730.52.30.0]

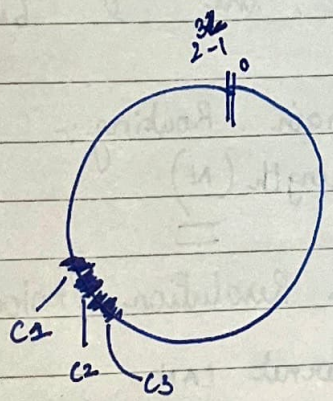
bitwise [S_2 = 730.52.30.128]

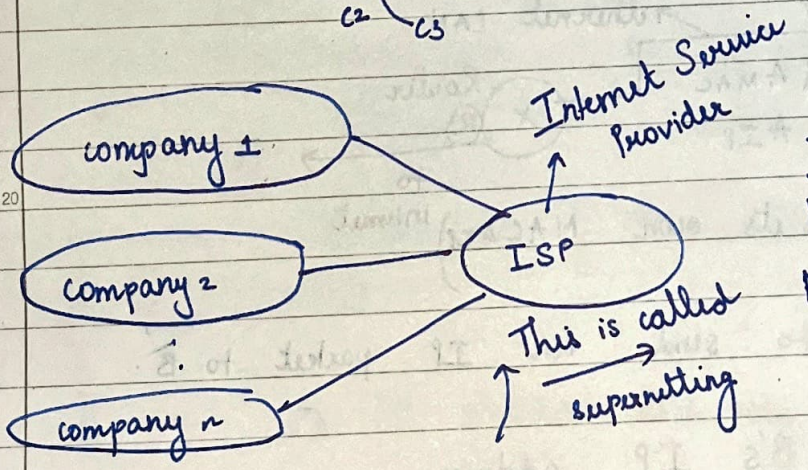
So packet was on interface @ do nothing is

Is (D and M_2) == S_2 then forward on c interface ing

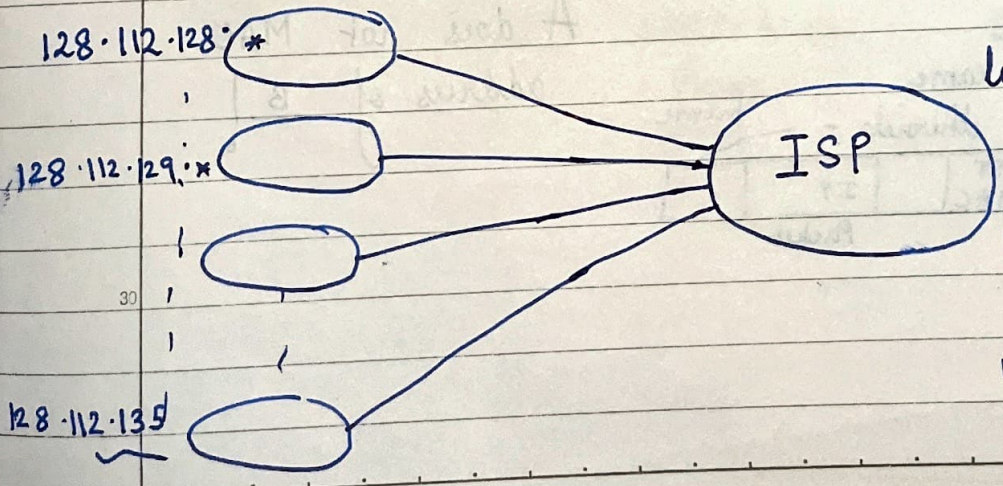
If not S_2 then forward on B.

Ex >





I want to combine them all in one if possible then how to do it !



Look at 3rd octate

128 : 10 - - 000

129 : 10 - - 001

.

.

135 10 ... 111

Date : _____

Last 3 bit only change.

→ So we can write all these as an common IP prefix

128 · 112 · 128 · 0/21

a·b·c·d/N → consider the N leading to get IP prefix

Because these do not change for these company.

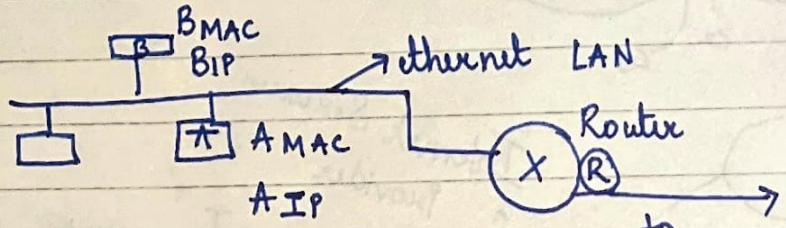
If given dest. IP address 'D'

If first N bits of 'D' match with the first N bits of a·b·c·d ; the 'D' belongs to that prefix.

CIDR: classes Inter Domain Routing :-

→ specify any prefix length (N)

ARP: Address Resolution Protocol



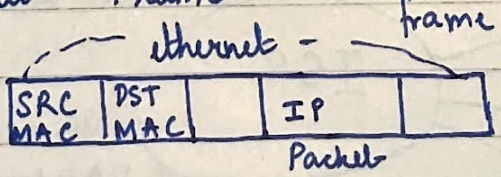
(R itself has its own MAC & IP)

A wants to send an IP packet to B.

A knows B's IP address :-

A does not MAC address of B !

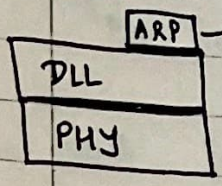
Ethernet Frame



Date : _____

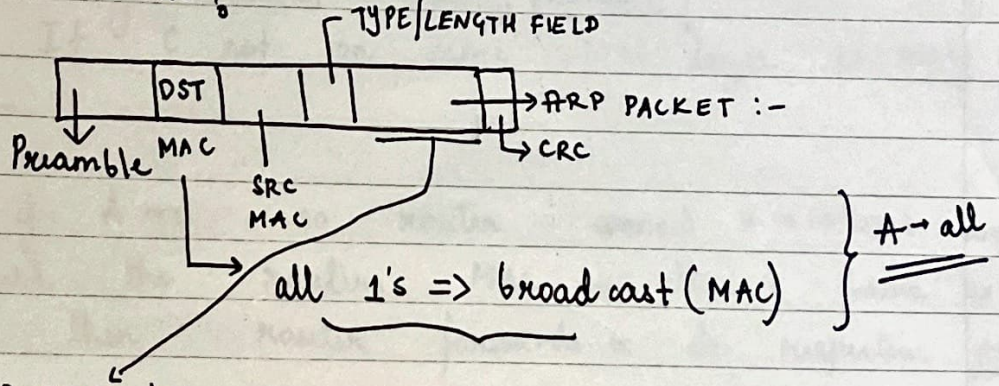
Now does A create ethernet header without knowing B's MAC address.

ARP to the rescue !



→ In practice we do not know exactly what is above what, by this I mean it varies with networks :-

Ethernet Frame !



ARP Request :

Request : SENDER MAC ; SENDER IP

(A_{MAC}) (A_{IP})

TARGET MAC ; TARGET IP

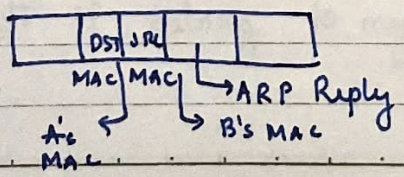
(all 0's) (B_{IP})

So if Target IP matches then only packet useful !

ARP Reply B → A SENDER MAC B_{MAC} SENDER IP B_{IP}

TARGET " A_{MAC} TARGET " A_{IP}

UNICAST FOR A



Date :

Informat^n (BMAC) stored in ARP cache at A

A maintains IP : MAC binding, to reduced every time asking

→ This is not a permenant binding.

Suppose some one changes card on device, or reassigned to some one else!

↗ this has a time-out (order of minutes)

→ What if B is not trying to send it someone on same ethernet

ARP only handles local protocol!

So If C not on same MAC layer no reply of ARP.

So if A → C so router connect A → internet we should the routers MAC in ethernet frame as DST, then router forwards to its respective AS

So A has to intelligent because what to put as MAC, whether to do ARP or not!

① How does A know if DST IP belongs on own network or not

② If DST IP on other network, how to know R_{IP} / MAC

1) A_{IP} ; $a_1.a_2.a_3.a_4$

eg)

Subnet mask Ex : 255 : 255 . 255 . 0

all 1's

If A's Target IP & subnet mask = A_{IP} & MASK then same subnet so ARP

This DST IP belong to my address.

Date :

If not the we try to find R_{MAC}

→ Assume we know RIP

→ We ARP we target address of R_{IP} & we get MAC of R.

→ How to get IP address ?

i) Limited broadcast

DST IP = all 1'ST

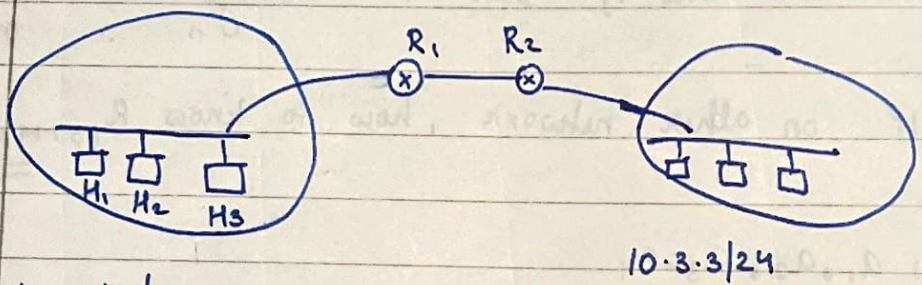
In this case all on same-subnet get message no other device because

DST IP & MASK = SUBNET IP

Now what is the issue, if we want to send broadcast out will this work? no !

Why → The L-3 router comes to know that this is a Limited Broadcast & stops it from leaving

ii) Directed Broadcast



10.1.1/24

subnet

Suppose we want to broadcast on 10.3.3/24 from 10.1.1/24.

→ If all IP dst bits = 1 then limited broadcast we send this back to subnet (I mean we do not send from $R_1 \rightarrow R_2$)

DHCP:-

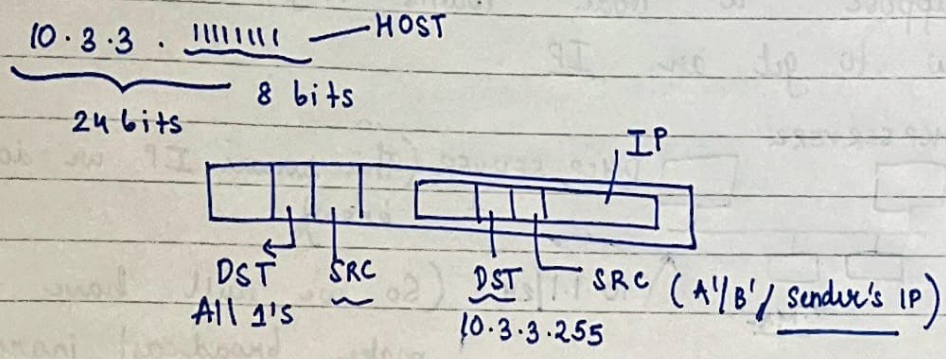
I want to my IP address. Defaults Routers IP addresses

Date :

So In order to do this we need to know the subnets IP or the LAN's subnets mask!

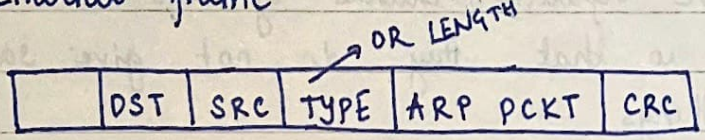
Because IP PCKT DST = subnets IP!

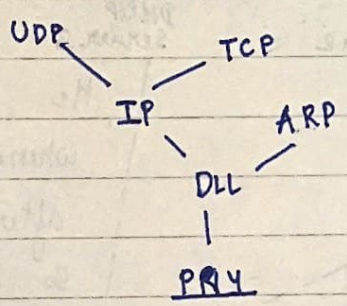
* all mac layer BITS = 1



To answer the previous Q we have DHCP Dynamic Host Configuration Protocol.

Ethernet frame





We need the Type in DLL frame because sometimes we may want to do ARP and IP.

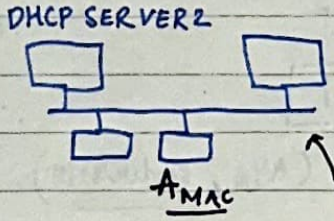
So thus we need a type/length:
if value of that chuck ≤1536 then this length if value > 1536 then this type field.
(eg : 0x0806) then ARP packet!

Date :

ARP: Know IP
Don't know Mac

DHCP: Node joins network
wants IP.

→ Suppose a Node wants to join a network how to get an IP



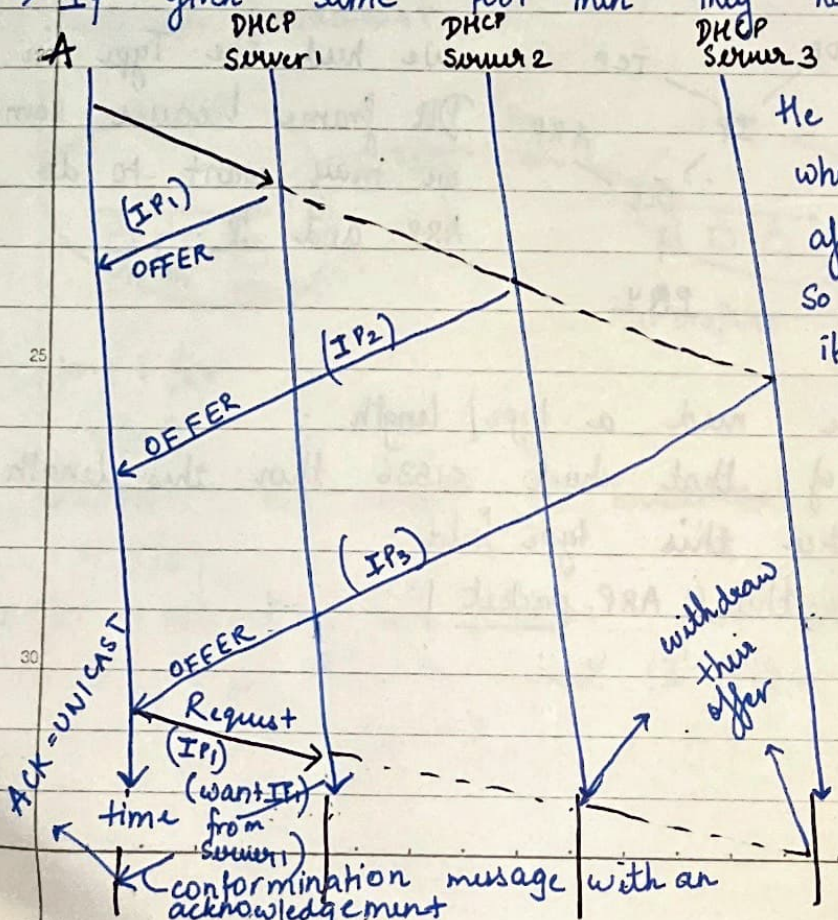
(This servers IP we do not know)

(So we will have to make broadcast inorder to get to DHCP server.)

Sync time in servers

→ We usually give separate chunks of IP's to these servers and so that they do not give same IP to diff things

→ If given same pool then they have to co-ordinate



He can accept whenever he wants after acknowledgement so after ΔT time accepts it.

DHCP (port) = 68 for client → This is for client

Date : _____

UDP TCP

IP

DLL

PHY

} iBGP sits on top of TCP/UDP even though it is a routing protocol, it is like an application layer

We want reply quickly so use UDP protocol

APP_1 ... APP_3

DHCP

UDP TCP

IP

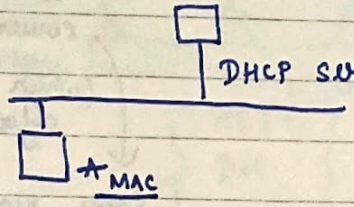
DLL

PHY

we get a packet

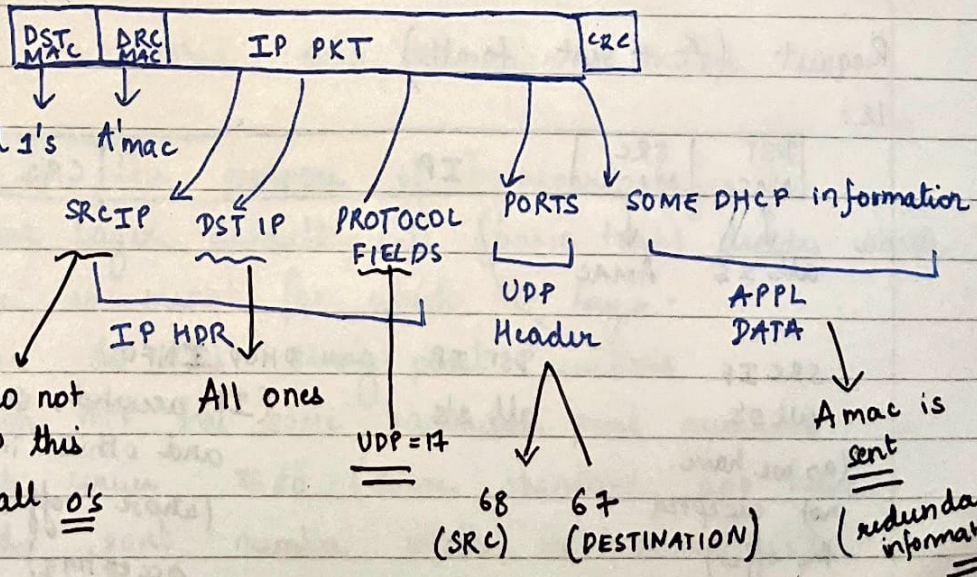
PKT

We have reached the IP field and IP decides next layer protocol after that which application it is meant, so we use PORT number to decide which application will get this pckt.



A sends discovery message

L_2

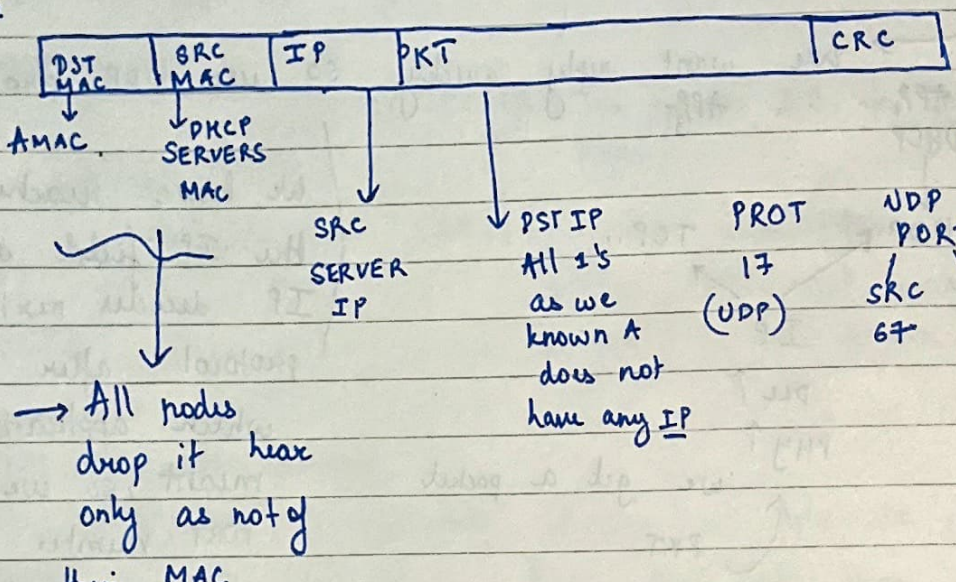


Date :

By time it reaches Appl layer all L-2, L-3 information is lost, so to know whose MAC want IP we should put this in DATA, so we do not lose this info!

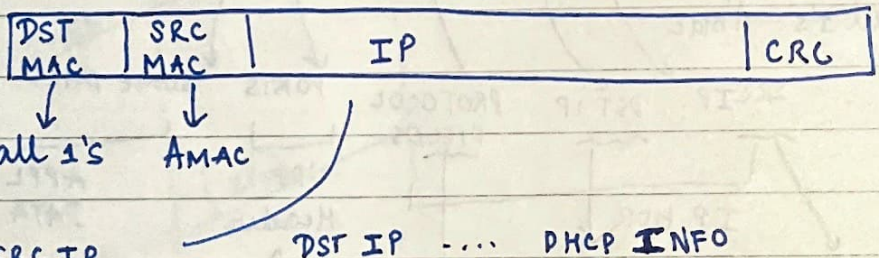
OFFER

L2

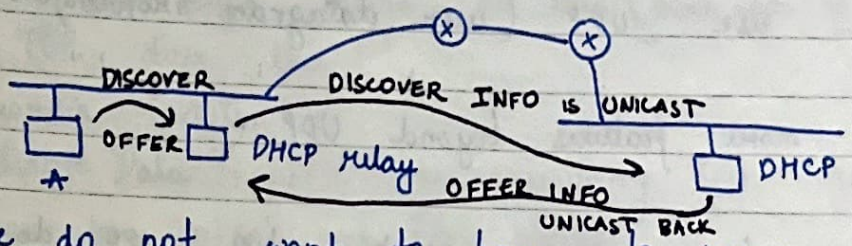


Request (A → sent to all)

L2:



Date :

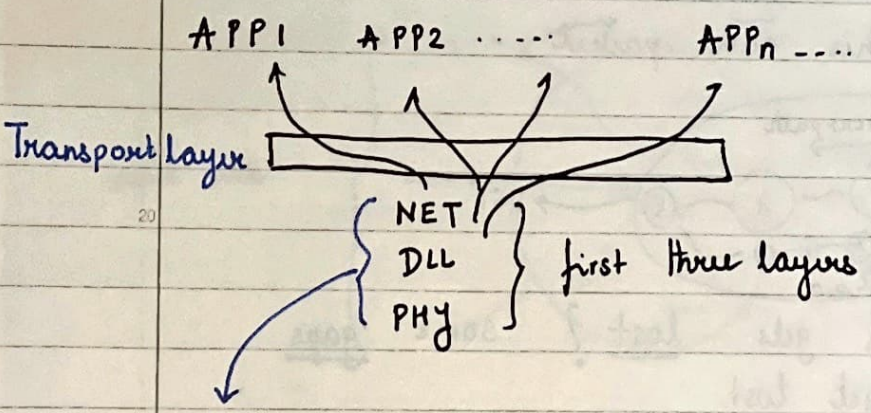


→ We do not want to have to many servers on LAN, so only few subnet have DHCP

→ So we have a DHCP relay which acts as a DHCP server, this forwards A's message to DHCP server as a unicast to Destined DHCP server and DHCP server replies via unicast

→ So DHCP RELAY has SRC IP has RTP (GIADDR) ↓ Gateway

Transport Layer (L-4)



} the original internet made APP layer very flexible (we can add as many apps as we want)

These are pushed into OS, network drivers

* What is the purpose of Transport layer?
Trasport layer demultiplexes (basic task) decides which message is meant for which APP layer.

→ So we do this using port numbers

→ So each APP has some particular port numbers.

→ " web server #80 (some standard app have reserved port number which we cannot use)

→ Port number in transport header

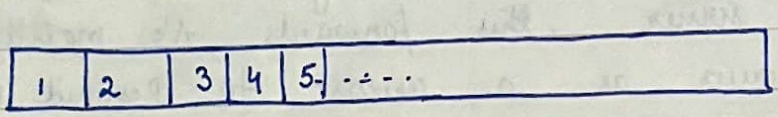
Date : _____

If we want a protocol doing only this we should use UDP (user datagram Protocol).

Some more features beyond UDP :-

So we have some work where we do file transfer

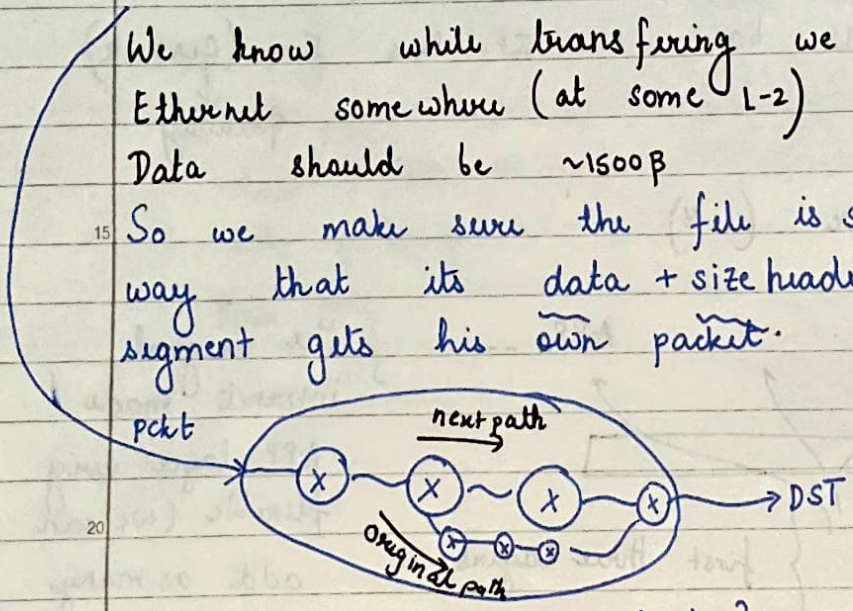
FILE TRANSFER



~ 1500B

We know while transfering we will encounter Ethernet somewhere (at some L-2) so max size of Data should be ~1500B

So we make sure the file is segmented in such a way that its data + size header ~1500B. So each segment gets his own packet.



What if packets gets lost ? some gaps

1) Packets could get lost

so 1 2 4 5 6 7 - - -

2) The weights of links could be changed, some link fails so a new path is taken

so packets could be faster on new one x there could reordering

1 2, 4, 5, 7, 8, 6, 9, 10

Even though Queue's implement FIFO, still reordering could happen.

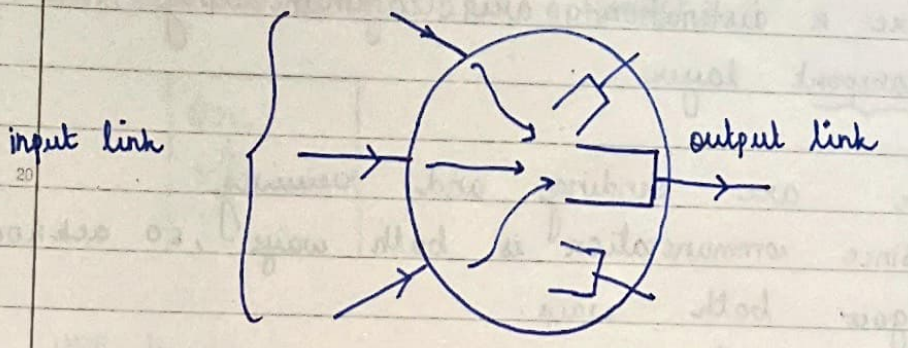
Date : _____

Now files look bull shit
So transport layer should do this (take care of this).
So TCP does this
: Reorder Packets
: Reliable Data Transfer : retransmitting lost segements
: Congestion Control and Flow control
So to this the source will have to do retransmission)

→ So TCP will not allow any other packet to be received after 12, untill 3 comes, so source retransmits (3) segment.

- So now reciever has to do some reordering.
- Routers are "Dumb" they cannot do this!

So TCP has to infer packet loss



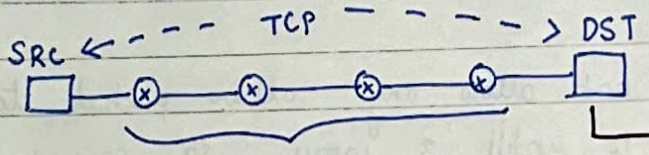
So if input rate > output rate
Queues start filling up and packets are buffered.
- So Queues have drop tail mechanism.
- If this happen congestion occurs:-
  → Queues full, pckts are dropped.
- So if apps are using UDP a lot they just keep sending data & congestion prolongs for a bit.

Date : _____

So we need TCP to do congestion control.
" reducing segment sending rate "

So if many are using TCP so many input wires reduce input rate, so Queue starts emptying

FLOW CONTROL



So if congestion in router along SRC - DST we call it congest^{n} control

* if congestion occurs here handling it here is called flow control.

Why does flow control differ from congestion control.
→ The src & distination are communicating directly at Transport layer

→ So we are sending and reciving
Now since communication is both ways, so acknowledge
- ment goes both ways.
eg Google search :-
(search →)
(← webpage)

→ So the reciever has a TCP buffer and it could full, this is because reciver could take time to process recived data.
→ So reciver can directly say without inferring (some mechanism of congestion control) that there is congestion.
→ So can communicate this to SRC easily !

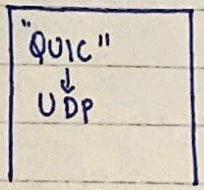
Date : _____

When to use UDP?

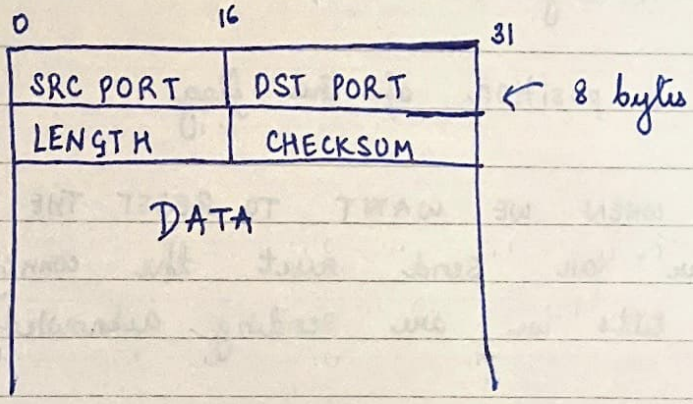
- DHCP servers use UDP and avoid TCP
- When the latency is very low on the path so
  321 ⟶ 123
  ↳ kind of all will same transport time
- When data is so small that it fits in one packet (no reordering needed) - (still message loss?)

If high latency we can use TCP!
Because TCP somehow does not allow wrong order and if we sent 3 2 1 ⟶ 1 — 3 (second) 2nd has not arrived, (got lost & buffered) it will only send 1 and withhold queue from 2-3-4-5-6....
Too much buffer!

Google Chrome uses something called "QUIC"
It sits on top of UDP optimization!

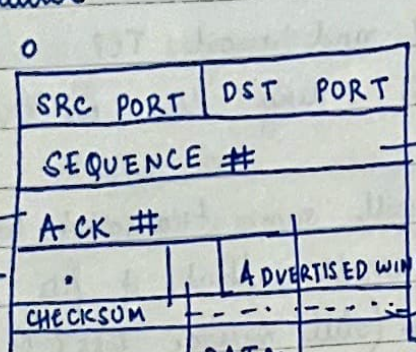


UDP Header



Date : _____

TCP Header



20 bytes of our head so each size of data.

→ what byte # we are expecting next
We have got everything till prev byte)

SEQUENCE NO
BYTE NO of 1ST byte in segment

FLAGS
↳ SYN | FIN | RESET | PUSH | URGENT | ACK

Initially we start a connection we send a syn pckt
fin packet (end of connection)

→ So 0/1 in position of this flag.

→ RESET IS WHEN WE WANT TO RESET THE CONNECTION
→ So Attacker can send reset the connection !
→ ACK bit tells we are sending acknowledgement.

Date : _____

ADVERTISEMENT WINDOW IS USED for telling how much of TCP buffer is full/empty so if too less space avail the server receiving can slow down sending.

Appl (port) → Appl (port)

UDP | TCP → We we send data as a stream of parts.

↓ Single Datagram (DATAGRAM SOCKETS)

↓ UDP: Packet recieved ← give immediately

(STREAM SOCKETS)

→ TCP splits message in segments we want to recieve in order at receiver

→ Here we need to remember a lot of state information

→ So for TCP we setup a connection & we will use this connection to monitor data transfer between Sender ⟷ Receiver (TCP)

CONNECTION ::

* This connection is responsible for Congestion control, flow control

(Listens at port (eg. 80))

TCP connection Establishment

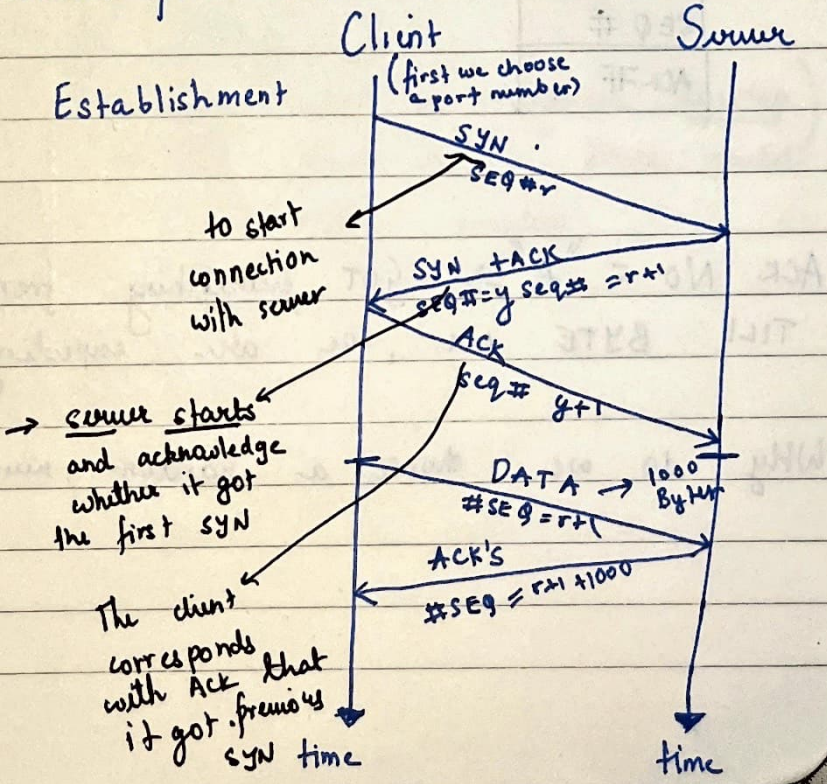
3-WAY HANDSHAKE

→ Only done for TCP

→ FLAG field of TCP Header

SYN | FIN | ACK

we set these bits



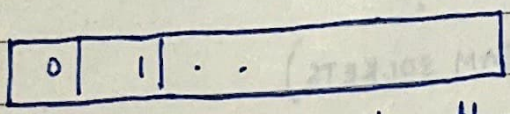
How is TCP designed

Data

→ So we allocate numbers to each number

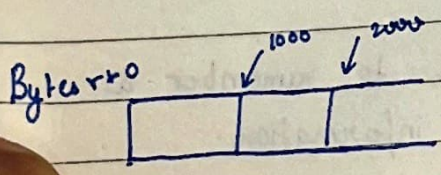
→ First issue these segments are not of same size

Data



If we number these segments then if they are not same then kya?

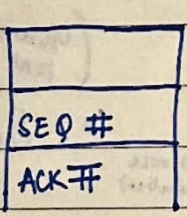
Instead we number them using bytes



use to send the starting number of byte in message and then using data length field we estimate length of packet

also we do not use 0, we a random number

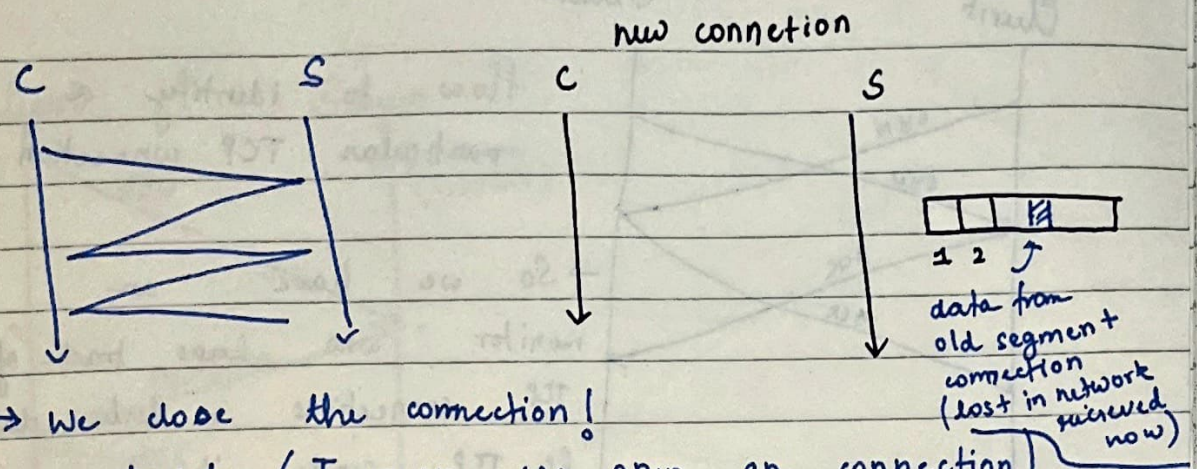
∴ r = random number



ACK NO = "A" => GOT everything from start (SYN SEQ #) TILL BYTE A-1, we are expecting byte A next :-

Why - do we choose a random number!

If segments number from 0

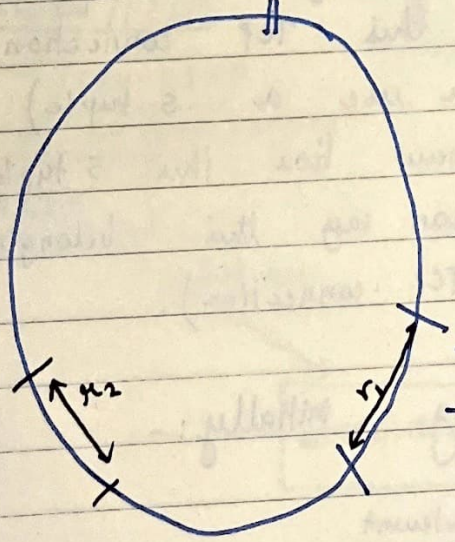


→ We close the connection!

Now we already (I mean we open an connection) immediately to server! (client port = same, server port = same)

So these could be so close in time, some data by first old connection (was stuck in some router) and we have started recieving info from new packets. So if it recieves it we put in some place with its new data. So to avoid this we used sequence number

SEQ # : 32 bit



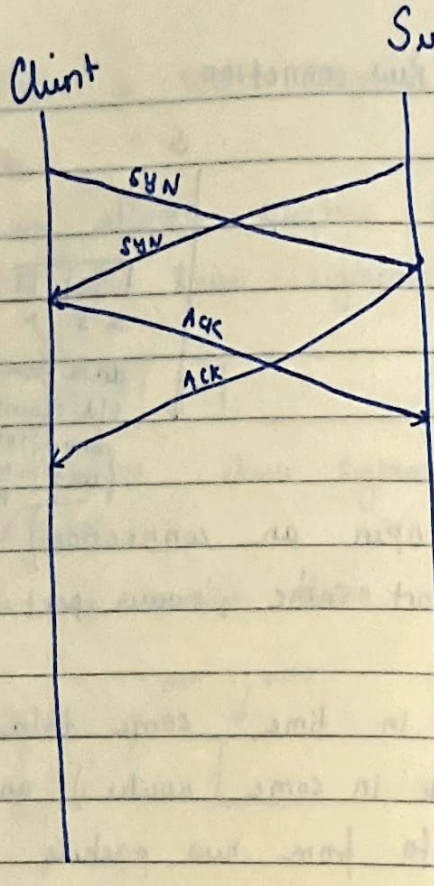
→ so for different connections we try to have different segment range for

→ (we want to minimize overlap)

→ If files large there could be some overlap.

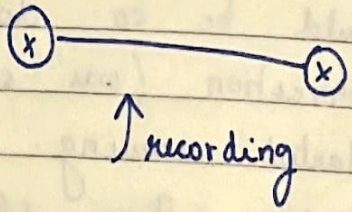
4-WAY hand shake :-

NXT PG



How to identify a particular TCP connection

→ So we have an monitor who keeps track of TCP connections & dubs data bf TCP sends it to respective places

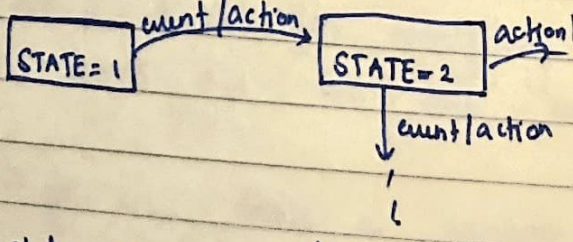


So

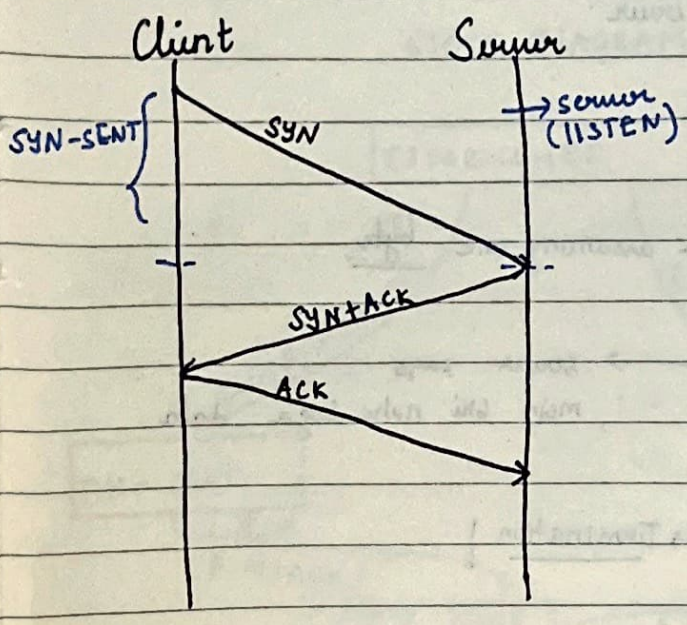
IP : SRC IP, DST IP
PORTS : SRC PORT, DST PORT
PROTOCOL : (INDICATES TCP) (IP HDR)

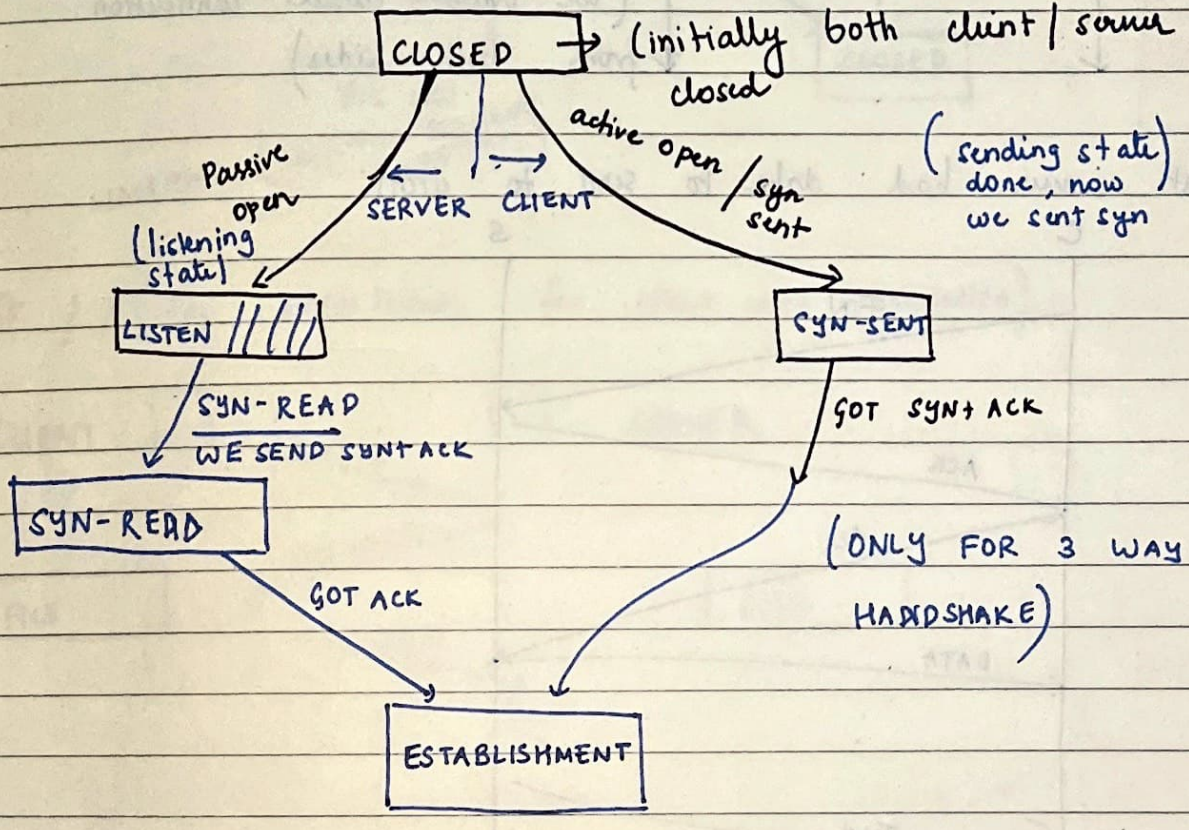
} If we want to indutify if this belongs to this TCP connection :-
(So we use a 5 tuple)
(so whoever has this 5 tuple we can say this belonging to this TCP connection).

So server is always listening initially :-
So state diagrams helps us :-



So state diagram for 3-WAY HANDSHAKE

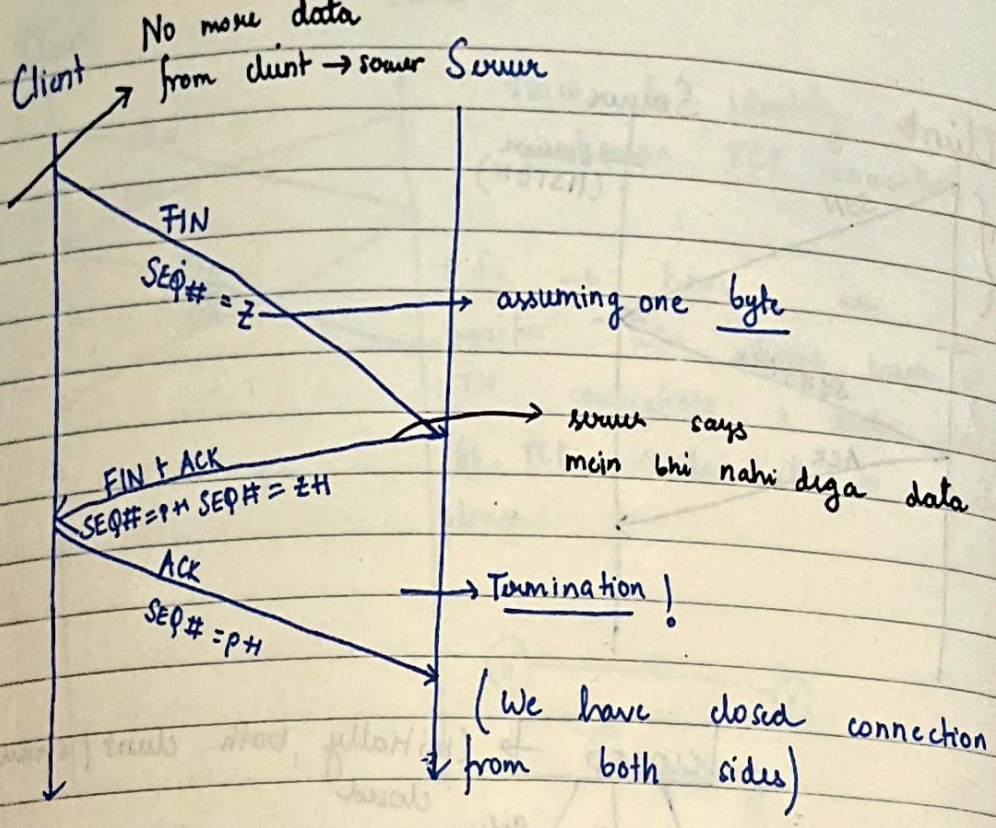




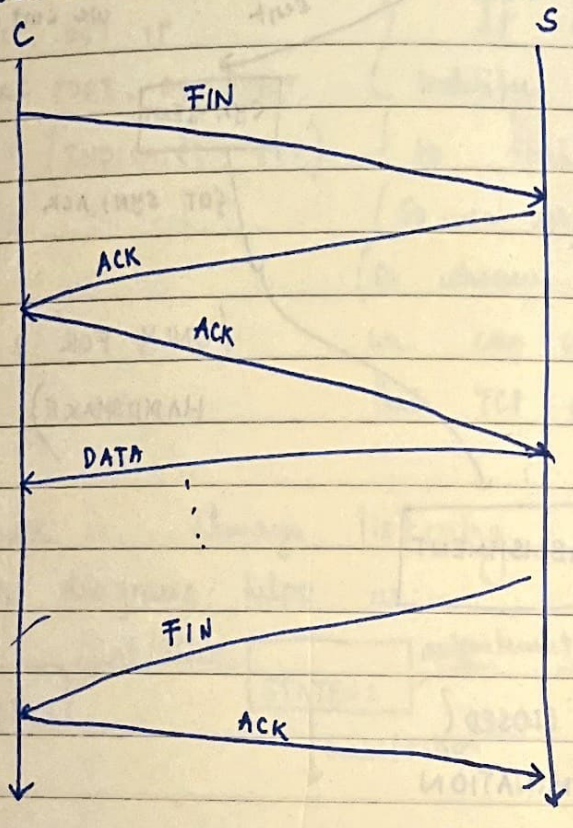
HOW CONNECTIONS IS CLOSED?

CONNECTION TERMINATION

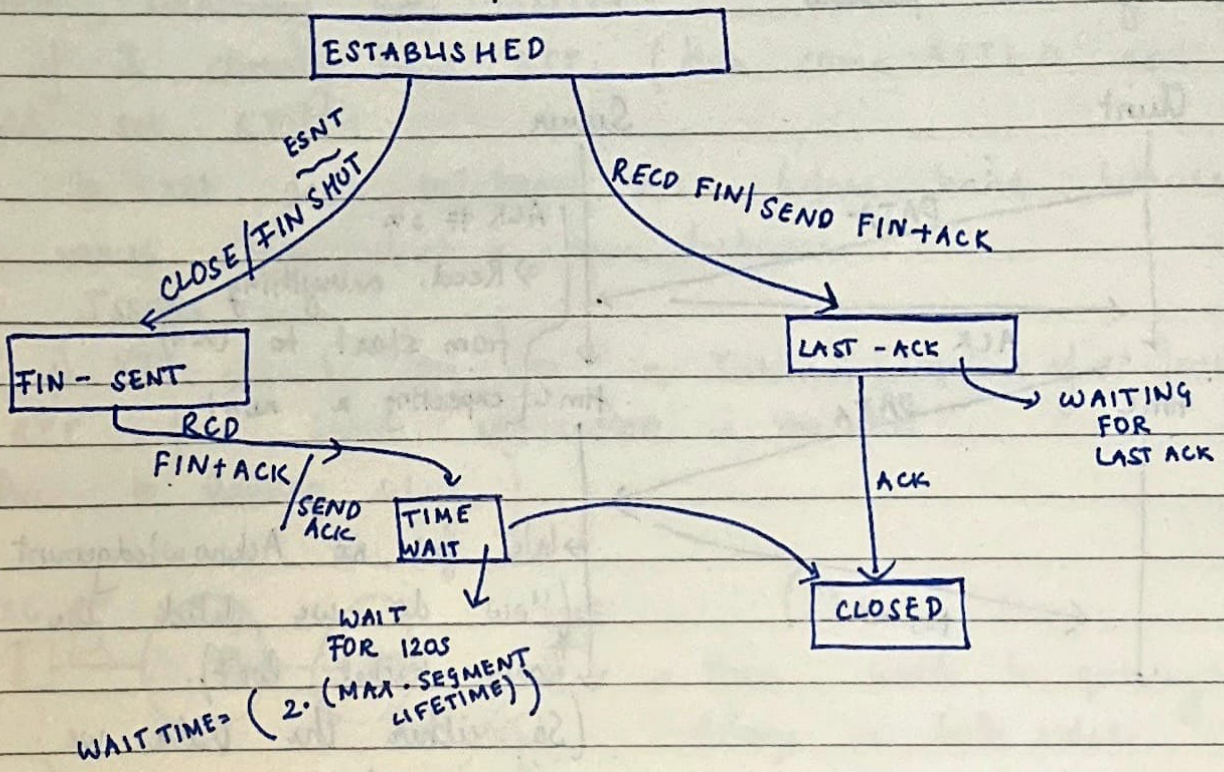
NXT PG



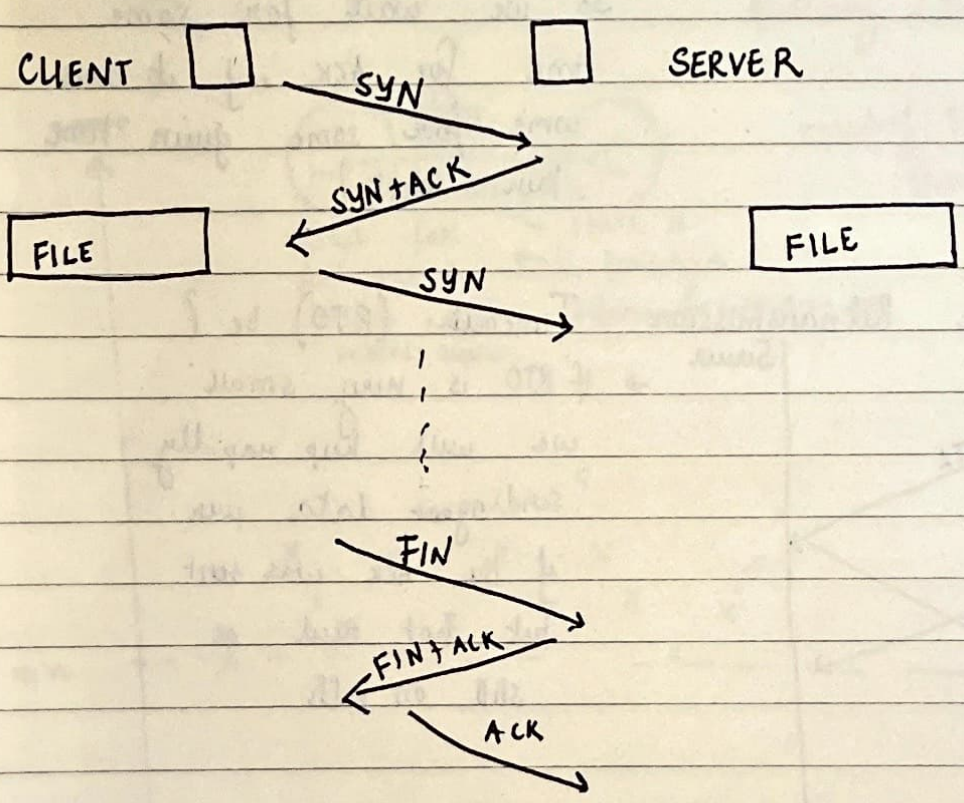
What server had data to send to client.



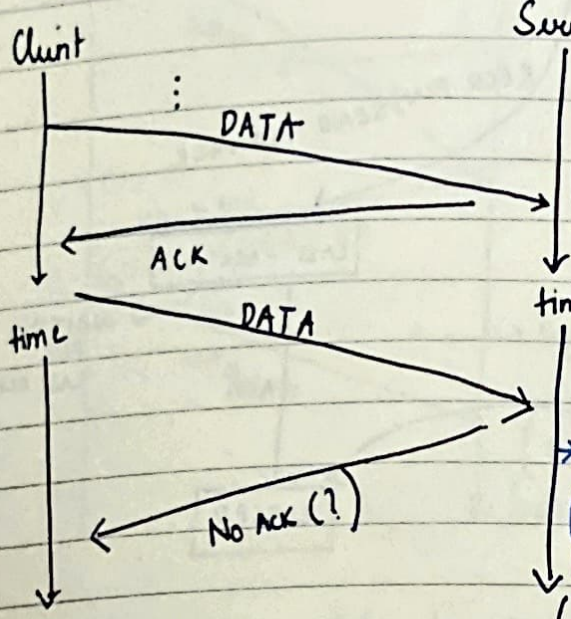
STATE DIAGRAM



If ? (See video lecture for other case - completion)



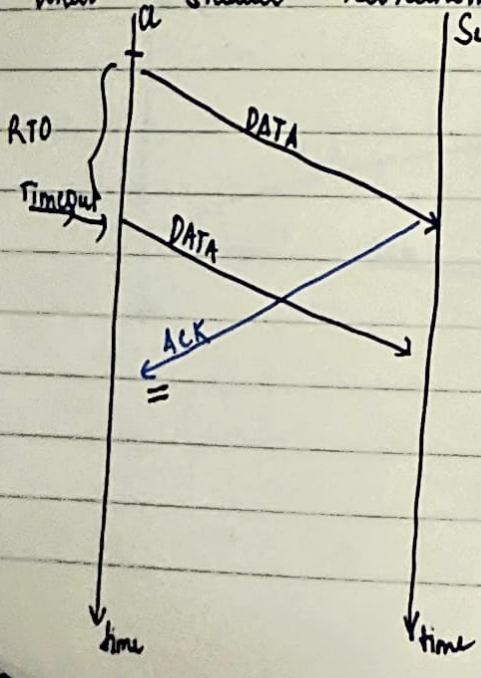
① TCP has to do retransmission of lost packets



ACK # = x
⇒ Recd. everything from start to (x-1) expecting x next

→ We got no Acknowledgment (How do we detect we have packet lost).
(So either the Data we sent is lost)
OR the ACK is lost.
So we wait for some time for ACK, if it does come for some given time then what!

What should Retransmission Timeout (RTO) be?



→ If RTO is very small we will keep rapidly sending data even if the ACK was sent but not recd as still on path

→ If RTO is very large, we will wait for a lot of time, unnecessary time wasted.

So if I already knew RTT (then some RTT + Δ we could set RTO).

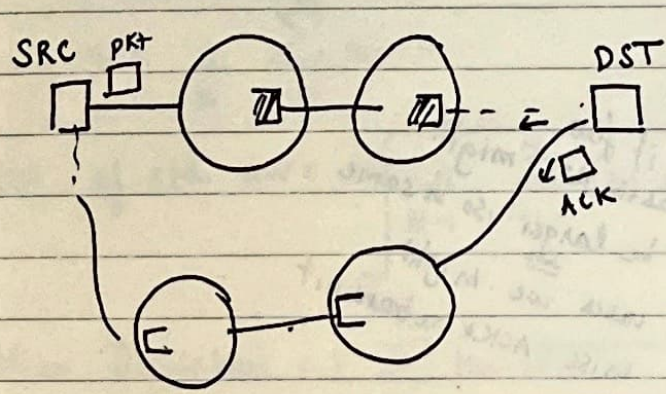
It is not good to know RTT before hand because it varies for client & server distance.

ISSUES

So, 1) RTT can be ms - sec in Internet (location of SRC/DST)

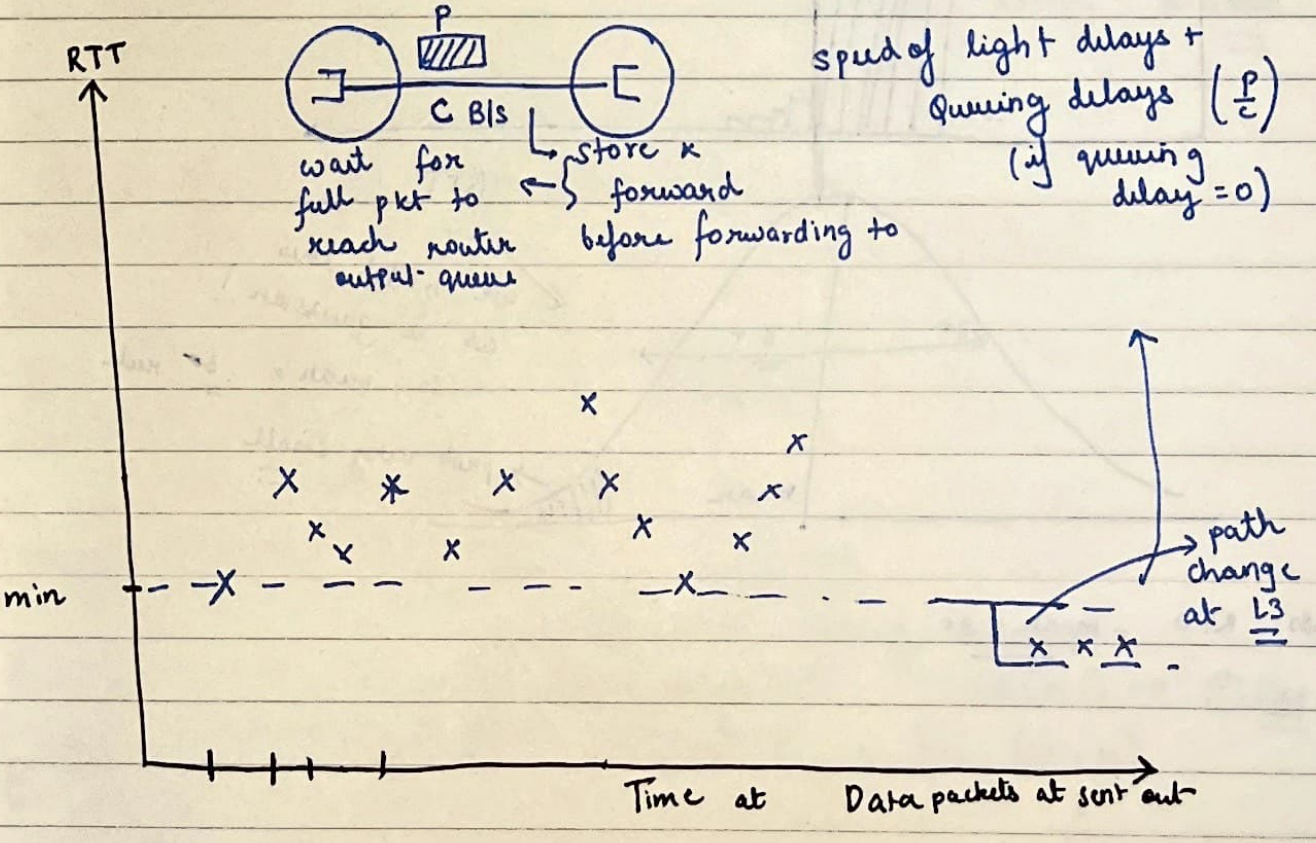
2) RTT for same connection is variable.

(Due to Queuing delay)



→ there could to queuing delays in both sided transmition

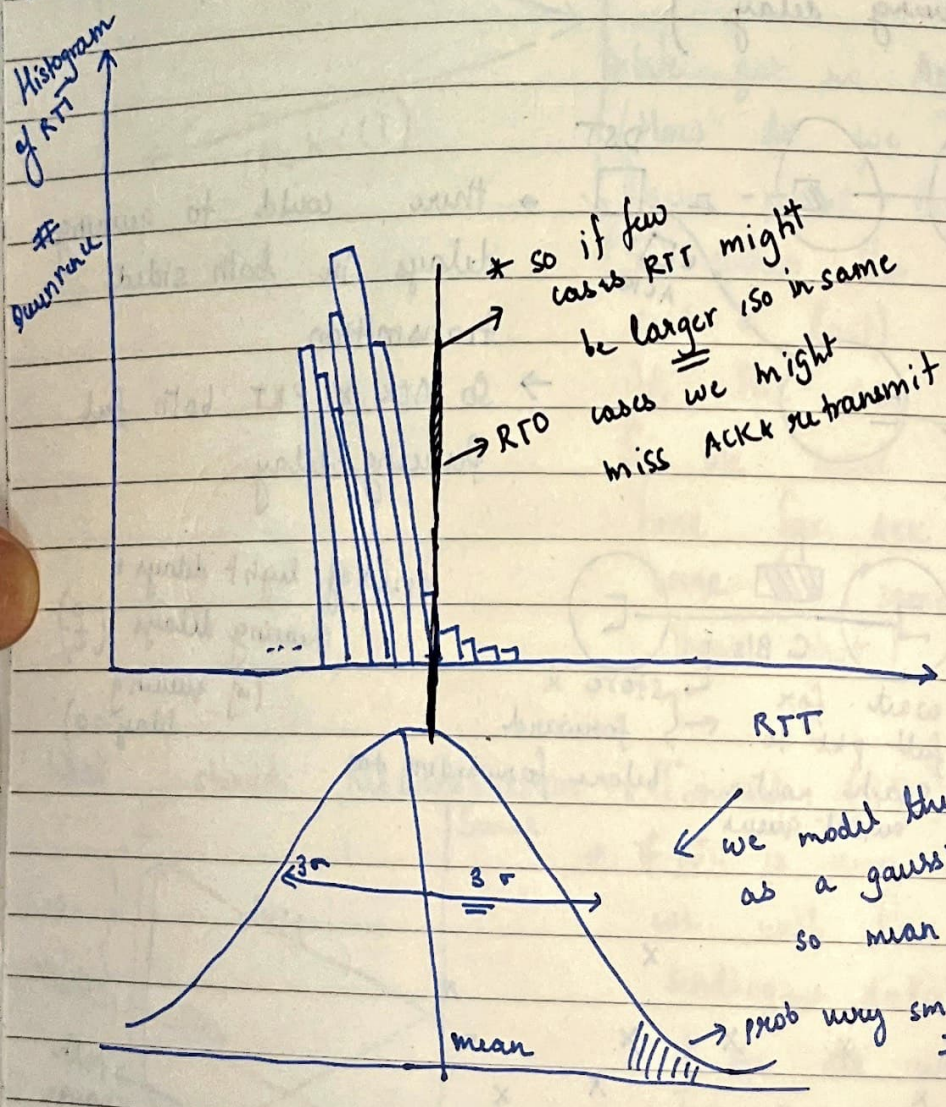
→ So ACK & PKT both fal Queuing delay.



corrupted or not

So in store & forward we use chuck sum to check whether packet is currupted or not
So CUT-THROUGH delay we immediately throug it in output queue the moment we start getting so we can reduce transmission delay.
We won't be able to check whether packets we have received is correct or corrupted, but in CUT-THROUGH we have speed.

So how should I choose RTO :-
So people plotted a histogram of RTT } Because we do not queue size, link speeds, buffer capacity at queues .....



so RTO = mean + 3σ

Idea: Measure RTOT over time, set RTO = mean + const * σ

→ Intuition:.

→ Also I want to look back in time quite less. So that I can avoid errors due to path changes, suddenly queues filling up, (conjestion)..

→ so limit look back period:-

∴ Random Numbers : $x_1, x_2, \ldots, x_N$

$$M = \frac{1}{N} \sum_{i=1}^{N} x_i$$

$$\text{Est of std dev} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - M)^2}$$

this is very difficult to do, since 1000's of packets are coming every second, how do square, square root, so much time wasted.

$$\text{Mean Deviation} = \frac{1}{N} \sum_{i=1}^{N} |x_i - M|$$

squaring & sqrt rooting avoided =

ALGO USED FOR TCP (for RTO)

RFC (request for comments)

Sample RTT ;
→ lastest RTT estimate ;
→ current

current estimate of mean actin RTT

→ exponential moving average.

EstimRTT = $(1-\alpha)$ EstimRTT + α SampleRTT $\alpha \in (0,1)$ → ② step

① Difference = Sample RTT − EstimRTT = like $(x_i - M)$

→ first step

Deviation = $(1-\beta)$ deviation + β |difference|

Timeout = $\mu \times$ EstimRTT + $\phi \times$ Deviation

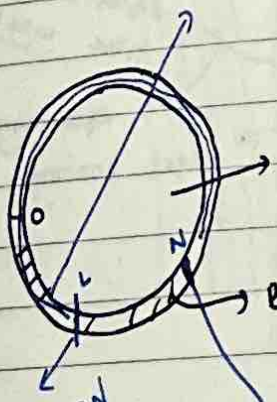
$\alpha = \frac{1}{8}$ $\beta = \frac{1}{4}$ } Default

TCP Vegas:- (Some Class)

CONGESTION & FLOW CONTROL

→ (Network does not touch the packet just pass it to DST)

What if congestion here?
(Too much load



TCP BUFFER at DST

BUFFER FILLED TILL HERE

APPLICATION HAS READ ONLY TILL HERE
(last read(L)
(By Appl)

Next expected Byte (N)

So the major arc of LN is avaiable as some data can be overwritten as read and processed.

So if APPL is slow buffer could easily get filled.
So Field in TCP hdr = Adv. Window = Free Buffer available

So if this buffer space is reducing at a high rate we want to communicate to APPL at src to slow down.

Buffer size = M

So remaing space = M - (N-L-1)
↳ (Because N is next expected byte)

so if N<L
: (L-N+1)

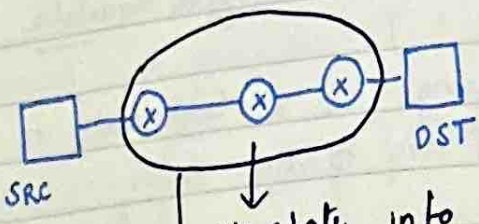
Because N has wrapped around.

Adv. Window = (M - (N-L-1)) mod M.

SRC using window how much max unacknowledged bytes I could I have send out.
↳ Definition of window

WINDOW = min (congestion widdow, Advertised Window)
↳ Method to control Data rate.
↳ Depends on congestion window!

TCP CONGESTION CONTROL

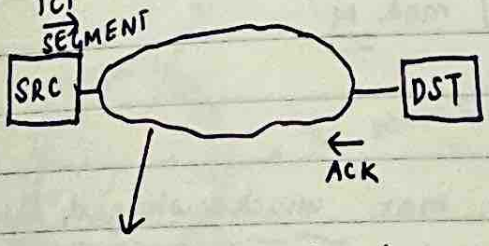


ARPANET: No state info about connection (No Appl. or Transport state stored)

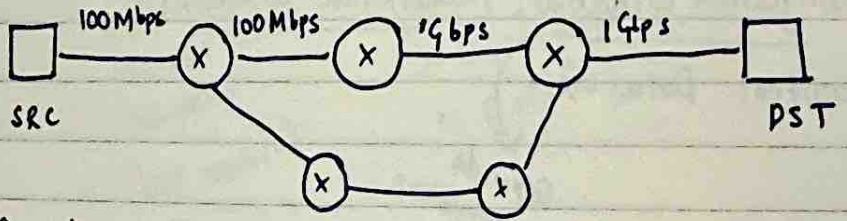
only pkt forwarding.

If network goes down, since we had no state, so no loss of huge amount of information.

So they put the complexity at end
So end nodes handle packet loss (packet forwarding) mech.
So for SRC & DST Network is a black box

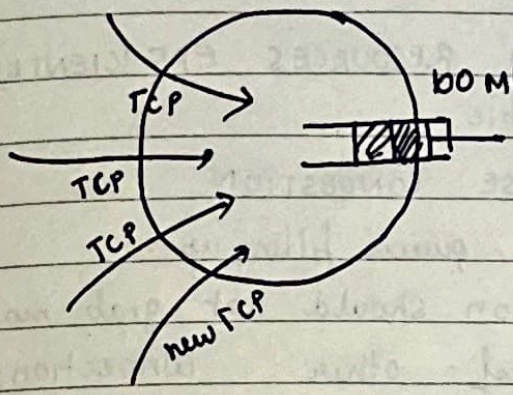


So these do not know Data rates in network but somehow they have to estimate this.



So they do not know bottle neck link speed & they do not know at what rate data should be sent.

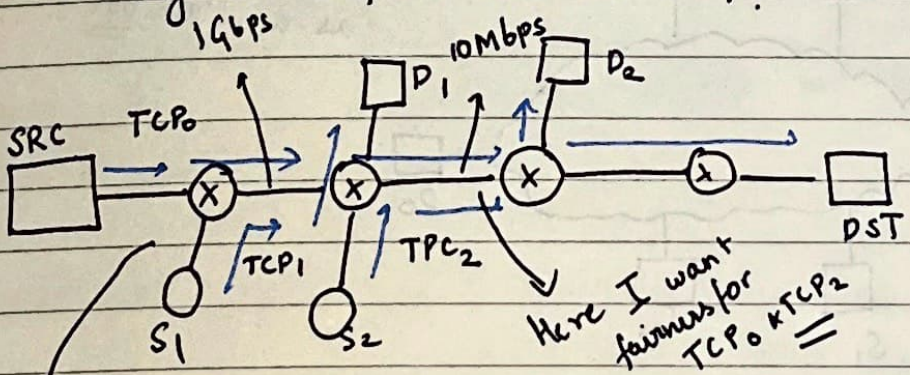
- End-Hosts Running TCP
  - DON'T KNOW LINK SPEEDS
  - DON'T KNOW UTILIZATIONS



So many packets of TCP comming from different source
Since we are fair so each gets 33 Mbps
But even if we know utilization & speeds, we do not know how many TCP connections are sharing the same path.

Due to this if a new TCP comes they how will SRC know?

Also knowing LINK speed does not help!



So I do not know how many connections I share the path with, I can say about particular link.

TCP_0 → Data Rate : 10 Mbps
TCP_1 → " " = 990"

TCP_1 is not hurting TCP_0 because TCP_0 cannot get more than 10 Mbps since 10 Mbps is bottle neck

So TCP_0 = 5 Mbps
TCP_1 = 995 "
TCP_2 = 5 "

so always there not fair distribution

So Can these source and DST talk?

So Congestion Control issues

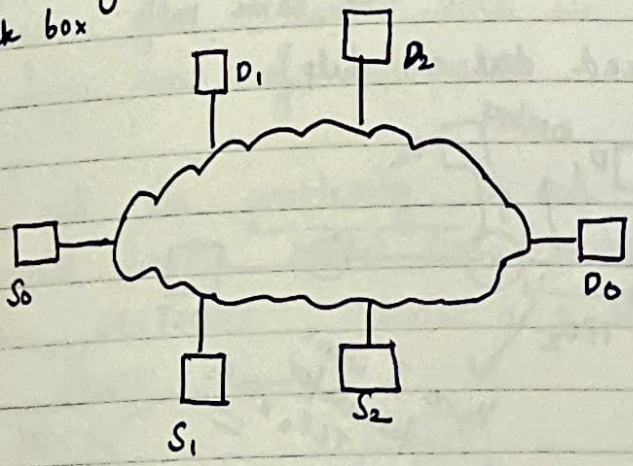
TCP Connection

. WANTS TO USE BANDWIDTH RESOURCES EFFICIENTLY
→ use as much as possible

• DO NOT WANT TO CAUSE CONGESTION
→ but do cause packet loss, queues filling up.

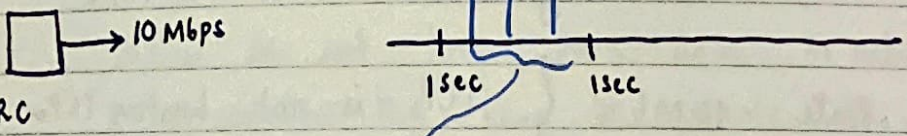
• FAIRNESS - ONE TCP connection should not grab most of the bandwidth at expense of other connections.

We want to develop an Algo that does not involve routers talking * SRC_1 x SRC_2 talking, so each sec network as a Black box



All set network as a black box !

Q) How to set data rate of TCP

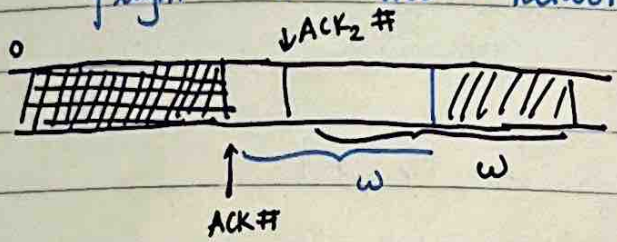


Suppose I am sending 10000 bits packets, so I can send 10^3 pkts as $10^3 \times 10^4 = 10$ Mbps

So we can send data for 1/2 sec or some part and then stay silent. So basically I burst into network.
But what if 1GBs was speed then I would burst into network by sending 10^6 packets, so we can reduce window size to 10 ms and then we can stay silent, we have counter that maintains this.

WINDOW BASED DATA RATE FILE

W: max amount of unacked data in flight in the network.

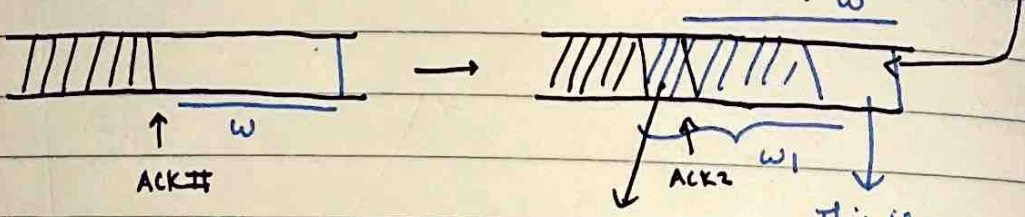


(the byte no server expecting

So since window size W, we can send W size of data from Ack no

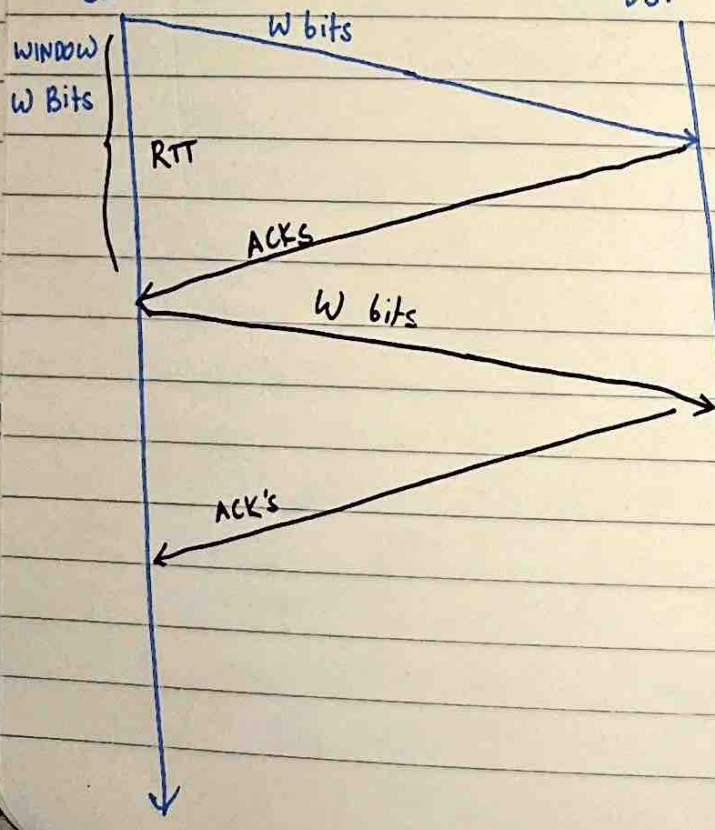
But we get a new Ack

So I am allowed to the new data that is not in network after sliding window to right!



Segment size

This is extra data we can send

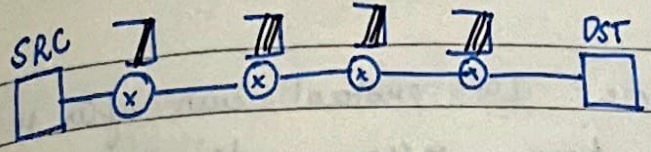


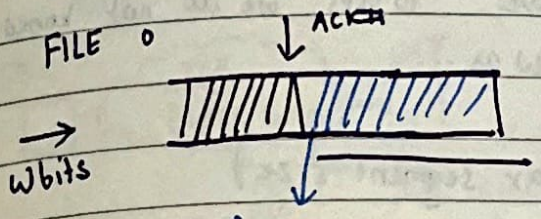
DATA RATE = $\frac{W}{RTT}$ bits/sec

We increase or decrease W to vary our data rate!

Clever to adjust data rate without any counter

If we feel congestion we will decrease W, if we feel empty network we send our more by ↑W.





These W are in flight

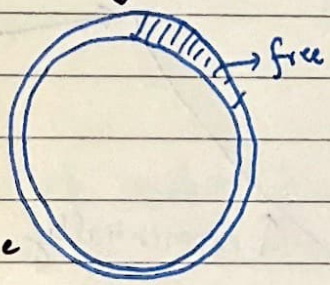
→ So we can see that at max we would have occupied W bits in queues all over the path, so we some how heuristically control congestion !

→ So may even have reached the DST, so would be queues. but worst case could be all/some in queues across path !

$W = \min$ (Cong. Window, Adv. Window)

↓ (CW)

CONGESTION WINDOW

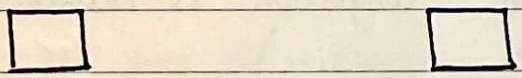


So for next few classes we will assume Adv. window ≥ Congestion window.

We $w = cw$ thus we do this !

SRC DST

INITIAL VALUE OF $w = ??$



Suppose $W = 10^3$ pkts $\rightarrow 10^7$ bits

1pkt $= 10^4$ bits

RTT: 1 ms =

So what is my data rate $= \frac{W}{RTT} = \frac{10^7}{10^{-3}} =$ 10 Gbps

very high!

We want to use the same TCP protocol even after 10 years, so we can all have 10 Gbps, we do not know anything, what should be set w as.

IDEA: SET $w = 1$ MSS (max segment size)

1 MSS ~ 10^4 bits RFC 5681

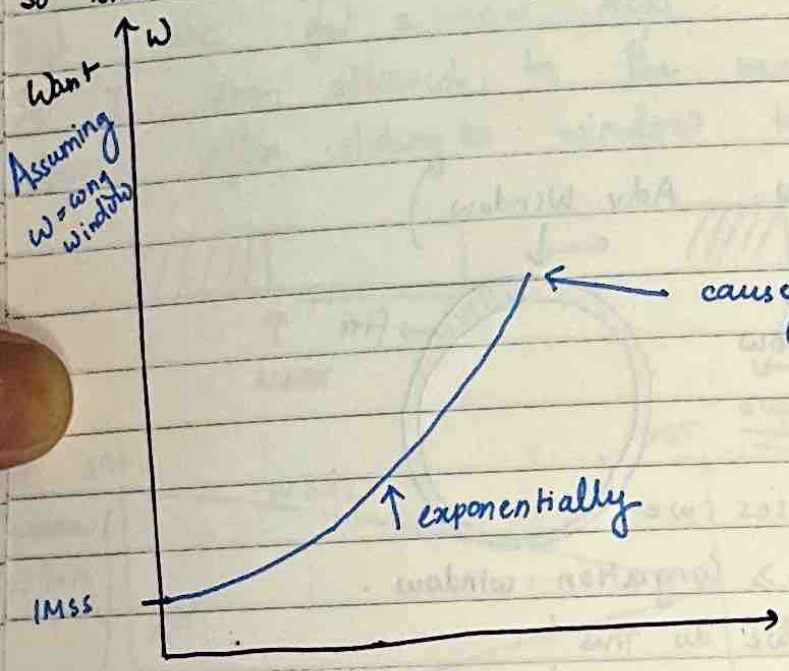
less than 1 MTU size of ethernet

so we send only one packet

So we are not using bandwidth efficiently

So this is called slow start: start with $w = 1$ MSS

(we want increase w rapidly to get appropiate



We start aggressively by increases no of MSS exponentially and then when congestion is caused we want (slow down) and (lowes wsize)

These two questions ↓ are unanswerde

TCP Tahoe

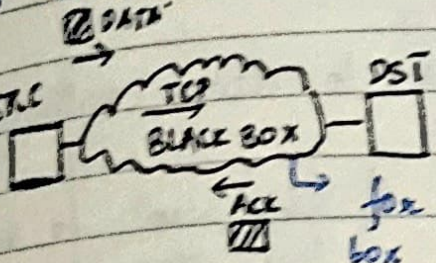
TCP Reno

TCP vegas

TCP AFRICA

TCP CUBIC

TCP Congestion Control



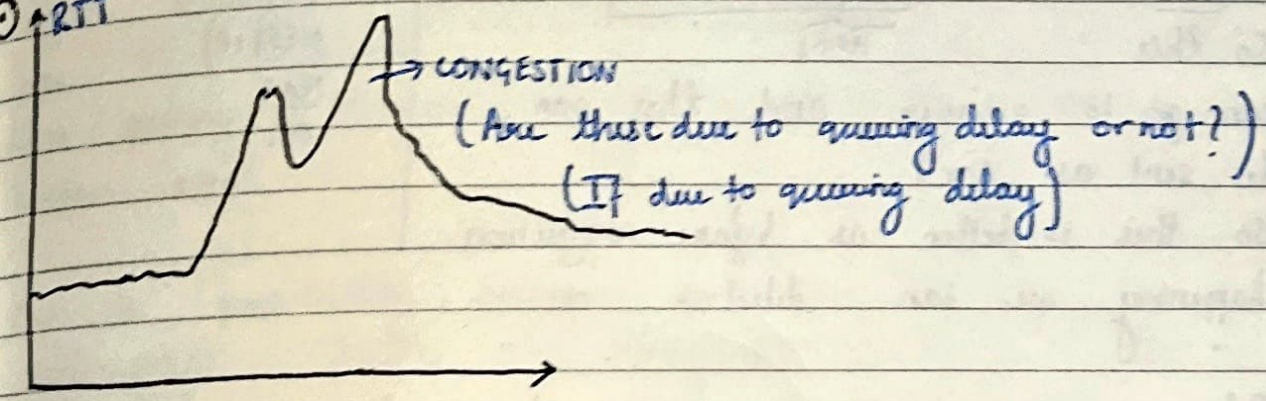
for source + dest, network is a black box

SIGNALS OF CONGESTION?

How do we figure congestion is there?

RTT may increase (due to increase in queuing delay)

① RTT



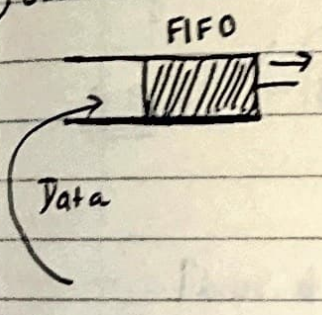
(Are these due to queuing delay or not?)

(If due to queuing delay)

② Other method is to look at packet loss!

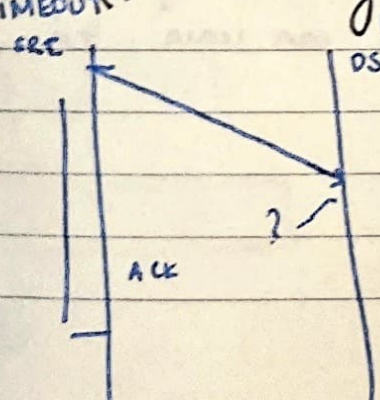
(If queue full we have packet drops)

(So we can somehow use this to figure out packet loss).



Q): How do we infer pkt drop has occurred!

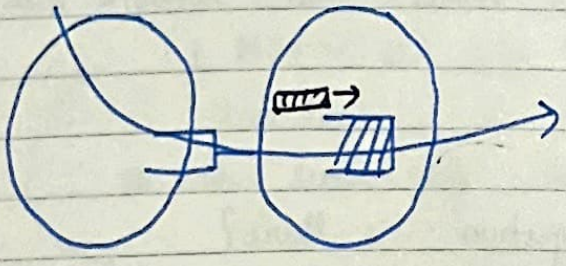
(We use timer (for receiving ACK) so we infer pkt drop!



(We have to figure how to detect early packet loss because timeout could be very large)

③ ECN: Explicit Congestion Notification!

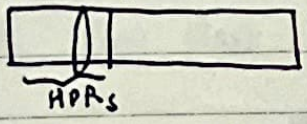
The routers should give feedback to source without changing protocol



(We want to keep fact intact that TCP does not happen at router level but we want some feedback)

TCP
NET (IP)
MAC.
DLL

So we set some bits in TCP header!



So this can go to reciever and this can be sent as ACK!

So this is better as before congestion happening we can detect.

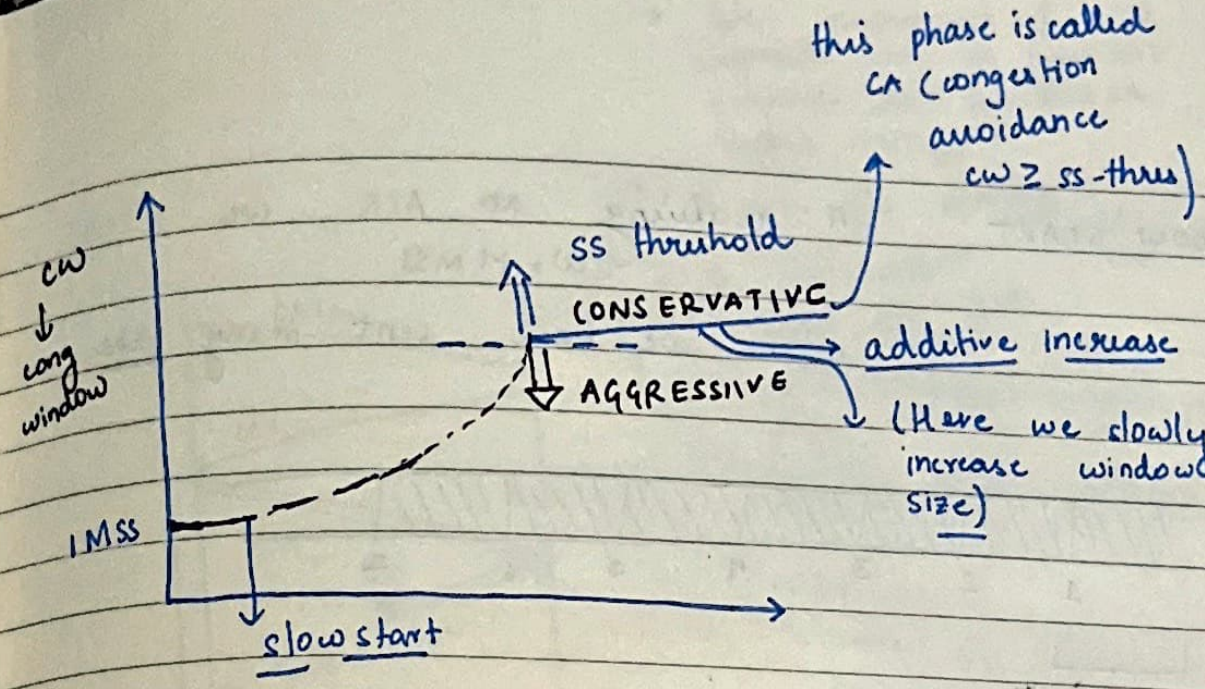
Before pkt is put in queue it modifies the TCP header bits, does not completely participate (some what of a hack)

ISSUE: Range of band width present (BW) is High

10 Kbps — 10 Gbps

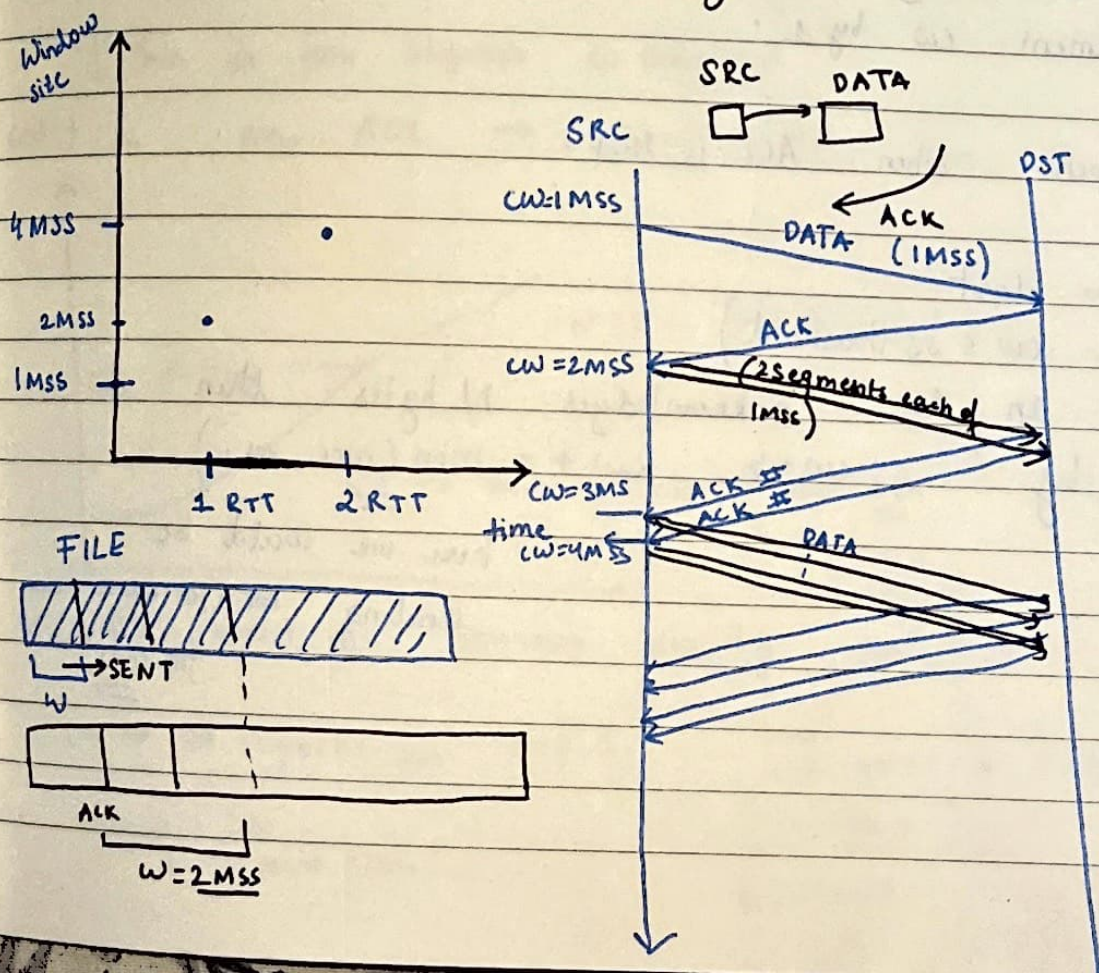
How should TCP figure out at what rate to send?

So we looked a slow start, we look CW and start it with smallest size and rapidly ↑ CW size till we hit danger mark, we can learn this overtime!



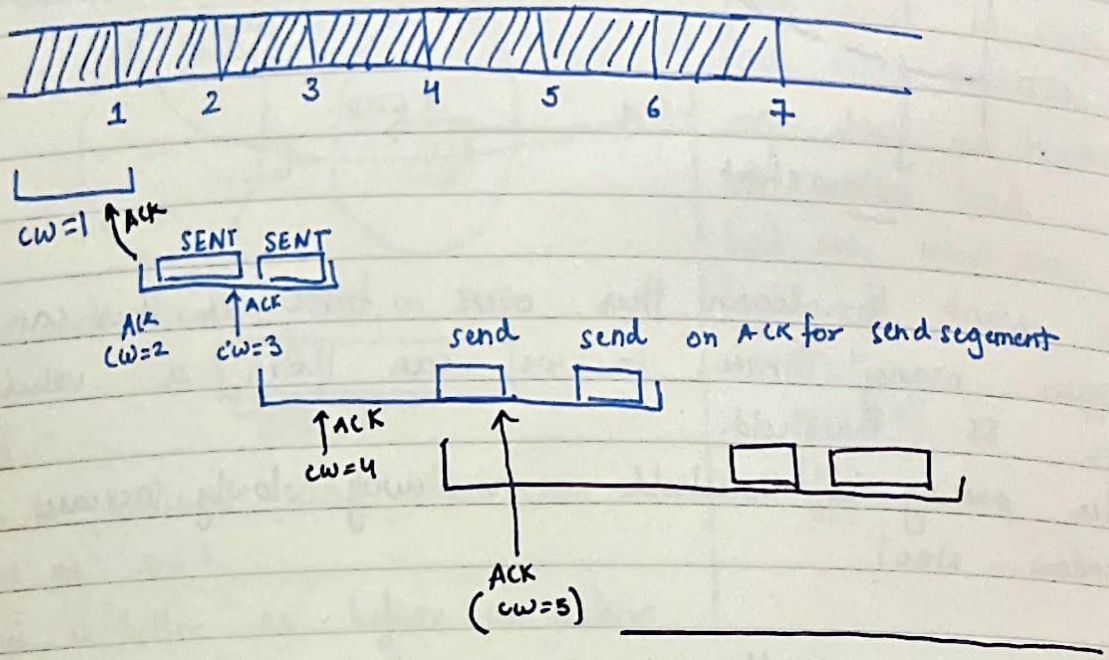
We want to learn this over time, as this can happen many times to we can learn a value for ss threshold
After passing ss threshold we very slowly increase window size!

How to practically increase cw!
SLOW-START Double cw every RTT.



(IN SLOW START : On receiving an ACK we
cw = +1 MSS

when we have sent many pkts



So everytime I get ACK I shift a window by right and increment cw by 1.

What happens when ACK is lost)

[RFC 5681]

In slow start
(ie cw ≤ ssthreshold)

then if an ACK anknowledges N bytes then cw increases by cw => cw += min(mss, N)

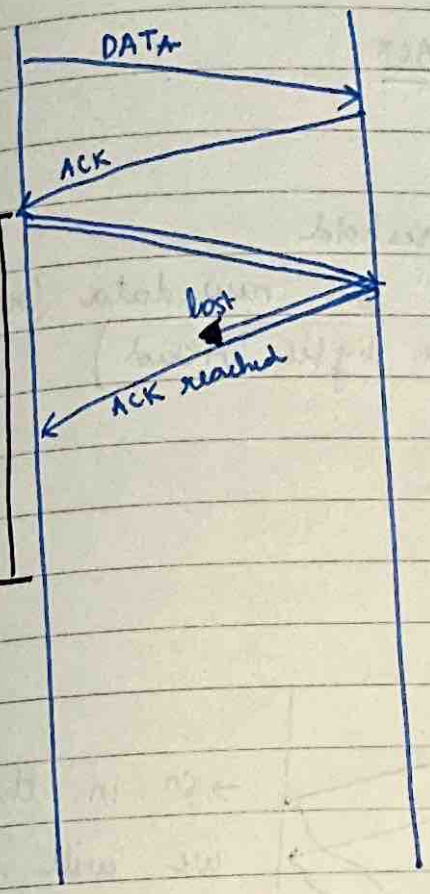
here we could be sending less (for cases when file ends)

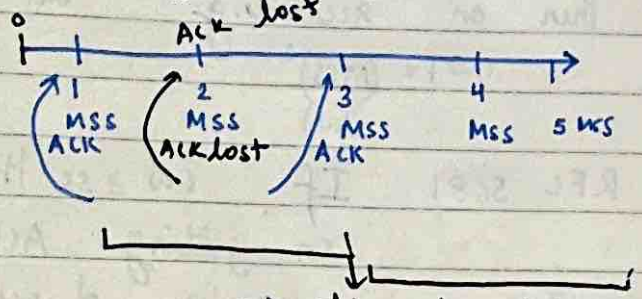
→ We retransmit if RTO expired and we have not received ACK for this or higher number!

→ So we got ACK = 4

So we assume at DS we have got 2 but ACK lost

CASE 1





→ This acknowledges for 2 bytes from 1-2 & 3 but for second lost ack

→ (so we cannot increase by 2)

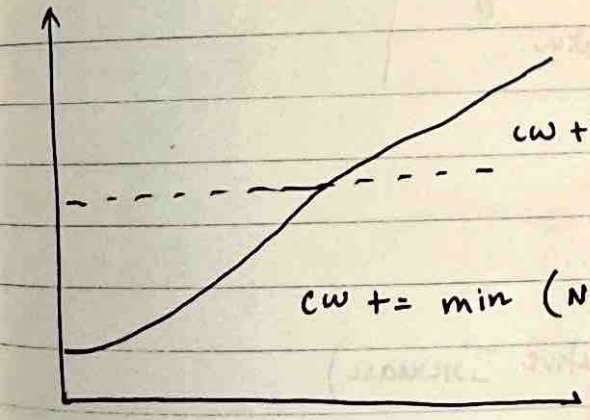
→ we should only increase by 1

Now window size = 3

and 3-4, 4-5, 5-6 sent

↓ This is for beyond ss. threshold!

cwt+=x per ACK → CWI= 1 MSS per RTT



So we want to increase cw by per RTT

cw → # segment per RTT ?

I want to have

so $\frac{cw}{1 \text{ segment size}}$

$nx = 1\,MSS$

$x = \frac{1\,MSS}{n} = \frac{(MSS)^2}{(cw)}$

CONGESTION AVOIDANCE (ADD. Inc)

↳ If $cw \geq$ SS-thrush

then on reciving an ACK

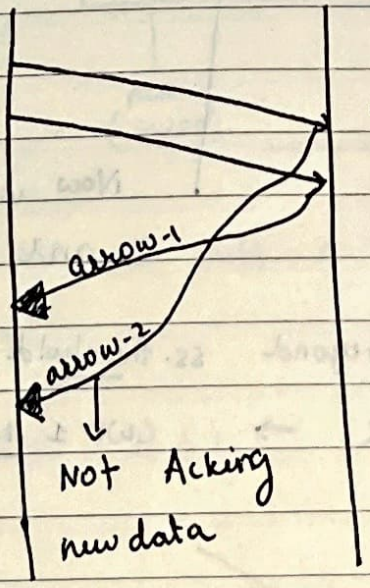
$$cw += \frac{(MSS)^2}{(cw)}$$

RFC 5681 : If $cw \geq$ ss threshold

on getting ACK for new data (non-zero number of new bytes Acked)

$$cw += \frac{(MSS)^2}{(cw)}$$

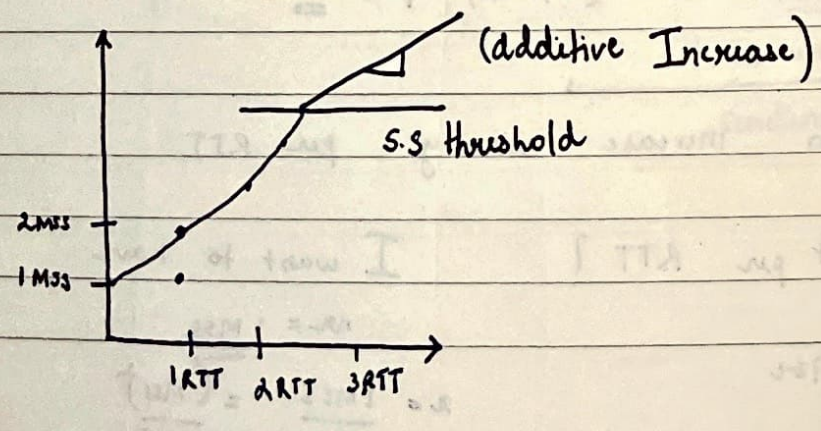
This case



→ so in this case we will only increase for arrow-1 not arrow-2

TCP Congestion Control.

Slow start, congestion control



slow start

cwt=1 MSS per ACK

for congestion avoidance

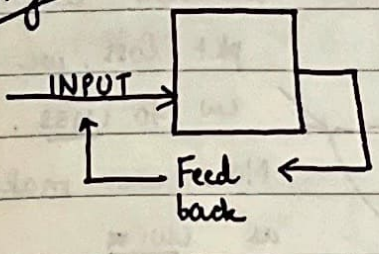
$cw \mathrel{+}= \frac{(MSS)^2}{cw}$ per ACK

How to detect and react to congestion?

STABILITY OF NETWORK

↓

No prolong condition



ACK ~ Feedback } This helps us control congestion

input (data

How to deal with feedback

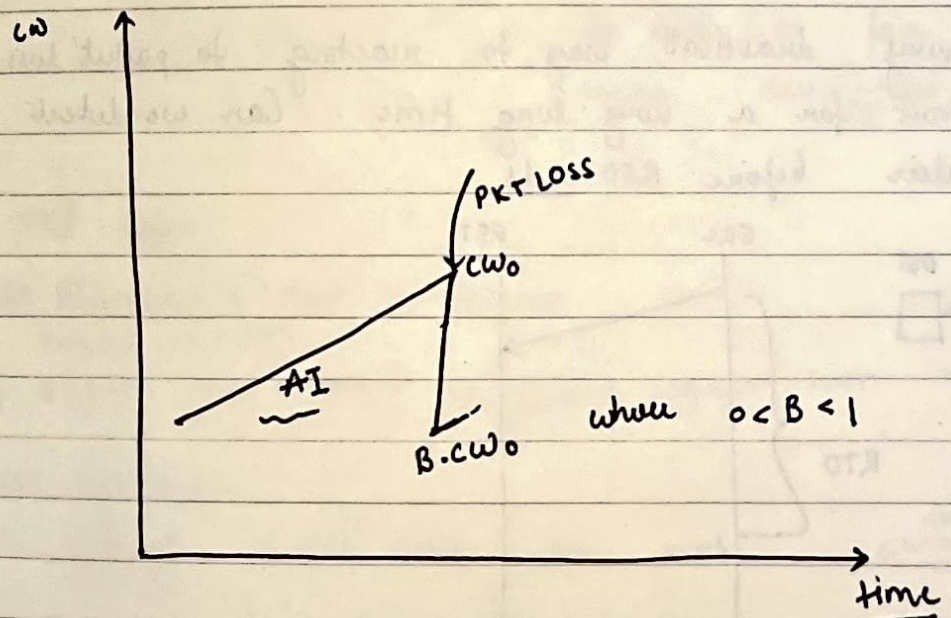
AI-MD:

Additive increase - multiplicative Decrease

↓ ↓

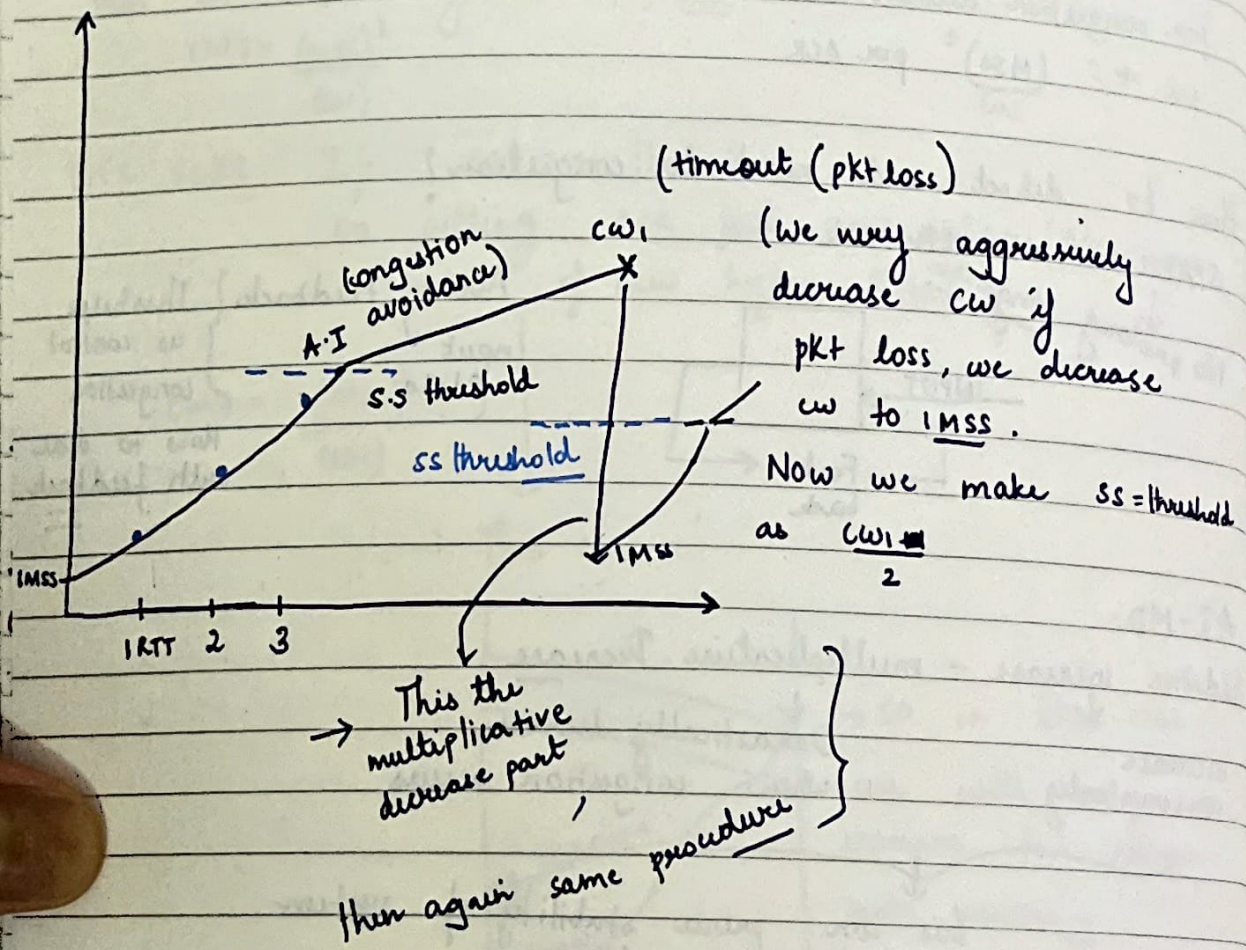
increase conservatively — Drastically decrease when congestion occurs

This can prove stability of system under some conditions.



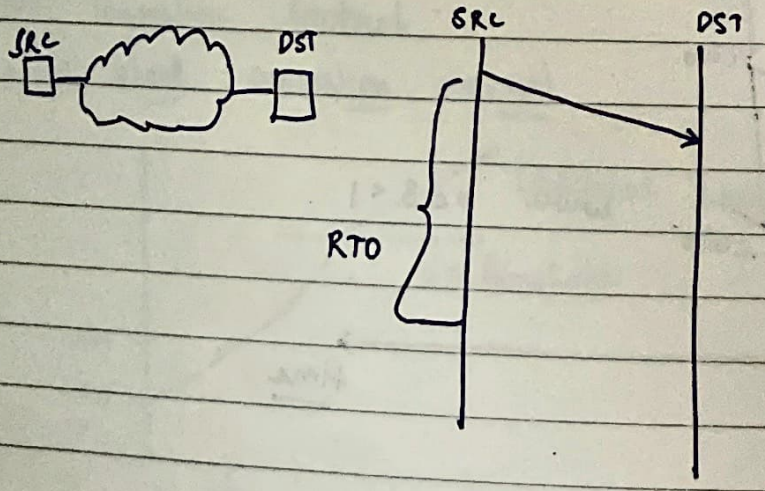
TCP TAHOE (CALIFORINA LAKE)

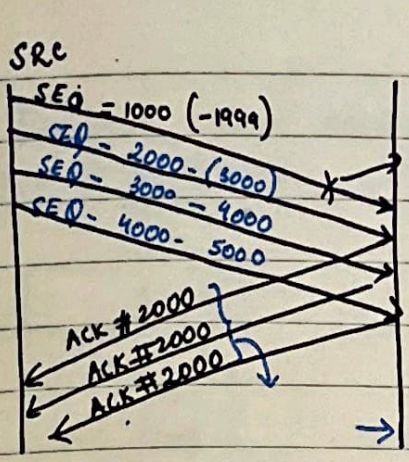
→ One of early versions: -



TCP RENO

- Timeout is a very drastic all way to reacting to packet loss. We have to wait for a very long time. Can we detect packet loss earlier before RTO ends.





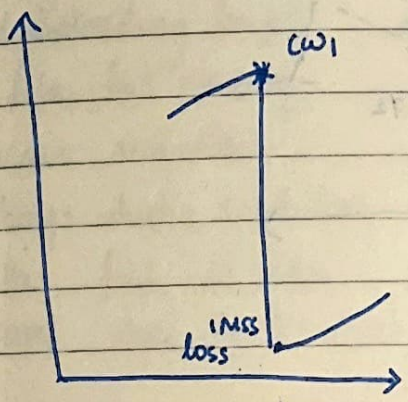
Does getting multiple acknowledge, it means pkt loss?

No, it could be packet reordering.

→ If we have lots of multiple ACK for same number the packet loss

(RULE) → 3 DUP Acks, assume pkt loss

Should we reduce cw to 1MSS for every pkt loss?



Let us differentiate between

T.O loss	T.D (triple duplicate acknowledge)
(Very drastic congestion) (No duplicate ACK getting back, no pkts getting through.	→ This means some packets are getting through → cw not very drastic

→ So we should react differently to both, so for TD ACKS we make $cw = \frac{cw'}{2} \rightarrow cw' = cw_{old}$

TCP RENO

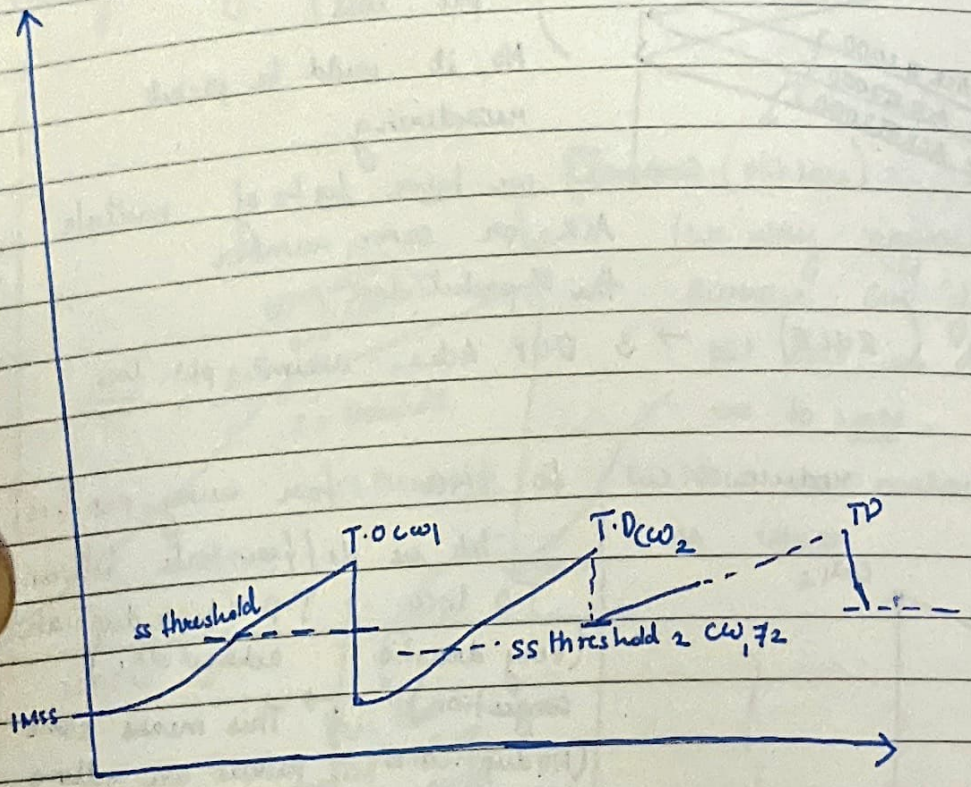
FAST RECOVERY & FAST RETRASMIT

If 3 DUP ACKs received → assume segment lost & retransmit it

FAST RECOVERY

: On getting 3 DUP ACK's we make $cw^{new} = \frac{1}{2} cw^{old}$

TCP RENO



TCP VEGAS

↳ RTT (send very much by this)

TCP RENO, TAHOE

→ triple duplicate

T.D → $cw = \frac{cw}{2}$

↓ retransmit lost segment → Fast recovery

Fast retransmit

ISSUE: Designed to cause congestion

└ We are actually causing congestion by increasing window.

So they are designed to create loss, The best thing is they use available bandwidth.

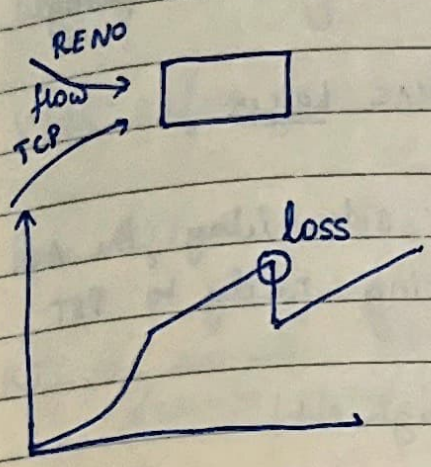
So they actually fill up the queue, so if we are sending UDP packets

↓ what could happen?

→ Due to Queues getting filled up UDP or loss sensitive can suffer.

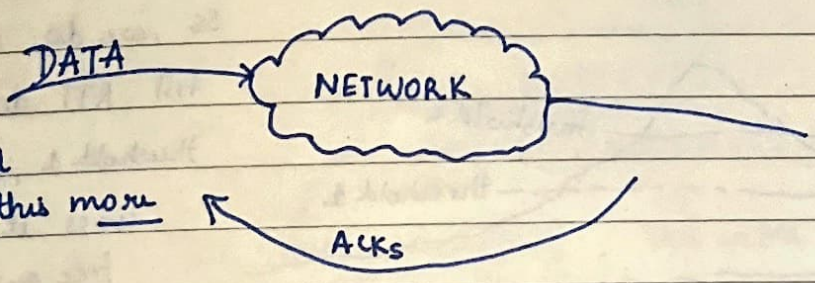
Also we get increase in JITTER↑

Also overall OWD & RTT increase.

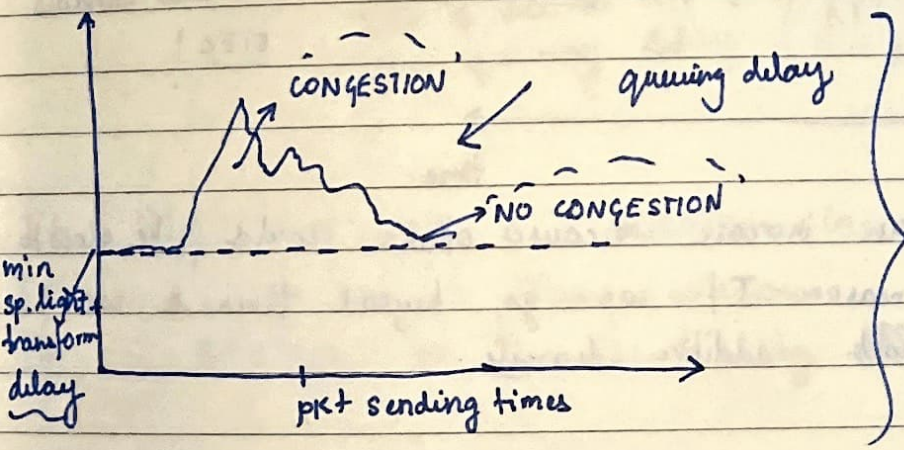


Only good for downloading large files, but playing video and other things which needs less Jitter fail in some cases due to TCP RENO.

TCP VEGAS



RTT
↓
We will consider this more



So they used this intution

TCP RENO IGNORES THIS

So looking at this graph we can take early action!
So is this intra delay due to Queuing while going to DST?

① This delay could be due to MAC Layer!
(Some protocol failure)

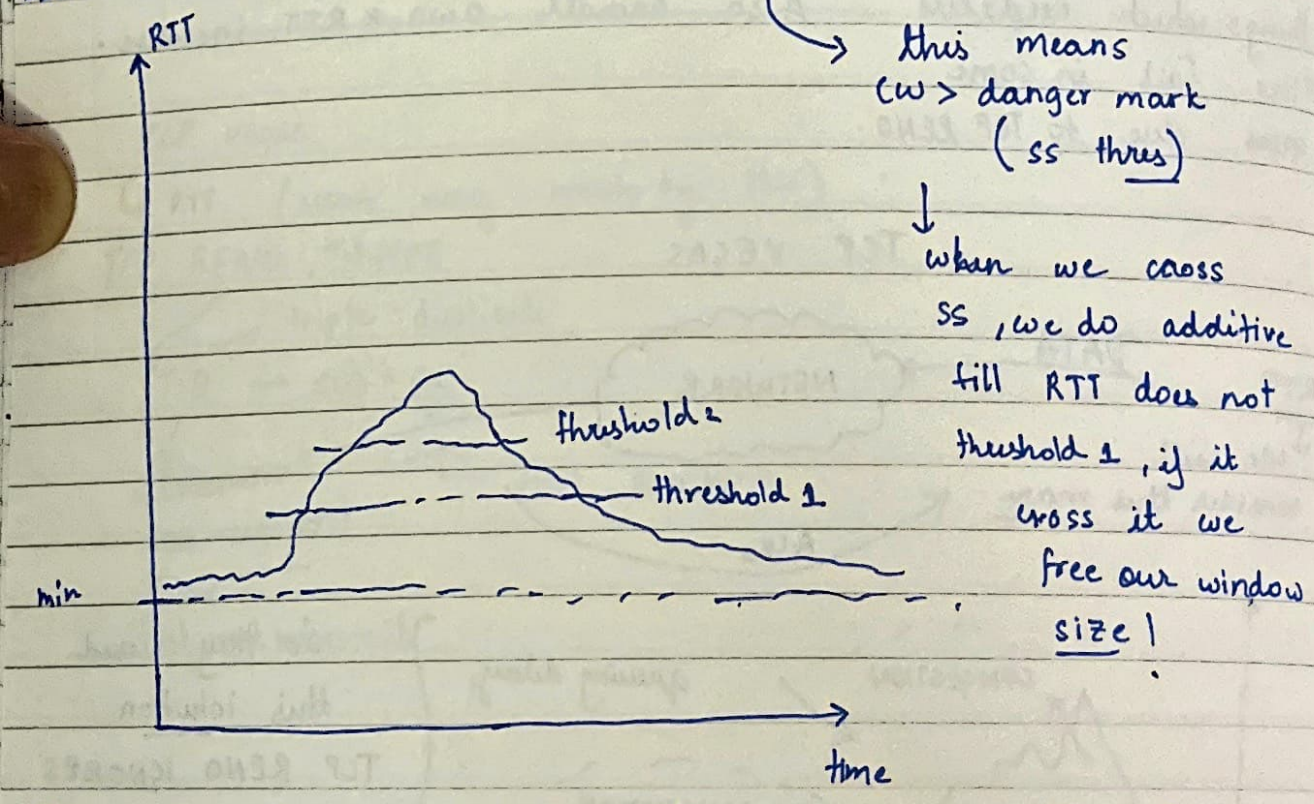
② Since this delay is due to both sides delay, the Ack could be reason. The data could be going nicely to DST but ACK's could be queued.

So why should we reduce SRC through put!

TCP VEGAS

S.S → as in RENO
T.O → " " "
T.D → " " "

But we can change congestion avoidance



→ this means cw > danger mark (ss thres)

↓

when we cross ss, we do additive till RTT does not threshold 1, if it cross it we free our window size!

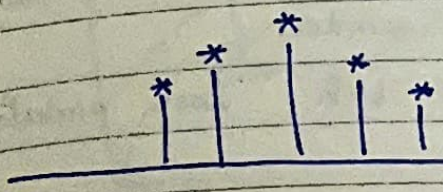
But still delay can increase because others could be slow to react or some other reason. If we go beyond thres 2 we start doing additive decrease.

→ The problem is what values should we choose? thres-1 (α) & thres-2 (β)

VEGAS RULES (FOR CONGESTION AVOIDANCE)

Base RTT — min observed RTT in some recent time window.

RTT → smooth estimate of the RTT (current smooth)



Suppose no congestion then

RTT = Base RTT (W: window)

Expected rate = $\frac{W}{\text{Base RTT}}$

Actual rate = $\frac{W}{RTT}$

Diff = Expected Rate − Actual Rate $= W\left[\frac{1}{\text{Base RTT}} - \frac{1}{RTT}\right]$

≥ 0

$= W\left[\frac{RTT - BASERTT}{(RTT) * (BASERTT)}\right]$

We are assuming this is due to the same queuing delay } this extra value in RTT is due to queuing delay

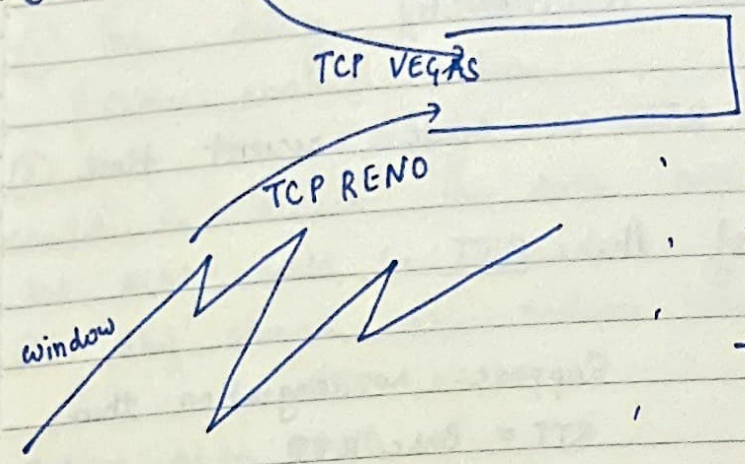
If Diff $< \alpha$ => A.I (additive increase in Ws)

If $\alpha \leq$ Diff $< \beta$ => Freeze window

If $\beta \leq$ Diff => Decrease W by 1 MSS per RTT

Suggested $\alpha = 30$ kbps $\beta = 60$ kbps

CAN TCP VEGAS be compatible with other versions of TCP?



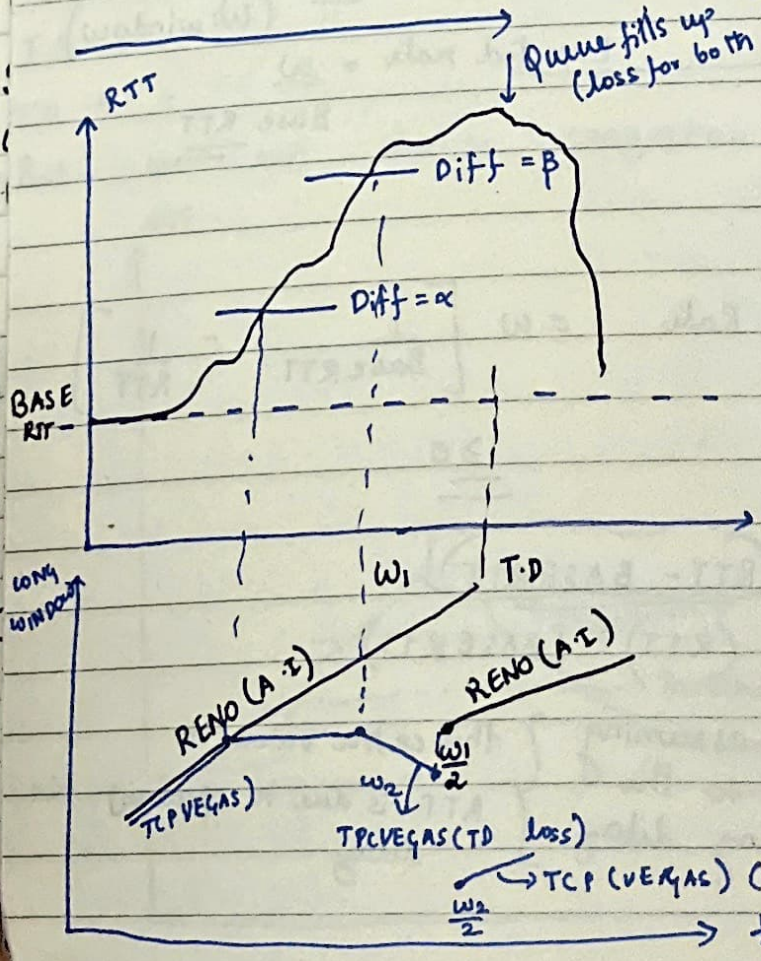
TCP starts of same way TCP RENO,

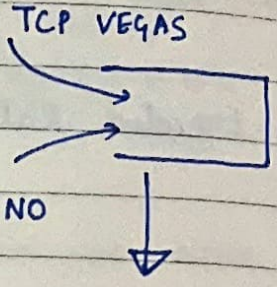
→ Queues started filling

→ TCP VEGAS Freezes window size

→ TCP VEGAS Then starts reducing window

→ Then both lose packets





→ TCP VEGAS gets pushed down!

→ So utility of person using TCP VEGAS decrease

TCP RENO Default TCP for a very long time

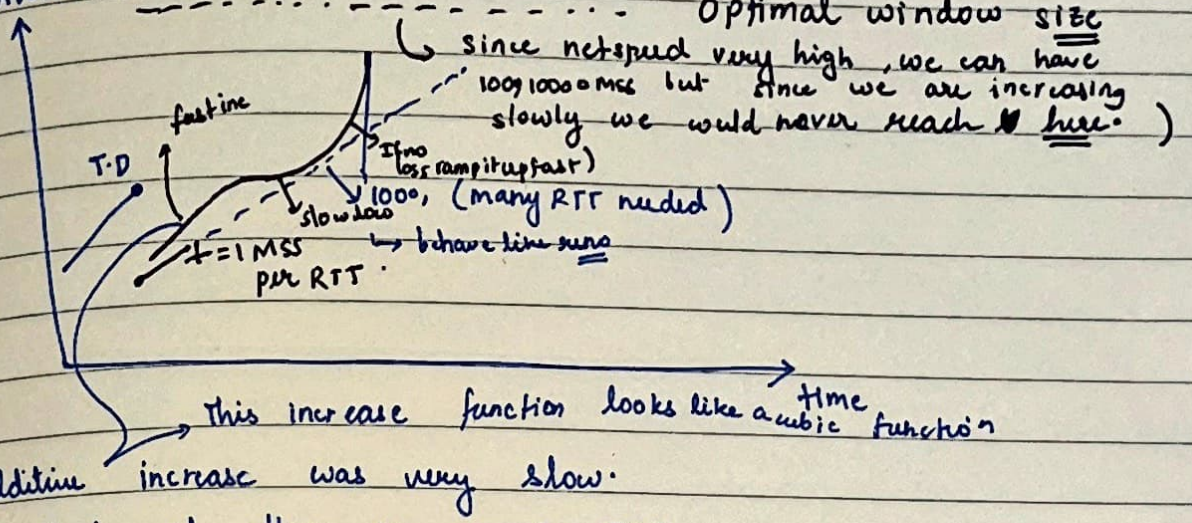
TCP RENO → 1988 – 2012 (default) by many OSes!

TCP CUBIC → (now default) (LINUX / MACOSx)

Why wasn't RENO good enough?

Now we have very high speed networks

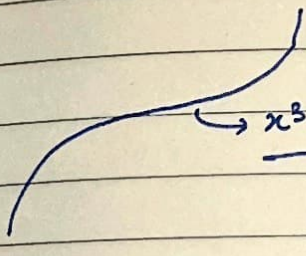
Cong. Avoidance

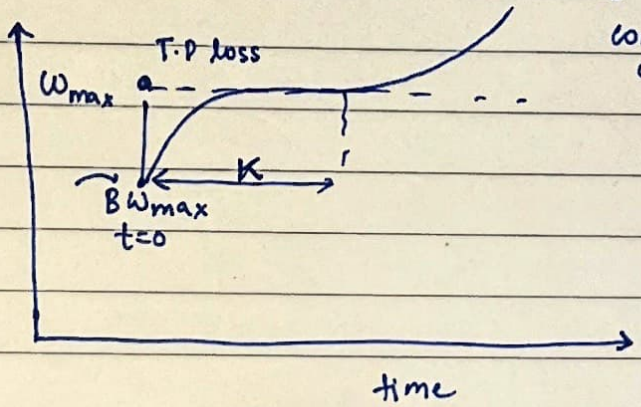


So additive increase was very slow.

↳ So what if others are using TCP reno and we are being aggressive, we could kill TCP reno.

So we want fast inc, but we want slow down (without killing TCP reno).





$B = 0.7$

$$Cwnd = c(T-K)^3 + W_{max} \quad (3)$$

let $c = 0.4$, $K = \left[\frac{W_{max}(1-\beta)}{c}\right]$

→ K does not depend on RTT

at $T=0$ $\quad cwn = -W_{max}(1-\beta) + W_{max} = W_{max}\beta$

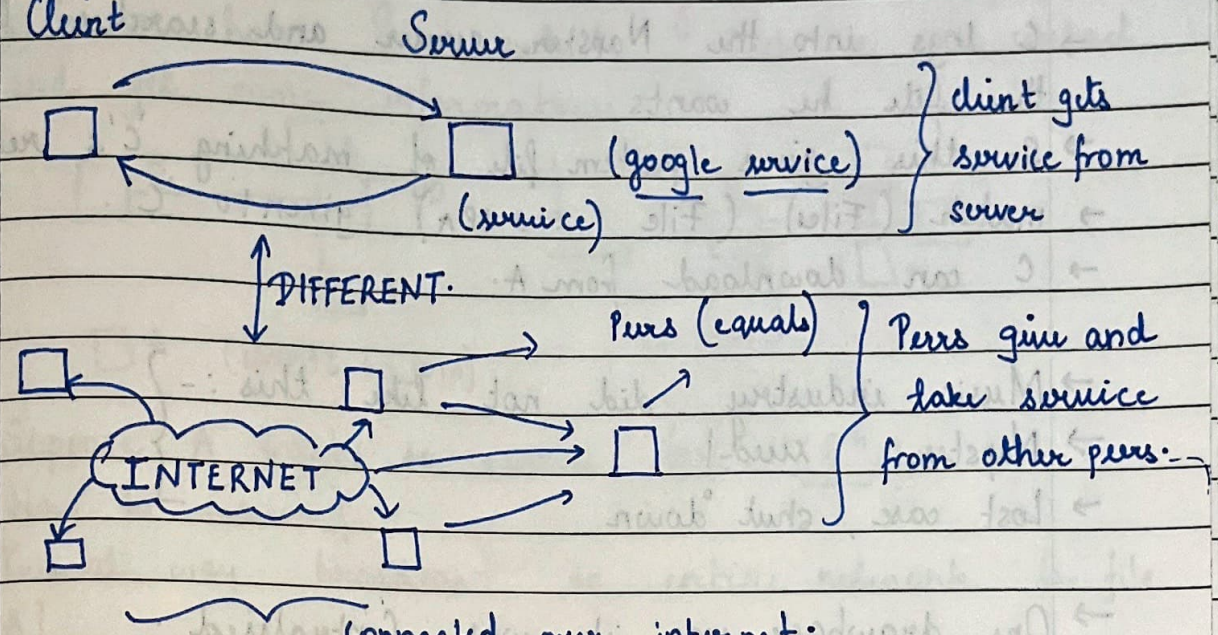
at $T=K$ $\quad$ window size $= W_{max}$

Application Layer

DATE

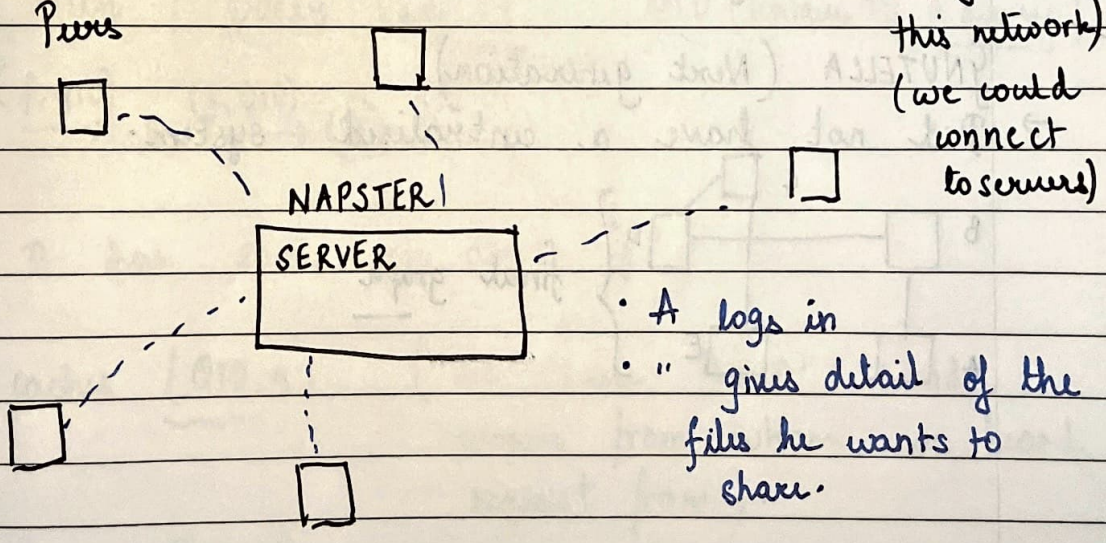
P2P networks..

1. Peer to Peer Networks.



- Botstrapping
- 1990's, why don't we share digital content with others
- peer → movie / music → peer } file: Michal Jackson's Music
- hosts

NAPSTER} (originated peer to peer network)



classmate

PAGE

NAP : Server

DATE

File 1	IP A
File 2	IP B

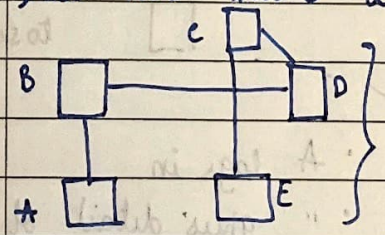
→ C logs into the Napster server and searches for the file he wants
→ So this gives pattern file of matching 'C's request.
→ match (File1) (File 1 ⟷ IP A) given to 'C'.
→ C can download from A.

→ Music industry did not like this :-
→ Napster sued
→ Lost case, shut down

→ One drawback was, it was Centralised.
→ It makes an easier target.
→ Single point of failure.
(not fault tolerant)
→ Somebody brought this down, 'ded'.

→ We need a decentralised server.
→ legally easy to take down.

GNUTELLA (Next generation)
→ Did not have a centralised system.



final graph

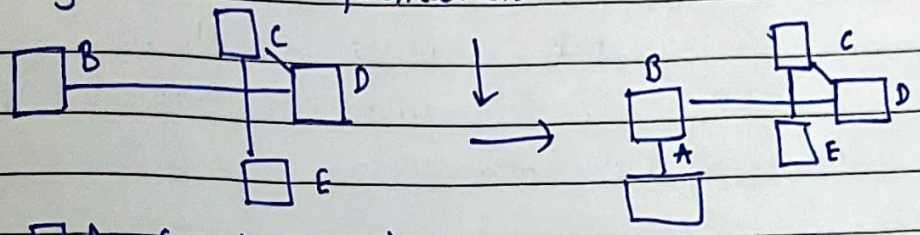
DATE

So how this network is created!

Bootstrapping

(1) Application may have IP's of some other peers

(2) So we have some website which we can contact and get some information.



A (wants to join)

Suppose A wants to search for "f"!

→ How to search?

→ Trivial way - broadcast to entire network for file "f" by A?

→ Over kill, we do a limited broadcast)

→ LIMITED BROADCAST (TTL = n, TLL -- (at every hop), TLL == 0 ⇒ don't fwd query).

Suppose n = 2

A — B — D → So let D have the file.

TTL=2 TTL=1 TTL=0 → So how do they transfer?

First Meth : QUERY had "f", QID (unique ID of query)

A —(f, QID)→ B —(f, QID)→ (D) (file NAME)

(So D has some info about "f")

B caches (QID, A) (we save "QID" and person from whom we heard request from).

D " (QID, B).

classmate

PAGE

DATE

(B searches according to QID)

D ⟶ B (QID, IP_D, Port_D ↑)

reply (IP_D, port number)
QID D

→ A can now request info from "D".

→ Now B can store some information about "f".

→ So B caches reply. (QID, "f", IP_D, Port_D).

→ So now G joins and asks for "f", B directly replies to G.

→ If we do not file we can search again by increasing ttl.

→ We keep ↑ing ttl till max value reached.

→ So in future GNUTELLA put A's info in Query.

→ So A & D directly communicate. (But now B cannot cache all this information). (The reply information)

→ Second Method:-
Query has IP_A, so directly send to A.

→ So the QID helps to prevent multiple forwarding of a query!

→ So each person makes sure to forward to all neighbors only once.

V.0.4 → max TTL = 7 } Why do we dec max TTL.
V 0.6 → max TTL = 4 } allowed?

When we have large networks!
Graphs become dense!

DATE

TTL=4 could make Query reach ~10^4 nodes
TTL=7 " " " ~10^7 "

→ So we decreased TTL overtime as GNUTELLA Become popular. Overkill hai!

→ Good → No centralised node!
→ We have many broadcasts! (Bad!!)
→ (Is there a clever way to avoid these broadcasts").
→ (Where info is stored?)

→ Suppose there are N-nodes!
→ can win by contacting only log N nodes
→ limited broadcast does not work.
→ How do we store info.
Suppose we have identifier of nodes, lets use IP addresses.

IP_A
IP_B
:
IP_E
IP_F
:
IP_Q – has f_1
IP_S (has f_2)

Files
f_1
f_2
:

suppose we could map these file name to IP address. (And we store this info somewhere else).

Can A in some clever fashion ask Q where is f_1 or some other node)

→ So Q tells 'E" I have "f_1"
So A asks "E" " " "

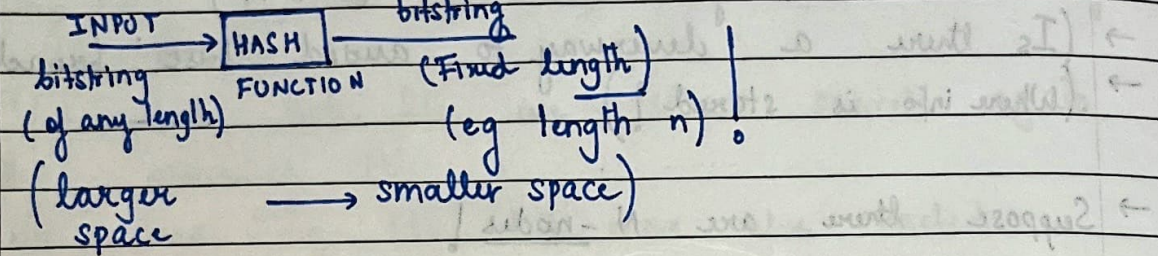
So all nodes having "f_1" tell E. So A gets info and downloads this from "Q".

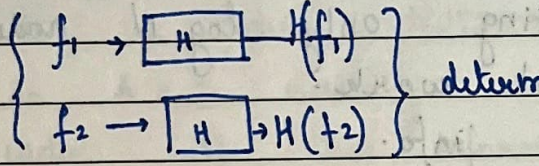
DATE

S stores it has "f_2" in "B".
So if 'f' node wants to search for f_2 it contacts "B".

So f_2 mapped to B
So f_1 " " E

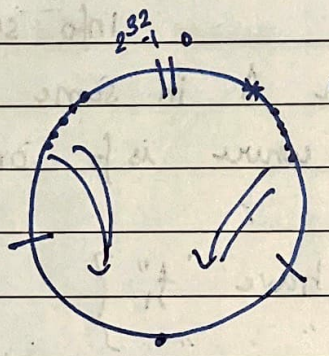
One method is to use a hash function.





H(f) are in domain file of file to IP Addresses.

→ So we choose closest IP to H(f).
But the issue is the networks has no uniform distribution of IP addresses.
→ So some nodes do a lot of work.

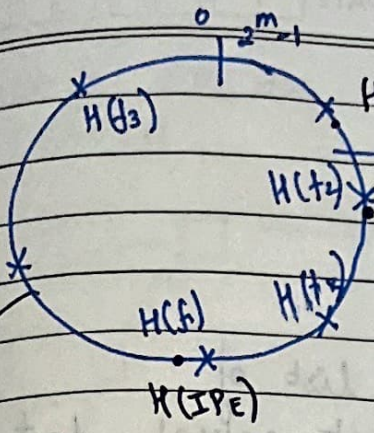


32 bit IP addresses space.

So one node storing info about a lot of files.

So they suggested to hash of IP addresses

DATE



m = length of hash o/p.
→ pseudo random numbers.
since H(f) → goes with / distributes uniformly (uniform distribution)
so now take hash of IP

(so f_1 is closest to E node)
(so E must store information about where f_1 is stored.
→ In this abstract space we are embedding information.
→ So we can zoom in and find where files are.

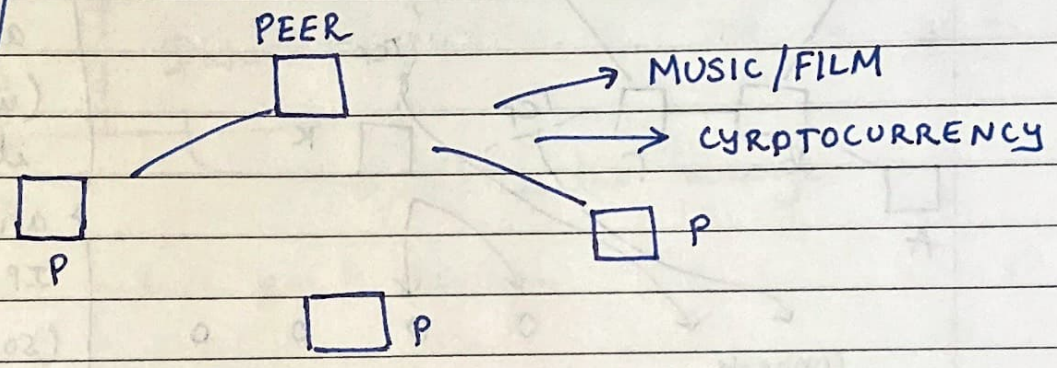
→ BIT TORRENT } many files can have many nodes.

P2P Networks :-

APP layer
TRAS
NET
DLL
PHY

Many applications have client to server.

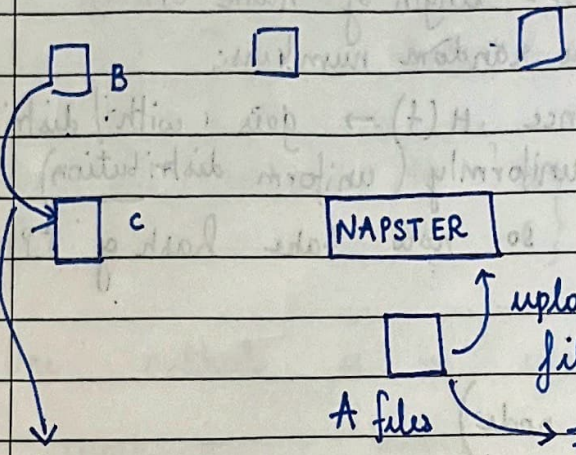
CLIENT SERVER.



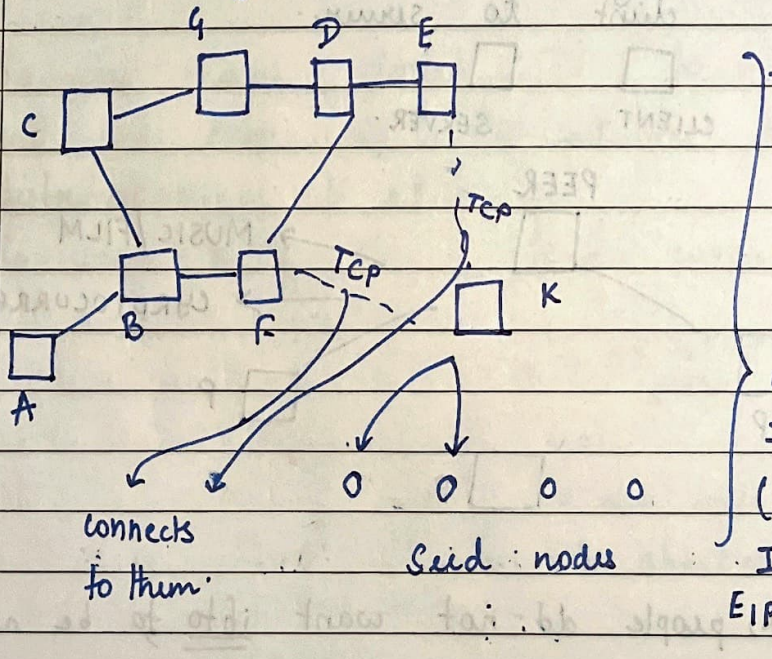
1ST GEN:-

→ NAPSTER } → As people do not want info to be always be on their laptop.
→ So a main server is useful.
→ But centralised.

DATE



→ Suppose C has a file for B
NAPSTER told B about it
Now B & C connect via TCP and share info
↳ reliability

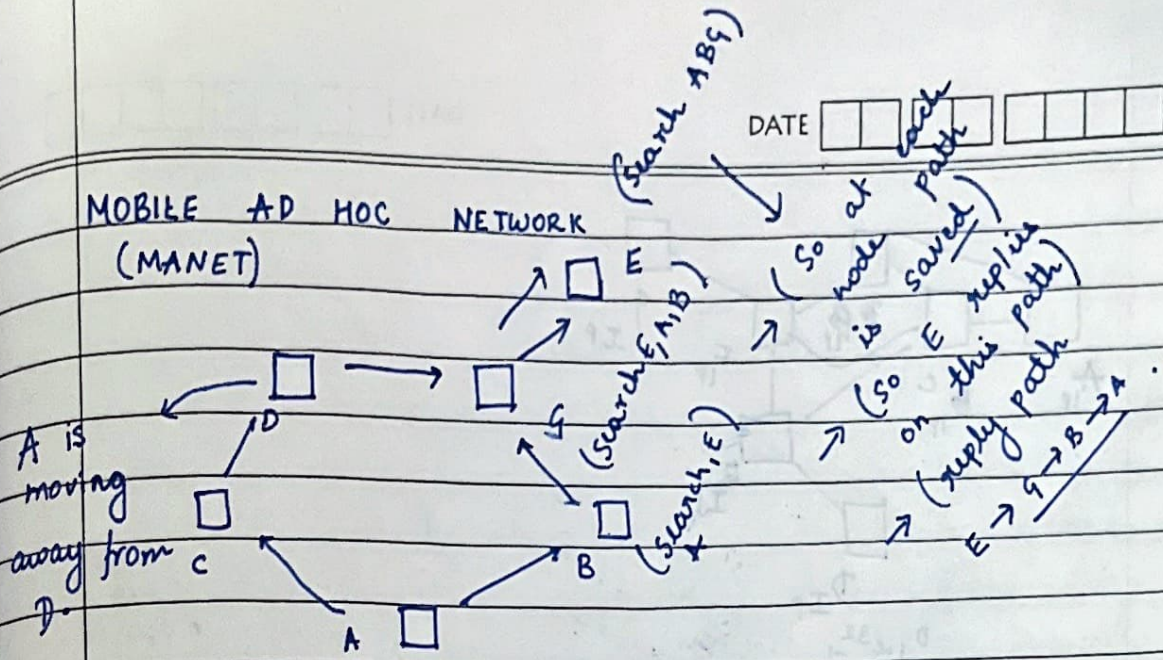


→ K wants to join, asks seed nodes (whose IP it knows) and asks for IP in network. (so it gives F's IP)
E_{IP}

→ As people start downloading files they make new friends.
→ The network becomes more dense!

DATE

MOBILE AD HOC NETWORK
(MANET)



tanks (moving around in a battle field | cellular networks are down).

How to do routing.

(Searching for E)

(Suppose D links breaks down)

So for routing we use source routing.
(Not IP routing)

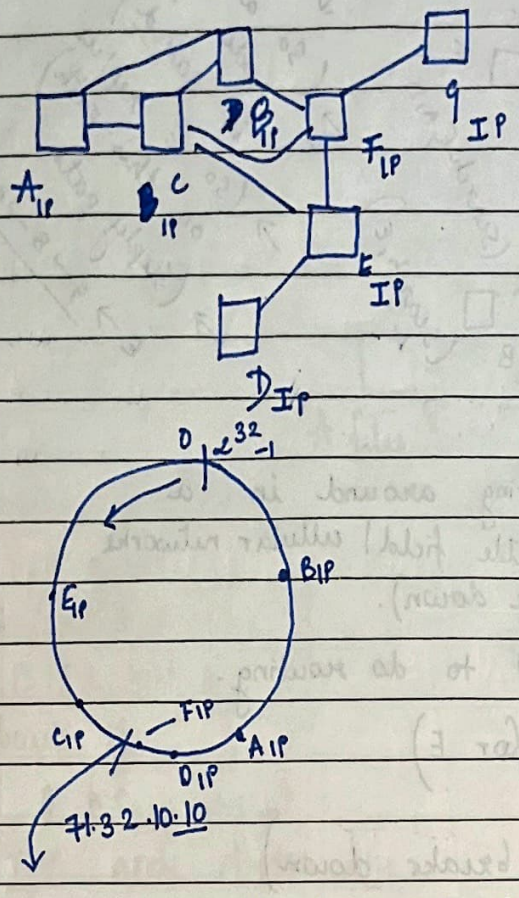
→ So message will have full route.

(A-B-C-E) (full route)	Data

A-E message → this is the message.

→ Because nodes are moving.

3RD GEN P2P: Distributed HASH TABLES (DHT)



So A wants to know who has IP closest to point 71.32.10.10

This a subproblem for P2P networks efficiency.

→ So A has IP address of B & C.

→ In my neighbouring node who has IP closest to this point.

→ So c gets my IP request

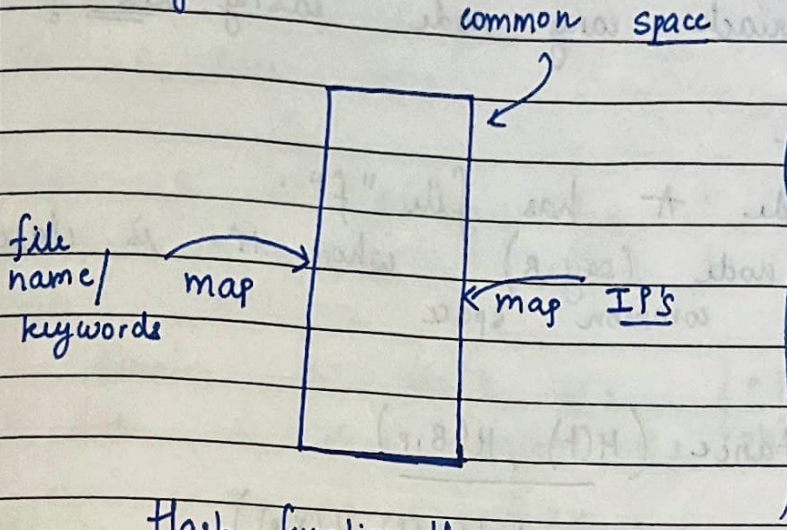
→ Now c checks in its neighbour who is closer.

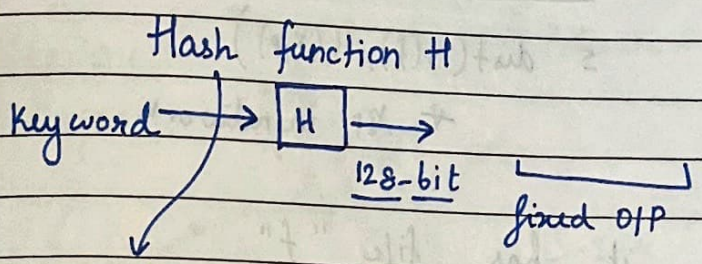
→ So we send it E_{IP} but if we had F_{IP} as neighbour

→ we would send it to C.

DATE

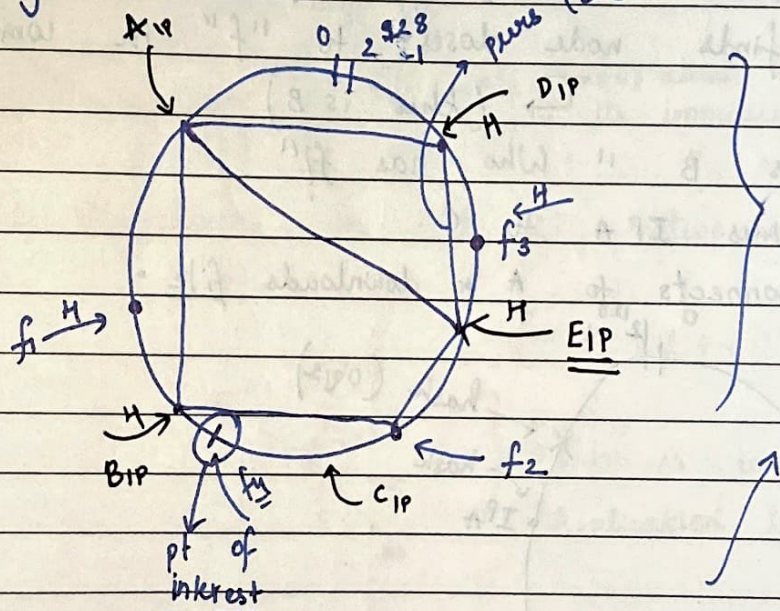
IDEA: file names keywords → map → IP





→ for even a single bit change the O/P is very random

So Eg.



↳ This pt is closest to IP of B.

A - D - E - C - B

(so A takes hash of their neighbours IP)

($IP_B \xrightarrow{H}$ $IP_D \xrightarrow{H}$ $IP_G \xrightarrow{H}$

PAGE

DATE

So we want to who has f4, so we give information:-
We can reach any node easily know!

→ HIGH LEVEL:-
Suppose node A has file "f".
A finds node (say B) whose IP is closest to f in common space

Means distance $(H(f), H(B_{IP}))$

$\leq dist(H(f), H(X_{IP}))$

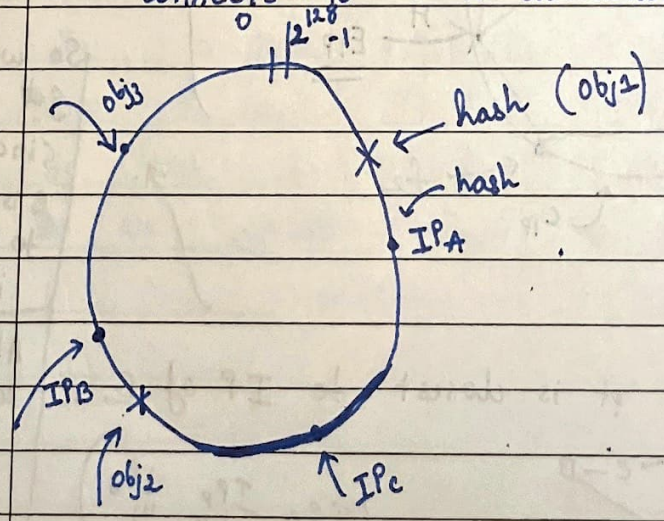
$\forall X_{IP}$ in network

→ A tells B it has file "f"

File	IP
"f"	IP A

→ Suppose C wants file "f"
C finds node closest to "f" in common space.
↳ (this is B)

- C asks B "Who has f?"
- B gives IP A to C.
- C connects to A & downloads file.



DATE

Node C has obj 1

→ Now hash of obj 1 is closest to hash of value IP$_A$ IP$_A$ it has.

→ So C tells obj 1

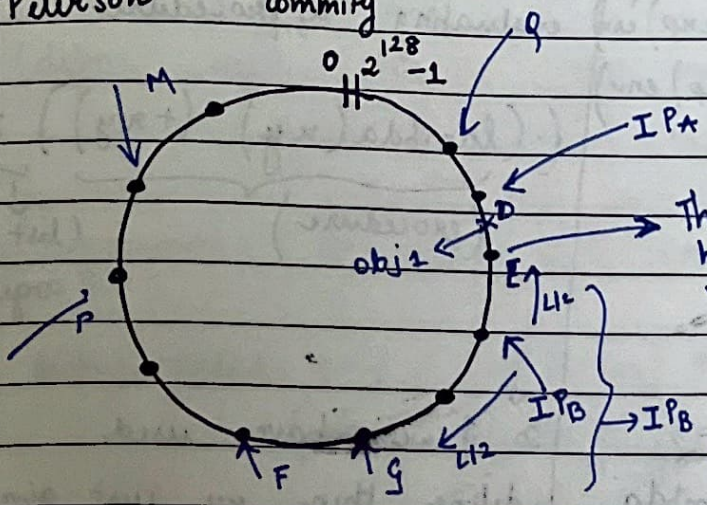
→ Node B is looking for obj 1.

- B finds A, which is closest to obj 1 in common space.
- B queries A about obj 1
- B tells A that C has obj 1

- B downloads obj 1 from C.

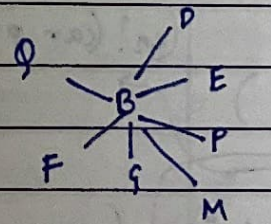
PASTRY - Peterson ← Druschel ← Adviser Peterson

Thais commity



→ Leaf Set: Neighbours in P2P networks of any node

(size L + O(log N))



→ This is just IP layer! (Application level)

→ M wants obj 1

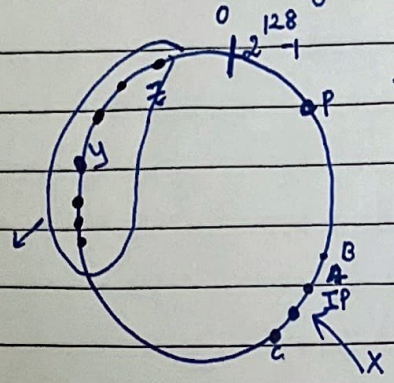
→ M should find peer closest to obj 1 in its own leaf set.

classmate

PAGE

DATE

→ So M will find B to be closest, it will forward message to B.
→ So now B now D is closest in its left set.
→ It forwards to D.
→ D has no neighbour closer, so D tells M.



→ Suppose Node X wants to join

→ So X does not know its next IP's (L/2) walk)
→ So X ask y to help it to join.
→ So y finds in its leaf set Z is closest
→ Z finds P, P finds A.
→ A finds itself to be closest.
→ Now A can tell L/2 people on its right and L/2 people of left to X.

- X" contacts some other node already in network (Y)
- " Y" routes message to node closest to hash (IP x).
- Each node on path (Y, Z, P, A). become part of leaf set of A.
- A shares info about L/2 on left & right
  ↳ join leaf set of x.

DATE

128 bits → No of digits looking for here O(N) (O(log N))

d 2 f e 3 7 - - - - (hex)

y → z → p → A

w.h.p

z will d as first hex digit

d 2 - ....

d 2 1 ...

hash IP$_z$: d _ _ ...

(we are looking for longer prefix match)

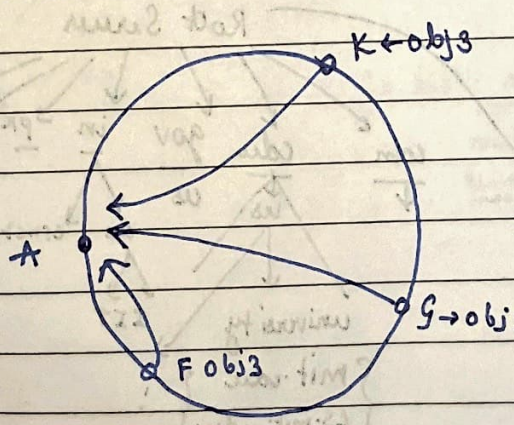
→ Now what if a node fails

→ So this can be done by communication of periodic messages at application layer.

→ So we can take care of this by asking other neighbours of their L & R.

→ Also we store copy of some information at some nodes.

BIT- TORRENT



Now files are with multiple user, so we → can download from all of them by allocating which chunks of data we want.

DATE

DNS, HTTP

→ DNS → domain name system.

→ IP
32.75.5.9 } Defense/Army were comfortable with this

→ But humans cannot understand this.

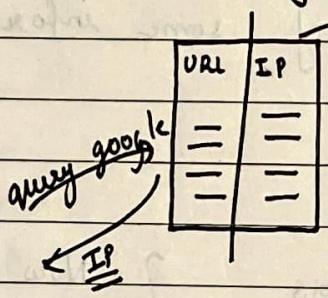
→ so for " google.com , itt.ac.in
→ these have IP mapping

→ www.google.com — IP ?
→ So server should know who to send message for this.

→ But network protocols do not understand this.

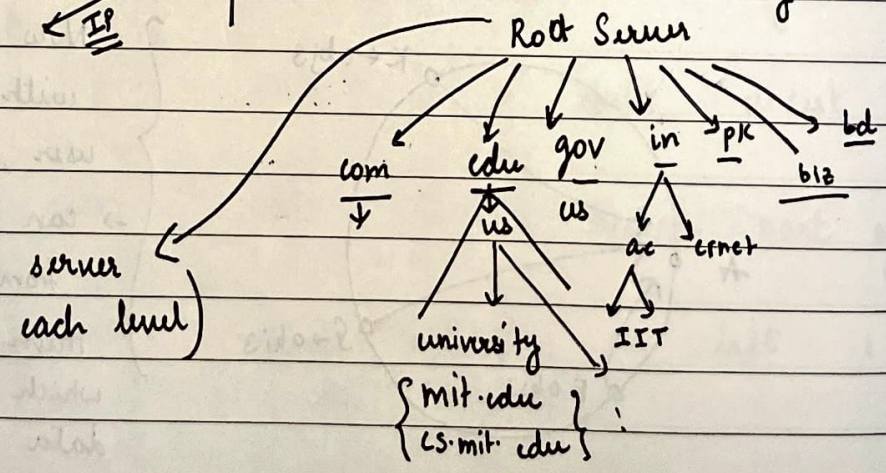
→ Also one more ad any url can change its IP.

→ But if we have a single server it is not scalable, because if it fails then what to do.



→ single DNS server → single pt of failure
→ Not scalable

↓ to save this we have heiarchy



(DNS server at each level)

DATE

→ so for IN → we have a DNS server and it should have IP address of DNS server of all lower in heiarchy at its node.

DNS :-

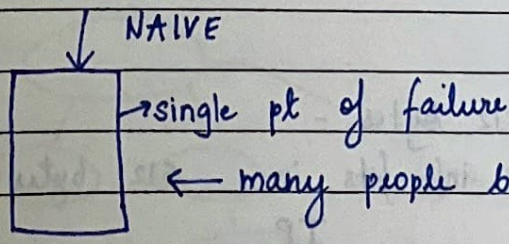
we want human readable strings like

www.google.com 8.32.7.9.

name @ ittb.ac.in :→ IP address :- 32.23.5.22.

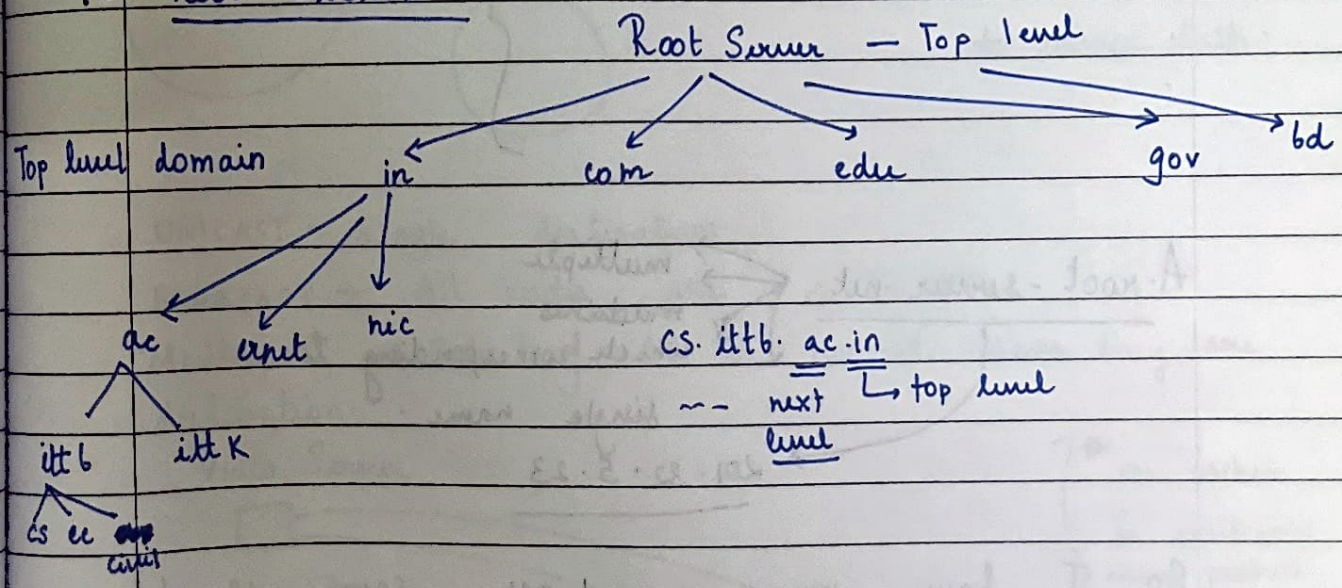
email server DNS

DNS takes us from name to IP address.



→ Hiarachial Solution.

:- Root Servers



classmate

PAGE

DATE

→ So at Root Domain we will have DNS servers. and at node we have DNS servers.

→ Also point to be noted :- DNS server of parent domain knows the DNS servers of immediate childre.

→ name of DNS server, IP address of DNS server.

Sinya.cse.itb.ac.in

↳ So I have to change its name, IP address corresponding to it by just telling its local friend.

ROOT SERVERS.

DNS pkt size : 512 bytes :-

13 ROOT servers → info fits in 512 bytes.

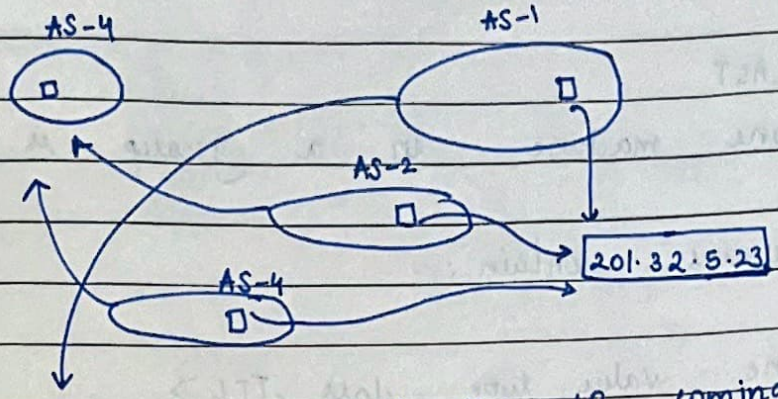
Names	IP
A. root-servers.net	
B. - - -	
.	
.	
.	

A.root-server.net → multiple machines which corresponding to a single name.

→ 201.32.5.23

So I have many machines same IP address, so how does layer 3 react to it

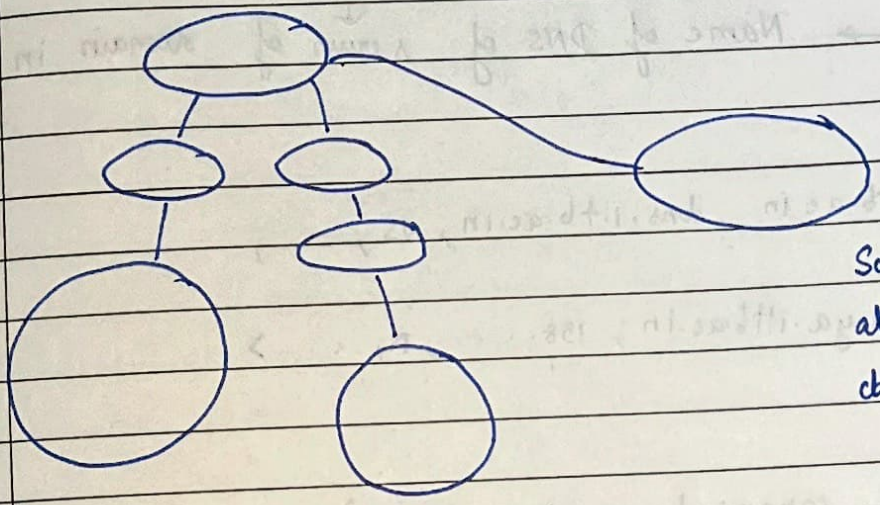
DATE



→ BGP advertisements of this IP coming from multiple as path.

→ so it seems a server connected to many, but diff machines.

→ so what happens in case of multiple AS.



* multiple paths

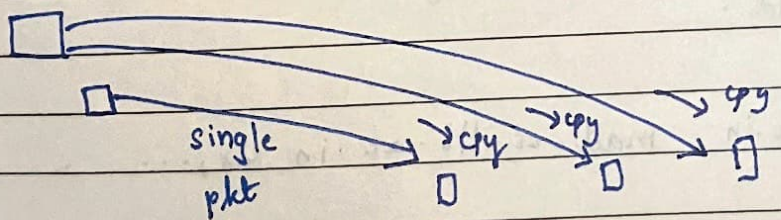
So we look at BGP attributes to choose Path.

UNICAST - Single destination.

BROADCAST - All nodes are destination.

Multicast → Subset of nodes in internet where they are destination.

Video Server



* one packet is split into many packets at network.

DATE

ANY-CAST

→ Any one machine in a group is destination.

→ DNS servers contain :-

<name , value, type , class , TTL >

(in: internet) eg

* how long this record is valid.

TYPE : "A"

→ value corresponds to a IP address.

→ "NS" → Name of DNS of server of domain in "Name"

Ex <iitb.ac.in , dns.iitb.ac.in , NS , ... ,

< surya.iitb.ac.in , 138... , A, ... >

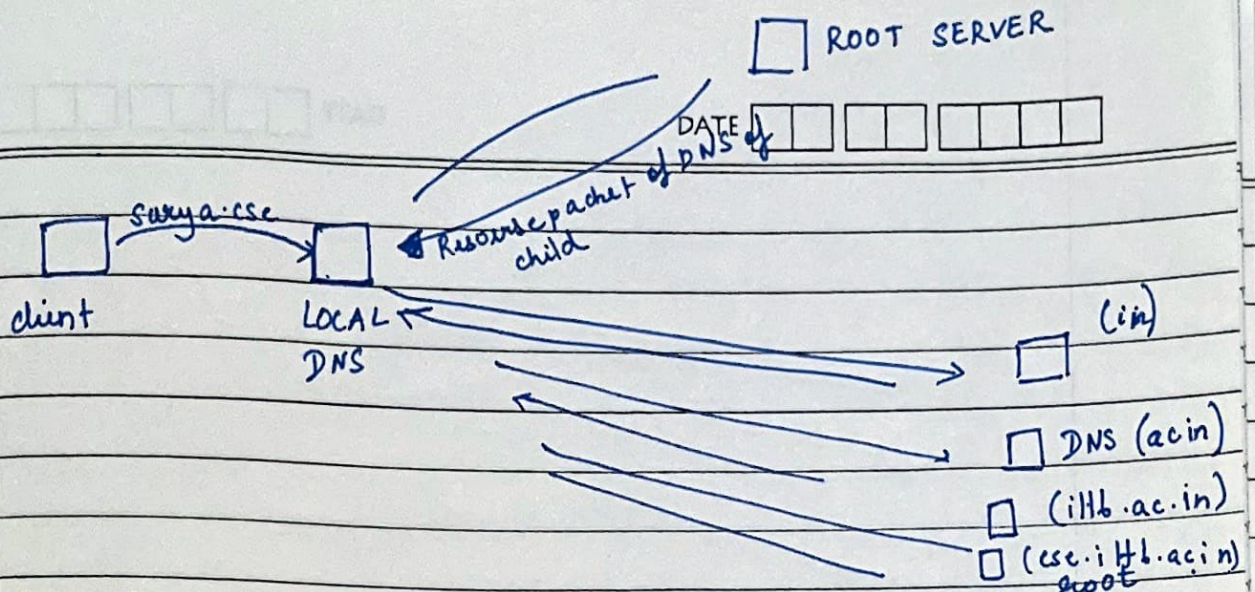
TYPE

CNAME → canonical name (alias)

cs.mit.edu
ee.mit.edu } → both have same IP

MX → email server of domain specified in name field.

<cs.iitb.ac.in , mail.cs.iitb.ac.in , MX , ... →



→ So local server that no seconds ask the server.

DATE

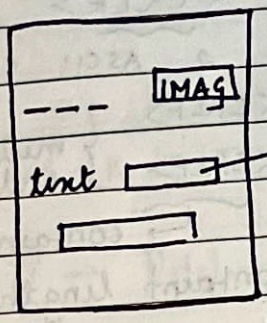
Application Layer :-

P2P, DNS :-

see detailed example in text book.

HTTP : Hyper - text transfer protocol.

→ Help us download and submit information to webpages.

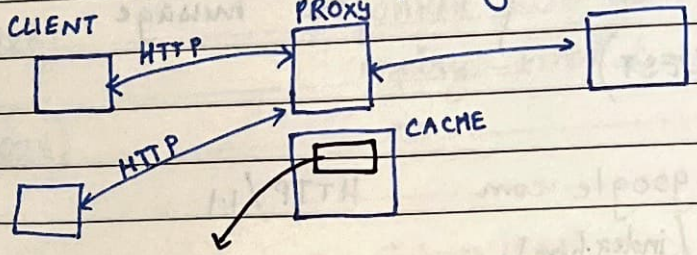


→ hyperlink (This is a powerful informat^{n} / invention)

So from one page we can navigate to another webpage.

So DOWNLOAD PAGES & SUBMIT INFORMATION :-

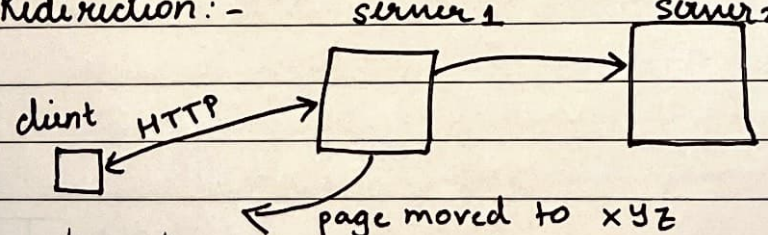
It also allows proxy.



copy.
(we will have some expiry time).

* So let us say people want download same page.
So at proxy we save it in cache and we distribute to users.

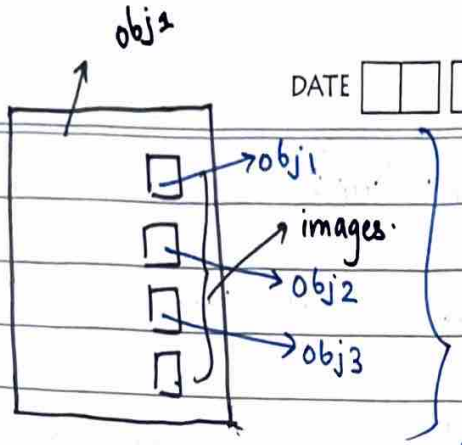
Redirection :-



we can shift information from one url to different url.

DATE

HTTP
TCP



How does HTTP send these informat simultaneously, one-by-one, open new connections for each obj.

So HTTP uses UDP QUICK (1.3)

(HTTP MESSAGE format
(start line) - - - - - - - - - - <CRLF>
2 ASCII CHAR

(MSG HDR) - - - - - - <CRLF>
- - - - - - - - - - <CRLF>
} multiple lines ending with crl.
↳ contains }

Header ends at <CRLF>
~~empty line just having~~ CRLF.

Message body

- content length : 1256 bytes
- Expiry time : - - ?
- ENCODING USED FOR MESSAGE } information about the message.

Example:- (OF REQUEST)
GET <URL>
http://www.google.com HTTP/1.1
Alternatively index.html.
This information can also be put in the header.

IN HDR: HOST : www.google.com.

DATE

REQUEST OPERATIONS

GET → RETRIEVE DOCUMENT.

HEAD → RETRIEVE ONLY META INFO (expiry time, last modified time, length, ...)

to know if we want download new copy of file or not.

OPTIONS → AVALIABLE OPTIONS (HTTP version at server).

POST → GIVE NEW INFORMATION TO SERVER.

way to send info

PUT → WE MODIFY SOME INFORMATION AT URL OF THE DERVER / store.

DELETE → Delete specified URL.

Responses : like

Example

(Start line) HTTP/1.1 202 ACCEPTED <CRLF>

(HDR) content length :-

expiry time : (CRLF)

(BODY) —

RESULT CODES :-

1xx Informational.

2xx SUCCESS.

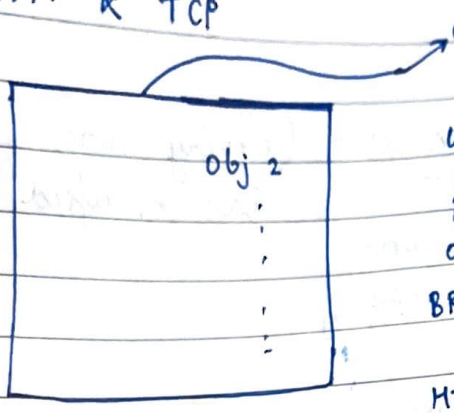
3xx REDIRECTIONS.

4xx CLIENT ERROR.

5xx SERVER ERROR.

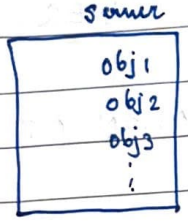
DATE

HTTP & TCP



When browser downloads the first one, it asks for all objects within.

BROWSER
↓ REQUESTS
HTTP
↓
TCP



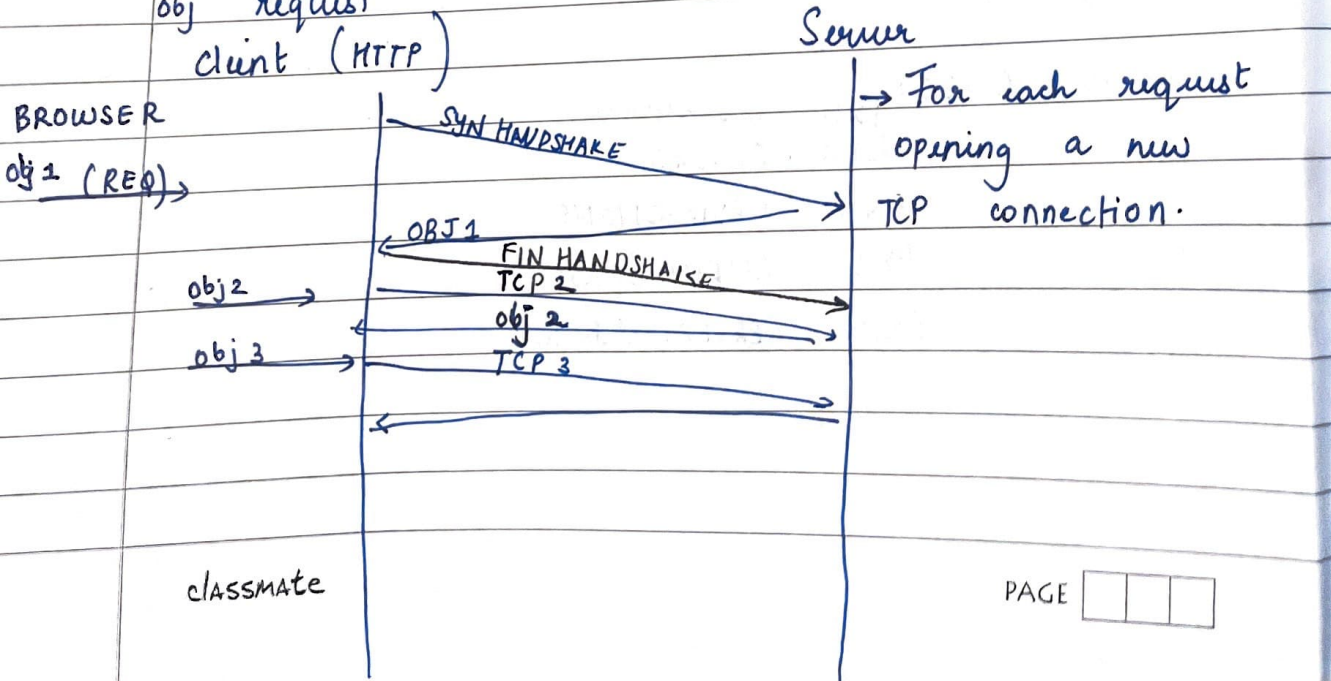
tricky thing to understand how to send all objs, if obj1 > obj2 !

Then if we keep transfering one obj then we can be stuck while sending a large object, while others could be quickly sent.

Q. SEND all request at once or sequentially
↳ one Request per TCP connections!

1.0

→ Send obj by opening new TCP connection for each obj request.



→ For each request opening a new TCP connection.

DATE

ISSUESES :-

1) EACH TCP connection takes time to learn optimal CW

2) overheads (Hand shake)

3) Server state is large due to multiple TCP connections being open.

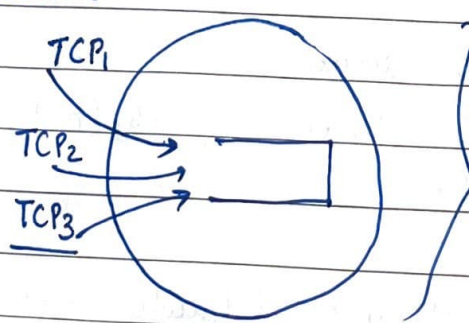
Server :

(port : 80) https : 43? 443?

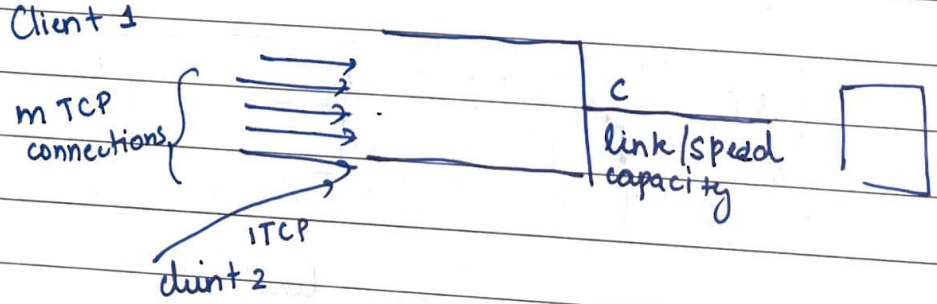
(8080)

http

4) FAIRNESS



* By opening our own many TCP connection. we get an unfair advantage



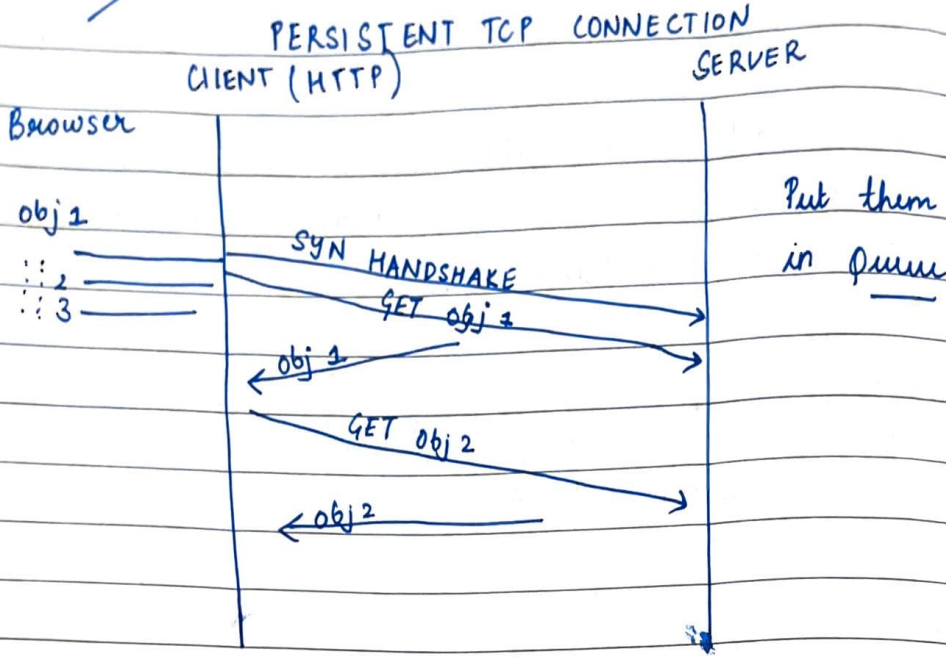
each link gets $\frac{C}{m+1}$ of capacity.

$\therefore$ Client 1 $= \frac{mC}{m+1}$

client 2 $= \frac{C}{m+1}$

DATE

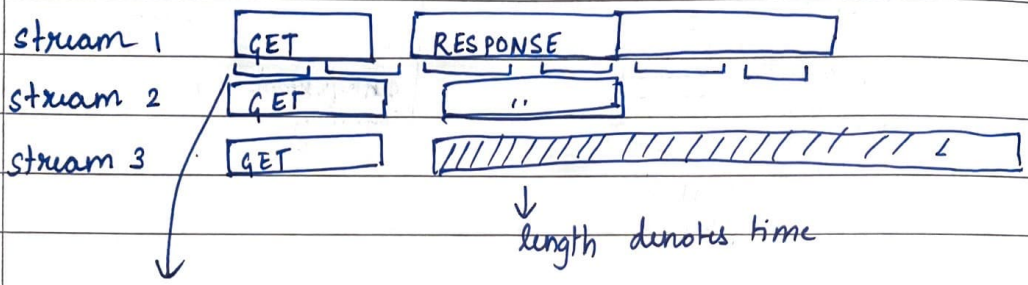
HTTP ~~1.0~~ rejected 1.1



Head of line blocking issue

→ obj 'n' gets delayed because obj(n-1) is taking time to download.

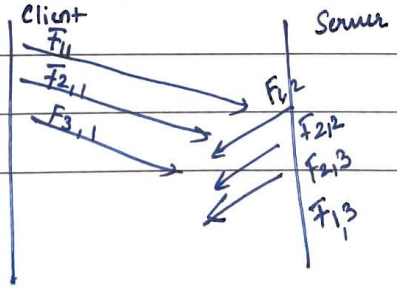
HTTP 2 → single TCP, streams & divide into streams



divide them into frames.

→ $F_{1,1}, F_{1,2}, F_{1,3}, F_{1,4}$ frames of first obj;

Over the same TCP connection



classmate

PAGE