

01 SATURDAY  
OCTOBER

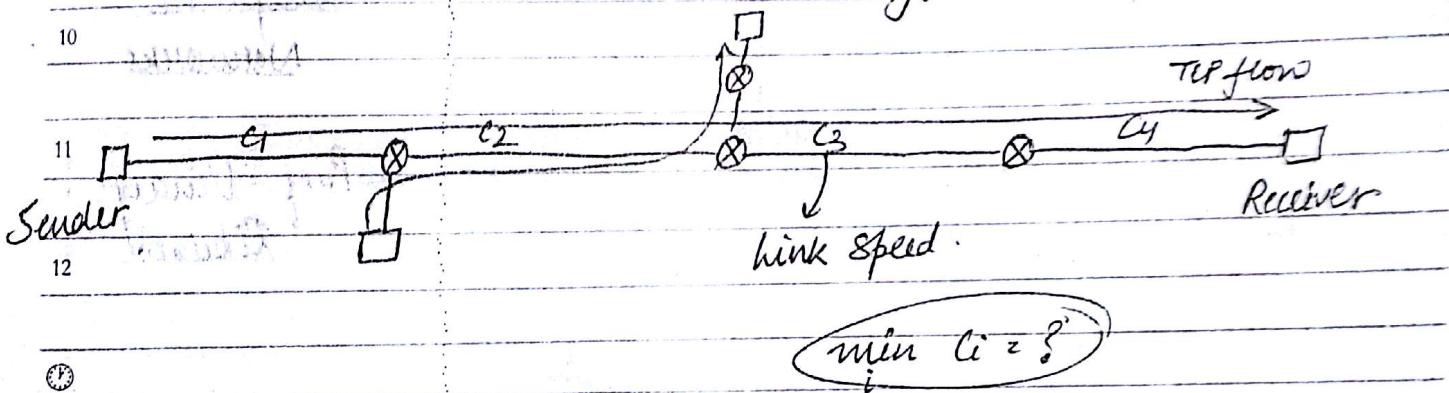
## TCP Congestion Control

September'16

October 11

M	T	W	T	F	S	S	M	T	W	T	F	S	S
			1	2	3	4	31				1	2	
5	6	7	8	9	10	11	3	4	5	6	7	8	9
12	13	14	15	16	17	18	10	11	12	13	14	15	16
19	20	21	22	23	24	25	17	18	19	20	21	22	23
26	27	28	29	30			24	25	26	27	28	29	30

Windows,  $W = \min$  (long. window (lw),  
Actual. window)  
↳ flow control.

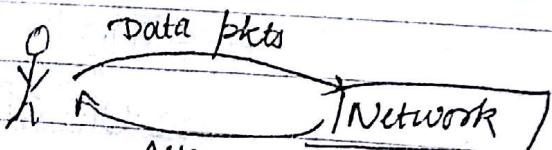


- 16 Use available bandwidth efficiently
  - 17 Must be "fair" to other TCP flows
  - 18 "Learn" correct Data Rate.
  - 19 ↳ Congest windows set by Congest Control Algorithm
  - 20 Avoid Congestion in Network.
  - 21 ↳ we suffer data rate by sending pcks. and receiving acks.

~~at~~ Ideas:

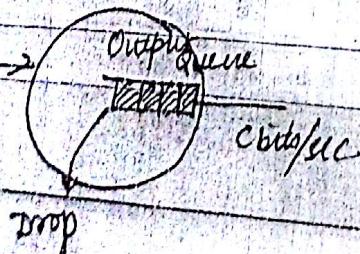
02 Sunday

Observe data plots.



Infer data rate from RTT.

RTT.



CW  
RTT

~~CW RTT > C ? Bottleneck Speed? MES~~

## MESSAGES

01 SATURDAY  
OCTOBER

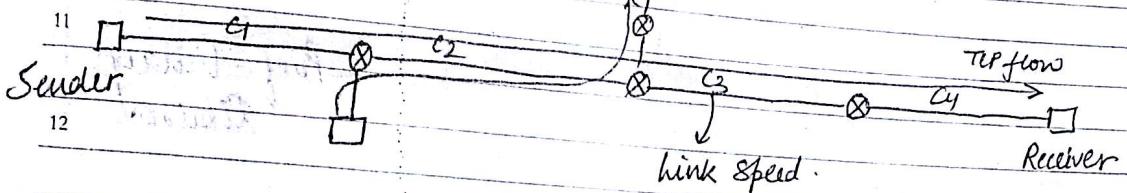
## TCP Congestion Control

Window,  $W = \min(\text{cong. window (cw)}, \text{Adv. window})$

→ Congest. Control.

Adv. window

↳ Flow Control



$\min_i c_i = ?$

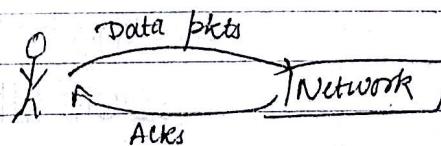
Want?

- 1 Use available bandwidth efficiently
- 2 Must be "fair" to other TCP flows
- 3 "Learn" Correct Data Rate.
- 4 ↳ Congest. window set by Congest. Control Algorithm
- 5 ↳ Avoid Congestion in Network.
- 6 ↳ Use proper data rate by sending ppts. and receiving acks.

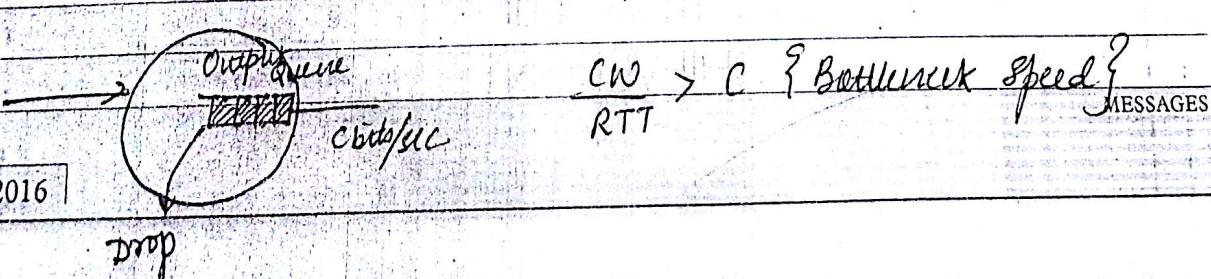
# Ideas:

02 Sunday

Observe data ppts.



Infer data rate from RTT.



October'16						
M	T	W	T	F	S	S
31			1	2		
1	2	3	4	5	6	7
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

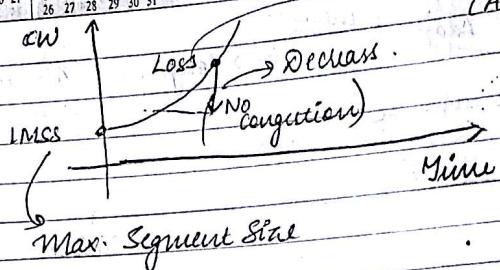
November'16						
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

December'16						
M	T	W	T	F	S	S
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

MONDAY  
OCTOBER

03

(Assumed due to congestion)

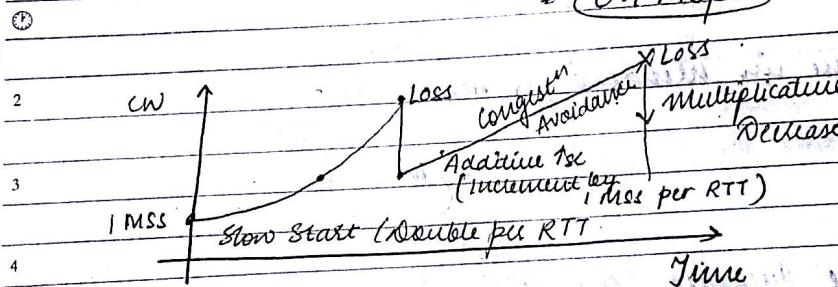


$$RTT = 120 \text{ ms}$$

$$MSS = 12000 \text{ bits}$$

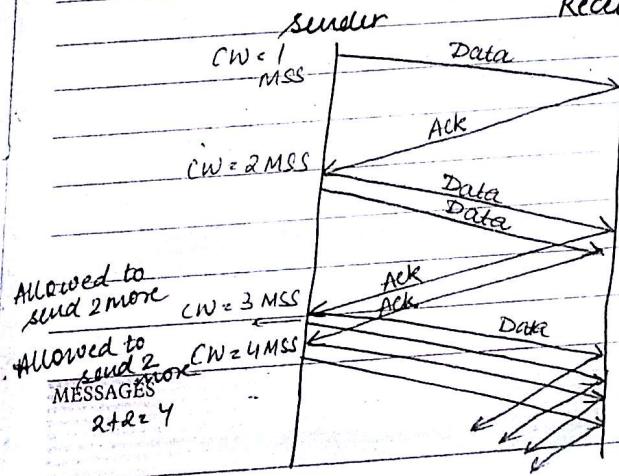
$$\frac{MSS}{RTT} = \frac{12 \times 10^3}{120 \times 10^{-3}} = 105 \text{ bits/sec.}$$

$$= 0.1 \text{ Mbps}$$



### # Slow Start

CW = 1 MSS initialised



On receiving every ACK, at sender

$$CW = CW + 1 MSS$$

Provided

$$CW < SS\text{-Threshold.}$$

↓  
Slow Start Threshold  
2016

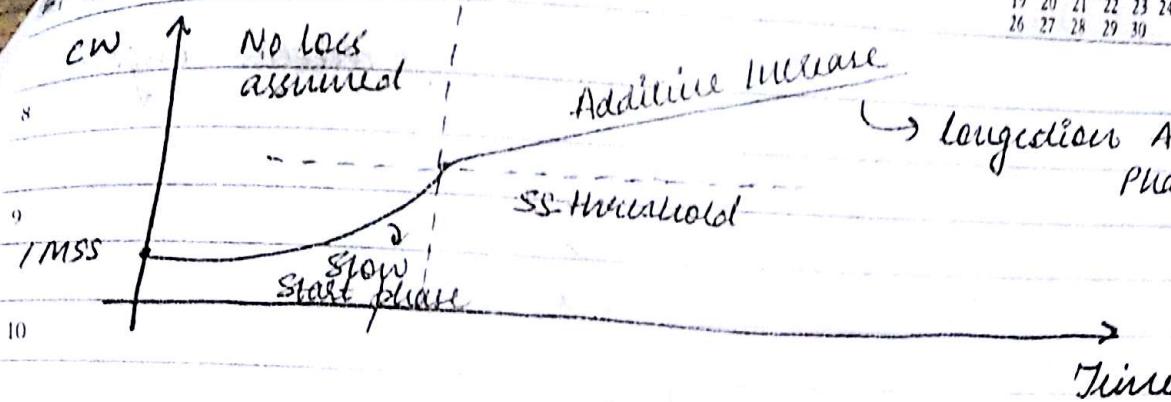
MESSAGES

NOVEMBER DECEMBER

04

TUESDAY

OCTOBER



# Additive Increase: Want to increase by 1 MSS per RTT

# ACKs per RTT  $\approx \frac{CW}{1 MSS}$  (# of pkts per RTT).

$$\begin{aligned} \text{Increase in window per Ack} &= \frac{1 MSS}{(CW/1 MSS)} \\ &= \frac{MSS \times MSS}{CW} \end{aligned}$$

# Additive Increase (Congestion Avoidance Phase)

On receiving ACK,

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

WEDNESDAY  
OCTOBER 05

October'16

M	T	W	T	F	S
31					1 2
3	4	5	6	7	8 9
10	11	12	13	14	15 16
17	18	19	20	21	22 23
24	25	26	27	28	29 30

November'16

M	T	W	T	F	S
1	2	3	4	5	6
8	9	10	11	12	13
15	16	17	18	19	20
22	23	24	25	26	27
29	30				

December'16

M	T	W	T	F	S
5	6	7	8	9	10 11
12	13	14	15	16	17 18
19	20	21	22	23	24 25
26	27	28	29	30	31

# TCP:

8

→ TAHOE

9

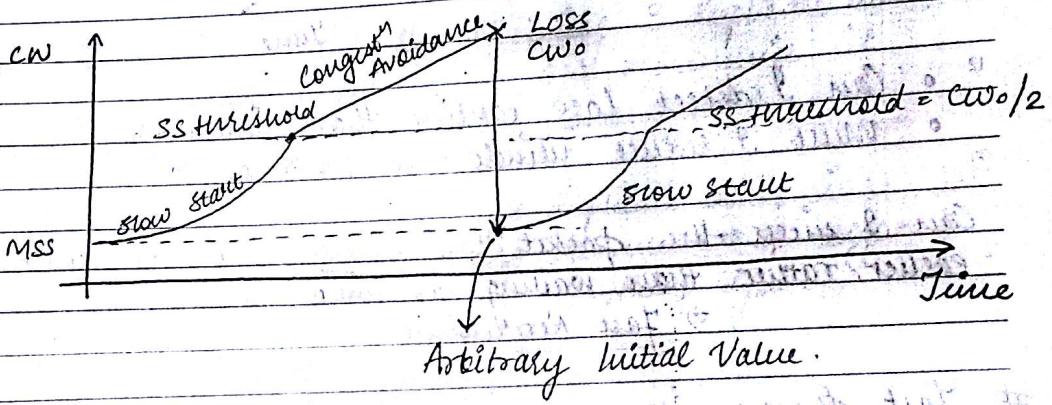
→ RENO

→ VEGAS

10

• TCP Tahoe

11



3

4

$$cw = \alpha \cdot cwo$$

5

6

MESSAGES

MESSAGES

2016

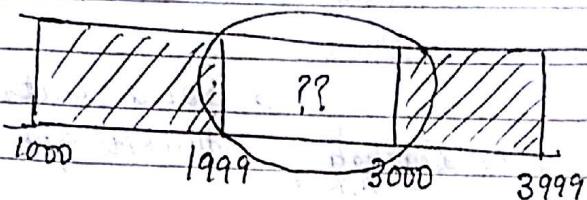
NOVEMBER DECEMBER

November'16							December'16						
M	T	W	T	F	S	S	M	T	W	T	F	S	S
1	2	3	4	5	6		1	2	3	4			
7	8	9	10	11	12	13	5	6	7	8	9	10	11
14	15	16	17	18	19	20	12	13	14	15	16	17	18
21	22	23	24	25	26	27	19	20	21	22	23	24	25
28	29	30					26	27	28	29	30	31	

FRIDAY OCTOBER 07

281-085 WK-40

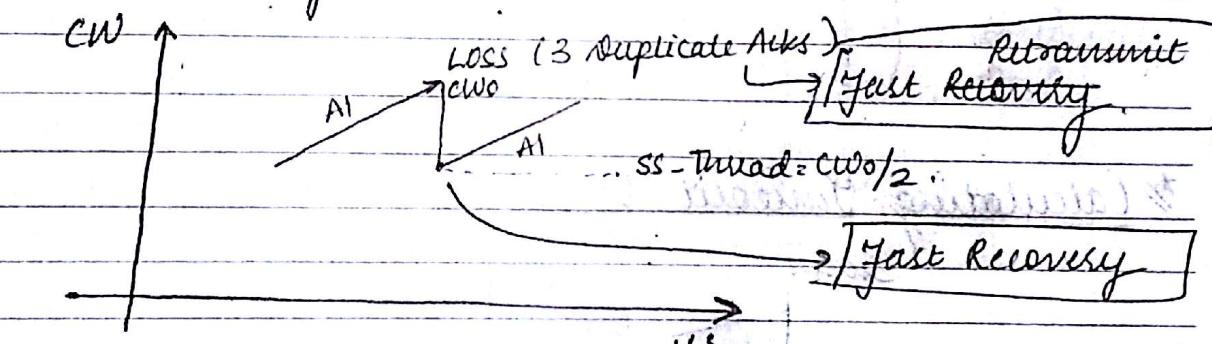
→ There might be many more losses after this, but we are sure about Seq. 2000 getting lost.



Note: Fast Retransmit: On receiving 3 duplicate acks, assume packet loss and retransmit it. (Camel Timer).

We don't do this after 1/2 duplicate acks, since it is possible that pkt reordering occurred. → If slight delay, no sense in resending everything again.

## # Fast Recovery



$$CW = \frac{CW_0}{2}$$

After 3 Duplicate Ack Loss

MESSAGES

2016

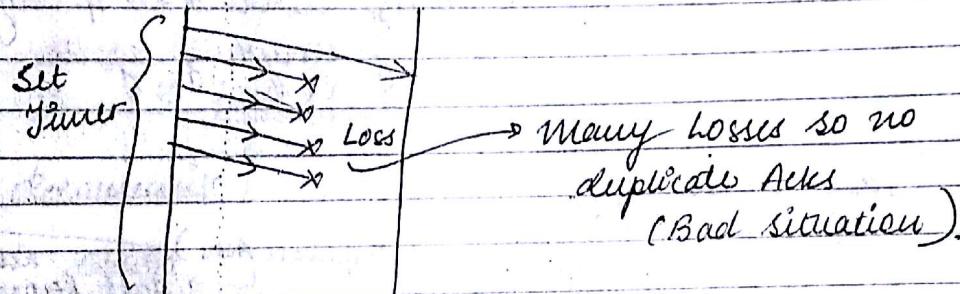
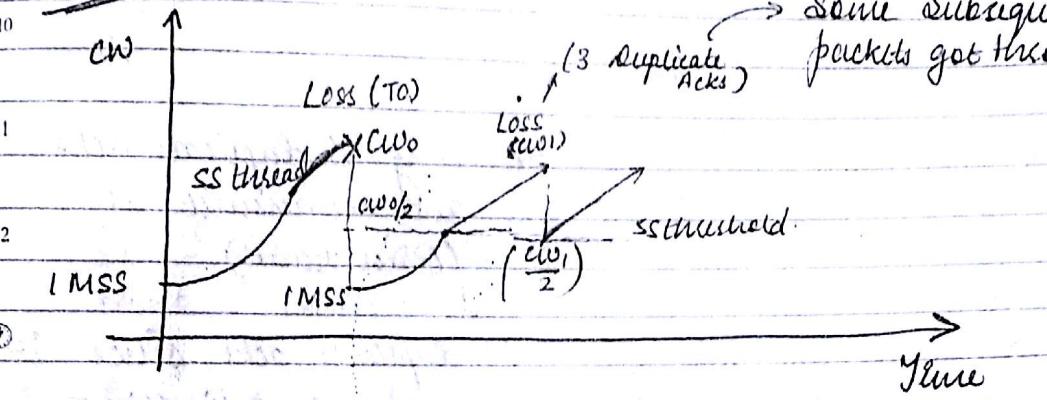


08 SATURDAY  
OCTOBER

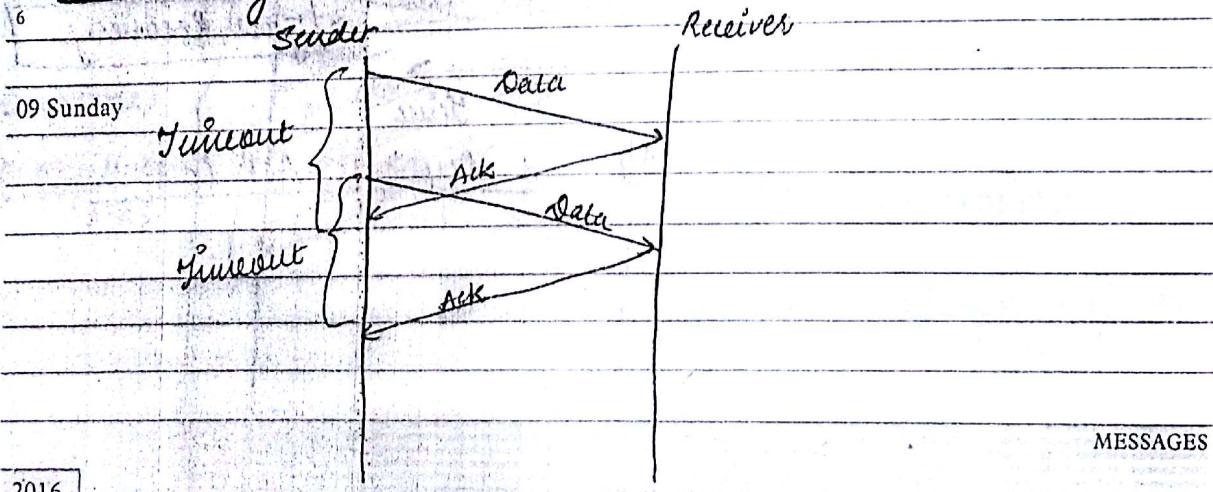
TCP Reno :

TCP Tahoe + TCP Retransmit + TCP Recovery.

TCP Reno



\* Calculating Timeout Value.



September'16							October'16							November'16							De
M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5
12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2
26	27	28	29	30			24	25	26	27	28	29	30								26

RT

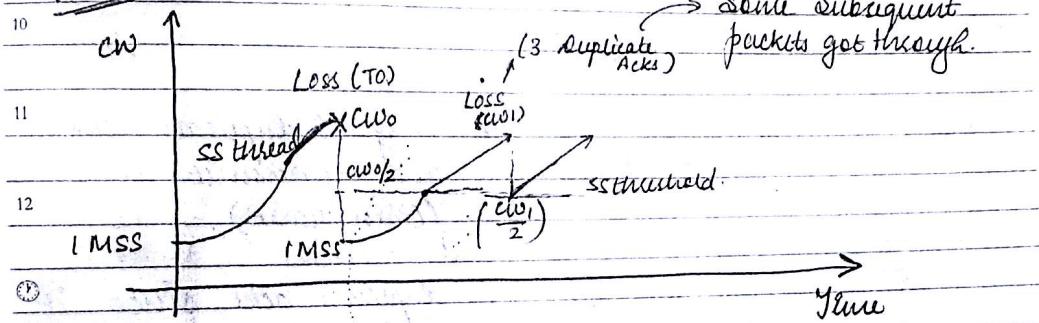
- 8
- 9
- 10
- 11
- 12
- 13
- 2 \* Sample
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- MESSAGES

**08** SATURDAY  
OCTOBER

YCP Reno:

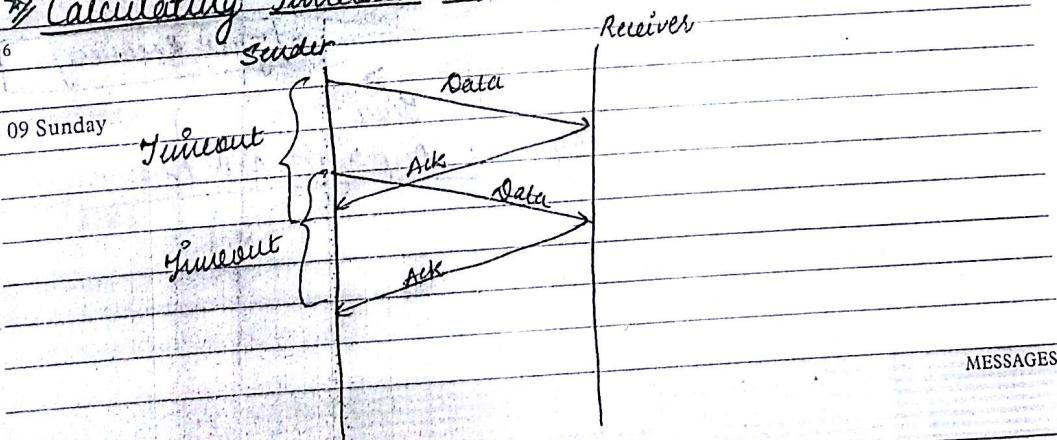
## TCP Tahoe + TCP Retransmit + TCP Recovery

TCP Reno



The diagram illustrates a sequence of events. On the left, a bracket labeled "Set Timer" groups several arrows pointing towards a vertical line. From this vertical line, multiple arrows point to the right, each labeled "Loss". A curved arrow points from the text "Many losses so no" to the "Loss" labels. Another curved arrow points from "duplicate ACKs" to the "Loss" labels. The final label "(Bad situation)" is at the bottom right.

## Calculating Present Value



2016

September '16					October '16				
M	T	W	T	F	S	M	T	W	F
			1	2	3	4			1
5	6	7	8	9	10	11	3	4	5
12	13	14	15	16	17	18	10	11	12
19	20	21	22	23	24	25	17	18	19
26	27	28	29	30			24	25	26

November'16							De
M	T	W	T	F	S	S	M
		1	2	3	4	5	6
7	8	9	10	11	12	13	5
14	15	16	17	18	19	20	12
21	22	23	24	25	26	27	19
28	29	30					26

RT

三

8

9

10

1

2 ~~\* Sample~~

3

4

5

6

1

4

100

1

36624816  
1992-93

1

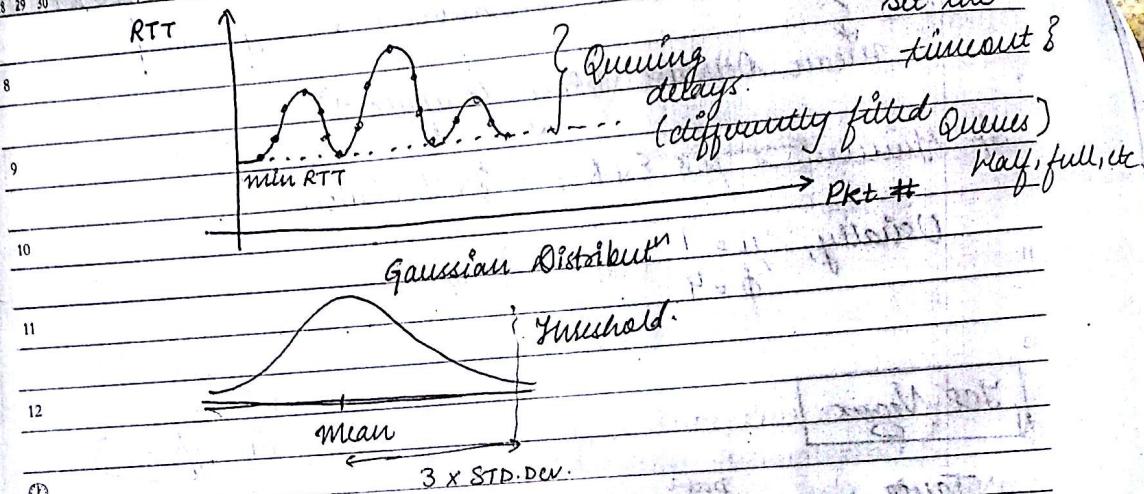
October'16						
M	T	W	T	F	S	S
31	1	2	3	4	5	6
1	2	3	4	5	6	7
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

November'16						
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

December'16						
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
21	22	23	24	25	26	27
26	27	28	29	30	31	

MONDAY  
OCTOBER 10

Where to  
set the



\* Samples of RTT  $x_1, x_2, x_3, \dots, x_N$

$$\bar{x} = \frac{1}{N} \sum x_i$$

$$\text{STD. Dev} = \sqrt{\frac{1}{N} \sum_i (\bar{x} - x_i)^2}$$

computationally  
involved.

Difficult to square root  
everytime

Instead use:

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

\* TCP Algorithm for Timeout:

1) Have estimated (EST) RTT.

2) Sample RTT  $\rightarrow$  Just calculated RTT.

3) Difference = Sample RTT - EST RTT  $|x_i - \bar{x}|$

2016

MESSAGES

MESSAGES



1 TUESDAY  
OCTOBER

$$44 \text{ Deviation} = (1-\alpha) \text{ Deviations} + \alpha / \text{difference}$$

3 mean Deviatn (easy computatn)

$$55 \text{ Minout} = \mu \times ESLRTT + \phi \times \text{Deviation}$$

Usually,  $\mu = 1$   
 $\phi = 4$ .

12 TCP Vegas:

September'16							October'16						
W	T	W	T	F	S	S	M	T	W	T	F	S	S
5	6	7	8	9	10	11	31	1	2	3	4	5	6
12	13	14	15	16	17	18	3	4	5	6	7	8	9
19	20	21	22	23	24	25	10	11	12	13	14	15	16
26	27	28	29	30			17	18	19	20	21	22	23
							24	25	26	27	28	29	30

November'16							Dec
M	T	W	T	F	S	S	M
1	2	3	4	5	6		
7	8	9	10	11	12	13	5
14	15	16	17	18	19	20	12
21	22	23	24	25	26	27	19
28	29	30					26

July	
S	M
12	4
19	11
26	18
	25

8 RM

10 LM

10 —

11 Failure

12 Q.1. A

Q.2. E

2 ots

3 Yatoo

Reno

4 TCP E

5 D

6 —

7 Mind

8 — R

9 — N

10 Lisa

11 M

12 —

13 —

14 —

15 —

16 —

17 —

18 —

19 —

20 —

21 —

22 —

23 —

24 —

25 —

26 —

27 —

28 —

29 —

30 —

31 —

1 —

2 —

3 —

4 —

5 —

6 —

7 —

8 —

9 —

10 —

11 —

12 —

13 —

14 —

15 —

16 —

17 —

18 —

19 —

20 —

21 —

22 —

23 —

24 —

25 —

26 —

27 —

28 —

29 —

30 —

31 —

1 —

2 —

3 —

4 —

5 —

6 —

7 —

8 —

9 —

10 —

11 —

12 —

13 —

14 —

15 —

16 —

17 —

18 —

19 —

20 —

21 —

22 —

23 —

24 —

25 —

26 —

27 —

28 —

29 —

30 —

31 —

1 —

2 —

3 —

4 —

5 —

6 —

7 —

8 —

9 —

10 —

11 —

12 —

13 —

14 —

15 —

16 —

17 —

18 —

19 —

20 —

21 —

22 —

23 —

24 —

25 —

26 —

27 —

28 —

29 —

30 —

31 —

1 —

2 —

3 —

4 —

5 —

6 —

7 —

8 —

9 —

10 —

11 —

12 —

13 —

14 —

15 —

16 —

17 —

18 —

19 —

20 —

21 —

22 —

23 —

24 —

25 —

26 —

27 —

28 —

29 —

30 —

31 —

1 —

2 —

3 —

4 —

5 —

6 —

7 —

8 —

9 —

10 —

11 —

12 —

13 —

14 —

15 —

16 —

17 —

18 —

19 —

20 —

21 —

22 —

23 —

24 —

25 —

26 —

27 —

28 —

29 —

30 —

31 —

1 —

2 —

3 —

4 —

5 —

6 —

7 —

8 —

9 —

10 —

11 —

12 —

13 —

14 —

15 —

16 —

17 —

18 —

19 —

20 —

21 —

22 —

23 —

24 —

25 —

26 —

27 —

28 —

29 —

30 —

31 —

1 —

2 —

3 —

October'16						
M	T	W	T	F	S	S
31	1	2	3	4	5	6
1	2	3	4	5	6	7
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

November'16

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

December'16

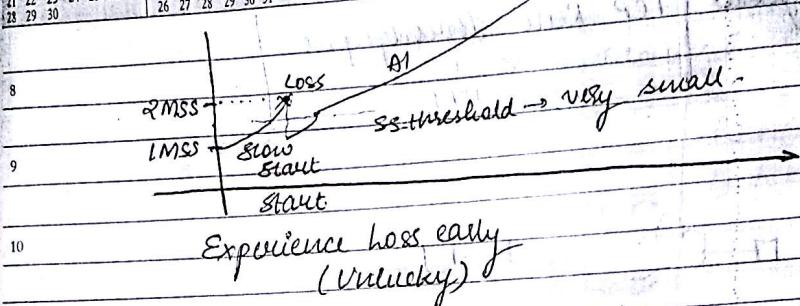
M	T	W	T	F	S	S
1	2	3	4	5	6	7
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

WEDNESDAY

OCTOBER

12

REG'D WK-A1

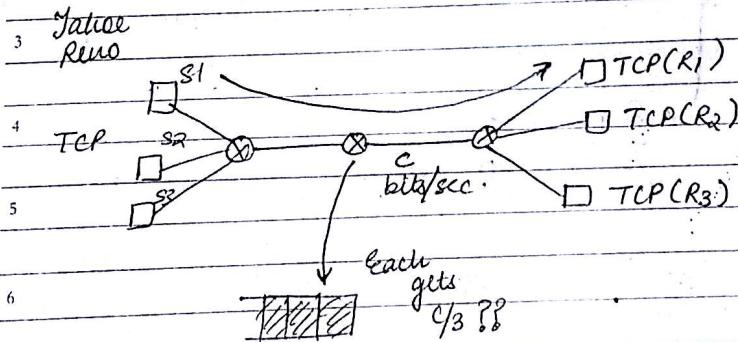


Fairness:

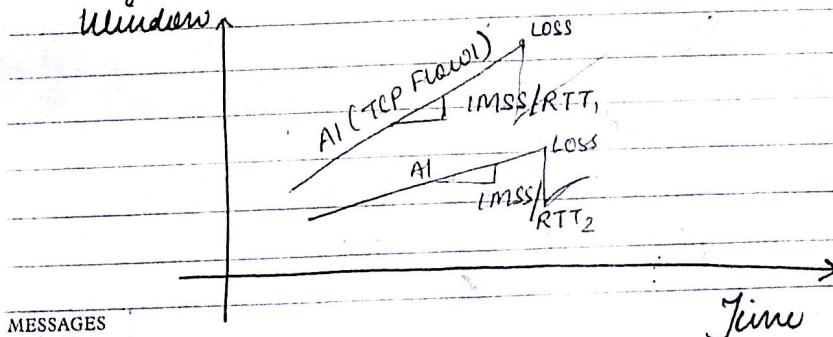
2016

- Q1: Are TCP flows of same type ( Tahoe, Reno, etc.) fair to each other? (Do they get same throughput?)
- Q2: Are TCP flows of different types fair to each other?

Queue  
Drop  
3 dup ack.



Window



2016

13 THURSDAY  
OCTOBER

RCP ~~Places~~

TCP Reno Throughput  $\propto \frac{1}{RTT}$

Vegas



RTT

Decrease Window

Queuing delay

Freeze window

RTT

Base RTT

September'16							October'16						
M	T	W	T	F	S	S	M	T	W	T	F	S	S
5	6	7	8	9	10	11	3	4	5	6	7	8	9
12	13	14	15	16	17	18	10	11	12	13	14	15	16
19	20	21	22	23	24	25	17	18	19	20	21	22	23
26	27	28	29	30			24	25	26	27	28	29	30

November'16							Dec	
M	T	W	T	F	S	S	M	
7	8	9	10	11	12	13	5	
14	15	16	17	18	19	20	12	
21	22	23	24	25	26	27	19	
28	29	30					26	

July  
1  
2  
3  
4  
11  
18  
25

clock

LL

10

11

Vegas

E

Expt

S

4

5

Ac

if

In cong  
avoidance  
phase,  
(cw > ss)

MESSAGES

MESSAGES

Sender

Data

Receiver

Acks

We are more concerned about congestion on data path than Acks path, we can measure one-way delay.

2016

October'16						
M	T	W	T	F	S	S
31				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

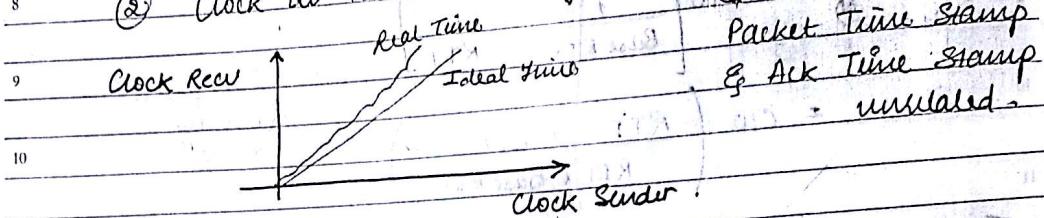
November'16						
M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

December'16						
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	
15	16	17	18	19	20	
19	20	21	22	23	24	25
26	27	28	29	30	31	

FRIDAY  
OCTOBER 14

- ① Data Rate higher than acks. ↗ relay one-way delay is not used?  
 ② Clock at receiver not synchronised ↘



### Vegas :

- ① Base RTT : Minimum RTT observed so far.  
 RTT : Current measured RTT.

$$\text{Expected Rate} = \frac{\text{longest window}}{\text{Base RTT}}$$

→ Data rate I would get if no congestion (no queuing delay).

$$\text{Actual Rate} = \frac{\text{longest window}}{\text{RTT}}$$

Actual Rate < Expected Rate

$$\text{Diff} = \text{Expected Rate} - \text{Actual Rate}$$

In congest<sup>n</sup> avoidance phase,  $\left\{ \begin{array}{l} \text{Diff} < \alpha \rightarrow \text{Additive increase like Reno} \\ \text{Diff} > \beta \rightarrow \text{Additive decrease of longest window} \\ (\text{cw} > \text{ss\_thresh}) \end{array} \right.$

$$\alpha < \text{Diff} < \beta \rightarrow \text{Don't change longest window}$$

2016

receivers

0

data

say

MESSAGES

NOVEMBER

DECEMBER

November'16						December'16					
M	T	W	T	F	S	M	T	W	T	F	S
1	2	3	4	5	6	5	6	7	8	9	10 11
7	8	9	10	11	12	13	12	13	14	15	16 17 18
14	15	16	17	18	19	20	19	20	21	22	23 24 25
21	22	23	24	25	26	27	26	27	28	29	30 31

MONDAY

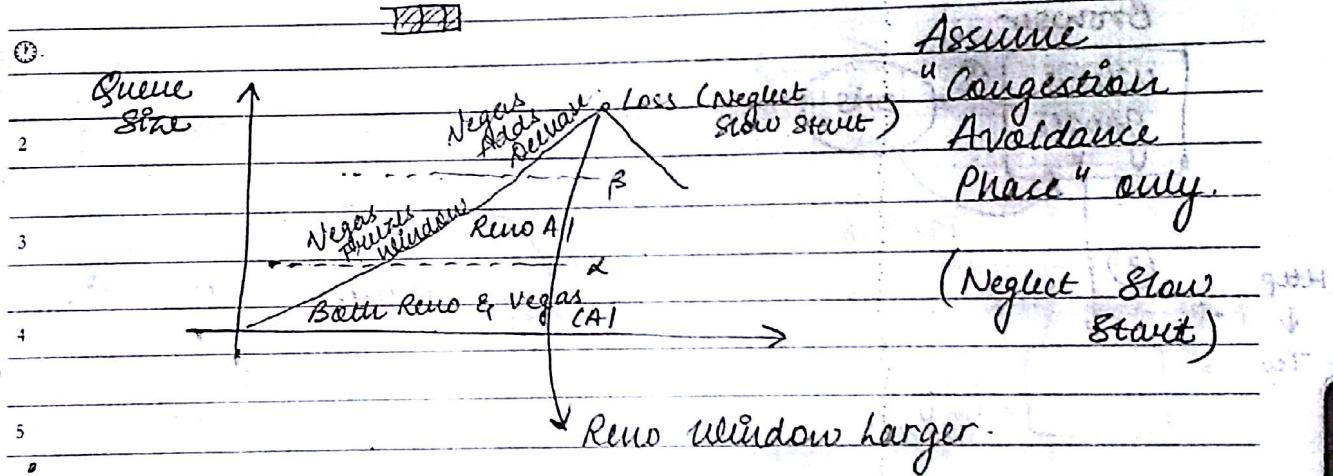
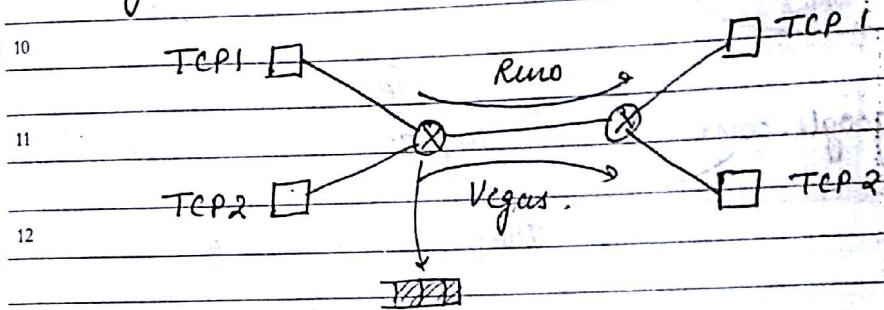
OCTOBER

17

2016-075 WK-42

Vegas couldn't succeed due to poor performance.  
No suitable values of  $\alpha, \beta$  were found.

### # Vegas v/s Reno



→ TCP Vegas is like a gentleman. Decreases the window size as soon as the queuing delay rises, while Reno keeps on.

MESSAGES

2016

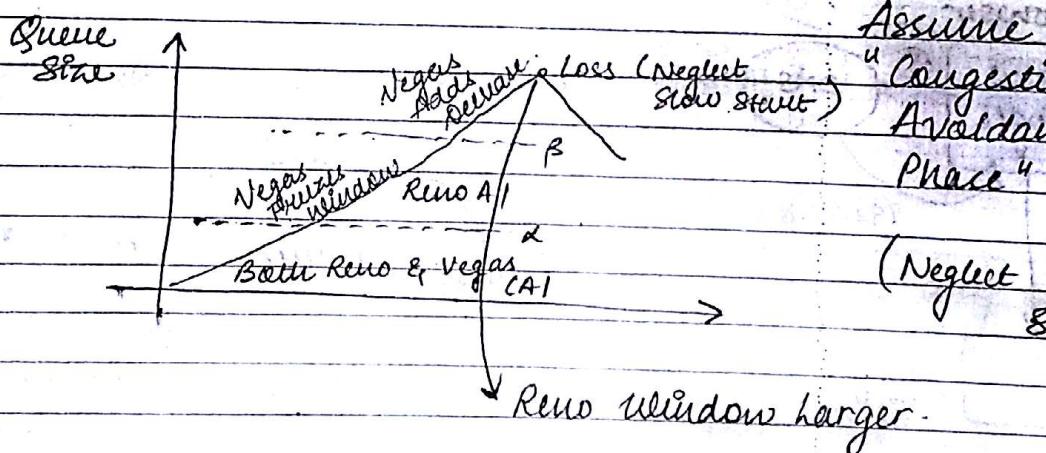
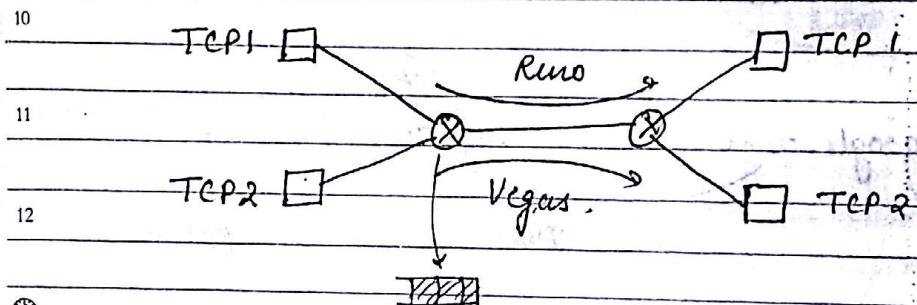
November'16							December'16						
M	T	W	T	F	S	S	M	T	W	T	F	S	S
1	2	3	4	5	6			1	2	3	4		
7	8	9	10	11	12	13	5	6	7	8	9	10	11
14	15	16	17	18	19	20	12	13	14	15	16	17	18
21	22	23	24	25	26	27	19	20	21	22	23	24	25
28	29	30					26	27	28	29	30	31	

MONDAY 17  
OCTOBER

291-075 Wk 42

Vegas couldn't succeed due to poor performance.  
No suitable good values of  $\alpha, \beta$  were found.

### # Vegas v/s Reno



→ TCP Vegas is like a gentleman. Decreases the window size as soon as the queuing delay rises, while Reno keeps on.

NOVEMBER  
DECEMBER

TUESDAY

OCTOBER

8 user

- Application → Port
- Transport → Protocol Number
- Network → IP address
- DLL → MAC address
- Physical

September'16

M	T	W	T	F	S	S
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		
					24	25

October'16

M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

November'16

M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

E	S	M
11	12	5
18	19	11
25	26	18
		25

July  
8 are  
le  
10 A.  
11

12 A.

13 E.

2

3

4 Cie  
play  
S)

5 → Yhe  
is  
the

6 Padv

# DNS

7 Isca

8 N

9 MESSAGES

10 MESSAGES

11 MESSAGES

12 MESSAGES

13 MESSAGES

14 MESSAGES

15 MESSAGES

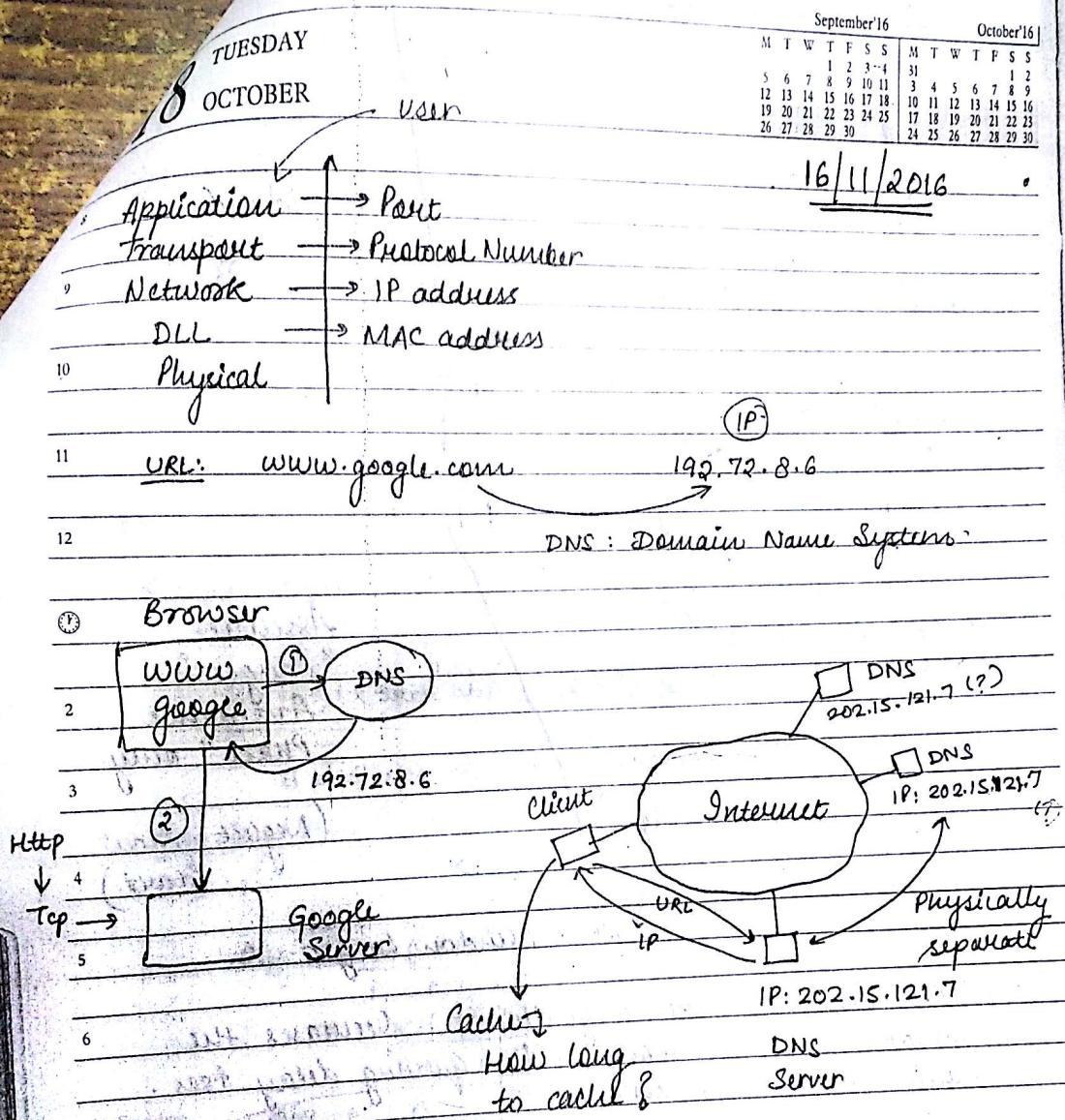
16 MESSAGES

17 MESSAGES

18 MESSAGES

19 MESSAGES

20 MESSAGES



2016

October'16						
M	T	W	T	F	S	S
31	1	2	3	4	5	6
2	3	4	5	6	7	8
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

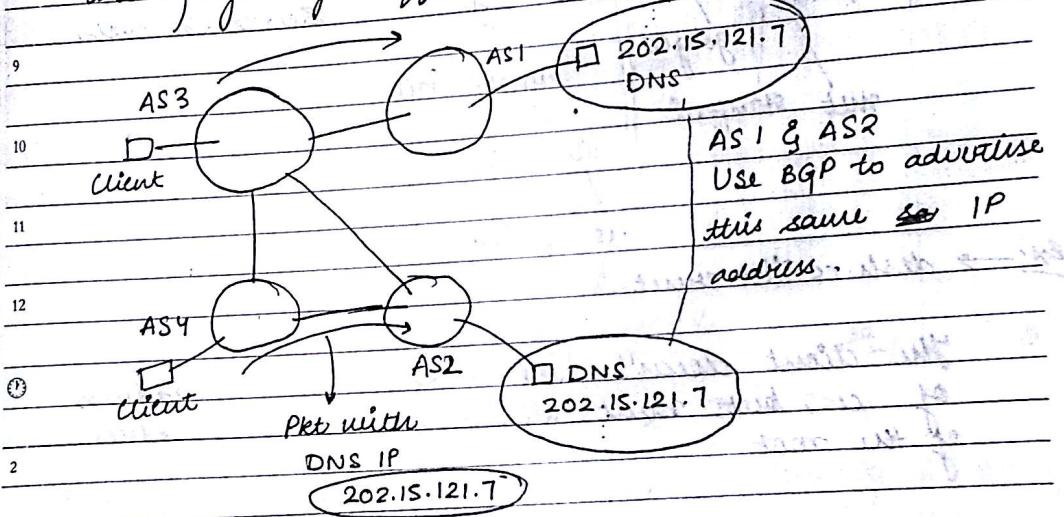
November'16						
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

December'16						
M	T	W	T	F	S	S
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

WEDNESDAY  
OCTOBER 19

2016-07-13 WK-42

They all use the same logical address but  
are physically different.



Clients send DNS packets which reach any one of physical servers (anycast).

→ This solves ①, ③ & ④ issues. But table size is still too large. What to do? As each one of them has to keep information of all others.  
⇒ Need a hierarchical structure for this.

### # DNS Hierarchy

Root  
Name  
Have DNS server for each domain.

NOVEMBER  
DECEMBER

MESSAGES

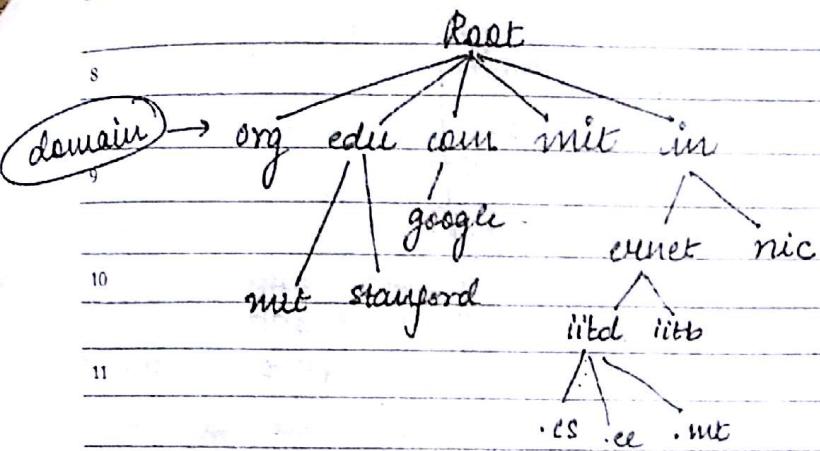
MESSAGES

2016

20 THURSDAY  
OCTOBER

September'16							October'16						
M	T	W	T	F	S	S	M	T	W	T	F	S	S
5	6	7	8	9	10	11	3	4	5	6	7	8	9
12	13	14	15	16	17	18	10	11	12	13	14	15	16
19	20	21	22	23	24	25	17	18	19	20	21	22	23
26	27	28	29	30			24	25	26	27	28	29	30

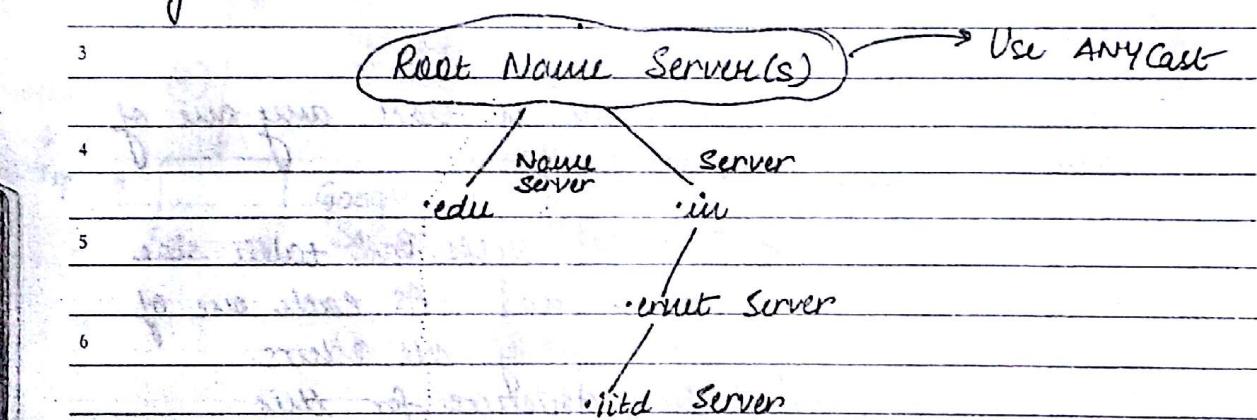
November'16													
M	T	W	T	F	S	S	M	T	W	T	F	S	S
7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30				



We have a DNS Server for each domain.

Ex:  $\rightarrow$  desh.cs.iitd.ernet.in

- ① The client doesn't need to know the IP address of .ce, but must atleast know the IP address of the root.



13 logical servers:

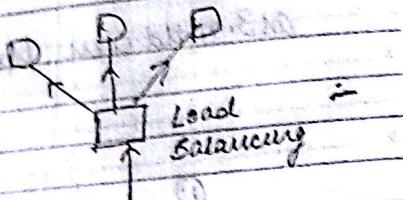
- A. root-server.net
- B. root-server.net

C. root-server.net  $\rightarrow$  One IP address, many ~~IP addresses~~ physical MESSAGES machines distributed in internet

December'18						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

November'18						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

FRIDAY  
OCTOBER 21



DNS cache

- 10 DNS → UDP
- 11    • Less overhead (No SYN/FIN)
- 12    • Faster

\* DNS server stores resource records.

- ① < Name, Value, Type, Class, TTL >
  - 2    +    +    +    +    \*
  - 3    Domain Info. we How to IN Validity in secs.
  - 4    Name are interpret (Internet) from now.
  - 5    looking for value?
- ↳ 2<sup>32</sup>-1
- ↳ It's different for that of packet TTL (IP TTL), which decreases by 1 after every hop.

- ② Type:
- 6    'A' : IP address

- 7    'NS' : Name of host running DNS server in domain corresponding to "Name" field.

'CNAME' : Canonical Name (Alias).

- 8    'MX' : Name of host running email server for domain specified in "Name" field.

Root has : < edu, as3.nstld.com, NS >  
DNS server for Edu.

< as3.nstld.com, 188.112.93.1, A >

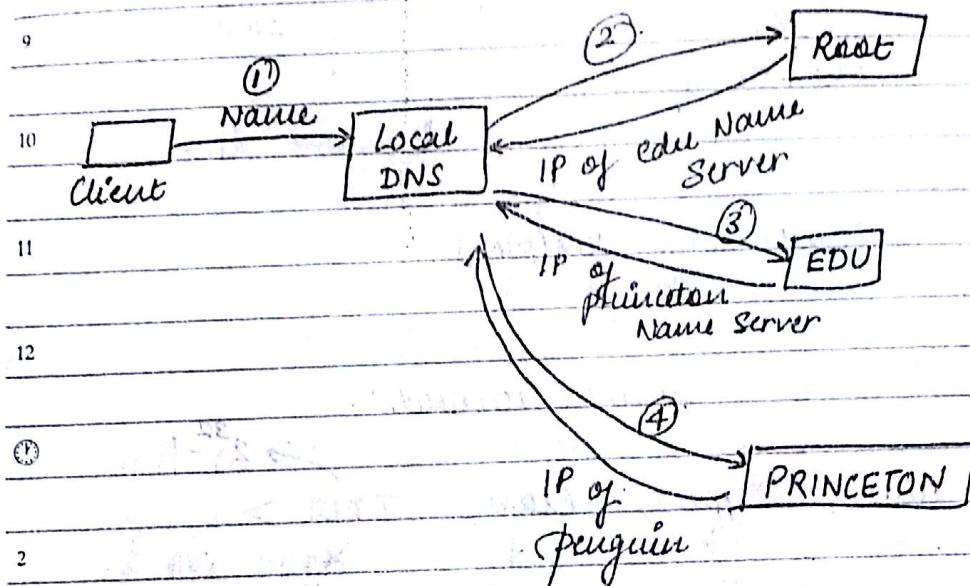
2016

many  
cal  
MESSAGES  
ited in  
net

22 SATURDAY  
OCTOBER

September '16							October '16					
M	T	W	T	F	S	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30				1	2	3	4	5	6
							7	8	9	10	11	12
							13	14	15	16	17	18
							19	20	21	22	23	24
							25	26	27	28	29	30

as 3. ns1ld.com has <princeton.edu, dns.princeton.edu, NS>  
<dns.princeton.edu, 112.191.7.16, A>



eg: ~~penguin.cs.~~ princeton.edu.

23 Sunday

2016

MESSAGES