

3 way handshaking (TCP follow করে)

↳ প্রথম Connection Establish করে then Response
বাটু যাবে then Request করে দ্বিতীয় handshaking

→ UDP টে Connection Establish এন্টার প্রয়োজন নাই গৈরি দ্বিতীয়—
No handshaking. • packets will be independent গৈরি unordered
(বিয়ালে No RTT delay) way টে মাবে।

↳ Advantage

→ অনেকোনো Delay নাই গৈরি transmission speed দ্বিতীয় গৈরি উচ্চতাপিত
পার্শ্বে থাবে।

→ No Congestion Control (UDP টে অত্যুক্তি-তে data পার্শ্বে
পার্শ্বে doesn't matter packet
যাবু স্টোর delivery fast চিকিৎসা যাছে নাকি।

disadvantages

→ packet loss
→ delivered out of order.

* file transfer G HDI TCP

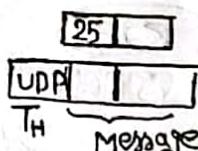
* Live streaming, zoom ইত্ব G UDP

SMTP
Client I

Application
Transport
Internet
Data Link
Physical

SMTP
Server

Mail Application



Transport

Internet

Multiplexing

Data link
Physical

সেভার মিশে দিয়ে মাইল মেজেটি সোকেট (25) এর ট্র্যান্সপোর্ট লেয়ারে গোচো, কোরি প্রক্রিয়াটি মিলেক্সিং। এবং প্র্যাপুর ট্র্যান্সপোর্ট লেয়ার হেডার মধ্যে TCP/UDP বিট মাঝে রয়েছে, কোরি সেগমেন্ট।

কোরি সেগমেন্ট বি থাকে সোর্স পোর্ট নো. কি ডেস্টিনেশন পোর্ট নো.

UDP Segment format:

header size
+
Msg size
No. of Bytes
of the
segment

Source port	Destination port
Length	Checksum
MSG / Data	

এখন detect করাব চলো
Original এর আওতা কিউ বিট
Extra পাণিলো কৃত থার্ট বিট
Error ক্ষেত্ৰে তা detect কৰা
যাব,
detect errors while
transmission

Error detect এবং স্বত্ত্বা ২ ধরণের math মোড়ে। একটা রকম Checksum.

- * find the checksum at sender and receiver end for the following sequence of bits. Analyze & Conclude whether the received message is corrupted or not

11010



110110

101010

110100

00111

নিম্ন কয়েক প্রেমিকে দ্বি ৪ টি আন বলা।

[Converting into decimal]

$$110110 = 54$$

$$101010 = 42$$

$$110100 = 52$$

$$+ 111100 = 60$$

$$\hline 11010000 = 208$$

decimal করানো করলে
 $(54 + 42 + 52 + 60) = 208$

২য় টাইল কয়েক একটা প্রয়োজন হচ্ছে কয়েক বিট Extra
সেগুন্ডী, (২ বিট) C2 msg প্রিয় 6 bit আছে summation 8 বিট।
then ৩য় summation এবং 6 bit বিট দ্বি গামে ২ বিট যোন
করুন।

$$\begin{array}{r} 010000 \\ \hline 11 \\ \hline 010011 \end{array}$$

আজ একটা স্বত্ত্বা Complement এসব।

1's complement of (010011) = 101100 checksum
for
sender.

Reciever এ রেজ আসলু কিম গোলু কিনা? পর্তি Check কো
জন্য checksum বাবে কৃব

$$\begin{array}{r}
 110110 \\
 101010 \\
 110100 \\
 \hline
 111100 \\
 \hline
 \frac{208}{+} = 010000 \quad [11 \text{ discarded}] \\
 \hline
 = 101100 \quad [\text{ব্যান্ডল আসলু} \\
 \text{11010000}] \\
 \hline
 \end{array}$$

checksum
sender (101100) ← 1's Compliment
কৃব

$$\begin{array}{r}
 \downarrow \\
 \text{1's compliment of } (111100) \\
 = \underline{\underline{000011}}
 \end{array}$$

[মুনোর আছু Cz
Error-free ইওয়া
মানি কৰ ০ রবে]

Sender:

1. Add all the messages
2. extra bit শুলো মান কৃব original bit শুলো গাখে
3. 1's Compliment কৃব (Extra থাকলো গোতা বাদ নিয়ে কৃব)

Reciever:

- 4) সব message bit গ্রেন
5. Extra bit বাদ নিয়ে - checksum এ গাখে মান কৃব,
6. 1's Compliment if এই bit 0 (zero) আজে এপ্যার
নাই।

2nd Method →

CRC (Cyclic Redundant Code)

→ It is used to detect

1. one bit errors

2. double bit

3. burst bit errors of length equal to the polynomial equation type.

$M(x)$ = Message bits

$R(x)$ = Redundant bits

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

Polynomial form ↑

*→ Always Count করতে হবে co-efficient কত?

Suppose, $x^4 + x^2 + 1$

$$1 \cdot x^4 + 0 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$$

10101 এই রূপান্তর message টির
দাপ্তরিক পরিকল্পনা

$$\begin{aligned} M(x) &= x^5 + x^4 + x^3 + 1 \\ &= 110011 \quad (6 \text{ bits}) \end{aligned} \quad \left\{ \text{টোক্স} \right.$$

$$\begin{aligned} R(x) &= x^4 + x^3 + 1 \quad [n=4] \\ &= 11001 \quad (5 \text{ bits}) \end{aligned}$$

$$\begin{array}{r} 11001) 1100110000 \\ \underline{-} \quad 11001 \\ \hline 10000 \end{array}$$

$$\begin{array}{r} 10000 \\ \underline{-} \quad 11001 \\ \hline 01001 \end{array}$$

বাকিয়ে zero values

Redundant message to be sent with original message

same bit = 0

other = 1

বাইনারিটি ঘন বুজন
X-OR operation of every bit

]

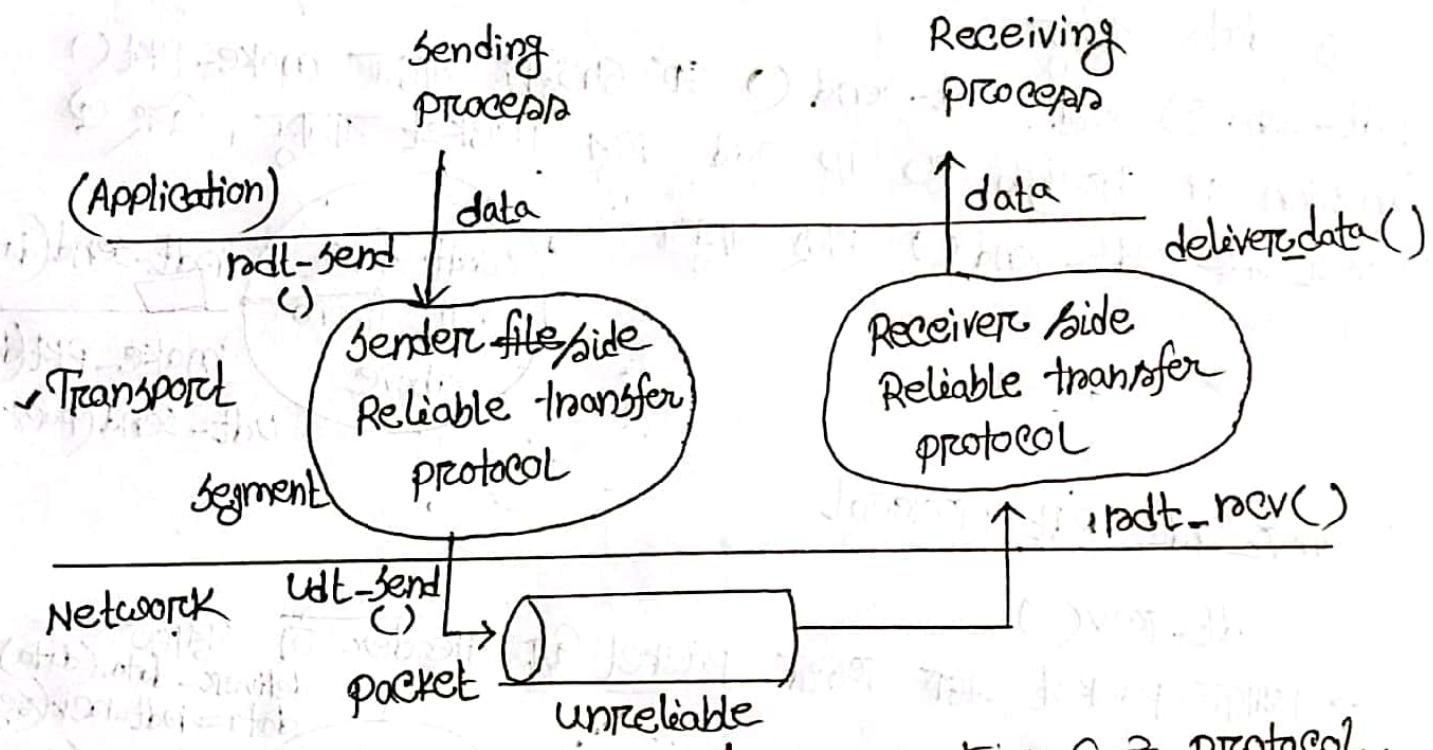
Receiver এতে কাণ্ডে যাবে Original message + Redundant bits (Sender)
 $M(x)$ + Calculated Redundant bits.

$M(x)$ Redundant
 [11001] message

$$\begin{array}{r}
 p(x) \\
 11001 \\
 \hline
 11001 \quad | \quad 1100111001 \\
 \hline
 11001 \\
 \hline
 11001 \\
 \hline
 00000
 \end{array}$$

[accepted]

Error-free



[rdt এতো protocol
rdt - send ও " " এর
function]

- Reliability provide করে কিন্তু protocol মাঝের বিলি reliable data transfer protocol. (RDT protocol)
- যাচাইয়া physical medium ৰ data layer হবে।
- Reliability নিকটে করবু বিভাগী way transport layer.
- From ip address add কোনো বলে packet
- reliability check করবু way messaging. আমি receiver কে message দিয়া বৈং আ গোচাক acknowledgement পাঠাবে।
গোচাক হুমকি না reliable
- Application layer এতো function.. Application layer এতো process
rdt - send() এতো function.
আই Call বাধবি এই function টো।
- Application layer এতো rdt - send() Call বাধবি এই segment
Create করু।

udt-send()

এটি unreliable বন্টে channel এর মাধ্যমে receiver এর

গোলুক data পাঠায়।

rdt-send() থেকে udt-send() কে চেঙ্গার গোলুক make-pkt() function দ্বাৰা through ip add কৰে packet বানায়, এবং এই packet দেখি udt-send() গোলুক ধার্য।

wait for
call from
above.

rdt-send(data)
make-pkt(data)
udt-send(packet)

Receiver side Rdt. protocol.

rdt-rcv()

→ Receiver packet থেকে ফেঁচে packet এর header টি মুছে
দেন।

wait for
call from
lower
layer

rdt-rcv(segment)
rdt-rcv(data)
data=Extract(packet)

segment=Extract(packet)

* Reliable data এ sender receiver
Data processor. দেখি পড়লাগ

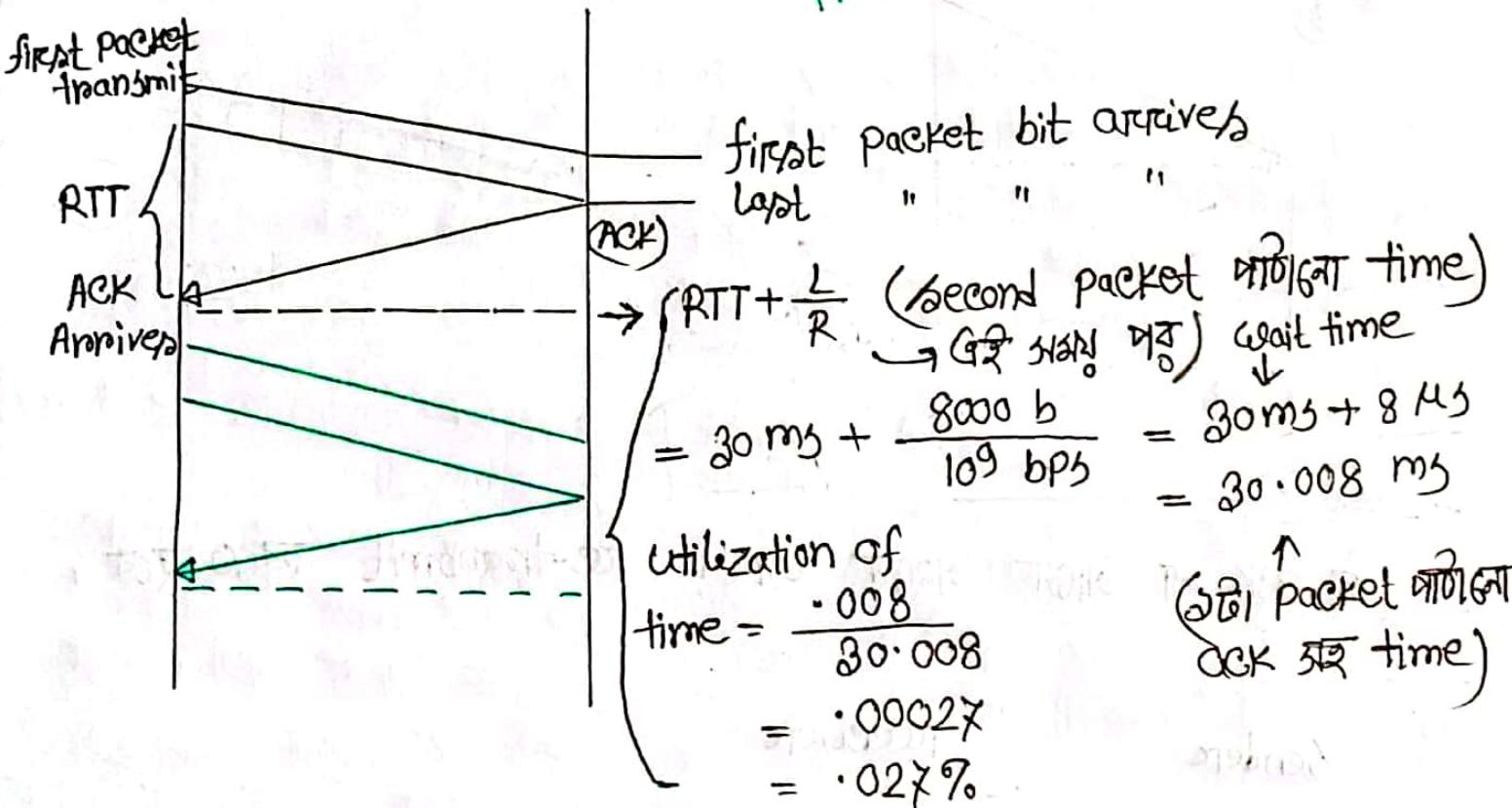
Stop-and-wait protocol

Reliability করিবলৈ Ensure করু দেই নিটো

Sender

Receiver

Non-pipeline (RTT version 2)



last packet টি মাঝার্থ পর্যন্ত ACK আসার সময়টি? RTT

প্রথম packet টি acknowledgement না আসা থেকে একটি packet পাঠিয়ে আসে, তারপর stop-wait protocol পাঠিয়ে আসে।

$$\text{Suppose, } R = 1 \text{ Gbps} = 10^9 \text{ bps}$$

$$L = 1000 \text{ B} = 8000 \text{ b}$$

$$RTT = 30 \text{ ms}$$

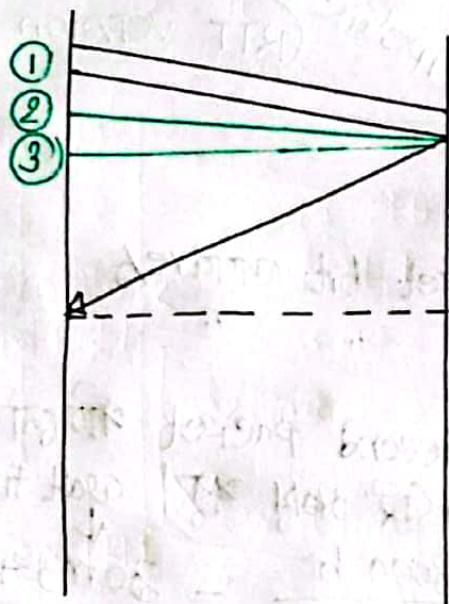
পাঠিয়ে pipeline করার জন্য updated version 2.1
 suppose 10টি packet করার
 যাই হবে,
 (10x utilization time)

pipeline-এ,

(nx utilization time)

$$\text{Utilization time} = \frac{n \times \frac{L}{R}}{\text{total time}}$$

pipeline kind of parallel



ধান্তি ack এবং গোচে বেরভেন্সে গোবৃত্ত টে-transmit করতে হবে।

sender

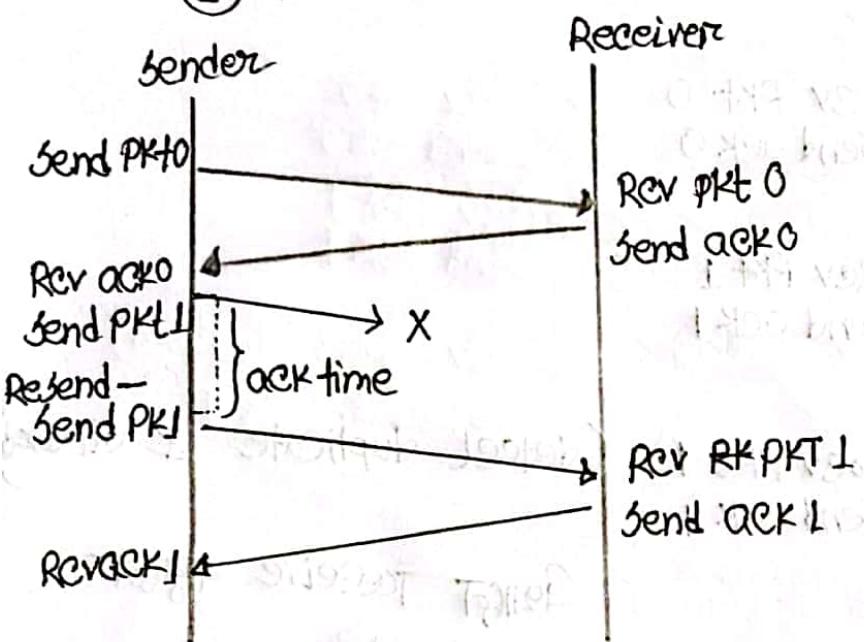
receiver

recv

ack()
recv pkt()
send ack()

Stop-watch protocol (Reliability action)

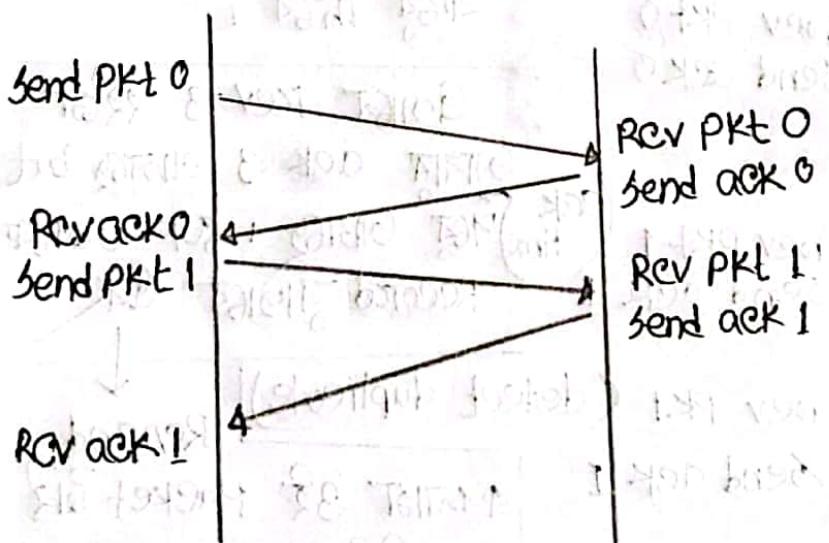
(b) packet loss



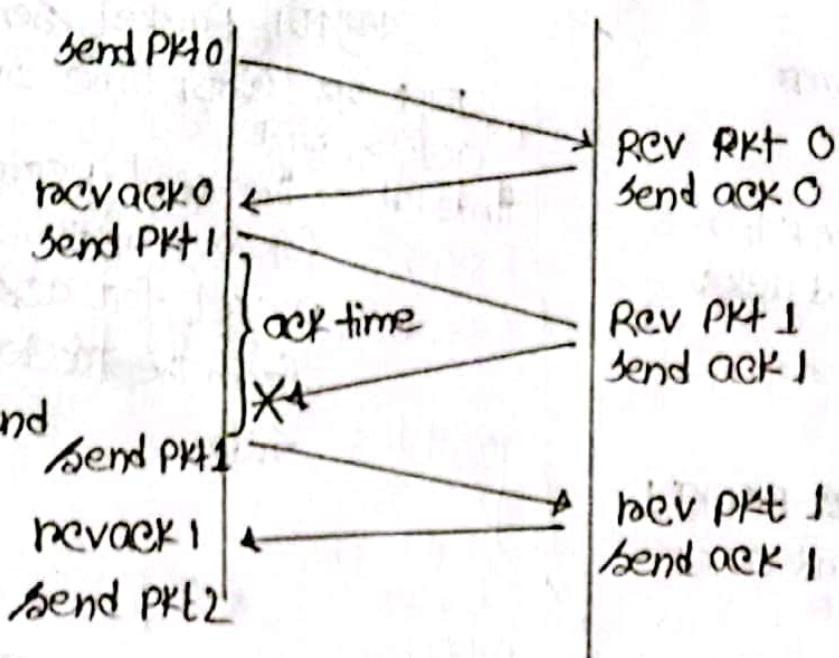
প্রয়োগটি Packet send করব পর কেটা time থাকবে
ACK পাইয়াও
timeout → the wait period
for a particular
packet for ACK
from the Receiver
side

ACK time এতে মন্তব্য করে না আজনো আর গোপনীয়
করব না 😊 . Resend করতে হবে ।

(a) No loss



(c) ACK loss



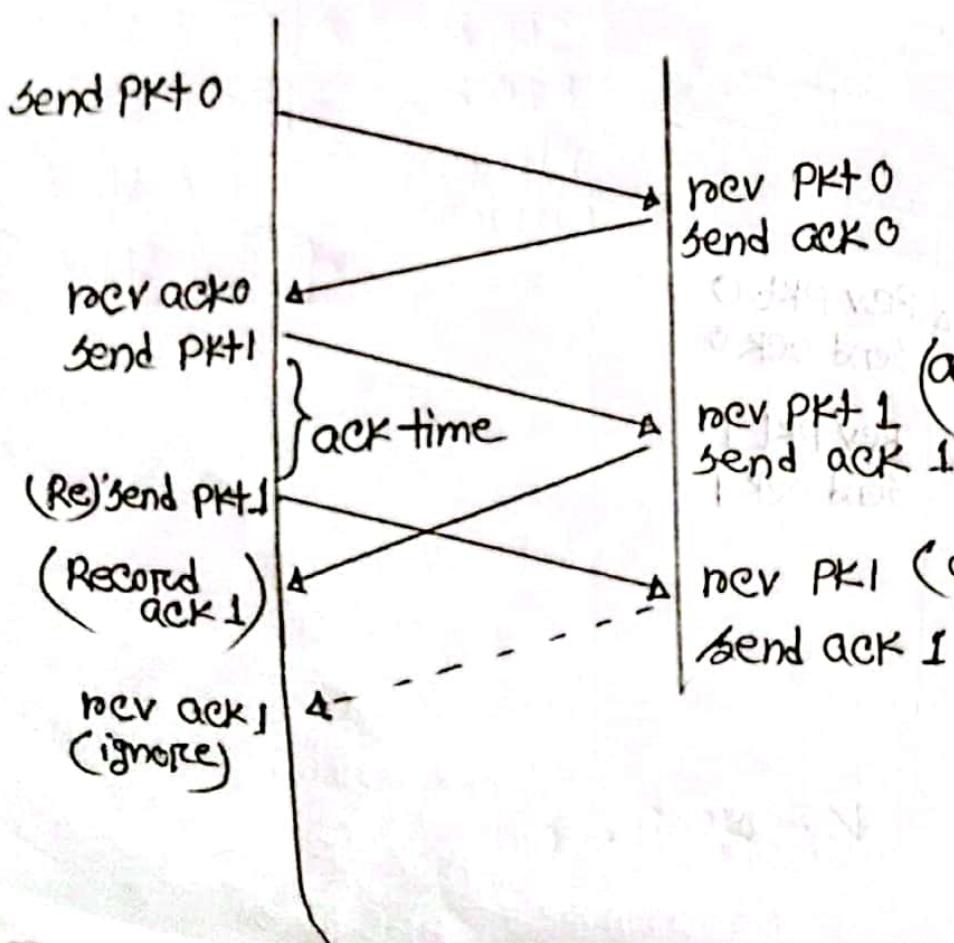
গোলনি তেক্ষণে বস্তুতে

but
ACK আসে নাই। এবং
যখন আবশ্য ট্রেন্ড
করুন গোলন তা শেষে
যে previously received
কর্তৃত্বের থই ক্ষেত্রে
কর্তৃত্ব দিবে।

গোলন TCV ৩ হবে
যোবার অক ৩ আসছে but
(ACK time) পর্যন্ত আসে নাই। তা উচিকে
Record রূপালীর ক্ষেত্রে

Resend
ক্ষমতা ওই packet তাই
যে উচিত সম্পর্ক যে
ACK আসবে ওভা
ignore করবে 😊।

(d) Delayed ACK



pipelined protocol.

It is a technique in which sender sends multiple packets without waiting for acknowledgements.

2. ~~CHAPTER~~ 1. GBN (Go-Back-N protocol)
2. SR (Selective Repeat protocol)

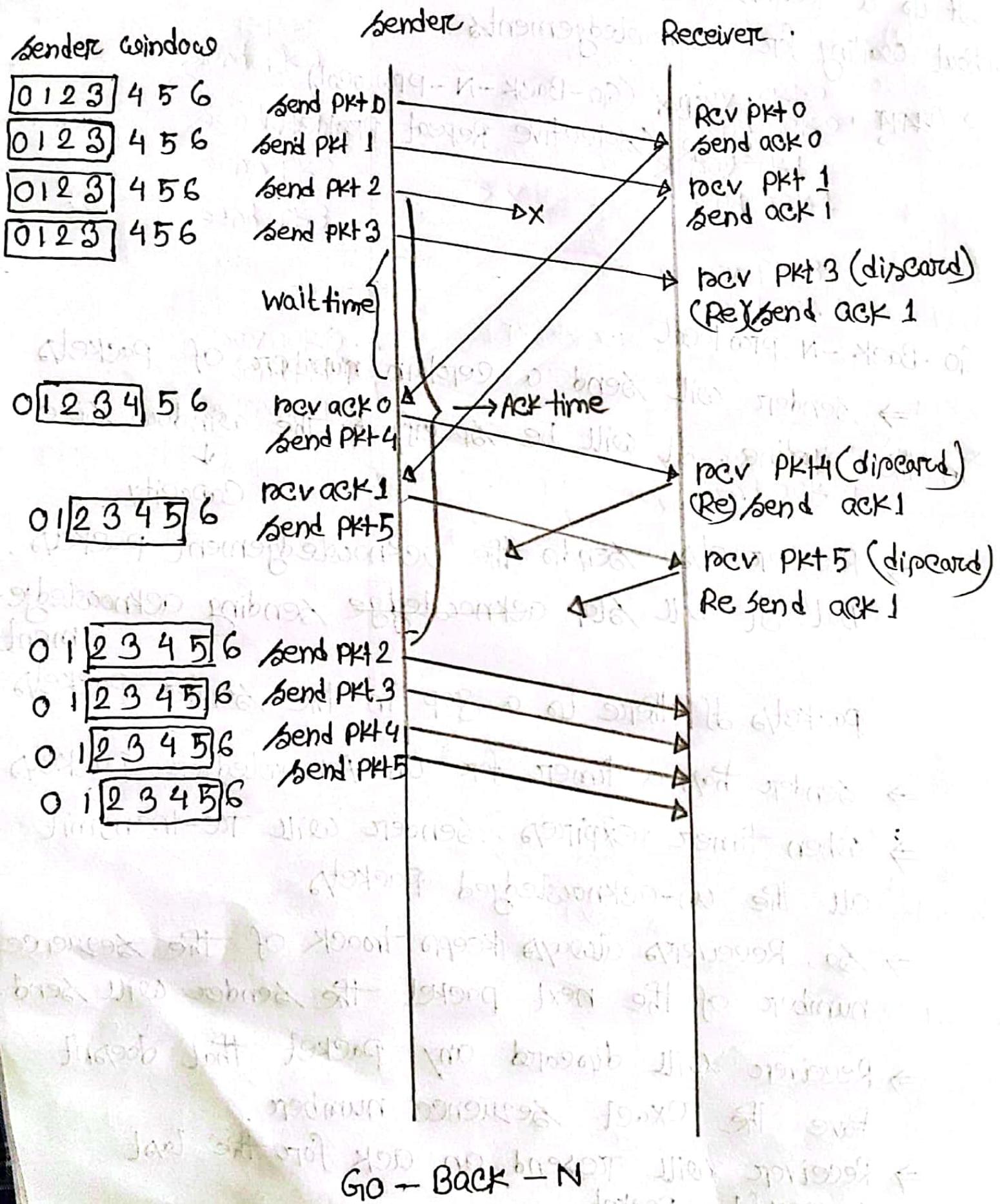
1. Go-Back-N protocol

→ Sender will send a certain number of packets in the pipeline. It will be specified by the window size
↓
Capacity.

→ Receiver also sends the acknowledgement packets. But it will stop acknowledging sending acknowledgement packets if there is a gap in the sender packets.

- Sender has a timer for un-acknowledged packets.
- When timer expires, Sender will re-transmit all the un-acknowledged packets.
- So, Receiver always keeps track of the sequence number of the next packet the sender will send.
- Receiver will discard any packet that doesn't have the exact sequence number.
- Receiver will resend an ACK for the last successful packet.

⇒ Once the sender has sent all its packets of the window, it will detect the lost ones.



থেকেন প্রথম first packet ACK না আসবে তাই wait time .
ACK থেকেন Tight shift করে window এবং ।

Go-N-Back receiver expect করে serially packet মাত্র ।
যে যদি কোটি packet না পাওয়া (PKT2) গুলো যে (like PKT3
পুরো পাওয়া packet কে transmit করতে এবং ACK কে
transmit করতে থাকে time out period পথান্ত ; এবং
মাত্র আজ এক packet ধূঁড়ে বাস্তু দিবে , time out
ক্ষেত্রে এখন check করবে last successful ACK খোন্তি হওয়া (PK1)
এবং উক্ত প্রেতের packet (PK2) থেকে অবাধ window
size transmit করবে ,

Selective Repeat

1. Sender can send a certain amount of packets in the pipeline based on window size .
2. Receiver sends individual ACK for each of the packet .
3. Sender maintains a timer for the un-ACK-ed packets .
4. When timer expires , it only retransmits the un-ACK-ed packets only .

* এখানে দেখাচ্ছি 4 টি পার্টির না , Only মিতি minor
গেটে মেটি - থাবি only যাবি time wait করতে ন হোচ্ছে ,

send PK 0

send PK 1

send PK 2

send PK 3

recv ACK 0

send PK 4

Re send PK 2

recv PK 0

send ACK 0

recv PK 1

send ACK 1

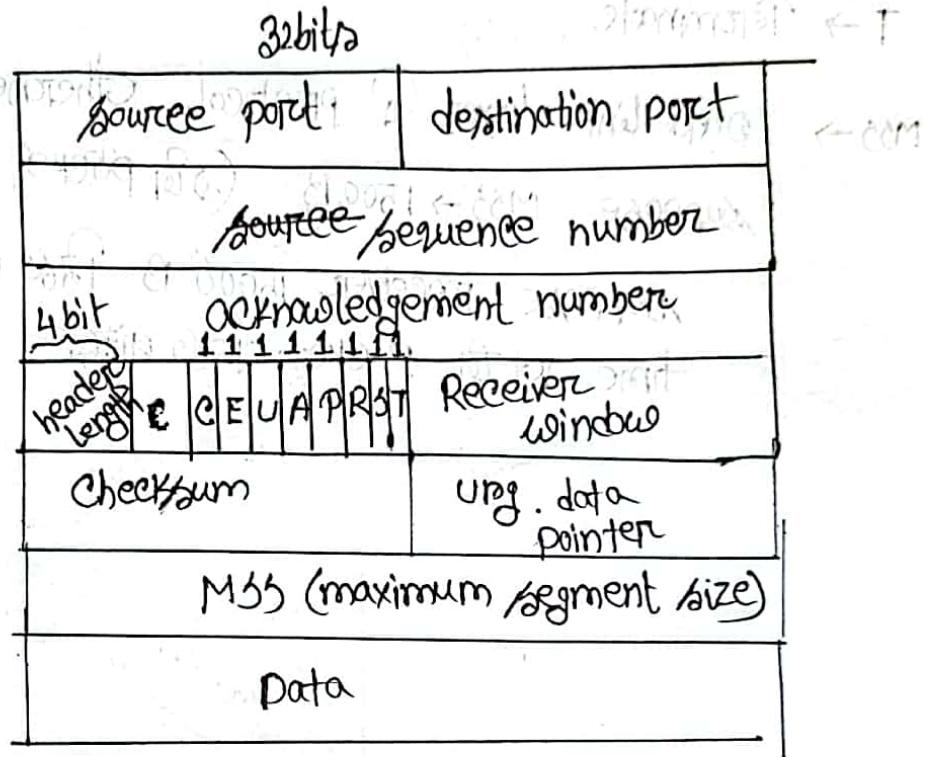
recv PK 3

send ACK 3

recv PK 4 (buffer)
send ACK 4

TCP Segment

Structure



Sequence number → একেবারে packet এর number ইয়ে থ্যু
যে byte পাঠিব তিনি randomly generate number
acknowledgement number → আগস্টা এর sequence আবার এর against G
ack-num generate কৃত
or packet
Receiver window → at a time Receiver করে byte data নিত
পার্শ্ব :

header length → মনে header এর length . Data এর ১০ header
যাবে গুরুত্ব length .

C,E → Congestion notification , 1 bit কল্পনা
Network G Congestion কেন্দ্ৰ নাকি ?

A → ACK bit

R → Reset

1 → SYN (সিনেক্ষন বাত্স)

ফোর্মেট দেখ

T → Terminate .

বেগনগুলি

MSS → Data link layer এ protocol ethernet .

Suppose MSS → 1500 B (এটা packet অর্থাৎ কত বাট্টি)

Suppose receiver 15000 B মিলে পাখতে দেবে at a

time 10 টি, packet. নিচে পাখতে

রেটেন্স 10.19.14.15.16.3.2.1.1

প্রথম 1500 B পাখতে

(যে ফর্মেট মানে) 1500

প্রথম 1500 B

তৃতীয় আরু তৃতীয় প্রদর্শন করে দেখো \leftarrow প্রদর্শন করে দেখো
প্রদর্শন করে দেখো \leftarrow প্রদর্শন করে দেখো
করে দেখো \leftarrow প্রদর্শন করে দেখো

চোখে আরু হাতে আপনার স্মিন্দে করে দেখো

বেগন করে দেখো করে দেখো \leftarrow বেগন করে দেখো
বেগন করে দেখো \leftarrow বেগন করে দেখো

বেগন করে দেখো \leftarrow বেগন করে দেখো

TCP 3 way handshaking

Client script TCP
Port 80 open.

SP = source port

DP = destination "

Seq = sequence no.

data -

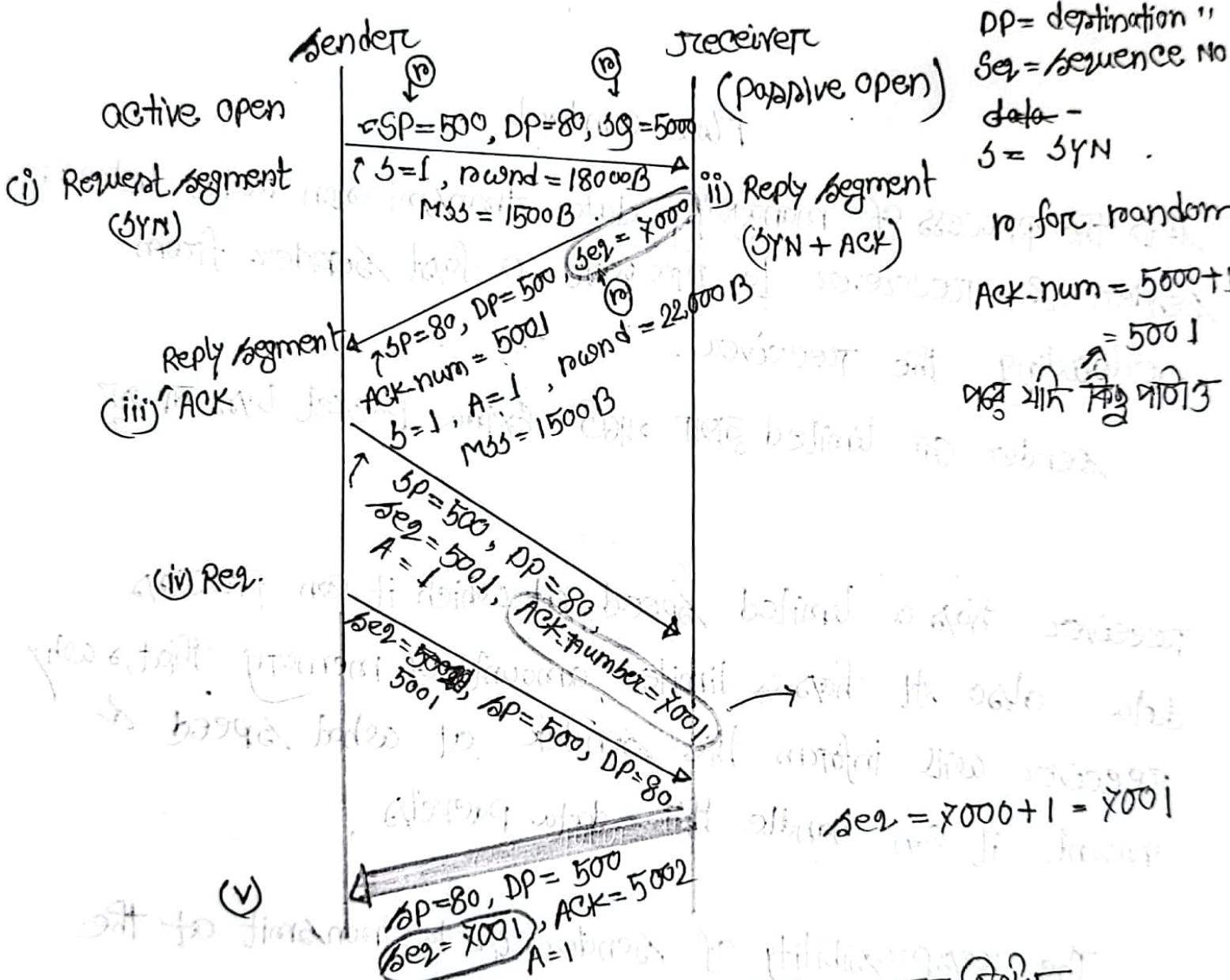
S = SYN.

for random

Ack-num = 5000 + 1

= 5001

প্রতি খানি স্টেট পাঠিয়ে



যখনই browser open করে কোম্পিউট গোড়াই থাকে তাৰে SP কৈল

port নম্বৰ হ'ব SP.

(Server request + ACK হ'ব)

i) Client SYN Req হ'ব

ii) Server SYN Req + ACK হ'ব Client ACK

iii) Client Server টো SYN ও এ হ'ব

Connection

Data यूज करी Request

(IV) + (V) के लिए

for eg. download - 70

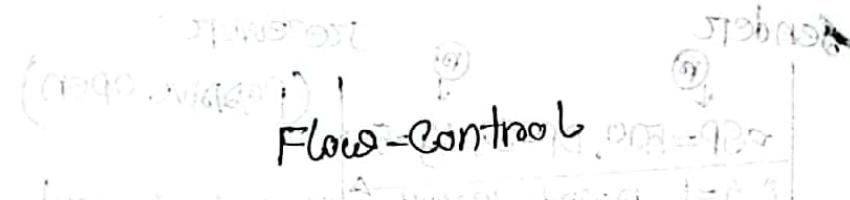
upload - 30

file download - 50

- upload

100% = 100

100% = 100



It is the process of managing data transmission rate between sender & receiver to prevent a fast sender from overloading the receiver.

sender का limited गति वाला Extra packet loss का क्षमा

receiver has a limited speed at which it can process data also it has a limited amount of memory that's why receiver will inform the sender at what speed & amount it can handle the data packets.

The responsibility of sender is to transmit at the agreed speed.

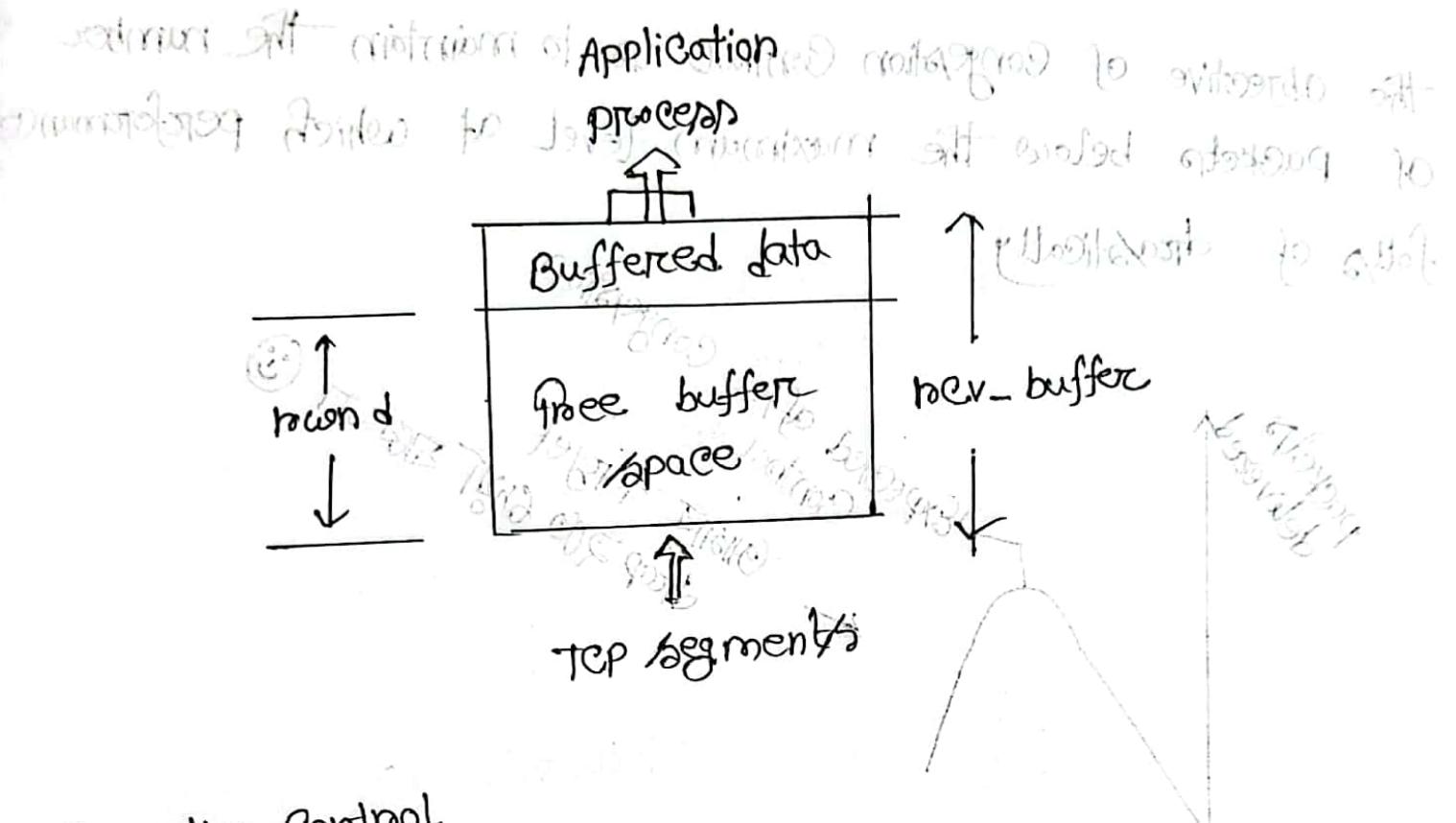
Flow Control always कर सकता है कि sender की गति

(idle time & forward pause)

to stop the data transmission

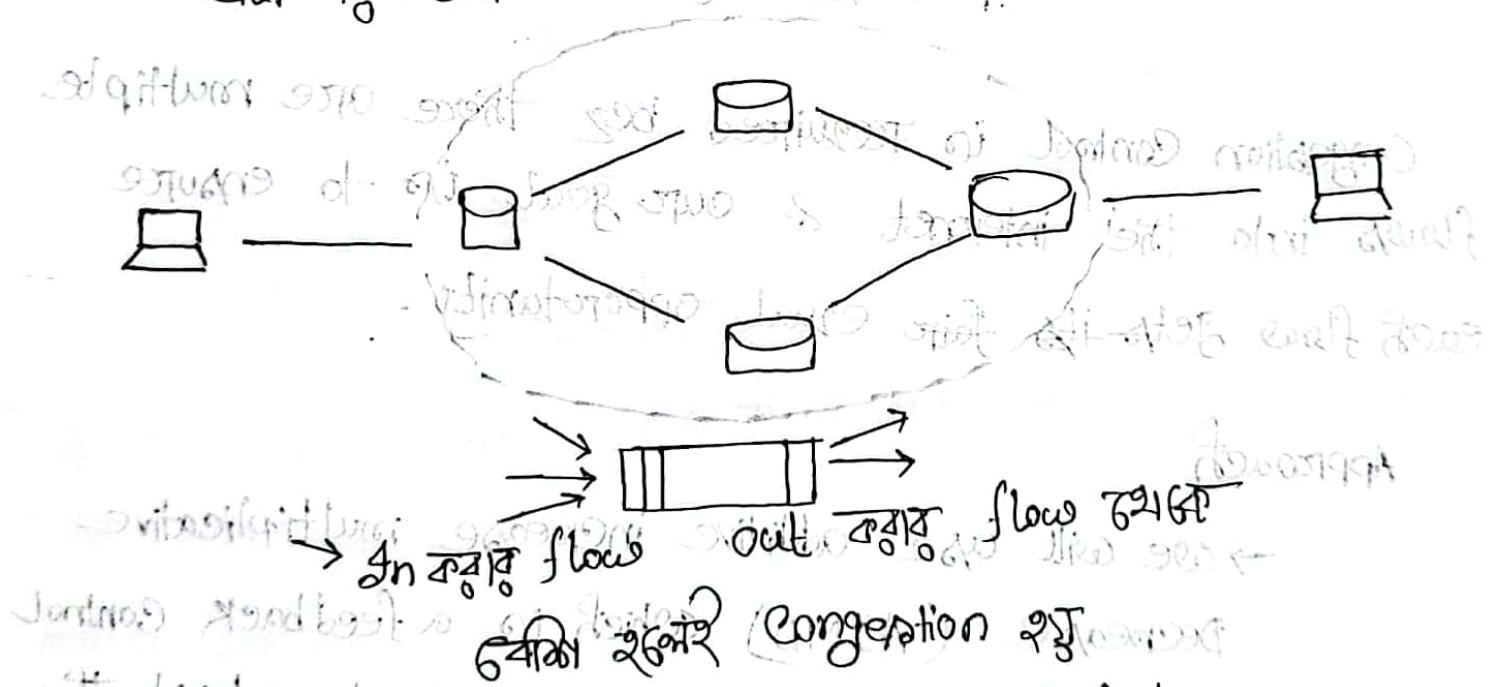
to hold the data transmission

to stop the data transmission



Congestion Control

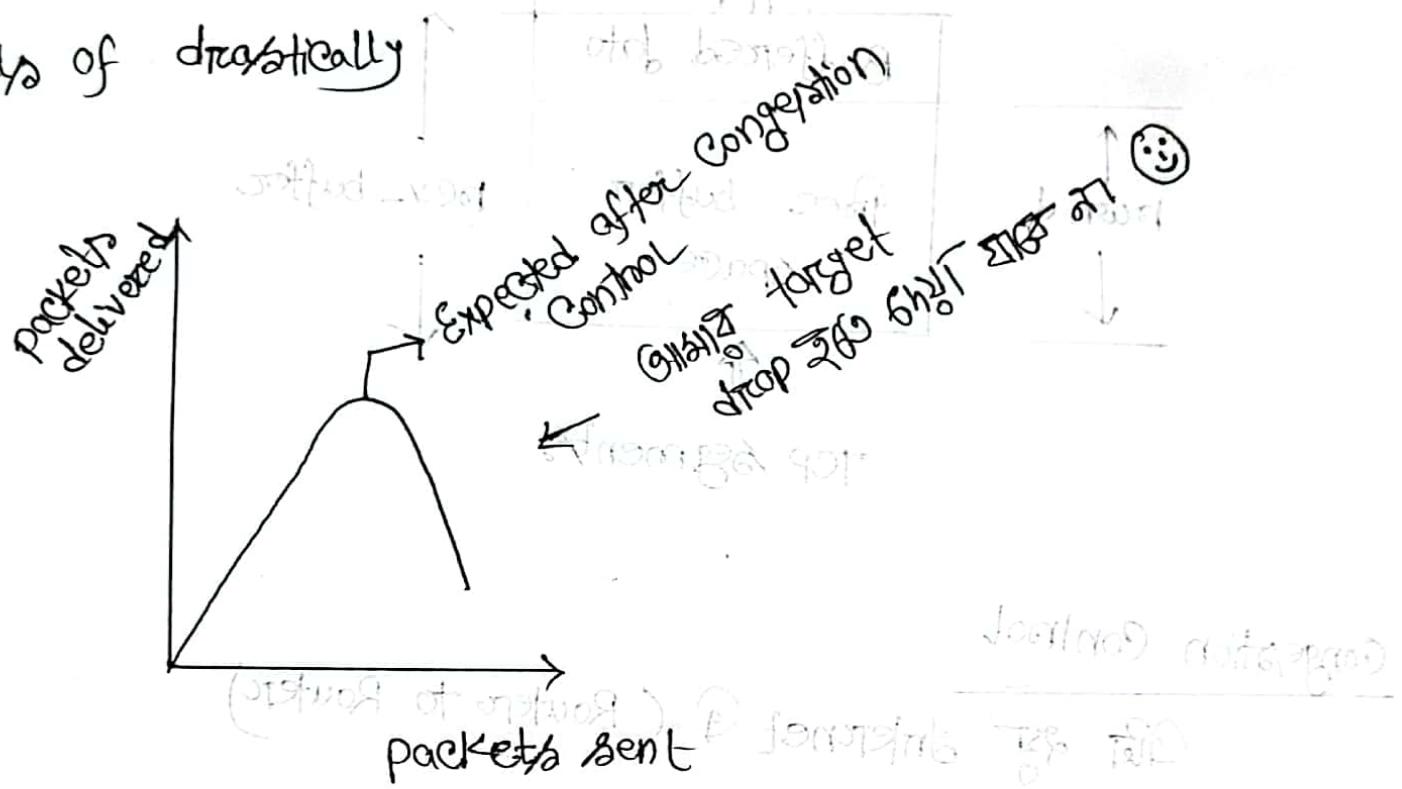
କେଣ୍ଟ କ୍ଷେତ୍ର ଇନ୍ଟର୍ନ୆ଟ (Routers to Router)



କେଣ୍ଟ କ୍ଷେତ୍ର (Congestion Control)
କେଣ୍ଟ କ୍ଷେତ୍ର କ୍ଷେତ୍ର (Congestion Control)

Congestion occurs when too many sources are sending too much data too fast for the network to handle

The objective of Congestion Control is to maintain the number of packets below the maximum level at which performance falls off drastically.



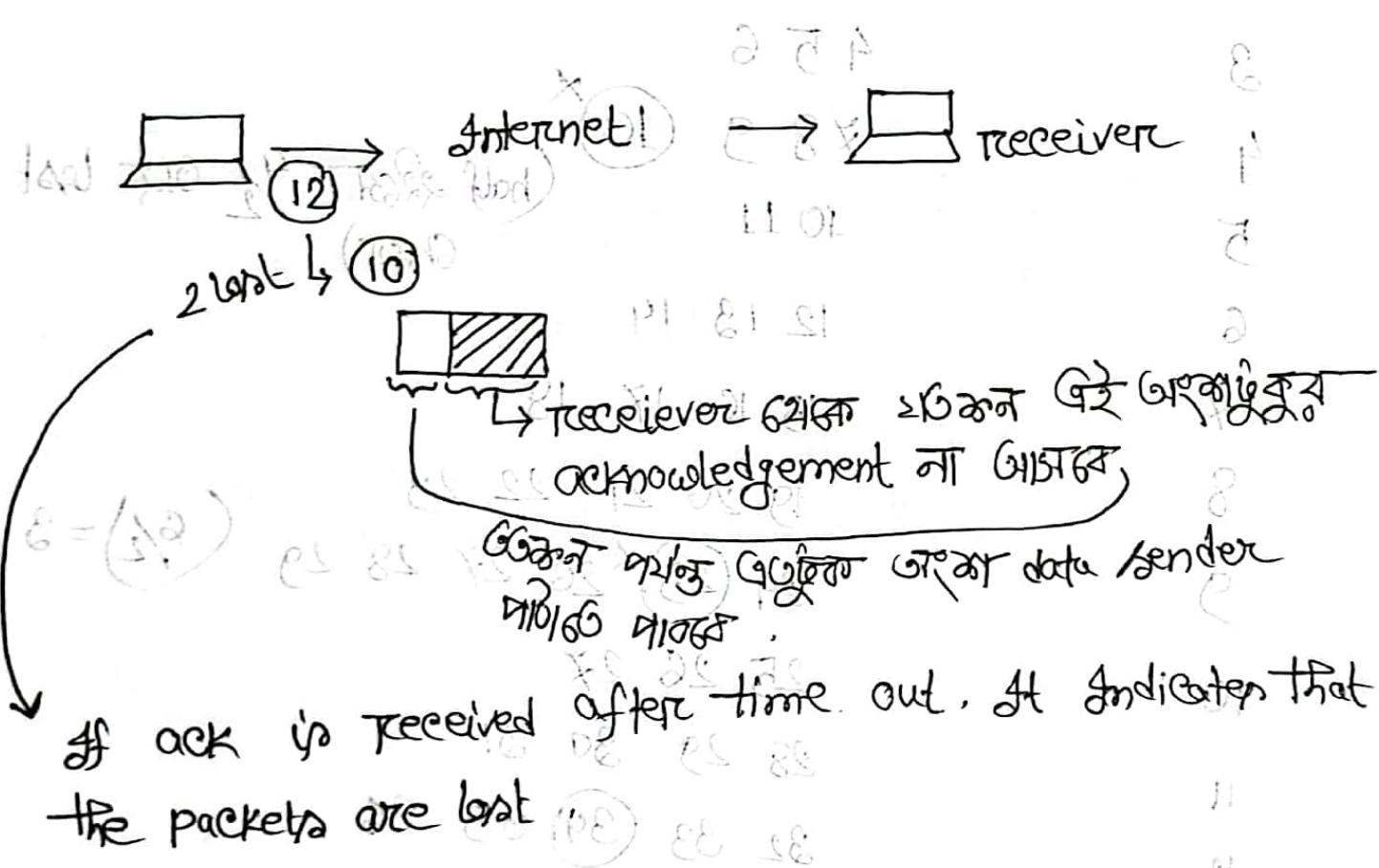
Congestion Control is required because there are multiple flows into the internet & our goal is to ensure each flow gets its fair equal opportunity.

Approach

→ we will use additive increase multiplicative decrease (AIMD) which is a feedback control algorithm whose main purpose is to adjust the bitrate of TCP flow. (how fast should the sender send packets of information without feeding into other power of)

Congestion window (Cwnd)

For each connection, TCP maintains a Congestion window. Congestion window is a variable that limits the amount of data sent to the network before an acknowledgement of the total number of unacknowledged packets is equal to the Cwnd, the sender stops sending data.



If ack is received after time out, it indicates that the packets are lost.

Additive increase →
Increase the sending rate or Cwnd by 1 MSS until packet loss.

Multiplicative decrease →
Decrease the sending rate by half when a loss occurs.

packet = 50

RTT = 100 (ms) മുകളിൽ നിന്നുമ്പോൾ

packet size = 1 KB അനുസരം 90% വിലപ്പെടുത്താൻ കൂടാൻ ശ്രദ്ധിക്കുന്നു

lost packets = 10, 25, 34, 45 എന്നും ഒരു അടിസ്ഥാനമായി

frames കുറവാണെന്നും lost of 1st burst, loss of 1st segment എന്നും കൂടാൻ ശ്രദ്ധിക്കുന്നു

RTT Seg NO to stations lost if

1 . job failed 1st time, lost, burst lost

2 23

3 4 5 6

4 ~~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 32 33 34 35 36~~ ~~37 38 39 40 41 42~~ lost

5

6 12 13 14

7 ~~15 16 17 18~~

8 ~~19 20 21 22 23~~

9 ~~24 25 26 27 28 29~~

$(6/2) = 3$

first 10 bursts H.O. AND ~~25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42~~ if

11

25 26 27

28 29 30 31

32 33 34

35 36

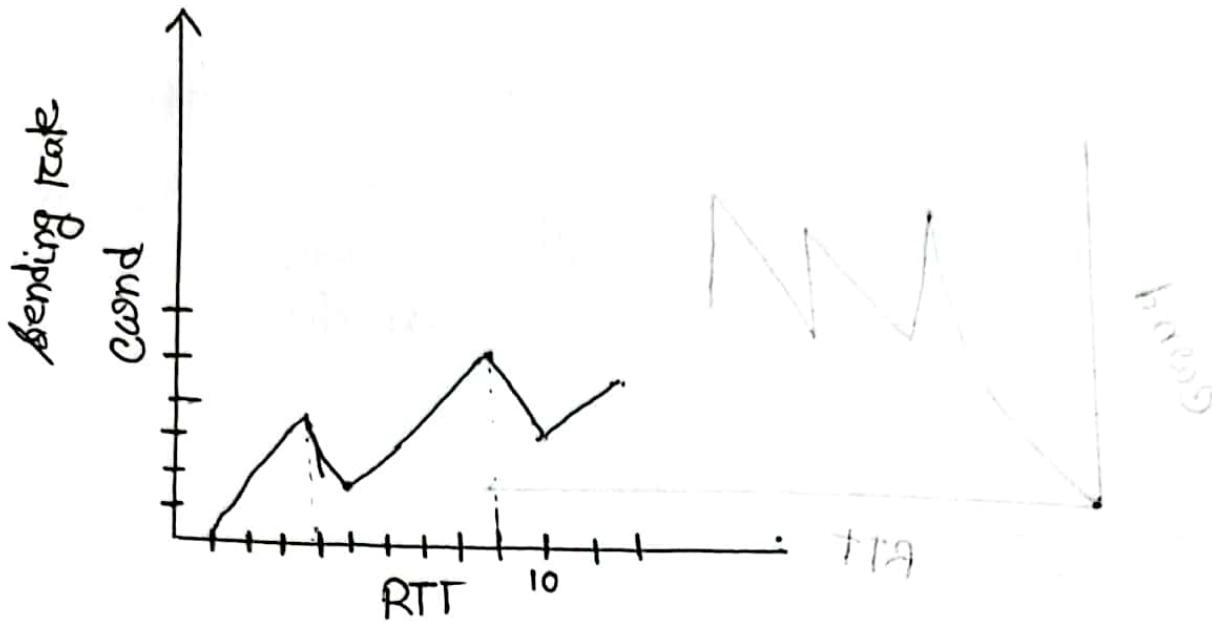
37 38 39 40 41 42

com. 1 13 ~~14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42~~ available

will get

1st 10 bursts H.O. ~~25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42~~ available

11 12 13 14 15



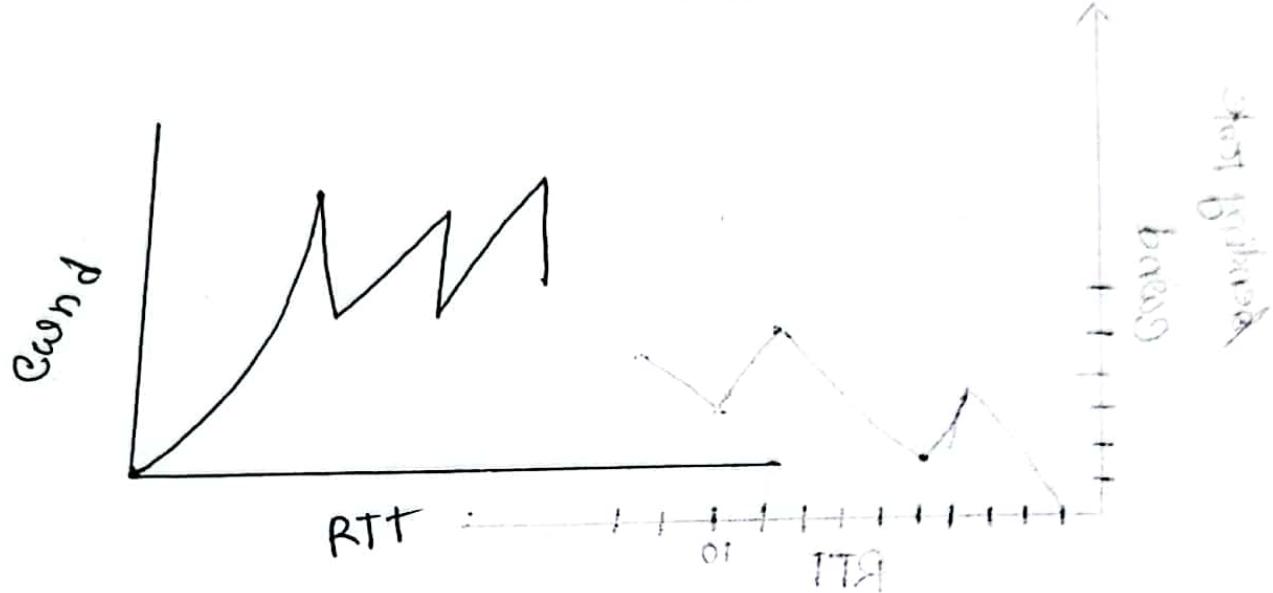
AIMD Graph

TCP slow start

under AIMD, it can take some time to reach the link capacity. slow start process helps to reach the link capacity faster.

during this fetch, connection rate or sending rate is increased exponentially until a loss is occurred

RTT	seq No.
1	1
2	2 3
3	4 5 6 7 8 9 10 11 12 13 14 15
4	10 11 12 13
5	14 15 16 17 18
6	19 20 21 22 23 24
7	(25) 26 27 28 29 30 31
8	25 26 27
9	



कृष्ण अमा

संचार एवं कृष्ण

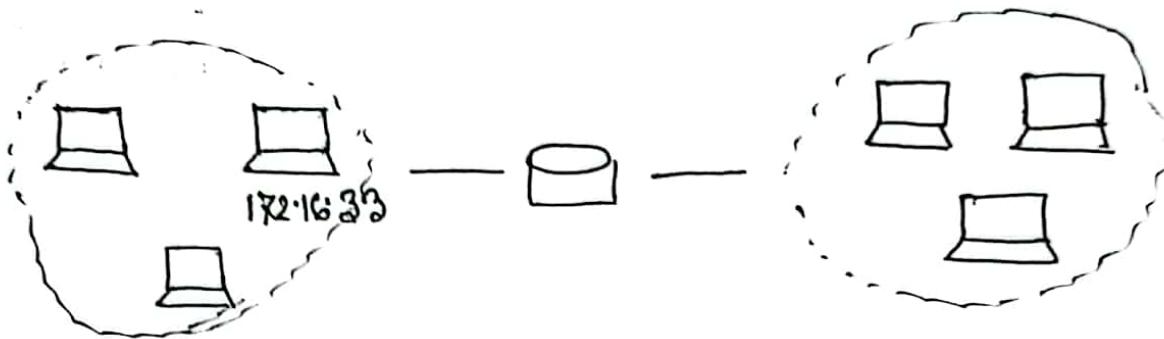
संचार के अन्तर्गत ऑफलाइन एवं ऑफलाइन
प्रॉटोकॉल की समीक्षा करें।

(i) शुरू किये जा रहे विभिन्न प्रॉटोकॉल के लिए उनकी विवरण दें।

प्रॉटोकॉल	प्रॉटोकॉल
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

Ip address G network portion & host portion.

Network ID → It identifies specific network. So our Routing table routes the data based on the network address.



Class A

0.0.0.0

127.255.255.255

8 bit	24 bit
Network IP	Host IP

private Ip { 10.0.0.0
10.255.255.255 }

reserve Ip { 0.0.0.0
0.255.255.255 }

ଓমি ইন্টেরনেট রে নেক্সু
ওয়ার্ল্ড প্রিভেট আইপি
প�লিক আইপি রে কন্টেন্ট
কন্টেন্ট মার্কেটিং কেন্দ্র নেট

[Same network G
private রে নেক্সু Data
Communication রে একত্র]

[অ্যারেবার রে রে নেক্সু
public]

It is used to communicate within the same network.
Each device under the same network has a unique
private ip. They can communicate with each other
without internet.