OSI Model

Application }
Presentation } message
Session }

Transport } segment (TCP), datagram (UDP)

(WAN) Network } dataPacket

(LAN) Datalink } Frame

Physical } bits

	Devices	Layer
Active device	Gateway	Application Presentation Session Transport
Active device	Router	Network
Active device	Bridge	Data link
Passive device	Repeater, hub	Physical

* Bridge is known as two - layer switch

* Router is known as three - layer switch

- Data link layer is responsible for node - to - node delivery within the layer; which can be identified by MAC address.

- **Network layer** is responsible for Source - to - Destination delivery between different network in the WAN using IP address.

- **Transport layer** is responsible for process - to - process delivery that can be identified by port address.

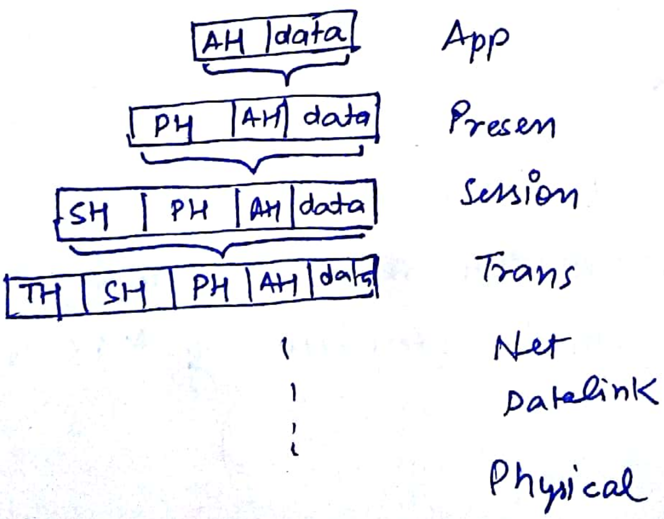
Port address ↔ Transport layer

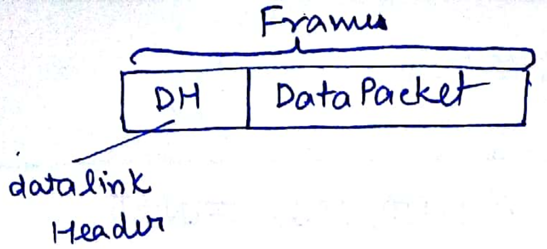
IP address ↔ Network layer

MAC address ↔ Data link layer

- **Socket** = IP address + Port address

- Network Architecture is known as protocol stack Architecture because the last header i.e added at the sender side is the first header ~~to~~ i.e removed at the reciever side.





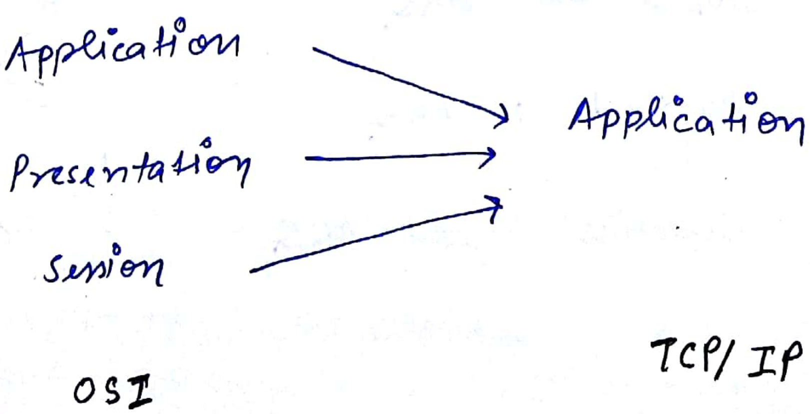
* lower-layer always encapsulate data of upper-layer.

Q. "M" is the message, "H" is the Header added at every layer. "N" layers are present in Heirarchy. Calculate Fraction of Headers in the whole content that is transmitted.

$$H + M$$
$$\downarrow$$
$$H + (H + M)$$
$$\downarrow$$
$$\vdots$$
Final msg $\leftarrow$ $NH + M$ (for N-layers)

so, fraction of header in whole msg $= \left(\dfrac{NH}{NH + M}\right)$

7 layers of OSI Modal are modified as 5-layers of TCP/IP modal. By including the Functionalities of Presentation & session layer in application layer



Scanned by CamScanner

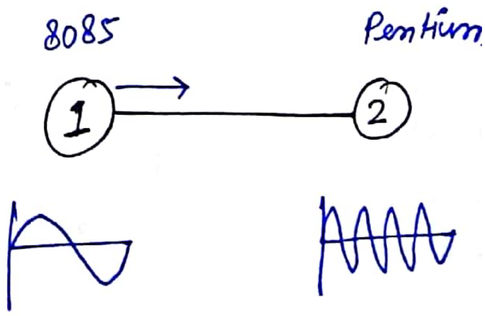
Layer	Services
Application	HTTP, FTP, SMTP, DNS, TELNET
Presentation	Syntax & sementics of data
Session	dialogue control, session management
Transport	flow control, error control, segmentation, congestion policies
Network	Routing algorithms, traffic shaping, fragmentation, IP addressing
Data Link	flow control, error control, Access control, framing
Physical	Physical & electrical properties of cable orchannel.

→ <u>Flow control policies or protocol :-</u>

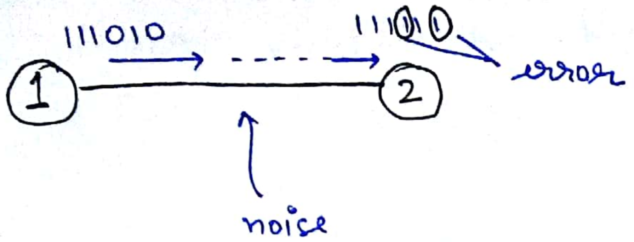
(i) stop & wait ARQ

(ii) Go Back N ARQ

(iii) Selective Repeat ARQ



Synchronization problem

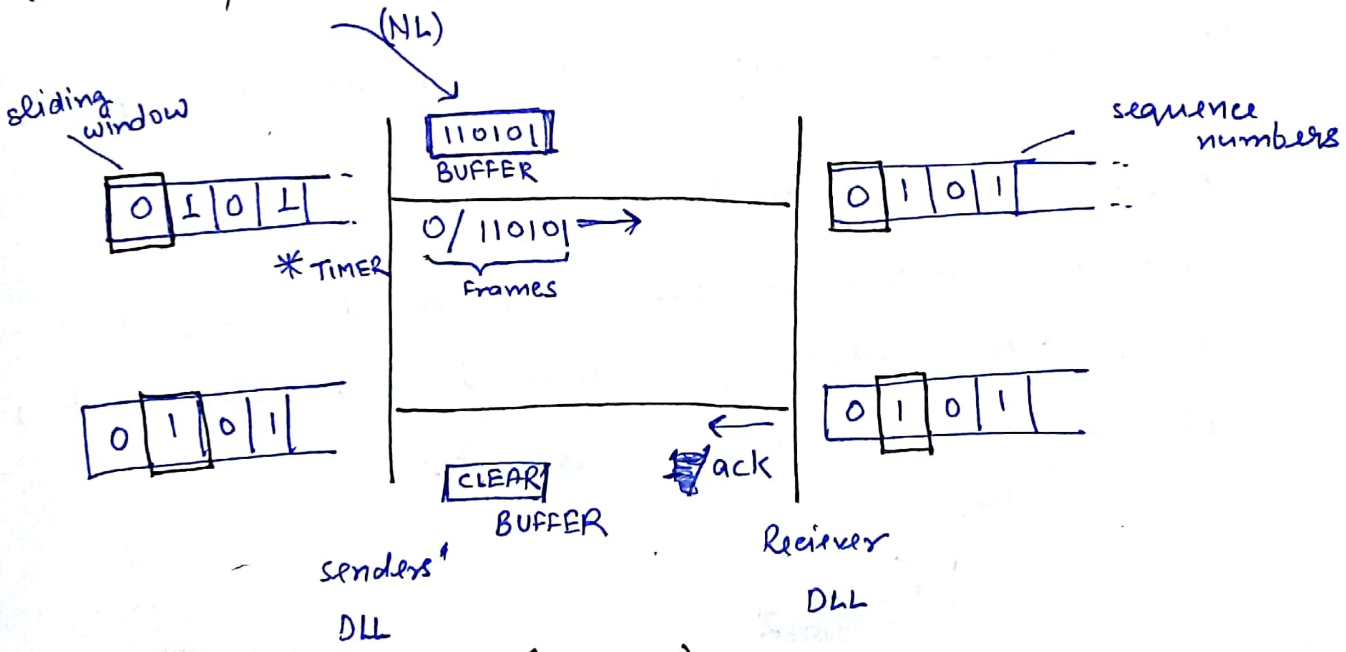


error control policies

- error correction policies
  - eg :- hamming code
- error detection policies
  - eg :- Parity, checksum, CRC

Data Link layer (LAN) :-

(i) Stop & Wait ARQ



(CASE 1)

Frame = sequence no. + datapacket

0/ 110101

- when frames is sended at reciever side sequence no. is matched.

if (sequence no in frame = reciever sequence window)

{

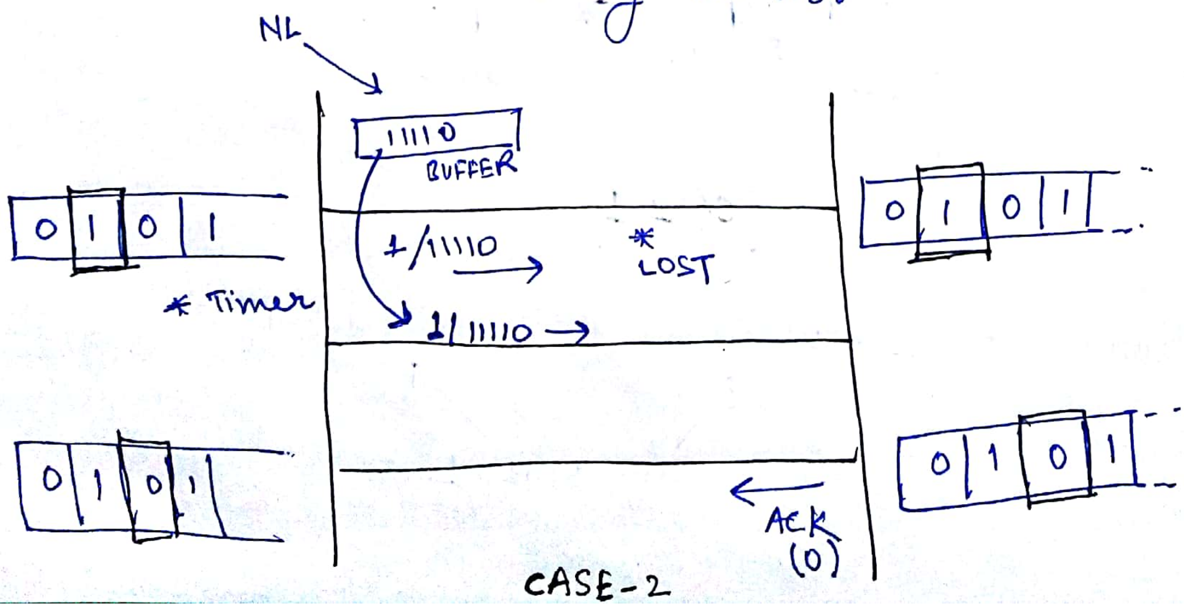
"frame is accepted"

}

- Once the data is reached to destination the sequence no. of the data is compared with reciever sliding window number. If there is a match data will be accepted and reciever windown will slide by one-bit.

If there is a mismatch data will not be accepted.

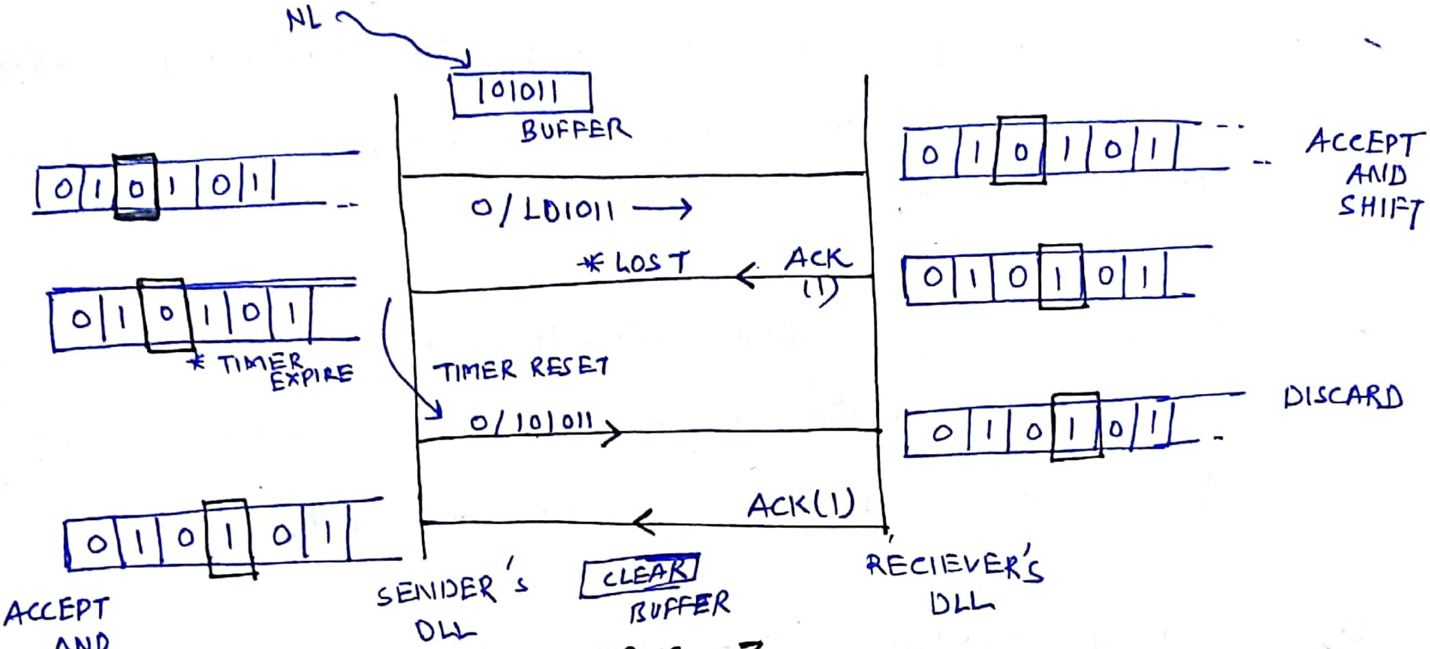
- Acknowledgement (ACK) no. will always be the sequence number of the next expected data, then only acknowledgement is accepted and slide window of sender is shifted by one bit.



CASE-2

- In this case if after sending if TIMER expires then it means data is lost.

- Then ARQ (Automatic ~~Reqeue~~ Repeat Request) protocol comes into existence. ARQ automatically resends the Buffer value with sequence no. (i.e frame) to reciever. ~~Fframe~~ ~~is~~ :



CASE - 3

- In this case ACK of reciver is lost i.e timer expires. Then Timer is RESET and frame is again sended which is not accepted due to sequence no. mismatch.

- In stop & wait ARQ, it supports individual frame & individual acknowledgment.

- Stop and wait is a theorical protocol without sliding window whereas stop & wait ARQ is a practical protocol with sliding windows.

<u>Conditions for sliding window protocols</u> :-

- In all sliding window protocols the maximum sender window size indicates the no. of frames that are transmitted in RTT.

- In all sliding window protocols the maximum sender window + maximum reciver window will always be equal to distinct sequence no. count.

Q. B.W = 10 Mbps , RTT = 50 μs , framesize = 5 bits

$$1 \text{ sec} = 10^7 \text{ bits}$$

$$50\mu s = 50 \times 10^{-6} \times 10^7$$

$$= 500 \text{ bits}$$

no. of frames which can be send $= \frac{500}{5}$

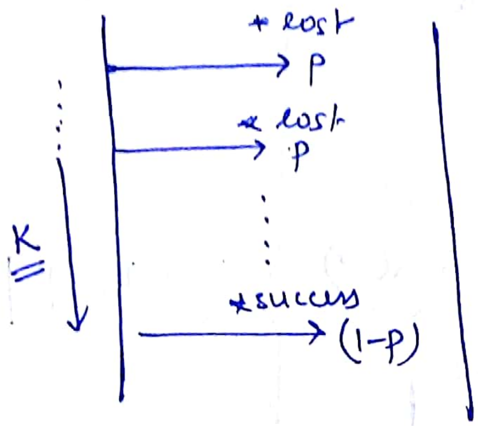
$= 100$ frames

B.W utilization in stop & wait ARQ $= \dfrac{\text{no. of frame in ARQ}}{\text{Total no. of frames allowed}}$

$= \frac{1}{100} \times 100\% = 1\%$

- In ARQ bandwidth utilization is less as we are transmitting only 1 frame in RTT.

Q. Probability of being lost is 'p', mean no. of transmissions of a frame is ___ .



- So let us assume total K times transmission is done

$$\overbrace{p \times p \times p \times \cdots}^{K-1} \times (1-p)$$

$E(K) = \sum_{K=1}^{\infty} K * p(K)$

mean

$= \sum_{K=1}^{\infty} K\,(p \times p \times p \cdots (1-p))$

$= \sum_{K=1}^{\infty} K\,(p^{K-1}(1-p))$

$= (1-p)\,[1 + 2p + 3p^2 + \cdots]$

$= (1-p)(1-p)^{-2}$

$= \boxed{\left(\frac{1}{1-p}\right)}$

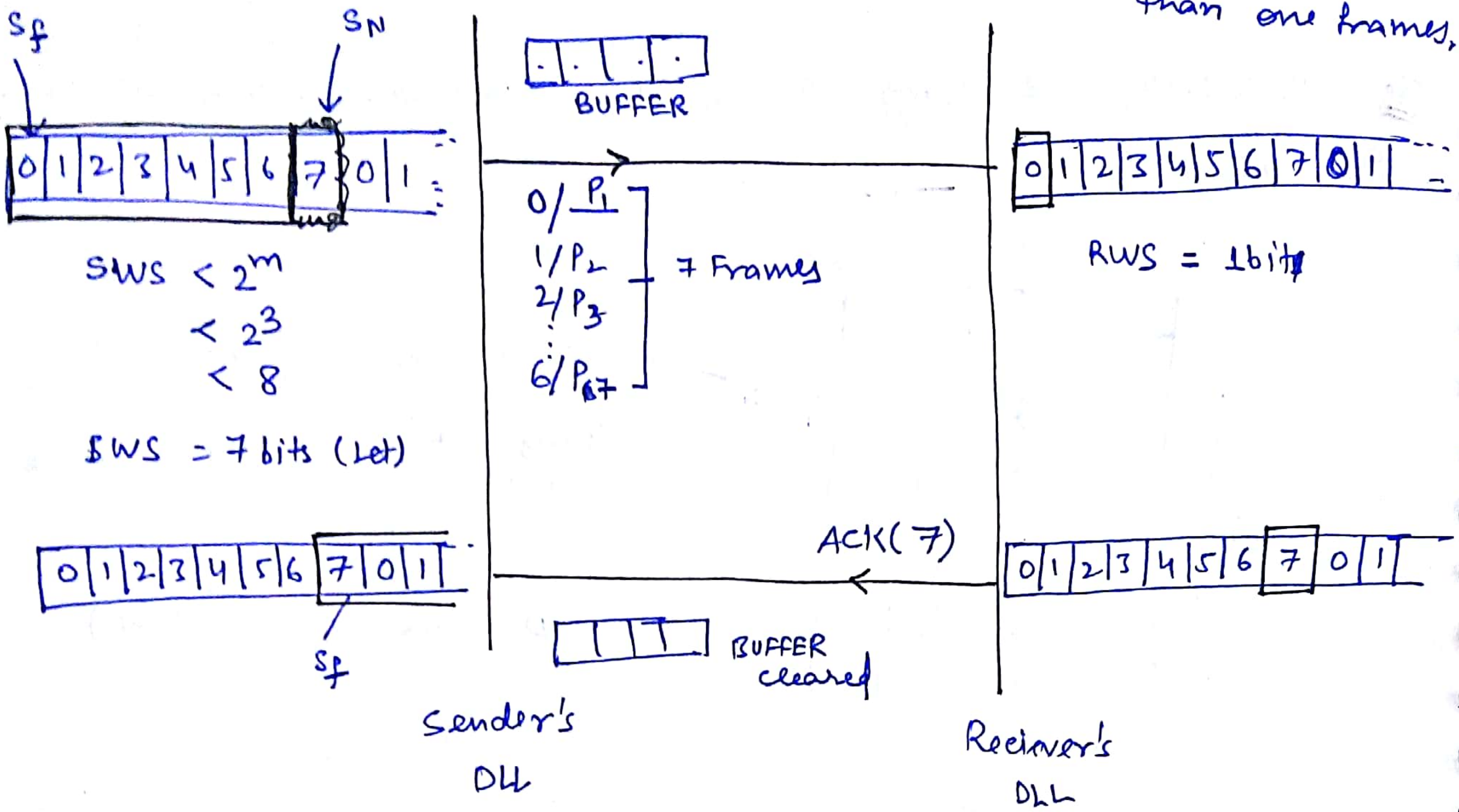
$\frac{1}{1-p}$ = ...

$1-p = 2$

$-p = 1-2 = ...$

(ii) Go Back N ARQ :-

* Buffer size is large as here it has to hold more than one frames,



condition for this protocol = size of sliding window $< 2^m$

where m is bits for sequence number

$m = 3$

000	—	0
001	—	1
010	—	2
011	—	3
100	—	4
101	—	5
110	—	6
111	—	7

$$SWS + RWS = 2^m$$

Scanned by CamScanner

* S_f is the first bit of window and S_N is the next bit of last bit of window.

* no. of frames sended = size of window in bits i.e 7

* Ack (7) is ~~sended~~ recived by sender then,

  if ~~……~~ACK $S_f = S_N$

  then all frames are sended successfully

  And $S_f = \overset{ACK}{S_N}$ (S_f sets to $\overset{ACK}{S_N}$)

Q. 5-bit sequence no. is used in Go Back ARQ. Calculate SWS and RWS.

$SWS < 2^m$ i.e $SWS < 2^5$

i.e $SWS < 32$. $SWS_{max} = 31$

$RWS = 1$

Q. Maximum sender window size (SWS) in go-back ARQ is 7 no. of sequence bits = ?

$SWS < 2^m$

$7 < 2^m$

$SWS_{max} = 2^m - 1$

$2^m = 1 + SWS_{max}$

$\left(m = \log_2 (1 + SWS_{max})\right)$

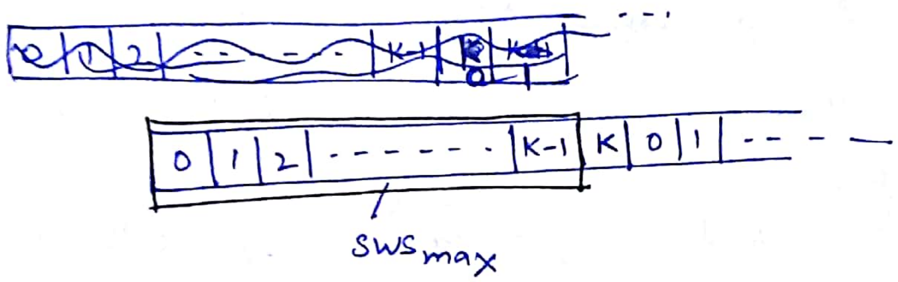
Scanned by CamScanner

Q. max sequence number in Go Back ARQ is "K".

max size of sender window is –

(a) K−1 (b). K+1

✓(c) K (d) none



- Go Back ARQ supports both individual Ack as well as cummulative ACK.

- whenever RWS = 1 ; it accepts only in-order frames.

Q. B.W = 10 Mbps , RTT = 50 MS, frame size = 25-bits.

Calculate window size and no. of sequence bits in GOBACK ARQ.

$1 \text{ sec} = 10^7 \text{ bits}$

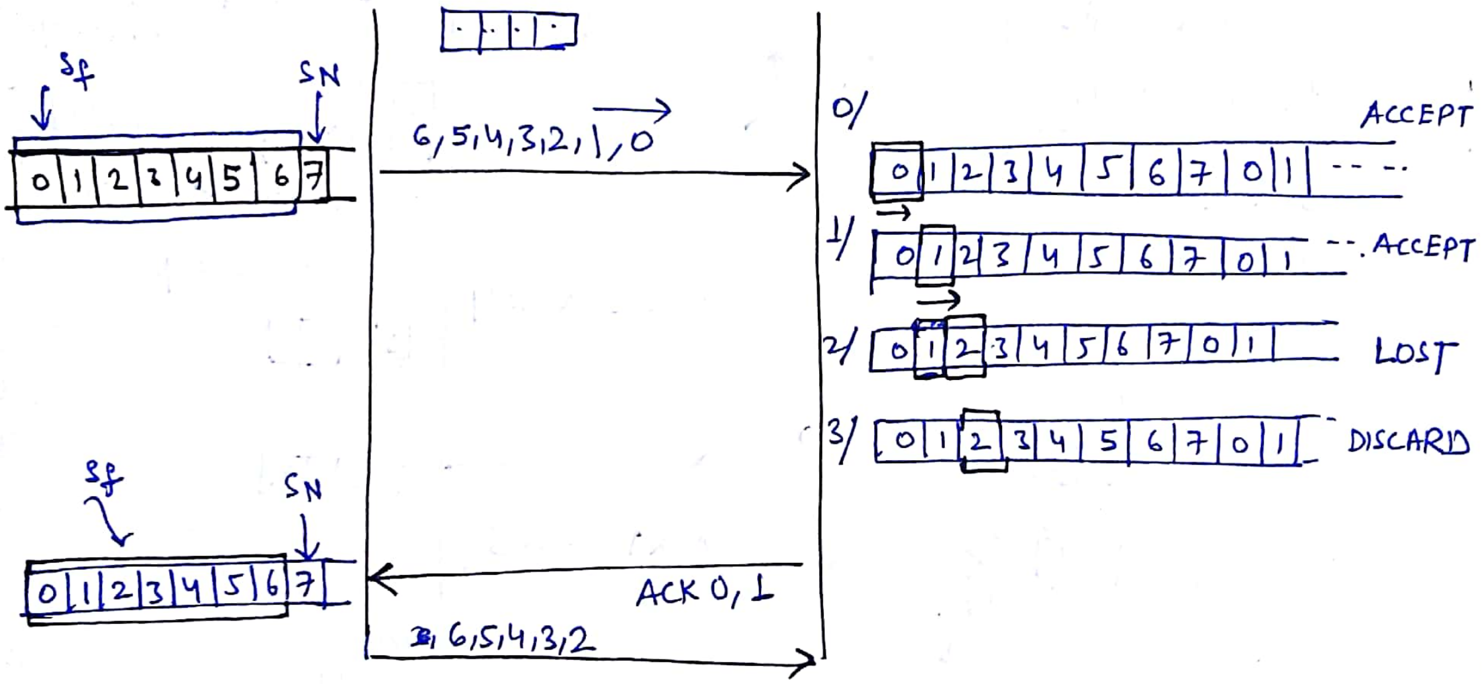
in 50 MS (RTT) = 500 bits

$\text{Total no. of frames send} = \frac{500}{25} = 20$

no. of sequence bits	SWS	RWS
~~3~~ 4	$2^4-1 = 15$	1
5	$2^5-1 = 31$	1

window size = 31

no. of sequence bits = 5

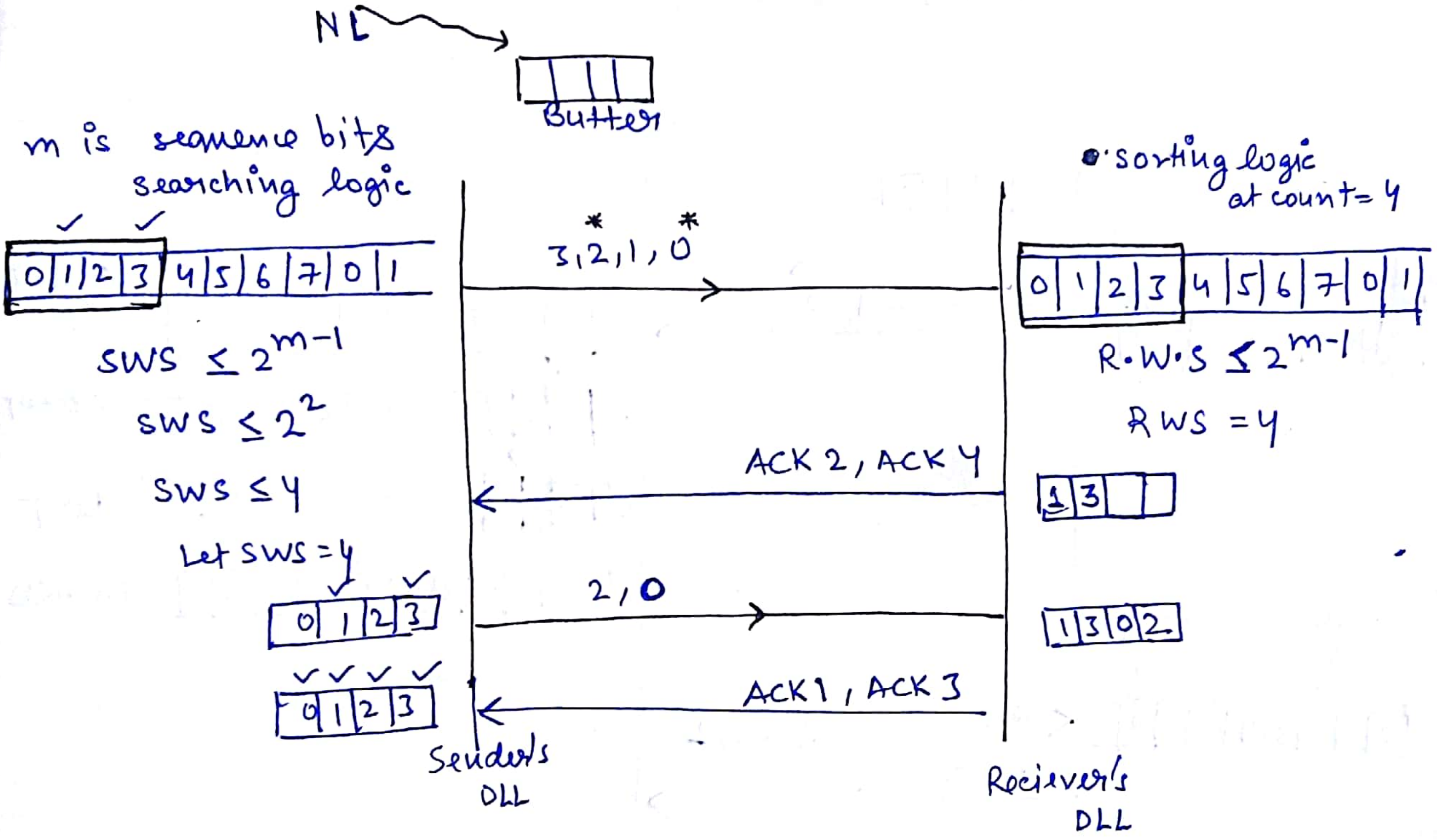


- in this case frame 2 and 4 are lost, then after of frame 0, and frame 1, no frames will be accepted as frame 2 is lost so the RW remains at position 1. Since sequence no $\neq$ RW position.

- So, if a frame is lost then that frame as well as all the following frames should be re-transmitted.

- In Go - back N ARQ there are more number of re - transmission, For noisy channels. So overall utilization is less.

Selective Repeat ARQ :-



- when 0 and 2 are lost, 1 and 3 frames are accepted which are kept inside an intermediate buffer.

- ACK 2, ACK 4 is sended, sender then ~~sen~~ resends 0 and 2.

- In this searching logic is applied during sequence matching

- count is the counter variable which counts the no. of frames accepted. As here no. of frames sended is 4 so ~~count~~

$$0 \le count \le 4$$

- when count = 4 i.e

1	3	0	2

intermediate buffer

Buffer is sorted

0	1	2	3

and then sended to Network layer.

Q. Max- sender window size in selective repeat ARQ is 4. Calculate no. of sequence bits?

$$SWS_{max} \le 2^{m-1}$$

$$4 \le 2^{m-1}$$

$$4 = 2^{m-1}$$

$$\log_2 4 = m-1$$

$$2 = m-1$$

$$\boxed{m = 3}$$

Q: B.W = 10 mbps , RTT = 50 μs , frame size = 5 bits

In selective repeat ARQ calculate no. of sequence bits?

$1\,s = 10^7$ bits

in $50\,\mu s = 50 \times 10^{-6} \times 10^7$

$= 500$ bits

No. of frames $=$ ~~frame~~ $\dfrac{\text{total bits sended}}{\text{frames size}} = \dfrac{500}{5}$

$= 100$ frames

Window - size $= 100$

since $SWS_{\cancel{max}} \leq 2^{m-1}$

$SWS_{max} = 2^{m-1}$

$100 = 2^{m-1}$

$\log_2 100 = m-1$

$m = \lceil \log_2 100 \rceil + 1$

$m = 7 + 1$

$\underline{(m = 8)}$

Q: Max -sequence no. in selective repeat ARQ is 7 max -size of sender window?

sequence range $=$ 0 to 7

i.e total numbers in sequence $= 8$

$2^m = 8 \qquad m = 3$

so, m (no. of sequence bits) = 3

$$S_{max} = 2^{m-1}$$

$$\boxed{S_{max} = 2^{3-1} = 2^2 = 4}$$

Q. Maximum sequence number is "N" in selective repeat ARQ. Calculate ~~number of sequence bits~~. maximum size of sender's window ?

$$SWS_{max} = \frac{N+1}{2}$$

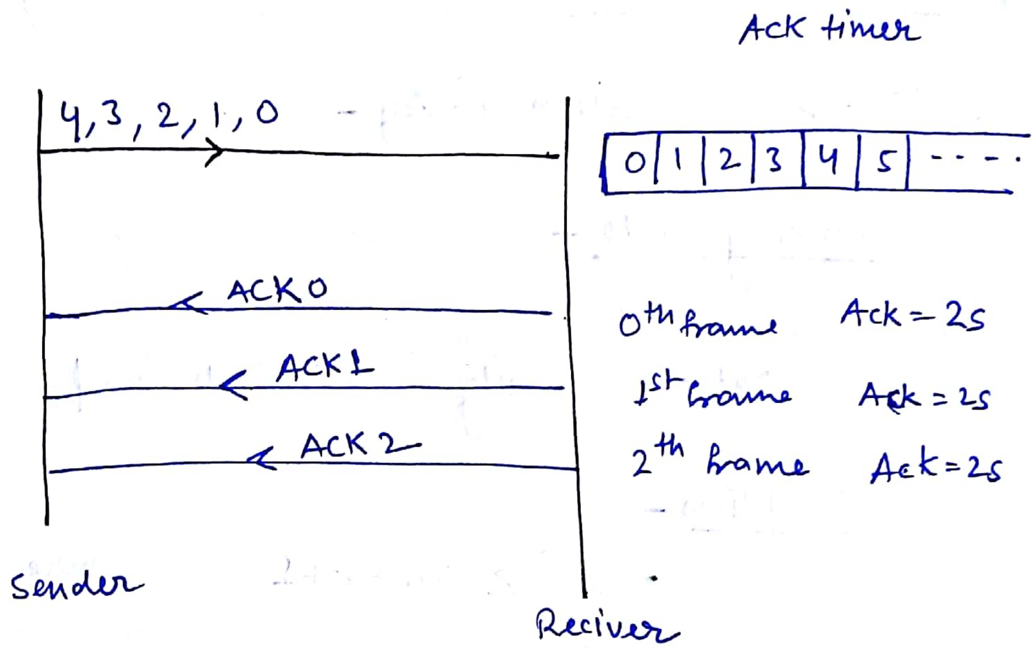
Buffer size requirement	Stop & wait ARQ	Selective Repeat ARQ	GoBack N ARQ
$m = 3$	1	4	7

* whenever the bandwidth is limited selective ARQ is preffered.

* whenever a packet comes to a router & router identifies that there are multiple paths are available for a packet, then the path which has more no. of ones in ~~sub~~ mask is preffered.

Special Cases of GoBack N ARQ :-

(i) Individual Acknowledgement -



- In case of Individual Ack's, the timer is started for every frame i.e recieved.
- This timer value should be small enough so that you can send the acknowledgement immediately.

(ii) Cummulative Acknowledgement -

- In case of cummulative ACK's, the timer is started for the first frame i.e recieved. This timer value should be large enough so that all frames might have enough. But it should be less then the sender timer. In easy words we can say timer value is moderate.

Error control policies of Data Link layer :-

Error correction policy -

Hamming code -

Data + parity = Code word

Condition -

- $2^r \geq m+r+1$ where m is data bits and r is parity bits

for eg :-

Data = 10011010

for $r = 3$

$2^3 \geq 8+3+1$ ✗

$r = 4$

$2^4 \geq 8+4+1$

$16 \geq 13$ ✓

so r should be 4

- Parity bits should be placed in power of 2 position.

Since Data = 10011010 where $m = 8$

	1	2	3	4	5	6	7	8	9	10	11	12
code word	$\underline{P_1}$	$\underline{P_2}$	1	$\underline{P_4}$	0	0	1	$\underline{P_8}$	1	0	1	0

- $\underline{P_1}$: 1 3 5 7 9 11

  $\underline{0}$ 1 0 1 1 1

  put (1 or 0) so that no. of 1's is even

  $\underline{P_2}$: 2 3 6 7 10 11

  $\underline{1}$ 1 0 1 0 1

  $\underline{P_4}$: 4 5 6 7 12

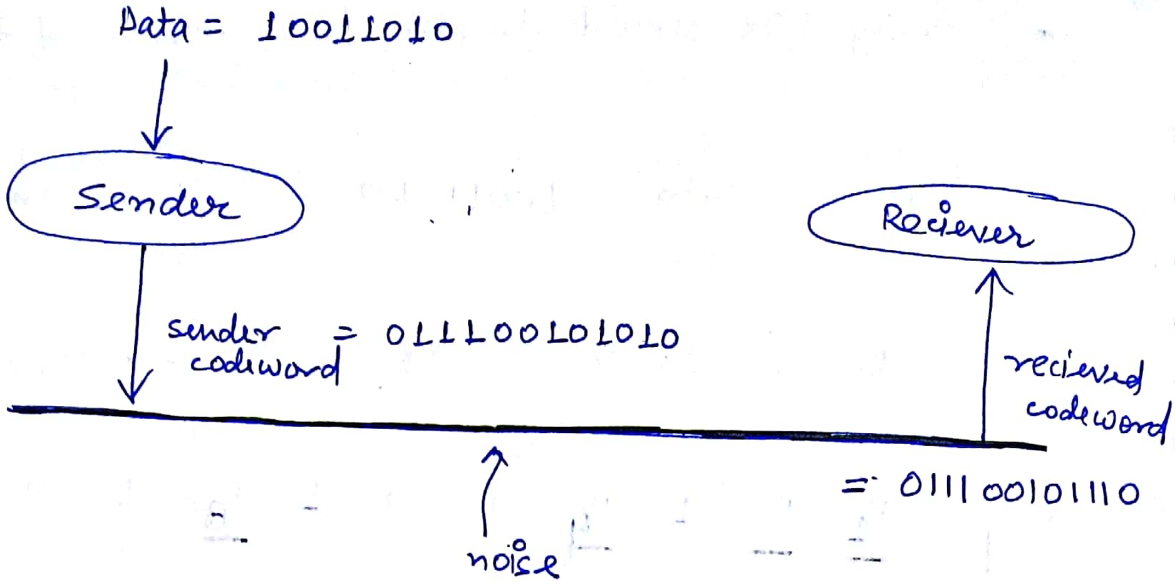
  $\underline{1}$ 0 0 1 0

  P_8 : 8 9 10 11 12

  $\underline{0}$ 1 0 1 0

So,

$$\boxed{\begin{aligned} P_1 &= 0 \\ P_2 &= 1 \\ P_4 &= 1 \\ P_8 &= 0 \end{aligned}}$$



Now the recieved code word is changed due to noise on channel.

Recieved codeword

	1	2	3	4	5	6	7	8	9	10	11	12
=	0	1	1	1	0	0	1	0	1	1	1	0
	P_1	P_2	1	P_4	0	0	1	P_8	1	1	1	0

<u>recieved Parity</u> -

$P_1 = 0$
$P_2 = 1$
$P_4 = 1$
$P_8 = 0$

<u>parity calculated at recieved</u> :-

P_1' :	1	3	5	7	9	11
	0	1	0	1	1	1

$P_1' = 0$ ✓

since P_1 recieved and P_1' calculated are same

$$P_1 = P_1'$$

then P_1 is correct.

P_2' :

2	3	6	7	10	11
0	1	0	1	1	1

$P_2' = 0$ ✗

$(P_2 \neq P_2')$

P_4' :

4	5	6	7	12
1	0	0	1	0

$P_4' = 1$ ✓

$(P_4 = P_4')$

P_8' :

8	9	10	11	12
1	1	1	1	0

$P_8' = 1$ ✓

$(P_8 \neq P_8)$

since P_2 and P_8 are incorrect then

position of error bit = $\underbrace{P_2 \quad P_8}_{8+2 = 10^{th} \text{ bit}}$

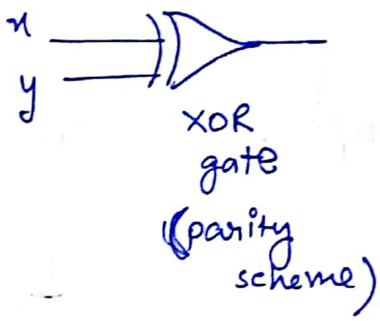
* **Limitation** – it can only correct 1 bit error

Error detection policy:-

Parity scheme :-

Data + Parity = codeword

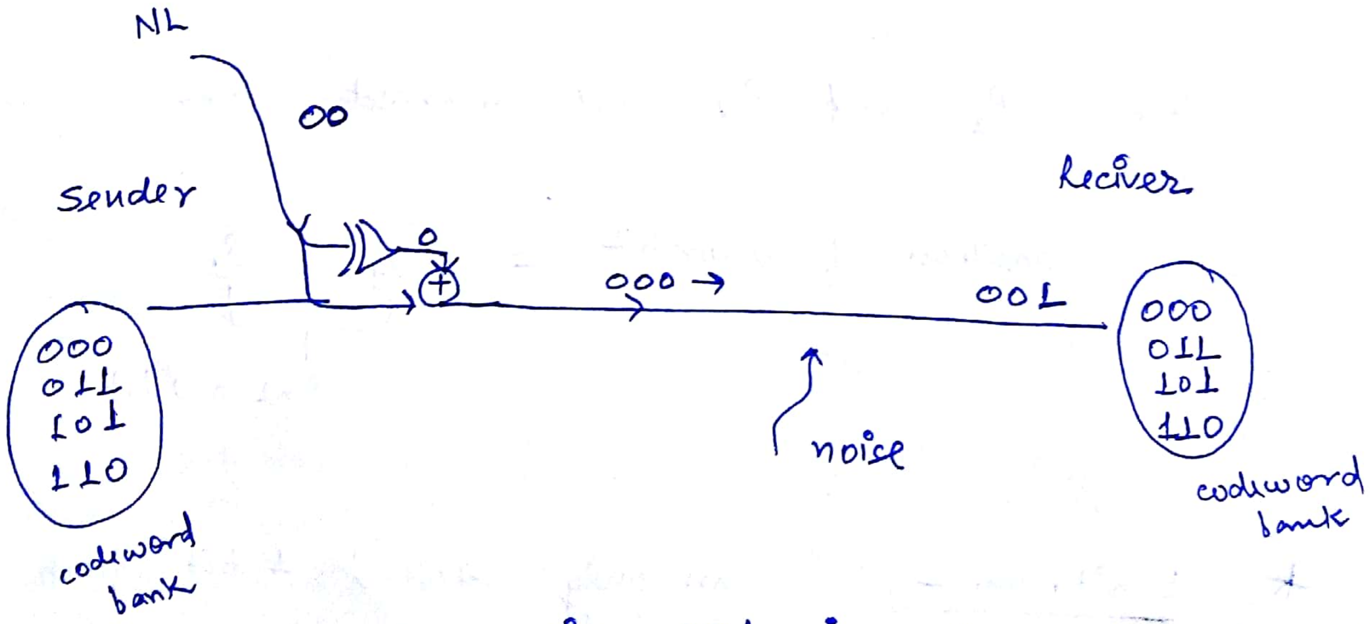
x	y	output parity
0	0	0
0	1	1
1	0	1
1	1	0



so, valid codewords are — 000, 011, 101, 110

(00 → 000, 01 → 011, 10 → 101, 11 → 110)

others are invalid.



since 001 is not a valid codeword so recieved data is not accepted.

- 1 bit error ~~codew~~ data are only detected correctly.

- Noise can generate both valid codewords as-well as invalid codewords.

- When a valid code ~~coard~~ word is converted into an invalid codeword by noise then error's can be detected.

eg. 000 ⟶ 001

- When a valid codeword is converted into another valid codeword then error's can't be detected.

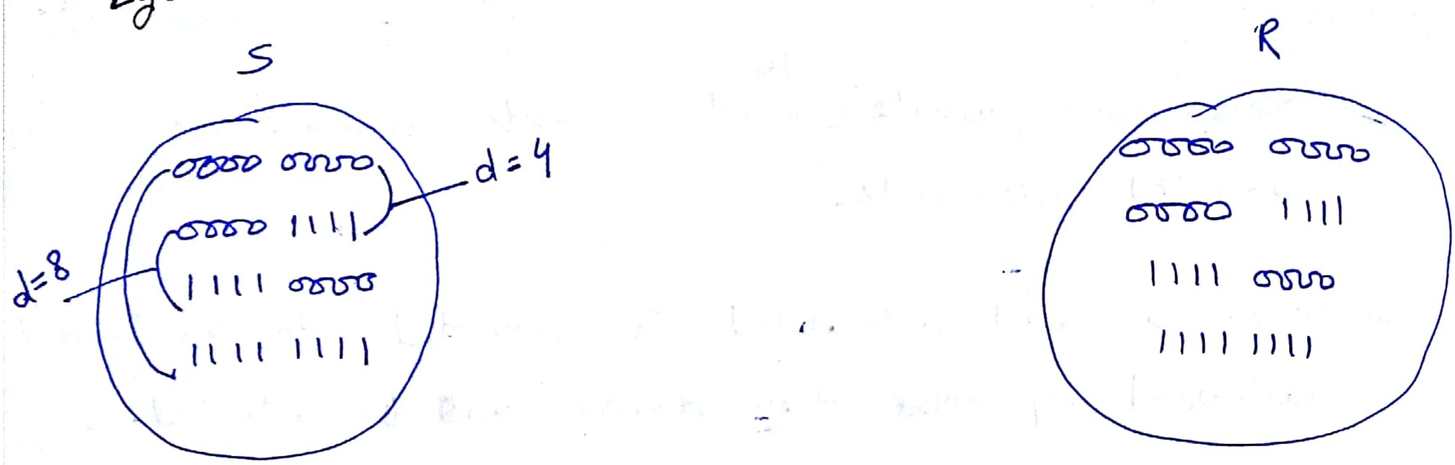
eg. 000 ⟶ 011

Hamming distance :-

- The numbers of bits that differ each other b/w two codewords is the hamming distance. We can also say that ~~xor~~ of no. of 1's in xor of two code-word is the hamming distance.

- To detect "d" errors, the minimum hamming distance should be (d+1).

Eg.



so minimum hamming distance $= 4$

d_{min}

so, it can detect errors upto 3-bits.

00000000 ⟶ ~~0~~0101010 (not accepted & error detected)

⟶ 00001111 (accepted & error not detected)

that's why not all 4-bit error words are detected.

> * Limitation of parity scheme is even number of errors can't be detected.

Scanned by CamScanner

Checksum :-

Data + checksum = codeword

* we can make our own rules for checksum -

CASE-1 (data change but checksum not)

Let Data = 74 12 89 =

checksum rules :

74
12
89
———
1 75

→ 75
+1
———
76

99
−76
———
23

→ Checksum = 23

codeword = 74128923

Sender → Reciever

codeword = 76168923

76
16
89
———
1 81

→ 81
+1
———
82

99
−82
———
17

17 ≠ 23

so error detected

CASE-2

(data remains same and checksum changed)

recieved codeword = 74128946

23 ≠ 46

so error detected

CASE-3 (Both data and checksum changed)

sender codeword = 741289 23

recieved code word = 761689 17

calculate codeword = 17

since 17 = 17

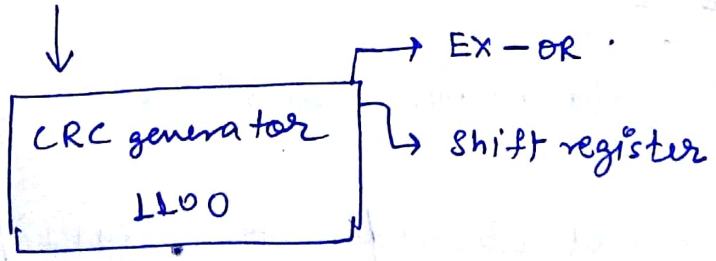
so it accepted the data , where error is not detected.

* Limitation of checksum is it noise modifies the data in such a way that the vertically placed bits will cancel each other then the calculated checksum will be equal to recieved checksum. such type of error can't be detected.

CRC (cyclic Redundancy check) :-

Data + CRC bits = codeword

Data = 10101



padding bits = CRC genertor bits − 1

= size (1100) − 1 = 4−1 = 3

10101 $\underbrace{000}$ padding bits

MSB is ~~compared~~ taken

$$
\begin{array}{r|l}
1100 & 10101000 \quad (1101 \\
 & 1100 \\
\hline
 & 1101 \\
 & \vdots \\
\hline
 & 100
\end{array}
$$

CRC = 100

code word = 10101 100

Recieved codeword = 10101111

$$
\begin{array}{r|l}
1100 & 10101111 \quad (~~101011~~ \; 11001 \\
 & \vdots \\
\hline
 & 011 \rightarrow \text{syndrome}
\end{array}
$$

if syndrome = 0 no error

≠ 0 error

CRC rules (i) CRC generator should not contains "x".

where ($x = 1 \times x^1 + 0 + x^0$)

(ii) If $x+1$ is taken as generator it can detect odd no. of errors.