- In multimode step Index, the density of core remains const. from centre to the edges.
- In multimode graded index, the density is highest at the centre of the core & drses gradually to its lowest at the edges
- Single mood use step Index fibre optics & a highly focused source of light that limits beams to a small range of angle.

Twisted cable has been replaced with fibre optics.

Used in fast ethernet (100 base FX, 1000Base X)

★ Data link layer

Error detection & correction

★ Types of errors :-

- Single bit error -
- Burst error

★ Redundancy - Adding extra bits to the signal code for error detection & correction

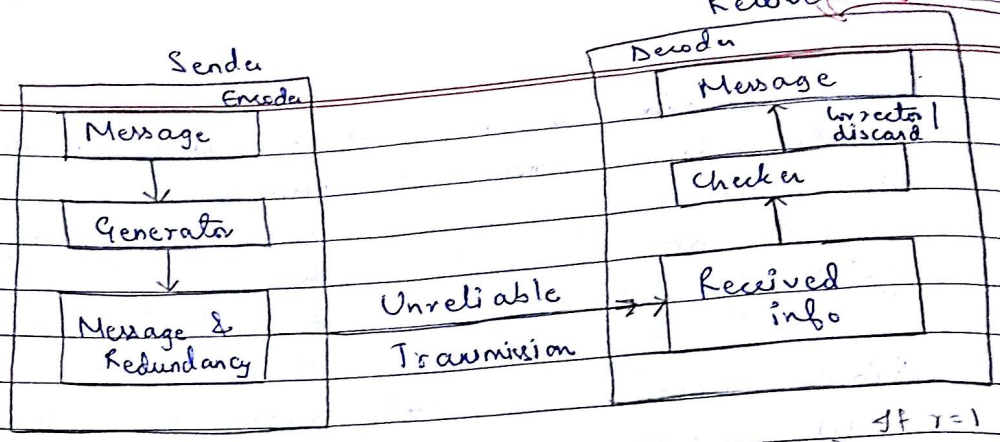
★ Error detection Vs Error correction

★ Forward error correction Vs Retransmission
↳ Self- correction

★ Coding -

P.T.O

Scanned by CamScanner



Only for single bit error

If $r=1$

If $r=3$; error correction

★ Block coding (Only for error detection)

In block coding, the message is divided into blocks, each having k bits called data word. r redundant bits are added to each block to make the length $n = k + r$. The resulting n bit block is called code word.

Eg:

k bit = 3 (data words $\} = 2^k$)

$r = 1$ (redundant bit)

$n = k + r$

n bit (codeword $= 2^{k+r}$)

Valid code word $= 2^k$

Invalid " $= 2^n - 2^k$

$k = 2$; $n = 3$

Data word	Codeword (valid) XOR
0 0	0 0 0 → XOR
0 1	0 1 1
1 0	1 0 1
1 1	1 1 0

Scanned by CamScanner

If we add only 1 redundant bit, then we can detect only single bit error.

Error detection code can detect only the type of error for which it is designed, other error will remain undetected.

For correction (capable of detecting only single bit error)

$k = 2$ $n = 5$

Data words	Codeword (valid)
0 0	0 0 0 0 0 → Duplicate
0 1	0 1 0 1 1 → XOR
1 0	1 0 1 0 1
1 1	1 1 1 1 0

Data Rate limits -

The data rate depends on 3 factors :-

- Bandwidth available
- level of signal
- Quality of channel (level of noise)

Noiseless channel → Nyquist Bitrate

$$\text{Bitrate} = 2 * \text{Bandwidth} * \log_2 L$$

Bitrate → bits/sec

L → no. of signal levels (used to represent data)

Noisy Channel → Shannon Capacity

$$\text{Capacity} = \text{Bandwidth} * \log_2 (1 + SNR)$$

Scanned by CamScanner

Q. Consider a noiseless channel with bandwidth of 3000 Hz transmitting a signal with 2 signal levels. Calculate the max bitrate.

Sol^n

$$\text{Birate} = 2 * \text{Bandwidth} * \log_2 L$$
$$= 2 * 3000 * \log_2 2$$
$$= 6000 \text{ bits/sec}$$

Q. Data is sent with bit rate 265 Kb/s over a noiseless channel with bandwidth 20 KHz. How many signal levels are req.

Sol^n

$$\text{Bitrate} = 265 \text{ Kbps}$$
$$= 265 \times 1024$$

$$\text{Bandwidth} = 20 \text{ KHz} = 20{,}000$$

$$\log_2 L = 6.625$$
$$L = 2^{6.625} = 98.7$$

$64 = 2^6$ $\quad$ $128 = 2^7$

Q. Assume SNR_{db} to be 36 & channel bandwidth is 2 MHz. Find channel capacity

$$SNR_{db} = 10 \log_{10} SNR$$

$$\text{Capacity} = 2{,}000000 * \log_2 (\cancel{\ })$$

$$SNR = 10^{3.6} = 3981$$

$$\text{Capacity} = 24 \text{ Mbps}$$

Scanned by CamScanner

☆ Hamming Distance

The hamming distance b/w two words is the no. of difference b/w corresponding bits.

when r = 1

00 → 000
01 → 011
10 → 101
11 → 110

000
011
XOR = 011 = 2

101
110
XOR = 011 = 2

d = 2

when r = 3

00 → 00000
01 → 01011
10 → 10101
11 → 11110

00000
11110
XOR = 1111 0 = 4

d = 4

☆ Min Hamming Distance

It is the smallest hamming distance b/w all pairs in a set of words.

when r = 1

(d_{min}) min hamming distance = 2

when r = 3

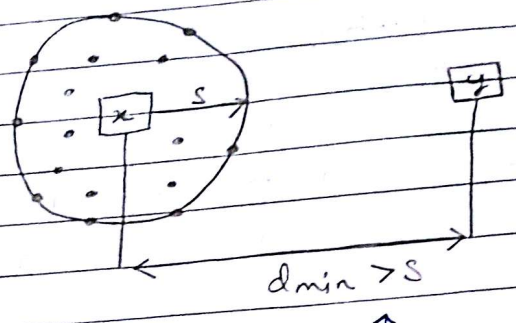
(d_{min}) min hamming distance = 3

00000
01011
XOR = 1 11
= 3

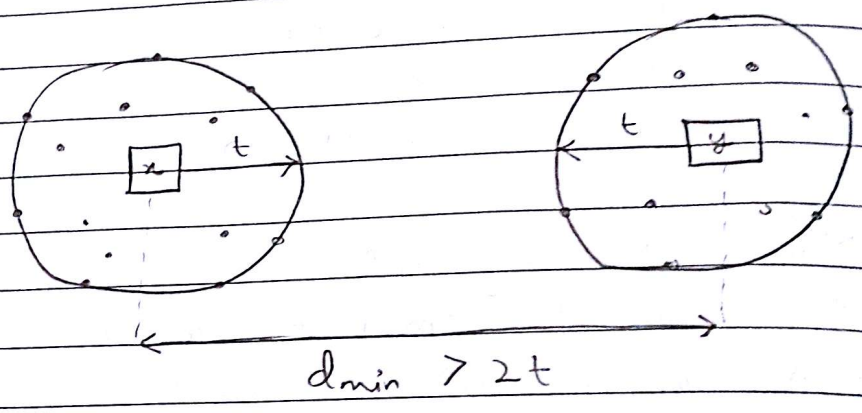
To guarantee the detection of upto S errors in all cases, the min hamming distance in a block code must be S+1

where S = no. of ~~min~~ errors we can detect

To guarantee correction of upto t errors in all cases, the min hamming distance in block code must be $d_{min} = 2t + 1$



For error detection



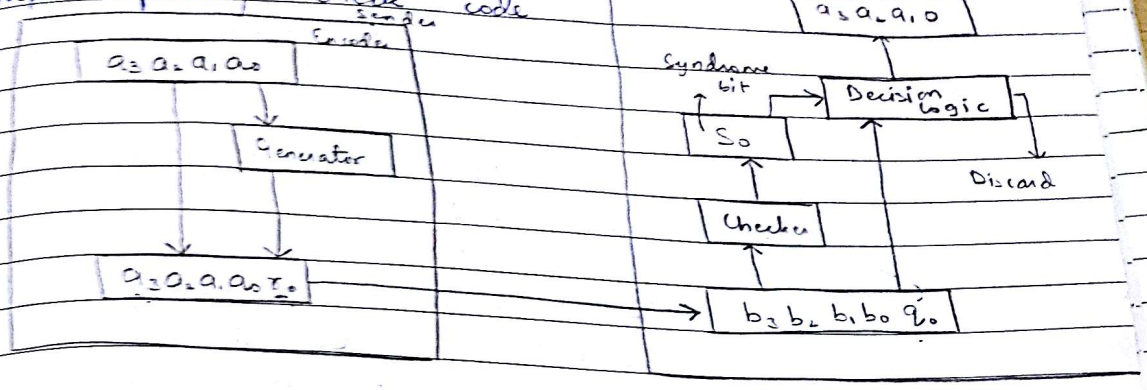
For error correction

Q $d_{min} = 4$

~~t ≠ 1~~ $t = 1$; Correction

$s = 3$; detection

★ Simple ~~parity~~ parity check code



$$r_0 = a_3 + a_2 + a_1 + a_0$$
$$S_0 = b_3 + b_2 + b_1 + b_0 + q_0$$

If $S_0 = 0$ bit accepted
$= 1$; error in received code
∴ discard

$C(n, k)$ — n: no. of bits in codeword; k: No. of bits in data word

⇒ If $k = 4$ & $n = 5$ i.e $C(5, 4)$

No. of Data words $= 2^4 = 16$

In even parity - we make no. of 1's in codewords even
" odd " " " " " " " odd

Using even- parity

Data word	Code word
0000	00000
0001	00011
0011	00110
0010	00101

Scanned by CamScanner

Suppose we want to communicate
datawords ← 1011 then code words = 10111 → codeword
if received = 10111
$S_0 = 0$; Accept
if received = 10101
$S_0 = 1$; Discard

if received = 10110 → Error in redundant bit even then we discard it as $S_0 = 1$
$S_0 = 1$; Discard

(2 bit error) if received = 00110
$S_0 = 0$; Accept → false acception i.e system not capable of detecting 2 errors.

(3 bit error) if received = 01011
$S_0 = 1$; Reject

works for odd no. of bit errors but not for even no. of bit errors.

2-D parity check

1	1	0	0	1	1	1	1
1	0	1	1	1	0	1	1
0	1	1	1	0	0	1	0
0	1	0	1	0	0	1	1
0	1	0	1	0	1	0	1

2-D parity check can detect upto 3 errors that occur anywhere in the table.
Errors affecting 4 bits may not be detected.

Scanned by CamScanner

Check Sum

7 11 12 0 6 = 36 = 100100

$\rightarrow$ 100

6 = 0110

9 = 1001

1's compliment of 6

Now numbers to be sent are

7 11 12 0 6 9

At receivers end

= 45 = 101101

$\rightarrow$ 101

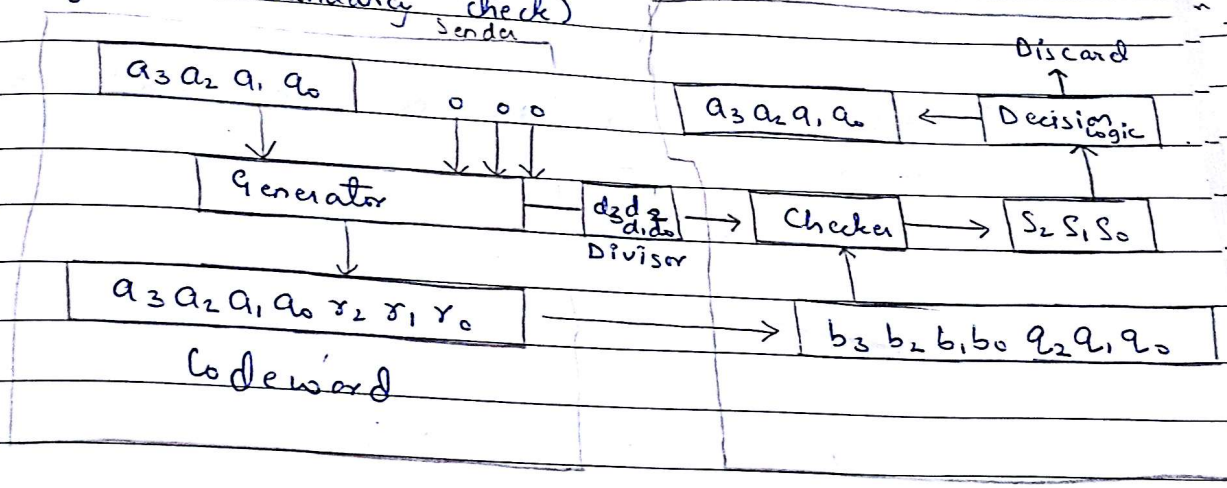
1111

0000 $\rightarrow$ 1's complement of 1111

$\therefore$ no error

If we change any number, then 1's comple at receiver's end will not be 0

CRC (Cyclic redundancy check)



In the encoder, the data word has k bits. The codeword has n bits. The size of the divisor is $\{n-k\}+1$. The no. of redundant bits is $n-k$.

Example :

```
          1010
(1011) ) 1001000
         1011
         ----
         0100
         0000
         ----
         1000
```

```
  1000
- 1011
  ----
  0110
  0000
  ----
   110
```

replaced with 000 i.e the redunda

∴ code word = 1001110 → redundant bits i.e the remainder

At receiver end, the code word is divided by divisor

```
        10010
1011 ) 1001110
       1011
       0101
       0000
        10111
        1011
        0000
         0000
         0000
         0000
```

∴ $S = 0$; hence accepted

→ If there was an error in fourth bit from the left

```
        1011
1011 ) 1000110
       1011
       0110
       0000
        1110
        1011
         1000
         -1011
           11
```

$S \neq 0$ ∴ Discarded

We cannot extract the dataword

Scanned by CamScanner

Dataword = 1010011010 ; 10 bit
Divisor = 10111 ; 5 bit

Q. Suppose the spectrum of a channel is b/w 5 MHz & 6 MHz. If SNR = 24 dB. Find max channel capacity. Also find max no. of signal levels in the transmitted signal.

Q. 8 bit dataword = 10011101
generator poly = $x^3 + 1$
Solve using CRC

$x^3 + 1 = 1x^3 + 0x^2 + 0x + 1$
$= 1001$

codeword = 10011101000

```
        10001101
1001 ) 10011101000
       -1001
        0001
        0000
         0011
         0000
          0110
          0000
           1101
           1001
            1010
            1001
             011
             1100
             1001
              1010
              1001
```

Imp...

Hamming code

8 bit dataword = 10011101

1	2	3	4	5	6	7	8	9	10	11
		D_3		D_5	D_6	D_7		D_9	D_{10}	D_{11}
P_1	P_2	1	P_4	0	0	1	P_8	1	1	0

P_1 = XOR (All odd positions)

= XOR (D_1, D_3, D_5, D_7, D_9, D_{11}) = XOR(1, 0, 1, 1, 1, 0)

= 1

P_2 = XOR (2, 3, 6, 7, 10, 11) = 1 + 0 + 1 + 1 + 0 = 1

P_4 = XOR (4, 5, 6, 7, 12) = 0 + 0 + 1 + 1 = 0

P_8 = XOR (8, 9, 10, 11, 12) = 1 + 1 + 0 + 1 = 1

∴ code word = 1110001 1101

Q Dataword = 10 1000110

1	2	3	4	5	6	7	8	9	10	11	12
		D_3		D_5	D_6	D_7		D_9	D_{10}	D_{11}	D_{12}
P_1	P_2	1	P_4	0	1	0	P_8	0	0	1	1

P_1 = 1 + 0 + 0 + 0 + 1 + 0 = 0

P_2 = 1 + 1 + 0 + 0 + 1 + 0 = 1

P_4 = 0 + 1 + 0 + 1 + 0 = 0

P_8 = 0 + 0 + 1 + 1 + 0 = 0

∴ code word = 0 1100100 0110

Scanned by CamScanner

For detection & correction

$C_1 = 0 + 1 + 0 + 0 + 0 + 1 + 0 = 0$
$C_2 = 1 + 1 + 1 + 0 + 0 + 1 = 0$
$C_4 = 0 + 0 + 1 + 0 + 1 + 0 = 0$
$C_8 = 0 + 0 + 0 + 1 + 1 + 0 = 0$

If there is an error in the 3rd bit from the left then what is the mechanism carried out at receiver end.

$\therefore$ codeword = 010001000010

$C_1 = 0 + 0 + 0 + 0 + 0 + 1 + 0 = 1$
$C_2 = 1 + 0 + 1 + 0 + 0 + 1 = 1$
$C_4 = 0 + 0 + 1 + 0 + 1 + 0 = 0$
$C_8 = 0 + 0 + 0 + 1 + 1 + 0 = 0$

Error in C_1 & C_2 = 1 + 2 = 3rd bit

$\therefore$ Flip 3rd bit

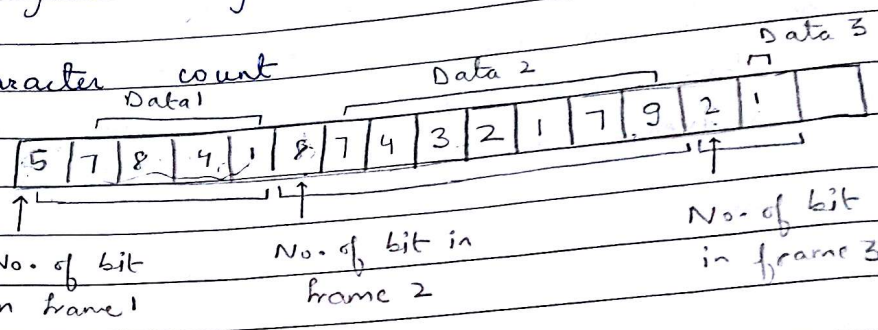
Scanned by CamScanner

Framing

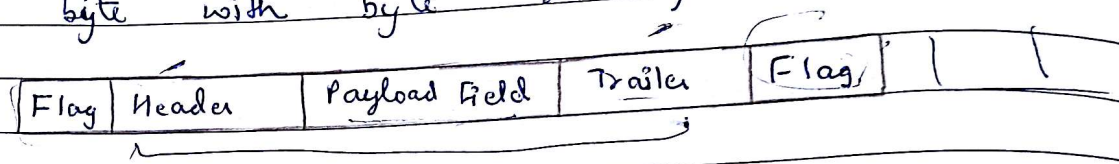
There are 4 methods for framing :-

- Character count
- Flag bite with byte ~~stuffing~~ stuffing
- Starting & ending flag with bit ~~stuffing~~ stuffing
- Physical layer coding violations

Character count



Flag byte with byte stuffing



If data contains value same as flag, we use an escape byte so that it is not read as flag.

~~Flag~~ A Flag B Flag

Flag A Esc Flag B Flag

A Esc B

A Esc Esc B

A Esc Flag B

A Esc Esc Esc Flag B

Scanned by CamScanner

1111101

Starting & ending Flag with bit stuffing

Flg 0 1111110

∴ Convert → 111111

1111101

→ 11111101

111110101

0110 11111111111111111 00010

Q Following char encoding is used in data link protocol

A : 00010111

B : 11111111

Flag : 01111110

Esc : 11100000

a) Show bit for A B Flag B Esc Flag

b) Identify orignal frame bit after destuffing using bit stuffing framing method

10111110 111111 0 000 10111110111110111110100000

Start Flag

0 111111 000 1

End Flag

a) Flag A B Esc Flag B Esc Esc Esc Flag Flag

b) 00010 1111 11111 11111111 00000

A B B ESC

Scanned by CamScanner

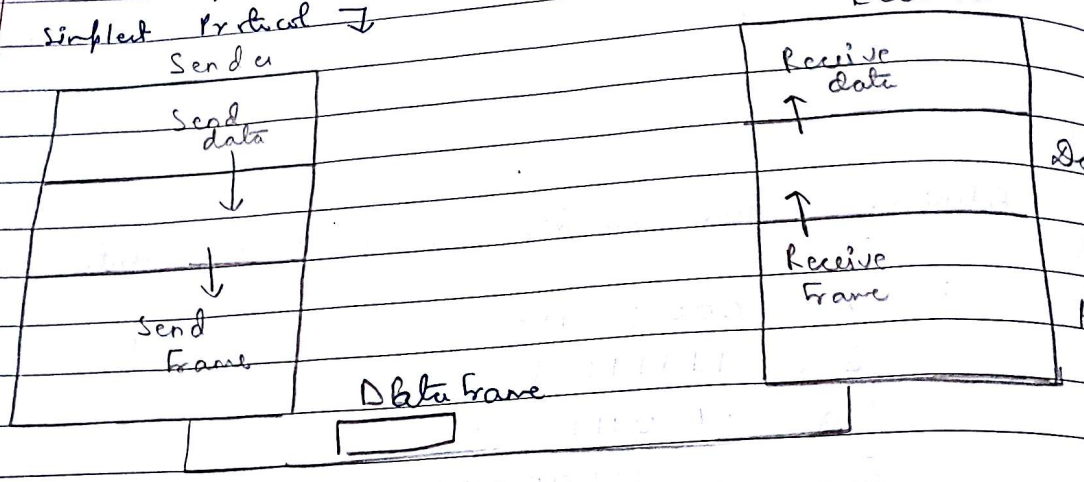
ARQ - Automatic repeat request

☆ Data Link layer Protocols

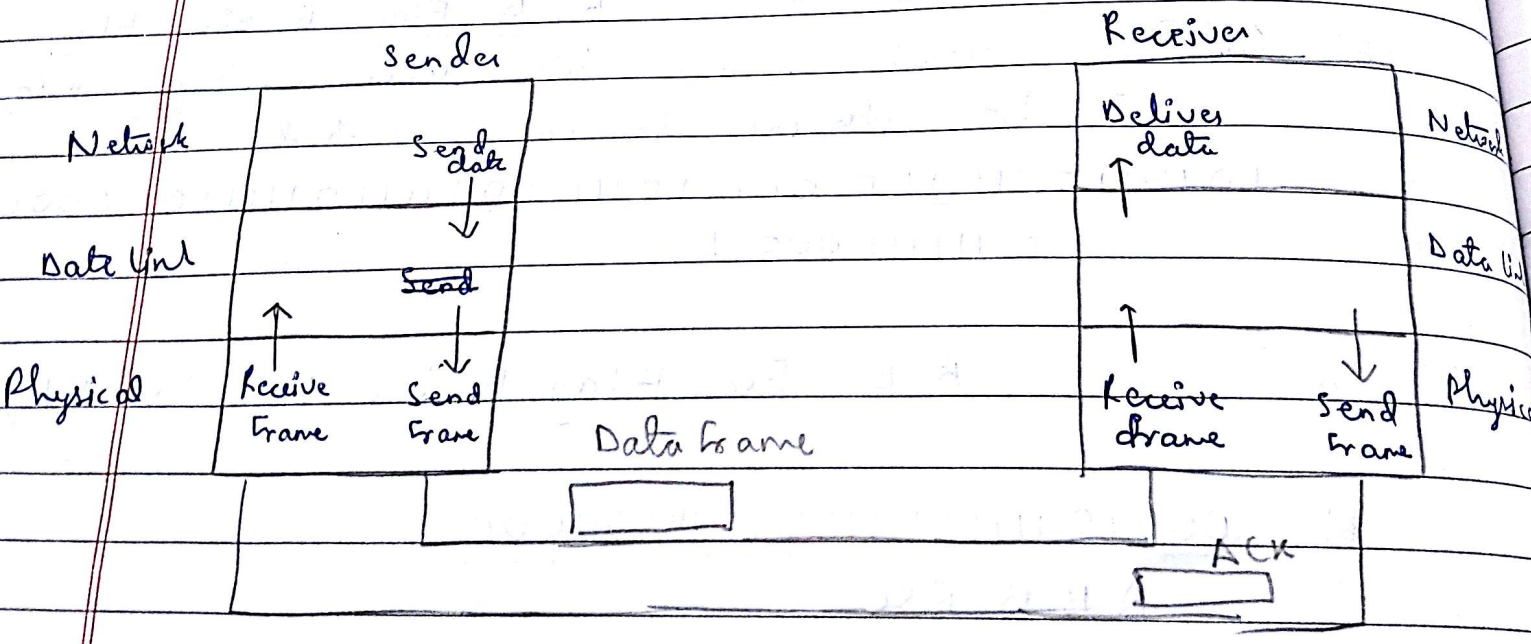
Data link layer protocols

- Noiseless channel
  - → Simplest
  - → Stop & wait
- Noisy channel
  - → Stop & wait ARQ
  - → Go back & ARQ
  - → Selective repeat ARQ

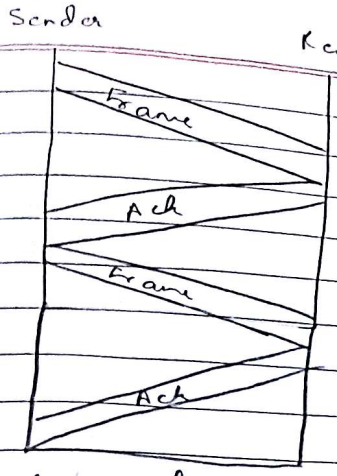
☆ Simplest Protocol ⇓



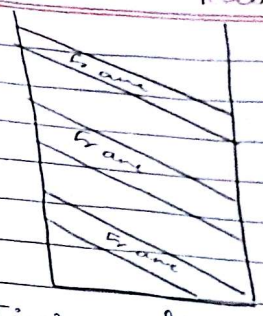
☆ Stop & wait Protocol-



Scanned by CamScanner

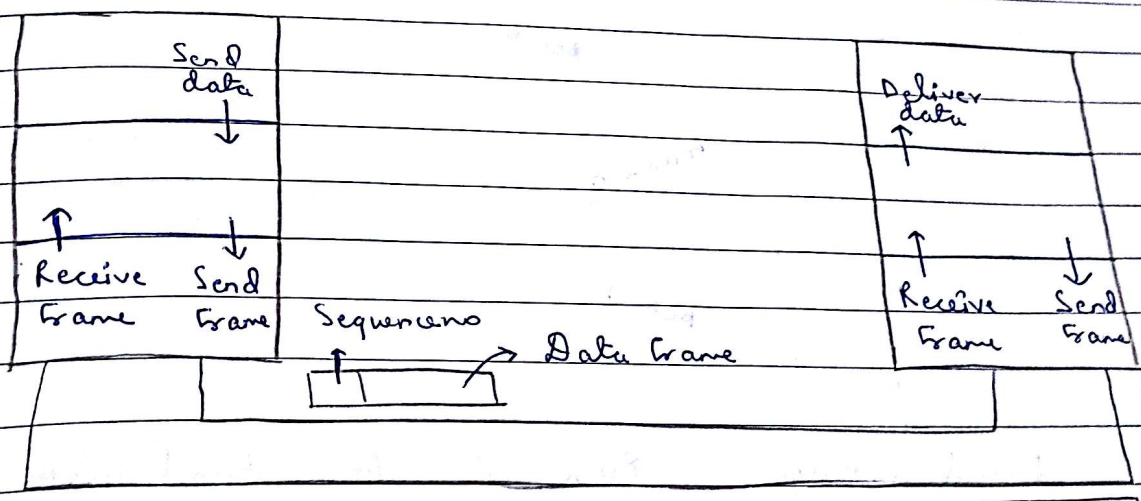


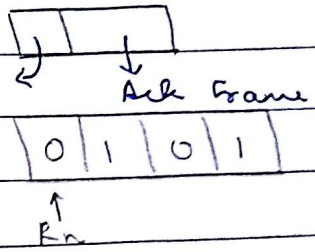
Timing dig for stop & wait



Timing dig for simplex

Stop & wait ARQ





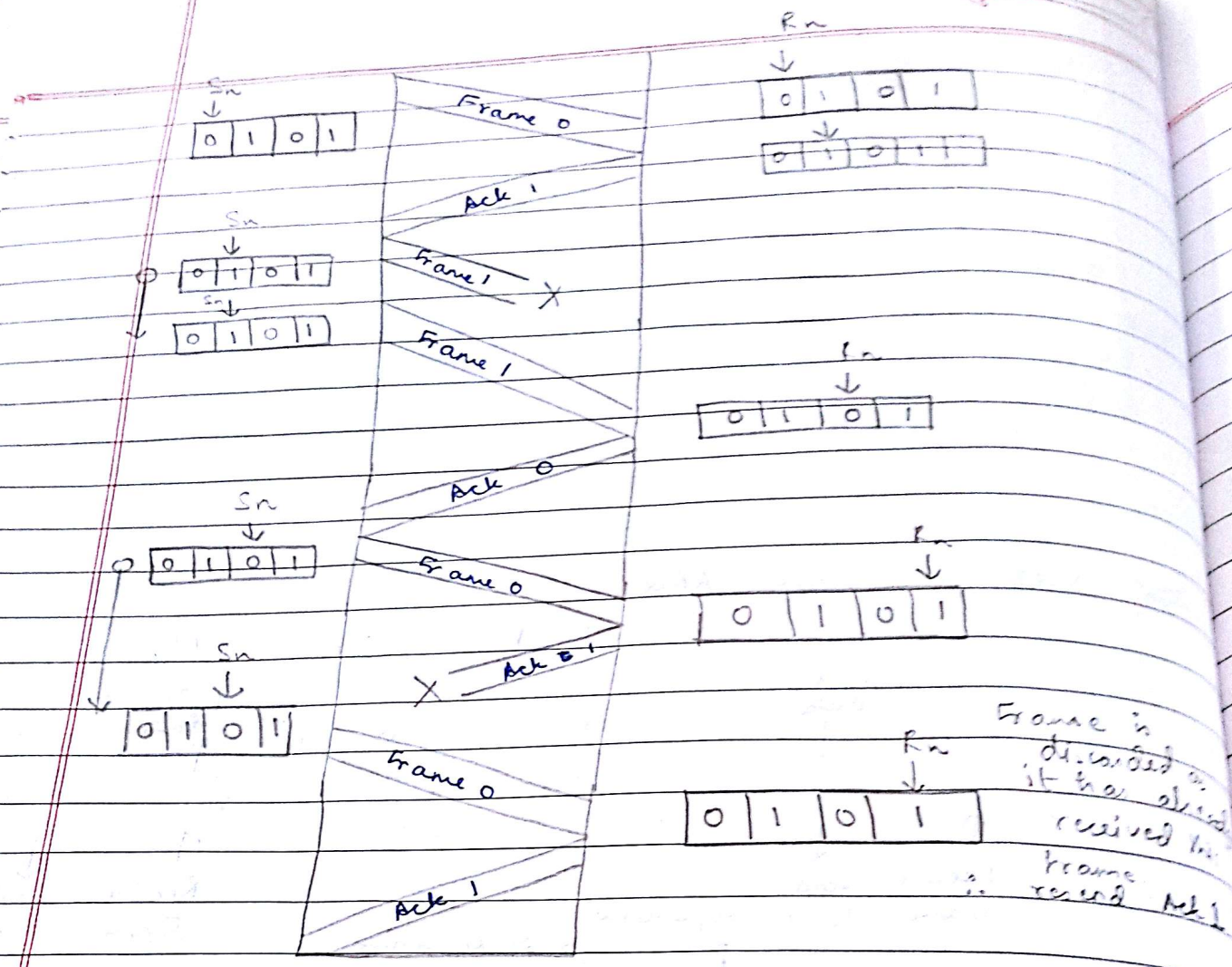
0	1	0	1

S_n

0	1	0	1

R_n

Scanned by CamScanner



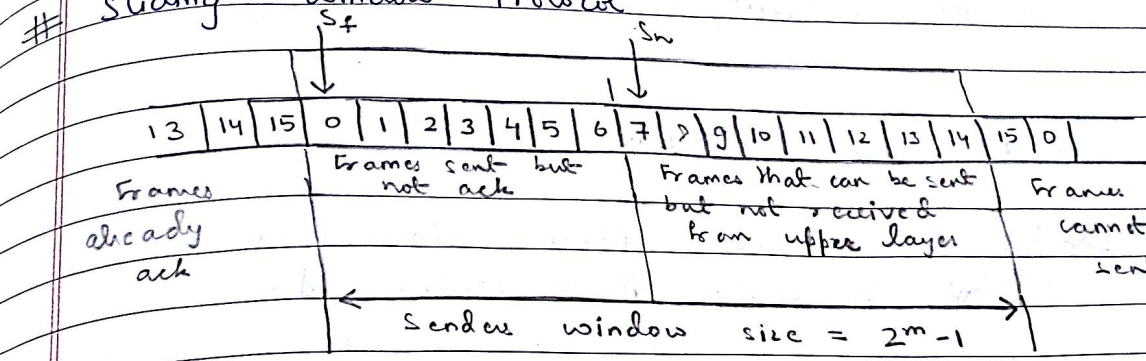
Disadvantage - Even if the bandwidth is high, even then we send only 1 frame.

Scanned by CamScanner

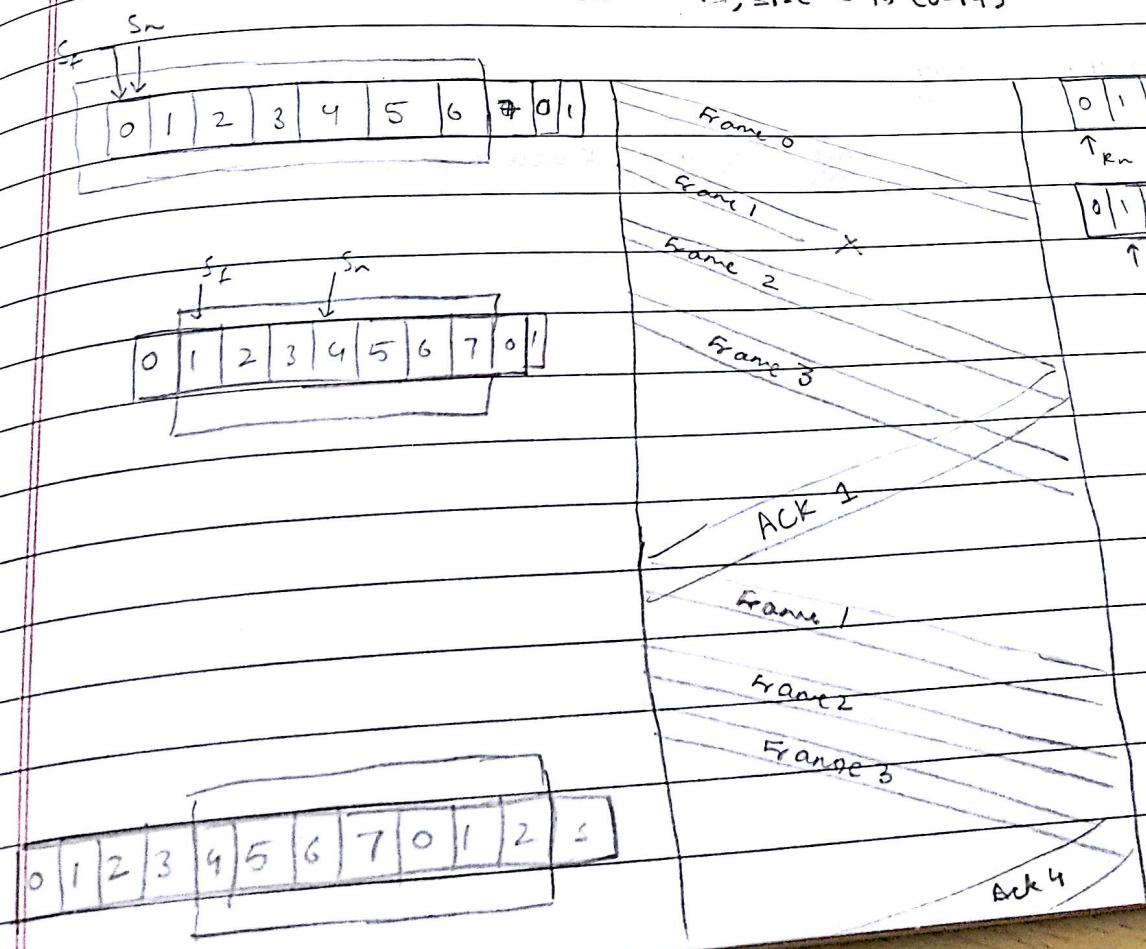
Go back & ARQ

* To improve the efficiency of transmission, multiple frames must be in transition while waiting for the Ack. In Go back & ARQ several frames are sent. The copy of these frames are maintained until the ack arrives.
The sequence nos are modulo 2^m where m is the size of sequence no. field in bits.
It uses the concept of sliding window protocol.

Sliding window Protocol



Here m = 4 ; size = 15 (0-14)



Q Why is the size of senders sliding window, less than 2^m, what happens if the size is greater than 2^m.

Soln If windows size is 2^m then the ack sent by receivers window will be 0 which will lead to an ambiguity i.e ~~whi~~ ack is of which 0.

Q Stop & wait ARQ is a spcl case of goback & ARQ. Justify?

Selective repeat ARQ

Sender & receiver sliding window size is 2^{m-1} to receive bits in random order.
??? why! what will happen if it is $> 2^{m-1}$

Piggybacking

Ack No	Sequence No	Data frame

Scanned by CamScanner

HDLC (High level data link control)

HDLC is a bit oriented protocol for comm over p2p & multipoint links

Transfer Modes (Do definition as well)

→ NRM (Normal Response mode)
→ ABM (Asynchronous balanced mode)

Frames

→ I - Frame (Information frame) Data
→ S - Frame (Supervisory frame) Ack, N-Ack
→ U - Frame (Un-numbered frame) no sequence no., control info

Frame Format

I-Frame

Flag (01111110)	Address (of secondary station)	Control	User Info	FCS (Frame check sequence [contain error detect & correction info])	Flag

S-Frame

Flag	Address (regardless of way of comm)	Control	FCS	Flag

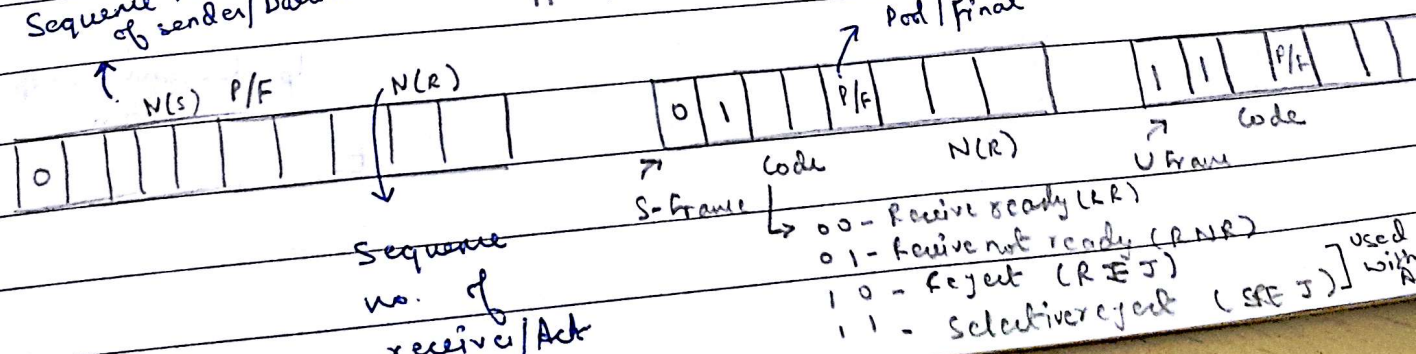
U-Frame

Flag	Address	Control	Managment Info	FCS	Flag

Control → 1 or 2 byte
Used for flow & error control

Control Frame ↓

0 → I frame
01 → S frame
11 → U frame



S-Frame Code:
00 - Receive ready (RR)
01 - Receive not ready (RNR)
10 - Reject (REJ)
11 - Selective reject (SREJ)] used with N

HDLC (High level data link control)

HDLC is a bit oriented protocol for comm over p2p & multipoint links.

Transfer Modes (Do defination as well)

→ NRM (Normal Response mode)
→ ABM (Asynchronous balanced mode)

Frames

→ I - Frame (Information frame) Data
→ S - Frame (Supervisory frame) Ack, N-Ack
→ U - Frame (Un-numbered frame) no sequence no., control info

Frame Format

Frame	Flag (01111110)	Address (Address of secondary station)	Control	User Info	FCS (Frame check sequence [contains error detect & correction info])	Flag
Frame	Flag	Address	Control	FCS	Flag	
Frame	Flag	Address	Control	Managment Info	FCS	Flag

Control → 1 or 2 byte
Used for flow & error control

Control Frame ↓

0 → I frame
01 → S frame
11 → U frame

0	N(S)								N(R)				0	1			P/F				1	1		P/F			

Frame

N(S): Sequence no of sender/Data frame
N(R): Sequence no. of receiver/Ack
P/F: Poll/Final — data send from primary to secondary (Poll) & vice versa (Final)

S-Frame: Code, N(R)
- 00 - Receive ready (RR)
- 01 - Receive not ready (RNR)
- 10 - Reject (REJ)
- 11 - Selective reject (SREJ)] used to deal with negative Ack

U Frame: 1 1 P/F Code Code

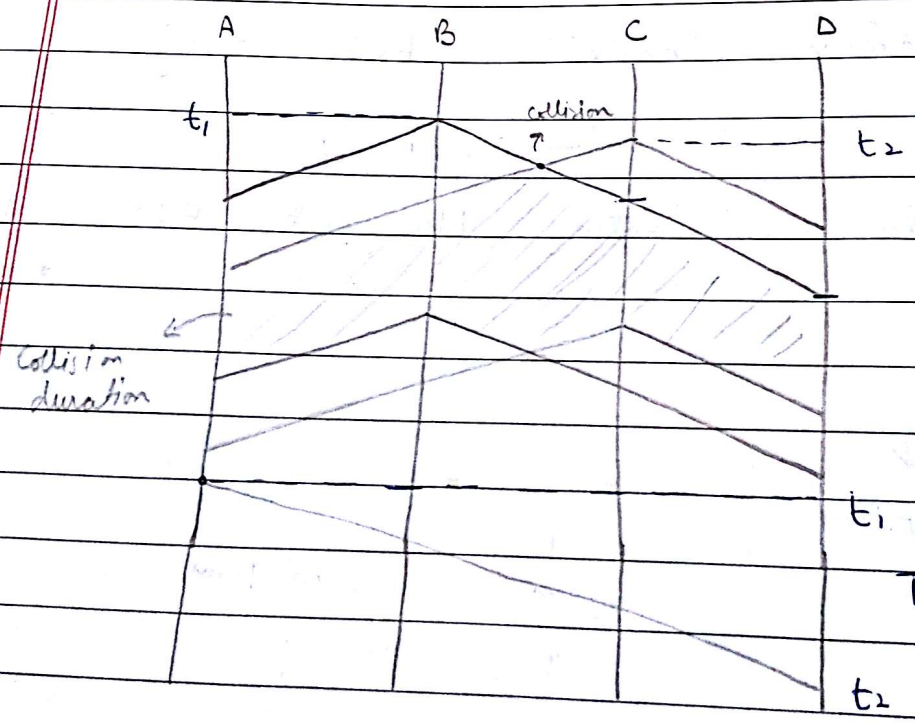
U- Frame contains an info feild used for system managment & not for the user.

☆ PPP ~~protocol~~ (point to point protocol)

Flag	Address	Control	Protocol	Payload	FCS	Flag
1 byte	1 byte	1 byte	1 or 2 byte	variable	2 or 4 byte	1 byte
01111110	11111111 - Broadcast	11000000	defines what type of data is sent i.e User or info or etc			

Notes to be done

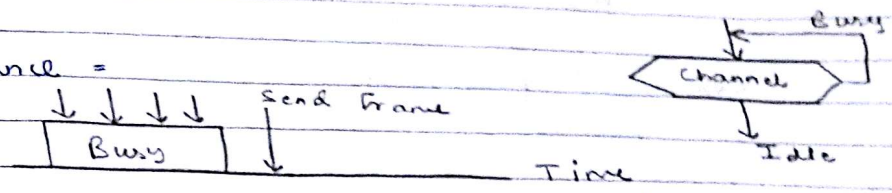
☆ CSMA (Carrier sense multiple access) → CA Collision avoid, → CD Collision detect



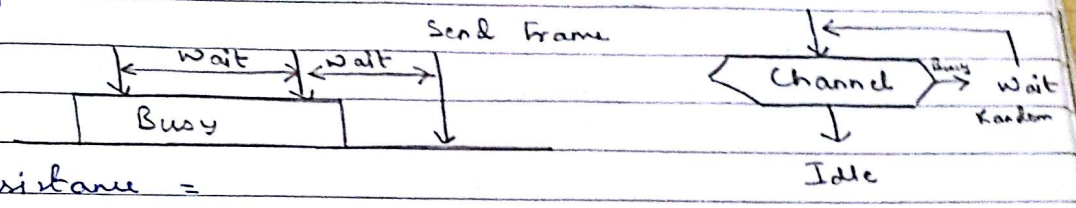
sensing

☆ Persistance Methods

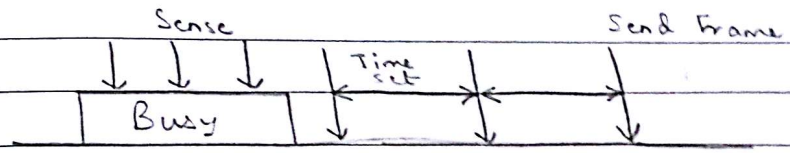
- 1- persistance =

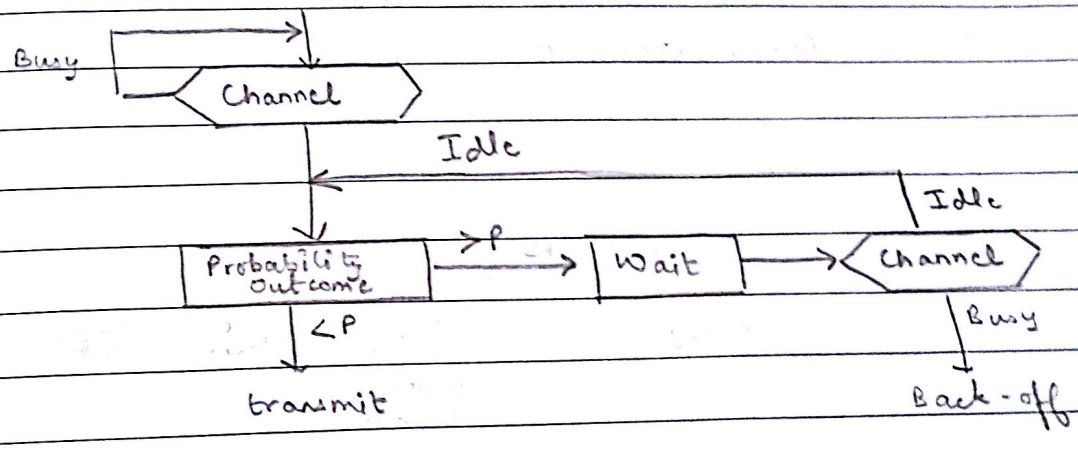


- Non- persistance =

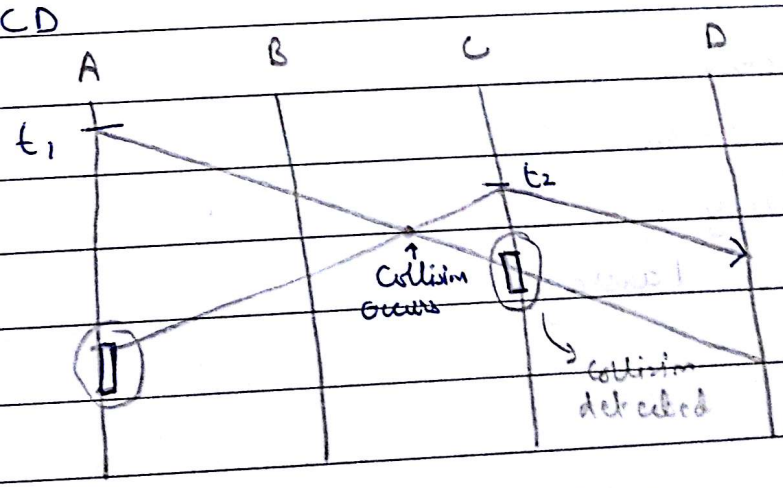


- p persistance =





☆ CSMA/CD



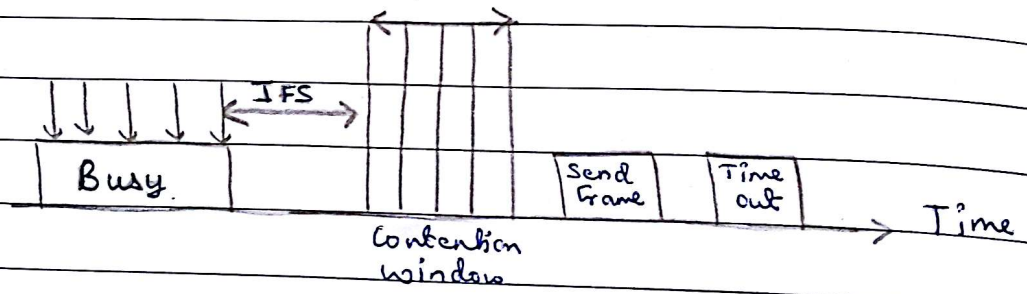
Scanned by CamScanner

~~ALOHA~~	ALOHA	CSMA/CD
• Addition of persistance process		
• In frame transmission & ack		
• Jamming signals		

CSMS/CD - Flowchart from book

★ CSMS/CA

→ Inter frame space (IFS)
→ Contention window
→ Acknowledgement

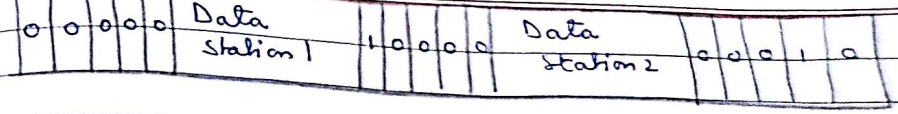


★ Control Access
→ Reservation
→ Pooling
→ Token Passing
→ Channelization

Scanned by CamScanner

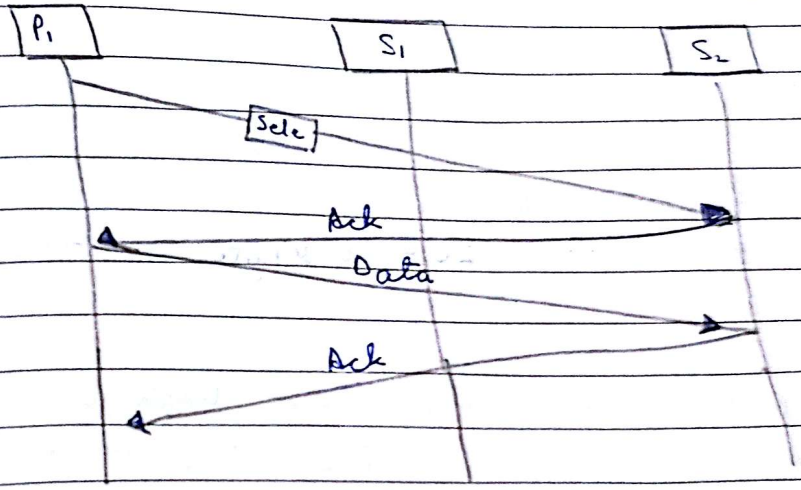
Scanned by CamScanner

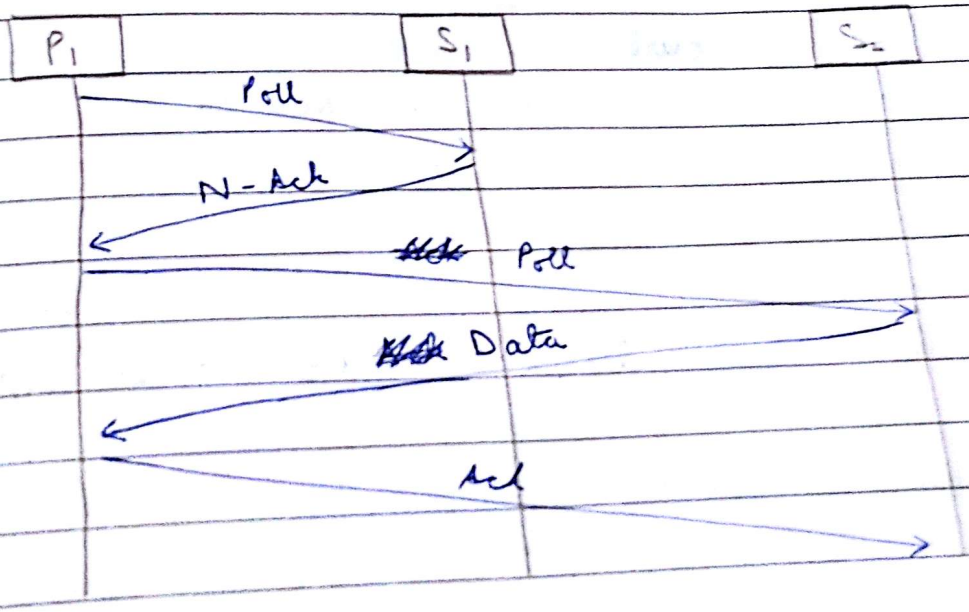
Reservation



Pooling

Uses Pool signal & select message





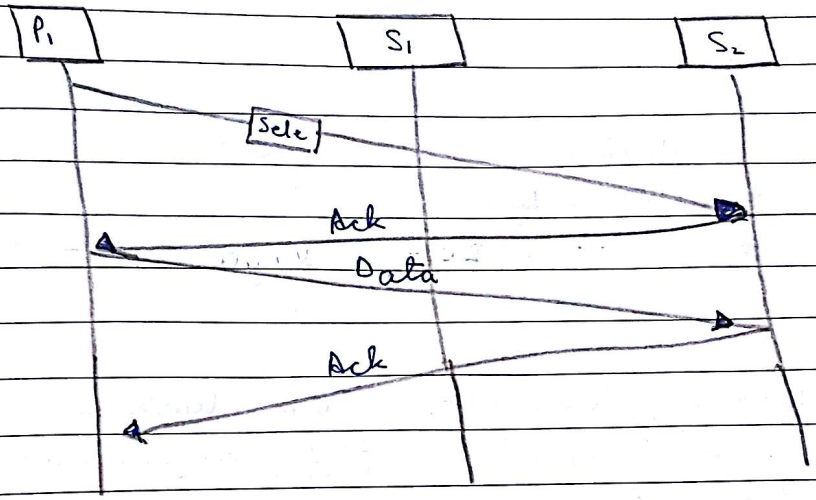
Scanned by CamScanner

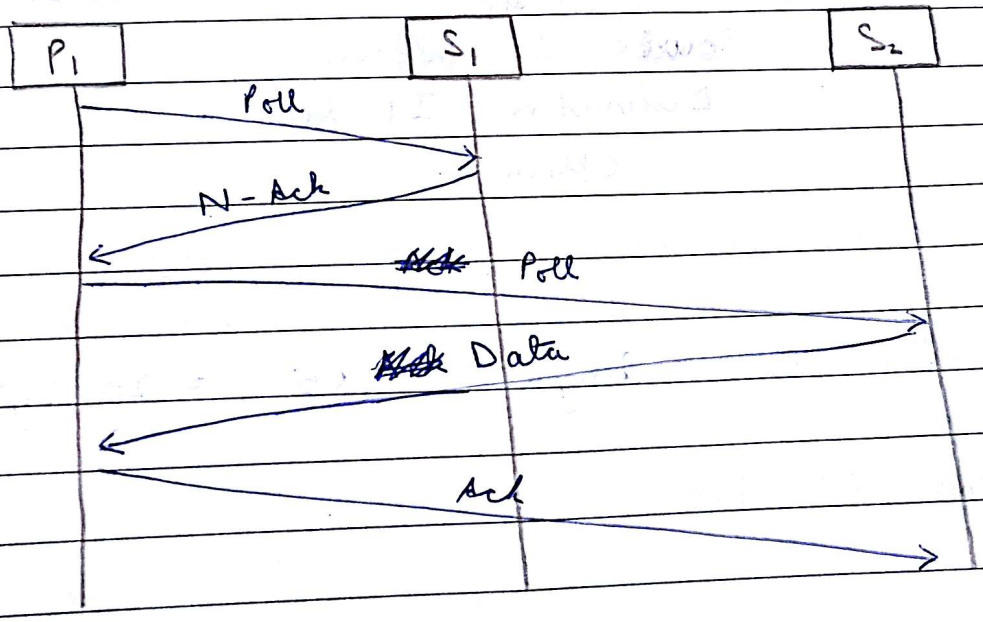
Reservation

0	0	0	0	0	Data Station 1	1	0	0	0	0	Data Station 2	0	0	0	1	0	

Pooling

Uses Pool signal & Select message





Scanned by CamScanner

IPv4 :

- IPv4 is unreliable connectionless datagram protocol.
- It provides no error control or flow control.
- For reliablity, it is paired with a reliable protocol. Eg: TCP
- It follows best effort delivery

Datagram

Header	Data

← 20-60 byte →

← 20 - 65536 bytes →

VER (4 bit)	HLEN (4 bit)	Service (8 bit)	Total length (16 bits)
Identification (16 bit)		Flag (3 bit)	Fragmentation Offset (13 bit)
Time to live (TTL) (8 bit)	Protocol 8 bit		Header checksum (16 bit)
Source IP Address			
Destination IP Address			
Option			

Used for fragmentation

← 32 bit →

Header min length = 4 × 5 = 20 bytes

⇒ Service (8 bit)

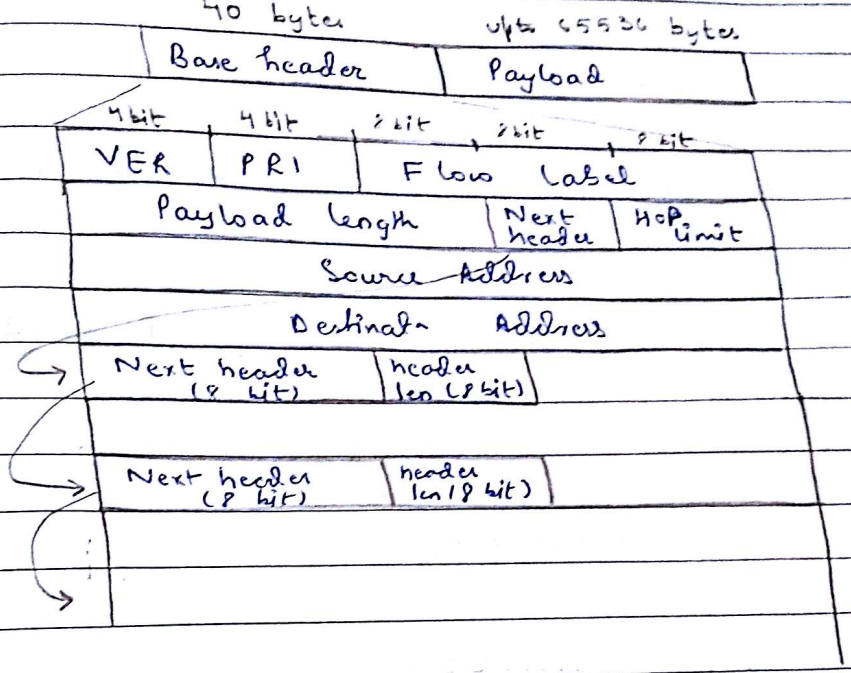
Precedence ← → TOS bits (Type of service) ← →

			D	T	R	C	

D → Min delay, T → Max throughput, R → Max Reliability, C → Min cost

⇒ Total length = length of data + length of header

IPv6 (128 bits)



- Large address space, larger than IPv4
- better header format.
- It supports resource allocation - type of service feild (TOS) has been removed & flow label has bee included.
- It supports more security.
- Total length field is elimentated & replaced by payload length.
- TTL field is replaced by hop limit.
- Header checksum is removed.
- Service type field is eliminated & replaced by priority & flow label fields.

Scanned by CamScanner

A.B.C.D/

Classless Addressing

A.B.C.D/x A.B.C.D/n no. of ~~ones~~ 1's

Classful Addressing is a spcl kind of classless addressing.

Q Let a classless address be 205.16.37.24/29. Find its range.

Sol:

~~Sol~~

24 = 00011000

Default mask = 11111000

111

Q An ISP is granted a block of address starting with 190.100.0.0/16. The ISP needs to distribute these addresses to 3 groups of cust as follows :-

→ 64 customers, each needs 256 addresses

→ 128 " , " " 128 "

→ 128 " , " " 64 "

Design the subblocks & find out how many addresses are still available after the allocation.

Sol: Group 1 :

Cust 1 190.100.0.0/24 ------ 190.100.0.255/24

Cust 2 190.100.1.0/24 ------ 190.100.1.255/24

⋮

Cust 64 190.100.64.0/24 ------ 190.100.64.255/24

64 X 256 = 16384

Group 2

Cust1 190.100.64.0 | 25 ---------- 190.100.64.127 | 25

Cust2: 190.100.64.128 | 25 ------ 190.100.64.255 | 25

⋮

Cust127: 190.100.127.128 | 25 ----- 190.100.127.255 | 25

128 x 128 = 16384

Group 3

Cust 1: 190.100.128.0 | 26 ------- 190.100.128.63 | 26

Cust 2: 190.100.128.64 | 26 ------ 190.100.128.127 | 26

Cust 3: 190.100.128.128 | 26 ---- 190.100.128.191 | 26

Cust 4: 190.100.128.192 | 26 ----- 190.100.128.255 | 26

⋮

190.100.159.192 | 26 ---- 190.100.159.255 | 26

128 x 64 = 8192

Address left = 65536 - (16384 + 16384 + 8192)
= 24576

Scanned by CamScanner

Q. I.P Address = 190.250.0.0/16

Group 1 : 64 customer each need 128 add
" 2 : 128 " " " 64 "
" 3 : 128 " " " 32 "

Congestion control

- Open loop
  → Avoidance measures
  1. Retransmission policy
  2. Window policy → uses selective repeat ARQ
  3. Acknowledgment policy
  4. Discarding policy
  5. Admission policy

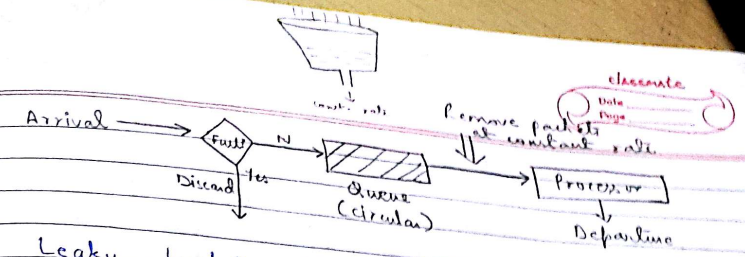
Closed loop
→ Prevention measures
1. Back pressure
2. Choke packets
3. Implicit signaling
4. Explicit signaling
   → Backward
   → Forward

QOS (Quality of Service)

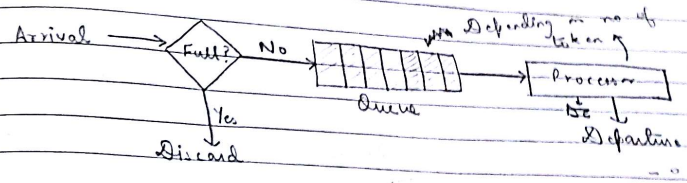
Characteristics
→ Reliability
→ Delay
→ Jitter
→ Bandwidth

Traffic Shaping
- Token bucket algo
- Leaky bucket algo

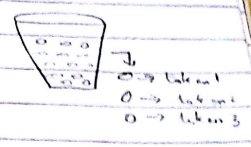
Scanned by CamScanner



Leaky bucket algo :-

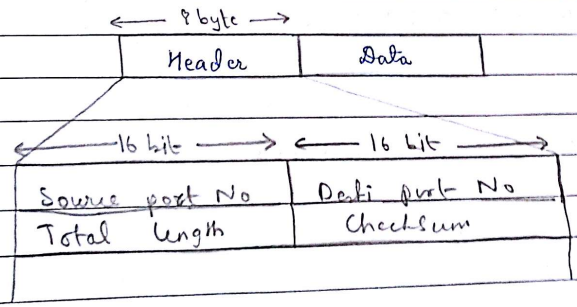


Token bucket algo :-



☆ TCP & UDP :-

UDP :-

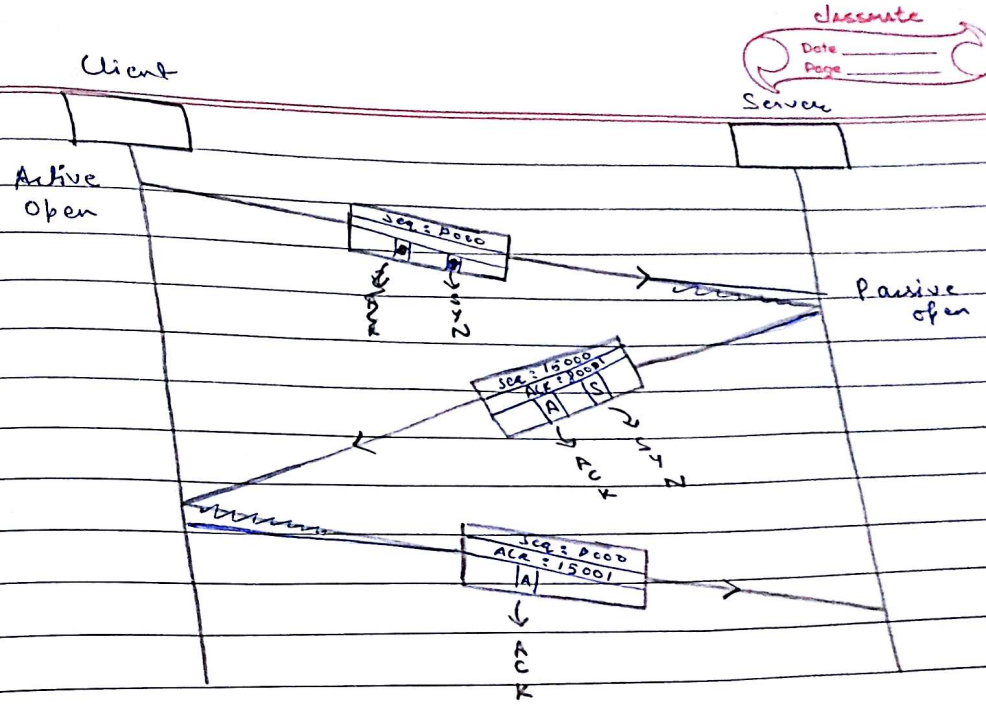


UDP is called connectionless & unreliable transfer protocol. It provide process - process comm & performs very limited error checking.
UDP is suitable for multicasting. It is used for mangment processes such as SNMP (simple N/work manag protocol). It is used for route updating protocol.

Scanned by CamScanner

<table>
<tr><td colspan="8">Sequence no</td></tr>
<tr><td colspan="8">Ack no</td></tr>
<tr><td>HLEN 4bit</td><td>Reserved 6 bit</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="2">Window size</td></tr>
<tr><td colspan="4">Checksum</td><td colspan="4">Urgent pointer</td></tr>
</table>

Scanned by CamScanner



Connection term → Three way handshaking
→ half close