CN short Notes :

CRC

① If $G(x)$ is divisible by $(x+1)$ then it can detect all odd length error

$G(x)$: generator polynomial
$E(x)$: Error polynomial
$C(x)$: code polynomial

At receiver side, $\frac{G(x) + E(x)}{G(x)}$

② If $G(x)$ contain two terms then can detect single bit error

$$= \frac{G(x) + x^i}{G(x)}$$

③ CRC can detect all burst error ~~less~~ less or equal than polynomial length

eg $G(x) = x^n + \dots + 1$

CRC can detect burst error of length $\boxed{n}$

Hamming code

for detection of d errors, the Hamming distance b/w two code word must be $\boxed{d+1}$

$\boxed{d_{min} = s+1}$

error

hamming distance

error

for correction of s bit error, distance b/w two code word must be $2s+1$

$$\boxed{d_{min} = 2s+1}$$

TCP Socket Calls

Server

Socket(): return socket descripter, which type of socket service made be must be made.

bind(): bind self port number

listen(): How many request can accept in Queue

accept(): build the socket, and waits for request

read()
write()
close()

client

socket() → connect() → read() → write() → close()

connect(): send connection request to server by sending SYN=1 segment.

Error control and detection

Bit stuffing (zeal(W))

Error control & ~~Detection~~ (zealw)

A	ESC	B	ESC	C	ESC	ESC	A

If 1 ESC, append 1 ESC
if 2 ESC, append 2 more ESC

Note:- (zealW) If the CRC divisor is n bits then highest power of generator i.e divisor $= n-1$

CRC (forouzan)

(i) Single bit error :- (All)
if $G(x) = x^k + x^i$, then $E(x) = x^i$ can detected.

(ii) Two isolated error - (All)
if (x^k+1) is not factor of $G(x)$, $k \geq 1$
for $E(x) = x^i + x^j = x^j(x^{i-j}+1)$ $\forall\ i-j>1$

two isolated error
$E(x) = x^k + x^{k+1}$
$E(x) = x^k(x+1)$
then $G(x)$ not divisible of x^k or $x^k(x+1)$, $k \geq 0$

(iii) Odd number of error
If $(x+1)$ is a factor of $G(x)$

(iv) ~~Bou~~ Burst up to length r (degree of G)
Burst of length k : $x^i(x^{k-i} + \dots + 1)$
If $G(x)$ is of form ~~$(x^i + x$~~ $\dots + x^0$ then
x^i is not a factor

(v) r-bit generator can detect 1 in 2^r errors

Hamming distance -

$d_{min} = S + 1$ {error detection up to d_{min}}
to be detected no of errors

Hamming distance of block code

to be correct

$d_{min} = 2S + 1$ {error correction up to d_{min}}

$\Rightarrow d + r + 1 \leq 2^r$ {eqn for redundant bit needed for d bit data block}

datablock size, redundant bit size

Checksum (zealW)

Qw- find check sum of 1110 and 1101

(CheckSum)
(zealW)

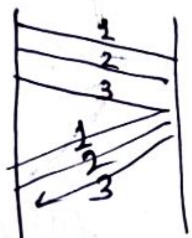
Ans- 1110 {Step① - Add the code block}
+ 1101
-----
1 1011 {Step② → Add carry to result}
+1

1100

1's Compl 0011 {Step③ → complement}

0011 → checksum

#Note:- GBack-N (ZealTS) (ZealT - CN1- 21)(TestID14)(CN-1)



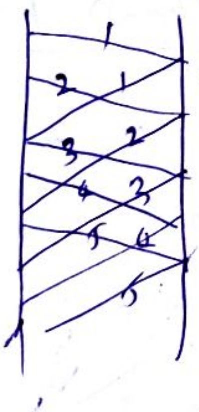
N=3 Window will ~~transmit~~ slide only when ack of first frame is recieved.

if $W_s * t_d \leq (t_n + 2t_p)$ then multiple frame is allowed at one.

[time to send conjucative frame]

[at that time ack will be Recived.]

$W_s t_d \leq (t_d + 2t_p)$



$\Leftarrow W_s * t_d > (t_d + 2t_p)$

#Note:- At Data link layer time out time is used for one frame.

but in Transport layer, time out timer is used for multiple frame.

OSI Model

⇒ (a) Physical Layer (OSI Model)

(i) define characteristic of interface b/w devices and transmission medium.

(ii) defines type of transmission medium

(iii) define transmission rate [# of bits send per second]

(iv) performs synchronisation of sender and receiver clocks.

(v) concerned with the connection of devices to the medium.

(α) point-to-point config. (β) multipoint config.

(vi) physical topology

(vii) direction of transmission

(viii) Deals with Encoding

(a) Manchester (b) Differential Manchester

Data Unit - Bit stream

⇒ (b) Data link Layer

(i) Framing (ii) Flow control (iii) Error control. (trailer added at end)

(iv) Access control

Data Unit - Frame

⇒ (c) Network Layer :-

(i) Logic Addressing (ii) Routing

Data Unit - Packets

⇒ (d) Transport Layer :

(i) Service Point Addressing (v) Connection control

(ii) Segmentation and Reassembly

(iii) Flow control

(iv) Error control Data Unit :- Segments.

(e) Session Layer

(i) Dialog control (establish, manage, terminate connection)

(ii) Synchronization

Data Unit :- Data

(f) Presentation Layer

Data Unit : Data

(i) Translation

(ii) Encryption and Decryption

(iii) Compression

(g) Application Layer

(i) NVT (ii) FTP (iii) Mail (v) Directory Service

Data Unit :- Data

---

there are m nodes and N slots

(i) probability to send frame by any slot is p.

(ii) " " " frame by any ~~slo~~ node $= \underbrace{{}^{m}C_1}_{\text{[1 out of m send]}} \underbrace{p}_{\text{[one success]}} \underbrace{(1-p)^{m-1}}_{\text{[other fails]}}$

(iii) " " " frame to slot n by any one.

so one fail at n-1 slot

$$\underbrace{\left(1 - {}^{m}C_1\, p(1-p)^{m-1}\right)^{n-1}}_{\text{failed in n-1 slot}} \underbrace{{}^{m}C_1\, p(1-p)^{m-1}}_{\text{success in given 'slot n'}}$$ [first time success in slot n]

Pure alloha

One way-propogation $= T_p$
Two " " $= 2T_p$
Round Trip Time (RTT) $= 2 * T_p + T_d$

Medium Acess control

⇒ ALOHA

$$\boxed{\text{throughput} = \left(\tfrac{1}{2}\right)e^{-1} = 0.1839}$$

$$\boxed{\text{throughput in frames/sec} = B \times \frac{1}{L} \times \text{throughput}} = \boxed{\frac{\text{throughput}}{T_d}}$$

⇒ Slotted ALOHA (fully from Zeal (W) questions)

$$\boxed{\text{throughput} = \frac{1}{e} = 0.3679}$$

(Zeal W) Ques. let three slotted Node A, B, C with slots slot 1, ...

① propability of node A at a slot $N = P(A)(1-P(A))^{N-1}$

and probability of A is $P(A)$ given

② probability of first node A success at slot $N =$

P(A fails at first N-1 slot) * P(success of a slot A)

probability of fail in N slot $= (1 - P(\text{success a node}))^{N-1}$

probability of success a node $= p(1-p)^{(\#\text{ of nodes} - 1)}$

$= p(1-p)^2$

$$(1 - p(1-p)^2)^{N-1}\, p(1-p)^2$$

[for Deep Go last page]

CSMA / CD (Read (w))

$\boxed{t_d \geq 2t_p} \Rightarrow \boxed{t_d \geq RTT} \Rightarrow \boxed{L \geq 2 \times T_p \times B}$

$\boxed{t_d \geq 2(t_p + \text{Delay})}$

(Read (w))

In IEEE Manchester encoding ⎍ → 1 and ⎍ → 0

Exponential back off algorithm :-

used to find wait time $= K \times T_{slot}$

↑ random choosen

from $\{0, 2^n - 1\}$ $\forall$ n = no of collisions.

if K is same for both host then collision will be happens.

$P(C_1) = 1$ $P(C_2) = 1$

$P(C_2|C_1) = 0.5$ $P(H_2|C_2) = \frac{1}{4}$

$P(H_1/C_1) = \frac{1}{4}$

success transmit

⇒ using the Exp. back off the probability of 2nd collision after 1st collision will decrease.

⇒ Capture Effect :- winning hosts have the highest probabity of winning again.

Solution :- use collision free algorithm : Token Passing

$P(H_1|C_2) = 5/8$

$P(H_2|C_2) = 1/8$

$P(C_3|C_2) = 1/4 = \frac{2}{8}$

Packetization (obr)

let N packets have to transmit through n hops and tasmission delay is td for one hop is td

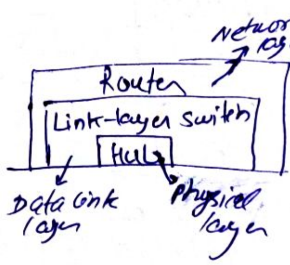
total time = $(N-1)t_d + nt_d$

Switch :- $\boxed{\text{Empty tabled Switch works as Hub}}$ (zeal ce)

(Yoozan)

Hub :- ① Hub are also worked as Repeater when used in star topology
② Hub broadcast the frame to all other port except from received.
③ A repeater has no fitering capability.
④ A Hub or a repeater is a physical-layer device.

Switch
① operates in both physical and data-link-layer
② regenerate the received signal
③ switch has a table used in filering decision.
④ Check MAC address contained in the Frame
⑤ don't change MAC in a frame.



	seprate devices	Not sepr
Collision Domain	Switch, Router	Hub
Broadcast Domain	Router	Hub, Switch

DVR

Problem → (1) Count-to-infinity

Solution - (a) Split-horizon

(b) Poison Reverse

LAN-technology [RBR notes]

⇒ Ethernet (IEEE 802.3)

(a) topology - Bus (b) Access control Method - CSMA/CD (c) NO ACK

(d) Encoding - Manchester (e) Operates at Data link layer

AL - Message

TL - Segment

NL - Datagram

DLL - Frame

PL - Single protocol Data Unit (I-PDU)

ETHERNET Frame format

7B (Preamble)	1B (SFD)	6B (DA)	6B (SA)	2 Length	Data	CRC 4B

64 bytes (from DA to CRC)

PL (Preamble, SFD) | DLL (DA, SA, Length) | DLL tail (CRC)

SFD - (Start frame Delimeter) (Indicate start of frame)

DA, SA → MAC Address

Type of Mac Address (48 bit)

(1) Unicast [LSB of 1st byte is 0]

(2) Multicast [1st Byte LSB 1's]

(3) Broadcast [all bits are 1's FF:FF:FF:FF]

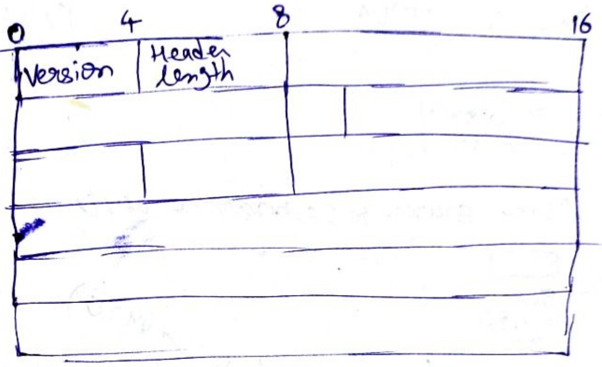
DATA	46B	1500B
FRAME	64B	1518B
	MIN	MAX

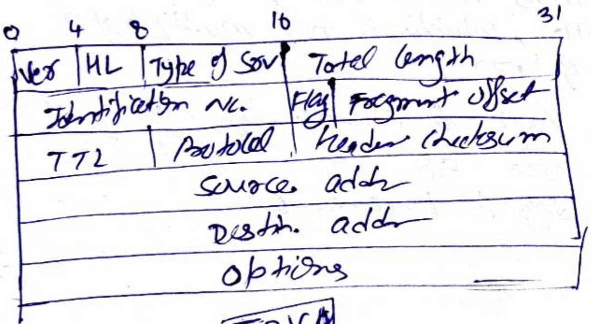
There are two Path

Path① A → B → C → D cost : 40

Path② A → E → D cost : 40

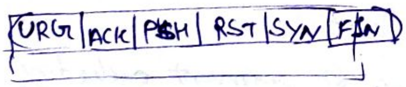
Path② is most preferred, because of less path distance nodes.

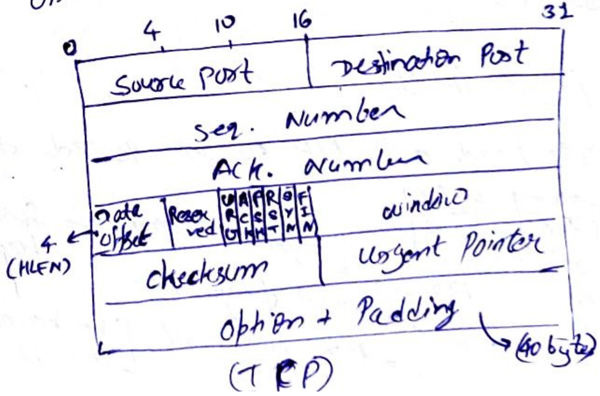




IPV4

Very high stronger Talking
Indian FLAG Frame FLAG
Tere Papa Likhchai
SPA
DPA
options





(TCP)

Fragmentation in IPV4

[Note:- should cut in multiple of 8]

|20| 1000| ⟶ |20| 480|

← 1020 → ← 500 →

Step (1) cut header from source & Destination MTU

|1000| | |

send 500 − 20 = 480 recv

Step (2) cut the send Data into chucks of '480' (MTU) and must have multiple of 8

e.g if given 476, which is not multiple of 8 then take chuck of '472'.

|480| |480| |40|

Step 3 give fragment number & add padding in last chunck if not multiple of 8.

e.g last chunck size 58, then add padding to be come 60

|480| |480| |40|

0 − ~~59~~ ~~60~~ − 119 120 124

fragment no. ~~last~~ |0| |60| |120|

Step 4 add IPV4 head and MF & DF & Id to each chuck

	← 500 →	← 500 →	← 60 →
	|20|480|	|20|480|	|20|40|
Identification	27	27	27
fragment	0	60	120
DF	0	0	0
MF	1	1	0

fragment number :- first byte starting of segment excluding Headers

→ Don't fragment

→ More fragment [1 - have more fragment after this fragment; 0 - It is last fragment]

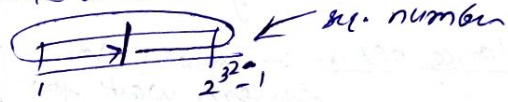
UDP Packet

0 15	31
SRC PORT	DEST. PORT
LEN(H+D)	UDP CHECKSUM
Application Data	

wrap Around time

wrap Around time :— total time required to used same sequence number after some time at given band width B.

e.g.



Note: If $\boxed{WT > LT}$ ⇒ then we don't need to worry
If not then we add some extra bit in options

- LT - life time
- WT - wrap time

Extra bit required for sequence number called timestamp

$\boxed{\text{wrap Ar. Time} = L.T \times B}$

$\boxed{\text{bit of seq. number} = \log \lceil \text{wrap Ar. Time} \rceil}$

efficiency of TCP

$$\boxed{\eta = \frac{w \times t_x}{RTT}} \qquad \left\{ t_x = \frac{L}{B} \right\}$$

if $\eta = 100\%$ then

$1 = \frac{w \times t_x}{RTT} \Rightarrow RTT = w \times t_x$

$\Rightarrow 2 \times T_p = w \times \frac{L}{B}$

$\Rightarrow \boxed{2 \times T_p \times B = w \times L}$

This term must be in byte.
Here $w = 1, 2 \dots N$
L = in Bytes

default window size is 2^{16}

option win size is 2^{14}

required $\boxed{\text{window size} = 2 \times t_p \times B}$ ← advertised window

window size converted into Byte

$\boxed{\text{bits in number} = \log \lceil \text{window size} \rceil}$

Advertised window in bytes

CSMA/CD Protocol

Persistance CSMA

(i) 1-persistance – If bus is free then definitely transmit data ($p=1$) [continue sense]

(ii) p-persistance – If " " " " p is probability to transmit data.

Non-persistance CSMA :- System starts sensing & bus not free than system wait for random time & again start sensing.

probab. of collision ⇒ | 1-pers. > p-pers. > Non-pers |

$$\boxed{\eta = \frac{1}{1+5a}}$$ ⇒ efficiency of CSMA/CD

TCP-Header

Source Port -
- 16 bit (unique)

Destination Port
- 16 bit (unique)

Seq. number
- 32 bit
- seq no. of first data byte

Ack. Number
- 32 bit
- last received data byte + 1.

Header length
- 4 bit
- [20B, 60B]
- Head field × 4 byte (scaling)

Reserved bits
- 6 bits

= Urg bit
used to treat data on urgent basis
- urgent data sent to application layer

= Ack bit

= PSH bit
- All segments in the buffer are pushed to receiving application

= RST bit
Reset the conn

= FIN bit
Terminate the conn.

= window size
- 16 bit

checksum
16 bit

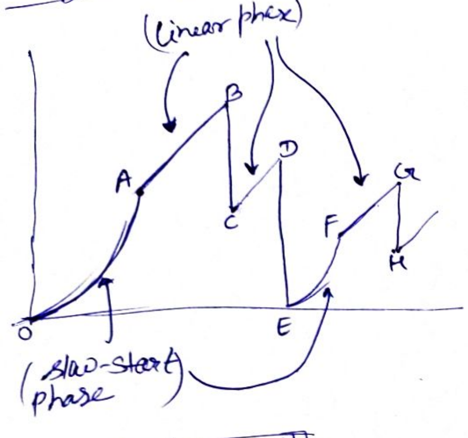
urgent pointer
- 16 bit
- How much data in current segment counting from the first data byte is urgent
- indicate the end of urgent data byte

Number of urgent bytes = urgent pointer + 1

End of urgent byte = seq. no. of first byte of segment + urgent pointer.

options
- 0-40 bytes
= Timestamp
= windowsize extension
= Parameter negotiation
- Maximum segment size on sync time.
= Padding

congestion control



Ⓐ SS-threshold value

Ⓑ Three acknowledgement lost

Ⓒ after 3-ack-lost

→ SS-threshold $= \frac{w}{2}$ ← (current window)

→ congestion window $= \frac{w}{2}$

Ⓓ Timeout timer lost

Ⓔ After timeout timer

→ SS-threshold $= \frac{w}{2}$ ← (current window)

→ congestion window = 1 mss

⇒ after each transmitt

slow start phase :- in each trasmission window get double

linear phase :- in each transmission window size get increment by 1.

⇒ after each ack

slow start phase = $\underset{\text{(In bytes)}}{C_{wind}} + \underset{\text{(In bytes)}}{1\ mss}$ (in each ack recived)

linear phase = $\boxed{\cancel{C_{wind} + \frac{1}{C_{wind}\text{(in mss)}} \times 1\ mss}}$ $\underset{\text{(In bytes)}}{C_{wind}} + \frac{1}{\underset{\text{(In MSS)}}{C_{wind}}} \times 1\text{MSS (In bytes)}$