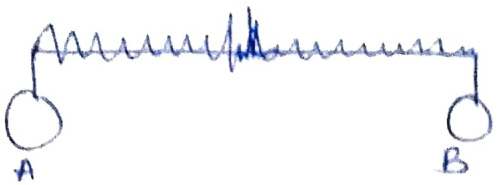
Back off Algorithm (CSMA/CD)



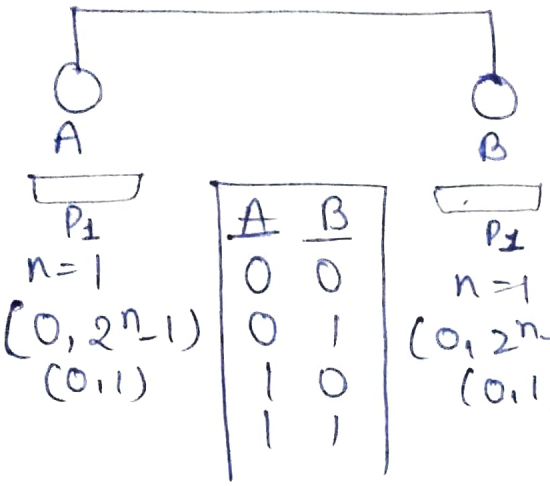
If A and B wants to retransmit data after detecting collision then when should they start.

In Back-Off Algorithm we use waiting time that how much time station wait to send the data again

n → collision number.

If both start transmitting at same time then no. of collision occur

n = 1



So, according to Algorithm A,B can choose any one number from $(0, 2^{n-1})$.

Let say we choose 0,1 so,

A should wait → 0 * Tslot = 0

B should wait → 1 * Tslot = 1

Using time slot we can find waiting time.

→ Instead of (0,1) we can choose any combination.

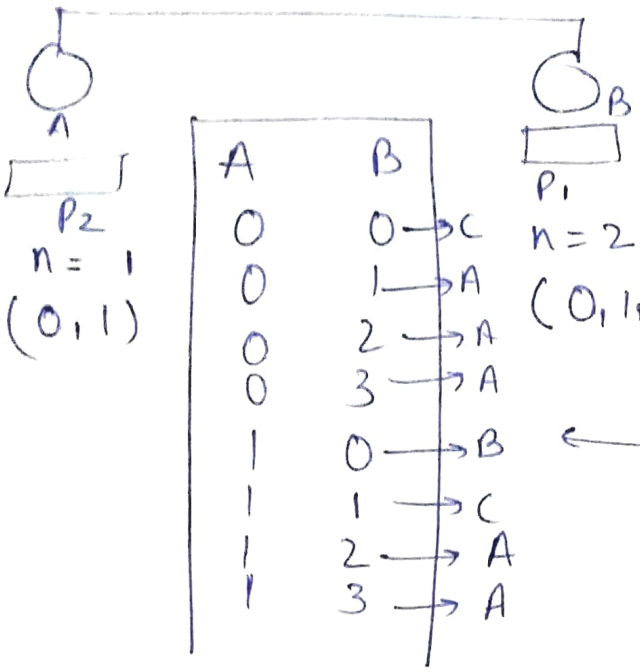
P(win of A) = 1/4 (0,1)

P(win of B) = 1/4 (1,0)

P(next collision) = 2/4 (1,1) (0,0)

→ Win of A means A will sent packet and B will wait for the Tslot.

Let us say after (0,1), B have its one (first) packet and A is ready to send 2nd Packet as



Again collision occur but this time first packet of B collides 2nd time with 2nd packet of

Vul A < Vulh

P(collision) = 2/8 = 1/4 = 25%

P(A won) = 5/8

P(B won) = 1/8

→ **Waiting time = K + Tslot**

K → randomly choose no. from $(0, 2^{n}-1)$

n → collision no.

→ The station won's the chance to transmit data first time its probability to transmit data next time get increases (station A)

This effect is called as 'capture effect'.

→ Probability increases or decreases exponentially.

→ This method (Back-off) applied for only two stations, so called as Binary-exponential Back-off Algorithm

Token passing —: Generally we measure time in seconds, but sometimes we measure it in bits.

→ 10 bits means time taken to transmit 10 bits.

→ If time given in bits called bit time and to convert it to sec do,

$$(time)_{sec} = \frac{(time)_{bit}}{Bandwidth}$$

→ If time given in 'm' then to convert in sec do,

$$(time)_{sec} = \frac{(time)_{metre}}{velocity}$$

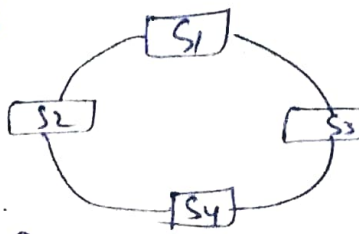
Q How many metre in 10 bits equivalut to B = 4 mbps & v = 2×10^8 m/s

$$\frac{10}{4 \times 10^6} \times 2 \times 10^8 = \text{metre (time)}$$

$$(time)_{metre} = 500 \text{ metres}$$

In token passing Advantages

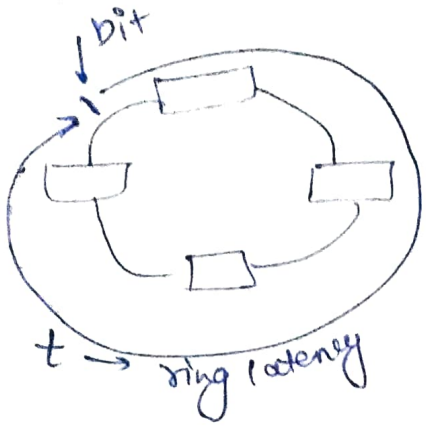
Ring Topology



→ the station that have token can transmit the data.

→ Data can transmit only in one direction.

Ring Latency → Time taken by a bit to complete a round in ring.



Ring latency = $\frac{d}{v} + (N * b)$

(N*b) time spend by the bit at all the station

N → No. of station

b → bit time (at station)

$(\frac{d}{v})$ → seconds (N*b) → bits.

So, $\boxed{\text{Ring Latency} = \frac{d}{v} + \frac{N*b}{B}}$ → seconds

or $\boxed{RL = \frac{d}{v} * B + N*b}$ → in bits

Cycle Time → Time taken by token to complete a round in ring.

(THT) Token holding time → time taken by station for which it holds the token.

Cycle time : $\left(\frac{d}{v} + N*(THT)\right)$

$= (T_p + N*(THT))$

Efficiency = $\frac{\text{Useful time}}{\text{Cycle time}}$

→ No. of packet transmit in a single cycle = N

→ N bcoz each station will receive a packet from token.

time require to send N packets = $N * T_t$

$\boxed{\eta = \frac{N * T_t}{(T_p + N * THT)}}$

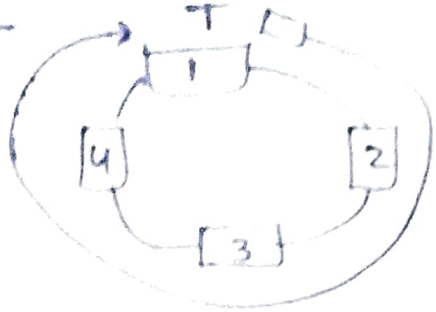
Token passing can be done in 2 ways

(1) Delayed token reinsertion (DTR)

(2) Early token reinsertion (ETR)

$$\eta = \frac{N * T_t}{T_p + N * THT}$$

(1) DTR



Token is at 1 at data is transmitted and waiting for data packet to come back.

$$THT = T_t + \text{Ring latency}$$

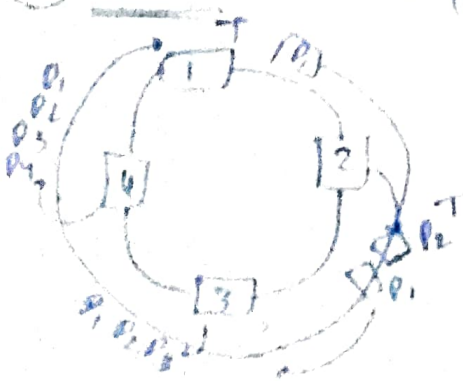
$$= T_t + T_b + \cancel{N * b} \to 0 \quad \text{[Just for sake of simplicity]}$$

$$\Rightarrow T_t + T_p$$

$$(\eta)_{DTR} = \frac{N * T_t}{T_b + N * (T_t + T_p)}$$

$$\Rightarrow \boxed{\frac{1}{1 + \frac{(N+1)a}{N}}}$$

(2) ETR



Token release immediately after transmitting packet at each station.

→ Each station have responsibility to remove its own packets from the link.

→ In second round each station will take off its own packets

$$THT = T_t$$

$$\eta = \frac{N * T_t}{T_p + N * T_t} = \left(\frac{N}{a+N}\right)$$

So,

$$(\eta)_{ETR} > (\eta)_{DTR}$$

→ If not given in exam go for ETR always.

Aloha and difference between flow and access control :- [Only remember the formula and value]

Aloha was used in Japan long back.

→ No Career sensing.
→ Collision Possible
→ Because of Acknowledgement no need of collision detection.
→ Retransmission of data after a random amount of time (Back off time).

Aloha types

→ Pure
→ Slotted

* Pure Aloha - Anyone can send data any time.

$$\eta = G * e^{-2G}$$

G → no. of stations who wants to transmit in T_t slot.

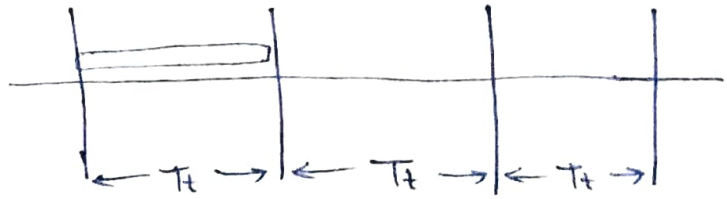
$$\frac{d\eta}{dG} = 0 \Rightarrow G = 1/2$$

$$(\eta_{max}) = \frac{1}{2e} \quad [At\ G = 1/2]$$

$$= \underline{0.184}\ (18.4\%)$$

⇒ η is very less due to lot of collision.

Slotted Aloha :



Every station can transmit in its time slot if it miss then it will wait for next slot.

$$\eta = G * e^{-G}$$

$$\frac{d\eta}{dG} = 0 \Rightarrow G = 1$$

$$(\eta)_{max} = \frac{1}{e} = 0.368$$

$$= \underline{36.8\%}$$

Imp. Question Day-4

→ Difference between flow control and access control. (Just read the answer to distinguish from Internet)

FlowControl

Stop and wait : $1/2a+1$

Go Back N / SR : $N/1+2a$

Access Control

TDM = $1/1+a$

Polling = $\frac{T_t}{T_t + T_{polling} + T_p}$

CSMA/CD = $\frac{1}{1+6.44a}$

Token → ETR : $1/1+(a/N)$; DTR : $1/1+\frac{a(N+1)}{N}$

Aloha → P : $G * e^{-2G} = 18.4\%$; S : $G * e^{-G} = 36.8\%$

Gate-2016 :- Question on slotted Aloha

Soln

P(A sending * P(B, C, D Not send)
+ P(B sending) * P(A, C, D Not send)
+ P(C sending) * P(A, B, D Not send)
+ P(D send) * P(A, B, C Not send)

= 0.4404

→ 0.462 is answer given because that is very close one

Error Control Methods

Error Control and CRC

Error may occur due to

① Packet lost (Burst error)
→ Buffer of router get lost.

② Bits get corrupted due to any reason. (Bit error)

If packet have some corrupted bits and gets received at another end. How can you handle it?

Error Handling

Error Detection (Only knows there is some error)

→ send two copy of same data & if one copy get corrupted you can detect the corrupt data. But this method doesn't use practically

→ Parity bits checking

→ CRC (Cyclic Redundancy check)

→ Checksum

Error Correction (know where the corrupt bits exits)

→ Hamming Code

→ ~~Hamming~~ Code have disadvantage-
① Send redundant data
② More time taken

→ CRC and checksum are mostly used in Computer Networks.

① **CRC** → (Cyclic Redundancy Check)

Msg = 0110101 no. of bits
Received = 0010010 gave error
⊕ 0100100 → 2 set bit

Sender   Receiver

1011011 | 000

| 1101 |
CRC Generator

If CRC generator is of n bits add (n-1) zeroes at the end of sender's bits.

Ex-OR also called sum mod 2 means add both bits and devide by 2 and get remainder

1 ⊕ 1	1 ⊕ 0	0 ⊕ 1	0 ⊕ 0
2%2	1%2	1%2	0%2
0	1	1	0

1 ⊕ 1 ⊕ 1 → 3%2 = 1   and so on many more

To get CRC using CRC Generator do X-OR between sender code and CRC generator

```
1101 ) 1011011000
       1101            (+)  X-OR
       0110011000
        1101
       00 0111000
          1101
          0011 00
            1101
            0|001|
```

Leading 1's को below से हटाना है each time.

Last (n-1) bits are CRC.

1011011 001
Data — Last three bits replaced with CRC.

```
1101 ) 1011011 001
       1101
       0110011001
        1101
       00011 1001
          1101
          001101
            1101
            0000
```

If (n-1) bits are all 0's with CRC means message is not corrupted.

→ If any bit get corrupted in message then we will not get all zeroes at the end.

→ CRC is able to detect but not able to correct it.

Ex → $x^3 + x + 1$

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

| 1011 |

Generate bits of CRC from polynomial.

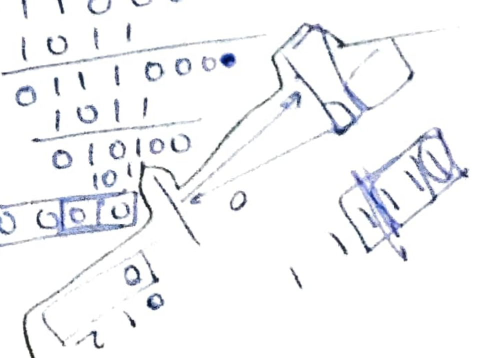
At sender

Data = 11010

CRC Gen. = $x^3 + x + 1$ = | 1011 |

11010 000 — (n-1) bits append

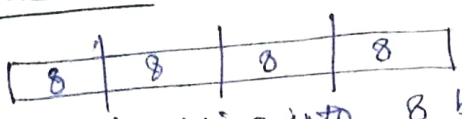
```
1011 ) 11010000
       1011
       01100000
        1011
       0111000
         1011
        010100
          101
       00|010|
```



At receiver 11010010 (obtained message)

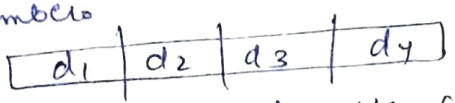
→ Above message will obtain if there is no errors

Checksum [DO atleast 3 questions for practice]



→ 8 bits checksum.

After devriding into 8 bits encode every 8 bits in decimal number

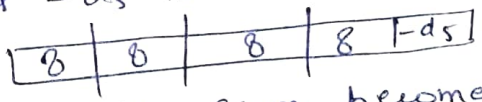


→ $(d_1 + d_2 + d_3 + d_4) = d_5$

then take 1's complement of d_5

(1's complement) $d_5 = -d_5$

then add $-d_5$ to end of msg



If instead of 8 bits sum becomes 9 bits number then do wrapped around (add extra bit as carry).

→ checksum can be of any bits (either 8, 16, 32) but here we take example of 8 bits.

→ Internet checksum is always 16 bit.

Summary

No any error detection and correction method is 100% reliable.

In CRC and checksum we take k bits are error handling bits

Assum data of n bits then no of data set possible over n-bits are 2^n but we have only k bits and $k << n$ then $2^n >> 2^k$. So during mapping of data and bits after sometimes (2^n) gets ended and start repeating the sequence for remaining data set in (2^n).

(many-one function)

So we can do that for n bit data send n-bit error handling means 2n bits needs to send and it will create a high redundancy and more data needs to transmito

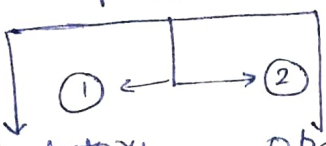
So no method is 100% reliable.

ISO/OSI stack

① ISO/OSI Layers

OS provide interprocess comm^n btw two 2 process of same host but in CN 2 process are in different hosts connected via Internet

functions (of every process)



Mandatory	Optional
1. Error Control	1. Encryption & Decryption
2. Flow Control	2. Checkpoints
3. Access Control	3. Routing
4. Multiplexing & Demultiplexing	4. many more
5. Addressing	

and many more

① ISO/OSI Model
② TCP/IP
③ ATM
④ X.25
⑤ IEEE and many more

But we have only first two in syllabus.

ISO/OSI (International standard Organisation)/ Open System Interconnect

AL
PreL
SesL
TL
NL
DLL
PL

we devide all the functions in 7 layers.

① we can use Devide and conquer
② Encapsulation is possible.
③ Abstraction is possible

PL → Only deals with H/W
DLL → Both H/W and S/W
NL → Complex (Routing Algo.)
TL → Thick Layer (More functions)

AL, PresL, SesL → Exist for User convenience

Please DoNot Touch Sachin Pathl Anjali

→ More Imp
(Network and Transport Layer)

Physical Layer

Electrical, Mechanical, functional and Procedural characteristic of physical links.

Copper wire → signal to electrical signals
Optical wire → signal to light
wireless → EM waves.

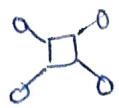
functionality

Transmission
- simplex
- Half Duplex (Walky Talky)
- full Duplex

→ In Gate assume bydefault full Duplex.

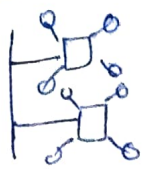
Topologies:- ① Bus

② Star



③ Ring

④ Hybrid



⑤ mesh

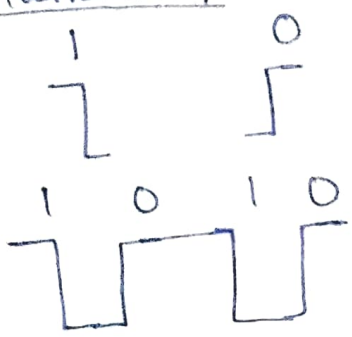
Complete graph

Encoding

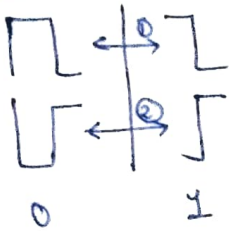
→ 0 or 1 to waves.

1 0 1 0 → seq. of bits

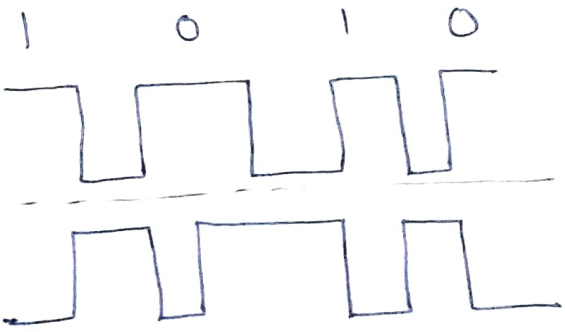
Manchester



Diff. manchester



zero is always going to start with edge



→ Both the diagram are correct

Baud Rate = 2 × Bit Rate.

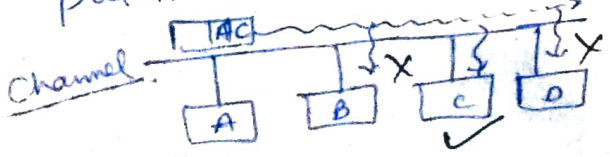
↳ No. of voltage sends per seconds

Data Link Layer (DLL)

1. flow Control (S&w, GBN, SR)
   → Most imp. GBN.
2. Error Control (CRC, checksum)
3. Framing
4. Physical Addressing
5. Access Control (CSMA/CD, Token passing)
   ↓ Ethernet ↓ Tokenring
6. HOP to HOP connectivity

framing at DLL

Take data from Network layer, put it in a frame and send it.



Packet reaches to everyone but B and D discard it to bcoz it hold the destination address of C.

Every sender/receiver just see the beggining (destination address) of the frame and if message उसका नहीं है तो receive करता।

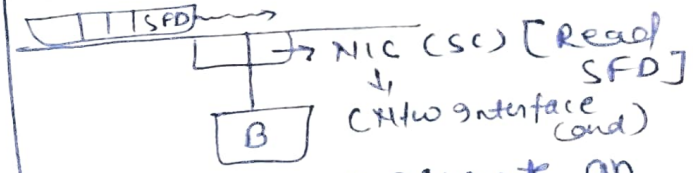
→ start of the frame delimeter

SFD is a pattern different from data.

SFD :- (1010 ... 11) ↔ (1+0)* (Regular exp.)

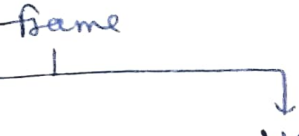
RE → NFA → DFA → SC.

Sequential Ckt (SC) can recognize the SFD



Using SFD we generate an alert among all that there is a message checked either it's for you or not.

But along with begginning it should have an end.

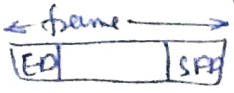


(No need to send end of frame bcoz fixed length is given)

→ If you send less than 1000 b it will attach some dummy bits, process called padding.

→ Must say about end of frame just like start of frame.

→ ① add length feild to say about end of frame
→ ② add end delimeter



Generally we use variable length frames.

→ length (Ethernet) (LAN's)
→ ED (Token ring)

-Disadvantages

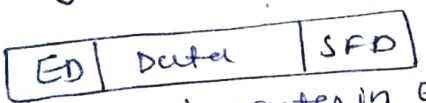
① In length method length is given before data and if some of the data bits get corrupted then it should fails bcoz it will not match up with actual data size.

② In ED if pattern of ED get match with data byte then create a problem that end will occur just before looking at ED (because rest bits of ED same as ED).

ED method

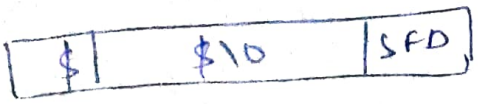
- Character stuffing (CS)
- Bit stuffing (BS)

① CS :-



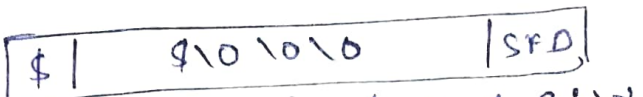
Use those character in ED that is not present in keyboard because those keys which are not present in keyboard never present in Data.

In earlier $ was not on keyboard so that time $ was used as ED but then after some time all the keyboard character started to use as ED too using various method



If a character preceded by '10' means that character is also ED so read $'10' $ as bit of message.

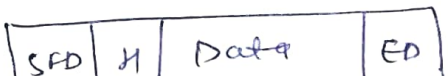
But it creates a problem that if '10's in bit sequence, How can we identify them

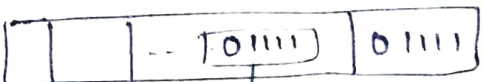


add a '10' for $ and a '10' for '10' too, to make a difference.

→ But it was costly so not in use right now.

② Bit Stuffing :- (Imp for Gate).





Sender Data :- 0 1 1 1 0 1
Add a '0' to break pattern.

Receiver Data → 0 1 1 1 1
Remove 0 and read the data.

But if data is '01110' it will create a problem then receiver thinks that you enter that '0' and receiver will remove that '0'.

To remove this problem you have to add a '0' for '0' in message as—

Data : 01110

now treat the same data send as

0 1 1 1 0 0 1

to distinguish that '0' is in pattern.

Add a zero (stuffed bit)

Ex → ED: 011111

Sender : 0111101

and if at sending side pattern 011110 exist then you have to add

0111100

Add a zero

because you don't know that after the '0', there may be a 1.

(2) ED: 01111

Data: 011100011110

Data after stuffing

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

Added zero

Reason for why I add 0 before last '1' in 01111 as 01101

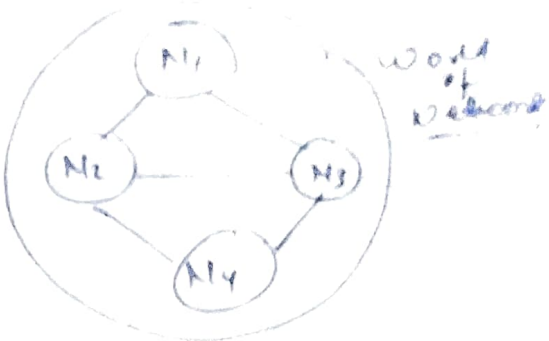
Reason is we want to break the sequence 0111 if we add 0 after 1st one 00111 then any exist lot of pattern with 01 00 randomly so to choose a pattern that have less probability of existence we choose the last bit to shift after zero.

Physical Addressing

Address

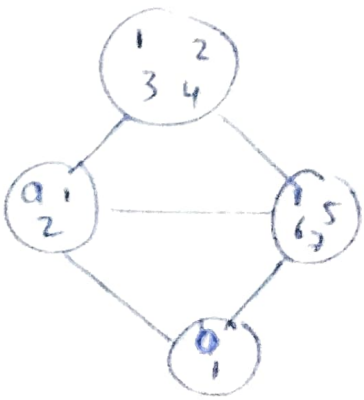
Physical Add

Logical Address



Physical Add. is unique for each network (N) and logical address is unique for complete world.

Ex →

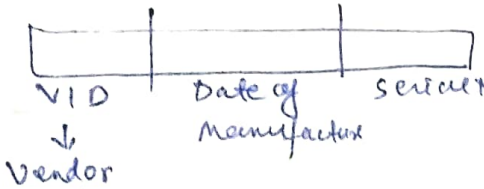


a and b are logical address so they can't repeat but physical address can repeat it would but not in same network

→ Physical and logical address are concepts but IP and mac address are exist in real.

→ IP number is publically unique. (32)

→ MAC is alloted to NIC. (48 bits)



→ Mac has 3 part not equally devided.

→ Mac is globally unique.

→ Both IP and mac can be used as logical address but we use IP as logical add. bcoz mac have extra info. that is not needed.

→ IP has useful info. for routing.

→ Mac is used as physical addrss bcoz it is permanent for the NIC in a particular network.

→ IP can also be use as physical add. but we don't use it.

→ Token ring and Ethernet both uses mac as physical address.

→ Not all N/w uses mac as physical address.

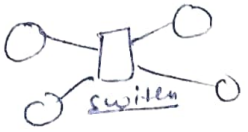
Network Layer → Logical Add.
DLL → Physical Add.

Parts of DLL →
- LLC → Logical Link Control (Flow control, manage)
- MAC → Medium Access Control (Menge framing) and phy. add.

Both manage error control

Network Layer (Layer-3)

1. Host to Host Connectivity (Can be in different N/w)
2. Logical Addressing
   → IP Add. in IP/TCP
3. Switching
   → Connect various N/w and make decision to send packet

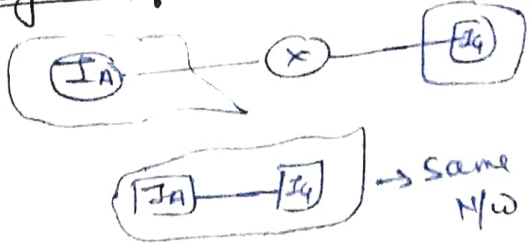


4. Router (Routing Table)
   → Use of routing table is switching.
   → Process of making routing table is routing.
5. Congestion Control.
6. Fragmentation

Transport Layer

1. End to End Connectivity
   → Connectivity of two processes in different host in diff. network
2. Flow Control (SR)
3. Error Control (Checksum)
4. Segmentation
5. Multiplexing and Demultiplexing
6. Congestion Control

How all the layers works together?

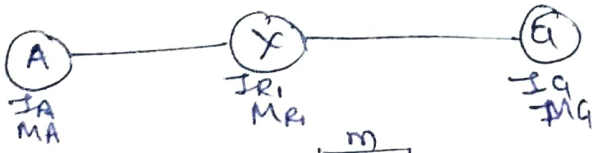


To check that both are in same network or not we use subnet mask concept

IA & SA → NIDAA

IG & SA → NIDGA

Assume that both are in different network.



AL → Message m

TL → Take message and add header (Port No.) | m | x | 80 | Ready |

NLD → Add IP Address | m | x | 80 | IA | IG |

DLL → | m | x | 80 | IA | IG | MA | MR1 | Add Physical Add.

PL → | m | x | 80 | IA | IG | MA | MR1 | Ready |

Address Resolution Protocol (ARP) use to convert (IP → Mac)

PL converts the packet into bits.

x → source port no. 80 → destination port no.

→ Connectionless technologies like (IP/TCP) doesn't send acknowledge

Session | Presentation layer

SL: ① Authentication & Authorisation

② checkpointing [save the state]

③ Synchronization

④ Dialogue control (One at a time)

⑤ Logical Grouping [Collect all Operations, put together and send it to server]

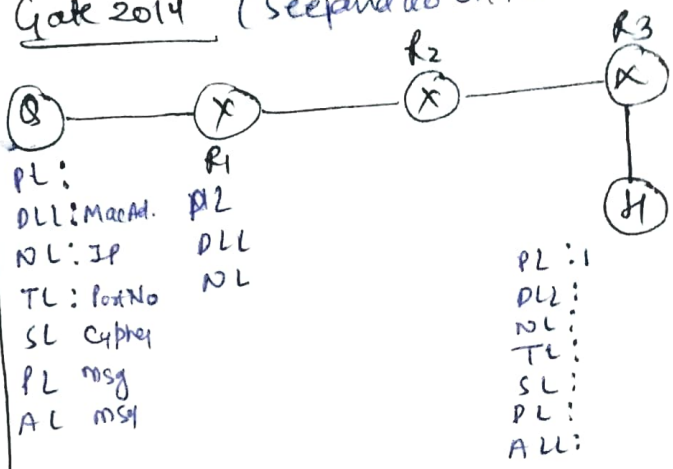
PL: ① Character translation → both sender and receiver must have same type of encoding

② Encryption and Decryption

③ Compression [.zip] → zip or unzip the data

Application layer implements both SL and PL.

Gate 2014 (See and do on Practice)



DLL → Ensure reliable transport of data over a physical point to point link.

NL → Routes data from one N/w node to next

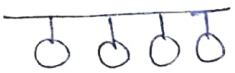
TL → All end to end comm^n b/w two process.

(Day-5) LAN Technologies

① Ethernet (IEEE 802.3)

→ Not use in practice now.

Topology → Bus



Access Control → CSMA/CD

No acknowledgement is implemented

Data Rate: Initially → 10 mbps

Now → 1 Gbps

100 mbps → FAST Ethernet

1 Gbps → 100 bit Ethernet

Encoding technique → Manchester

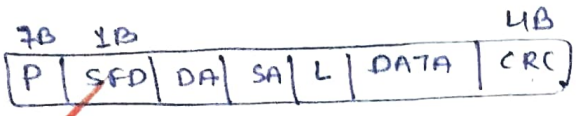
→ DLL deals mainly with LAN.

AL - Message
TL - segment
NL - Datagram
DLL - frame
PL - Single Protocol Data Unit (PDU)

Terms for Data as different layer

→ It is connectionless.

Ethernet frame format

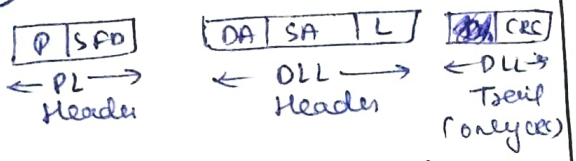


P → Preamble (7 bytes)
L → Length

Preamble is 7 bytes of (10)* (10.....10)

DA → Destination Add.

SFD: 1 byte : 10101011



Preamble and SFD use for -
① they indicate start of frame
② Use for synchronization.

DA & SA (6 bytes each)
→ They are mac address (mac add. as physical add.)
→ Changing mac address is illegal (but you can change it).

Types of mac add.

① UNICAST → One host to another.
② MULTICAST → One host to many.
③ BROADCAST → one to all.

→ In Unicast the least significant bit of 1st byte is '0' in mac add.

1A : 2B : 34 : 48 : CD : 20
we prefer hexadecimal no. (Just a convention)

0001101[0] : 10101011 :
lsb is '0'.

→ Source address should always be Unicast.

→ If lsb of 1st byte is '1' then it is Multicast

A3 : C5 : B4 : 67 : 89 : 10
10100001[1]
lsb of 1st byte

→ We are provided with Unicast add. and you have to create you own Multicast add.

→ In broadcast all the 6 bytes are 1s.

FF:FF:FF:FF:FF:FF

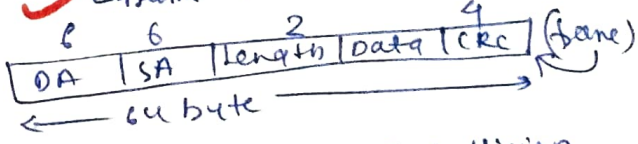
→ Broadcast is subclass of multicast.

L → Length (2 byte) (16 bits)
So Data can be of 2^{16}.

→ Data can't be of variable size.

from $L \geq 2 * T_b * B_w$ (CSMA/CD)
On putting standard value of T_b and B_w for Ethernet we get

$L_{frame} \geq 64$ Byte means



L_{min} = 64 byte to detect collision.
and max. restriction = 1500 B.

So

	min	max
Data	46 B	1500 B
frame	64 B	1518 B

Limitations

64 − (6+6+2+4) = 46 B Data

CRC (4 Byte)
You can use standard 32 bit polynomial.
$x^{32} + \text{---}$:

→ Ethernet is simplest way to connect channel

→ Not applicable for real time applications, because of collisions.

→ Min. frame size → 64 B. (More padding require).

→ Not useful for interactive applications.

→ Every station is treated equally (No priority). But server needs to get more priority over client so it is not suitable for client-server approach.

→ Jamming signals (generate during collision) count because for padding.

→ A station in Ethernet never stop sensing the channel because it has no acknowledgement

→ Station doesn't transmit the packet after collision

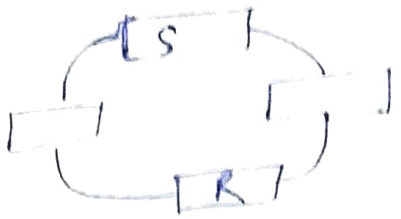
→ Back-off Algo. used for collision detection.

Token Ring (802.5) IEEE

→ Token Bus (802.4)

1) Topology used → Ring
2) Access Control Method → Token Passing
3) Unidirectional flow (simplex)
4) Data Rates 4mbps, 16mbps
5) Acknowledgement (Piggy backing)
6) Diff. Manchester Encoding

Operation → Token Passing



Problem → ① If sender gets removed before removing data packet from link then that packet get stuck in the link forever. Called as orphan packet.

② If packet get corrupt and sender reject it then packet become stray packet

Soln ① Insert a monitor in loop on seeing the packet first time it will allocate a bit in packet as mark it monitor see the same bit again it will remove the packet.

② If CRC is changed then monitor will remove the packet.

→ Above two problems are source problem.

Destination Problem :-

① Destination is down means destination is not picking up the packet

② Destination is busy (No space to take) Send packet again

③ ERROR in data (Correct & detect)
④ Copied data.

5/3/22

Token ring is not in syllabus, so dont need to go for more than the given information.

Switching

① Circuit Switching
② Packet switching
③ Message switching

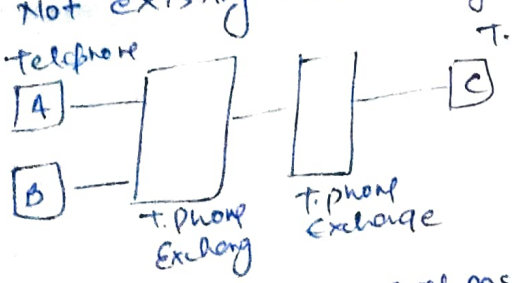
Types of Packet switching
① Virtual ckts.
② Datagram.

Telegram N/W → Telephone N/W → Computer N/W

If we feel any problem in CN we go back in Telephone N/W and search for how they solve it

Circuit Switching

Not existing now a days.



Assume, M → size of msg
B → Bandwidth
X → No. of T. Phone Ex.
d → distance b/w two T. Phone Ex
v → velocity

(Time)transmission = setup time
$(Xd/v)(T_p) + M/B(T_t)$
$+T$ (Teardown the link)

→ Applied at Physical layer
→ It is absolute (No one in using right now).

i) Headers are not required.
ii) No reordering can be done.

Packet Switching

Every msg have headers and save first in m/m of exchange (switch)

$TT = \left(\frac{XM}{B} + \frac{Xd}{v}\right)$

→ No setup and teardown time
→ It is store and forward N/W.

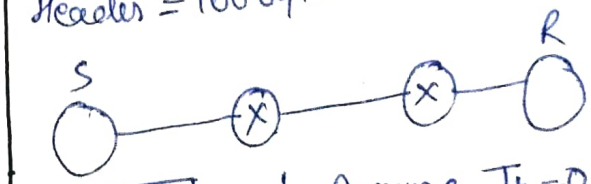
→ for a very large message use circuit time bcoz (T_t) will be very high for Packet switching and for small data go for packet switching.

Packetisation in packet switching

→ Pipeline is useful if you want to send burst of data.

Data = 1000 bytes B_w = 1 MBps = 10^6 Bps

Headers = 100 bytes



1000	100

← Packet → 1100 Bytes

Assume $T_p = 0$ for sake of simplicity

At every switch we store first and then transmit data

$T_t = \frac{L}{B} = \frac{1100}{10^6} = 1.1$ msec

$TT = 3 \times T_t = 3.3$ msec

If we want to send 5 packets

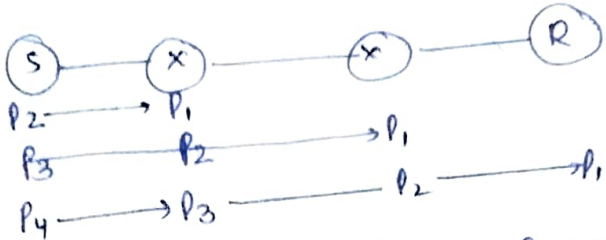
Data = $\frac{1000}{5}$ = 200 Bytes per packet

Bw = 1 MBps

Header = 100 Bytes

final packet size = 200 + 100 = 300 Bytes

$T_t = L/B = \frac{300}{10^6} = 0.3$ msec



(TT for P_1) = $3 \times T_t = 0.9$ msec

But for all other packets it takes only 1 $T_t = 0.3$ ms because all the packets are also under process along with P_1.

(TT) = $0.9 + 4 \times 0.3$

= 2.1 msec

So, pipelining decreases the total time (TT).

If we send 10 packets :-

Data = $\frac{1000}{10}$ = 100 Bytes

Same Bw and header

Packet size = 200 Bytes

$T_t = \frac{200}{10^6} = 0.2$ msec

(TT) = (0.2 * 9) + 0.6

(P_2 - P_{10}) → P_1

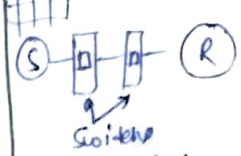
= 2.4 msec

→ By dividing data into small packets we get benefits.

→ Devide data into smaller packets but to get more efficiency or less (TT) do division in a way that (Packet size) should be significant larger than (Header).

Virtual Circuits and Datagram

VC



First packet reserved the buffer for upcoming packets on the way inside switch.

→ Every packet follow same path.

DG

All the packets goes independtly and can follow the different path

→ No any buffer is require always in switches.

→ Voice call is virtual circuit.
→ Internet data is datagram.

→ For a long distance call more switches are involved so they it is more costly.

→ Headers required at VC's because they response that every packet with same header will follow same path.

1) I-packet need Global header.

2) All other need local header (HOP to HOP) because Global header already creates a buffer for all in VC's.

→ VC's are connection oriented.

→ Datagram are not connection oriented.

→ In datagram all packets needs a separate header.

→ Datagram have no reserved buffer, so connectionless.

→ Path of every packet is same in VCs, so all will appear in order.

→ Path is not same is DG's so, they may appear out of order.

→ VC's are highly reliable

→ DG's are not reliable bcoz any packet can discard anytime

→ VCs are mostly costly and DG's are cost efficient.

→ Data calls uses datagram so they are less costly, so sometime your voice is not clear on whatsapp call or other bcoz some packet might get lost.

ATM → (Asynchronous Transfer mode uses VC's)

IP → (Internet Protocol uses Datagram).

→ ATM is not in syllabus.