1. INTRODUCTION:

- 7 layers.
- 5 layers.

Purpose/ Use. / Remedies / Protocol.

- Physical layer. (Unit 2)
- Data link layer. (Unit 3)
- Network layer. (Unit 5)
- Transport layer. } (Unit 6).
- Application layer. }

— INTRODUCTION:

- need of networking.
- n/w elements
- Applications
- N/w hardware
- N/w Topology.
- Model
- Design issues
- Service Primitives
- Services types.

N/W Elements:

1. Link.
2. Switch.
3. Bridge. : used to either connect two computers or two networks.
4. Router. : used to connect more than one networks., does processing for

identifying the address to which the message is to be sent. (Unicast / Multicast)

5. Hub: Sends the message to every network connected to it (broadcast).

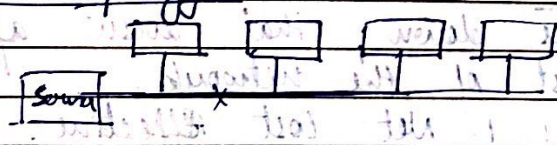
6) Repeater: Amplifies the signal.

* Applications:
1. Business Application
2. Home Applications
3. Mobile Applications
4. Social Applications

* Network Hardware:
↦ Based on interprocessor distance.
1. Personal Area Network (PAN) - 1 meter range
2. Local Area Network (LAN) — 1 Km range
3. Metropolitan Area Network (MAN) - 10 Km
4. Wide Area Network (WAN). 100–1000 Km.
5. Internet. > 10000 Km.

* Network Topology:
— The way in which computers are connected

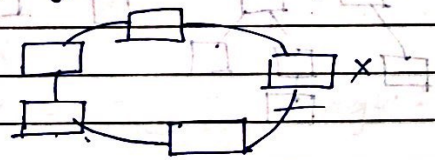
1. Bus Topology:



Pros: 1. Simple in architecture.
2. Cost Effective.

Cons: 1. If the main line goes down whole network goes down.

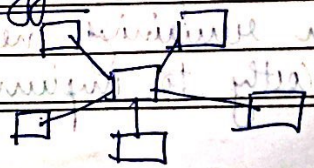
2. Ring Topology: (Token Passing Topology).



Pros: 1. Simple.

Cons: 1. If one connection breaks, the communication continues to an extent and then goes down.
2. If a system goes down, the passing of token will not occure and the whole network goes down.

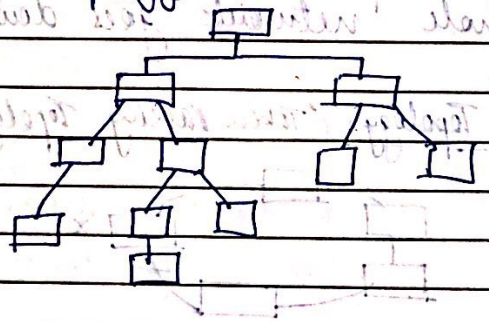
3. Star Topology:



Pros: 1. Even if a link or a system goes down, that doesn't affect the rest of the network.

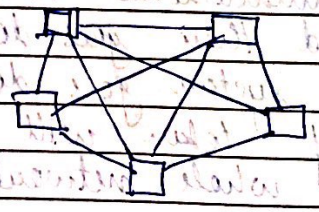
Cons: 1. Not cost effective.
2. If server goes down whole network goes down.

4. Tree Topology:



~~Cost~~ Pros: Cost effective.

5. Mesh Topology:



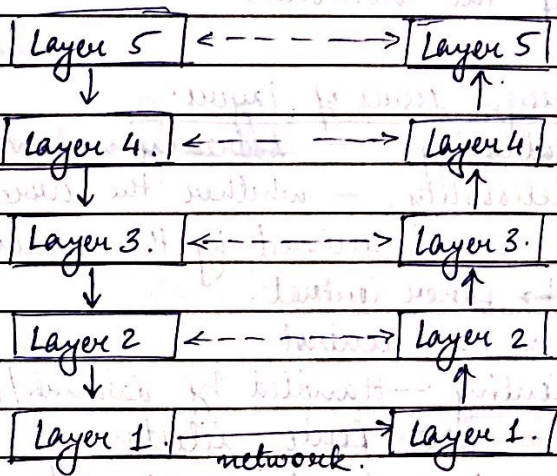
Pros: Even if a link or system goes down remaining network is functional.

Cons: Costly to implement.

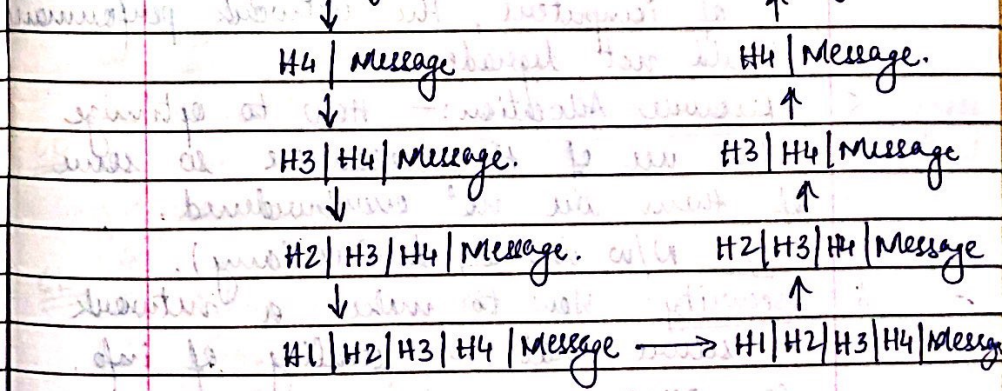
*** LAYERED TECHNOLOGY:

Sender. Receiver.

NODE-I. ←→ NODE-II.



Ex:



OSI – Open System Interconnection.

★ The number of tasks to be performed that many number of layers should be introduced. But large number of layers would slow down the speed of the network.

★ Design Issues of Layers:

(network layer)
1. Addressing. – Addressing mechanism.
2. Reliability. – whether the correct message is received by the receiver.
   ↳ Error Control.
   ↳ Flow Control.
3. Routing – Handled by datalink/transport layer
   – Best Route Selection.
   – Dynamic change of package route (intelligence).
4. Scalability – When increasing the number of computers, the network performance should not degrade.
5. Resource Allocation:– How to optimize the use of the resources so some of them are not overburdened.
   Ex: N/w device, devices (any).
6. Security: How to make a network secure that no stealing of info. can occur.

★★★ Connection Oriented & Connectionless Service:

1. Service: Set of primitives which are provided from one layer to another.

2. Protocol: Set of rules at which both communicating parties agree, at same layer.

– Connection Oriented Service: (Slower) Reliable.
   ↳ Connect.
   ↳ Communicate.
   ↳ Disconnect. Ex: Telephone call.

– Connectionless Oriented Software Service: (Faster) less Reliable.
   ↳ Communicate. Ex: Postal Service.

★ Services:

		Examples:
GO	1. Reliable Message Stream.	Sequence of Pages
GO	2. Reliable Byte Stream.	Movie Download.
GO	3. Unreliable Connection.	Voice over IP.
CL	4. Unreliable Datagram.	Junk Mail
CL	5. Acknowledge Datagram (Ack given source provider and not from Receiver)	Text Messaging.
CL	6. Request Reply.	Database Query.

Ex: chatting mechanism. Reliable.

P.T.O.

* Reliable communication:

— Message Stream:
Here when we send two messages and their boundaries are preserved: and we receive two messages respectively.

— Byte Stream:
Here we send two messages, which are combined as a stream and sent as a single message.

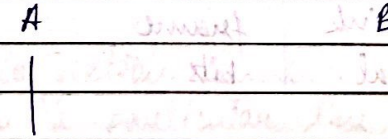
* Service Primitives:
To establish connection - ariuvte what are the steps that are to be followed.

1. Listen: Ready for connections to establish (server).
2. Connect: Establish a connection with the client.
3. Accept: Due to limited buffers only some connections will be accepted.
4. Receive: Accept request is received by the client and then both server and client sends the messages.
5. Send: Communication between the client and server
6. Disconnect: After communication is done the connection is disconnected.

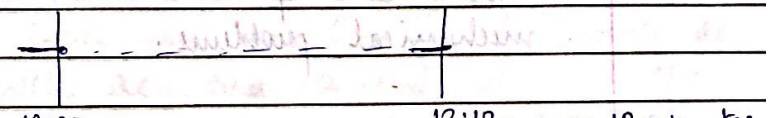
** OSI Model: (Reference Models):

* Terminologies:

1. Transmission Delay:
The total time taken by a frame to completely travel on a network.



2. Propogation Delay:
The amount of time required for the first bit to reach the receiver.



= 10 minutes.

P.T.O.

Every layer provides service to its upper layer.

Page No.
Date :

* Reference Models:

→ Intl. Standard Org. — Open System Interconnect (ISO — OSI).

Open System Interconnection — 7 Layers.

Application	Message.
Presentation	Presentation Protocol Data Unit
Session	Session Protocol Data Unit.
Transport	Transport — " — / Segment.
Network	Packet.
Data Link	frame
Physical.	bits.

1. Physical Layer:
   ↳ Primary task is to convert frame into bits.
   ↳ If it sends 1, it checks whether 1 is received or not and vice versa.
   ↳ Takes care of electrical and mechanical problems.

2. Data Link Layer:
   ↳ Converts the message in the form of Packet from Network Layer to frame in Data Link Layer.
   ↳ 2 functionalities.
   • Flow Control: If sender is fast and receiver is slow, then DLL asks sender to slow down data sending

Page No
Date :

   ↳ Doesn't think of message transfer on network.
   • Error Control: Controls the data transfer without any errors.
   ↳ DLL takes care of end-to-end data delivery.

3. Network Layer:
   ↳ Responsible to route a packet on a network, choosing the path for the message (packet).
   ↳ Congestion Control: If there is congestion on a particular line, the packet will be route to some another route to decrease congestion.
   ↳ Network Layer is thin in terms of broadcasting.

4. Transport:
   ↳ Looks for the Quality of service as well as end-to-end delivery of data.
   ↳ Provides Flow Control and Error Control with the presence of Network, but DLL provides both without the Network.

5. Session Layer:
   ↳ Responsible for establishing, maintain

ing and disconnecting the session.
↳ Also checks for synchronization of data, token management and dialog control.

6. Presentation Layer:
↳ Handles syntax and semantics of the message.
↳ Data compression, and other functions encryption / decryption.

7. Application Layer:
↳ Generates the application out of the message.
Ex: http protocol is used in application layer, DNS.

2. TCP / IP:

	OSI.
Application	→ Application, Pres, Sess.
Transport	→ Transport.
Internet	→ Network.
Link layer	→ Data L, Physical.

TCP/UDP
IP

- In OSI, the model is defined and then each of the layers are given its responsibilities / protocols.
- In TCP/IP, first the protocols are made and on those protocols' basis the model is constructed.

3. Internet Model:

Application.	
Transport	→ CLO service.
Network	→ CO. + CLO. service.
Data Link.	
Physical.	

UNIT 3 - DATA LINK LAYER.

- **Topics:**
  1. Services provided by DLL
  2. Framing.
  3. Error Detection and Correction.
  4. Flow Control.
  5. Protocols.

★ **Services:**

1. UDP { 1. Unacknowledged Conn.ⁿless service (Unreliably)
   { 2. Acknowledged ——"—— (Reliably).
   TCP. { 3. Acknowledged Connectionless service.

2. **Framing:** Need to recognize framing boundaries.
3. Error Control. – noise can ruin a Xmission.
4. Flow Control. – Control the rate of flow of data.

★ **Framing:**

How to break the bit stream into frames.

1. ↳ Time Gaps: Ways of dividing bit stream by time gaps.
   ↳ Character Count:
   ↳ Starting and ending characters, with character stuffing.
   ↳ Starting and ending flags, with bit stuffing.
   ↳ Physical layer encoding violation.
2. Character Count: can be specified in the header which denotes the number of character a frame consists of.

Without Error:

5	1	2	3	4	8	0	2	6	7	5	8	9	2	1

5 chars. 8 chars. 2 chars.

With Error:

5	1	2	3	4	8	0	2	6	7	5	8	9	2	1

↑ 7. ↑ 7. (1 frame suffered).
(All the frames are corrupted).

To use character count: Use a field in the header to specify the numbers of characters in the frame.
↳ Count can be garbled by a trans-mission error.
↳ Resynchronization problem.
↳ A character stream (a) without errors.
(b) with one error.

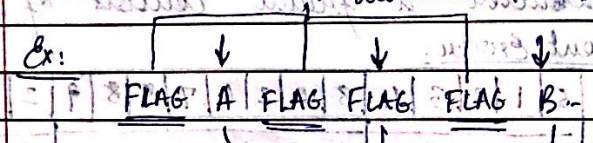
(a)

P.T.O.

More space is required in character stuffing, but less in bit stuffing.

3. Character Stuffing: → control information
Using flag bytes with byte stuffing.
(a). A frame delimited by flag bytes.
(b). Four examples of byte sequences before and after stuffing.

Ex: (to determine the boundaries of packets)



Stuffing occurs at sending side and Destuffing occurs at receiving side.

4. Bit Stuffing:
Starting and ending flags, with bit stuffing
↳ Begin and end with a flag bit "01111110".
↳ Prevent a flag from appearing in data by bit stuffing.
↳ Bit stuffing inserting 0 after 5 continuous bit "1" data appear.

This prevents the occurrence of the flag within the message to disrupt.

(a) 01111110 | 10101011111110100110. | 01111110
Flag — which disrupts the message.

* Error Control:
Ensure reliable frame delivery: every frame arrives without damaging, frames arrive in order.
↳ To provide the sender with some feedback.
* Positive Acknowledgement.
* Negative Acknowledgement.
↳ To provide timeout timers.
* Resend as necessary.
↳ To number of frame.
* To distinguish retransmissions from originals.

1. Error Types:
↳ Isolated Errors.
↳ Burst Errors.

2. Approaches:
↳ Error Detection and Correction.
↳ Error Correction.

* Error Detection: ED just detects whether there is an error in the message or not.
1. Parity.
2. Checksum.
3. Cyclic Redundancy Code (CRC).

★ Error correction: EC finds where the error is and corrects it.
↳ Hamming Code.

1. Parity: (a) (Vertical Redundancy Check):
↳ Even:

101101.0 ← as even ⟶ 1111010 ✗

Parity ↓ 1010010 ✓

01110101 1 ← Parity.

If there is odd bit error, then it detects an error.
Other words, it is a one bit error detection mechanism.

(b) (Longitudinal Redundancy Check) ← (Interleaving) — used to detect burst error

Sender:	Receiver:
100101	100010
011011	011011
101111	101111
011100	011100
001101	001101
	000000

All zeros when there is no error, else it will consist of non-zero elements.

↳ Works only when there is only one bit error in the whole column.
↳ Longitudinal Redundancy check checks for multiple bit error.

2. Checksum: Used at higher levels like Transport layer.
↳ If size of checksum field is n, then divide the stream in n bits.

Ex: 32 bit stream: (in 4 segments).

10011001 | 11100010 | 00100100 | 10000100.
A B C D.

Add the streams:

```
  A   10011001        01111100  E
+ B + 11100010      + 00100100  + C
    1 01111011        10100000  F
      ⟶ 1             10000100  + D
      01111100  E   1 00100100  G
                      ⟶ 1
                Sum   00100101  H
     Compliment: [11011010] (Checksum).
```

This process is called as 1's compliment arithmetic.

The sender sends the original message alongside the checksum.
So the size of data stream at receiver is 32 + 8 = 40 bits.

At Receiver:

```
   A          B          C          D
10011001 | 11100010 | 00100100 | 10000100
11011010.
   E
```

Sum = 00100101 A+B+C+D.

  + 11011010

Sum 11111111 A+B+C+D+E.

Complement: 00000000

If all the bits are zero, then there is no error.

↳ Checksum can detect multiple errors.

★ **Burst Error:**

```
1 0 0 1 0 0 1 1 1 1 0 0 1 0 1 0 1 1 1 0 1
          ↓   ↓       ↓ ↓
1 0 0 1 1 0 0 1 1 1 0 1 1 0 1 0 1 . 1 0
        |_______________|
           Burst error.
```

when initial and last error bits should be 1.

★★★ 3. **Cyclic Redundancy Check: (CRC)**

↳ CRC is used for error detection at Data Link Layer commonly.

↳ Uses Generator Polynomial. [G(x)]

↳ r is the degree of $G(x)$.

↳ $M(x)$ is the message.

↳ The size of the message at receiver is,

$M(x)+r$. (Append r at the end).

↳ The whole message is divided by $G(x)$.

↳ The substraction performed in division is modulo 2.

↳ we perform XOR rather than subtraction

↳ Remainder is added to the message.

Ex: $M(x) = 1101011111$

$G(x) = 10011$ OR x^4+x+1.

Transformed Message = 11010111110000

```
10011 | 11010111110000.
        10011
        010011
         10011
         0000011110
              10011
              011010
               10011
               010010
                10011
                000010.
```

Bits appended are 4 because in division by 5, at most 4 bits will be remainder.

CRC code is : (msg + Remainder)

11010111110010

At Receiver :

The message is divided by $G(x)$.

If the remainder = 0, then there is no error present.

```
10011 ) 11010111110010
        10011
        010011
         10011
        0000011110
             10011
             11011010
              10011
              1010011
               10011
               000000.
```

To get the exact message, receiver will determine the 'r' and will truncate off the bits from back.

* CRC can detect errors when:

- ↳ If error is factor of $G(x)$. (Cannot detect)
- ↳ Single bit error.

* ↳ It can detect burst error of length $\leq r$.

- Actual polynomial used in a system has a degree $r = 32$.

* ↳ If in the error, odd number of bits are present, it can detect error if r = odd. (Odd number error).

** Error Correction:

1. Hamming Code :

- ↳ can detect multiple bit error.
- ↳ can correct single bit error.

$n = m + r$ (n → Hamming code, m → Actual message, r → redundant bits).

Here, the redundant bits are added in between the message.

If m-bit message, $2^r - 1 \geq n$.

Ex: Let us consider,

Actual msg: 1001101

Parity bits → P_1 P_2 D_3 P_4 D_5 D_6 D_7 P_8 D_9 D_{10} ....

(Data bits: D_3, D_5, D_6, D_7, D_9, D_{10})

↳ Parity bits are added in the message in the location 2^n, where $0 \le n \le \infty$.

Rough Hamming code:

$P_1\ P_2\ 1\ P_4\ 0\ 0\ 1\ P_8\ 1\ 0\ 1$.

Now to calculate Parity Bits,

$P_1 = P_1, 1, 0, 1, 1, 1. \quad = 0$

$P_2 = P_2, 1, 0, 1, 0, 1. \quad = 1$

$P_4 = P_4, 0, 0, 1, \quad = 1.$

$P_8 = P_8, 1, 0, 1 \quad = 0.$

} consider 2^n number of bits and leave 2^n bits.

∴ Hamming code is

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

At Receiver:

m = 0 1 1 1 1 0 1 0 1 0 1

$P_1 = 0, 1, 1, 1, 1, 1. \quad = 1.$

$P_2 = 1, 1, 0, 1, 0, 1 \quad = 0.$

$P_4 = 1, 1, 0, 1 \quad = 1.$

$P_8 = 0, 1, 0, 1 \quad = 0.$

} involved in error at, it is even parity, so it should be 0.

$P_1\ P_2\ P_4$

∴ The error is at location 5.

Actual Message = 0 1 1 1 0 0 1 0 1 0 1 1

* If error rate is high, then better way is to correct the error on receiver side, but if the error rate is low it is better to detect and dont send acknowledge to get the message from sender again.

Ex: Consider a 4 bit message 1001 is to be send by using Hamming code method therefore find out the Hamming code at the sender. Calculated Hamming code is sent on the network. Consider, the message received at the receiver contains an error on last bit when compared to calculated Hamming code. Find out the position of the errored bit.

Solu: m = 1001.

$P_1\ P_2\ 1\ P_4\ 0\ 0\ 1$.

Rough Hamming code: $P_1\ P_2\ 1\ P_4\ 0\ 0\ 1$.

Now, Calculate Parity bits:

$P_1 = P_1, 0, 1, 1. \quad = 0.$

$P_2 = P_2, 1, 0, 1 \quad = 0.$

$P_4 = P_4, 0, 0, 1 \quad = 1.$

∴ Hamming code is,
0011001.

At Receiver:
m = 0011000
$P_1 = 0,1,0,0 = 1.$
$P_2 = 0,1,0,0 = 1.$
$P_4 = 1,0,0,0 = 1.$
$P_1 + P_2 + P_4 = 7.$
∴ The error is at location 7.
Actual Message = 00111001

Eg: A bit stream 1001110 is transmitted using the standard CRC method. The generator polynomial is x^3+1. Show the actual bit stream transmitted. Suppose that the 3rd bit from the left is inverted during transmission. Show that this error is detected at the receiver's end. Give an example of bit errors in the bit stream transmitted that will not be detected by the receiver.

Soln: M(x) = 1001101.
G(x) = x^3+1 OR 1001.

Transformed message = 1001101000.

```
        1101
1001 | 1001101000
       1001
       -----
        0001101
          1001
          ----
          0000100
             1001
             ----
             0000100
```

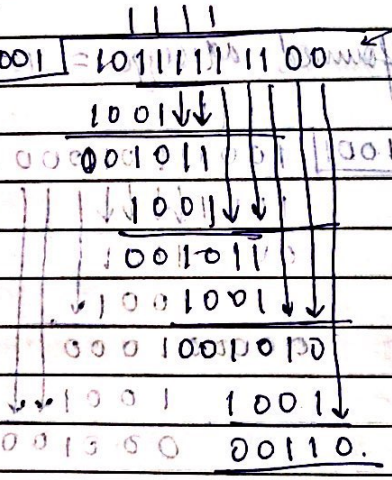
CRC code: 1001101100

At Receiver:
```
        1101
1001 | 1001101100
       1001
       -----
        0010
           1001
           ----
           0011
              1001
              ----
              0010
```
∴ Error is detected.

IEEE 802 = 32 degree polynomial.

Page No.
Date :

Burst error of length greater than r will not be detected.



If burst error is less than length of r, then error will always be detected. But if the length of burst error is greater than r, then it is not guaranteed whether the error will be detected or not.

Ex: A 1024 bit message is sent that contains 992 data bits and 32 CRC bits. CRC is computed using IEEE 802 standard, 32 degree CRC polynomial. For each of the following, explain whether the errors during message transmission will be detected by

CRC can detect — single bit, all odd bits, 2 isolated bit, burst < r.

Page No
Date :

the receiver:

1. There was a single bit error.
2. There were 2 isolated errors.
3. There were 18 isolated errors.
4. There were 47 isolated bit errors.
5. There was a 24 bit long burst error.
6. There was a 35 bit long burst error.

1 — Yes, because CRC detects all single bit errors

2 — Yes.

18 even 3. — Maybe/No, because CRC detects only odd bits errors perfectly but cannot guarantee for even multiple bits.

47 odd 4. — Yes, because odd number of isolated bits are always detected.

5. — Yes, because the length of the burst < r

6. — No, because the length of the burst > r.

P.T.O.

★ Flow Control:

Protocols

For Noiseless channel	For Noisy channel

★ Assumptions:

★ Wait_for_Event (&event):

At sender it waits for NL to send the data, as well as it sends it to PL and starts the events to get the ack for message.

→ Frame contents:

1. **Kind:** It determines whether the message consists only control info or data or both.
2. **Seq_no:** To identify the ordering of frame at receiving side.
3. **ack:** used for acknowledgements.
4. **info:** It consists of a single packet of data. For control info, this field is null.

O.T.P

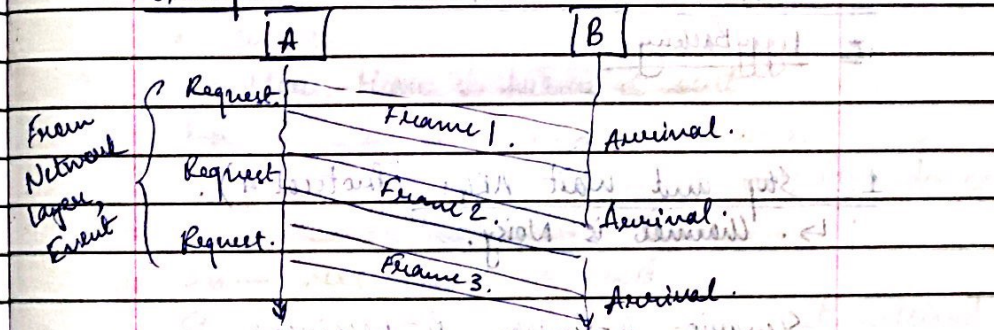
In real, the noiseless channels are non existent.

★ UNRESTRICTED SIMPLEX PROTOCOL:

↳ Processing time is negligible.

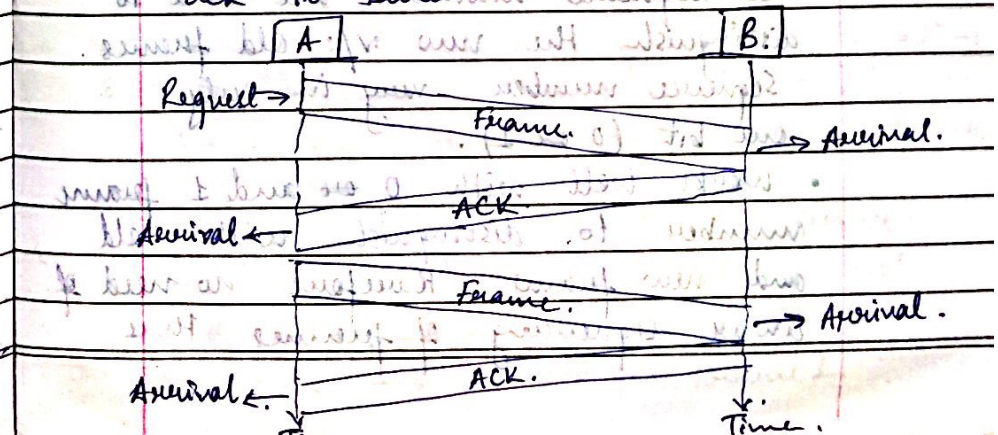
- **Assumptions:**
  - ↳ Infinite Buffer Space.
  - ↳ Data flow is unidirectional.
  - ↳ Communication channel is noiseless.
  - ↳ Sender and receiver are always ready.

Example:



★ Simplex Stop and wait Protocol:

1. One way data traffic.
2. Errors free communication channel.
3. Finite buffer space in DLL at receiver.
4. Transmission is duplex as receiver sends ack to sender.



Sequence number are assigned to each of the frames in the form $0 - 2^n - 1$

Ex: $n=3$, $0 - 2^3 - 1 = 0 - 7$.

* Its an error free channel so no chance of the ACK getting corrupted or lost.

* NOISY CHANNELS:

↳ SLIDING WINDOW PROTOCOLS:

* PiggyBacking:

1. Stop and Wait ARQ: (Protocol 4).

↳ Channel is Noisy.

Scenario: A-Sender, B-Receiver:

- A sends data.
- B receives data, sends Ack to A.
- Ack get lost.
- After timeout on A, it resends the frame which causes duplication of frame at receiver.
- So sequence numbers are used to distinguish the new v/s old frames. Sequence number vary in only one bit (0 or 1).
- Works well with 0 and 1 frame number to distinguish with old and new frame, therefore no need of linear sequencing of frames like increasing the frame size.
- Stop and wait wastes the bandwidth so solution is to send multiple frames, which violates the rule of Stop & wait ARQ.

2. Go Back N: (Protocol 5). $(2^n - 1)$

- Sender:

↳ More than 1 frame is sent.

↳ $n = 2^m - 1$, suppose $m=3$,

$\therefore n = 0 - 7$. (sequence no.) (Window size)

S_n — next frame to send.

S_f — frame for which Ack isn't received.

R_n — next frame is expected.

Here, if one frame is lost all the frames succeding that frame will have to be retransmitted.

$\therefore$ Wasting bandwidth in high error rate.

3. Selective Repeat: (Protocol 6): $\left(2^n/2\right)$. $\frac{-1}{2} = \frac{2^n - 1}{2}$.

↳ Window size for sender and receiver is same.

↳ Greater buffer required at receiver.

↳ R_n — Here points to window not a specific frame.

↳ NAK 1 ⟶ No Ack for frame 1.

★ HDLC Protocol:

↳ Stands for High level Data Link Control. (Bit control protocol)

↳ Most widely used

• Stations (nodes) in network:

1. Primary:
   ↳ controls operation of link.
   ↳ issues commands (frames).
   ↳ Maintains seperate logical link to each secondary station.

2. Secondary:
   ↳ Under control of primary station.
   ↳ Issues Responses (frames).

3. Combined:
   ↳ May issue commands and responses.

• Link Configuration:

1. Unbalanced: Primary & Secondary stations
2. Balanced: Both combined stations.

• Transfer Mode:

1. Normal Response Mode (NRM).
   ↳ Uses Unbalanced link configuration.
   ↳ Primary initiates.
   ↳ Secondary transmit in response.

2. Asynchronous Response Mode (ARM):
   ↳ Unbalanced
   ↳ Secondary initiate without permission from primary.
   ↳ Primary — connect, disconnect, error correction, etc.
   ↳ Rarely Used.

3. Asynchronous Balanced mode (ABM):
   ↳ Balanced.
   ↳ Either station initiate transmission without permission.

★ • FRAME STRUCTURE:

01111110 ↓ Flag	Address	Control	Information	FCS	01111110 ↓ Flag
8	8	8 or 16	variable	16 or 32	8

Address: = 8 bits, extendable to 16 bit.

Control: I — Information
S — Supervisory.
U — Unnumbered.

FCS: Frame Check Sequence
consists of Error Control
Ex: checksum, CRC, etc

7	6	5	4	3	2	1	0
1	1	0	0	0	0	P/F	

P.T.O.

-11- (flow control and error)

In S — user info is absent.
In I — user info is provided → data task.
In U — user info replaced by management information.

★ Address Field:
- Identifies & puts receiver address.
- Usually 8 bits long.
- All (11111111) is broadcasting.

★ Control Field:
First or first two bits are used to determine type of frame.

8 bits:

I — | 0 | N(S) | P/F | N(R) |
1 2 3 4 5 6 7 8 → Where to be taken

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

Only 8 bit U : | 1 | 1 | M | P/F | M |
1 2 3 4 5 6 7 8

N(S) — Send seq. no.
N(R) — Receive seq no.
S — supervisory function bits.
M — Unnumbered function bits.
P/F — Poll / Final.

16 bits:

I : | 0 | N(S) | P/F | N(R) |
1 2 ... 8 9 ... 16

S : | 1 0 | S | 0 0 0 0 | P/F | N(R) |
1 2 3 4 5 6 7 8 9 10 16

U : 8 bit always

★ PPP is always used for broadcasting, no multicasting and unicasting.

- Poll / Final:
  - ↳ Use depends on context.
  - ↳ Command Frame:
    - ⇒ P bit : send for poll from primary.
  - ↳ Response Frame:
    - ⇒ F bit : from secondary.

Sender: Command — P=1 if response is commanded; P=0 if response is not commanded.

Secondary Response — F=1 if response is piggybacked; F=0 no response is sent.

★ S Frame:

S:
(00) 1. RR — Receiver Ready
(01). 2. RNR — Receiver Not Ready.
(10). 3. REJ — Reject on frame N(R) .. Go back N.
(11). 4. SREJ — Selective reject on N(R) ... Selective Repeat

★ U-Frame:

P.T.O.

Point-to-Point Protocol (PPP)

- character oriented protocol.
- Integral no. of bytes can be send, cannot send 31.25 bytes
  (∵ HDLC, whereas, it is bit oriented).

***Q** Station A needs to send a message consisting of 9 packets to station B, using a sliding window protocol flow error control strategy. (window size = 3). All packets are ready and immediately available for transmission. If every 5th packet that A transmits get lost (but no ack from B ever get lost) then what is the number of packets that A will transmit for sending the message to B.

Soln: Consider window size = 3 at sender.

frame = 1 2 3 4 5 6 7 8 9.

Sender	Receiver
1	1 ✓
2	2 ✓
3	3 ✓
4.	4 ✓
(5)	lost
6	6 × } discarded.
7.	7 × }
5.	5 ✓
6	6 ✓
(7).	lost 7.
8	8 × } discarded.
9	9 × }
7	7 ✓
8	8 ✓
(9)	lost.
9	9 ✓

(Margin notes: every 5th packet lost; brackets marked "window")

Total = 16

∴ Total No. of packets sent = 16.

P.T.O.

***Q** Station A uses 32 byte packets to transmit messages to station B using sliding window protocol. The round trip delay between A and B is 80 msec. and the bottleneck bandwidth on the path between A and B is 128 Kbps. What is the optimal window size that A should use?

(Margin note: This is the only propagation time of round trip.)

Soln: Formula:

$$\text{Utilization} = \frac{n}{1+2a}$$

n = window size.

a = propagation time / transmission time.

We consider maximum utilization.

∴ Max utilization = 100%.

$$\frac{n}{1+2a} = 1$$

$n = 1 + 2a$.

Round trip propagation time = 80 msec.

∴ one way, propagation delay = $80/2 = 40$ msec

Transmission time = frame size / bandwidth.

transmission time = $\frac{32 \times 8}{128 \times 1000}$ bits.

$= 2$ msec.

∴ $n = 1 + 2a = 1 + 2 \times \frac{40}{2} = 41$ (propagation time / transmission time)

∴ **window size = 41.**

*Q Frame of 1000 bits are sent over 10^6 bps. (bits per sec) between two hosts. The propogation time is 25 msec. Frames are to be transmitted into this line to maximally pack them in transmit (within the link). Suppose that sliding window protocol is used with this sender window size of 32 and acknowledgements are always piggybacked. After sending 32 frames what is the maximum minimum time, the sender will have to wait before starting transmission of next frame.

Soln: propogation time = 25 msec. (no processing time given), round trip = 50 m
transmission time = $1000/10^6$ sec = 1 msec.

At sender:

1 frame is transmitted in 1 msec.

∴ 32 frames transmitted in 32 msec.

This sender cannot send the 33rd frame until it gets the acknowledgement for the 1st frame.

Round trip for a frame = 50 msec.

sender sits idle for = 50 − 32 = 18 msec.

∴ Minimum amount of time sender waits = **18 msec.**

UNIT – 4.

SWITCHING and MAC LAYER.

★ Types of switching:

1. Datagram.
2. Packet Switching / Virtual Circuit.

★ CIRCUIT SWITCHING: (Reliable but slow):

The switches must know how to find the route to the destination and how to allocate bandwidth (channels) for establish a connection.

- Types:
  1. Frequency Division Multiplexing
  2. Time Division Multiplexing.

★ PACKET SWITCHING:

- Basic Operations:
  - ↳ Data are transmitted in short packets.
  - ↳ Store and forward. (used for processing and routing purpose).

1. Datagram:
2. Virtual Circuit: (Destination address not required)
   Every node has virtual circuit Identifiers used to identify all the links connected to it.

fastest – Datagram.
Reliable – Circuit switching.

★ Diff between Circuit Switching and Virtual CS is that in CS the physical path is fixed and will never change, but in VCS (the route can be changed ↳ depending upon the traffic or link broken, etc.

★ Medium Access Sub Layer: (used for broadcasting and shared channel acce

- Network
- ← Logical Link Control sublayer/unit 3)
- ← Medium Access Sublayer.
- Physical.

- IEEE standard of 802:

- Channel Allocation:

Problem: DLL is unable to allocate a single shared channel to multiple users.

Solution:

1. Static Channel Allocation.
2. Dynamic Channel Allocation.
   - a) Continuous Time.
   - b) Slotted Time.
     - a) Carrier Sense.
     - b) No Carrier Sense.

*** Multiple Access Protocols:

1. Random Access Protocols (no controller present) of. Also any station can begin at any time (Continuous Time).

a). Pure ALOHA:

↳ under continuous Time.

↳ There will be collision and the colliding frames will be damaged.

↳ If frame is destroyed the sender just wait a random amount of time and sends it again.

↳ If no frame is transmitted at the time our station is transmitting the frame, 100% efficiency is achieved.

↳ To determine whether the signal is original or collided, identify the intensity level of the signal. (intensity increases when signals collide).

Energy level

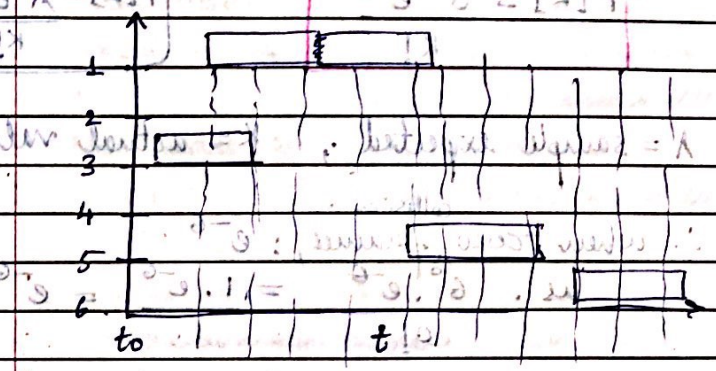
↳ Uniform frame sizes are used.

* ↳ Throughput:

↳ Assumption: frames are of equal length

↳ Vulnerable Time: Time wasted due to collision.

Let 't' be the time a frame takes to receive at the receiver.



The worst case, when two frames are collided $\leq 2t'$ vulnerable period is wasted as 2 frames will have to be completely retransmitted. (for frame 5 & 1).

N = No. of frames to be transmitted.

G = total number of frames that have to be transmitted due to collision.

$G = N$ if no collision.

$G > N$ if collision present

$\therefore G \geq N$.

P_0 = prob. of frames not collide.

S = Throughput i.e., successfully receive

$S = G \cdot P_0$

Consider, k frames are generated during an interval, expected G.

Scanned with CamScanner

Using Poisson's formula,

$$P[K] = \frac{G^K e^{-G}}{K!}$$

Poisson's dist.

$$P[K] = \frac{\lambda^K e^{-\lambda}}{K!}$$

λ = sample expected ; K = actual value.

∴ when zero collision frames, : e^{-G}.

i.e. $\frac{G^0 \cdot e^{-G}}{0!} = \frac{1 \cdot e^{-G}}{1} = e^{-G}$.

∴ throughput here is 100%.

in time interval t ⇒ G expected.

in time interval 2t ⇒ 2G expected. (vulnerable time)

Probability of no frames are generated in vulnerable time 2t!

∴ $P[K] = e^{-2G}$ | K = 0.

∴ $S = G \cdot P_0 = G \cdot e^{-2G}$

$S = G \cdot e^{-2G}$

Max. throughput occurs at G = 0.5 with S = 1/2e

= 0.184.

∴ Max 184 out of 1000 frames will be received by the receiver.

In the equation, put G = 0.1, 0.2 .... and plot the graph. The maximum G will be at 0.5

b) SLOTTED ALOHA:

Assumptions:

↳ All frames same size.

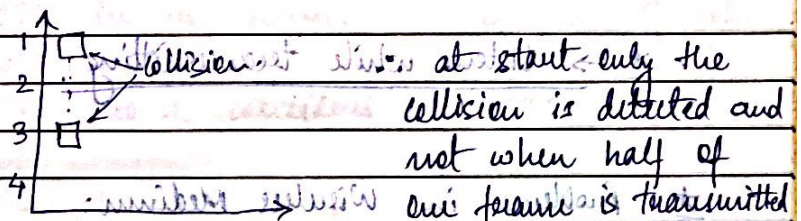
↳ Time is divided into equals slots.

↳ nodes are synchronized.

↳ nodes start to transmit frames only at start of slots.

Vulnerable time is t

∴ $S = G \cdot e^{-G}$



collision at start only the collision is detected and not when half of one frame is transmitted

Max Throughput = 370 out of 1000 frames.

also part of Random Access Protocols.

2. a) Carrier Sense Multiple Access Protocols:

↳ Channel : A

1). 1 – Persistent CSMA → sense continuously

2). Non – Persistent CSMA → if busy, wait random time and sense again

3). p – Persistent CSMA same as 1-persistent with p probability

In p-persistent, if the channel is idle then a random number is generated between 0 and 1 and waits that random time before sending to minimize collision.

no. of station to send. $n p < 1$. p: probability, 0.8

★ CSMA – CD (Collision Detection):

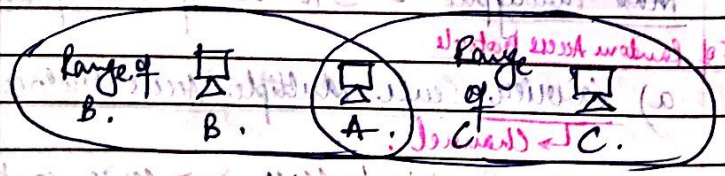
↳ Used in wired systems.

Ex: Ethernet.

★ ↳ Listens while transmitting.

★ Problems in Wireless Medium:

1. Hidden Station problem:

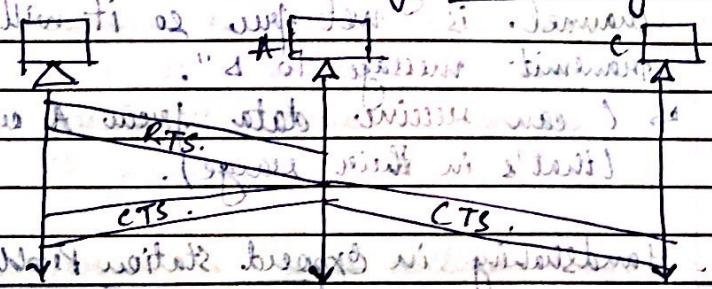


↳ B and C are hidden from each other with respect to A.

↳ ∴ A will receive signal both from B and C. So it will receive garbled signal. But B and C will not detect collision.

Clear to send

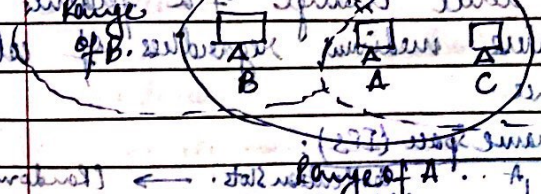
↳ The CTS frame in CSMA-CA helps avoid collision. using handshaking.



If A gets RTS from B, it will send CTS in its range to B and C both will receive CTS. But C didn't send RTS, so it concludes that CTS is not for C.

If A gets RTS from B and C at the same time then RTS is collided and A cannot read and ignores.

2. Exposed Station problem:

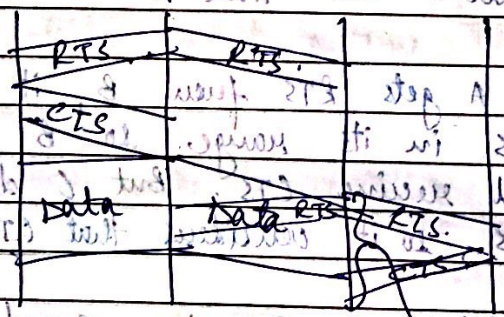


↳ C is exposed to transmission from A to B.

↳ A will send signal to B but it will also be detected by C. So C will assume that channel is not free so it will not transmit message to "D".

↳ C can receive data from A or D. (that's in their range).

- Handshaking in Exposed station Problem:

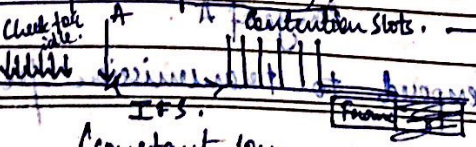


A to B transmission exceeded.

CSMA – CA (Collision Avoidance):

↳ For wireless medium, collision detection is not possible because there is always energy level change of a frame in a wireless medium regardless of collision occurrence.

↳ Interframe Space (IFS):



(constant for any particular station)

2. Controlled Access Protocols:

- Provides order access to shared medium so that every station has chance to transfer (fair protocol).
- Eliminates collision completely.
- Throughput = 1.

1. Reservation Access Method:

For reservation consider frame initialized: [0|0|0|0|0] (1, 2, 3, 4, 5)

Consider stations 1, 4, 3 wants to send data, while they finds reservation frame is available, they put their entries as (1, 2, 3, 4, 5)

[1|0|1|1|0] → Controller.

So whoever reserved the frames, are given a fixed time interval to send their data. In this order of entry reservation doesn't matter but controller mechanism use that too.

2. Polling:

↳ Stations take turns accessing the medium.

↳ Centralized and distributed.

↓ no primary, secondary.

"primary" or secondary.

1. Centralized Polling:
↳ Primary sends a poll frame to each node and if the node has data to send, it sends the data or NAK is sent.

2. Distributed Polling:
↳ Each station is assigned with a priority and they send data and pulling notice to next frame, send data, and so on...

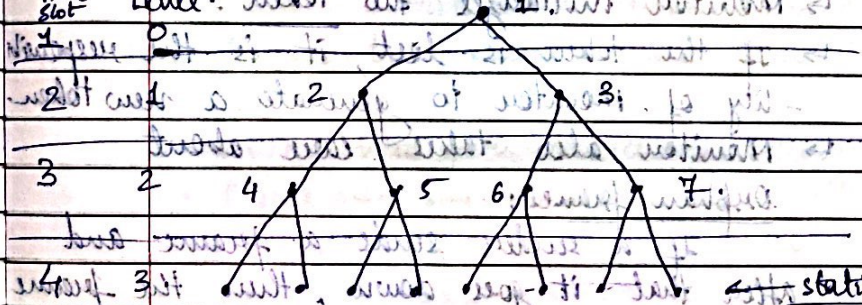
★ 3. Control Token / Token Passing:
↳ The control token technique uses a control or permission token to share the communication resources between a number of nodes. The technique can be applied to both bus and ring network topologies.

★★ COLLISION FREE PROTOCOLS:

1. Bit-Map Protocol:
↳ It is very similar to Reservation Protocol

2. Binary Countdown Protocol:
Rules:
↳ Compete by "OR" with other bits in the same order.
↳ If there is a "0 to 1" transformation" in the result - give up.

3. Adaptive Tree Walk Protocol:
↳ Stations are present at the leaf node only.



★ Data is sent in Depth first order.

★ LAN:

1. Token Ring Network (IEEE 802.5):
↳ Token Passing Method of Controlled Access Method is used.
↳ Max 250 nodes.

Monitor - who actually monitors the functioning by itself being a normal node of the network and takes

necessary actions.

Ex: C.R.

↳ Send claim token frame to request to become a Monitor.

If claim token frame of itself is received before any other claim token frame become monitor.

↳ Monitor initializes the token

↳ If the token is lost, it is the responsibility of Monitor to generate a new token

↳ Monitor also takes care about Orphan frames:

If a sender sends a frame and after that it goes down, then the frame will keep on moving because A sender is not available to delete the frame, so Monitor does that work by setting Monitor bit as 1 as a frame passes by the Monitor.

- Access control (1 Byte):
  1. Token bit
  2. Monitor bit
  3. Priority bit.

(3 bits) (000 - low priority, ... 111 - high priority).

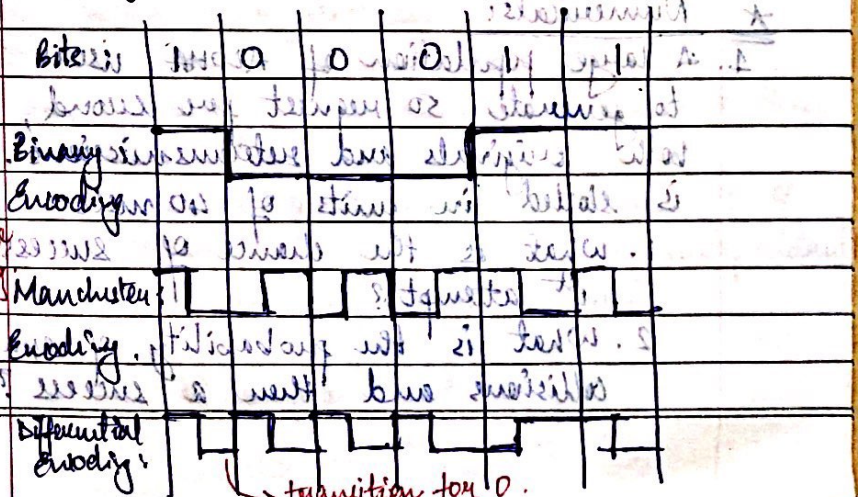
2. Fiber Distributed Data Interface (FDDI):
   ↳ Has a dual ring
   ↳ Can support 1000 nodes.

★ 3. Ethernet (802.3):

↳ Ethernet cabling:

- 10 Base 2 — means it operates at 10 Mbps, uses baseband signaling and supports segments upto 200 metres
- Device need to connecting Ethernet cable:
  1. Repeater — physical layer device — receives, amplifies, retransmits the signal.
  2. Hub
  3. Switch

- All ethernet system uses Manchester encoding (high at 0.85V & low at −0.85V).



starts at 1=high 0=low at middle. does opposite of starting

Ex: Differential Manchester, transition take place for 0 & not for 1.

- Ethernet MAC Sublayer Protocol:
- Ethernet Transmit Algorithm:
  - ↳ called 1-persistent.
- Binary Backoff Exponential Algorithm:
- Switched Ethernet:
  - ↳ Deals with high load
- Fast Ethernet: (802.3u) → 100 Mbps:
  - ↳ backward compatible with standard Ethernet
- Gigabit Ethernet (802.3z) → 1000 Mbps:

★ Numericals:

1. A large population of ALOHA users manages to generate 50 request per second, including both originals and retransmissions. Time is slotted in units of 40 msec.
   1. What is the chance of success on 1st attempt?
   2. What is the probability of exactly k collisions and then a success?

Solution: Chance of success on first attempt:

$$P[k] = \frac{G^k \cdot e^{-G}}{k!} \quad \text{(Slotted ALOHA formula)}$$

for $k = 0$,

$$P[0] = \frac{G^0 e^{-G}}{0!}$$

$$\therefore P[0] = e^{-G}$$

Frame transmission time = 40 msec.

(As in slotted Aloha, slots depend upon the minimum transmission time for a frame)

50 packets/sec are generated.

$$G = 50 \times 40 \times 10^{-3} = 2 \text{ requests/sec.}$$

$$P[0] = e^{-G}.$$

[Probability of exactly k collisions and then success: In slotted Aloha, the probability of success or collision, whatever, is independent of the previous transmission.]

(No. of attempts = k+1.)

P[K collision & next success] = P[K collision] × P[(K+1)th success].

↳ only 1 succ.

P[1 collision] = $1 - e^{-G}$

↳ success probability

P[1 collision] = 1 − P(success)

∴ P[1 collision] = $1 - e^{-G}$ → probability.

For K collisions it P[K collisions] = $(1-e^{-G}) \times (1-e^{-G}) \times \dots$ K times

∴ P[K collision] = $(1-e^{-G})^K$

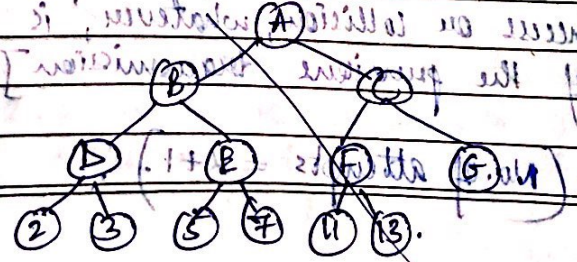
∴ P[K collisions & next success] = $(1-e^{-G})^K e^{-G}$.

$= (1-e^{-2})^K e^{-2}$.

(G=2 calculated in part 1).

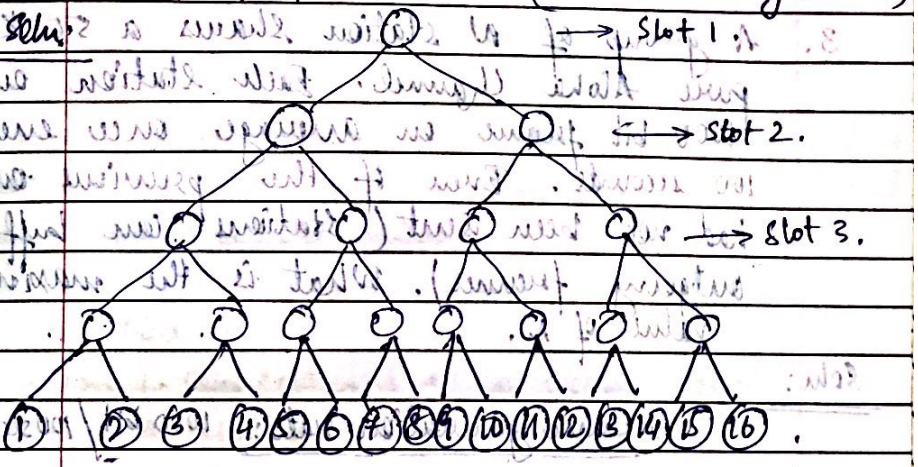
P[K collision & next success] = 0.137×0.865^K.

2. 16 stations, numbered 1 through 16 are contending for the use of shared channel by using Adaptive Tree Walk Protocol. If all the station whose addresses are prime numbers, have suddenly become ready at once, how many bit slots ↳ or attempts are needed to resolve contention.



{2, 3, 5, 7, 11, 13} → (who are ready to send)

Soln:



Slot 1: 2, 3, 5, 7, 11, 13, collision move to left subtree.

Slot 2: 2, 3, 5, 7.

try sending, again collision.

Slot 3: 2, 3.

try sending, again collision.

Slot 4: 2.

successful transmission.

Slot 5: 3 sends.

Slot 6: 5, 7 try sending, again collision.

Slot 7: 5 sends.

Slot 8: 7 sends.

Slot 9: 11, 13.

try sending, again collision.

Slot 10: 11 sends.

Slot 11: 13 sends.

3. A group of N station shares a 56 Kbps pure Aloha channel. Each station outputs 1000 bit frame on average once every 100 seconds. Even if the previous one had not been sent (stations can buffer outgoing frames). What is the maximum value of N.

Soln:

Frame generation rate = 1000 bit / 100 sec = 10 bps

Channel = 56 Kbps = bandwidth.

{ But in pure Aloha, maximum bandwidth that can be achieved is, 0.184 }

Even though bandwidth = 56 Kbps.

usable bandwidth / but successful bandwidth = $56 \times 10^3 \times 0.184$

= 10304 bps.

N = no. of stations

$n = \frac{\text{usable bandwidth}}{\text{frame generation rate}} = \frac{10304}{10}$

= 1030.4 ≅ 1030

Unit – 5.

Network layer.

⇒ Network layer is concerned of routing the packets such that they are delivered to the destination.

Offers:

1. Connectionless & Connection Oriented service
2. Routing → who packet will travel on network.
3. Congestion Control.

★ Routing:

↳ Packets are often routed from the source to the destination hop by hop.

Types:

Non-Adaptive
1. Static. — predetermined routes.

Adaptive
2. Dynamic. — changes decision on current scenario.
3. Default.

• Optimal Principle:

↳ Sink Tree: The shortest path from a source to all other destinations.

1. Static Routing Algorithm.
2. Dynamic Routing Algorithm.
3. Hierarchical Routing.

Network structure is known to each router.

II. Dynamic Routing Algorithm:

1. Distance Vector Routing:

↳ Each router maintains a routing table giving the best known distance to each destination and which line to use to get there.

↳ Every router will maintain routing information all other routers but this routing information is shared only with the neighbours.

drawback:

- Count-To-Infinity Problem:

I. Static Routing Algorithm:

1. Shortest Path Routing:

↳ Shortest path from a node to all others.

↳ Dijkstra Algorithm is used.

↳ Stop only when all unexplored node in list is left.

2. Flooding:

↳ Every incoming packet is sent out on every outgoing line except the one it arrived on.

↳ Good for Broadcasting but not for Unicasting and Multicasting.

3. Flow-Based Routing:

↳ Can be applied in few cases only.

↳ For a given line the capacity and average flow are known in advance. It is possible to compute Mean Packet

$$T_i = 1 / (\mu c_i - \text{lambda}_i)$$

where,

lambda i is mean flow in packets/sec

given - mean packet size is 800 bits.

This $1/\mu$

Weight $\sum \lambda_i = 82$ (from table).

∴ all divided by 82 is weight column.

Ex: $14/82 = 0.171$.

Actual Mean Delay = T x Weight

II. Dynamic RA:

2. Link State Routing:

Each router must

32-bit sequence numbers are used in today's networks.

1. Discover its neighbours and learn their network addresses.
2. Measure the delay or cost to each of its neighbours.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

1. Discover about Neighbours: Send Hello Packets.

2. Measure Line Cost:
Echo packet are transmitted and waits to receive to calculate propagation delay.

3. Building Link State Packets.

A	
Seq.	
Age	
B	4
E	5

★ 4. Hierarchical Routing:
↳ Kamoun and Kleinrock:
↳ N Routers
↳ ~~ENT~~ Two hierarchical numbering

↳ e (log) entries for each router.

★ Congestion:
Goal:
↳ make sure the subnet is able to carry the offered traffic.
Causes:
↳ When packets arrive it stores it in input buffer. But when packets arrive too fast, it does not find space in buffer.
↳ Even large buffer can not prevent congestion, because of delay, timeout and retransmission. because if the packet is at the top of input buffer it would require just to process and pass it to the next node. But in that time timeout occurs and sender sends the packet again.

Congestion Control.	Flow Control.
All nodes are involved.	Just sender & receiver
~~The~~ Network Layer.	Data Link Layer.

• Prevention Policies:
1. Open Loop (Static Decision):
2. Closed Loop (Dynamic Decision):
1. Explicit Feedback: Packets are sent back

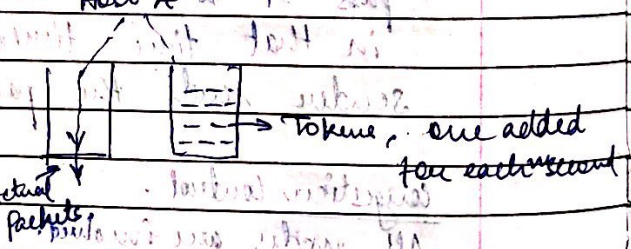
from the point of congestion.

2. Implicit: Sender observes for the time required for Acks.

1. Open Loop:

A. Leaky Bucket: (LB)
↳ never changes according to scenario, even if the input rate increases.
↳ It operates at the same rate that it is programmed to.

B. Token Bucket: (TB):
↳ Allows the output rate to vary depending on the size of the burst.



C: Token Bucket Capacity, ρ: token arrival rate,
S: burst length, M: maximal output rate.

$\Rightarrow C + \rho S = MS$

Bandwidth ∝ to size of Token Bucket.

Ex: 1. 25 MB/sec.

$40 \times 10^{-3} \times 25 \times 10^6$ B/sec.

$= \frac{1000 \times 10^3}{500}$ B.

$= 2 \times 10^3$ B/sec.

$= 2 \times 10^6$ MB/sec

2. $\rho = 2$ KB/msec. for 2 MB/sec.

$C = 250$ KB.

$M = 25$ MB/sec.

$= 25$ KB/msec.

$\therefore C + \rho S = MS.$

$M - \rho S = C.$

$S = \frac{C}{M - \rho}.$

$= \frac{250}{25 - 2}.$

$= \frac{250}{23} = 11$ msec.

In 11 ms 275 KB data is sent,

$\Rightarrow$ 725 KB data left.

New data will flow at rate 2KB/msec i.e., token arrival rate $\Rightarrow$ remaining 725 KB of data will be transferred in $\frac{725}{2} = 362$ msec.

★ Closed Loop Congestion Control:

↳ Admission Control.

★ 1. Choke Packets:

2. Hop-by-Hop Choke Packets:

3. Load Shedding:

4. Jitter Control:

Ex: (From previous page)

(i) Capacity of token bucket = 250 KB.

$C + Ss = Ms$.

$S = \frac{C}{M-S} = \frac{250\ KB}{25\ MB/sec - 2\ KB/msec}$

$\Rightarrow \frac{250\ KB}{25\ KB/sec - 2} \cong 11\ msec.$

25 KB/msec for 11 msec.

$= 25 \times 11 = 275\ KB$. → tokens over.

$1000 - 275 = 725\ KB$ remaining.

now, tokens generate at 2KB/msec.

∴ Remaining 725 KB will go at 2KB/msec

$\frac{725}{2}$ msec = 362 msec.

∴ 725 moves with 2KB/msec for 362 msec

(ii) C = 500 KB.

(iii) C = 750 KB.

(iv) find o/p from a 500KB token bucket feeding a 10 MB/sec leaky bucket.

∴ M = 10 MB/sec = 10 KB/msec.

$S = \frac{500}{10-2} = \frac{125}{8} = 62.5\ msec \cong 62\ ms$

★ Numericals:

1. A Computer on a 6 Mbps Network is regulated by a token bucket. The token bucket is filled at the rate of 1 MBps. It is initially filled to capacity with 8 Mbs. How long can a computer transmit at the full 6 Mbps.

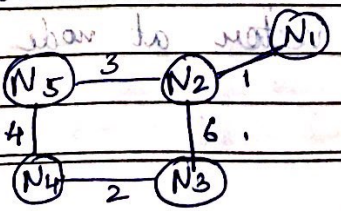
Solution: $\rho = 1$ Mbps.

C = 8 Mbps.

M = 6 Mbps.

$S = \frac{C}{M-\rho} = \frac{8}{6-1} = \frac{8}{5} = 1.6\ sec.$

2. Consider a network with 5 nodes, n1 .. n5, as shown below



The Network uses Distance vector routing protocol. Once the routers have stabilized the Distance Vector at different nodes are as following:

$N_1 : (0,1,7,8,4)$.

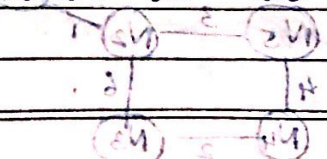
$N_2 : (1,0,6,7,3)$.

$N_3 : (7,6,0,2,6)$

$N_4 : (8,7,2,0,4)$.

$N_5 : (4,3,6,4,0)$.

Each Distance vector is the distance of best known path at that instance to node $N_1 \ldots N_5$; where distance to itself is 0. Also all links are symmetric and the cost is identical in both directions. In each round all nodes exchange their distance vector with their respective neighbors. Then all nodes update their distance vector. In between two rounds, any change in the cost of the links will cause to change two incident nodes to change only that entry in their distance vector. The cost of the link N_2/N_3 reduces to 2. After the next round of update what will be the new distance vector at node N_3.



Soln: New distance vector routing table for N_3 is $N_3 : (3,2,0,2,5)$.

N_3 receives Distance Vector from N_4 & N_2. But N_4 has no change in routing table, so it considers N_2.

$\therefore N_2 = (1,0,2,7,3)$.

initial $N_3 = (7,6,0,2,6)$.

after receiving updated N_2,

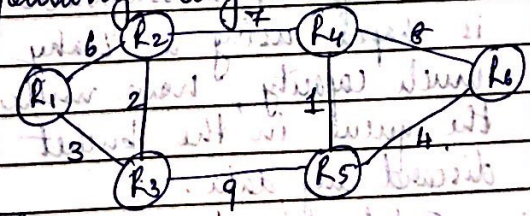
$(N_3 \text{ to } N_2) + (N_2 \text{ to } N_1) = 2+1 = 3$.

$\therefore N_3 \text{ to } N_1 = 3$.

similarly for all.

now $\boxed{N_3 = (3,2,0,2,5)}$

3. Consider a network with 6 routers $R_1 \ldots R_6$ connected with links as shown in the following diagram.



All the routers use DVR to update routing table. Each router starts with

the routing table initialized to contain entry of each neighbour with the weight of the respective connecting link. After all the routing table stabilized, how many links in the network will ~~be used~~ never be used to carry any data.

Solu: $R_1 = (0, 5, 3, 12, 12, 16)$. $(R_1 - R_2), (R_4 - R_6)$

$R_2 = (5, 0, 2, 7, 8, 12)$ $(R_1 - R_2), (R_4 - R_6)$

$R_3 =$ —"—

$R_4 =$ —"—

$R_5 =$ —"—

In all cases above $(R_1 - R_2)$ & $(R_4 - R_6)$ are the links which are never used.

4. Computer A has (19.5) MB. to send on a network & transmits the data in a burst 6 Mbps. The maximum transmission rate across routers in the network is 4 Mbps. If a Computer A's transmission is shaped using a Leaky Bucket, how much capacity, how much capacity must the queue in the bucket hold not to discard any data.

Solu: Total data to send = 19.5×8 MB

Transmission rate from computer = 6 Mbps

Time taken to send this data = $\frac{19.5 \times 8}{6}$ sec

Now, network transmission rate = 4 Mbps.

∴ Leaky bucket outputs as 4 Mbps.

I/p to bucket is at 6 Mbps.

∴ $6 - 4 = 2$ Mbps.

∴ Data stored in bucket is at rate = 2 Mbps

∴ Total data stored = $2 \text{ Mbps} \times \frac{19.5 \times 8}{6}$

$= \frac{8}{16} \times 19.5$ MB.

$= 156/3 = \boxed{52 \text{ MB. = Capacity of bucket.}}$

UNIT-6.

TRANSPORT LAYER.

- Implemented in the Operating System.
- deals with quality of service for end to end
- Services:
  - ↳ Efficient, reliable, cost-effective service.
  - ↳ is connection oriented
    - ↳ connection less.
- Service Primitives: Basic:
  1. Listen.
  2. Connect.
  3. Send.
  4. Receive.
  5. Disconnect.
- Berkely Service Primitives:

1. Addressing:
   TSAP — Transport service Access Point.
   — Interfaces between Application layer & Transport Layer (Access Point).
   — Converts App. data into segment.

   stable TSAP address = constant address.

   • Process Server is the solution for stable TSAP address problem.

2. Connection Establishment:

3. Connection Release:
   - ↳ Asymmetric. : abrupt → result in data loss.
   - ↳ Symmetric.
     - ↳ Both parties should agree to release connection.
     - ↳ Two-army problem.

     Solution:

4. Flow Control and Error Control.

★ Multiplexing:

★★★ CRASH RECOVERY:

	Sender		Receiver.
Initially.	S0		
send data	S1	data →	
Receive data	S0	← Ack.	

Sender stays in S1 till Ack is received.

★★ User Datagram Protocol:
— provides connectionless service.
— 8 byte header.

DLL = fixed window size.
TL = Varying window size.

• Transmission Control Protocol:

* TCP Congestion Control or Internet C. C:

Sender → Receiver (rwnd)

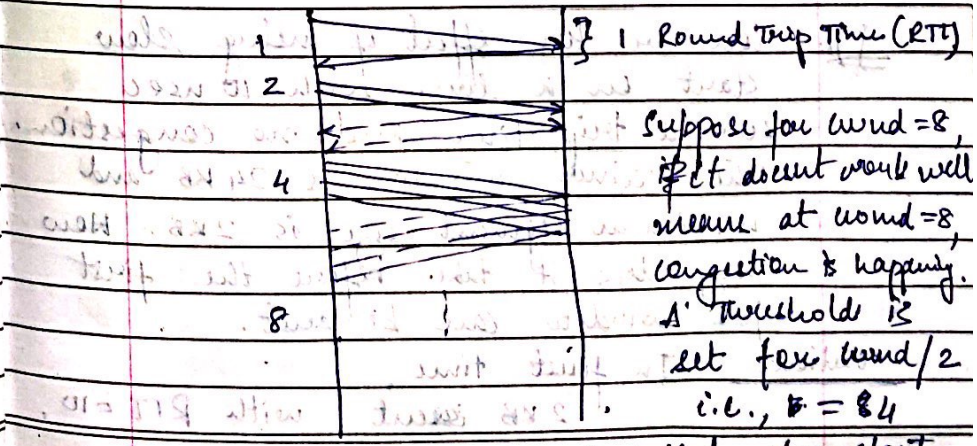
congestion window (cwnd).

Initial cwnd 64 KB.

1). Slow Start + Congestion Avoidance / Slow Start Additive Increase. (Exponentially Increase)

If congestion is detected, the cwnd = 1, where 1 frame size is 1 Byte

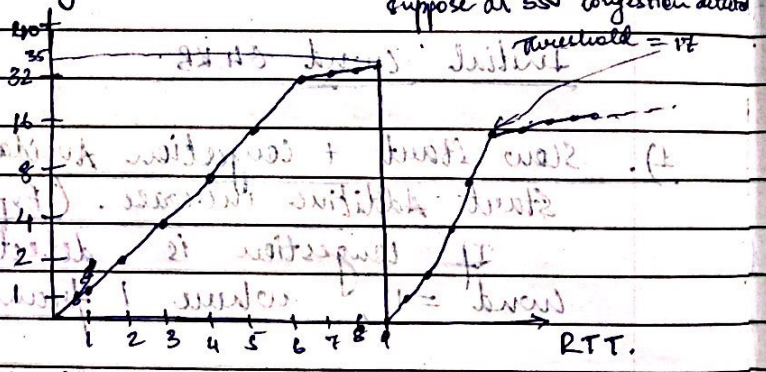
Then if Ack is received, then increase rate in 2^0.



Suppose for cwnd = 8, if it doesn't work well means at cwnd = 8, congestion is happening. A threshold is set for cwnd/2 i.e., t = 64

This is called slow start threshold.

When the cwnd reaches Threshold it increases by 1. and when again congestion is detected its new Threshold value is set and then again cwnd starts from 1 to Threshold value in exponential way and then increase by 1.



2). Fast Retransmission:

Q1. Consider the effect of using slow start on a line with 10 msec round trip time and no congestion. The receiver window is 24 KB and the maximum segment size is 2 KB. How long does it take before the first full window can be sent.

Solution: In first time, 2 KB sent with RTT = 10.

Then,
4 KB = 10.
8 KB = 10
16 KB = 10.
32 KB — cant because receiver = 24 KB.
∴ 10 + 10 + 10 + 10 = 40 msec before 24 KB.

2. Suppose that the TCP congestion window is set to 18 KB and a timeout occurs. How big will the window. if the next 4 transmission burst are successful. Assume that the maximum segment size is 1 KB.

Solution: cwnd = 18 KB & timeout occurs. congestion detected.
Threshold = 18/2 = 9 KB.
Then we start at cwnd = 1 segment.
first four burst = 1, 2, 4, 8 segments.
data = 1, 2, 4, (8 KB).

3. Let the size of congestion window of TCP connection of 32 KB when timeout occurs. The RTT of connection is 100 msec, and maximum segment size is 2 KB. Calculate the time taken (in msec) by a TCP connection to get back to 32 KB congestion window.

Page No.
Date :

Soln: ∴ Slow threshold = 16 KB. ∴ = 32/2 = 16

2, 4, 8, 16, 18, 20, 22, 24, 26, 28, 30, 32.

Headed threshold. Increased here by 1 segment

Here.

11 × 100 = 1100 msec to reach 32 KB window size

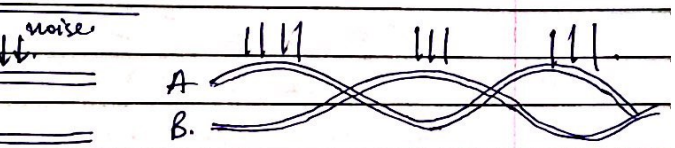
Page No.
Date :

UNIT – 2

PHYSICAL LAYER.

- Guided Media:

1. Twisted Pair:



- Unshielded.
- Shielded.

2. Co-axial:

3. Fiber Optics:

- Unguided Media:
  ↳ use Electro-Magnetic waves.

1. Radio Waves:

2. Terrestrial Microwave.