Lecture 39, 40, 41, 42, 43 → HDLC, BISYNC, DDCMP, PPP, Bit stuffing.

39.

(1) HDLC [High-level Data Link Control].

⇒ SDLC (by IBM) → Standardised by ISO to HDLC.

(2) ⇒

| Beginning sequence (8 bit) | Header (16 bit) | Body (variable acc to data) | CRC (16 bit) | Ending sequence. (8 bit) |
|---|---|---|---|---|

Frame format.

(3) ⇒ Beginning = Ending seq = 01111110.

Used for start & end of frame.

Also if link idle for a long time, beg & ending seq transmitted to keep link alive.

DATE / /

(4) Header → Address and Control field

Body → Payload (variable size)

CRC → Cyclic Redundancy check — Error Detection.

(5) Types of HDLC frames
- → I - frame
- → S - frame
- → U - frame.

↓

Type of HDLC identified by control field in header

**I - frame** (Information)

⇒ 1st bit is 0.

**S - frame** (Supurvisory)

⇒ 1st 2 bits 0 (Error detection)

U - frame (Un-numbered).

⇒ 1st 2 bits 11 (Mislineous activites)

1/10.

(6) **Bit Stuffing**

⇒ If we have 01111110 in middle of data instead of in beg & ending the computer will insert a zero after 5 ones.

⇒ 011111010. → to minimize errors.

41

## (7) BI-SYNC (Data link).

⇒ Binary Synchronous Communication Protocol.

⇒ A byte oriented approach protocol.
(character)

⇒ Frame format

(1 byte)

⇒ 8 bits (SYN) + 8 bits (SYN) + 8 bits (SOH) + Header + 8 bits STX
+ Body + ETX 1 byte + 16 bits CRC (error detection).

⇒ First 8 bits (1 byte) is for synchronization.
+ 8 bits

⇒ Second SOH → Start of header.
then header

⇒ 3rd → Start of text (STX) and End of Text (ETX).

⇒ 4th → Body variable size.

⇒ 5th → CRC → Error detection.

* Like bit stuffing, in BI-SYNC there is character or byte stuffing

42.

## (8) PPP (Point to Point Protocol) [V. Imp]. [Data link layer protocol Usually used in WAN for Internet.]

⇒ Frame format

[Handle heavy loads].

| 8 bits | 8 | 8 | 16 | | 16 | 8 |
|---|---|---|---|---|---|---|
| Flag | Address | Control | Protocol | Payload | Checksum | Flag |

→ Flag → Marks beg & end of frame.

Address → Usually all 11111 in case of broadcast.

Control → Remains constant (11.000 000).

Protocol → Tells the type of data in payload.

Payload → variable length, max length can be decided.

Checksum → Error detection technique.

Flag → end. flag (as mentioned above).

43.

⑨ <u>DDCMP</u> (Digital Data Communication Message Protocol).

→ Data Link layer.

→ It's using a byte counting approach.

<u>Frame format</u>

→

| (8) | (8) | (8) | (14) | (42) | Var | (16) |
|---|---|---|---|---|---|---|
| SYN | SYN | Class | Count | Header | Body | CRC. |
| | | ↓ like protocol in PPP | | | | ↓ Error detection |

<u>Drawback</u> → Count field, if corrupted then the receiver doesn't know where data ends (or) undercounts it

# Lecture 44 – 50 → Error Detection protocols.
[VRC, LRC, checksum, CRC]

44.

## ① Error Detection & Correction

→ Error is a corruption of data during transmission

→ For reliable communication, error must be detected and corrected.

→ Detection & Correction are implemented at transport layer (or) data link layer.

## ② Types of errors

Types of errors → Bit Error.
→ Burst Error.

### Bit Error

→ only a single bit error.

→ i.e only 1 bit in data unit has been changed.

0 | 1 | 0 | 0 | 1 | 1 | 0 | 1      0 | 1 | 1 | 0 | 1 | 1 | 0 | 1

### Burst Error

→ In burst error 2 or more bits are changed.

→ length of burst error is not only bits corrupted but everything b/n them too.

## ③ Detecting Errors → Done by Receiver.

P.T.O

→ But for receiver to detect (or) correct errors, some extra bits are sent by sender. → called redundant bits. (Present in format).

③ → **Redundancy**

→ Generating fn. → generates Redundance bits add to data in sender.

Checking fn → Checks redundance bits and on basis of that accepts (or) rejects data.

④ → **Error correction**

→ tedious than detection.

i) Retransmit data

(or) ii) Error correcting code for specific tasks.

45.

⑤ Error detection techniques

→ VRC
→ LRC
→ Checksum
→ CRC.

⑥ **VRC** [Vertical Redundancy Check].

→ Also known as parity check.

→

Data [110001] → Even Parity Generator → 1 in this case

↓

Counts no: of 1s.

If odd → 1 appended

If even → 0 appended

∴ final data transmitted → $\boxed{1\,|\,1\,|\,1\,|\,0\,|\,0\,|\,0\,|\,1}$.

Drawbacks → Cannot detect burst error. if number of error is even.
Can only detect when no: of ones = odd.

~~→ Can always detect bit error.~~

46.
(7) LRC [longititional]

⇒ Block of bits organized in rows and columns.

∴ ⇒ Also called 2-Dim Parity.

⇒ Parity bit is calculated for each column.

```
11101110
11011101
10111001
↓
0- - - ..0
```

} Every column generates
⇒ odd no: of 1s → 1.
even no: of 1s → 0

⇒ Better in detecting burst errors. than VRC.

47.
(8) Checksum

→ ~~We check the sum. and attach to message~~

At Sender. → Break original message into 'k' blocks.
⇒ Sum them
⇒ Add the carry if generated
⇒ Do 1s compliment.

PAGE No
DATE / /

Many problems of Checksum in GATE and University Exams. ∴ Do problems. if necessary.

⇒ Receiver again follows same steps & compares it's answer with checksum value received.

⇒ ~~Detect all errors involving odd number of bits & most errors involving even number of bits.~~

~~48.~~

## CRC [Cyclic Redundancy Check]

(9) ⇒ CRC Generation at Sender side ⇒ Find length of divisor 'L'.
⇒ Append (L − 1) bits to original message.
⇒ Perform binary division operation
⇒ Remainder of division = CRC. → Must be of L−1 bits

Instead of subtraction XOR performed.

Eg:∴ [Data ⇒ 100100] [Divisor ⇒ 1101]
[CRC ⇒ 001 after steps]
[Final ⇒ 100100001.]

[Dividend ⇒ 100100000]

~~49. & 50.~~

(10) ⇒ CRC at Receiver side → when divisor [1101] divided by final [100100001] the remainder should be 0.

Sample Problem solved for practice.

PAGE No.
DATE / /

# Lecture 51 – 56 → Network Performance, Latency (Delay) Bandwidth & RTT.

51.

① Network performance is measured in 3 ways.

→ Bandwidth
→ Throughput
→ Latency.

② → Highway → Can take 1000 per minute. (bandwidth)
→ But due to congestion, can only take 100 cars now (throughput).
→

③ → Bandwidth. (Capability).

→ Amt of data (max) that can be transmitted by a network per second.

2 ways to express → 1 GBps / for ethernet (bps)
→ Range of freq (Hertz)

④ → Throughput (Reality)

→ Actual amt of data that passes through the medium

⑤ Latency (delay).

→ Time taken for entire message to arrive at the destination from the time the first bit is sent out from the source.

Components → Transmission delay. (~~b/w network~~)
→ Propogation delay (~~Sender prob~~).
→ Queuing delay
→ Processing delay (~~Receiver problem~~).

PAGE No.
DATE / /

CSS-labs.org → Java applet

52,53

(6) Components of latency.

→ Transmission delay → Time taken to place complete data on the transmission medium from device A.

→ Propagation delay → Time taken for data to go from device A to device B.

→ Queuing delay (changes i.e variable) → Holding time for intermediate nodes before they pass it to next.

→ Processing delay → Time by receiver to accept message.

(7) formula of (Trans delay) → $\frac{\text{message size}}{\text{Bandwidth}}$

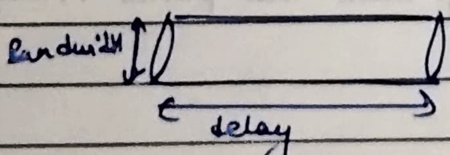
(Prop delay (time)) → $\frac{\text{Distance}}{\text{Prop speed}}$

Only 2 formulas.

Generally speaking Queuing & processing delay are not taken into account.

54

(8) Bandwidth delay product defines number of bits that can fill the link.




Volume = Bd × delay
(bits) (bps) (time)

55&56

(9) RTT → (Round trip time) = 2 × Propagation delay.

→ Length of time for signal to be sent

" " for Acknowledge to be received.