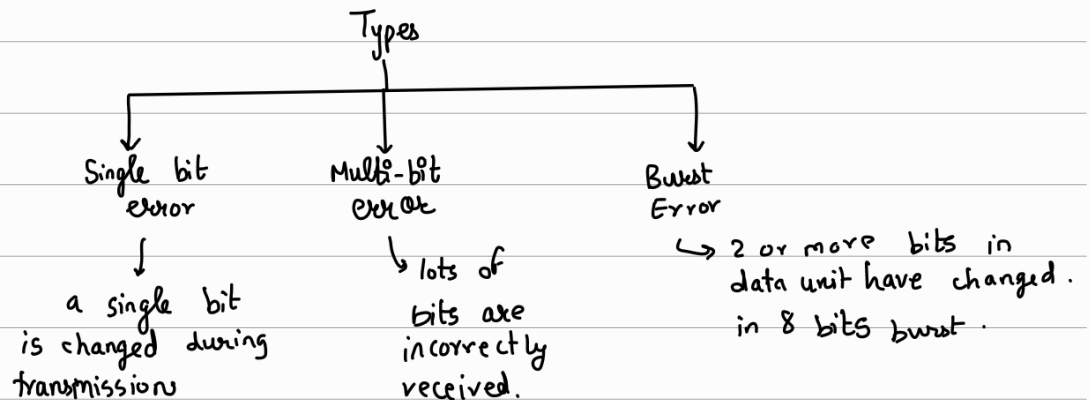


UNIT IV

Error Detection and Correction.

what is an error?

- Corruption of data in transmission is error.
- Error may be caused by noise, human errors, natural disasters, etc.



Redundant bits

To detect and correct data sender will add some extra redundant bits to the data.

Redundancy achieved through coding (block coding).

receiver can detect error given that

1. receiver has a list of valid codewords.
2. Original codeword has changed to an invalid one.

what is burst error?

Datawords: • k-bits long blocks of the original message.

• we add r redundant bits.

codeword \rightarrow n length ' $n = k + r$ '

<u>e.g.</u>	Datawords	Codewords
	00	000
	01	011
	10	101
	11	110

Hamming distance: • Hamming distance between two words (of same size) is number of difference between the bits.

e.g. (000, 011)

0 0 0
0 1 1
 \rightarrow x ✓ ✓

2 ✓ \rightarrow 2 difference, 2 hamming distance

IF s-error occurred during transmission, the Hamming distance is s.

• Min hamming distance should be 1 more than the size of codeword.

if we don't do this +1, it might be mistaken for data bit.

Parity bit - a bit added to make parity (1's occurrence even).

• appended with dataword \rightarrow forms codeword.

Parity Check Decoder

DW \rightarrow 1011

CW \rightarrow 10111
 \uparrow
 r_0 - Parity bit

Cases:

1. No error. 10111 received $\rightarrow 1+0+1+1+1 \cdot 1.2 \rightarrow 0$

1011 extracted.

2. 1 bit changes at r_0 , 10011 $\rightarrow 1+0+0+1+1 \cdot 1.2 \rightarrow 1$ Syndrome bit

3. 1 bit changes at r_0 , 10110 $\rightarrow 1+0+1+1+0 \cdot 1.2 \rightarrow 1$ Syndrome bit

2,3 \rightarrow dataword is discarded.

4. Even number of errors cannot be detected, as syndrome will be 0.

5. Three bits are changed, syndrome will be 1, any odd number of errors can be found here. (works for odd errors).

Cyclic Redundancy
Check (CRC)

DW \rightarrow R bits

CW \rightarrow n bits

a divisor of size ' $n-k+1$ ' is used

DW \rightarrow 1001 $\xrightarrow{\text{while in generator}}$ 000

$$\begin{array}{r} 1010 \\ 1011 \overline{) 1011000} \\ \underline{-1011} \\ 010000 \\ \underline{-0000} \\ 10000 \\ \underline{-1010} \\ 00100 \end{array}$$

check this
out again!

Checksum:

• Message divided into m -bits

e.g.

received checksum, \rightarrow take 1's complement \rightarrow add 1's complement with data words \rightarrow if checksum is more than ' m '

bits, divide into m bits and add them together

\rightarrow take 1's compli again \rightarrow Syndrome = 0000 not corrupted.

Algo:

1. Split data in ' m ' bits.
2. Convert bits (of size m) to decimals and add them. (converting is optional).

Data = 01110011, $m = 4$
 \hookrightarrow block of 4 bits

3. Take complement of addition (on sender side itself.)

4. (On receiver) add m bit data words with checksum value.

$$\Rightarrow \underbrace{0+1+1+1}_{7} + \underbrace{0+0+1+1}_{3}$$

5. Take 1's complement of result.

$$\Rightarrow (10)_{10} \Rightarrow (1010)_2$$

- a. if result = 0
 \uparrow
Syndrome

$$1's \text{ complement} \Rightarrow (0101)_2$$

\Downarrow receiver

no corruptions.

$$\underbrace{0+1+1+1}_{7} + \underbrace{0+0+1+1}_{3} + \underbrace{0+1+0+1}_{5}$$

- b. else
data corrupted.

$$(15)_{10} \rightarrow (1111)_2$$

$$1's \text{ complement} \Rightarrow (0000)_2$$

\therefore Syndrome = 0000 \therefore No errors/data corruptions.

Internet Checksum

- Traditionally 16-bit checksum.

8-bit Fletcher:

- data octets (bytes)

•

Adler checksum

→ 32-bit checksum.

→ 3 differences

Forward Error
Correction

- Hamming Distance
- XORing
- Chunk interleaving
- Combining Hamming distance & interleaving
- Compounding High and Low resolution packets.