Hamming Codes

practice 17: (a) 01011 ~~da~~ codeword (or C/W)

from table 10.2 : ~~da~~ corresponding data 01

(b) C/W → 11111 ; from table, ~~d/w~~ this C/W

not exist ∴ c/w is corrupted.

according to example 10.3, do I assume I have 1 bit corrupted?

minimum hamming distance দিয়ে determine করা যায় কয়টা bit correction করা যাবে।

Table 10.2 has $d_{min} = 3$ ∴ ~~2 bit~~ t bit can be corrected.

$$\cancel{dim}\ d_{min} = 2t + 1 \Rightarrow 3 = 2t + 1$$

$\therefore t = 1$ ∴ 1 bit can be corrected

Comparing recieved c/w with table 10.2's codeword.

table c/w	recieved c/w	difference
00000	11111	5 bit
01011		2 bit
10101		2 bit
11110		1bit

matches, t

∴ replace 11111 with 11110 as c/w

now, d/w is 11

17(c) c/w → 00000 , d/w → 00

(d) c/w → 11011 , corrupted.

table c/w	received c/w	difference
0 0 0 0 0		4 bits
0 1 0 1 1	1 1 0 1 1	~~2~~ bits 1 bit
1 0 1 0 1		3 bits
1 1 1 1 0		2 bits.

~~However, I can only correct 1 bit corruption.~~

replace c/w 11011 with 01011

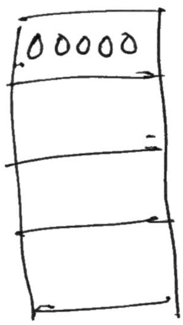
∴ c/w → 01011 , d/w → 01

(18) linearity. ⊕ will give another valid c/w.

~~00000 ⊕ 01011~~

01011 ⊕ 10111 = 11100 (violation)

∴ not linear.



Binary CRC

d/w → 1011001

divisor → 111 (length 3)

dividend → 1011001$\underbrace{00}$

(3-1)=2

2 extra zeros adding after d/w

```
111) 101100100 (111101
     111
 ⊕   0101
      111
 ⊕    0100
       111
 ⊕    ~~0111~~
       0110
        111
 ⊕      0011
         110
         111
 ⊕       0010
```

$\underbrace{10}$ remainder

same amout of digits

this is senders side.

remainder size is same as the size of exta zeros added to make a augmented d/w (aka dividend)

hence the codeword is : $\boxed{\overbrace{1011001}^{d/w}\overbrace{10}^{remainder}}$

on reciever side.

divisor → 111

c/w → 101100110

```
111 ) 1 0 1 1 0 0 1 1 0 ( 111101
      1 1 1
  ⊕
      0 1 0 1
        1 1 1
  ⊕
        0 1 0 0
          1 1 1
  ⊕
          0 1 1 0
            1 1 1
  ⊕
            0 0 1 1 1
                1 1 1
                1 1 1
  ⊕
                0 0 0 0
                syndrome
```

syndrome is all zero, that means correct codeword.

∴ data word accepted.

Polynomial Cyclic Codes

dataword → 1001 → $x^3+0+0+x^0 \to x^3+1$

divisor → 1011 → $x^3+0+x^1+x^0 \to x^3+x+1$

dividend (augmented d/w) → $\underbrace{x^3(x^3+1) \to x^6+x^3}_{\text{highest orderded term} \times \text{whole d/w}}$

gender side

$$x^3+x+1 \overline{)\ \cancel{x^6}+\cancel{x^3}\ } (x^3+x$$

$$(-)\ \underline{\cancel{x^6}+x^4+\cancel{x^3}}$$

$$\cancel{x^4}$$

$$(-)\ \underline{\cancel{x^4}+x^2+x}$$

$$x^2+x$$

no more >= term available

$\therefore x^2+x \to$ remainder

divisor) dividend (- - -

dividing until dividend has a higher or equal ordered term as the divisor

$\therefore$ The codeword is $\underbrace{x^6+x^3}_{\text{augmented d/w}}+\underbrace{x^2+x}_{\text{remainder}}$

~~so~~ recievers side:

divisor $\rightarrow$ x^3+x+1

code word $\rightarrow$ $x^6+x^3+x^2+x$

$$
\begin{array}{r l}
x^3+x+1 \,\big) & \cancel{x^6}+\cancel{x^3}+x^2+x \;\big(x^3+x \\
& \cancel{x^6}+x^4+\cancel{x^3} \\
(-) & \overline{\qquad\qquad\qquad\qquad} \\
& x^4+x^2+x \\
& x^4+x^2+x \\
(-) & \overline{\qquad\qquad\qquad\qquad} \\
& 0 \leftarrow \text{syndrome/remainde}
\end{array}
$$

$\therefore$ syndrome is $\emptyset$, the dataword is accepted

Question 1:

Dataword $\rightarrow$ $x^6+x^4+x^3+x+1$

divisor $\rightarrow$ x^4+x^2+x+1

augmented d/w $\rightarrow$ $x^4(x^6+x^4+x^3+x+1)$

$\rightarrow x^{10}+x^8+x^7+x^5+x^4$

$$
\begin{array}{r l}
x^4+x^2+x+1 \,\big) & \cancel{x^{10}}+\cancel{x^8}+\cancel{x^7}+x^5+x^4 \;\big|\; x^6+x^2+x \\
& \cancel{x^{10}}+\cancel{x^8}+\cancel{x^7}+x^6 \\
(-) & \overline{\qquad\qquad\qquad\qquad} \\
& x^6+x^5+x^4 \\
& x^6+x^4+x^3+x^2 \\
(-) & \overline{\qquad\qquad\qquad\qquad} \\
& x^5+x^3+x^2 \\
& x^5+x^3+x^2+x \\
& \overline{\qquad\qquad\qquad\qquad} \\
& x
\end{array}
$$

the remainder, x has less order than the divisor(4). So terminate operation

$\therefore$ code word $x^{10}+x^8+x^7+x^5+x^4+x$

Checksum (Hex)

example: packets are 16 bit (i.e, 4 Hex digit)

sender side

sender adds all his packet to calculate checksum

```
  4 6 6 F
  7 2 6 F
  7 5 7 A
  6 1 6 E
----------
(1) 8 F C 6
 └─overflow──→ 1
----------
    8 F C 7
```

this is warp sum

using calculators not()

I get FFFF7038

keeping the last 4 Hex digits only as per questions packet size → $\boxed{7038}$

this is the checksum

reciever get the checksum value from sender and add it with all frames to cross check checksum

```
  4 6 6 F
  7 2 6 F
  7 5 7 A
  6 1 6 E
  7 0 3 8
----------
(1) F F F E
 └─overflow──→ 1
----------
    F F F F
```

~~not()~~

not()

$\boxed{\text{FFFF 0 0 0 0}}$

keeping last 4 Hex digit

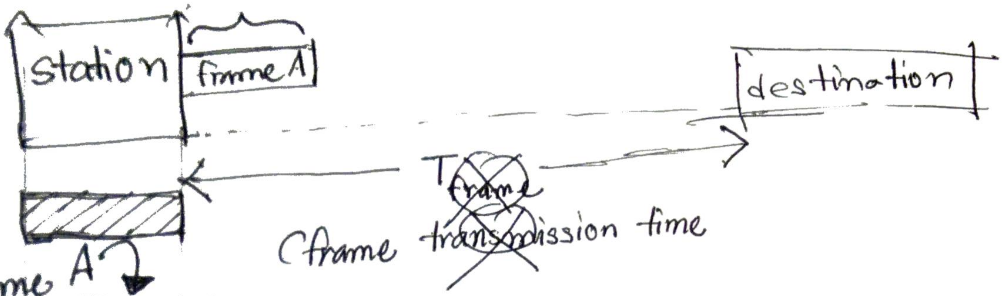
$\boxed{0000}$ (all zero)

data is correct

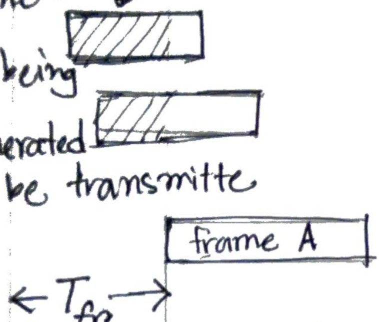
ALOHA → one of the random access protocols.

Pure Aloha

Throughput



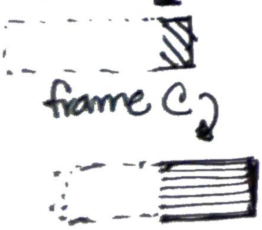
frame A is being generated to be transmitte



takes T_{fr} time to release entire frame into the medium

frame B

frame C



during frame A's creation (succesfully) frame B was 25% created and C was 75% created. so now the system have 2 frame of totall 100% size

that means, on average 50% of a frame generated. (0.5 G = 0.5)

Example 12.3

frame size 200 bit

channel speed 200 kbps.

$$\therefore T_{frame} = \frac{200 \text{ bit}}{200 \times 10^{3} \text{ bit}} s$$

$$\boxed{T_{fr} = 10^{-3} s \approx 1 ms}$$

(a) 1s → 1000 frames

$\therefore$ 1ms → 1 frame

$\therefore G = 1$ (avg frame generation in T_{fr} time)

throughput,

$$\therefore S = G \times e^{-2G}$$

$$= 1 \times e^{-(2 \times 1)}$$

$$= 0.1353$$

(13.5% success)

(c) 1000 ms $\rightarrow$ 250 frame

$\therefore$ 1 ms $\rightarrow$ 0.25 frame

$\therefore G = 0.25$

$\therefore S = 0.25 \times e^{-(2 \times 0.25)}$

$S = 0.1516$

(15.16% success)

Slotted ALOHA

Example 12.4

frame size 200 bit

channel speed 200 kbps

$\therefore T_{frame} = 1 ms$

(a) 1000 ms $\rightarrow$ 1000 frame

$\therefore$ 1 ms $\rightarrow$ 1 frame

(T_{fr} time amount)

$\therefore G = 1$

throughput for slotted ALOHA.

$\therefore S = G \times e^{-G}$

$= 1 \times e^{-1}$

$= 0.367$

36.7%

for this

(b) 1000 ms $\rightarrow$ 500 frame

$\therefore$ 1 ms $\rightarrow$ 0.5 frame

$\therefore G = 0.5$

$\therefore S = 0.5 \times e^{-0.5} = 0.303$

or (30.3%)

(c) $G = \frac{250}{1000} = 0.25$

$\therefore S = 0.25 \times e^{-0.25}$

$= 0.194$

(19.4%)