TD 03 : Détection et correction des erreurs.

1. Soit le message composé de la chaîne : "NET", le contrôle de transmission de chaque caractère est assuré par un bit de parité impair, donner la représentation binaire du message transmis. On suppose que les caractères sont codés selon le code ASCII, en utilisant 7 bits. Rappel : Le code ASCII des caractères transmis sont : N : 1001110, E : 1001001, T : 1010100

*1001110**1** 1001001**0** 1010100**0***

2. Montrer qu'un code dont la distance de Hamming est *d* peut détecter *d-1* erreurs.

Par définition la distance de Hamming d'un code est égale à la distance minimale entre deux mots valide du code. Si la distance d'un code est égale à d, l'altération de d-1 bits d'un mot de code valide ne donnera pas un autre mot valide du code.

3. Un code correcteur d'erreur contient les quatre mots suivants :
   0000000000
   0000011111
   1111100000
   1111111111
   3.1 Que vaut la distance de *Hamming* de ce code ?.

   5.

   3.2 Combien d'erreurs peut-il détecter ? Et combien d'erreurs peut-il corriger ?

   il peut détecter 4 erreurs et corriger 2.

   3.3 Le récepteur reçoit le mot 1110000000, quel est le mot initial ?

   Le mot valide le plus proche est 1111100000. On propose ce mot comme correction.

4. Soit le message suivant : 0011111101. On rajoute à ce message un CRC calculé par le polynôme générateur $g(x)= X^2+X+1$. Quel est le message codé ?

   P(X) $= x^7+x^6+x^5+x^4+x^3+x^2+1$
   P(x). $X^2 = (x^7+x^4+1).\ g(x) + (\mathbf{x+1})$
   *Le message envoyé est : 0011111101**11***

5. Le nombre et les types d'erreurs détectables par le CRC dépendent des qualités du polynôme générateur G(X). On veut démontrer les propriétés suivantes :

   P1. Pour détecter des erreurs simples G(x) doit posséder au moins deux termes.
   P2. Pour détecter les erreurs doubles, le polynôme générateur G(x) ne doit pas diviser tout binôme de degré 1<i< n-1 où n est la taille du message à protéger.
   P3. Pour détecter les erreurs en nombre impair G(X) doit être un multiple de X+1
   P4.Un polynôme générateur de degré M, détecte les paquets d'erreurs de longueur L<= M.

Soit T(x) le polynôme correspondant à un message envoyé. Si le message a subit des erreurs ceci est traduit par le changmement de quelques bits. le message reçue T'(x) ser donc

T'(x) = T(x) + E(x) où E(x) est le polynôme correspondant aux erreurs survenues pendnat le transmission du message. Pour que g(x) soit capbale d e détecter les erreurs il ne doit pas diviser sans reste le polynôme E(x)

P1. Dans ce cas le polynôme E(x) = ^{xi} *où i <= degré de T(x).*
Si g(x) possède au moins 2 termes on peut écrire g(x) = $x^j + x^k$.
Quelque soit i,j,k (avec j≠k) le polynôme = $x^j + x^k$*. ne peut pas diviser* x^i *sans reste.*

P2. Une erreur double s'écrit E(x) = $x^{i}+x^{j}$ *avec i ≠ j. le degré de E(x) est <= n-1. Pour détecter une erreur double g(x) ne doit pas diviser sans reste tout binôme de degré <= n-1.*

P3. Pour démontrer cette propriété on va procéder à une démonstration par l'absurde.
Supposons que G(x) est divisible par x+1 et qu'il ne détecte pas les erreurs en nombre impair.
On a g(x) = (x+1) H(x) et on a E(x) = g(x) Z(x) donc E(x) =(x+1) H(x) Z(x) .
Pour x=1 on E(x) = 1 puisque le nombre d'erreur est impaire mais (x+1)H(x)Z(x) = 0 d'ou l'absurde.

P4. Un paquet d'erreur de taille m est une suite de m bits dans le message dont le premier et le dernier bit sont faux. Les bits intermédiaire peuvent être correctes ou faux. Une telle erreur s'écrit E(x) = M(x).xi où M(x) est de degré m-a et (n-m>=i>=0.
Pour que g(x) ne détecte pas cette erreur il faut qu'il divise E(x) sans reste. g(x) a au moins deux termes donc il ne divise pas xi. pour diviser E(x) sans reste il faut diviser M(x) sans reste. Or si M(x) est de degré inférieur à g(x) ne peut pas diviser M(x) sans reste.

6. Soit le message suivant : 1101011011. On rajoute à ce message un CRC calculé par le polynôme générateur X^4+X+1
   6.1 Quelle est la suite binaire générée par le codeur de code cyclique ?
   *1101011011**1110***
   6.2 Quelle est la trame transmise à la couche physique ?

   *1101011011111**0**0 (ajout d'un bit de transparence)*

   6.3 Les erreurs simples et en nombre impair sont-elle détectées ?

   On détecte les erreurs simple. mais il ne détecte pas les erreur en nombre impaire puisque il n'est pas multiple de x+1.

   6.4 Quelle est la taille des paquets d’erreurs détectés ?
   4