

Protocoles réseaux

TD n° 3 : Code de Hamming

Codes correcteurs

On rappelle qu'un code *détecte* k erreurs si, pour tout mot reçu avec au moins une erreur et au plus k par-rapport au mot envoyé, le code signale une erreur. Un code *corrige* k erreurs si, pour tout mot y' reçu avec au moins un bit d'erreur et au plus k par-rapport au mot y envoyé, y est le seul mot du dictionnaire tel que l'on peut obtenir y' en changeant au plus k bits dans un mot du dictionnaire. y' et k étant connus, le décodeur peut retrouver l'unique y et le produire, corrigeant ainsi l'erreur.

Code de Hamming

Attention, pour Hamming on numérote les bits d'un mot en partant de 1. Coder un mot A de m bits se fait en deux étapes :

D'abord, on insère un blanc (en décalant les autres bits vers la droite) à toutes les positions de bits dont le numéro est une puissance de 2. Ainsi :

$$A=011101000111101 \quad \text{devient :} \quad _ _ 0 _ 111 _ 0100011 _ 1101.$$

Puis on remplit le blanc en position 2^i comme bits de parité de certain bits du mot. Plus précisément, le bit numéro 2^i code alors la parité des bits de numéro j tels que j modulo 2^{i+1} est supérieur ou égal à 2^i . Ainsi 1 contrôle ceux de numéro 1,3,5,7... ; 2 contrôle 2,3,6,7,10,11... ; 4 contrôle 4,5,6,7,12 etc etc. Visuellement, en gras les bits contrôlés par le bit 1 : **$_ _ 0 _ 111 _ 0100011 _ 1101$**
 par 2 : **$_ _ 0 _ 111 _ 0100011 _ 1101$** par 4 : **$_ _ 0 _ 111 _ 0100011 _ 1101$** par 8 : **$_ _ 0 _ 111 _ 0100011 _ 1101$**
 et enfin par 16 : **$_ _ 0 _ 111 _ 0100011 _ 1101$** .

On remplace donc chaque blanc par le bit de parité des bits qu'il contrôle, et c'est ce mot qui est transmis.

Questions :

1. Transmettez (codez) 0110010 puis 011101000111101 puis 001100001101000111001111
2. Quelle est la redondance r ajoutée ? Donner une relation liant m (taille du mot initiale), n (taille après codage) et r .
3. Comment décoder un mot ? Comment vérifier s'il y a eu des erreurs ? Comment corriger une erreur, le cas échéant ?
4. Recevez (décodez) 1010110 puis 0010010110011100101. Il faut donner le mot décodé, et le mot corrigé en cas d'erreur (en supposant une seule erreur), puis le mot avant codage.
5. Combien d'erreurs ce code peut-il détecter ? Produisez un exemple d'erreur non détectée.
6. Combien peut-il en corriger ?
7. Montrer que, pour certaines valeurs de n , m et r (lesquelles ?) tout mot à n bits est soit un mot du dictionnaire, soit diffère sur un seul bit d'un et d'un seul mot du dictionnaire. En d'autres termes, que tout mot reçu est soit interprété comme étant correct, soit corrigé. Un code vérifiant cette propriété est dit *parfait*.

Exercice : Bit de parité sur Hamming (examen 2017)

On considère le code suivant : étant donné un mot w de m bits, on applique d'abord le code de Hamming à w , ce qui donne le mot w' , puis on rajoute à la fin de w' son bit de parité ($w'' := w'.b$ avec $b := |w'|_1 \% 2$), ce qui donne le mot w'' qui est transmis.

Pour les questions qui suivent, il n'y a pas à prouver les propriétés ni à détailler les algorithmes de codage/décodage du bit de parité ainsi que du code de Hamming vues en TD. Mais si nécessaire il faut les citer.

1. Codez $w = 01101011010$ en détaillant les étapes.
2. Montrer que ce code détecte 3 erreurs. Donner un exemple où 4 erreurs arrivent sans être détectées.
3. On admettra que ce code corrige une erreur mais pas deux. Donner l'algorithme de décodage, qui donne aussi une estimation du nombre d'erreurs qui ont eu lieu (en supposant qu'il y a eu moins de 4 erreurs) et transmet soit le mot décodé soit «erreur incorrigible».
4. Quelle est le nombre de bits ajoutés par le codage? Comparer (en termes de puissance de détection, de correction, et de bits ajoutés) avec Hamming.