

Protocoles réseaux

TD n° 2 : Codes correcteurs

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.

Exercice 1 : pouvoir d'un code

1. Montrer que si un code détecte k erreurs alors il en corrige au plus $k/2$.
2. Réciproquement montrer que si un code corrige k erreurs alors il en détecte au moins $2k + 1$.

Exercice 2 : matrice de parité

Une variante est le bit de parité par bloc. On écrit 8 octets sur une matrice 8x8, puis on rajoute une 9eme ligne et une 9eme colonne. Le bit de la i eme colonne (resp. ligne) est choisi pour que le nombre de 1 dans cette colonne (resp. ligne) soit pair.

1. Que vaut le bit en position (9;9) ?
2. Combien d'erreur sur un bit ce code détecte-t-il ?
3. Et combien en corrige-t-il ?

Exercice 3 : un tout premier protocole FEC

1. Proposer un code qui permet de corriger une erreur sur un bit **de numero connu** (on sait quel bit est erroné).
2. Maintenant on suppose que le mot reçu est sur l'alphabet $\{0, 1, ?\}$, et que chaque bit envoyé est soit reçu correctement soit transformé en '?'. Proposer un code qui permet de corriger une seule erreur (au plus un '?' est reçu).
3. Passons à un modèle où l'on dispose d'une donnée sur k blocs de b bits chacun, et que l'on envoie des blocs (éventuellement plus nombreux que k ou plus long que b) sur le canal. Des erreurs comme à la question 2 (des '?') peuvent survenir dans **au plus un bloc**. Proposer un code qui permette de reconstituer, *sans retransmission*, le bloc erroné.
4. On suppose maintenant que les données reçues sont binaires (modèle habituel, plus de '?') mais toujours que des erreurs peuvent survenir dans au plus un bloc. Que faire ?
5. Donner la redondance de ces codes. Est-ce un gros gâchis de bande passante ?
6. Question de culture générale : savez-vous ce qu'est un *RAID array* ? Connaissez-vous les protocoles RAID-3, RAID-4 et RAID-5 ?

Pour ceux qui veulent aller plus loin, sachez qu'il est possible d'anticiper la perte de x blocs en envoyant $k + 2x$ blocs seulement, grâce au code de **Reed-Solomon**.

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 :

$A=011101000111101$ devient : `__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 numero 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 erreur ? Comment corriger *une* erreur, le cas échéant ?
4. Recevez (décodez) 1010110 puis 1011011111011011 puis 0010010110011100101. Il faut donner le mot décodé, et le mot corrigé en cas d'erreur (en supposant une seule erreur).
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'autre termes, que tout mot reçu est soit interprété comme étant correct, soit corrigé. Un code vérifiant cette propriété est dit *parfait*.