Année 2009-2010

Examen de Protocoles Réseaux

éléments de corrections

corrigé partiel et non officiel,
à lire d'un œil critique

Exercice 1 :

Question 1 :

Le tableau donné est le suivant :

Mots sources	Mots codés
000	x00xx
001	00101
0xx	010xx
011	xx1x1
100	1001x
101	101x1
110	1100x
111	111xx

Le but est de remplacer les x par les bits correspondants, sachant que les mots de la première colonne ont été multiplié par une matrice inconnue pour donner les mots de la seconde colonne.

Méthode

Quelques évidences :

- La matrice par laquelle on peut multiplier des mots de taille 3, et obtenir des mots de taille 5, est une matrice 3x5

Cette matrice sera donc de la forme :

$$\begin{bmatrix} Y & Y & Y & Y & Y \\ Y & Y & Y & Y & Y \\ Y & Y & Y & Y & Y \end{bmatrix}$$

- Le mot 000 ne peut que donner 00000, quelle que soit la matrice par laquelle on le multiplie

- Le seul mot incomplet 0xx dans la colonne de droite est manifestement 010, car c'est le seul mot sur 3 bit qu'il manque, et qu'il est d'ailleurs à sa place si on compte en binaire (en 001 et 011).

Remarque : On peut éventuellement être paranoïaque, s'imaginer que l'enseignant a mis 2 fois le même mot dans le tableau pour brouiller les pistes, mais on s'aperçoit qu'aucun des autres mots ne peut avoir un code de la forme 010xx, c'est donc bien un nouveau mot, c'est donc bien 010.

Ce qui nous donne le tableau :

Mots sources	Mots codés
000	00000
001	00101
010	010xx
011	xx1x1
100	1001x
101	101x1
110	1100x
111	111xx

La seconde ligne du tableau est un vecteur unitaire (c'est-à-dire qu'il ne comporte qu'un seul 1), il correspond donc à une ligne de la matrice génératrice. Comme c'est le 3ème bit qui est à 1, il correspond à la 3ème ligne de la matrice.

En effet, si on a :

$$\begin{array}{c} \\ \\ \\ (001) \end{array} \begin{bmatrix} Y & Y & Y & Y & Y \\ Y & Y & Y & Y & Y \\ Y & Y & Y & Y & Y \end{bmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Si on sait multiplier 2 matrices, on voit que la dernière ligne de la grande matrice est forcement égale au résultat.

Y	Y	Y	Y	Y
Y	Y	Y	Y	Y
0	0	1	0	1

De même pour les lignes 3 et 5 du tableau, qui sont les 2 autres vecteurs unitaires, elles nous donnent une partie des 2 lignes restantes de la matrice :

1	0	0	1	Y
0	1	0	Y	Y
0	0	1	0	1

À présent, la ligne 4 du tableau nous dit ceci :

1	0	0	1	Y
0	1	0	Y	Y
0	0	1	0	1

(011) (x x 1 x 1)

On sait ainsi que :

- le 1er x vaut 0 + 0, soit 0
- le 2ème x vaut 1 + 0 soit 1
- le dernier Y de la 2ème ligne respecte Y + 1 = 1, donc ce Y vaut 0

Ce qui donne pour le tableau :

Mots sources	Mots codés
000	00000
001	00101
010	010x0
011	011x1
100	1001x
101	101x1
110	1100x
111	111xx

Et pour la matrice :

$$\begin{pmatrix} 1 & 0 & 0 & 1 & Y \\ 0 & 1 & 0 & Y & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Passons à la 6ème ligne du tableau :

$$(101)\begin{pmatrix} 1 & 0 & 0 & 1 & Y \\ 0 & 1 & 0 & Y & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & x & 1 \end{pmatrix}$$

On reprend le même raisonnement :

- 1 + 0 = x, donc x = 1
- Y + 1 = 1, donc Y = 0

Ce qui donne :

Mots sources	Mots codés
000	00000
001	00101
010	010x0
011	011x1
100	10010
101	10111
110	1100x
111	111xx

Et :

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & Y & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

En bref...

Pour la ligne 7 du tableau on obtient :

- 0 + 0 = x, donc x = 0
- 1 + Y = 0, donc Y = 1

Pour la ligne 8 :

- pour le 1er x : 1 + 1 + 0 = x, donc x = 0 (modulo 2)
- pour le 2ème x : 0 + 0 + 1 = x, donc x = 1

On revient sur la ligne 4, qui est la dernière où il reste un x :

- 1 + 0 = x, donc x = 1

Ce qui donne :

Mots sources	Mots codés
000	00000
001	00101
010	01010
011	01111
100	10010
101	10111
110	11000
111	11101

Et :

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Remarque : Expliquer et détailler des calculs matriciels est assez long et fastidieux, mais comme aucune justification n'était demandée, on pouvait faire ces calculs de tête en moins de 5 minutes lors de l'examen.

Correction

Mots sources	Mots codés
000	00000
001	00101
010	01010
011	01111
100	10010
101	10111
110	11000
111	11101

Question 2 :

Matrice déjà calculée, cf question 1.

Correction

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Question 3 :

De 2 choses l'une :

- Soit il y a une erreur dans l'énoncé, et il faut lire "Les 3 premières colonnes de la matrices correspondent-elles à la matrice identité ?", et non pas "lignes".
  Dans ce cas, on a déjà la bonne matrice, les 3 premières colonnes correspondent à l'identité (une matrice carré avec des 1 sur la diagonale nord-ouest/sud-est, et des 0 partout ailleurs).
- Soit l'enseignant préfère une autre notation, strictement équivalente, qui consiste à écrire les mots en colonnes, et à multiplier la matrice génératrice par un mot pour le coder (ce qui n'est pas équivalent à multiplier un mot par la matrice génératrice comme on le faisait jusqu'ici, la multiplication matricielle n'étant pas commutative).

Dans le 2nd cas, il faudrait donc transposer la matrice qu'on a trouvé, ce qui donnerait :

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

On a bien les 3 premières lignes qui correspondent à l'identité.

Comme je préfère la notation qui consiste à écrire les mots en ligne, je vais supposer qu'il y a une erreur dans l'énoncé, et poursuivre avec la 1ère matrice génératrice.
Ceux qui préfèrent la notation des mots en colonnes peuvent transposer les matrices des questions précédentes et suivantes. Les raisonnements et les réponses rédigées restent les mêmes.

Question 4 :

Correction

- Les 3 premières lettres du mot codé sont une copie du mot source (puisqu'elles correspondent au mot source multiplié par l'identité).
- Les 2 dernières lettres forment le checksum spécifique du mot codé. Elle sont sensées permettre de détecter une éventuelle erreur.

Question 5 :

Méthode

Pour construire la matrice de contrôle, on peut prendre le lignes qui sont à gauche de l'identité (I_3) dans la matrice génératrice, et rajouter l'identité (I_2) en dessous.

Correction

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Question 7 :

5.1) La distance minimale du code

Méthode

On cherche à calculer la distance minimale entre 2 mots du code, c'est-à-dire le nombre minimum de bits qui varient entre 2 mots codés différents.
Et en regardant la matrice, on peut facilement déterminer cette distance.

Correction

La distance entre 2 mots ne peut évidemment pas être égale à 0 (il s'agirait alors du même mots, et pas de 2 mots différents).
On voit dans la matrice que la distance entre 2 mots ne peut pas être égale à 1, car si on change 1 bit du mot à coder, on ajoute ou soustrait une ligne de la matrice, ce qui modifie 2 bits sur le mot à coder, car toutes les lignes de la matrice génératrice ont 2 bits à 1.
On voit aussi facilement dans la matrice qu'une distance de 2 entre 2 mots est possible, car, par exemple, les 2 premières lignes de la matrice sont identiques à 2 bits près, donc ces 2 mots (10010 et 01010) sont des mots du codes (puisqu'ils sont les codes respectifs de 100 et 010) et ont pour distance 2.

Donc, la distance minimale entre 2 mots du code est de 2.

Autre méthode

Ici, comme le code est petit, et qu'on a déjà listé les 8 mots qu'il contient, on pouvait aussi simplement exhiber 2 mots à distance 2 dans la colonne des mots codés du tableau, et affirmer qu'on n'y trouve aucun couple de mots à distance 1.

5.2) Le nombre d'erreurs corrigibles

Correction

Non, le code ne peut pas toujours corriger 1 erreur, car la distance minimale du code est inférieure à 3.
Par exemple, comme 10010 et 01010 appartiennent tout 2 au code et sont à distance 2, si le destinataire reçoit 00010, il ne pourra pas déterminer s'il s'agit de 10010 avec une erreur sur le 1er bit, ou de 01010 avec une erreur sur le 2ème bit.

Autre méthode

8 mots sources possibles sur 3 bits, et à chacun de ses 8 mots on associe un code de 5 bits, donc on a 8 mots codés possible.
Pour chacun de ses 8 mots il existe 5 mots possibles avec 1 erreur (l'erreur pouvant survenir sur chacun des 5 bits), 8 * 5 = 40, il y a donc 40 possibilités d'avoir 1 erreur.
En comptant les 8 mots sans erreurs, il y a 48 possibilités d'avoir 0 ou 1 erreur.
Or, on ne peut recevoir que 32 mots différents sur 5 bits.
Il y aura donc au moins 16 mots *ambigus* (terme technique que je viens d'inventer), c'est-à-dire dont on ne pourra retrouver l'origine.
Donc, non, le code ne peut pas toujours corriger 1 erreur.

Encore une autre méthode

Non, le code ne peut pas toujours corriger une erreur, car les lignes de la matrice de contrôle (ou *C*) ne sont pas toutes distinctes, or c'est en regardant à quelle ligne de *C* correspond le *syndrome* du mot (celui là je l'invente pas, il existe, j'vous jure !), c'est-à-dire le mot reçu multiplié par *C*, qu'on déduit sur quel bit se trouve l'éventuelle erreur.
Par exemple, si il y a une erreur sur le 1er bit, le destinataire trouvera le syndrome 01 (qui correspond à la 1ère ligne de la matrice *C*), mais ce syndrome ne lui permettra pas de déterminer si l'erreur se trouve sur le bit 1, 2 ou 4 (puisque les lignes 2 et 4 de *C* sont aussi 01).

Question 7 :

cf paragraphe précédent (normalement, on multiplie le mot codé par la matrice de contrôle *C*, ça nous donne le syndrome, on regarde à quelle ligne de *C* il correspond, et le numéro de cette ligne nous donne le numéro du bit à corriger, mais ici, ça ne marchera pas, car plusieurs lignes de *C* sont identiques)

(Suite non traitée)