LELEC1930 - Introduction aux télécommunications
Séance 3 - Encodage

Prof. : Jérôme Louveaux   Assist. : Jérome Eertmans

Rappel

Attention : *dans les exercices suivants, les opérations sont effectuées modulo 2. Le résultat est soit 0, soit 1. On aura donc que* $1+1=0$, $1+1+1=1$, $1+1+1+1=0$, *etc.*

Code correcteur

Lors de la transmission d'un message, il n'est pas rare d'observer une déformation du message lors de son chemin. De ce fait, l'information reçue peut ne contenir que de manière partielle le message original.

En anticipation à ce problème, le message est alors envoyé sous une forme encodée, souvent plus longue, afin d'augmenter le taux de réussite quant à la bonne réception du message. De manière générale, on utilise **un dictionnaire** permettant d'encoder un message de k bits sur n bits. On définit le **taux du code** comme étant alors $\frac{k}{n}$.

$$\begin{array}{c} 00 \rightarrow \\ 01 \rightarrow \\ 10 \rightarrow \\ 11 \rightarrow \end{array} \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

FIGURE 1 – Dictionnaire d'encodage d'un message de 2 bits sur 6 bits.

La distance minimale d'un code, notée d_{min}, est la plus petite distance entre deux mots du dictionnaire. La distance entre deux mots, aussi appelée distance de Hamming, est le nombre de bits différents entre ces deux mots. Dans l'exemple de la figure 1, la distance minimale est de 3 bits.

Un code dont la distance minimale vaut d_{min} détecte $d_{min} - 1$ erreurs et peut corriger $\lfloor \frac{d_{min}-1}{2} \rfloor$ erreurs, où $\lfloor \cdot \rfloor$ est l'arrondi vers le bas. Ceci définit la **capacité correctrice** d'un code.

Le décodage d'un bloc reçu se fait en recherchant le mot le plus proche de ce dernier en termes de distance. Exemple pour la figure 1 :

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \longrightarrow 00 \tag{1}$$

Un **code de Hamming** est un code correcteur linéaire qui permet la détection d'une erreur si elle n'est présente que sur une lettre du message. Les codes linéaires peuvent être complètement représenté par une **matrice génératrice** $\underline{\underline{G}}$ de taille $k \times n$. Cette matrice s'obtient en assemblant les mots de codes correspondant à la base canonique [1]. À partir de n'importe quel mot d'information $\underline{i}$, on peut calculer son mot de code, $\underline{c}$, en effectuant le produit matriciel $\underline{c} = \underline{i}\,\underline{\underline{G}}$.

1. 1000, 0100, 0010 et 0001 dans le cas d'un mot d'information de 4 bits.

On définit alors la **matrice de contrôle** H. Cette dernière est plus complexe à calculer dans un cas général. Néanmoins, lorsque la matrice $\underline{\underline{G}}$ a une structure particulière formée d'une sous-matrice identité (ici représentée en noir sur la figure 3) accolée à une autre sous-matrice (ici représentée en bleu), dans ce cas, la matrice de parité peut s'obtenir en transposant la partie bleue de la matrice $\underline{\underline{G}}$ et en y ajoutant une sous-matrice identité à droite pour compléter une matrice de taille $(n-k) \times n$. C'est le cas notamment pour la matrice du code de Hamming (voir ci-dessous).

Pour un bloc reçu $\underline{c}$, le vecteur syndrome est donné par $\underline{s} = \underline{c}\,\underline{\underline{H}}^T$. Il indique la présence ou non d'erreur(s) : le vecteur nul indique l'absence d'erreurs.

$$\begin{array}{l} 0000 \to \\ 0001 \to \\ 0010 \to \\ 0011 \to \\ 0100 \to \\ 0101 \to \\ 0110 \to \\ 0111 \to \end{array} \begin{bmatrix} 0&0&0&0&0&0&0 \\ 0&0&0&1&1&1&0 \\ 0&0&1&0&0&1&1 \\ 0&0&1&1&1&0&1 \\ 0&1&0&0&1&0&1 \\ 0&1&0&1&0&1&1 \\ 0&1&1&0&1&1&0 \\ 0&1&1&1&0&0&0 \end{bmatrix} \quad \& \quad \begin{array}{l} 1000 \to \\ 1001 \to \\ 1010 \to \\ 1011 \to \\ 1100 \to \\ 1101 \to \\ 1110 \to \\ 1111 \to \end{array} \begin{bmatrix} 1&0&0&0&1&1&1 \\ 1&0&0&1&0&0&1 \\ 1&0&1&0&1&0&0 \\ 1&0&1&1&0&1&0 \\ 1&1&0&0&0&1&0 \\ 1&1&0&1&1&0&0 \\ 1&1&1&0&0&0&1 \\ 1&1&1&1&1&1&1 \end{bmatrix}$$

FIGURE 2 – Dictionnaire du code de Hamming (7,4).

$$\underline{\underline{G}} = \begin{bmatrix} 1&0&0&0&1&1&1 \\ 0&1&0&0&1&0&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&1&1&1&0 \end{bmatrix} \qquad \underline{\underline{H}} = \begin{bmatrix} 1&1&0&1&1&0&0 \\ 1&0&1&1&0&1&0 \\ 1&1&1&0&0&0&1 \end{bmatrix}$$

FIGURE 3 – Matrice génératrice et matrice de contrôle du code de Hamming (7,4).

Compression des données

Souvent, on souhaite réduire la taille des messages envoyés afin qu'ils soient transmis plus rapidement ou prennent moins de place sur un espace de stockage. Pour ce faire, les données sont compressées en utilisant un algorithme de codage. C'est le cas des fichiers `.zip` ou encore du codage de Huffman. Ce dernier est décrit dans les slides du cours et un exemple détaillé est disponible sur la page Wikipédia.

Le code de Huffman se base sur une distribution connue des symboles afin de créer un code de compression. Le code de compression se présente sous forme d'un arbre où prendre une branche à gauche indique un 0 et une branche à droite un 1. Ci-dessous, l'arbre obtenu pour la phrase "this is an example of a huffman tree", pour lequel le code final est : {'e' : 000, 'a' : 010, ' ' : 111, ...}.

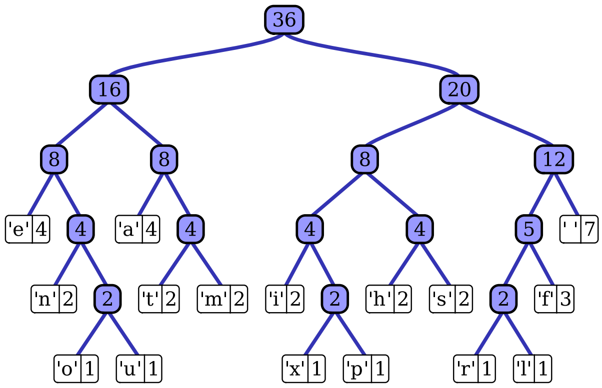


FIGURE 4 – Arbre d'Huffman, repris de Wikipédia.

Exercices

Exercice 1 : Dictionnaire d'un code

À partir du code suivant :

$$\begin{array}{c} 00 \to \\ 01 \to \\ 10 \to \\ 11 \to \end{array} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

1. Calculez le taux du code.
2. Effectuez le décodage du bloc reçu suivant : $y = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$.
3. Quelle est la capacité correctrice du code ?
4. Il s'agit d'un code linéaire. Quelle est la taille de sa matrice de contrôle ?

Réponse à l'exercice 1 :

Le code présenté encode 2 bits (k) sur 7 bits (n).

1. Le taux est donc de $\frac{2}{7}$.
2. Le code le plus proche de y est $\begin{bmatrix} 1 & 0 & {\color{red}0} & 1 & 1 & {\color{red}0} & 1 \end{bmatrix}$, ce qui correspond au mot 01.
3. On calcule que la distance minimale est de 4. La capacité correctrice est donc de 1. C'est le nombre minimum d'erreurs que l'on peut corriger : pas maximum. Sinon, on n'aurait pas pu corriger y.

Exercice 2 : Code de Hamming

Considérez le code de Hamming (7,4) et répondez aux questions suivantes :

1. Calculez le mot codé pour les bits d'entrée 1101.
2. Quel est le taux de ce code ?
3. Quelle est la distance minimale du code ?
4. Calculez le syndrome du code 1111110.
5. Quel est le mot à l'origine du code 1001101 ?
6. Quel est le mot à l'origine du code 0000110 ?

Réponse à l'exercice 2 :

Pour répondre aux questions, il est utile de relire le rappel.

1. 1101 devient 1101100.

2. Le taux est de $\frac{4}{7}$.
3. La distance minimale est de 3.
4. Le syndrome de $\underline{c} = 1111110$ est $\underline{s} = \underline{c} \times \underline{\underline{H}}^T = 001$.
5. Le mot le plus proche est 1001.
6. Le mot le plus proche est 0001.

Exercice 3 : Code de Huffman

1. Parmi les codes suivants, lesquels **ne sont pas** des codes de Huffman ?
   (a) $\{1, 01, 00\}$
   (b) $\{01, 11, 10\}$
   (c) $\{00, 01, 10, 110\}$
   (d) $\{0, 1, 01, 11\}$
2. On considère une source d'information, ayant 5 valeurs possibles, avec les probabilités données ci-dessous. On considère aussi deux encodages possibles C_1 et C_2 :

x	$p(x)$	C_1	C_2
a	0.50	0	01
b	0.15	10	00
c	0.15	110	10
d	0.10	1110	110
e	0.10	1111	111

   Quel code fournit la meilleure compression d'information ?
3. Construisez l'arbre de Huffman pour la phrase "Ésope reste ici et se repose".

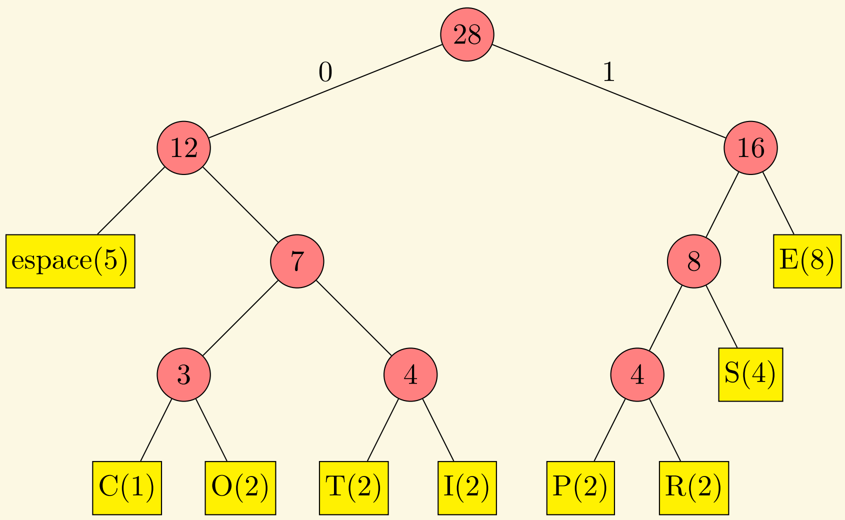
Réponse à l'exercice 3 :

1. Pour vérifier si un code est valide, il faut tracer un arbre et vérifier que chaque feuille possède une paire. Par exemple, si le code 01 existe, alors le code 00 ou ses deux feuilles doivent exister. Une feuille (un code) ne peut pas posséder de sous-feuille.
   (a) Code de Huffman.
   (b) Non valide, car 01 n'a pas sa paire.
   (c) Non valide, car 110 n'a pas sa paire.
   (d) Non valide, car 0 et 1 sont des feuilles et ne peuvent donc pas avoir de sous-feuille.
2. Pour estimer la compression de chaque code, on peut calculer le nombre de bits moyen et comparer :

$$\overline{C}_1 = 0.50 \cdot 1 + 0.15 \cdot 2 + 0.15 \cdot 3 + 0.10 \cdot 4 + 0.10 \cdot 4 = 2.05$$
$$\overline{C}_2 = 0.50 \cdot 2 + 0.15 \cdot 2 + 0.15 \cdot 2 + 0.10 \cdot 3 + 0.10 \cdot 3 = 2.20$$

   Le premier code est donc légèrement meilleur.

3. Voici un exemple d'arbre que vous pouvez obtenir. Des variations sont possibles, car plusieurs lettres apparaissent le même nombre de fois.



Exercice 4 (bonus) : Planning de fréquence

On veut établir le planning de fréquence d'un réseau cellulaire. En raison de différents effets de propagation, on suppose que la densité de puissance reçue à une distance r ($r > 10\,\mathrm{m}$) d'une station de base est approximée par

$$S(r) = \frac{P_0}{r^3}, \tag{2}$$

où P_0 est la puissance de l'antenne (supposée identique pour toutes les stations de base). Pour simplifier, on suppose que toutes les cellules sont des cercles de $1\,\mathrm{km}$ de rayon centrés sur leur station de base respective. Quelle doit être la distance minimale entre 2 stations de base réutilisant la même fréquence, de façon à ce que signal utile en provenance de la première station de base soit toujours au minimum $20\,\mathrm{dB}$ au-dessus de l'interférence en provenance de la deuxième station de base ?

Réponse à l'exercice 4 (bonus) :

Afin de répondre à l'exercice, il est utile de faire un petit dessin, voir la figure 5. L'utilisateur (en rouge), souhaite donc recevoir un signal de la part de TX_1 au minimum $20\,\mathrm{dB}$ au-dessus de TX_2.
Le pire cas de figure est celui illustré : l'utilisateur est en bordure de zone ($d_1 = 1\,\mathrm{km}$) et se trouve au plus proche de TX_2.

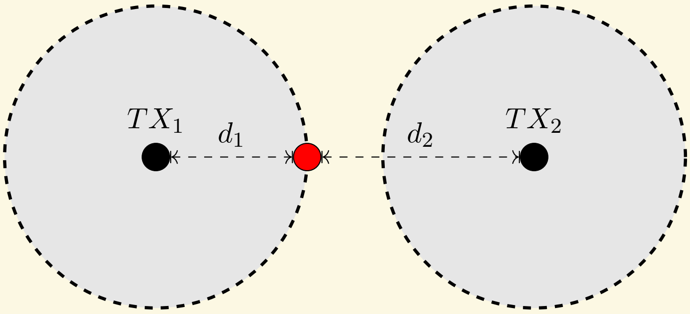


FIGURE 5 – Schéma du problème.

La distance $d_1 = 1$ km. On veut aussi que le rapport des puissances soit de 20 dB, c.-à-d. un rapport de 100 :

$$\frac{S_1}{S_2} = 100 = \frac{d_2^3}{d_1^3} \Rightarrow d_2 = 4.64\,\text{km}. \tag{3}$$

La distance entre les deux antennes doit donc être de 5.64 km.