Université de Strasbourg

Codes correcteurs et détecteurs

Projet noté de réseaux locaux et inter-connexions

Louis-César Pagès

I. Introduction

Présentation du projet

A l'aide d'un programme en C, on veut illustrer un phénomène récurrent qui advient lors de communications dans un réseau local. En effet, il se peut que les données envoyées par un émetteur soient « brouillées » durant leur transmission. Le code reçu par un destinataire est alors corrompus. Cela ce caractérise le plus souvent par des bits intervertis dans le message original.

Ici, le medium « medium.c » peut être configuré pour simuler plusieurs scénario d'erreurs lors de la transmission. Ces types sont détaillés dans le sujet de ce projet.

Dans la suite, on tente d'appliquer des solutions vues en cours pour détecter et résoudre ces problèmes de transmissions. Ces solutions sont implémenter dans le fichier .

Codes détecteurs

Bits de parité, code polynomial vont nous permettre de détecter des erreurs.
Le bits de parité consiste à calculer la parité du code (nombre de 1). Si cette parité existe alors le bit de contrôle aura comme valeur 1 sinon 0.

Le code polynomial consiste en la « multiplication » du message que l'on veut coder par un polynôme générateur. Si lors de du déchiffrage le reste de la division du message codé n'est pas nul, alors il y a eu une erreur.

Code correcteur

Le code Hamming est un code linéaire qui peut détecter jusqu'à deux erreurs de bit simultanées Il nous permet aussi de corriger les erreurs d'un seul bit.

II. Exercice 1

Avec le bit de parité, on peut détecter les erreurs survenu dans les bits d'informations et si. On peut détecter toutes les erreurs de types 1 qui surviennent dans les bits d'informations. Si, lors de la transmission le bits de parité (9ième bits) est modifié, il y a un risque pour que le récepteur détecte une erreur alors que les bits d'informations sont corrects.

Si le type d'erreur modifie les informations par pair (2n 1 ou 2n 0 modifiés) dans les bits d'informations, alors l'erreur ne peut être détectée.

III. Exercice 2

Avec le codage polynomial, toutes les erreurs survenus lors de la transmission peuvent être détectées à condition qu'elles ne produisent pas un multiple de G(x), le polynôme générateur.

IV. Exercice 3

1. La valeur k-ième d'une combinaison linéaire quelconque des k-1 de G est 1.
2. En construisant la matrice G', le poids minimum devient p-1.
3. Après la construction du code raccourci de Hamming(7,4), on obtient :

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \longrightarrow G' = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \longrightarrow H' = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Ici, le nouveau code raccourci C' a donc un poids minimum de 2. (Poids des mots non nuls de poid minimum = 2). Pour trouver sa capacité de détection, il faut donc déterminer si il existe un nb k d'erreurs tel que :

$$d \geq 2k + 1$$

La distance d'un code linéaire étant son poids minimum, 2, il n'existe pas d'entier k permettant de résoudre l'équation.

Le code ne peut donc pas corriger d'erreurs.

4. Si l'on répète cette opération 3 fois sur le code de Hamming(7,4) on obtient un code qui correspond à un polynôme générateur pour un codage polynomial.
   Son poids minimum est de 1, il ne peut donc rien corriger et peut détecter plusieurs erreurs.
5. Oui il y a plusieurs matrice G possibles pour la question 3. Oui on peut obtenir des réponses différentes, tout dépend de la place des vecteurs dans la matrice initial G. Au final le polynôme générateur trouvé aura une une forme différente avec des monômes nuls ou non. Ce qui implique, parfois une capacité de détection différente.
6. Il existe un code de Hamming(n,k) pour toute valeur de n tel que : $n=2^{n-k}-1$
7. Pour s'assurer que le code corrige 1 erreur, il faut vérifier que son poids minimum soit supérieur ou égale à 3. Pour une dimensions k = 8 ; une solution serait de trouver une matrice Hamming() de dimensions supérieur à 8 puis de « raccourcir »le code (Comme dans la question 3). On sait qu'un code de Hamming à pour longueur 2^x-1 et de dimension 2^x-k-1 . Hamming(15,11). Après raccourci, on obtient une longueur de 12 pour protéger 8 bits d'information.
8.

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