Codes Correcteurs

Julie Badets, Corentin Frade, Quentin Rouland, Émeric Tosi

27 février 2014

Résumé

Introduction

Sommaire

Chapitre 1

Code de répétition

1.1 Introduction

On transmet simplement plusieurs fois le même message.

Une seule erreur peut être détectée à coup sur.

1.2 Probabilité de détection

Probabilité de détecter une erreur :

1.3 Rendement

Le rendement de ce code est très mauvais, on double au minimum la taille du message :

$$Rendement = \text{Taille du message} * \text{Nombre de répétition(s)}$$

Pour un message sur 7 bits (un simple caractère encodé en UTF-7 par exemple) et avec 1 répétition seulement :

$$Rendement = \frac{7}{7 * 2} = 50\%$$

Pour le même message mais avec 2 répétitions :

$$Rendement = \frac{7}{7 * 3} \approx 33.3\%$$

1.4 Implémentation

Chapitre 2

VRC : Bit de parité

2.1 Introduction

le VRC (Vertical Redundancy Check), plus connu sous le nom de bit de parité, est simplement le rajout d'un bit en fin de message pour assurer la parité du message. Ce dernier bit la valeur nécessaire pour assurer un nombre pair de bit à 1 dans le message final. Il est donc à 0 pour un nombre pair de bit à 1 dans le message de départ, ou est à 1.

Une seule erreur peut être détectée à coup sur. Cependant les erreurs détectées sont celles où un nombre impairs de bit ont changé d'état.

2.2 Probabilité de détection

Probabilité de détecter une erreur :

$$P(Détection) = P(1erreur) + P(3erreurs) + P(5erreurs) + P(7erreurs)$$

$$P(Détection) = \binom{n}{k} p^k (1-p)^{n-k} + \binom{n}{k} + \binom{n}{k} + \binom{n}{k}$$

$$P(Détection) \approx 74\% \text{ avec } P(Erreur) = 10\%$$

2.3 Rendement

Le rendement de ce code est très bon :

$$Rendement = \frac{\text{Taille du message}}{\text{Taille du message} + 1}$$

Pour un message sur 7 bits (un simple caractère encodé en UTF-7 par exemple) le rendement est déjà excellent :

$$Rendement = \frac{7}{7+1} = 87.5\%$$

2.4 Implémentation

Chapitre 3

LRC :Contrôle parité croisée

3.1 Introduction

À partir de plusieurs message encodés grâce au VRC. Nous prendrons le cas d'une matrice carrée composée de 7 messages de taille 7 chacun. Il sera appliqué horizontalement à la matrice (à chaque message) le bit de parité. La matrice passera donc à une taille de 7 sur 8. Le dernier message qui sera généré grâce à l'application du bit de parité verticalement sur la matrice.

cb erreur peut être corrigée. ?

3.2 Probabilité de détection

Probabilité de détecter une erreur :

3.3 Rendement

Le rendement de ce code est bon :

$$Rendement = \frac{n * \text{Taille du message}}{(n+1) * (\text{Taille du message} + 1)}$$

Dans le cas que nous étudions :

$$Rendement = \frac{7 * 7}{(7+1) * (7+1)} \approx 76.6\%$$

3.4 Implémentation

Chapitre 4

CRC : Code de redondance cyclique

4.1 Introduction

le CRC (Cyclic Redundancy Check), contrôle de redondance cyclique, représente la principale méthode de détection d'erreurs utilisée dans les télécommunications et consiste à protéger des blocs de données, appelés trames. À chaque trame est associé un bloc de données, appelé code de contrôle (parfois CRC par abus de langage).

On choisit un polynôme générateur, fixé et donc connu des deux entités qui se transmettent le message. Grâce à celui ci, l'émetteur peut générer le code de contrôle qui est le reste de la division avec le message à envoyer. Le récepteur divise ce qu'il a reçut, retrouve le message et sait si il y a eu un problème.

Il existe plusieurs variantes du CRC selon le choix du polynôme : CRC 12, CRC 16, CRC CCIT v41, CRC 32, CRC ARPA

Deux erreurs peuvent être détectées grâce au CRC 16. Cependant les erreurs détectées sont celles où un nombre impairs de bit ont changé d'état ou celles qui sont des suites de bit qui ont tous changés (rafales), de taille inférieur au degré du polynôme.

4.2 Probabilité de détection

Probabilité de détecter une erreur :

$$P(Détection) = P(1erreur) + P(3erreurs) + P(5erreurs) + P(7erreurs)$$

$$+P(9erreurs) + P(11erreurs) + P(13erreurs) + P(15erreurs) + ...XXX...$$

$$P(Détection) = XXX\% \text{ avec } P(Erreur) = 10\%$$

4.3 Rendement

Le rendement de ce code est dépendant de la taille du message :

$$Rendement = \frac{\text{Taille du message}}{\text{Taille du message} + \text{Degré du polynôme}}$$

Pour un petit message de 7 bit (un simple caractère encodé en UTF-7 par exemple) avec du CRC 16 cela donne un rendement très médiocre :

$$Rendement = \frac{7}{7+16} \approx 30\%$$

Cependant avec un message de taille plus importante comme par exemple un message de 128 bit avec du CRC 16 le rendement est excellent :

$$Rendement = \frac{128}{128+16} \approx 89\%$$

Chapitre 5

Hamming

Table des figures