Présentation des codes redondants cycliques (CRC) & Illustration et limites du Code Hamming

Présenté par :

AHOUANDJINOU Bill & ALONOUMI Cédric Bosco

4^{e} année d'Ingénieurie

Ecole Polytechnique d'Abomey-Calavi – Université d'Abomey-Calavi

EPAC – UAC

Année académique : 2021 – 2022

Sommaire

Section introductrice

Généralités

- Des systèmes de plus en plus performants ont été mis au point pour perfectionner la détection d'erreurs de transmission.
- Ces codes sont généralement appelés codes **autocorrecteurs** ou **autovérificateurs**.
- Techniques simples de détection d'erreurs : le **contrôle de parité**, le **contrôle de parité croisé**.
- La plupart des systèmes de contrôle d'erreur au niveau logique sont basés sur un ajout d'information
- Les méthodes standard de calcul d'une somme de contrôle est appelée **Contrôle de Redondance Cyclique** (CRC)

Contrôle de Redondance Cyclique

Principe

- Principe : traiter les séquences binaires comme des polynômes binaires
- La séquence binaire 110101001 correspond à la forme polynomiale suivante : $X^8 + X^7 + X^5 + X^3 + 1$.
- Une séquence de n bits constitue donc un polynôme de degré maximal $n - 1$.
- Un polynôme **générateur** prédéfini et noté $G(X)$ est connu de l'émetteur et du récepteur
- La détection d'erreur consiste pour l'émetteur à effectuer un algorithme sur les bits de la trame afin de générer un CRC, et de transmettre ces deux éléments au récepteur. Il suffit alors au récepteur d'effectuer le même calcul afin de vérifier que le CRC est valide.

Procédure de calcul

- Soit M le message correspondant aux bits de la trame à envoyer et $M(X)$ le polynôme associé.
- Appelons M' le message transmis, c'est-à-dire le message initial auquel aura été concaténé le CRC de n bits.
- Le CRC est tel que $M'(X)/G(X) = 0$.
- La mise en garde importante est d'utiliser l'opérateur XOR à la place de l'addition (soustraction).

Application

- Soit M le message correspondant aux bits de la trame à envoyer et $M(X)$ le polynôme associé.
- Soit M' le message transmis et $G(X)$, le polynôme générateur associé au générateur G.
- On prend M de 16 bits : 1011000100101010 et $G(X) = X^3 + 1$ (représenté en binaire par 1001).
- Etant donné que $G(X)$ est de degré 3, il s'agit d'ajouter 3 bits nuls à M, soit : 1011000100101010000.
- Le CRC étant égal au reste de la division de M par G, en prenant soin d'appliquer l'opérator logique XOR à la place de la soustraction, on retrouve $CRC = 001$.
- On a $M' = 1011000100101010001$.
- Si le récepteur du message effectue la division de M' par G, en respectant la même règle concernant l'opérateur XOR, il obtiendra un reste nul si la transmission s'est effectuée sans erreur.

Implémentation

- Code C++ au niveau de l'émetteur.
- Code C++ au niveau du récepteur.

Intégrité des données

- Lorsqu'ils sont stockés avec les données, les CRC et les fonctions de hachage cryptographique ne protègent pas en eux-mêmes contre la modification intentionnelle des données.
- Contrairement aux fonctions de hachage cryptographique, le CRC est une fonction facilement réversible, ce qui le rend impropre à une utilisation dans les signatures numériques.
- Le CRC est une fonction linéaire, en conséquence, même si le CRC est chiffré avec un chiffrement de flux qui utilise XOR comme opération de combinaison, à la fois le message et le CRC associé peut être manipulé sans connaître la clé de chiffrement.

Polynômes générateurs

Les polynômes générateurs les plus couramment employés sont :

- CRC-12 : $X^{12} + X^{11} + X3 + X2 + X + 1$
- CRC-16 : $X^{16} + X^{15} + X^2 + 1$
- CRC CCITT V41 : $X^{16} + X^{12} + X^5 + 1$
- CRC-32 (Ethernet) : $X^{32} + X^{26} + X^{23} + X22 + X16 + X12 + X11 + X10 + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- CRC ARPA : $X^{24}+X^{23}+X^{17}+X^{16}+X^{15}+X^{13}+X^{11}+X^{10}+X^9+X^8+X^5+X^3+1$

Le choix ou la conception d'un générateur dépend du concepteur et des applications liées.

Code Hamming

Présentation

- L'objectif d'un code correcteur est la détection et la correction d'erreurs après la transmission d'un message.
- Cette correction est permise grâce à l'ajout d'informations redondantes.
- Un **code de Hamming** : la redondance permet exactement la correction d'une altération sur une unique *lettre* du message.

Fonctionnement

Se référer au document plus explicite.

Application

Au niveau de l'émetteur :

- Pour $m = 4$, on a $k = 3$ et $n = 7$. $M = m_4 m_3 m_2 m_1$ et $K = k_3 k_2 k_1$
- On obtient $N = m_4 m_3 m_2 k_3 m_1 k_2 k_1$.
- $H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$
- En prenant $M = 1010$, on a : $N = \begin{pmatrix} k_1 \\ k_2 \\ 0 \\ k_3 \\ 1 \\ 0 \\ 1 \end{pmatrix}$ et

  $H \times N = K = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$
- Le message envoyé est : $N = 1010010$.

Application

Au niveau du récepteur :

- Il reçoit $Q = q_7 q_6 q_5 q_4 q_3 q_2 q_1$.
- $H \times Q = C$
- Si $N = Q = 1010010$ et C est la matrice nulle.
- Exemple de cas d'erreur : $Q = 1000010$.
- On obtient $C = 101$ (en binaire) $= 5$ (en décimal)
- Correction de cette erreur.

Implémentation

- Code python
- Simulation avec Simulink

Limites

- La limite principale du code Hamming (n, m, k) repose sur sa capacité de correction.
- La capacité de correction : $\frac{k-1}{2}$ erreur(s) pour n bits de données.

Conclusion

Conclusion

Il est très probable de recevoir des messages erronnés après une transmission. Nous pouvons, de ce fait, prendre des précautions de traitement avant l'envoi pour essayer de détecter ces éventuelles erreurs en utilisant des codes détecteurs (code CRC) et/ou correcteurs (code Hamming) d'erreurs. Malgré l'efficacité de ces codes, ils ont des limites qui restreignent donc leur usage.

Bibliographie

- https ://www.commentcamarche.net/contents/97-controle-d-erreur-crc
- https ://waytolearnx.com/2019/06/techniques-de-detection-derreur.html
- https ://www.techno-science.net/glossaire-definition/Code-de-Hamming-7-4.html
- https ://www.mathworks.com/help/comm/ug/error-detection-and-correction.html

Merci de votre attention !