

Modèle OSI

Couche 2 : Détection des erreurs

(Suite Partie II)

Christian Attiogbé
Christian.Attiogbe@univ-nantes.fr
(novembre 1999, janvier 2001)



Détection des erreurs de transmission

Des erreurs sont dues aux canaux de transmission.

Il faut **pouvoir les détecter et les corriger**.

De façon générale pour transmettre k bits, on **ajoute r bits**,
soit au total $k + r = n$ bits transmis.

On parle de **code(n, k)** ou de **mot de code**.

Il existe **plusieurs méthodes de détection et de correction des erreurs**.

Elles sont situées dans la **couche 2 du modèle de référence**



Méthode VRC : Vertical Redundancy Check

C'est la méthode de la parité verticale.

A chaque caractère, on ajoute un bit de parité (1 ou 0) de façon à ce que le nombre total de 1 soit paire ou impaire.

parité paire lorsque le nombre de 1 est paire et de

parité impaire lorsque le nombre de 1 est impaire.

Exemple : VRC avec parité paire

Caractère A (code 41 en ASCII) : 1 0 0 0 0 0 1 **0**

Caractère C (code 43 en ASCII) : 1 0 0 0 0 1 1 **1**



Méthode VRC

La détection d'erreur avec le VRC consiste à :

recalculer le bit de parité à la réception et

vérifier que le nombre total de 1 correspond à la parité choisie.

La méthode **VRC n'est pas très fiable**

(si deux bits sont erronés, la détection échoue).

Son taux d'efficacité est estimé à 50 %.



Méthode LRC : Longitudinal Redundancy Check

Parité longitudinale.

Elle permet de tester l'ensemble d'un bloc de données.

Un **OU logique** est effectué sur l'**ensemble des bits de même rang** du message.

Le résultat obtenu est appelé **BCC (Block Check Character)** ou **clé** de fin de message, il est transmis comme dernier caractère du message à émettre.



Méthode LRC

Le récepteur fait la même opération et compare les deux clés. Cette méthode atteint un taux d'efficacité de 98%.

Exemple : LRC avec parité paire (sur 7 bits)

| | |
|--------------------------------|---------------|
| Caractère A (code 41 en ASCII) | 1 0 0 0 0 0 1 |
| Caractère C (code 43 en ASCII) | 1 0 0 0 0 1 1 |
| <b style="color: red;">LRC | 0 0 0 0 0 1 0 |

(Sur 8 bits, il faut d'abord calculer le VRC)



Méthode CRC : Cyclic Redundancy Check

C'est la méthode 'Code de Redondance cyclique'. C'est la plus élaborée.

Le principe consiste à l'**émission** à :

- diviser les bits du message à émettre considéré comme un polynôme par un autre polynôme (dit générateur).

Le reste de la division constitue le CRC.

- emettre le message avec le CRC

A la **réception** on recalcule un CRC et les résultats comparés. Cette méthode atteint un taux d'efficacité de 99,9%



Les codes cycliques

On les appelle encore **codes polynomiaux**.

Ils assurent un bon contrôle des erreurs et sont de ce fait retenus par l'ISO.

L'information à envoyer est enregistrée dans une **trame** (bloc des données au niveau 2 du modèle OSI).

Une trame est formée de m bits de données et de r bits de contrôle.

Le **mot de code** est formé des $m + r$ bits.

Les mots de code sont représentés **sous forme polynomiale**.



Les codes cycliques

Exemple :

La suite de bits **1 0 1 1 0 0 1 1** est représentée par :
le polynôme $x^7 + x^5 + x^4 + x^1 + x^0$

La suite **1 1 0 1 0** est représentée par
le polynôme $x^4 + x^3 + x$.



Code cyclique : principe de la détection

On utilise $R(x)$ le reste de la division $M(x)/G(x) = Q(x)$
 $G(x)$ appelé **polynôme générateur**, $Q(x)$ le quotient

Soit M l'ensemble des bits d'informations à transmettre.

$M = m_n \dots m_1 m_0$ où m_i est un bit.

On **associe à M le polynôme**

$$M(x) = m_n x^n + \dots + m_1 x^1 + m_0.$$

Soit $G(x)$ le polynôme générateur. $G(x)$ est de degré γ .

On peut exprimer $M(x)$ par $M(x) = G(x).Q(x) + R(x)$



Code cyclique : Principe de la détection ...

$R(x)$, le reste de la division polynomiale de $x^\gamma.M(x)$ par $G(x)$, représente le code d'erreur.

Le bloc transmis sera la suite de bits correspondants au polynôme : $M'(x) = x^\gamma.M(x) - R(x)$.



Code cyclique : Principe de la détection ...

A la réception,

$M'(x)$ est divisé par $G(x)$ (par le récepteur).

Le reste $R'(x)$ de la division doit être nul s'il n'y a pas eu d'erreur de transmission.

Toutes les opérations sont binaires.

Les polynômes générateurs sont normalisés.

Le plus utilisé est celui de la recommandation v41 de l'UIT :

$$G(x) = x^{16} + x^{12} + x^5 + 1$$



Code cyclique : Illustration

Soient M un bloc de m bits et $M(x)$ le polynôme correspondant, $G(x)$ un polynôme de degré r , le polynôme générateur (avec $m \geq r$).

On multiplie $M(x)$ par x^r , en ajoutant r bits 0 à la fin de M .

On divise $x^r.M(x)$ par $G(x)$ pour obtenir le reste $R(x)$.

$$x^r.M(x) = G(x).Q(x) + R(x) \text{ et}$$

$$R(x) = x^r.M(x) - G(x).Q(x)$$

$$\text{On calcule } M'(x) = x^r.M(x) - R(x)$$



Code cyclique : Illustration

La séquence de bits obtenus M' est expédiée.

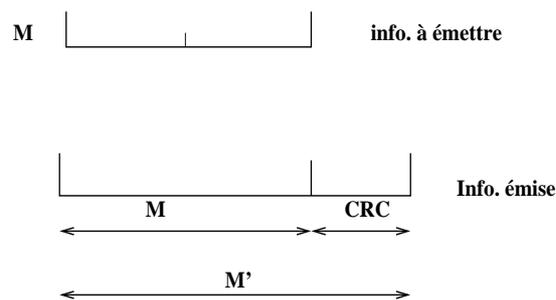


Fig. 1 – Idée du calcul du mot de code



Détection Illustration

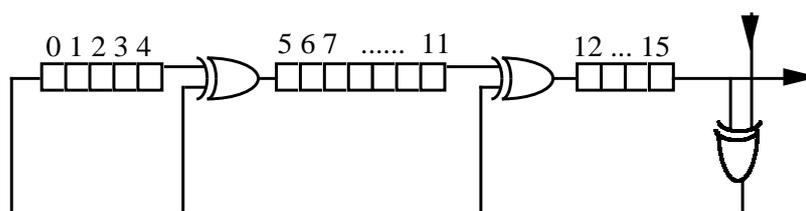


Fig. 2 – Circuit V41 du calcul de CRC

- Initialement, le registre est à 0 (tous les bits)
- Les bits du message M sont en entrée (DI)



Détection Illustration

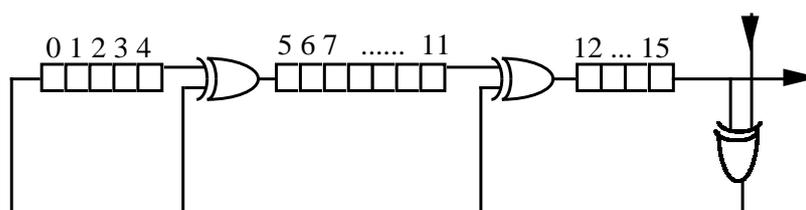


Fig. 3 – Circuit V41 du calcul de CRC

- bit par bit, on fait entrer M, on obtient une nouvelle configuration du registre.
- à la fin de tous les bits (M), le contenu du registre est le CRC (mais à l'envers).



CRC - V41

Le CRC V41 détecte

- toutes les erreurs simples
- toutes les erreurs doubles
- toutes les erreurs impaires ($x+1$)
- 100 % des erreurs de longueur ≤ 16 bits
- 99,998 % des erreurs de longueur > 16 bits



Les codes courants

| | | |
|---------|---|---------------------------------|
| crc_12 | $x^{12} + x^{11} + x^3 + x^2 + 1$ | car. de 6 bits |
| crc_16 | $x^{16} + x^{15} + x^2 + 1$ | car. 8 bits, USA |
| crc_32 | $x^{32} + x^{26} + x^{23} + x^{22} + x^{16}$ $+ x^{12} + x^{11} + x^{10} + x^8 + x^7$ $+ x^5 + x^4 + x^2 + 1$ | liaisons synch. hauts débits |
| crc_V41 | $x^{16} + x^{12} + x^5 + 1$ | car. 8 bits, Europe |



Détection d'erreur - correction

Dans tous les cas de détection d'erreur,

lorsqu'une erreur est détectée, il faut la corriger.

L'émetteur et le récepteur amorce un dialogue (à l'initiative du récepteur) pour effectuer la correction de l'erreur.



Détection d'erreur - correction

Les méthodes de correction d'erreur sont :

- **acquiescement systématique de chaque bloc de données,**
- **acquiescement d'une suite de plusieurs blocs** avec **reprise séquentielle,**
- **acquiescement d'une suite de plusieurs blocs** avec **reprise sélective.**

Nous étudierons ces méthodes dans les procédures/protocoles de la couche 2 (la famille HDLC).



Conclusion - détection/correction d'erreur

Codes cycliques - CRC

Checksum