Sommaire

1 Contrôle des données 1
1.1 Introduction 1
1.2 Contrôle à parité 2
1.3 Le contrôle de parité croisé 3
1.4 Le Contrôle de Redondance Cyclique 4
1.5 D'autres types de code : 5
2 Correction des données 5
2.1 Principe de la correction 5
2.2 Les algorithmes de correction 6
2.2.1 Le code de Hamming 6
2.2.2 Le Code de Reed-Solomon 6

1 Contrôle des données

1.1 Introduction

Le contrôle des données consiste à assurer la véracité des données transmises. Ce contrôle est mis en place afin d'assurer la cohérence de celles-ci, que ce soit pour garantir une transmission de donnée à travers une ligne de communication ou pour un stockage des données en mémoire ou sur une unité de stockage. Cette recherche de cohérence est liée au fait que les composants ou les transmissions peuvent subir des perturbations dégradant les données.

Afin d'assurer cette véracité, plusieurs mécanismes plus ou moins complexes ont été mis en place. Tous ces mécanismes sont basés sur l'ajout d'informations complémentaires "codant" ce contrôle.

Le contrôle de parité fonctionne selon un principe très simple. Aux n bits que comporte le code à l'origine, on ajoute un bit supplémentaire.

1.2 Contrôle à parité

Le contrôle de parité a été le premier système mis en place et reste l'un des plus simple.
Son principe est basé sur l'utilisation d'un bit supplémentaire assurant la fonction de parité paire ou impaire.

En cas de parité paire:

- le bit de parité vaut 0 si le nombre de bits précédents est pair
- le bit de parité vaut 1 si le nombre de bits précédents est impair

En cas de parité impaire:

- le bit de parité vaut 0 si le nombre de bits précédents est impair
- le bit de parité vaut 1 si le nombre de bits précédents est pair

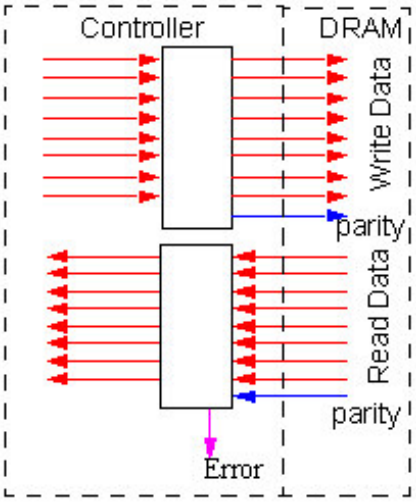
Ce type de codage est utilisé :

- En transmission série (norme V24)
  - On parle de transmission en 7 bits parité paire, 8 bits parité impaire..
- En contrôle de plan de mémoire dynamique

Le principe fait appel à deux mécanismes :

- Un système réalise le codage du bit de contrôle
- Un système réalise la vérification du bit de contrôle

<u>Mécanisme de codage et de contrôle de parité sur un plan mémoire</u>



Ce contrôle est très utilisé, cependant est fragile car incapable de détecter plus d'une erreur, voire même la donnée peut être entièrement erronée sans que cela ne se remarque.
Si plus d'un bit est en erreur, celle-ci ne sera pas détectée.

Exemple en parité paire :

- 1100 0011 0 Codage de départ
- 1100 1011 0 **<u>Erreur</u>** : octet → 5 chiffres 1 donc le bit de parité devrait être 1
- 1100 0000 0 **<u>Erreur non détectable</u>** (2 bits sont passés de 1 à 0)

1.3 Le contrôle de parité croisé

Ce type de contrôle ne consiste pas uniquement à contrôler l'intégrité des données d'un caractère mais à contrôler l'intégrité des bits de parité d'un bloc de caractères.

Ce type de contrôle est basé sur deux mots de contrôle :

- **Le VRC (Vertical Redundancy Check)** Contrôle vertical de redondance. Désigne la parité appliquée à un mot et non pas à la suite des mots (parité longitudinale).
- **Le LRC (Longitudinal Redundancy Check)** Contrôle longitudinal de redondance : système de détection d'erreurs par parité s'appliquant à la totalité d'un bloc, par opposition à la parité « verticale » qui s'applique à chaque mot de ce bloc.

Exemple :

Dans ce cas, l'information est codée sur 7 bits, le 8 éme bit étant réservé qu codage du bit de contrôle.

	LRC ↓							
	1	1	1	0	1	1	0	1
	1	1	0	1	1	0	1	1
	0	0	0	1	0	0	0	1
VRC →	0	0	1	0	0	1	1	1

La vérification du bloc est plus robuste aux erreurs que le précédent puisqu'il assure un double contrôle horizontal et vertical. En cas d'erreur, il est également possible de corriger celle-ci puisqu'elle est localisable :

Exemple d'erreur corrigeable :

	LRC ↓							
	1	1	1	0	1	1	0	1
	1	1	1	1	1	0	1	1
	0	0	0	1	0	0	0	1
VRC →	0	0	1	0	0	1	1	1

L'erreur ici est détectable et corrigeable. Ce code fait partie des premiers codes à correction d'erreur

Autre exemple :

	LRC ↓							
	1	1	1	0	1	1	0	1
	1	1	0	0	1	0	1	1
	0	0	0	1	0	1	0	1
VRC →	0	0	1	0	0	1	1	1

Dans ce cas, les 2 erreurs sont détectées mais ne peuvent être corrigées

Exemple d'erreur non corrigeable :

	LRC ↓							
	1	1	1	0	1	1	0	1
	1	1	0	0	0	0	1	1
	0	0	0	1	0	0	0	1
VRC →	0	0	1	0	0	1	1	1

Dans ce cas, les colonnes en cause sont repérées, par contre on ne sait pas quelle ligne est en défaut.

1.4 Le Contrôle de Redondance Cyclique

Ce type de contrôle est souvent désigné par ses lettres CRC.

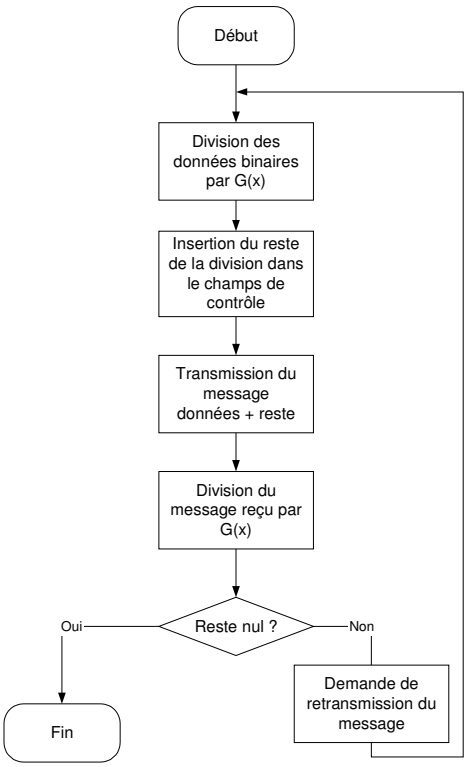
Ce mécanisme consiste à protéger des blocs de données en ajoutant un code de contrôle. Ce code « CRC » contient des éléments redondants par rapport aux données transmises de manière à permettre la détection des erreurs, mais également de les réparer dans certains cas. Il est utilisé dans le cas de transmission d'une grande série d'octets.

Ce code est basé sur le fait que toute chaîne binaire permet de construire un polynôme, chacun des bits donnant sa valeur au coefficient polynomial correspondant.
Ex :

$0101 \rightarrow 0x^3 + 1x^2 + 0x^1 + 1x^0 \rightarrow x^2 + 1$

La mise en place du code CRC nécessite de choisir un polynôme de référence appelé polynôme générateur nommé souvent G(x).

Algorithme de codage et décodage CRC :



- Le CCITT a normalisé l'utilisation d'un polynôme de degré 16 ($x^{16} + x^{12} + x^5 + 1$) pour les transmissions de données sur Transpac.
- Ethernet, utilise également un contrôle basé sur un CRC. Le "reste" porte le nom de FCS (Frame Check Sequence) et est codé sur 32 bits. . Le polynôme utilisé est du 32éme ordre:
  (. $x^{32} + x^{23} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$)
- Certains polynômes ont été normalisés, et on parle alors de CRC16, CRC12 …Dans tous les cas, il est important qu'émetteur et récepteur utilisent le même polynôme de référence.

x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x1 + 1

1.5 D'autres types de code :

Indépendamment des codages vus ci-dessus qui sont des codes normalisés, il est possible d'utiliser toute autre méthode de calcul du caractère de contrôle. Celui-ci peut être obtenu à partir d'opérations sur les caractères de la trame (addition, soustraction, fonctions logiques), et peut également être codé sur un ou plusieurs octets.

L'objectif reste dans tous les cas de fiabiliser la transmission en détectant les erreurs de transmission.

2 Correction des données

2.1 Principe de la correction

S'il est facile de détecter les erreurs (voir les méthodes ci-dessus), il est plus difficile de les corriger, puisqu'il faut que la transmission des données assure une redondance de l'information afin de pouvoir assurer cette correction. De plus, les algorithmes à mettre en œuvre sont plus complexes et donc plus long en temps de traitement.

En transmission de donnée, les méthodes de correction sont rarement appliquées (la redondance impliquant des trames beaucoup plus longues), les algorithmes assurent la détection et en cas d'erreur demandent la ré-émission de la trame en défaut.

Par contre, en cas de stockage des données, cette correction doit être mise en œuvre puisqu'en cas d'erreur, c'est la donnée elle-même qui est corrompue, et il est nécessaire de la reconstituer.

Les cas d'utilisation les plus courants sont :

- Les méthodes ECC pour les plans mémoires des serveurs
- Les technologies RAID pour les disques durs

2.2 Les algorithmes de correction

2.2.1 Le code de Hamming

Le plus célèbre d'entre eux est le code de Hamming (datant des années 50), ou plutôt la famille de code de Hamming (code normal, étendu, cyclique). L'utilisation de la méthode dépend essentiellement de la tolérance à l'erreur que l'on souhaite (correction d'1 ou de plusieurs bits en défaut).

Le principe est basé sur l'algèbre linéaire et sur l'utilisation de matrices (matrices de Hamming). Il consiste à rajouter des codes de contrôle en plus des informations à transmettre. Le nombre de ces codes de contrôle dépend directement du niveau de fiabilité que l'on souhaite obtenir.

Application à la mémoire ECC :

- 64 bits → 8 bits pour l'ECC
- 32 bits → 7 bits
- 8 bits → 4 bits

Application à la technologie Raid :

En technologie Raid, différentes techniques sont utilisées (codage par bloc ou secteur…), et le contrôle n'est pas systématiquement assuré par un code de Hamming au niveau du stockage, celui-ci étant déjà compris dans la transmission de l'information (codage au niveau des controleurs). Néanmoins, on peut retenir comme principe l'utilisation d'un disque de contrôle pour 3 disques de données.

2.2.2 Le Code de Reed-Solomon

Les supports sont soumis à de nombreux problèmes :

- Défaut de lecture du Laser
- Défaut de surface : empreintes de doigts, rayures, salissures …attaques :

La correction d'erreur sur un CD peut corriger jusqu'à des blocs de 3500 erreurs consécutives en utilisant plusieurs niveaux de codes détecteurs et correcteurs d'erreurs, séparés par un niveau d'entrelacement des informations : Cross Interleave Reed-Solomon (CIRC)