3 — Détection et correction des erreurs binaires

Réseaux
IUT de Villetaneuse
Département Informatique, Formation Continue
Année 2012–2013

http://www.lipn.univ-paris13.fr/~evangelista/cours/2012-2013/reseaux-fc

Les erreurs de transmission

- erreur de transmission = un bit b transmis par l'émetteur est mal interprété par le récepteur
- causes possibles aux erreurs de transmission :
  - le bruit
  - la désynchronisation entre le récepteur et l'émetteur
  - la déformation du signal sur les longues distances

    ...
- Pour résoudre ce problème, l'émetteur et le récepteur utilise un **code** de détection d'erreur.

Principe des codes de détection d'erreurs

- Avant transmission, l'émetteur va rajouter des bits à ceux transmis.
- Ces bits sont calculés selon un algorithme connu à la fois de l'émetteur et du récepteur.
- Ils sont ensuite utilisés par le récepteur pour déterminer si la transmission s'est bien déroulée.
- À part pour la détection d'erreurs ces bits ne transportent aucune information utile.

⇒ On dit que l'émetteur introduit de la **redondance**.

- Remarques
  - Aucun code n'est parfait : un code pourra toujours laisser passer des erreurs (i.e., ne pas les détecter).
  - En cas d'erreur, certains codes permettent de corriger l'erreur. On parle alors de **code correcteur**.
- Nous allons voir trois familles de codes :
  - les codes à contrôle de parité
  - les codes CRC
  - les codes de Hamming

Codes à contrôle de parité — Le VRC

- **VRC** = Vertical Redundancy Check
- on insère à chaque mot de N bits un **bit de parité** pour que le nombre de bits à 1 dans le mot (modifié) de $N+1$ bits soit
  - pair (VRC pair) ;
  - ou impair (VRC impair).
- Le récepteur doit ensuite contrôler que le nombre de bits à 1 dans chaque mot est pair (ou impair).
- Exemple : on veut envoyer les trois caractères ASCII suivants : `IUT`.
  - (Les caractères ASCII sont codés sur 7 bits.)

	ASCII	VRC pair	VRC impair
I	1001001	**1**	**0**
U	1010101	**0**	**1**
T	1010100	**1**	**0**

⇒ séquence envoyée avec un VRC pair :

1001001**1** 1010101**0** 1010100**1**

Codes à contrôle de parité — Le LRC

- **LRC** = Longitudinal Redundancy Check
- après une séquence de N mots de b bits, on insère un **mot de parité** de b bits pour que le nombre de bits à 1 dans chaque colonne soit
  - pair (LRC pair) ;
  - ou impair (LRC impair).
- Exemple :

	ASCII	
I	1001001	
U	1010101	
T	1010100	
	1001000	LRC pair
	0110111	LRC impair

⇒ séquence envoyée avec un LRC pair :

$$\underbrace{1001001}_{\texttt{I}}\,\underbrace{1010101}_{\texttt{U}}\,\underbrace{1010100}_{\texttt{T}}\,\underbrace{\mathbf{1001000}}_{\text{LRC}}$$

Codes à contrôle de parité — VRC + LRC

- On peut combiner les deux codes :
  - rajouter un bit de parité à chaque mot (VRC)
  - rajouter un mot de parité à la fin d'une séquence de mots (LRC)
- Exemple :

	ASCII	VRC pair	VRC impair	
I	1001001	**1**	**0**	
U	1010101	**0**	**1**	
T	1010100	**1**	**0**	
	1001000	**0**	**1**	LRC pair
	0110111	**1**	**0**	LRC impair

⇒ séquence envoyée avec un VRC pair + LRC impair :

$$\underbrace{1001001\mathbf{1}}_{\texttt{I}}\,\underbrace{1010101\mathbf{0}}_{\texttt{U}}\,\underbrace{1010100\mathbf{1}}_{\texttt{T}}\,\underbrace{\mathbf{01101111}}_{\text{LRC}}$$

Code à redondance cyclique — Principe

- **CRC** = Code à Redondance Cyclique
- On parle aussi de codes polynômiaux.
- On représente une séquence de bits comme un polynôme binaire.
  - Tous les coefficients du polynôme sont 0 ou 1.
- Une séquence de n bits est un polynôme à n termes de x^0 à x^{n-1} :
  - Le bit le plus à gauche indique le coefficient de x^{n-1} ;
  - celui à sa droite indique le coefficient de x^{n-2} et ainsi de suite.
  - Enfin, le bit le plus à droite indique le coefficient de $x^0 = 1$.
- Exemple : la séquence 100101 correspond à $x^5 + x^2 + 1$.
- L'émetteur et le récepteur utilisent tous les deux un **polynôme générateur** $G(x)$ (qui est constant).
- Ce polynôme $G(x)$ va servir :
  - à l'émetteur pour déterminer les bits qui seront effectivement transmis ;
  - et au récepteur pour déterminer si la transmission s'est déroulée sans erreur.

Code à redondance cyclique — Codage

Soient

- M le message à envoyer
- et k le degré du polynôme générateur $G(x)$ (le degré est la puissance du plus grand terme du polynôme).

Algorithme :

1. Calculer $M(x)$ le polynôme correspondant à M.
2. Calculer $P(x) = M(x) \times x^k$
   - Cette opération revient à faire un décalage à gauche de k bits sur M.
3. Effectuer la division de $P(x)$ par $G(x)$
   - Le reste de la division est un polynôme $R(x)$ de degré inférieur à k.
4. Calculer $M'(x) = P(x) + R(x)$.
   - $M'(x)$ est forcèment divisible par $G(x)$.
5. Transmettre le message correspondant à $M'(x)$ (plutôt que $M(x)$).
   - Le message transmis est le message initial auquel on ajoute k bits qui correspondent au reste de la division.

Code à redondance cyclique — Décodage

Soit

- M' le message reçu

Algorithme :

1. Calculer $M'(x)$ le polynôme correspondant à M'.
2. Effectuer la division de $M'(x)$ par $G(x)$.
3. Si le résultat de la division vaut 0 $\Rightarrow$ pas d'erreur détectée
4. sinon $\Rightarrow$ erreur détectée
   - (Les codes à redondance cyclique ne permettent pas de corriger les erreurs.)

Code à redondance cyclique — Exemple

Soient

- $G(x) = x^3 + x + 1$ le polynôme générateur ;
- et $M = 1101$ le message à envoyer.

Codage par l'émetteur

1. $M(x) = x^3 + x^2 + 1$
2. $P(x) = M(x) \times x^3 = x^6 + x^5 + x^3$
3. La division de $P(x)$ par $G(x)$ donne le reste $R(x) = 1$.
   (voir diapositive suivante pour le détail de l'opération)
4. $M'(x) = P(x) + R(x) = x^6 + x^5 + x^3 + 1$
5. transmettre $M' = 1101001$

Décodage par le récepteur

1. $M'(x) = x^6 + x^5 + x^3 + 1$
2. La division de $M'(x)$ par $G(x)$ donne un reste de 0.

$\Rightarrow$ pas d'erreur détectée, la transmission s'est bien passée

Code à redondance cyclique — Exemple de division

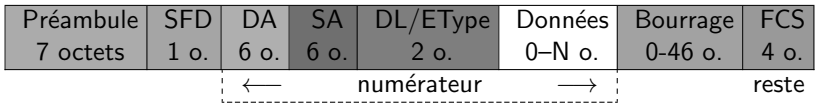
- On divise $P(x) = x^6 + x^5 + x^3$ par $G(x) = x^3 + x + 1$.
- L'addition se fait modulo 2 (sans retenue) :
  - $1 + 1 = 0 + 0 = 0$
  - $0 + 1 = 1 + 0 = 1$
- À chaque étape on élimine le terme le plus élevé du numérateur.
- La division s'arrête lorsqu'il est impossible d'éliminer ce terme.

$$
\begin{array}{rl}
x^6 + x^5 + x^3 & \mid\, x^3 + x + 1 \\
\hline
+\ x^6 + x^4 + x^3 & \mid\, x^3 \\
\hline
x^5 + x^4 & \\
+\ x^5 + x^3 + x^2 & \mid\, + x^2 \\
\hline
x^4 + x^3 + x^2 & \\
+\ x^4 + x^2 + x & \mid\, + x \\
\hline
x^3 + x & \\
+\ x^3 + x + 1 & \mid\, + 1 \\
\hline
R(x) = 1 &
\end{array}
$$

- On a donc : $P(x) = (\underbrace{x^3 + x + 1}_{G(x)}) \times (\underbrace{x^3 + x^2 + x + 1}_{\text{résultat de la division}}) + \underbrace{1}_{R(x)}$.

Code à redondance cyclique — Cas d'Ethernet

Rappel sur la structure des trames Ethernet



- Ethernet utilise un CRC basé sur un polynôme de degré 32 pour détecter les erreurs de transmission.
  - degré 32 $\Rightarrow$ le reste fait au max. 32 bits = 4 octets
- L'émetteur de la trame effectue la division des bits des champs DA + SA + DL/Etype + Donnés par ce polynôme générateur.
- Il place le résultat de la division dans le champ FCS.
- A la réception, le récepteur effectue la division sur ces même champs puis vérifie qu'il trouve bien le FCS.
  - Si ce n'est pas le cas la trame est ignorée.

Codes de Hamming

- inventé par Richard Wesley Hamming, mathématicien
- On utilise un **dictionnaire** D de n mots de b bits.
- Ce dictionnaire est connu de l'émetteur et du récepteur.
- À l'émission, on ne transmet que des mots de D.

$\Rightarrow$ Chaque séquence de $log_2(n)$ bits doit être associé à un des mots de D.

- Si le récepteur reçoit un mot m qui n'est pas dans D, il remplace (si possible) m par le mot m' de D dont il est le plus proche.
- Pour trouver ce mot :
  1. On calcule la **distance** de m par rapport à chaque mot de D.
     - distance entre 2 mots m et $m' = d(m, m')$ = nombre de bits qui différent dans les 2 mots
  2. On prend le mot m' pour lequel la distance à m est la plus petite.
  3. Si il y a plusieurs mots pour lesquels la distance est minimale $\Rightarrow$ m ne peut pas être corrigé.

Codes de Hamming — Exemple (1/2)

- On a $D = \{$ 0000, 1011, 1100, 1111 $\}$.
- Comme D contient 4 mots et $log_2(4) = 2$, l'émetteur transmettra un mot de D pour chaque couple de bits du message à envoyer.
- On supposera que : 00→0000, 10→1011, 01→1100 et 11→1111

Côté émetteur

- L'émetteur veut transmettre 00 10 01.
- Après remplacement par des mots de D il transmet 0000 1011 1100.

Codes de Hamming — Exemple (2/2)

Côté récepteur

- Le récepteur reçoit 0000 0011 1000.
- 0000 est bien un mot de $D \Rightarrow$ OK, il est accepté
- 0011 n'est pas un mot de $D \Rightarrow$ on le compare à chaque mot de D
  - $d(0011, 0000) = 2$
  - $d(0011, 1011) = 1$
  - $d(0011, 1100) = 4$
  - $d(0011, 1111) = 2$
  - Distance de 1 par rapport à 1011 $\Rightarrow$ 0011 est corrigé en 1011.
- 1000 n'est pas un mot de $D \Rightarrow$ on le compare à chaque mot de D
  - $d(1000, 0000) = 1$
  - $d(1000, 1011) = 2$
  - $d(1000, 1100) = 1$
  - $d(1000, 1111) = 3$.
  - Distance de 1 par rapport à 0000 **et** 1100 $\Rightarrow$ 1000 ne peut pas être corrigé.

Comparatif des codes

Critères de comparaison :

- simplicité ($\Rightarrow$ rapidité d'exécution)
- redondance : nombre de bits de redondance utilisés
- détection : probabilité de détecter une erreur
- correction : possibilité de corriger les erreurs

Code	simplicité	redondance	détection	correction
VRC, LRC	**très simple**	moyenne	faible	non
VRC + LRC	**très simple**	moyenne	faible	**oui**[1]
CRC	compliqué	**faible**[2]	**excellente**[3]	non
Hamming	simple	élevé	moyenne[4]	**oui**

[1] On peut corriger un seul bit erroné : le VRC permet de trouver le mot contenant ce bit et le LRC nous donne la position du bit dans ce mot.

[2] On ajoute juste le reste de la division au message, quelle que soit sa taille.

[3] Avec certains polynômes générateurs on peut détecter 95% des erreurs.

[4] Le nombre d'erreurs que l'on peut détecter/corriger dépend du dictionnaire.