

TD5 – Couche liaison (Datalink layer)

{Nicolas.Ollinger, Emmanuel.Godard, Yann.Esposito}@lif.univ-mrs.fr

17 novembre 2004

☞ Les canaux de transmission imparfait entraînent des erreurs lors des échanges de données. La probabilité d'erreur sur une ligne téléphonique est comprise entre 10^{-4} et 10^{-7} . Ce TD est consacré à la compréhension des codes de détection et de correction d'erreurs. Il s'agit de méthodes mises en place au niveau de la deuxième couche OSI.

Pour l'envoi de k bits, on rajoute r bits qui permettent de tester si une erreur a eu lieu.

Dans la suite on utilisera la terminologie suivante :

- la totalité des $k + r$ bits sera appelé une *trame* ;
- la suite de k bits sera appelé le *message* ;
- la suite des r bits ajoutés au message sera appelé le *code correcteur* ou tout simplement le *code*.

1 Bit de parité

☞ Lors de l'envoi de k bits, on compte le nombre qui est égal à 1. Si ce nombre est pair on ajoute le bit 0 sinon on ajoute le bit 1 aux k bits envoyés.

1.1. Peut-on détecter si un seul bit est erroné ? deux ?

1.2. On considère que la probabilité d'erreur est $p = 10^{-6}$ et que $k = 7$.

(a) Donnez la probabilité qu'une trame soit erronée.

(b) Donnez la probabilité de faire un mauvais contrôle : c'est-à-dire accepter un message erroné ou ne pas accepter un message correct. En déduire une estimation du nombre moyen de bons contrôles par mauvais contrôle.

1.3. Donnez une estimation du temps moyen pour qu'une trame erronée soit acceptée sur une connexion 10Mo/s.

1.4. Si une erreur est détectée, que peut-on faire ?

2 Parité double

☞ Dans ce cas les trames sont considérées comme des matrices $n \times n$ (généralement $n = 8$) bits. On ajoute alors un bit de parité par colonne et par ligne.

2.1. Décrivez les avantages et les inconvénients de cette méthode par rapport à la précédente.

2.2. Erreurs de contrôles

(a) Qu'est-ce que faire un mauvais contrôle dans ce cas ?

(b) Dans quel cas ces mauvais contrôles se produisent-ils ? Quels sont les cas les plus probables ? Quelle est la probabilité pour chacun de ces cas qu'ils se produisent ?

3 Code de Hamming

☞ Il s'agit d'utiliser $\log_2 k$ bits de parité et d'avoir un moyen rapide de détecter la position de l'erreur.

On veut envoyer un message de m bits où $m = (2^n - 1) - n$ bits. On parle de code $x - y$ où $x = n + m$ et $y = m$. Par exemple le code de Hamming 7 - 4 a une efficacité de $4/7 = 57\%$, 31 - 26 efficacité de 83%. Les bits de contrôle de parité C_i sont en position 2^i . Les bits du message D_j occupent le reste de l'envoi.

Lorsque l'on veut envoyer le message $D_3D_2D_1D_0$, on ajoute le code correcteur $C_2C_1C_0$ aux positions 2^i comme suit :

D_3	D_2	D_1	C_2	D_0	C_1	C_0
-------	-------	-------	-------	-------	-------	-------

Ensuite, à la réception, on recalcule les valeur des bits aux positions 2^i

D_3	D_2	D_1	C'_2	D_0	C'_1	C'_0
-------	-------	-------	--------	-------	--------	--------

Si $C'_2C'_1C'_0$ vaut 0 alors on considère qu'il n'y pas d'erreur, sinon, la valeur correspond à l'indice auquel une erreur est survenue.

Lorsque $C'_0 = 1$ alors les valeurs possibles pour $C'_2C'_1C'_0$ sont : 001, 011, 101 et 111 c'est-à-dire 1, 3, 5 et 7. Il s'agit des indices utilisés pour le calcul de C_0 et de C'_0 .

3.1. Calculez les indices de calcul pour C_1 et C_2

Si $I_1 \dots I_n$ sont les indices utilisés pour le calcul de C'_i , alors la valeur de $I_1 \oplus \dots \oplus I_n$.

3.2. On souhaite envoyer le mot 1010 sachant que le mot envoyé va être de la forme

1	0	1	C_2	0	C_1	C_0
---	---	---	-------	---	-------	-------

Calculez les valeurs de C_0 , C_1 et C_2 pour qu'à la réception les $C'_2C'_1C'_0$ soit égal à 0.

3.3. Vérifiez que si une erreur à lieu, la valeur de $C'_2C'_1C'_0$ en donne la position.

4 Détection d'erreur par CRC (Cyclic Redundancy Check)

☞ On considère chaque message comme un polynôme à coefficients dans $\{0,1\}$. Par exemple 1101 correspond au polynôme : $x^3 + x^2 + 1$ On considère ensuite un polynôme commun à l'émetteur et au récepteur. Il s'agit du polynôme *générateur*. On va alors décaler vers la droite le message d'autant que le degrés du polynôme générateur. On va alors diviser le message décalé par ce polynôme. Il suffit alors de supprimer (concaténer au message) le reste de la division. Le récepteur fait la division et vérifie que le reste est nul.

On peut lorsque le polynôme commun est de degrés 16 détecter toutes les erreurs simples ou doubles, toutes les erreurs comportant un nombre impair de bits et tous les paquets d'erreurs de longueur inférieure ou égale à 16. Avec une très bonne probabilité, les paquets d'erreurs de longueur supérieure.

Par exemple Ethernet utilise un champs CRC à 32 bits avec le polynôme générateur :

$$X^{32} + X^{26} + X^{23}X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

4.1. Cette méthode est-elle une méthode de détection d'erreur ou de correction d'erreur ?

4.2. Le polynôme diviseur est : 101 c'est-à-dire $x^2 + 1$. On veut coder le mot 11010100. Quel est la trame que l'émetteur doit envoyer ?