Réseaux 1
Couche Liaison de données

Abdelkader OUALI

Université de Caen Normandie
Laboratoire GREYC

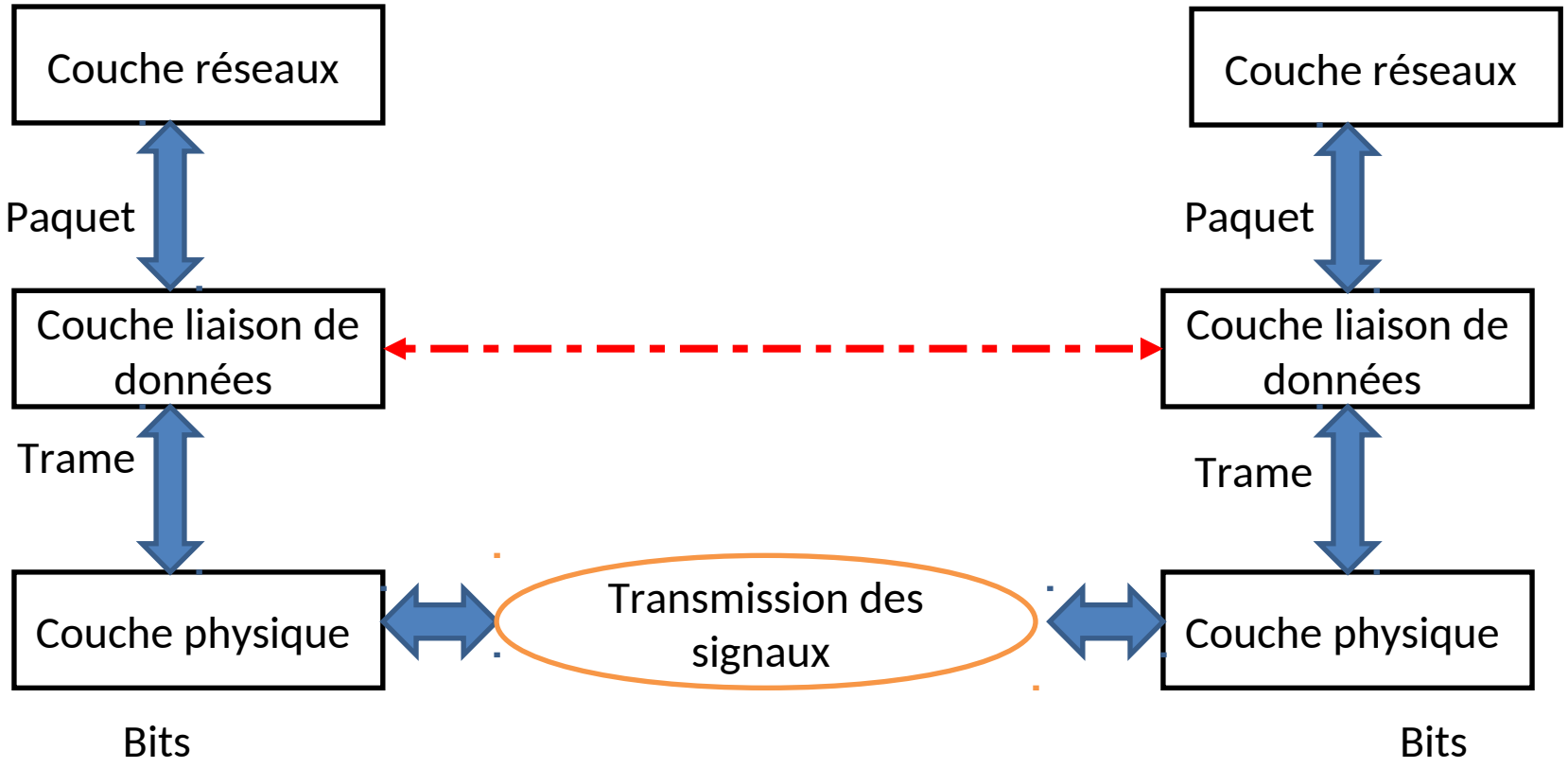
abdelkader.ouali@unicaen.fr

Plan

- Couche liaison de données
  - Types d'erreurs
  - Détection d'erreur
  - Correction des erreurs
- Contrôle d’accès au support
  - Accès point à point
  - Accès multiple

Définition

- Ensemble des **matériels** et **logiciels** permettant d'assurer une transmission **fiable** des données sur la liaison physique
- L'**unité d'information** associée à la couche 2 du modèle **OSI** est **trame** ou **L-PDU**



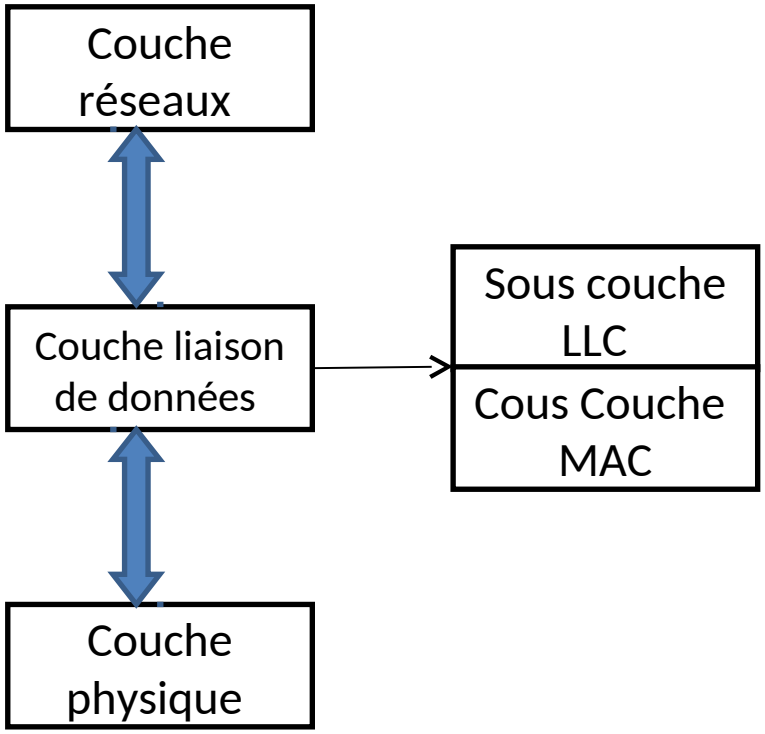
Rôles de la couche liaison de données

- **Découpage en trame**
  - **Délimiter** les données **issus** de la couche **réseaux**
- **Contrôle d'accès au media de transmission**
  - Quelle machine a le droit d'utiliser le support pour envoyer les données
- **Adressage**
  - **Identification** physique des machines
- **Contrôle d'erreurs**
  - Assurer le transfert **sans erreurs** des données
- **Contrôle de flux**
  - Assurer un transfert **fiable** de données

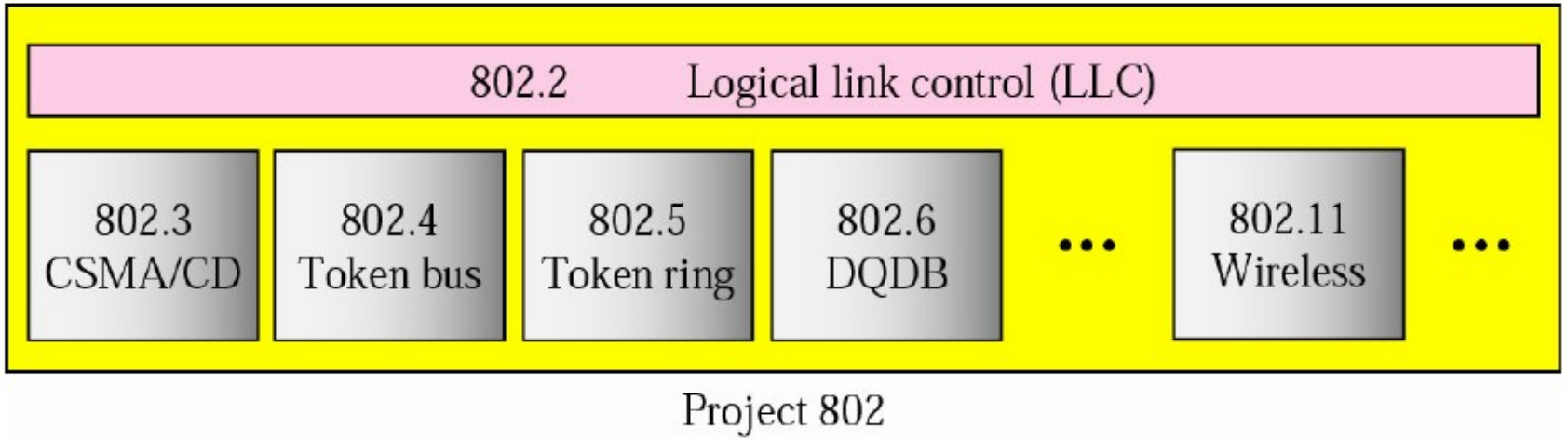
Couche liaison de données
Deux sous-couches

➢La couche liaison de données est découpée en deux sous couches :

- **LLC** (Logical Link Control)
- **MAC** (Media Access Control)



Standard IEEE pour les LANs



Project 802

Sous Couche LLC
Contrôle d’erreurs

Sous-couche LLC

- La sous-couche **LLC** (Logical Link Control) est l'**interface** avec la couche **Réseau** en lui offrant une couche **logique** plutôt que physique
- Elle a pour rôle :
  - La protection contre les erreurs de transmission .
  - Assurer le transfert des trames et le contrôle de flux entre les stations du réseau.
- **Indépendante** de la méthode d'accès utilisée par la sous couche **MAC**

Erreurs de transmission

- Des erreurs sont dues aux canaux de transmission.
- **Un** ou **plusieurs** bits d'information peuvent être **changés** en même temps durant la **transmission** des données
- Les données peuvent être **corrompues** ou **perdues**
- Les erreurs sont causées par :
  - L'interférence (Bruit)
  - La distorsion

Comment prévenir les erreurs
(solutions physiques)

- Pour réduire les interférences
  - **Blinder** les fils
  - S'assurer que les câbles sont **loin** des **sources** d'interférence (Bruit)
- Pour réduire la distorsion
  - **Ajuster** l'équipement de transmission et améliorer la qualité de la connexion
  - Utiliser des **amplificateurs** et des **répéteurs**
  - Utiliser du **câble** de **meilleur qualité**
- Mais **le risque d'erreurs existe toujours**, mais il doit pas dépasser un certain seuil 10^{-8} à 10^{-10}

Contrôle d'erreurs

- Il faut pouvoir les **détecter** et **corriger** les erreurs.
- De façon générale pour transmettre k bits d'information , on ajoute **r bits** dit **bits de contrôle**
- On parle de code(n, k) ou de mot de code.
- Les bits de contrôle sont **calculés en fonction** des bits de l'information
- Au total on transmis **n= k + r** bits

- **À la réception** les bits de contrôles seront **recalculer** afin de s'assure si l'information est **bien reçus** ou non

Technique de contrôle d'erreurs

- Il existe **plusieurs méthodes** de contrôle d'erreurs :
  - **Détection (code détecteur)**
    - **Détecter** le changement de **un** ou **plusieurs** bits d'information .
    - **Pas de possibilité** pour corriger ces erreurs
  - **Détection et correction (code correcteur)**
    - **Détecter** le changement de **un** ou **plusieurs** bits d'information .
    - **Capacité** de corriger ces erreurs

Détection d’erreurs

➢Les techniques les plus utilisées pour la détection d’erreurs sont :

- **VRC (Vertical Redundancy Check)**

  Parité vertical

- **LRC (Longitudunal Redundancy Check)** :

  Parité longitudinale

- **CRC (Cyclic Redundancy Check)**

  Vérification polynomiale

Parité : VRC(Vertical Redundancy Check)

- Mécanisme le plus ancien
- Calculer **la parité est rajouter un bit** à l'information envoyé

  - **Parité paire :** si le **nombre de 1** dans l'information est **paire** alors le bit de parité est égale à **1** , sinon **0**
  - **Parité impaire :** si **le nombre de 1** dans l'information est **impaire** alors le bit de parité est égale à **1** , sinon **0**

- **Exemple :**

  Si on utilise une parité impaire pour l'information 1100100

  alors, on rajoute 1.

  L'information à envoyer serait donc 1100100**1**

Détection des erreurs grâce au code VRC

- La détection d'erreur avec le **VRC** consiste a :
  - **Recalculer** le bit de parité à la réception et le **comparer** avec le bit de parité **reçu**

Exemple : l'information reçues 1100100 **1**

le bit de parité qui correspond à l'information : 1100100 est égal à 1 donc l'information reçues est **correcte**

- **Capacité de détection :** si un seul bit change alors la parité change → on détecte l'erreur mais on peut pas savoir quel est le bit qui a changé.
- Si le nombre de de bits qui change est pair → impossible de détecter l'erreur

1100100 → 11**1**0**0**00

Parité longitudinale : LRC

- Appliquer le principe de la parité (**paire** ou **impaire**) aux colonnes d'un bloc de données
- Exemple le message DATA:

Caractère	Code ASCII
D	1000100
A	1000001
T	1010100
A	1000001
LRC	**1101111**

Parité longitudinale : LRC

- C'est un **code meilleur** que le **VRC**
- **Impossible** de **détecter** l'erreur si **deux bit sont changés** en même temps sur la même colonnes
- **Pour plus d'efficacité**
  - **Rajouter** un **contrôle** sur les **lignes**

Caractère	ASCII	Bit de parité
D	1000100	1
A	1000001	1
T	1010100	0
A	1000001	1
LRC	1101111	1

Vérification polynomiale

- Une information **en binaire** peut être écrit sous la **forme polynomial** suivant les puissance de 2

$$(1110)_2 = 1*2^3 + 1*2^2 + 1*2^1 + 0*2^0$$

Dans le cas général :

$$(u_k u_{k-1} \ldots\ldots\ldots u_1 u_0)_2 = u_k.X^k + u_{k-1}.X^{k-1} + \ldots\ldots\ldots + u_1.X^1 + u_0.X^0 \text{ avec } u_i \in [0,1]$$

Exemple :
La suite **1100101** est représentée par le polynôme

$$1100101 = 1.X^6 + 1.X^5 + 0.X^4 + 0.X^3 + 1.X^2 + 0.X^1 + 1.X^0$$
$$= X^6 + X^5 + X^2 + 1$$

Calcul du CRC

- On choisit un polynôme appelé polynôme générateur
  - G(X) de **degré *n***

    Exemple :

    Polynôme générateur G(X)=$\mathbf{X^4+X^2+X}$ de **degré 4**
- Soit une information sur **m bits** représentée sous la **forme d'un polynôme M(X)** de **dégrée m**
- **Pour calculer le CRC :**
  - **Multiplier** le polynôme **M(X)** par $\mathbf{X^n}$ (**n** est le **degré** du **polynôme générateur**)
  - **Division** de ($\mathbf{X^n * M(X)}$) / **G(X)**,
  - **Quotient** Q(X) et le **reste** R(X)

    $$\mathbf{X^n * M(X) = Q(X) * G(X) + R(X)}$$
  - Le **CRC** correspond au **reste** de la division R(X)
- Donc l'**information a envoyé** est égale à : **M(X)R(X)**

Exemple

Soit l'information 11100111 et le polynôme générateur X^4+X^2+X

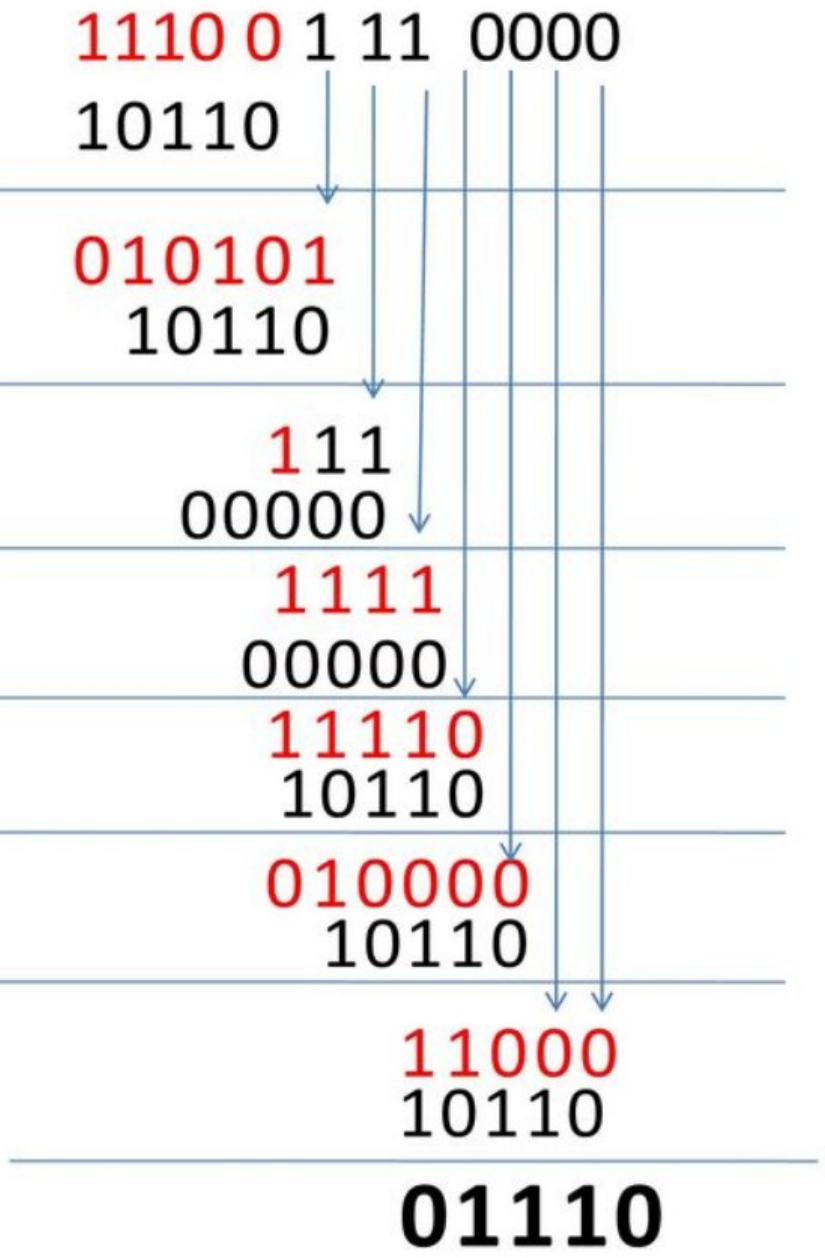
$G(X)= X^4+X^2+X$

$M(X)=X^7+X^6+X^5+X^2+X^1+1$

Multiplier M(X) par X^4

$$X^{11}+X^{10}+X^9+X^6+X^5+X^4$$

$$X^4.M(X)=111001110000$$



10110

11001101

L'information à envoyer :
11100111 **1110**

Remarque :

Toutes les opération se font en binaire modulo2

1+1 =0

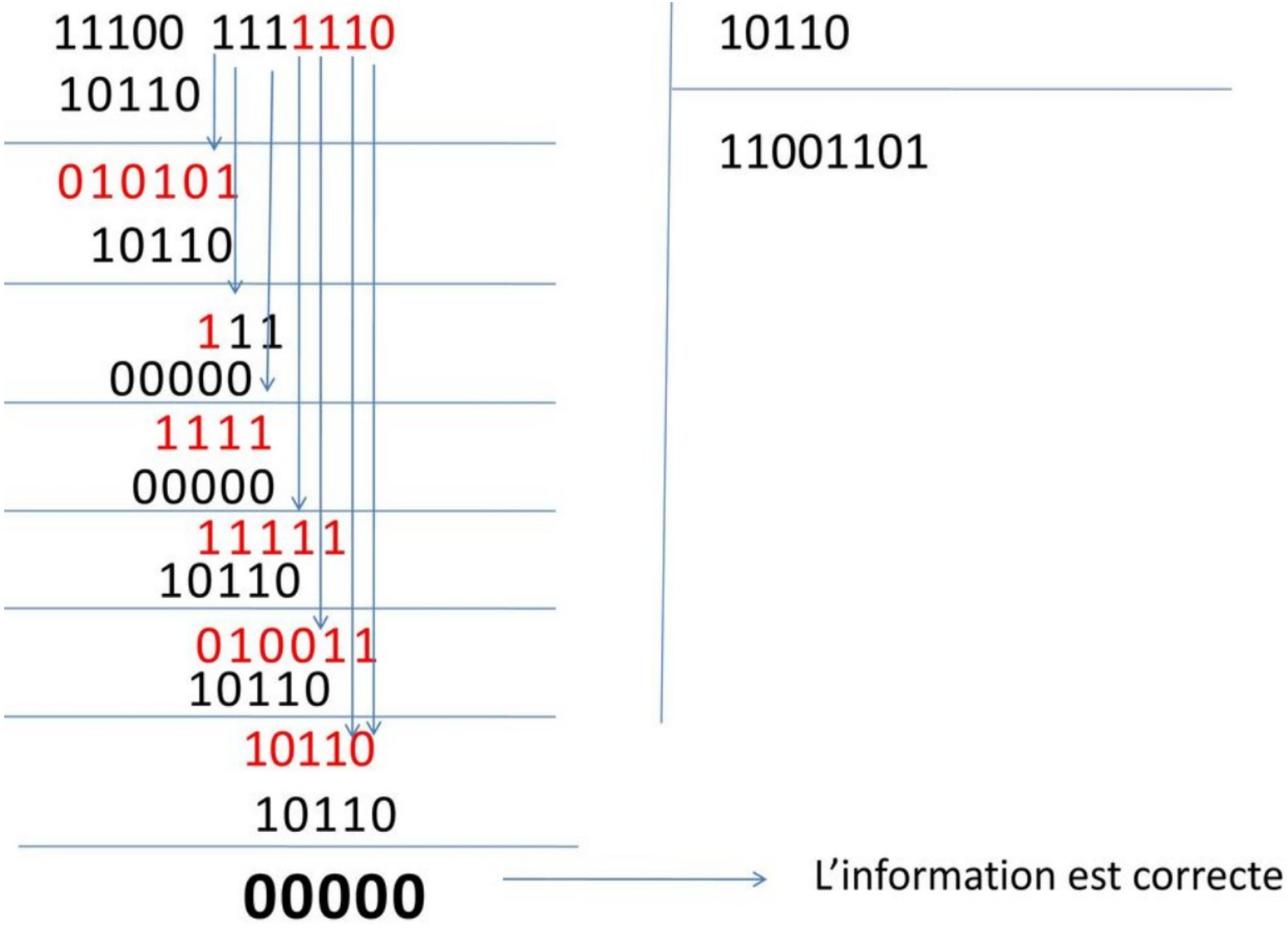
1+0 = 1

0+0 =0

Effectuer des ou exclusifs

Détection des erreurs

- À la réception de l'information
  - **diviser le polynôme qui représente** l'information sur le polynôme générateur
- Si le reste de la division **est nul** alors
  - l'information est **correcte**
- Si non
  - il y une **erreur**


11100 11111110
10110
010101
10110
111
00000
1111
00000
11111
10110
010011
10110
10110
10110
00000
10110
11001101
L'information est correcte

Calcul du CRC par des additions successives

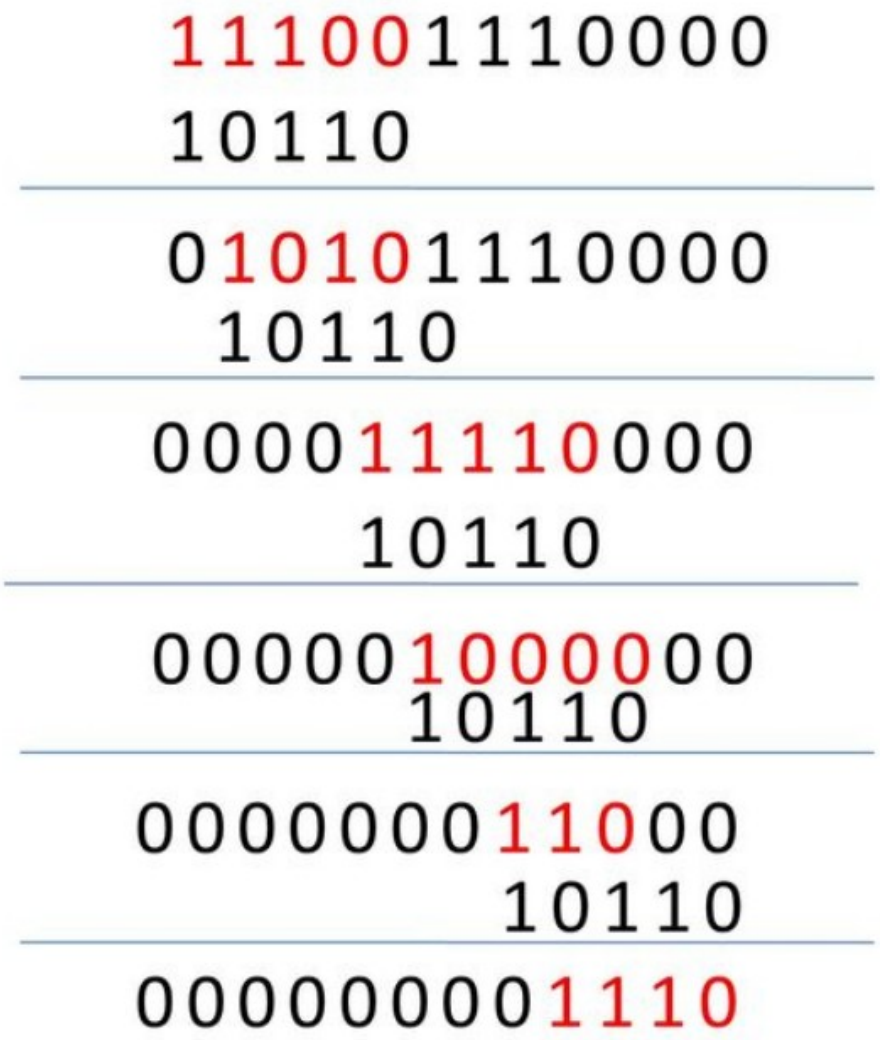
- Soit G(X) un polynôme générateur de degré n . On le transforme en un mot binaire
- **Exemple :**

  Avec le polynôme générateur X^4+X^2+X, on obtient 10110.
- On ajoute **n zéros** au **mot binaire à transmettre** où n est le degré du polynôme générateur
- **Exemple :**

  On souhaite transmettre le mot 11100111 en utilisant le polynôme générateur X^4+X^2+X, on obtient alors 11100111 **0000**.

- On va **additionner itérativement** à ce mot, le mot correspondant au polynôme générateur jusqu'à ce que le mot obtenu soit inférieur au polynôme générateur

- Ce **mot obtenu** correspond au **CRC** à ajouter au mot **avant de l'émettre**



Le code CRC = 1110 donc l'information a transmettre

1110011 1110

A la réception , refaire la même opération .
Si le résultat est nul alors l'information est correcte.

```
11100111 1110
10110
---------
01010111 1110
 10110
---------
00001111110
    10110
---------
00000100110
     10110
---------
00000001010
       10110
---------
00000000000
```

Le reste de la division est nul donc l'information est correcte

Exercice :

- Soit le polynôme générateur $G(X)= X^4+X^2+X$
- Calculer le CRC pour les deux informations suivantes

  - 1111011101
  - 1100010101

Normalisation des polynômes générateurs

- Le CCITT a recommandé un certain nombre de polynômes :

- $CRC-12 = x^{12} + x^{11} + x^{3} + x^{2} + x + 1$
- $CRC-16 = x^{16} + x^{15} + x^{2} + 1$
- $CRC-CCITT = x^{16} + x^{12} + x^{5} + 1$
- $CRC-32 = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x + 1$

Correction d'erreur par retransmission

- La méthode la plus **simple** pour **corriger** une erreur c'est de demander une **retransmission**

- Un récepteur qui **détecte** une erreur demande une **retransmission** jusqu'à ce qu'il **n'y est plus d'erreur**.

Codes correcteurs

- Le principe des **codes correcteurs** est le même que celui des codes détecteurs
- **Lors de l'émission**
  - **Rajouter** des **bits de contrôle** supplémentaires
- **A la réception**
  - **Détecter** les erreurs grâce au bits de contrôle et **possibilité de corriger** ces erreurs
- Le code de **Hamming** est parmi ces codes

Code de Hamming

- Un code de **Hamming** est un code C(n,k) avec
  - $n = 2^r - 1$, et
  - $r = n - k$
- Exemple :
  - n=7 → r=3 → k=4
  - n=15 → r=4 → k=11

- Le codage de Hamming se base sur le calcul de **la parité**
- Au lieu de rajouter un **seul bit**, rajouter **plusieurs bits** de parité
- Chaque bit de contrôle est une **fonction** de plusieurs bits d'information

Principe du code de Hamming

- Soit un mot de code (7,4) → rajouter 3 bits de contrôles notés $C_0\,C_1\,C_2$
- et l'information de départ est sur 4 bits $m=U_0U_1U_2U_3$
- Les bits de contrôle sont insérés dans les bits de l'information De la façon suivante :
  - il prennent les position 2^i { 1,2,4,8,…)
  - et les bits d'information prennent les autres position

C0	C1	U0	C3	U1	U2	U3
2^0	2^1	3	2^4	5	6	7

Comment calculer les bits de contrôle

- Chaque bits de l'information possède une position dans le mot de code final
- Écrire cette position en puissance de 2

Exemple :

U_0 est dans la position $3=1+2 = 2^0+2^1$
U_1 est dans la position $5=1+4 = 2^0+2^2$
U_2 est dans la position $6=2+4 = 2^1+2^2$
U_3 est dans la position $7=1+2+4= 2^0+2^1+2^2$

Un bit de l'information ayant la position **J** participe au calcul du bit de contrôle ayant la position 2^i si 2^i **existe dans la décomposition en code binaire de la position J**

Dans l'exemple précédant :

- La position de C_0 est 2^0 → donc les bits U_0, U_1, U_3 participe dans le calcul de C_0 → $C_0=U_0+U_1+U_3$
- La position de C_1 est 2^1 → donc les bits U_0, U_2, U_3 participe dans le calcul de C_1 → $C_1=U_0+U_2+U_3$
- La position de C_2 est 2^2 → donc les bits U_1, U_2, U_3 participe dans le calcul de C_2 → $C_2=U_1+U_2+U_3$

Application

- pour l'information 1010 trouver le code de Hamming correspond (+ joue le rôle d'un ou-exclusif)
  - $C_0 = U_0+U_1+U_3 = 1+0+0 = 1$
  - $C_1 = U_0+U_2+U_3 = 1+1+0 = 0$
  - $C_2 = U_1+U_2+U_3 = 0+1+0 = 1$
- Donc l'information à envoyer : 1011010

Détection de l'erreur

- À la réception du message
  - **Recalculer** les bits de contrôle de la même **manière que lors de l'émission**
- Si égalité alors
  - Passage au bit suivant
- Sinon
  - Incrémenter un **compteur C** par la position du bit de contrôle
- Après avoir recalculer tous les bits de contrôle
  - Si le compteur est **égale a zéro** alors
    - pas d'erreur
  - Sinon
    - **numéro du bit erroné**

Exemple

- Si on envoi le message 1011010 et on reçoit le message 1011000 normalement il y a une erreur
- Pour la détecter :
  - $C'_0 = U_0 + U_1 + U_3$ → $C'_0 = 1+0+0 = 1$, **correcte,** $C=0$
  - $C'_1 = U_0 + U_2 + U_3$ → $C'_1 = 1+0+0 = 1$, **erreur,** $C=2$
  - $C'_2 = U_1 + U_2 + U_3$ → $C'_2 = 0+0+0 = 0$, **erreur,** $C=2+4$
  - Donc le bit erroné est le bit N° 6

Fin