

## Couche Liaison

# Contrôle d'erreur

# Objectifs

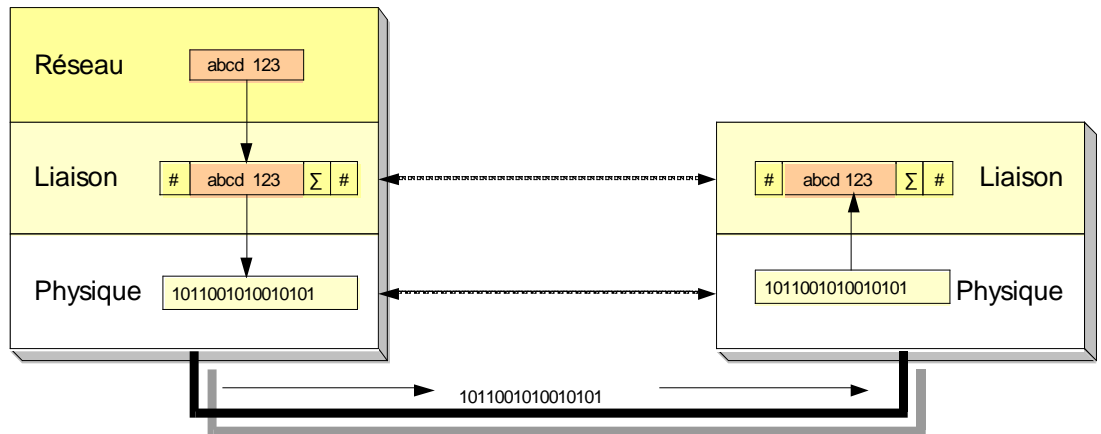
---

- Connaître les critères d'efficacité d'un contrôle d'erreur
- Pouvoir expliquer et appliquer les techniques de parité simple et croisée
- Savoir calculer le rendement d'un code détecteur/correcteur
- Savoir citer les avantages des codes polynomiaux
- A partir d'un polynôme générateur  $G(x)$  donné, être capable de calculer des mots de code
- A partir d'un polynôme générateur  $G(x)$  donné, être capable de détecter des erreurs sur une trame

# 2. Couche liaison

## Simuler une liaison parfaite, sans erreurs, à la couche supérieure

- Travaille entre systèmes voisins, à travers une seule liaison
- Découper les séquences de bits en paquets (appelés trames)
  - Reconnaître les frontières entre les trames
- Détecter et corriger des erreurs de transmission
- Régulation de flux



# Contrôle d'erreurs

---

- Les protocoles de la couche physique ne sont pas parfaits mais subissent des erreurs bits
- Le taux d'erreur bit dépend surtout du média de transmission
- La couche liaison implémente deux fonctions:
  - Détection d'erreurs
    - Codes détecteurs
  - Correction d'erreurs (seulement pour un service fiable)
    - Codes correcteurs
    - Retransmission de trames

# Critères d'efficacité

---

1. Capacité à détecter / corriger des erreurs multiples
2. Capacité à détecter / corriger des rafales d'erreurs d'une certaine longueur
3. Probabilité d'accepter une trame erronée comme correcte
4. Rendement: rapport entre les bits de données et la longueur totale des paquets

# Terminologie

---

- Quand on veut transmettre  $m$  bits de données
- On ajoute  $r$  bits de contrôle
- Et on obtient un mot de code de longueur  $n$

$$n = m + r$$

# Rendement

---

- Définition: rapport entre les bits de données et la longueur du mot de code

$$R = \frac{m}{n}$$

- Exemple:
  - Mot de données sur 8 bits ( $m$ )
  - Ajout de 4 bits de contrôle ( $r$ )
  - $n = m + r = 8 + 4 = 12$
  - $R = \frac{8}{12} = 67\%$

# Parité simple

---

- Méthode la plus simple de détection d'erreurs
- Ajout d'un bit de parité à la fin d'un bloc de données
- Deux types:
  - **Parité paire:** s'assurer que la somme des bits est paire (y compris le bit de parité). **Ex:** 010100, 011101
  - **Parité impaire:** s'assurer que la somme des bits est impaire (y compris le bit de parité). **Ex:** 010101, 011100



# Parité simple: efficacité

---

- Toutes les erreurs bit sur un nombre impair de bits sont détectables pour une longueur de données quelconque
- En pratique, seulement la moitié des erreurs bit sont détectées
- La plupart des erreurs bits apparaissent en rafale, donc un nombre pair d'erreurs est aussi probable qu'un nombre impair

# Parité croisée

- Aussi appelé parités verticales et horizontales
- Permettent de construire un code correcteur simple
  - Arranger la séquence de bits en une matrice
  - Calculer une parité par ligne et par colonne

1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	0
1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	0
0	0	1	1	0	0	1	1
1	0	1	1	1	1	1	0

Parités  
horizontales

Parités verticales

# Parité croisée: efficacité

- Permet de corriger toutes les erreurs simples
- Permet de détecter toutes les erreurs sur 2 ou 3 bits
- Permet de détecter toutes les rafales plus courtes que la longueur d'une ligne
- 4 erreurs bits peuvent passer sans être détectées

1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	0
1	1	0	0	1	0	1	0
1	0	0	0	1	1	1	0
0	0	0	1	1	0	1	1
1	0	1	1	1	1	1	0

Parités  
horizontales

Parités verticales

# Codes polynomiaux

---

- Avantages:
  - Permettent de contrôler des mots d'une longueur variable
  - Réalisation simple et efficace en matériel
  - Très bonnes capacités de détection d'erreurs

# Codes polynomiaux

---

- Un bloc de  $m$  bits est vu comme un polynôme  $M(x)$  de degré  $m - 1$  avec des coefficients binaires

$$110001 \rightarrow M(x) = x^5 + x^4 + x^0$$

- Tous les mots de code, représentés par des polynômes  $T(x)$ , sont des multiples d'un polynôme générateur  $G(x)$
- Autrement dit, un mot de code correspond à un polynôme  $T(x)$  de degré  $m - 1 + r$  avec  $T(x) \bmod G(x) = 0$

# Calcul d'un mot de code

---

1. Ajouter  $r$  zéros après le bit de poids faible du bloc. Il contient ainsi  $m + r$  bits, correspondant au polynôme  $x^r M(x)$  appelé  $T(x)$ .
2. Division modulo 2 du polynôme  $T(x)$  par  $G(x)$ .
3. Soustraire modulo 2 le reste de la division de la chaîne de bits correspondant au polynôme  $T(x)$ .
4. Le résultat de cette opération est la trame à transmettre.

# Détection d'erreur

---

- Le récepteur calcule  $\frac{T(x)}{G(x)}$
- Il accepte la trame comme correcte si le reste est égal à 0
- En cas d'erreurs bits, la trame reçue correspond à un polynôme  $T'(x) = T(x) + E(x)$
- Une erreur est détectée si le reste de  $\frac{E(x)}{G(x)}$  n'est pas nul

# Polynômes générateurs normalisés

---

- CRC-12:  $x^{12} + x^{11} + x^3 + x^2 + x^1 + 1$
- CRC-16:  $x^{16} + x^{15} + x^2 + 1$
- CRC-CCITT:  $x^{16} + x^{12} + x^5 + 1$
- CRC-32:  $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$
- Le CRC-32 est utilisé pour Ethernet:
  - Détecte toutes les rafales d'erreurs de 32 bits
  - Probabilité qu'une rafale plus longue ne soit pas détectée:  
$$P = 4.6 \cdot 10^{-10}$$