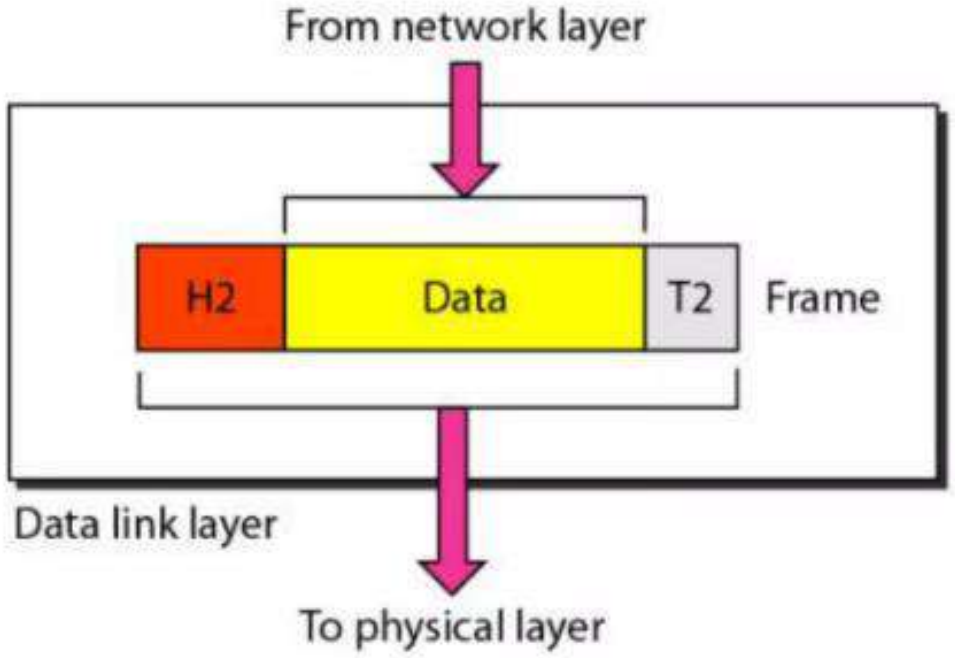
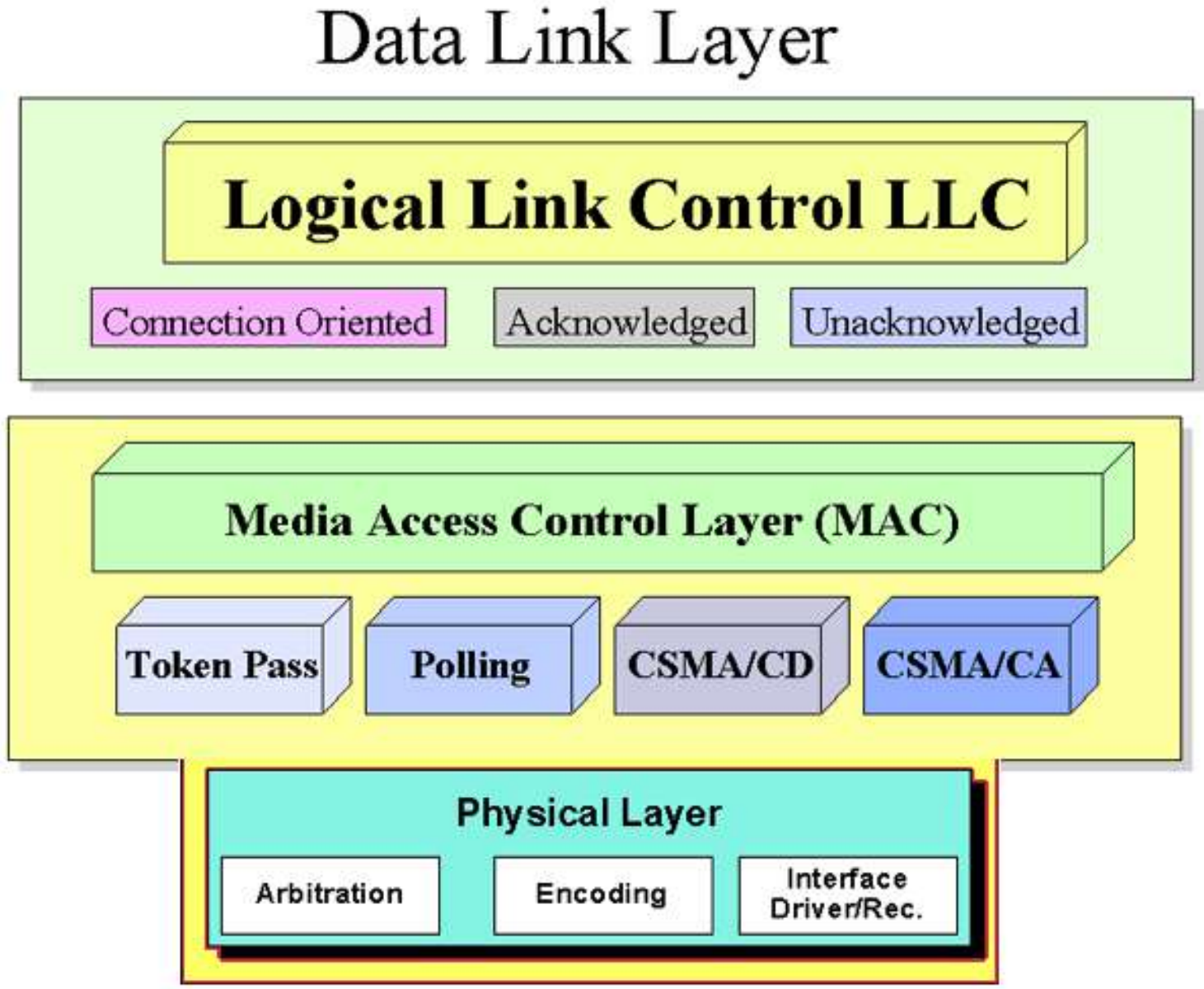
Chapitre 4

Eléments de la Couche Liaison des données



1 - Introduction



1. Introduction

Principales fonctionnalités de gestion de la couche liaison

1. DELIMITATION et IDENTIFICATION des trames (Protocole) : PPP, SLIP, HDLC…

2. GESTION d'une connexion liaison
   établissement et libération d'une liaison de données sur un ou plusieurs circuits physiques.

3. SUPERVISION du fonctionnement de la liaison de données selon.

   - Le mode de transmission (synchrone ou asynchrone)
   - La nature de l'échange (simplex, half-duplex ou full-duplex)
   - Le type de liaison (point-à-point ou multipoint)
   - Le mode de l'échange (hiérarchique ou symétrique)

4. IDENTIFICATION de la source et du destinataire (Adressage)

5. CONTROLE D'ERREURS et CONTROLE DE FLUX (Procédure)

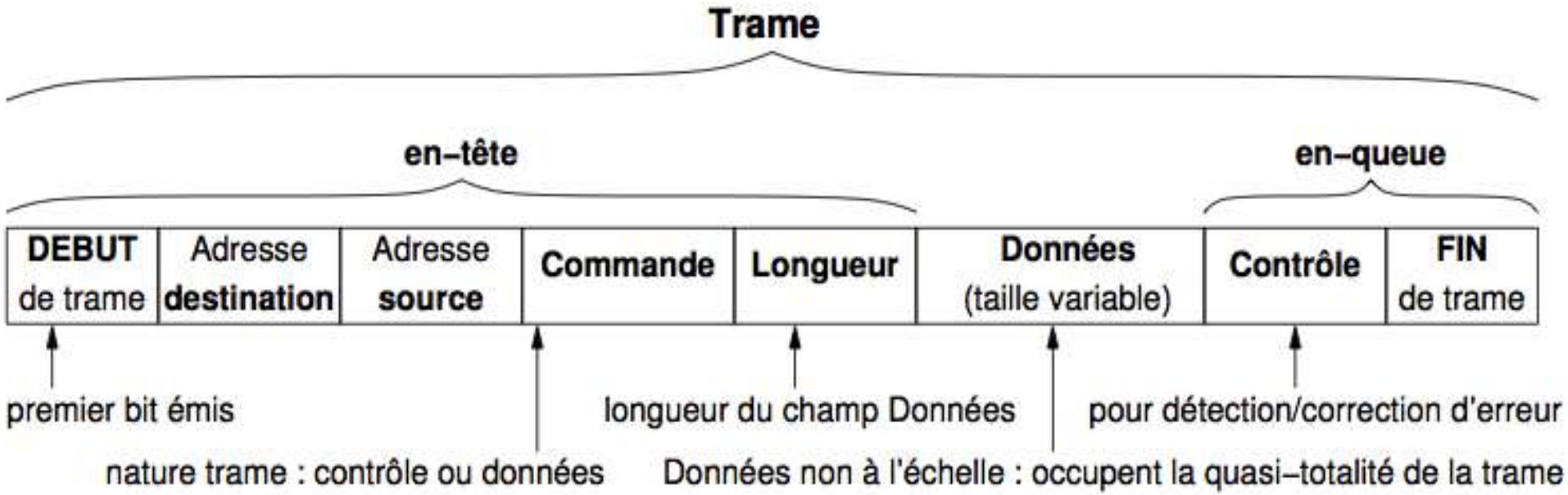
2. Notion de Trame

- suite de bits structurée
- comporte des champs de différentes tailles
- a une longueur minimum et maximum, parfois fixe (ex : cellules ATM de 53 octets)
- un champ peut avoir une taille variable (0 si optionnel)
- le protocole permet de reconnaitre tous les champs

- champ données : peut être absent dans une trame de contrôle
- délimiteurs : début et fin de trame, pas forcément sous la forme de champs
- adresse physique (MAC) de la destination : indispendable sur une liaison multipoints
- adresse physique (MAC) de la source (émetteur)
- champs de correction/détection d'erreurs
- champs de commande : indiquant s'il y a des données, des accusées de réception, demandant un accusé de réception, etc.

2. Notion de Trame

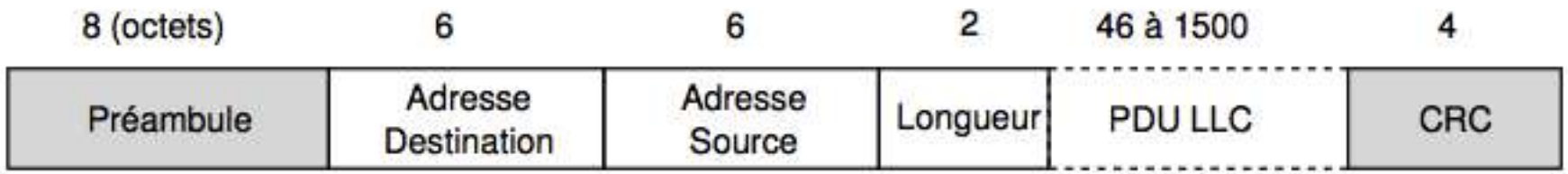
Forme générale d'une Trame



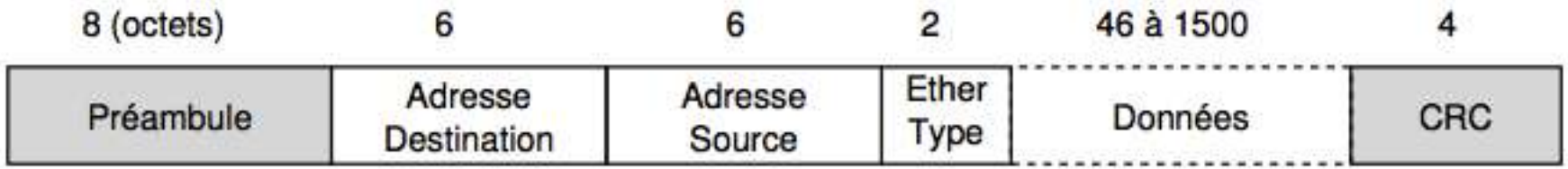
2. Notion de Trame

Exemple de Trames

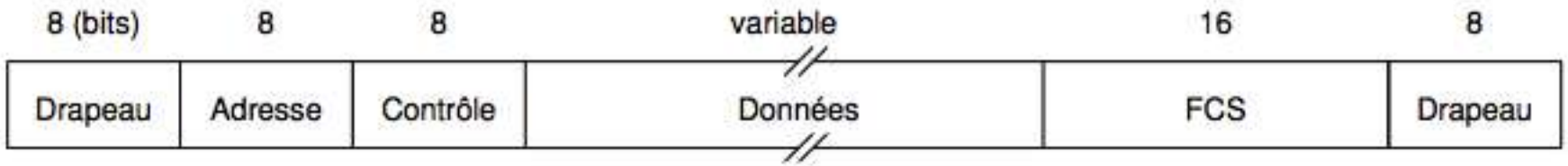
- IEEE 802.3 :



- Ethernet V2 :

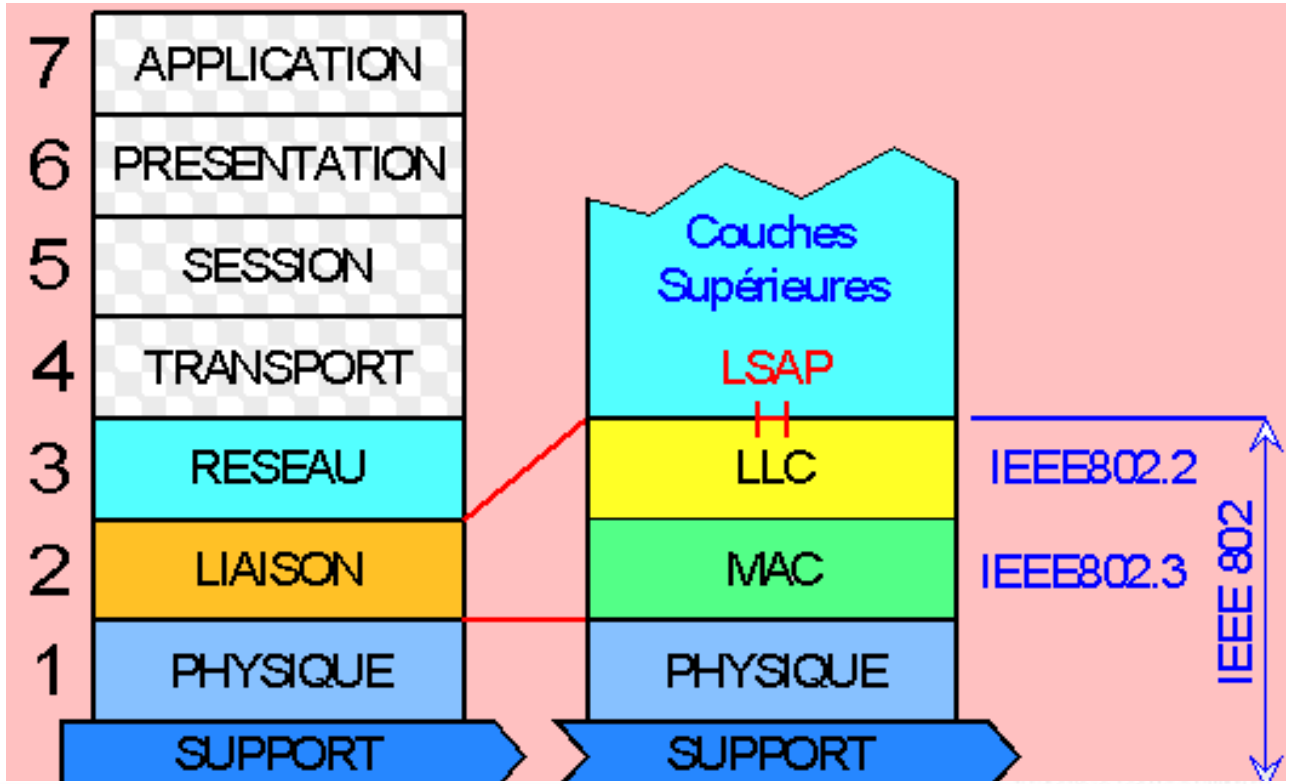


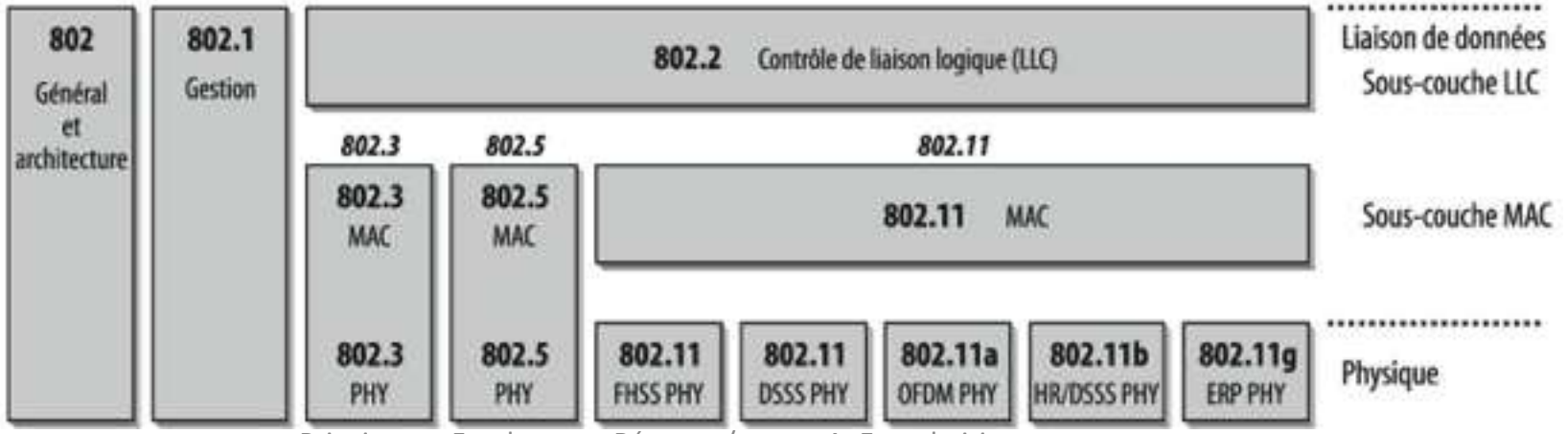
- HDLC (*High-Level Data Link Control*) et LAP-B (*Link Access Procedure Balanced*) :



2. Notion de Trame

IEEE 802.2 : Norme LLC
IEEE 802.3 : Ethernet

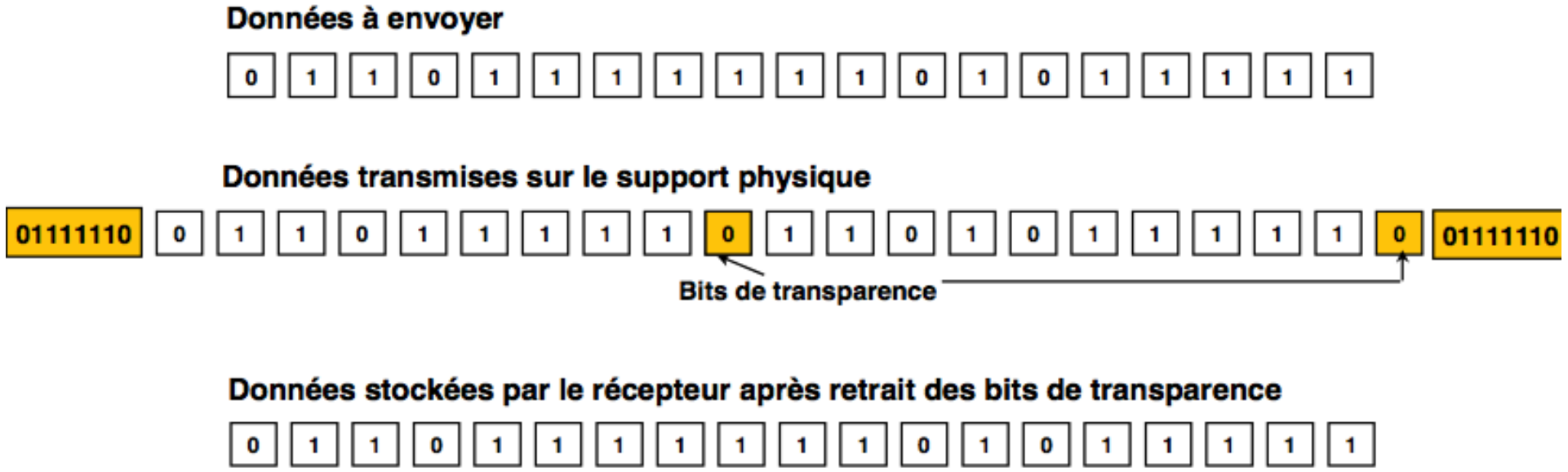




2. Notion de Trame

Principales fonctionnalités de gestion de la couche liaison

1. DELIMITATION et IDENTIFICATION des trames (Protocole) : PPP, SLIP, HDLC…



•Un mécanisme de transparence permet la également de regler les problèmes d'apparition du fanion dans le bloc de données.

• Avantages : (1) indépendant du code utilisé – (2) trame de taille variable et longue

• Exemples : ISO HDLC, PPP

3. Identification / Adressage

Qu’est ce qu’une adresse MAC ?

MAC = Media Access Control

Chaque équipement ethernet → une adresse MAC unique, qui est “gravé" dans le matériel → BIA (Burn-In Address)

Une adresse MAC identifie un équipement (un noeud) àl’intérieur d’un réseau local.

Une adresse MAC est constituée de six octets généralement affiché en valeur hexadécimale : 00-0A-CC-32-FO-FD

NB: L’adresse MAC est écrite sur la carte d’interface réseau du PC.

3. Identification / Adressage

Adressage MAC

3. Identification / Adressage

Adressage MAC

EXEMPLE D'ADRESSE DE VENDEUR (RFC 1700)

Début d'adresse MAC (en hexadécimal)	Vendeur
00:00:0C	Cisco
00:00:1D	Cabletron
08:00:20	Sun
08:00:2B	DEC
08:00:5A	IBM

3. Identification / Adressage

Système d'Adressage MAC

MAC Address (Medium Access Control)

L'adresse de niveau 2 d'un élément de réseau

Format : 6 octets exprimés en hexadécimal séparés par ":" ou "–"

- l'**OUI (Organization Unique Id)** 3 octets
- l'adresse matérielle spécifique **(Product ID)** 3 octets.

Source MAC Address

L'adresse MAC de la station émettrice

Destination MAC Address

L'adresse MAC de la station destinataire

- **Unicast MAC Address** 0A:00:81:2F:42:51

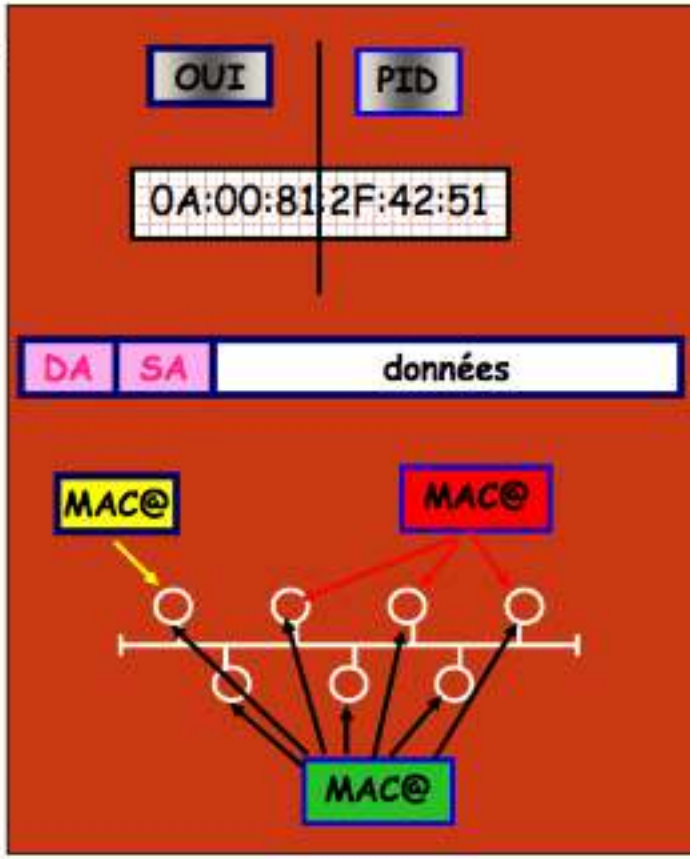
Une adresse MAC désignant une seule station

- **Multicast MAC Address** 01:xx:xx:xx:xx:xx **(premier octet impair)**

Une adresse MAC désignant plusieurs stations (un groupe)

- **Broadcast MAC Address** FF:FF:FF:FF:FF:FF

Adresse MAC de diffusion qui désigne l'ensemble des stations du domaine de collision concerné.



3. Identification / Adressage

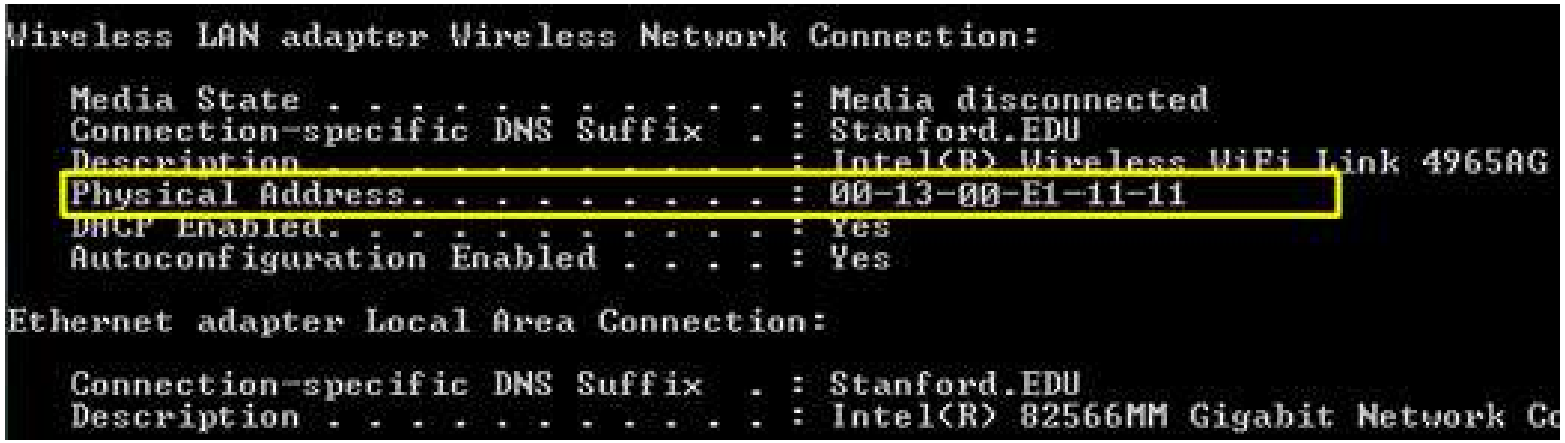
Obtention de l'adresse MAC d'un équipement ?

ipconfig

ifconfig

arp –a

……..



4. Gestion des erreurs

Protocoles de détection et de correction des erreurs

Pourquoi ?

• Des canaux de transmission imparfait entraînant des erreurs lors des échanges de données.

• Probabilité d'erreur sur une ligne téléphonique : $P=10^{-4}$ (cela peut même atteindre 10^{-7}).

⇒ Utilisation de méthodes de détection des erreurs et éventuellement de correction des erreurs.

Principe général

Chaque suite de bits (trame) à transmettre est augmentée par une autre suite de bits dite de redondance ou de contrôle.

Pour chaque suite de k bits transmis, on ajoute r bits. On dit alors que l'on utilise

un code C(n, k) avec n = k + r.

4. Gestion des erreurs

Protocoles de détection et de correction des erreurs

Principe général (suite)

À la réception, on effectue l'opération inverse et les bits ajoutés permettent d'effectuer des contrôles à l'arrivée.

Il existe deux catégories de code :

les codes détecteurs d'erreurs,

- Le mécanisme de détection des erreurs vérifie l'octet ou le bloc de données reçu et, si une erreur est détectée, il signale l'erreur sans la corriger.

les codes correcteurs d'erreurs.

Le protocole de correction des erreurs comprend les mécanismes permettant de détecter l'occurrence des erreurs tout autant que leur correction.

4. Gestion des erreurs

Détection des erreurs

+ Les méthodes classiques de détection des erreurs sont:
  - **contrôle transversal de la parité** (*Vertical Redundancy Checking*, VRC),
  - **contrôle longitudinal de la parité** (*Longitudinal Redundancy Checking*, LRC),
  - le **code de redondance cyclique** (*Cyclic Redundancy Checking*, CRC).

+ Quelque soit la méthode choisie, une information est ajoutée aux données avant leur envoi. Cette information permet au récepteur de vérifier si un changement s'est produit dans les données durant leur transmission.

+ Si le bloc de données contient *k* bits, *k* + *n* bits seront transmis, où *n* correspond au nombre de bits ajoutés pour permettre la vérification. C'est la redondance; elle contient une partie de l'information déjà présente.

+ Les données ne sont pas validées en tant que telles; la vérification se fait sur la correspondance entre les données de la source et celles reçues à la destination. Si les données sont fausses au départ et si elles sont transmises intactes, aucune erreur ne sera signalée.

+ Un code se caractérise par son *efficacité*, son *taux d'erreur brut* et sa *redondance*. On définit l'*efficacité* d'un code par le rapport du nombre d'erreurs détectées au nombre total d'erreurs du message.

4. Gestion des erreurs

Détection des erreurs

Vertical Redundancy Check (VRC) : on effectue un calcul de parité pour chaque caractère. Permet de déterminer si un nombre impair d'erreur s'est produit lors de la transmission d'un caractère.

Longitudinal Redundancy Check (LRC) : on effectue un calcul de parité sur les bits de même rang. S'utilise souvent avec VRC pour renforcer le code.

Exemple : VRC + LRC

lettre	lettre codée	LRC
H	0001001	0
E	1010001	1
L	0011001	1
L	0011001	1
O	1111001	1
VRC	0100001	0

4. Gestion des erreurs

Détection des erreurs

Cyclic Redundancy Check (CRC) : appelé aussi contrôle polynomial, il est très utilisé dans les protocoles modernes car il permet de détecter des erreurs sur plusieurs bits.

Cette méthode utilise les polynômes suivants :

- Polynôme associé à une information $P(x)$: soit $b_n \dots b_0$ la suite de bits correspondant à l'information à transmettre. Alors $P(x) = b_n x^n \dots b_0 x^0$.
- Polynôme générateur $Q(x)$: polynôme caractérisant le contrôle.
- Polynôme reste $R(x)$: correspond à l'information redondante ajoutée en fin de trame. Si r est le degrè de $Q(x)$ alors $R(x)$ est le reste de la division euclidienne de $x^r \times P(x)$ par $Q(x)$:

$$R(x) = (x^r \times P(x)) \mod Q(x) = c_{r-1} x^{r-1} \dots c_0 x^0$$

L'information redondante correspondante à $R(x)$ est $c_{r-1} \dots c_0$.

4. Gestion des erreurs

Détection des erreurs

+ *Validation par le cycle de redondance* : La validation par cycle de redondance (CRC) est une méthode efficace qui permet de valider plusieurs octets à la fois.

Le concept du CRC peut être décrit de la façon suivante : à partir d'une séquence de *k* bits, une séquence de *n* bits supplémentaire est générée de telle sorte que la juxtaposition *k* + *n* forme un nombre divisible sans reste par un autre nombre prédéfini.

+ Exemple: Supposons le nombre prédéfini P = 110101 et la séquence de bits à transmettre *k* = 1010001101. La séquence *n* = 01110 est alors ajoutée de façon à produire *k* + *n* = **1010001101**01110.

La séquence *k* + *n* est transmise et reçue par le récepteur qui la divise par P. Si le résultat donne un reste nul, il n'y a pas d'erreur de transmission, dans le cas contraire, une erreur est signalée. Lorsque les cinq derniers bits sont enlevés, la valeur *k* = 1010001101 est retrouvée.

+ Il existe plusieurs méthodes pour construire un CRC :

- **la méthode arithmétique** : application de la fonction **modulo 2**;
- **la méthode polynomiale** : application des propriétés d'une fonction polynomiale dont les coefficients sont fonction de la valeur binaire à transmettre.

4- Gestion des erreurs

Détection des erreurs

- **Émission d'un mot :**
  - On choisit un polynôme générateur puis on le transforme en un mot binaire.
  - Exemple : avec le polynôme générateur $x^4 + x^2 + x$, on obtient `10110`.
  - On ajoute m zéros au mot binaire à transmettre où m 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 `111001110000`.
  - On va ajouter 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.
  - On effectue donc une division euclidienne dans laquelle on ne tient pas compte du quotient.

4- Gestion des erreurs

Détection des erreurs

- Exemple d'émission d'un mot :

```
1 1 1 0 0 1 1 1 0 0 0 0
1 0 1 1 0
---------------------
0 1 0 1 0 1
  1 0 1 1 0
---------------------
  0 0 0 1 1 1 1 0
      1 0 1 1 0
---------------------
      0 1 0 0 0 0
        1 0 1 1 0
---------------------
        0 0 1 1 0 0 0
            1 0 1 1 0
---------------------
            0 1 1 1 0
```

- Le CRC est donc 1110 et le mot à transmettre 11100111 1110.

4- Gestion des erreurs

Détection des erreurs

- **Réception d'un mot :**

```
1 1 1 0 0 1 1 1 1 1 1 0
1 0 1 1 0
---------------------------
0 1 0 1 0 1
  1 0 1 1 0
---------------------------
  0 0 0 1 1 1 1 1
      1 0 1 1 0
---------------------------
      0 1 0 0 1 1
        1 0 1 1 0
---------------------------
        0 0 1 0 1 1 0
            1 0 1 1 0
---------------------------
            0 0 0 0 0
```

- Le reste de la division est nulle, il n'y a donc pas d'erreur.

4- Gestion des erreurs

Détection des erreurs

- **Exercices :**

  On utilisera le polynôme générateur $x^4 + x^2 + x$.

  1. On souhaite transmettre le message suivant :`1111011101`, quel sera le CRC à ajouter ?
  2. Même question avec le mot `1100010101`.
  3. Je viens de recevoir les messages suivants : `111100010101`0, `1100010101010110`, sont-ils corrects ?

4- Gestion des erreurs

Détection des erreurs

- Correction : quel CRC à ajouter avant d'émettre le message 1111011101 ?

```
1 1 1 1 0 1 1 1 0 1 0 0 0 0
1 0 1 1 0
---------------------------
0 1 0 0 0 1
  1 0 1 1 0
---------------------------
  0 0 1 1 1 1 1
    1 0 1 1 0
---------------------------
    0 1 0 0 1 0
      1 0 1 1 0
---------------------------
      0 0 1 0 0 1 0
        1 0 1 1 0
---------------------------
        0 0 1 0 0 0 0
          1 0 1 1 0
---------------------------
          0 0 1 1 0 0
```

- Le CRC est donc 1100 et le mot à transmettre 1111011101 1100

4- Gestion des erreurs

Détection des erreurs

- Correction : quel CRC à ajouter avant d'émettre le message 1111011101 ?

x^{13}	x^{12}	x^{11}	x^{10}		x^8	x^7	x^6		x^4			x^4	$+x^2$	$+x$
x^{13}		x^{11}	x^{10}											
	x^{12}				x^8	x^7	x^6		x^4			x^9	$+x^8$	$+x^6$
	x^{12}		x^{10}	x^9								$+x^5$	$+x^3$	$+x^2$
			x^{10}	x^9	x^8	x^7	x^6		x^4			$+x$		
			x^{10}		x^8	x^7								
				x^9			x^6		x^4					
				x^9		x^7	x^6							
						x^7			x^4					
						x^7		x^5	x^4					
								x^5						
								x^5		x^3	x^2			
										x^3	x^2			

4- Gestion des erreurs

Détection des erreurs

- Correction : quel CRC à ajouter avant d'émettre le message 1100010101 ?

```
1 1 0 0 0 1 0 1 0 1 0 0 0 0
1 0 1 1 0
---------
0 1 1 1 0 1
  1 0 1 1 0
---------
  0 1 0 1 1 0
    1 0 1 1 0
-------------
    0 0 0 0 0 1 0 1 0 0
        1 0 1 1 0
-----------------
        0 0 0 1 0 0 0
```

- Le CRC est donc 1000 et le mot à transmettre 1100010101 1000.

4- Gestion des erreurs

Détection des erreurs

- **Réception d'un mot :**

```
1 1 1 0 0 1 1 1 1 1 1 0
1 0 1 1 0
---------------------------
0 1 0 1 0 1
  1 0 1 1 0
---------------------------
  0 0 0 1 1 1 1 1
      1 0 1 1 0
---------------------------
      0 1 0 0 1 1
        1 0 1 1 0
---------------------------
        0 0 1 0 1 1 0
            1 0 1 1 0
---------------------------
            0 0 0 0 0
```

- Le reste de la division est nulle, il n'y a donc pas d'erreur.

4- Gestion des erreurs

Détection des erreurs

- Correction : le message reçu 1111000101010 est-il correct ?

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

- Le reste est nul ⇒ il n'y a pas d'erreur dans le mot transmis.

4- Gestion des erreurs

Détection des erreurs

- Correction : le message reçu 11000101010110 est-il correct ?

```
1 1 0 0 0 1 0 1 0 1 0 1 1 0
1 0 1 1 0
---------------------------
0 1 1 1 0 1
  1 0 1 1 0
---------------------------
  0 1 0 1 1 0
    1 0 1 1 0
---------------------------
    0 0 0 0 0 1 0 1 0 1
          1 0 1 1 0
---------------------------
          0 0 0 1 1 1 0
```

- Le reste est 1110 ⇒ il y a une erreur dans le mot transmis.

4- Gestion des erreurs

Détection des erreurs

Exemple : Soit 1000001110000100 l'information à transmettre. Alors le polynôme correspondant est :

$$P(x) = x^{15} + x^9 + x^8 + x^7 + x^2$$

Soit le polynôme de contrôle de degrès 12 suivant :

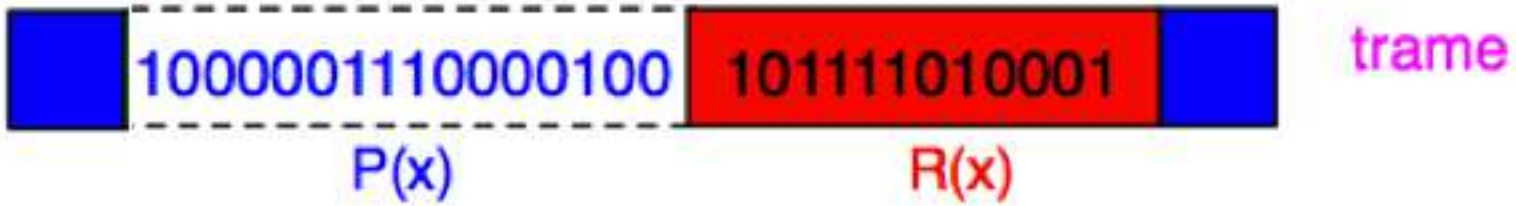
$$Q(x) = x^{12} + x^{11} + x^3 + x^2 + x + 1$$

La division de $x^{12} \times P(x)$ par $Q(x)$ donne le polynôme reste :

$$R(x) = (x^{12} \times P(x)) \mod Q(x) = x^{11} + x^9 + x^8 + x^7 + x^6 + x^4 + 1$$

L'information redondante à ajouter en fin de trame est donc : 101111010001.

La trame envoyée contient donc : 1000001110000100 101111010001.



4- Gestion des erreurs

Détection des erreurs

- Exemples de codes polynômiaux :

  (i) L'avis V41 du CCITT conseille l'utilisation de codes polynômiaux (de longueurs n = 260, 500, 980 ou 3860 bits) avec le polynôme générateur $G(x) = x^{16} + x^{12} + x^5 + 1$.

  (ii) Le polynôme CRC-16 est utilisé par le protocole HDLC :

  . $G(x) = x^{16} + x^{15} + x^2 + 1$.

  (iii) Le polynôme suivant est utilisé par Ethernet :

  . $G(x) = x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+1$.

4- Gestion des erreurs

Protocoles de correction des erreurs :

+ Pour pouvoir corriger les erreurs, il faut être capable de distinguer les différents caractères qui sont émis, même lorsqu'ils ont un bit erroné.

+ Il existe principalement deux types de correction d'erreur :

- correction d'erreur sans circuit de retour (*Forward Error Correction*, FEC).
- détecteur d'erreurs avec demande de répétition (*Automatic Repeat Request*, ARQ).

+ Dans la correction d'erreur sans circuit de retour (FEC), on utilise un type de codage qui permet à la réception de corriger un certain nombre d'erreurs. Pour réaliser cela on utilise du logiciel ou du matériel installés aux deux extrémités.

- Dans la détection d'erreurs avec demande de répétition (ARQ), lorsqu'une erreur est détectée, le récepteur signale l'erreur au transmetteur en utilisant la voie de retour (*backward channel)* afin qu'il lui retransmette de nouveau le bloc ou paquet d'information erroné. Le récepteur réitère sa demande de retransmission tant qu'il n'a pas reçu une séquence de bits sans erreur. La retransmission peut être soit : une retransmission avec arrêt et attente (ARQ-ACK), une retransmission continue (ARQ-NAK), une retransmission à répétition sélective.

4- Gestion des erreurs

- Le *taux d'erreur brut* d'un code (probabilité d'occurrence de message faux) est le nombre total de messages erronés par rapport au nombre total de messages transmis.

- Certains messages faux peuvent échapper aux mécanismes de contrôle. Pour tenir compte de ces faux messages, on définit un *taux d'erreur effectif*: $q = T(1-e)$, où T désigne la probabilité d'occurrence des faux messages (taux d'erreur brut) et $(1-e)$ la probabilité qu'un faux message reste non détecté (pourcentage d'erreurs non détectées).

- La *redondance* d'un code est définie comme le rapport entre le nombre de bits de contrôle au nombre de bits de l'information utile par mot de code : $R = (n-m) / m$, où n désigne le nombre total de bits et m le nombre de bits constituant l'information utile.

- On peut déterminer la *distance* en calculant le nombre de bits différents entre deux mots de code. Pour cela on effectue un OU EXCLUSIF et on compte le nombre de 1 du résultat. Exemple: (10110011) + (11100011) = 01010000; donc 10110011 et 11100011 diffèrent de 2 bits.

- Le nombre de bits de différence entre deux mots de code s'appelle la *distance de Hamming*. La robustesse d'un code correcteur ou détecteur dépend de sa distance de Hamming du code complet.