



# Rappel 1

OSI

TCP/IP

7	Application <i>Application</i>	échanges de données d'application (selon l'application)	Application
6	Présentation <i>Presentation</i>	mise en forme des données pour la transmission	
5	Session <i>Session</i>	synchronisation de processus	
4	Transport <i>Transport</i>	transfert de blocs d'octets entre processus	Transport
3	Réseau <i>Network</i>	transfert de blocs d'octets entre systèmes (pas forcément raccordés au même médium)	Internet
2	Liaison de données <i>Data Link</i>	transfert fiable de blocs d'octets entre systèmes raccordés au même médium	Accès Réseau
1	Physique <i>Physical</i>	transfert de bits entre systèmes raccordés au même médium	

# Pour info

---

## ✦ Couche physique

### ◆ Transfert des données

- Câble électrique
- Fibre optique
- Ondes

### ◆ Inconvénient

- Atténuation
- Taux d'erreur

La couche physique permet d'envoyer physiquement des bits entre une source et un destinataire qui sont reliés entre eux, mais avec possibilité d'erreurs.

# Plan du chapitre

---

## Transfert de données sur un réseau local

- 1) La couche 2 (liaison de données)
  - 1) La couche MAC
  - 2) La couche LLC
- 2) Codes détecteurs/correcteurs d'erreurs
- 3) Protocole filaire
  - Ethernet
  - Le switch
- 4) Protocoles non filaires
  - 802.15,802.16, 802.11,...

# La couche MAC - Généralité (1)

But :

**Transmission de bits entre systèmes raccordés au même réseau local (même médium)**

Liaison de données	Logical Link Control ( 802.2 )					LLC
	CSMA/CD (802.3)	Bus (802.4)	Jeton (802.5)	DQDB (802.6)	Sans fil (802.11)	
Physique	Paire torsadée	Fibre optique		Ondes		

....

MAC : Medium Access Control

En général, tout signal émis par l'un des systèmes raccordés peut être entendu par tous les autres

Équipement : *switch*, HUB, *point d'accès*

# Couche 2 - Généralité

---

## Les fonctionnalités de la couche 2

- Gestion des données
  - Problème : Repérage des différentes trames sur le support physique (suite de bits)



utilisation d'un fanion

- Détection et correction des erreurs
  - Numérotation des trames
  - Utilisation de code détecteur/correcteur d'erreurs
- Régulation du trafic
  - utilisation d'acquittement
  - Emission entre une source et un destinataire

- **Trouver le destinataire**

Appellation unique de chaque entité MAC sur le médium

Identifie le système sur le médium

Appellation physique → **adresse MAC**

# L'adresse MAC

---

## ✦ Identification du niveau 2 :

### ✦ Unicité sur le médium

### ✦ **adresse sur 6 octets**

- 3 premiers octets :OUI (Organizationally Unique Identifier)
  - ◆ Concerne entreprise (oui.txt)
- 3 octets suivants : un numéro unique pour l'entreprise
- Existe des @mac de groupe ou de broadcast
- **@MAC permet d'identifier d'une manière unique un ordinateur sur un réseau local**

Test : ipconfig /all dans un shell windows : cmd

# La couche MAC - Généralité (2)

Une entité-MAC par système

- ◆ reçoit les demandes d'émission (de la couche supérieure)
- ◆ décide quand émettre (**méthode d'accès au médium**)
- ◆ écoute ce qui est transmis, et décide de recevoir ou non

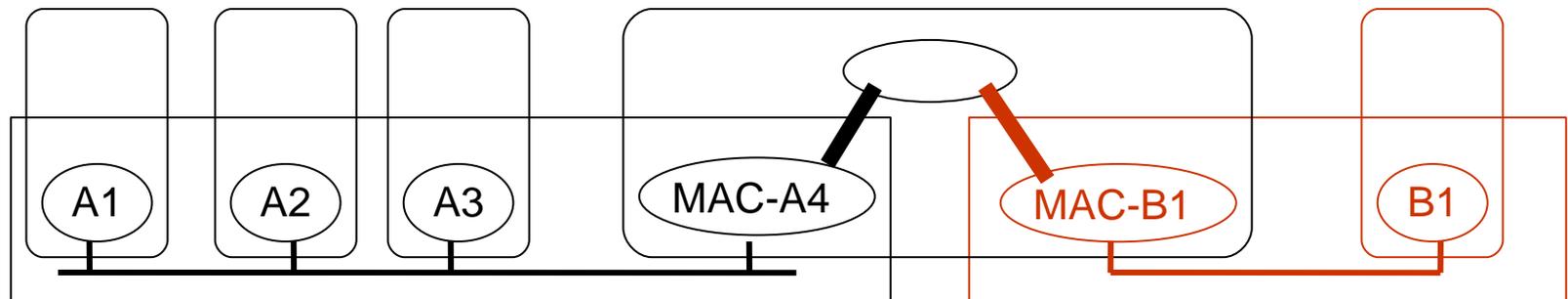
Si plusieurs couches MAC

Pas de communication entre deux couches MAC

*Sauf si*

un système est raccordé aux 2 supports et contient :

- une entité MAC pour chaque médium
- une entité-utilisateur de couches MAC liée à chacune des deux couches MAC ➡ **Routeur (utilisation couche niveau 3)**





# Traitement des erreurs

# Le traitement des erreurs (1)

- ◆ -  $T_e$  = taux d'erreur
- $T_e = \text{Nb bits erronés} / \text{Nb bits transmis pendant une période d'observation}$   
 $10^{-9} < T_e < 10^{-3}$
- En général, les erreurs se produisent **en rafale**



## Principes

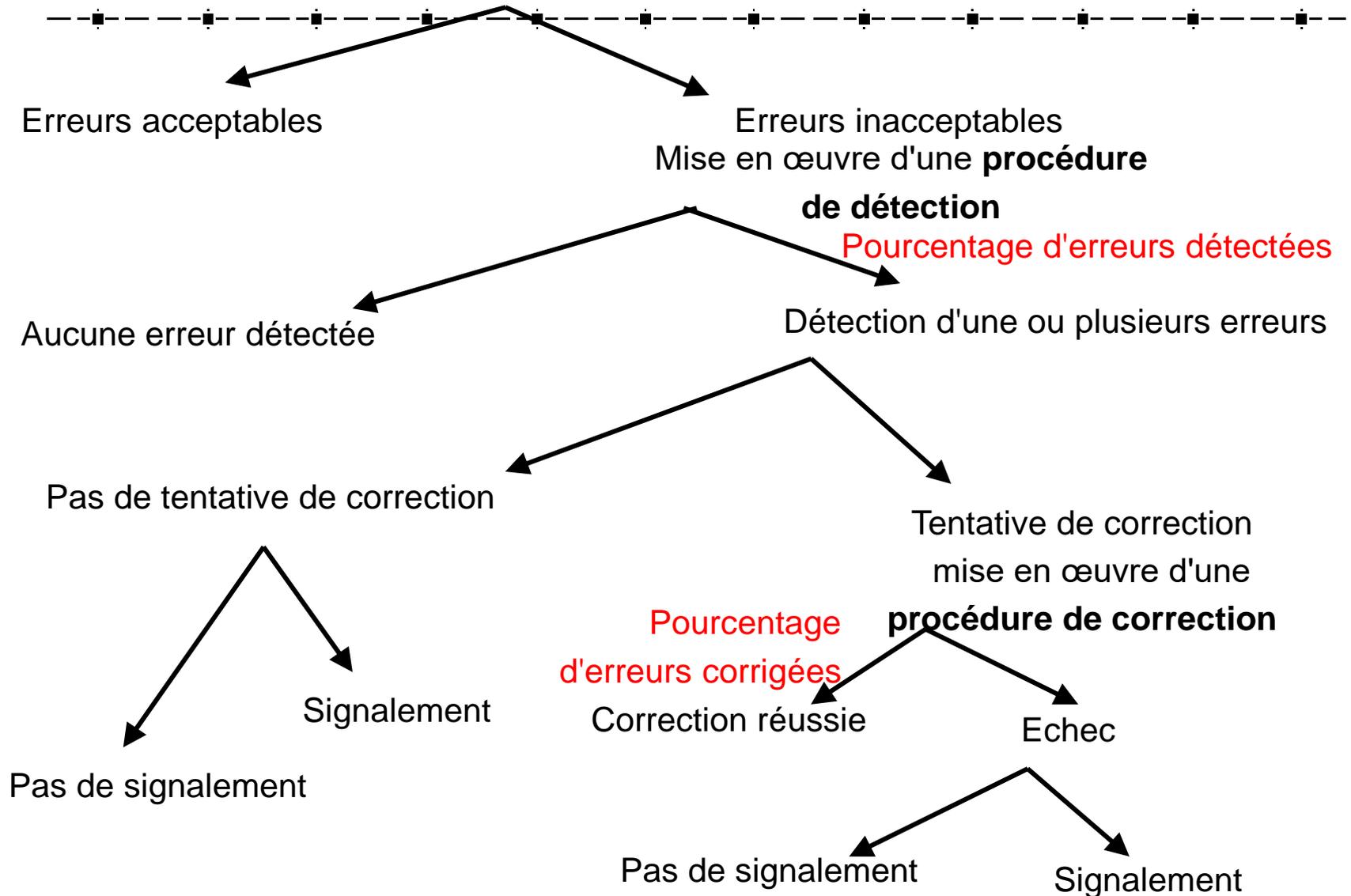
Un **vocabulaire** commun à l'émetteur et au récepteur

L'émetteur n'émet que des **mots** du vocabulaire

Le récepteur reçoit une chaîne de bits qui :

- ◆ n'est pas un mot du vocabulaire → détection d'une ou plusieurs erreurs
- ◆ est un mot du vocabulaire → Pas d'erreur de transmission ou plusieurs erreurs qui se compensent

# Stratégie de traitement d'une erreur



# Le traitement des erreurs (2)

## ✦ Que faire avec ses erreurs

- ◆ Echo, répétition, **redondance** ?

On envoie donc *m bits de données et r bits de redondance*  
*lg message = m+r*

## ◆ Deux stratégies

- Suffisamment de redondance pour que le récepteur puisse reconstituer les données
  - ◆ Utilise des **codes correcteurs d'erreur**
- Juste de qu'il faut en redondance pour pouvoir **détection** les erreurs de la transmission et **demande une retransmission**
  - ◆ Utilise des **codes détecteurs d'erreurs**

# Distance de Hamming

- ✦ C'est le critère qui permet d'évaluer le pouvoir détecteur d'un code ainsi que son pouvoir correcteur.
- ✦ Distance de Hamming **entre 2 mots** (noté  $d_h$ )  
= nbre de positions qui ont des valeurs distinctes.  
EX :  $d_h ( 110011, 101010 ) = 3$   
(Astuce = nbre de 1 du OU exclusif)
- ✦ Distance de Hamming **d'un code C** (noté  $D_H(C)$ )  
= le minimum des distances de Hamming entre 2 mots du code  
EX :  $D_H ( \{110, 101, 011\} ) = 2$   
 $D_H ( \{0011, 0101, 1001, 0110, 1010, 1100\} ) = 2$

# Pouvoir détecteur d'un code

✦ Définition : On parle **d'erreur d'ordre  $k$**  lorsqu'un mot émis  $u$  diffère par  $k$  bits du mot reçu  $v$ , ie  $d_h(u, v) = k$ .

✦ Pour **détecter une erreur d'ordre 1**, quelle doit être la distance de Hamming du code ?

Réponse :  $D_H(C) = 2$

en effet, dans ce cas, 1 erreur simple ne peut pas changer un mot du code en un autre mot du code.

✦ Même question pour une **erreur d'ordre  $k$**  ?

Réponse :  $D_H(C) = k+1$

# Pouvoir correcteur d'un code

- ✦ Pour pouvoir **corriger une erreur d'ordre 1**, une distance de Hamming  $D_H(C) = 2$  est-elle suffisante ?  
Réponse : **non... il faut  $D_H(C) = 3$**

En effet, si  $D_H(C) = 2$ , 1 erreur simple peut nous produire un mot exactement « au milieu » de 2 mots du code, à distance 1 de chacun !

si  $D_H(C) = 3$ , 1 erreur simple produit un mot qui reste le plus proche du mot transmis : on peut donc le retrouver !

Exemple : A=000, B=011, C=101

- ✦ Pour corriger une **erreur d'ordre k**,  
il faut  **$D_H(C) = 2k+1$** .

# Code détecteur d'erreurs (1)

## Contrôle de parité

- ◆ **code VCR ( Vertical Redundancy Check)**
  - Ajoute 1 bit
  - Nb de 1 pair dans les données envoyées
  - Distance de Hamming : 2
    - ◆ **Code détecteur**
  
- ◆ **code LCR ( Longitudinal Redundancy Check)**
  - Parité verticale et horizontale
  - Distance de Hamming : 3
    - ◆ **Code détecteur/correcteur d'erreur**

# Code détecteur d'erreurs (2)

## Codes Polynomiaux

- ◆ CRC : Cyclic Redundancy Check
- ◆ FCS : Frame Control Check
- ◆ Toutes séquence de n bits peut être représentée par un polynôme à coefficients binaires
- ◆ On travaille en modulo 2

$G(x)$  : polynôme générateur de degré  $r$

$M(x)$  : message à encoder

$$1100101 \quad \leftrightarrow \quad 1*x^6 + 1*x^5 + 0*x^4 + 0*x^3 + 1*x^2 + 0*x^1 + 1*x^0$$
$$\quad \quad \quad \leftrightarrow \quad x^6 + x^5 + x^2 + 1$$

ex :  $G(x) = x^{16} + x^{12} + x^5 + 1$

CRC-CCITT

$G(x) = x^8 + x^2 + x + 1$

ATM

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

CRC-32-ethernet

# Code détecteur d'erreurs (3)

## Méthode

- On divise le polynôme  $M(x) \cdot x^r$  par  $G(x)$  et on obtient le reste  $R(x)$

$$M(x) \cdot x^r = G(x) \cdot Q(x) + R(x)$$

- On envoie la séquence de bits de longueur  $n=m+r$  tel que :

$$N(x) = M(x) \cdot x^r + R(x)$$

- $N(x)$  est multiple de  $G(x)$  car :

$$\begin{aligned} N(x) &= M(x) \cdot x^r + R(x) = G(x) \cdot Q(x) + R(x) + R(x) \\ &= G(x) \cdot Q(x) \end{aligned}$$

- On décode en faisant la division, **le reste doit être nul**

**Mathématique en base 2 :**

Addition

+	0	1
0	0	1
1	1	0

Soustraction

-	0	1
0	0	1
1	1	0

# Exemple

✦  $G(x) = x^2 + 1$  ,  $\rightarrow 101$

✦ Message à envoyer : 1100 , c'est à dire  $x^3 + x^2$

Division de  $(x^3 + x^2) * x^2 = x^5 + x^4$  par le polynôme  $G(x)$

$$\begin{array}{r|l} 110000 & 101 \\ 011 & \text{xxxx} \\ 011 & \\ 011 & \\ 011 & \end{array}$$

Reste 11

Message envoyé : 110011

$$\begin{array}{r|l} \text{Vérification : } 110011 & 101 \\ 011 & \\ 011 & \\ 010 & \\ 000 & \end{array}$$





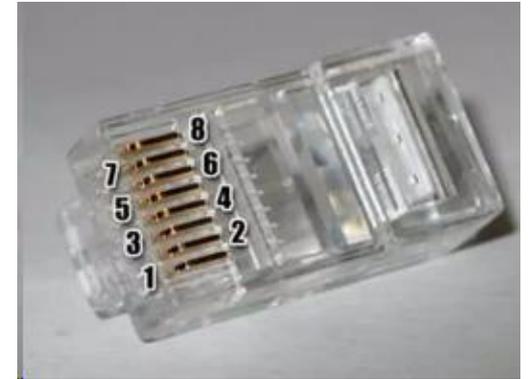
## Protocole de niveau 2

# IEEE 802.3..., Ethernet -II

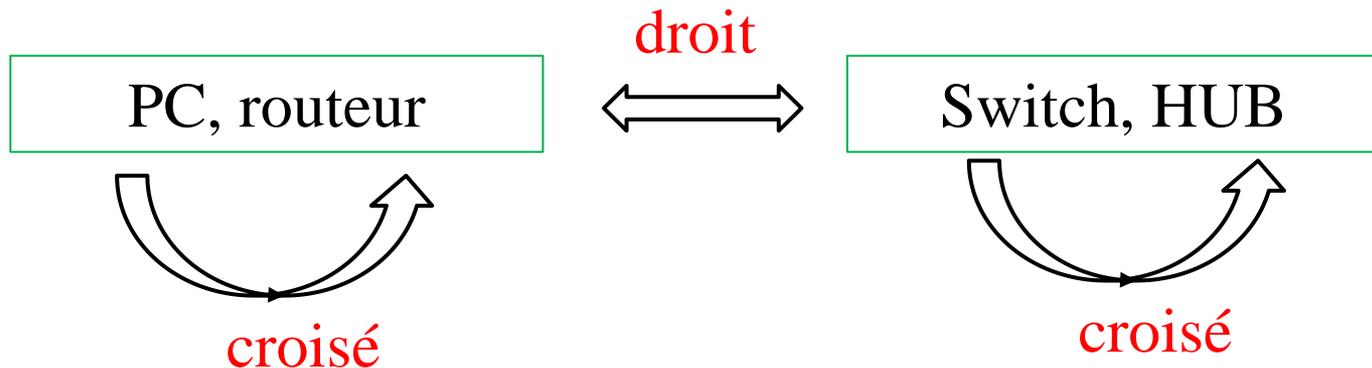
❖ Protocole ethernet date de 1973, inventé par Xeros Parc

❖ Connecteur RJ45

- ❖ 8 broches de connexions électriques
- ❖ Débit 10/100 Mb/s, broche 1-2 et 3-6
- ❖ Débit 1Gb/s, 8 broches utilisées



❖ Câble droit ou câble croisé



Actuellement, le décroisement peut se faire au niveau logiciel grâce à la détection de la polarité du signal

# IEEE 802.3..., Ethernet -II

---

- ◆ Technique : CSMA/CD  
Carrier Sense Multiple Access / Collision Detection
- ◆ Topologie : Bus et maintenant maillage
- ◆ Méthode d'accès : par compétition

➔ Sur un bus :  
(écoute du médium : si libre, émission  
sinon attente)  
Réémission après un temps d'attente si collision -> **half-duplex**

➔ **Topologie maillée** :  
**full-duplex** → aucune collision possible

- ◆ Débit : 10 Mb/s, 100 Mb/s, 1 Gb/s, 10Gb/s

# IEEE 802.3..., Ethernet -II

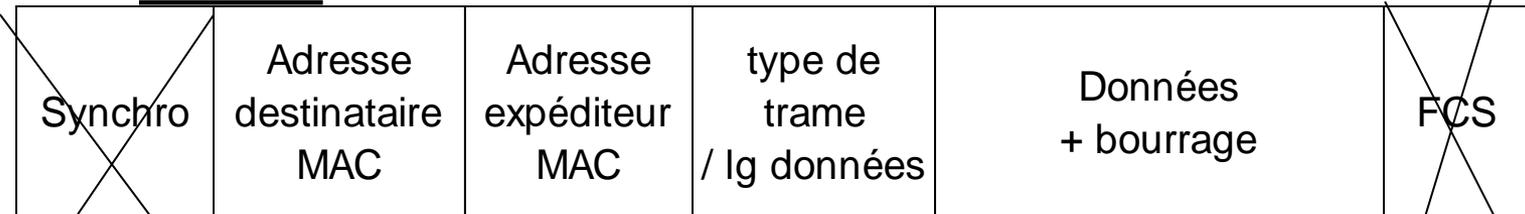
- ◆ Equipement :

Half-duplex -> HUB

Full-duplex → switch (commutateur)



- ◆ Trame :

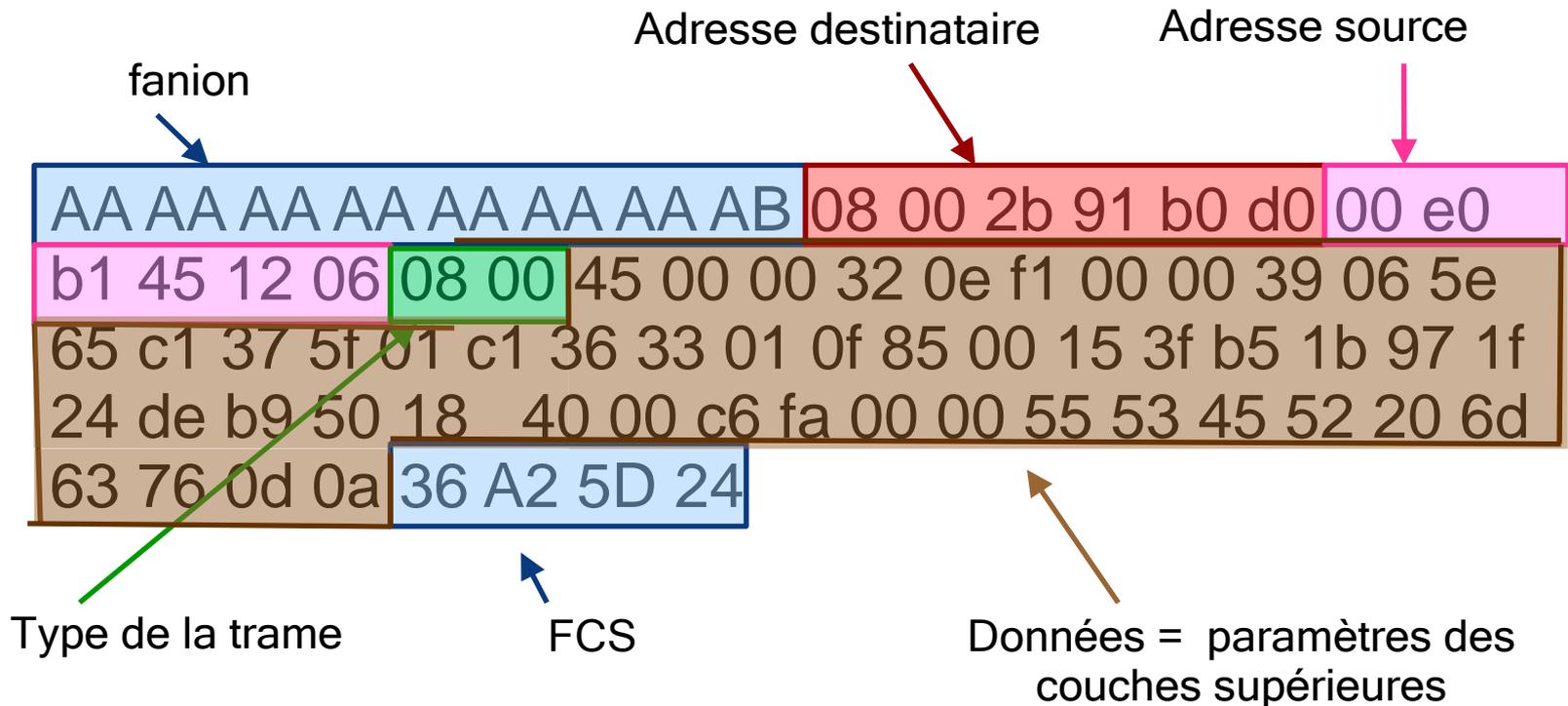


- ◆ Ethernet 802.3: lg données < 0x05DC

**Ethernet -II** : protocole de niveau 3 encapsulé > 0x05DC

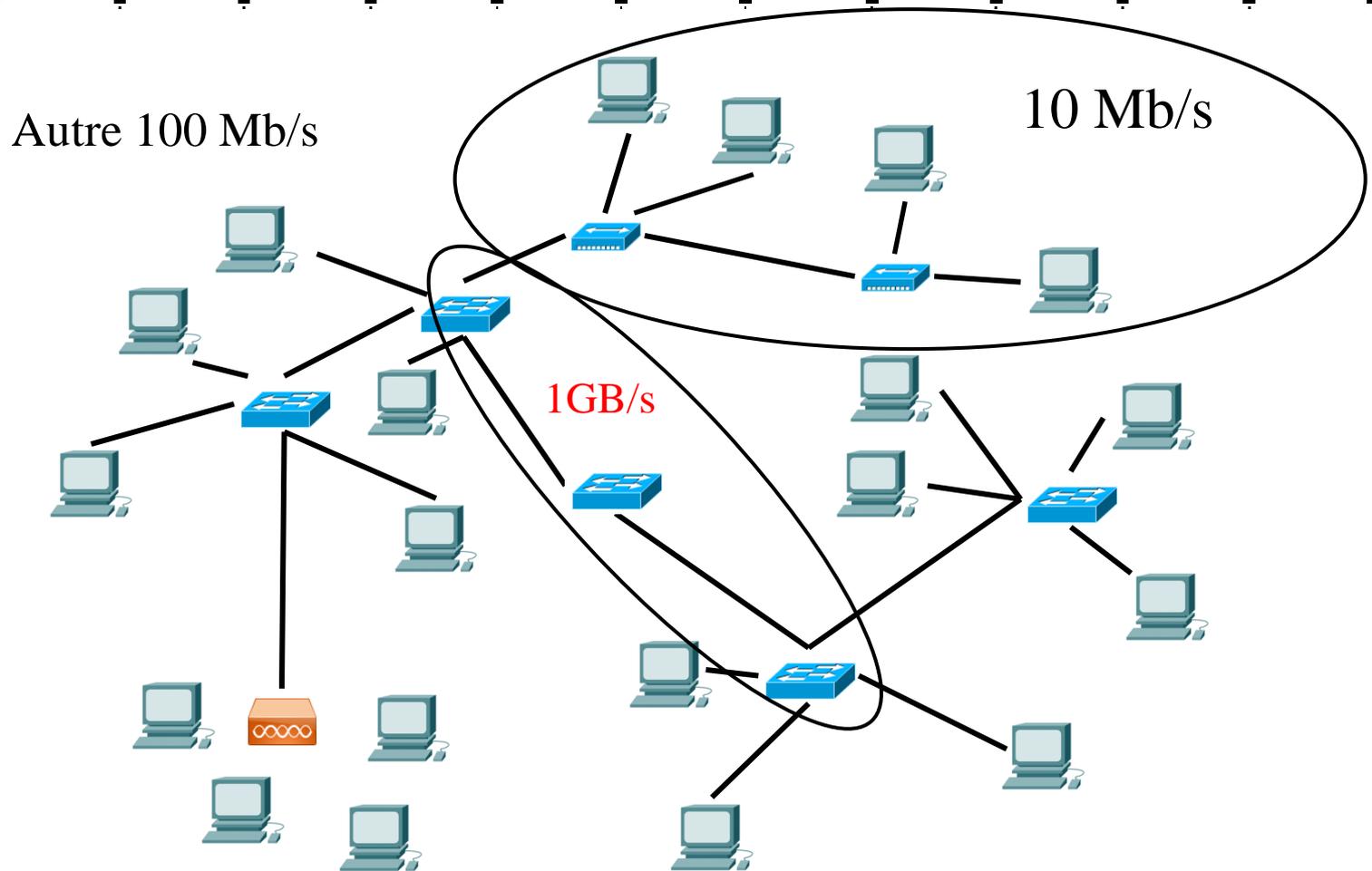
# Exemple ethernet -II

Exemple d'une trame en hexadécimal :

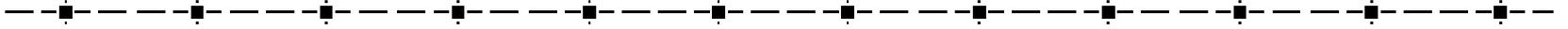


**Le fanion et le contrôle d'erreur (FCS) ne sont jamais représentés sauf ici.**

# LAN



Un LAN : ensemble d'équipements reliés entre eux par des switches, HUB ou Point d'accès.



Equipement filaire de niveau 2

SWITCH

HUB

# SWITCH / HUB

---



HUB



SWITCH

# Un switch

- 
- ✦ Permet la connexion point à point (full-duplex)
  - ✦ *Apprentissage des @MAC*
    - ◆ Table d'adresse mac
  - ✦ Laisse passer les trames du niveau 2 en broadcast (domaine de diffusion)
  - ✦ **Permet de sécuriser le réseau car les données ne sont transmises que sur le lien destinataire**
  - ✦ Pb : possibilité de saturer la table des @MAC → Hub
  - ✦ Vol d'@MAC
  - ✦ **Sécurisation des ports du switch**