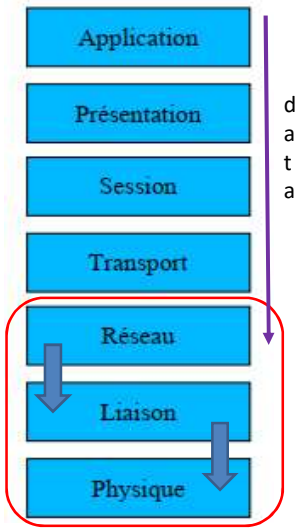
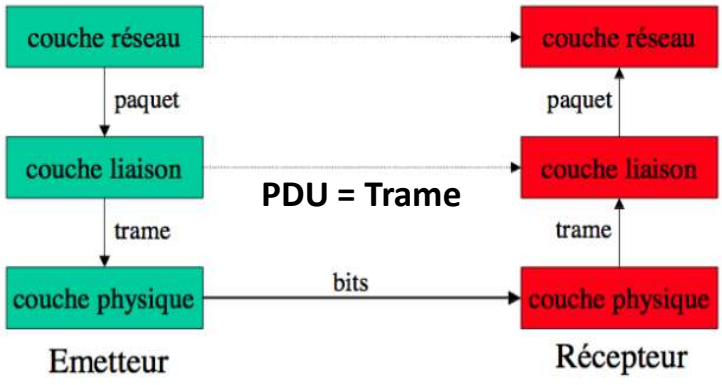
CHAPITRE III

COUCHE LIAISON DE DONNÉES

MODE DE FONCTIONNEMENT DANS LE MODÈLE EN COUCHES





Introduction

La couche liaison de données permet de:

- Détermine la manière dont les bits tenant de la couche physique sont regroupés en trames
- Traiter les erreurs de transmission
- Effectue un contrôle de flux pour réguler le volume de données échangées
- Définit le mode d'accès au réseau
- Définit le protocole de communication (HDLC, PPP, ETHERNET, ….)

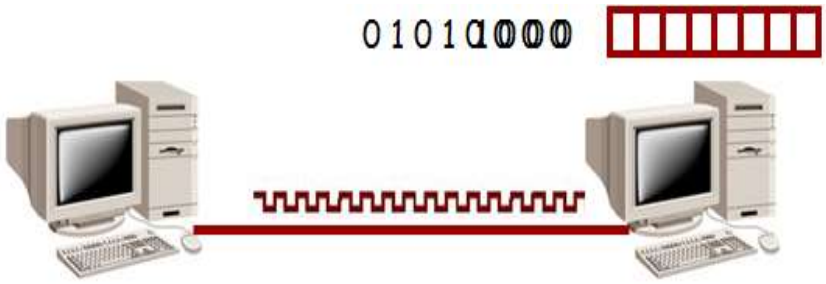
Adressage

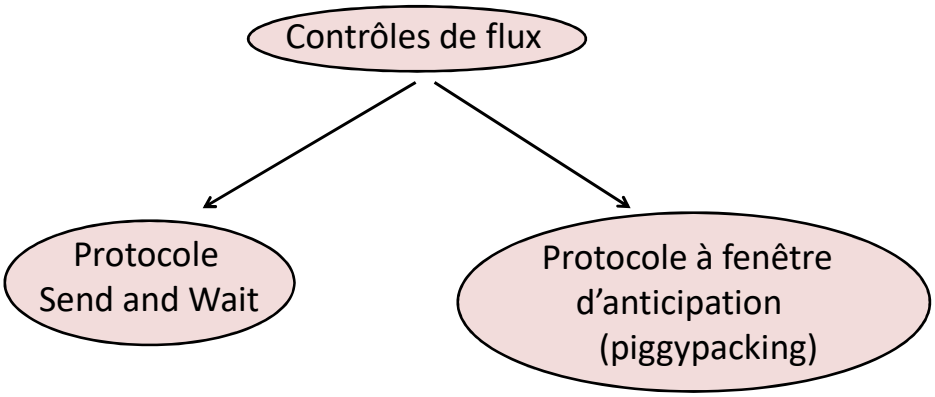
1. Adressage

- Permet d'identifier les machines sur le réseau au niveau de la couche liaison
- Permet d'identifier une liaison: Source-Destination
- Exemple: Ethernet, adresse MAC (Physique) 48:52:25:36:FA:B1

2. Contrôle de flux

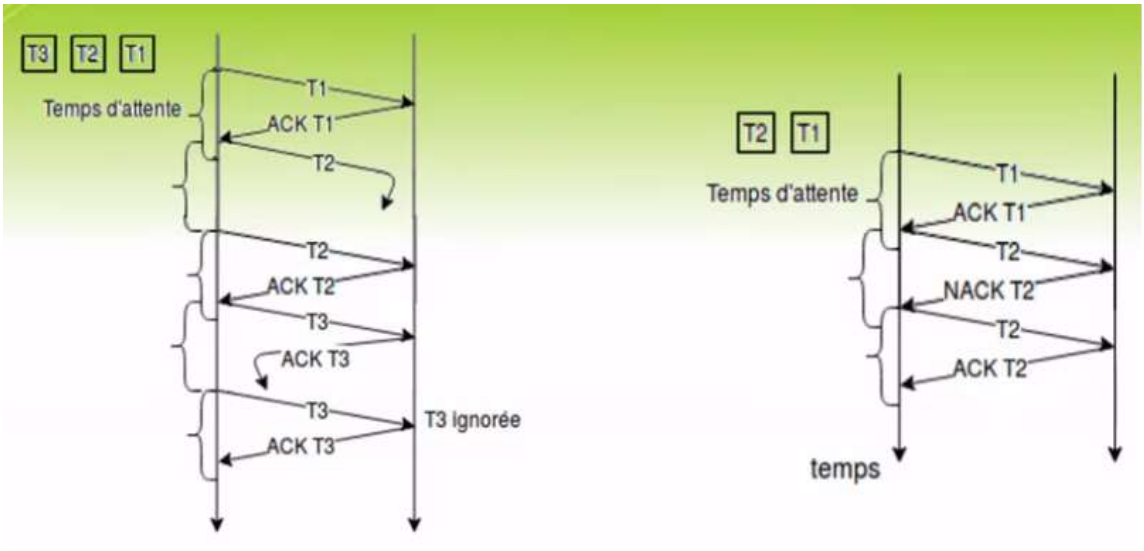
- But: régulariser l'émission des trames sur la capacité du récepteur





Contrôle de flux

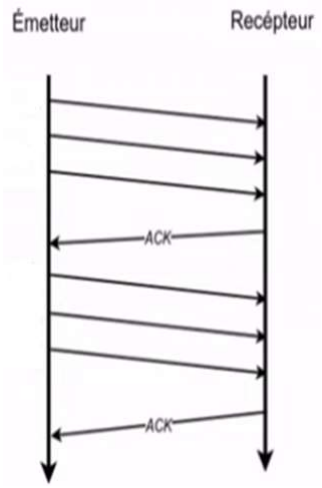
a. Protocole Send and Wait



Contrôle de flux

b. Protocole à fenêtre d'anticipation (piggypacking)

- L'émetteur peut envoyer un ensemble de trames sans attendre d'acquittement pour chaque trame, on appelle ça le **pipelining.**
- L'acquittement qui correspondant aux trames de données envoyées est appelé le **piggypacking.**



3. La norme 802.3 : Ethernet

- C'est la norme la plus utilisée pour les **réseaux locaux**
- Ethernet a été conçu par XEROX corporation dans des années 70.
  - ➢ Ether l'espace à travers lequel étaient censées se propager les ondes
  - ➢ Net abréviation de Network
- Il fait suite au développement d'un projet ALOHA (interconnexion par *liaison radio* des îles Hawai), avant de considérer la méthode **CSMA/CD**. Le réseau final permet de partager une liaison haut débit de plus de 100 mètres entre différents ordinateurs en bus sur un câble coaxial à 10 Mbits/s**.**
- Extension à des topologies en étoile

3.1 Normalisation des réseaux Ethernet

Les réseaux Ethernet sont les plus utilisés car le prix de revient n'est pas très élevé. Ils sont classés en différentes catégories selon leurs caractéristiques : type de support, longueur de segment, débit binaire, type de transmission. Cela a conduit à la normalisation représentée par la désignation suivante :

D TRANS L

- **D :** Désigne le débit binaire maximal sur le tronçon exprimé en Mbit/s.
- **Trans :** Désigne le type de transmission, ***Broad*** pour analogique et ***Base*** pour numérique.
- **L :** Peut prendre plusieurs valeurs :

➢**T ? : {Tx , T4, T,… } :** Pour exprimer une topologie en étoile utilisant un hub et de la paire torsadée. La longueur d'un segment est égale à 100 mètres maximum. **Exemple :** 100 base TX, 10 base T.

➢**F ? : { Fx , F,… } :** Pour exprimer une topologie en étoile en utilisant un hub et de la fibre optique. La longueur d'un segment est égale à 500 mètres maximum. **Exemple :** 100 base FX, 10 base F.

➢**V** : valeur pour désigner la longueur maximale en centaines de mètres d'un segment de câble coaxial dans un réseau en topologie bus. **Exemple :** 10 base 2, 10 base 5, 10 Broad 35.

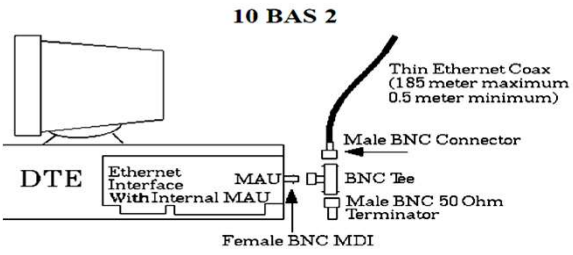
➢**XX :** Toute autre codification normalisant le Giga Ethernet dont les performances dépassent le 1 Gbits. **Exemple : LH, SX, ZX,…**

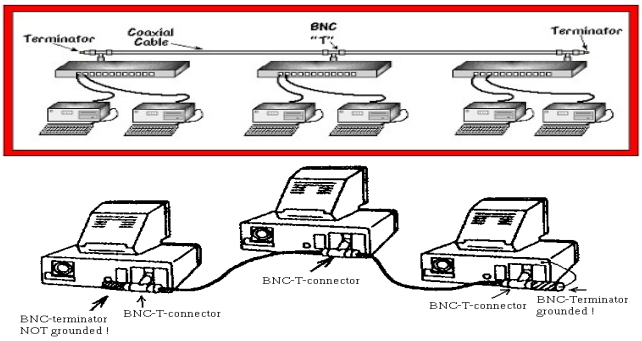
3.2 Description de quelques réseaux Ethernet

Il existe actuellement trois déclinaisons d'Ethernet normalisées par IEEE : Standard Ethernet 10Mbps (norme **802.3**), le Fast Ethernet à 100 Mbps (norme **802.3u**) et le Gigabit Ethernet (norme **802.3ab**).

I. Ethernet Standard 10 Mb

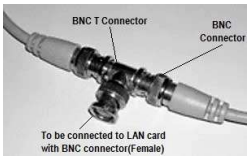
- **10 Base 2 :** est un réseau utilisant un câble coaxial fin avec des connecteurs BNC en T. Il est facile à installer, par contre, comme les connecteurs affaiblissent le signal on ne peut mettre que 30 stations sur le câble.





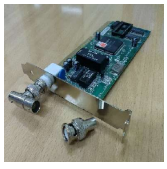
Bouchon

Connecteur BNC (T)



Raccordement BNC

Câble coaxial fin

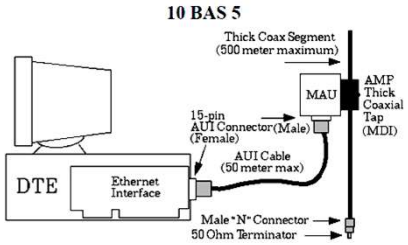


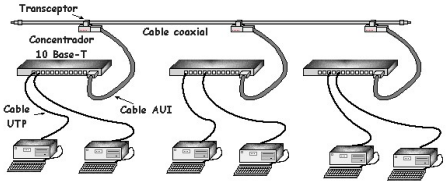
Carte réseau 10 base 2

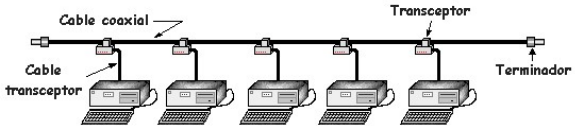
Hub avec extension BNC

Norme 802.3

- **10 Base 5 :** Utilise un *câble coaxial épais* ce qui permet d'augmenter les distances couvertes tout en remplaçant les connecteurs BNC par des adaptateurs MAU interfaçant le câble principal avec le câble de liaison reliant l'adaptateur à la carte réseau. Ce câble appelé « *drop câble* » peut être soit de la paire torsadée, un câble parallèle, où du câble coaxial fin, et sa longueur ne dépasse pas 50 mètres.



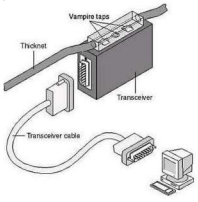


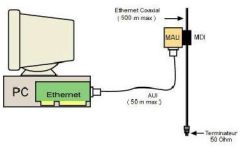


Norme 802.3

Câble coaxial épais

Raccordement MAU

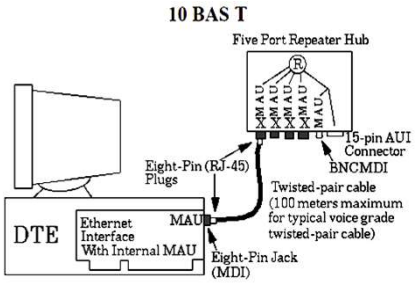


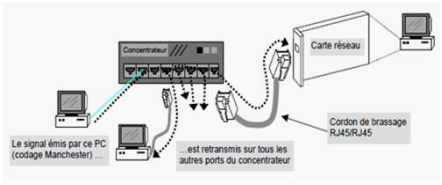


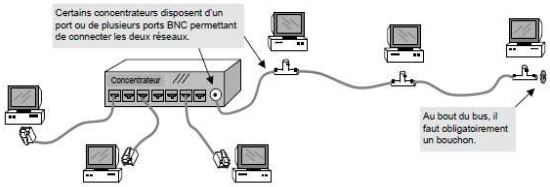
Raccordement avec carte réseau

Norme 802.3

- **10 Base T :** Utilise un Hub en topologie étoile, la paire torsadée relie chaque station au Hub. La paire torsadée est de *catégorie 2* (2 paires de fils). La distance d'un segment est de 100 mètres.
- **10 Base T4 :** Utilise de la paire torsadée de *catégorie 4* ; donc plus robuste aux erreurs de transmissions.

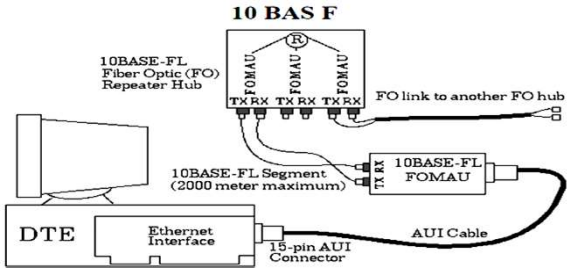






Norme 802.3

10 Base F : Utilise la fibre optique comme câble principal en maintenant le même type de carte réseau que celui du 10 base 5, avec un FOMAU externe assurant la conversion des signaux lumineux en signaux électriques.



FOMAU

Norme 802.3

10 Broad 36 : est un standard initialisé en 1985 et mis au point par le groupe de travail IEEE 802.3b du sous-comité de standardisation IEEE 802.3. Celui-ci permet la transmission de données jusqu'à un débit de 10Mbit/s sur du câble coaxial 75 ohms et sur une longueur pouvant atteindre 3600 mètres.

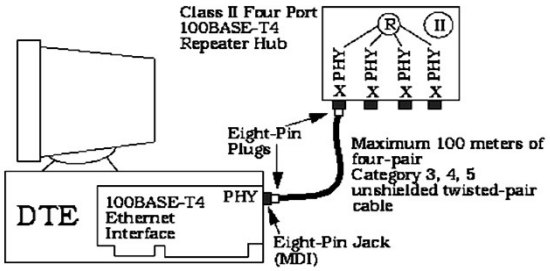
- Il se présente comme un 10 base 5, seulement le câble coaxial est différent (câble CATV) et le MAU externe est remplacé.
- Il utilise des MAU spéciaux : convertisseurs (Num/Analogiques) en quelque sorte des modems pour la transmission du signal analogique.
- Il permet de ce fait une plus grande couverture et une meilleure fiabilité du signal.

Norme 802.3

Norme	10 base 5	10 base 2	10 base T	10 broad 36	10 base F
Support	Coaxial 50Ω Câble jaune	Coaxial 50Ω Câble noir RG58	Paire torsadée	Coaxial 50Ω Type CATV	Fibre optique
Vitesse	10 MBPS	10 MBPS	10 MBPS	10 MBPS	10 MBPS
Longueur de segment	500 m	185 m	100 m	1875 m	1 km
Taille du réseau	2,5 km	925 m	4 à 5 hubs en cascade	3675 m	-
Distance Min inter station	2,5 m	0,5 m	-	-	-
Nombre de stations par segment	100 max	30 max	-	1024	-
Codage	Manchester	Manchester	Manchester	Analo PSK	-
Topologie	Bus	Bus	étoile	Bus	Etoile
Câble	Semi rigide avec rayon de courbure 30 cm	Souple	Catégorie 2 ou 3	Souple	Multimode
Connecteurs et prises	Prises piquées MAU externe	Connecteurs BNC en T vissés	-	-	Utilise des FOMAU
Remarques		MAU intégré dans carte réseau	HUB	Utilise des Modems	HUB FO

Fast Ethetnet

- **100BASE T4** : Permet le 100 Mbit/s (en HALF-duplex seulement) sur du câble de catégorie 3 , 4 ou 5).

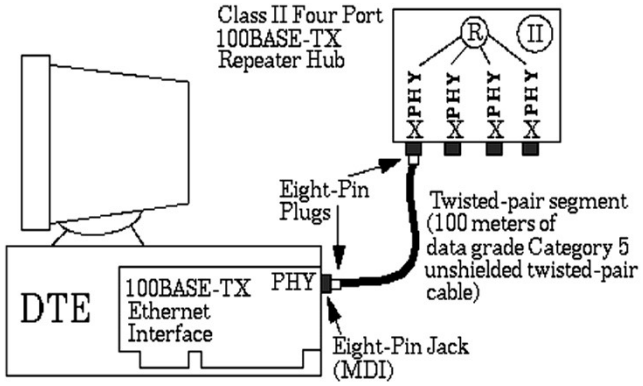


Fast Ethernet

- Amélioration de la norme IEEE 802.3 (addenda nommé 802.3u) en 1995
- Entièrement compatible avec 10BASE-T
  - Topologie en étoile : hub ou commutateur avec paires torsadées
  - Protocole CSMA/CD
  - Même format de trame
- Ce sont le codage du signal et la catégorie des câbles qui changent.
- Trois types de câblages autorisés
  - 100Base-T4 (UTP3)
  - 100Base-TX (UTP5)
  - 100Base-FX (fibre optique

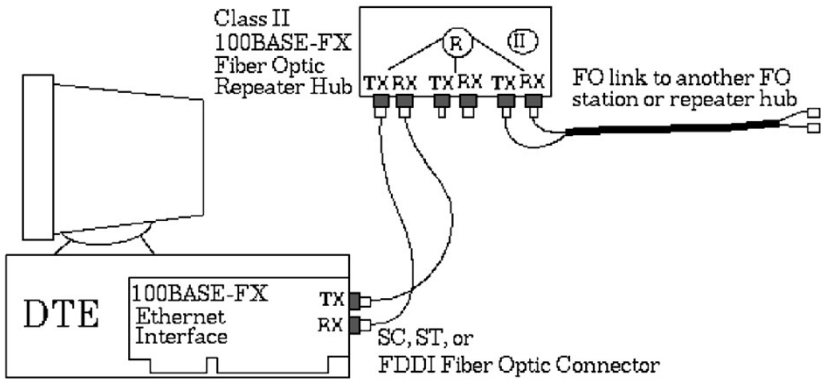
Norme 802.3

100 Base Tx : Standard qui fonctionne en FULL duplex qui utilise de la paire torsadée de catégorie 5 (STP) et des cartes réseaux et un Hub puissant (100 Mbs). Possible avec Hub ou switch.



Norme 802.3

100 Base Fx : Reprend la même architecture que le 10 base F en intégrant le FOMAU dans la carte réseau ce qui permet d'augmenter le débit jusqu'à 100 Mbs.



Norme 802.3

Nom	Type	Longueur max segment	Mode de transmission	Codage
100Base-T4	Paire torsadée UTP 3, 4, 5	100m	Half-duplex	8B/6T
100Base-TX	Paire torsadée UTP5 ou STP	100m	Full-duplex	4B/5B puis MLT-3
100Base-FX	Fibre optique multimode	2000m	Full-Duplex	4B/5B puis NRZI
		400m	Half-Duplex	

Norme 802.3

Gigabit Ethernet

- Norme IEEE 802.3z, ratifiée en 1998
- Entièrement compatible avec toutes les normes Ethernet précédentes
- Mode full-duplex ou half-duplex
- Paire torsadée ou fibre optique

Giga Ethernet

- **1000 BASE-T :** 1 Gbit/s sur câble de paires torsadées de catégorie **5** (classe D) ou supérieure, sur une longueur maximale de 100 m. Il opère en *full duplex*. La topologie est ici toujours en étoile et utilise obligatoirement des commutateurs (*switch*).
- **1000BASE-CX :** Une solution pour de courtes distances (jusqu'à 25 m) pour le 1Gbit/s sur du câble de paire torsadée spécial.
- **1000BASE-SX :** 1 Gbit/s sur fibre optique multimode.
- **1000BASE-LX :** 1Gbit/s sur fibre optique monomode et multimode.
- **1000BASE-LH :** 1Gbit/s sur fibre optique, sur longues distances.
- **1000BASE-ZX :** 1Gbit/s sur fibre optique monomode longues distances.

Ethernet 10 Gigabits

Pour les réseaux locaux, réseaux métropolitains et réseaux étendus. Il est actuellement spécifié par un standard supplémentaire, l'IEEE 802.3ae dont la première publication date de 2002.

- **10G BASE-CX4** : utilise un câble en cuivre de type *infiniband 4x* sur une longueur maximale de 15 mètres par segment.
- **10G BASE-T** : transmission sur câble catégorie 6, 6 A ou 7 (802.3an), en full duplex sur 4 paires sur une longueur maximale de 100 mètres.
- **10G BASE-SR** : opère sur de courtes distances sur de la *fibre multimode*, il a une portée de 26 à 82 mètres, en fonction du type de câble. Il supporte aussi les distances jusqu'à 300 m sur la fibre multimode à 2 000 MHz.
- **10G BASE-LR et 10G BASE-ER :** Ces standards supportent jusqu'à 10 et 40 km respectivement, sur fibre monomode.
- **10G BASE-SW, 10G BASE-LW et 10G BASE-EW :** Ces variétés utilisent le *WAN PHY*, qui est un standard physique conçu pour intégrer et inter-opérer les trois réseaux pour former un WAN. Ils utilisent le même type de fibre, en plus de supporter les mêmes distances.

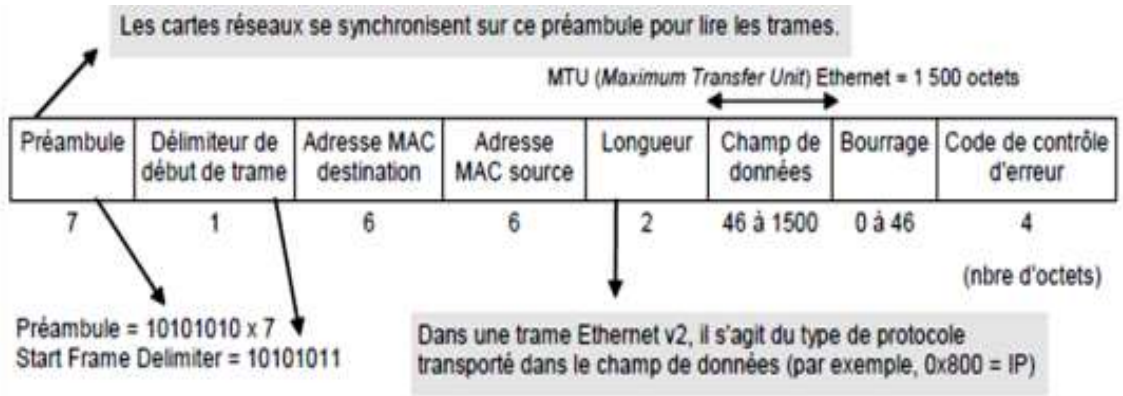
Norme 802.3

Type	Vitesse	Distance	Type de câble
10BASE-T	10 Mb / s	100m	Cuivre
100BASE-TX	100 Mb /s	100m	Cuivre
100BASE-FX	100 Mb / s	412 m 2 Km	half Duplex Multi-mode Fibre optique Full Duplex multi-mode Fibre optique
1000 Base LX	1000 Mb / s 1000 Mb / s	3Km 550m	Single-mode Fibre optique (SMF) Multi-mode Fibre optique (MMF)
1000 Base SX	1000 Mb / s 1000 Mb / s	550m 275m	Multi-mode Fibre optique (50u) Multi-mode Fibre optique (62.5 u)
1000 Base C (pas supportée par les applications industrielles standards)	1000 Mb / s	25m	Cuivre, 4 paires UTP5
1000BaseT - 1000 Base TX IEEE 802.3 ab ratifié le 26 juin 1999,	1000 Mb / s	100m	Cuivre, câble catégorie 5e, transmission sur 4 paires (250 Mbits/paire)
1000 BASE LH	1000 Mb / s	70 km	Fibre optique

Norme 802.3

3.3 Structure d'une trame ETHERNET

Une trame Ethernet a une taille minimale de 64 octets.



Norme 802.3

- ***Amorce (préambule):*** Représente l'annonce de l'envoi de la trame. Elle est composée de 7 octets positionnés à 10101010. Cette amorce permet de synchroniser les stations réceptrices.
- ***Start Frame Délimiter :*** délimiteur de début de trame 10101011.
- ***Adresse destination, Adresse source :*** Ce sont les adresses MAC physiques du réseau, codées sur 6 octets
- ***Longueur du champ d'information (Lenght) :*** Ce champ indique sur 2 octets la longueur des données de la trame LLC encapsulée. Ce nombre est compris entre 0 et 1500 octets.
- ***Données (Data) :*** Champ de la trame LLC (*Logical Link Control*).
- ***Bourrage (PAD) :*** Octets de bourrage ajoutés si la trame LLC ne contient pas 46 octets pour satisfaire la taille minimale d'une trame.
- ***FCS (Frame Control Sequence) :*** Constitué d'un mot de 32 bits, ce champ représente le code de vérification d'erreur sur la trame. La détection d'erreur sur une trame 802.3 se fait à l'aide d'un code polynomial dont le polynôme générateur est : $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + 1$.

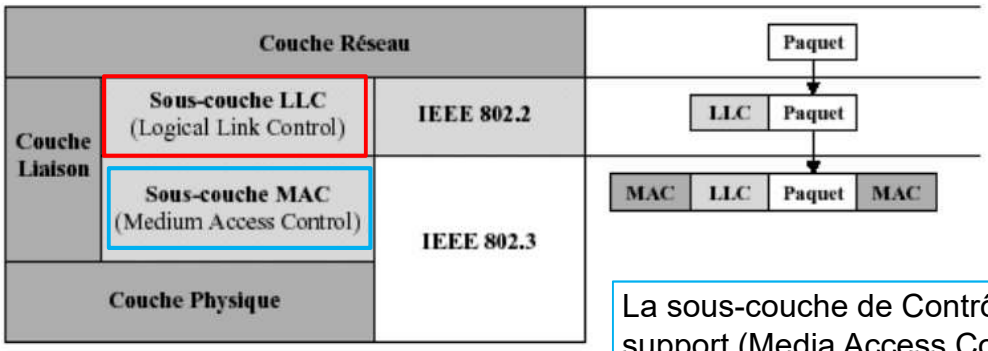
Norme 802.3

Adresses MAC Ethernet

- Dans une trame, émetteurs et destinataires sont identifiés grâce aux adresses MAC, et dont le format est standardisée par l IEEE.
- Chaque carte réseau Ethernet se distingue par une adresse MAC unique. Elle est constitué de 6 octets (48 bits) de type : *X : X : X : X : X : X* où chaque *X* varie de 0 à 255 mais plus souvent donné en hexadécimal (Exemple : 4D : EE : 52 : A4 : F6 : 69).

OUI (*Organizationally Unique Identifier*) = Partie de l'adresse affectée par l'IEEE à un fabricant de carte : 02608C pour 3com, 00000C pour Cisco, etc.	Partie de l'adresse affectée par le fabricant de la carte : de 000001 à FFFFFE

Les sous-couches liaison de données



La sous-couche Contrôle de la liaison logique (Logical Link Control - LLC)

- Fournit mécanismes pour le contrôle de flux,
- Gestion des accusés de réception (acquittements)
- Détection et correction d'erreurs.

La sous-couche de Contrôle d'accès au support (Media Access Control - MAC)

- Réguler les émissions sur un support donné (méthodes d'accès)
- La description des formats de trame (cellule élémentaire du transport d'information).
- Adressage : méthodes de repérage des stations émettrices et réceptrices.

4. Contrôle des erreurs

- Deux stratégies pour le contrôle des erreurs de transmission : la **détection/retransmission** et la **correction**

--La **détection/retransmission** consiste à ajouter juste assez de **redondances** dans les données à transmettre afin que le récepteur puisse détecter les erreurs sans pouvoir les corriger.

-La **correction** consiste à inclure dans les blocs de données suffisamment de **redondances** pour que le récepteur puisse restituer les données originales à partir des données reçues,

4.1 Généralités sur les codes

Rappel sur les opérations binaires

Somme modulo 2	$0 \oplus 0=0$	$0 \oplus 1=1$	$1 \oplus 0=1$	$1 \oplus 1=0$
Multiplication	0.0=0	0.1=0	1.0=0	1.1=1

Principe général

- Chaque suite de bits à transmettre est augmentée par une autre suite de bits dite de **redondance** ou de **contrôle**.

- Pour chaque suite de **k** bits transmise, on ajoute **r** bits. On dit alors que l'on utilise un code **C(n,k)** avec **n = k +r**.

n: taille du code;

k: taille de l'information utile.

- À la réception, les bits ajoutés permettent d'effectuer des contrôles de bonne réception.

  Dans le cas d'une réception sans anomalie, il suffira d'extraire l'information utile.

Définition 1

On appelle un code de longueur *n*, noté *C(n,k)* telle que *n* est la taille du code et *k* est la taille de l'information utile, un ensemble *C* de séquences de *n bits* (mots distincts), construits sur l'ensemble {0,1}.

- Une séquence de *n* bits est dite un ***mot de code***.
- Un mot de code de *n* bits n'appartenant pas à *C* sera dit *invalide*.
- Un code *C(n k)* contient 2^k mots de codes valides.

Exemple : Soit un code *C(4,2)* : { 00**10** 10**00** 01**11** 11**10** };

la séquence 10**01** définit un mot de code invalide

Contrôle des erreurs

A la réception d'une séquence de *n bits,* deux cas sont possibles :

- La séquence correspond à un mot du code et la transmission sera considérée comme étant **correcte**.
- La séquence n'est pas valide. Dans ce cas, on est en présence d'une erreur et le récepteur peut alors soit corriger l'erreur, soit demander une retransmission.

Example : Soit le code *C(4,2)* : { 0010 1000 0111 1110 }.

- A l'émission : On veut transmettre 00, on récupère son code 0010.
- Erreur de transmission : 1er cas 0000 **(erreur simple)**

- A la réception, on vérifie si le mot est valide (s'il appartient au code) :
  Erreur mot non valide (erreur détectable)

Contrôle des erreurs

Définition 2 (*Efficacité d'un code*) : L'*efficacité* d'un code est d'autant meilleure que les mots du code sont plus distincts les uns des autres. Elle dépend de la distance minimale de *Hamming* entre les différents mots de codes.

$$L'efficacité \text{ du code} = \frac{\text{nombre de messages reconnus faux}}{\text{nombre de messages faux}}$$

Plus l'efficacité est proche de 1, plus le code est performant

- **Définition 3 :** Soit *X* le bloc émis et *X'* le bloc reçu. On appelle ***vecteur d'erreur de bloc E :***

$$E = X \oplus X'$$

Si ***E = 0***, alors il n y pas d'erreur. Si ***E ≠ 0,*** il y a une erreur,

4. 2 Distance de Hamming

- ***Distance de Hamming*** : C'est le nombre de bits en lesquels 2 mots d'un code diffèrent. c'est le nombre de bits à 1 dans le résultat du XOR.

- ***Distance de Hamming minimale d'un code :*** On appelle *distance de Hamming minimale*, notée Dh_{min}, d'un code ***C***, le minimum des distances entre 2 mots quelconques de ce code.

Exemple : M = 10001001 et M' = 10110001 => *Dh* (M, M') = 3.

$$\begin{array}{r} 10001001 \\ \underline{10110001} \\ 00111000 \end{array}$$

- La Dh_{min} d'un code *C(n,k)*, est obtenu en comparant ses 2^k mots de code valides. Exemple:

  { 0010 1000 0111 1110 } Dh_{min} =**2**.

- Dh_{min} d'un code permet d'évaluer son pouvoir détecteur d'erreurs, ainsi que son pouvoir correcteur. En effet, si la Dh_{min} entre deux mots de code est *d*, il faut *d* erreurs pour transformer un mot en un autre.

*Un code **C** peut:*

- *détecter des erreurs d'ordre **DH(C) – 1***
- *corriger des erreurs d'ordre **(DH(C) – 1)/2***

Distance de Hamming du code	Ordre maximal des erreurs **détectables**	Ordre maximal des erreurs **corrigibles**
1	-	-
2	1	-
3	2	1
4	3	1
5	4	2
6	5	2

Code de Hamming

Le code de Hamming est une technique utilisée pour la détection d'une erreur simple (sur un seul bit) par l'ajout de bits de contrôle de parité. Chacun des bits de contrôle représente la parité d'un sous-ensemble de bits de l'information utile.

A l'émission trois étapes sont utilisées pour trouver le code de Hamming :

– *Calcul du nombre total de bits de contrôle.*
– *Déterminer les positions des bits de contrôle.*
– *Calcul des valeurs des bits de contrôle.*

Code de Hamming

Calcul du nombre de bits de contrôle (de parité)

Le nombre de bits de contrôle c'est le plus petit nombre ***r*** vérifiant l'inéquation suivante :

$$2^r \geq n + r + 1$$

Où ***n*** est le nombre de bits de l'information utile $M = m_0 m_1 m_2 \ldots m_{n-1}$

Code de Hamming

Déterminer les positions des bits de contrôle

- Les bits de contrôle seront placés aux positions qui sont des puissances de 2 : 2^0, 2^1, 2^2, 2^3, 2^4, …
- Donc le bit de contrôle C_i sera placé dans la position 2^i, avec i = 0, 1, 2, 3,…
- Les positions sont numérotées à partir de la gauche de ***1*** jusqu'à ***n + r.***

C_0	C_1	m_0	C_2	m_1	m_2	m_3	C_3	.....	m_9	C_4	.....	C_5	.....
2^0	2^1	3	2^2	5	6	7	2^3	.....	15	2^4	.....	2^5	.....

Code de Hamming

Calcul des valeurs des bits de contrôle.

Utiliser la représentation binaire non signée (RBNS), sous forme d'une décomposition de puissances de 2, pour calculer les valeurs des bits de contrôle à insérer aux positions déterminées à l'étape précédente : un bit de l'information utile occupant la position ***j*** participe dans le calcul du bit de contrôle de la position 2^i si la *RBNS* de ***j*** contient 2^i. le résultat c'est le xor de tous les bits de l'information utile vérifiant la condition précédente pour chaque bit de contrôle.

NB. La RBNS pour un entier c'est la somme de puissances de 2 de ce nombre;

$$J = \sum_{k=0}^{s} a_k 2^k, avec\ a_k = 0\ ou\ 1$$

Par exemple : $J = 7 = 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 = 2^0 + 2^1 + 2^2$

$J = 14 = 0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 = 2^1 + 2^2 + 2^3$

Code de Hamming (Exemple)

- Pour ***n = 4*** et $M = m_0m_1m_2m_3 = 1011$, quel est le code de Hamming associé ?

Calculons ***r*** le nombre de bit de contrôle selon l'inéquation $2^r \geq n + r + 1$

- Pour ***r = 1*** : $(2^1 = 2) \geq (4 + 1 + 1 = 6)$ … $faux$
- Pour ***r = 2*** : $(2^2 = 4) \geq (4 + 2 + 1 = 7)$ … $faux$
- Pour ***r = 3*** : $(2^3 = 8) \geq (4 + 3 + 1 = 8)$ … $vrai, donc\ r = 3$

On aura le code ***C = (7, 4)*** : 4 bits d'information utile et 3 bits de contrôle C_0, C_1, C_2

Code de Hamming (Exemple)

Déterminer les positions des bits de contrôle

Pour $n = 4$ et $M = m_0m_1m_2m_3 = 1011$,

Les positions des 3 bits de contrôle C_0, C_1, C_2 sont respectivement $2^0, 2^1, 2^2$:

- C_0 dans la position 1
- C_1 dans la position 2
- C_2 dans la position 4

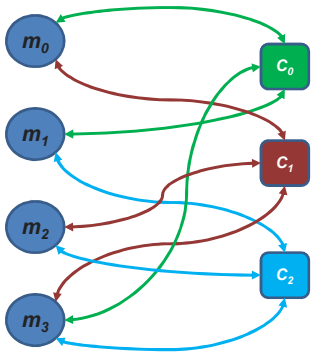
C_0	C_1	m_0	C_2	m_1	m_2	m_3
1	2	3	4	5	6	7

Code de Hamming (Exemple)

Calcul des valeurs des bits de contrôle

- Pour chaque position des 4 bits m_0, m_1, m_2, m_3 la représentation *RBNS est donnée comme suit :*

Bit m_i	*position*	*RBNS*	*Bits de contrôle*
m_0	3	$2^0 + 2^1$	C_0, C_1
m_1	5	$2^0 + 2^2$	C_0, C_2
m_2	6	$2^1 + 2^2$	C_1, C_2
m_3	7	$2^0 + 2^1 + 2^2$	C_0, C_1, C_2



Graphe biparti

Code de Hamming (Exemple)

Calcul des valeurs des bits de contrôle

Pour M = $m_0m_1m_2m_3$ = 1011

- C_0 est lié à m_0, m_1 et m_3 donc $C_0 = m_0 \oplus m_1 \oplus m_3 = 1 \oplus 0 \oplus 1 = 0$
- C_1 est lié à m_0, m_2 et m_3 donc $C_1 = m_0 \oplus m_2 \oplus m_3 = 1 \oplus 1 \oplus 1 = 1$
- C_2 est lié à m_1, m_2 et m_3 donc $C_2 = m_1 \oplus m_2 \oplus m_3 = 0 \oplus 1 \oplus 1 = 0$

Donc $C_0 = 0$, $C_1 = 1$ et $C_2 = 0$

- *Le code de Hamming résultant est : C = 0110011*

C_0	C_1	m_0	C_2	m_1	m_2	m_3
0	1	1	0	0	1	1

Code de Hamming

Détection de l'erreur

Le récepteur doit faire la procédure inverse, à partir du code reçu :

- il faut extraire les bits de l'information utile,
- recalculer les bits de contrôle
- faire la comparaison avec ceux reçu.

Si tous les bits de contrôle calculés sont identiques a ceux reçu, alors pas d'erreur, sinon il y a un bit erroné.

La position du bit erroné c'est la somme des positions des bits de contrôle erronés.

Code de Hamming

Exemple Détection de l'erreur

Soit à vérifier le code de Hamming (7, 4) : *C = 0110001*

Les bits de contrôle dans les positions 2^0, 2^1 et 2^2 sont $C_0 = 0$, $C_1 = 1$ et $C_2 = 0$

L'information utile M = $\mathbf{m_0 m_1 m_2 m_3 = 1001}$

Code de Hamming

Détection de l'erreur (Exemple)

Nous avons $C_0 = 0$, $C_1 = 1$ et $C_2 = 0$ (bits reçus)

L'information utile reçu M = $\mathbf{m_0 m_1 m_2 m_3 = 1001}$

- C_0 est lié à m_0, m_1 et m_3 donc $C_0 = m_0 \oplus m_1 \oplus m_3 = 1 \oplus 0 \oplus 1 = 0 =$ $bit C_0$ *reçu, **donc pas d'erreur;***
- C_1 est lié à m_0, m_2 et m_3 donc $C_1 = m_0 \oplus m_2 \oplus m_3 = 1 \oplus 0 \oplus 1 = 0 \neq$ $bit\ C_1$ *reçu, **donc erroné;***
- C_2 est lié à m_1, m_2 et m_3 donc $C_2 = m_1 \oplus m_2 \oplus m_3 = 0 \oplus 0 \oplus 1 = 1 \neq$ $bit\ C_2$ *reçu, **donc erroné;***

Code de Hamming

Détection de l'erreur (Exemple)

- Nous avons C_1 et C_2 *erronés, leurs positions sont 2 et 4 donc le bit erroné est le* $6^{ème} = 2 + 4$.
- *C = 0110001 ➔ il faut corriger par inversion du* $6^{ème}$ *bit*
- *Donc le code correcte est C = 0110011*
- *L'information utile corrigée est M = 1011*

4.3 Méthode basée sur la parité

L'information transmise est découpée en *blocs* de *k* bits, puis on leur rajoute *r* bits de redondance. On crée alors un *code de bloc* de longueur $n=k+r$. Sur 2^n combinaisons possibles, seules 2^k combinaisons sont valides. Les bits de redondance sont calculés de différentes méthodes.

4.3.1 Parité transversale (ou verticale) VRC (*Vertical Redundancy Checking*)

- L'information est sectionnée en blocs de *k* bits qui sont généralement des caractères, puis on ajoute à chaque bloc un bit de parité *(r=1)* de telle sorte que la somme des *k*+1 bits modulo 2 soit 0 (*parité paire*) ou égale à 1 (*parité impaire)*.

Exemple : Envoi d'un bloc de 4 caractères de longueur 3 (*k*=3) :

- Information utile : 110 001 011 000.
- Information envoyée : 1100 0011 0110 0000.

VRC permet de détecter une erreur simple sur chacun des mots transmis.

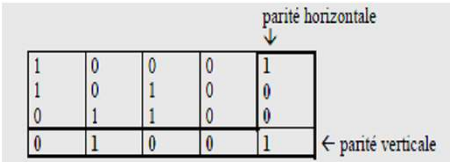
Contrôle des erreurs

4.3.2 Parité LRC/VRC: On combine généralement la parité transversale et la parité longitudinale de la façon suivante : les caractères munis de leur bit de parité transversale sont regroupés en blocs, et on ajoute à la fin de chaque bloc un caractère supplémentaire pour la parité longitudinale. Ce contrôle est appelé *Vertical Redundancy Checking / Longitudinal Redundancy Checking, LRC/VRC.*

Exemple : Envoi d'un bloc de 4 caractères avec contrôle LRC/VRC :

- Information utile : 110 001 011 000.
- Information envoyée : 1100 0011 0110 0000 1001.

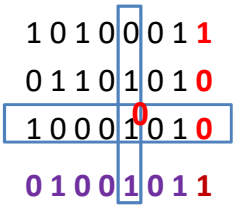
1	1	0	0
0	0	1	1
0	1	1	0
0	0	0	0
1	0	0	1



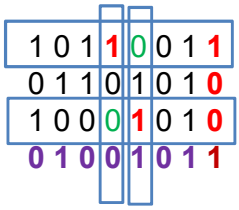
Contrôle des erreurs

- Comme une erreur simple modifie simultanément la parité d'une ligne et d'une colonne, la *correction* est possible en inversant le bit situé à l'intersection de la ligne et de la colonne ayant une parité incorrecte.

Exemples



On peut corriger
(Erreur simple)



Impossible de corriger
(Erreur double)

4. 4 Codes polynomiaux

Ce sont des codes de blocs très utilisés dans la pratique car facilement implantables et qui donnent d'excellents résultats. On considère que les bits d'une séquence sont les coefficients d'un polynôme. Ces coefficients ne prennent que les valeurs 0 ou 1. Un bloc de *n* bits est vu comme la série de coefficients d'un polynôme de *n termes*, allant de x^{n-1} à x^0. Un tel polynôme est dit de degré *n-1*. Le bit le plus à gauche (fort) est le coefficient de x^{n-1}, son voisin est le coefficient de x^{n-2}, ainsi de suite:

$$P(x) = p_{n-1}\, x^{n-1} + p_{n-2}\, x^{n-2} + \ldots + p_1\, x + p_0\, x^0 \qquad (p_i \in \{0,1\})$$

Par exemple, la séquence "**001101**" comprend 6 bits. Elle peut être représentée par un polynôme à 6 termes (degré 5) dont les coefficients sont 0, 0, 1, 1, 0 et 1, ce qui donne le polynôme :

$x^5.0+x^4.0+x^3.1+x^2.1+x.0+x^0.1=$ $\mathbf{x^3+x^2+1}$

- Formellement, supposons un code *C(n, k)*. ***L'information utile est représentée par le polynôme*** *Z(x)*.

  Z(x) est au maximum de degré *(k-1)* puisque l'information comporte *k* bits :

  $$Z(x) = u_{k-1}\, x^{k-1} + \ldots + u_1\, x + u_0\, x^0. \qquad (u_i \in \{0,1\})$$

- Pour utiliser un code polynomial, l'émetteur et le récepteur doivent d'abord se mettre d'accord sur le choix d'un *polynôme générateur G(x)*. Pour des raisons pratiques, les coefficients de poids fort et faible du générateur doivent être égaux à 1. *G(x)* est choisi de degré *r=n-k*.

  $$G(x) = g_r\, x^r + \ldots + g_1\, x + g_0\, x^0. \qquad (g_i \in \{0,1\})\ g_{r-1} = g_0 = 1$$

Contrôle des erreurs

Codage à l'émission

- Il faut ***multiplier*** *Z(x)* par x^r pour créer un décalage à gauche de *r* bits.

Exemple : Soit un code *C(11,7).* Nous considérons, *Z(x)* =1011011**,** on multiplie par x^4 et on obtient : 10110110000.

- On ***divise*** le ***produit obtenu*** par ***G(x)*** ; on obtient:

  $Z(x)\, x^r = Q(x)\, G(x) + A(x)$ Où :

  Q(x) : polynôme quotient.

  A(x) : polynôme reste de la division, au maximum de degré : *r-1***.**

- Donc on ***obtient le mot de code à envoyer***, représenté par le polynôme *Y(x)* de degré *n-1* suivant :

$$Y(x) = Z(x)\, x^r + A(x) = Q(x)\, G(x)$$

- On envoie la séquence de bits de longueur *n=k+r* associée au polynôme *Y(x)*. La séquence envoyée est construite en collant (rajoutant) à l'information utile, le total de contrôle représenté par *A(x)*. Le polynôme *Y(x)* obtenu est aussi **divisible par *G(x).*** Il en résulte que les mots valides du code polynomial *C(n,k)* sont donc les **polynômes multiples de *G(x).***

Contrôle des erreurs

Exemple

Soit le code *C(9,6)* avec ***G(x)= x^3+1.***

- On veut transférer l'information "001101".

Z(x) ***= x^3+x^2+1.***

$Z(x).x^3 = x^6+x^5+x^3$

$$\begin{array}{r|l} X^6+X^5+X^3 & X^3+1 \\ -(X^6+X^3) & X^3+X^2 \\ \hline X^5 & \\ -(X^5+X^2) & \\ \hline X^2 & \end{array}$$

- On divise $Z(x).x^3$ par *G(x)*, on obtient :

  $Z(x) \,.\, x^3 = G(x) \,.\, Q(x) + A(x)$, avec ***$A(x)=x^2$*** et $Q(x)=x^3+x^2$.

- D'où : $Y(x) = Z(x) \,.\, x^3 + A(x) = x^6+x^5+x^3+x^2 =$ ***$x^6+x^5+x^3+x^2$.***

Le polynôme *Y(x)* correspond à la séquence : 001101***100***

-*A(x)* doit être écrit sur *r bits*. Si A(x)=x on l'écrit 010 et non 10.

Décodage à la réception

- A la **réception**, un calcul semblable s'effectue sur le mot reçu, mais il faut, que le **reste** soit **nul**. Dans le cas contraire, c'est qu'une erreur est survenue en cours de transmission.

- Soit *Ÿ(x)* le polynôme de degré *n-1* dénotant le ***mot de code reçu***.

$$S(x) = \ddot{Y}(x) \ MOD[\ G(x)]$$

- Aucune erreur n'est détectée si le syndrome $S(x) = 0$. Sinon elle est détectée si $S(x) \neq 0$. La détection d'erreur consiste à vérifier que le mot reçu est bien un mot du code $C(n,k)$ c'est-à-dire que $\ddot{Y}(x)$, est divisible par $G(x)$.
- Si le polynôme est divisible par $G(x)$, alors il suffit alors d'extraire l'information utile en supprimant les r derniers bits de la séquence reçue. Si le polynôme n'est pas divisible par $G(x)$, alors une erreur a eu lieu pendant la transmission et le récepteur demandera une retransmission du message.

Exemple : On envoie la séquence de bits"001101**100**". On remarque bien que cette séquence est construite en concaténant à l'information utile "001101", la séquence "**100**" qui correspond à la redondance.

Cas 1 : On reçoit "011101100" (Noter l'erreur de transmission sur le deuxième bit).

$\ddot{Y}(x) = x^7 + x^6 + x^5 + x^3 + x^2$.
On calcule $S(x)$:

$$\begin{array}{r|l} X^7+X^6+X^5+X^3+X^2 & X^3+1 \\ -(X^7+X^4) & \hline X^4+X^3+X^2+X \\ X^6+X^5+X^4+X^3+X^2 & \\ -(X^6+X^3) & \\ X^5+X^4+X^2 & \\ -(X^5+X^2) & \\ X^4 & \\ -(X^4+X) & \\ X & \end{array}$$

Le reste étant non nul, une erreur de transmission est détectée.

Cas 2 : On reçoit "001101100" (Noter l'absence d'erreur de transmission).

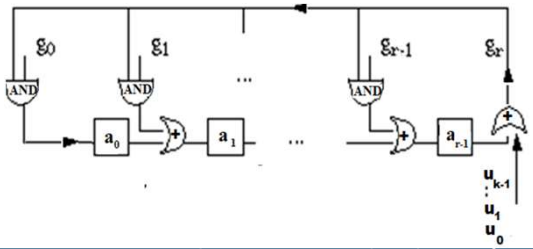
$\ddot{Y}(x) = x^6 + x^5 + x^3 + x^2$.
On calcule $S(x)$:

$$\begin{array}{r|l} X^6+X^5+X^3+X^2 & X^3+1 \\ -(X^6+X^3) & \hline X^3+X^2 \\ X^5+X^2 & \\ -(X^5+X^2) & \\ 0 & \end{array}$$

Aucune erreur de transmission n'est détectée et l'information utile est la séquence de bits obtenue en supprimant les 3 derniers bits (r=3) de la séquence "001101**100**".

4.5 Circuit logique d'un codeur polynomial

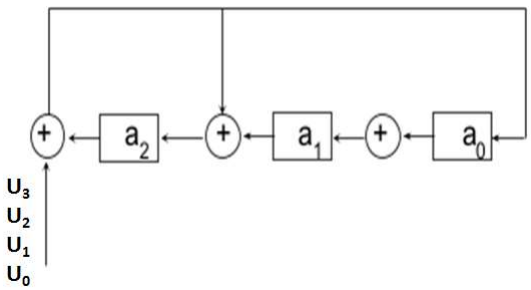
- La division se fait à l'aide d'un **circuit logique** appelé ***diviseur bâti*** autour d'un registre à décalage. Le registre est constitué de ***r*** bascules a_{r-1}, …, a_0, représentant les ***bits de contrôle***, liés par des opérateurs de ***OU exclusif*** $\oplus$. Les coefficients du polynôme *Z(x)* sont injectés dans le circuit un à un, à chaque cycle d'horloge, commençant du coefficient le plus fort au plus faible.
- Initialement, les registres a_i sont à zéro. Chaque coefficient en entrée est sommé avec le bit a_{r-1}. La sortie de cet opérateur va en entrée vers a_0. Si dans le polynôme *G(x)*, le coefficient de x^i est égal à 1, une branche de la sortie $x_i \oplus a_{r-1}$ est créée en entrée de l'opérateur *OU exclusif* mis avant la bascule du bit a_{i-1}. Après le passage des *k* bits de *Z(x)* en *k* cycle horloge, le calcul est achevé et les registres a_i contiennent les coefficients du polynôme *A(x)*.



Exemple

Soit un code C(7,4) avec ***G(x)= x^3+x^2+1***

u_i	u_i+a_2	a_0	a_1	a_2
		0	0	0
1	1	1	0	1
1	0	0	1	0
0	0	0	0	1
1	0	0	0	0



Contrôle des erreurs

Propriétés des codes polynomiaux

Soit Y(x) un mot envoyé, et Ÿ(x) le mot reçu correspondant tel que Ÿ(x)=Y(x)+E(x). On a alors les propriétés suivantes :

- Toute erreur simple est détectée si G(x) comporte plus d'un coefficient non nul.
- Les erreurs doubles sont toutes détectées si G(x) ne divise pas x^i+1 où i appartient à {r, n-1}, n étant la taille du code.
- L'erreur sur un message comportant un nombre impair d'erreurs est toujours détectée si le polynôme générateur G(x) est divisible par (x+1).
- Un code polynomial détecte toutes les salves (suite d'erreurs), de longueur inférieure ou égale à r avec r le degré de G(x).

Contrôle des erreurs

Polynômes générateurs utilisés : Le choix du polynôme générateur est très important : de lui dépendra le pouvoir de détection de certains types d'erreur de transmission. Les principaux polynômes utilisés :

- LRCC-8 : x^8+1
- LRCC-16 : $x^{16}+1$
- CRC 12 : $x^{12} + x^{11} + x^3 + x^2 + x + 1$
- CRC 16 Forward : $x^{16}+x^{15}+x^2+1$
- CRC 16 Backward : $x^{16}+ x^{14}+x + 1$
- CRC CITT Forward : $x^{16}+ x^{12}+ x^5 + 1$
- CRC CITT Backward : $x^{16} + x^{11} + x^4 +1$
- CRC-32= $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x +1$.

Contrôle des erreurs

Nom	Générateur	Factorisation	Exemples d'utilisation
TCH/FS -HS-EFS	$X^3 + X + 1$	irréductible	GSM transmission de voix
GSM TCH/EFS	$X^8 + X^4 + X^3 + X^2 + 1$	irréductible	GSM pré-codage canal à plein taux
CRC-8	$X^8 + X^7 + X^4 + X^3 + X + 1$	$(X + 1)(X^7 + X^3 + 1)$	GSM 3ème génération
CRC-16 X25-CCITT	$X^{16} + X^{12} + X^5 + 1$	$(X + 1)(X^{15} + X^{14} + X^{13} + X^{12} + X^4 + X^3 + X^2 + X + 1)$	Protocole X25-CCITT ; contrôle trames PPP FCS-16 (RFC-1662)
CRC-24	$X^{24} + X^{23} + X^{18} + X^{17} + X^{14} + X^{11} + X^{10} + X^7 + X^6 + X^5 + X^4 + X^3 + X + 1$	$(X + 1)(X^{23} + X^{17} + X^{13} + X^{12} + X^{11} + X^9 + X^8 + X^7 + X^5 + X^3 + 1)$	communications UHF et satellites (SATCOM) ; messages OpenPGP (RFC-2440)
CRC-24 (3GPP)	$X^{24} + X^{23} + X^6 + X^5 + X + 1$	$(X + 1)(X^{23} + X^5 + 1)$	GSM 3ème génération
CRC-32 AUTODIN-II	$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$	irréductible	IEEE-802.3, ATM AAL5, trames PPP FCS-32 (RFC-1662) ; contrôle d'intégrité des fichiers ZIP et RAR ;

Pour corriger, on utilise des polynômes avec des propriétés particulières.

Exemple: Codes cycliques, codes **BCH** (**B**ose, Ray-**C**haudhuri et **H**ocquenghem); Codes de Reed Solomon.

Contrôle d'accès multiple

5. Contrôle d'accès multiple

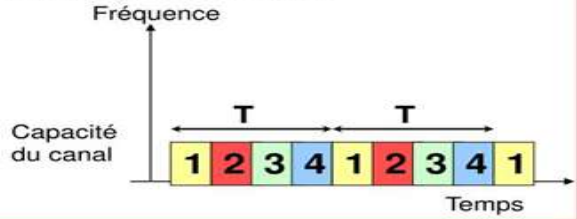
- Une méthode d'accès définit la politique d'accès aux supports du réseau lorsque plusieurs machines veulent communiquer en même temps, cette politique est implémentée dans la carte réseau au niveau de la sous couche MAC.
- Il existe de nombreuses techniques normalisées:
  - *Centralisées ou distribuées* : une station primaire désignée est chargée de régler les conflits d'accès, ou le contrôle est distribuée entre toutes les stations.
  - *Statiques ou dynamiques*.
  - *Déterministes ou probabilistes (Aléatoires):* garantie au bout d'un temps défini l'accès au support ou non (probabilité).
  - *Equitables ou non :* vis-à-vis des possibilités d'accès des stations
  - *Avec ou sans contentions*: existence de collisions de trames.

Contrôle d'accès multiple

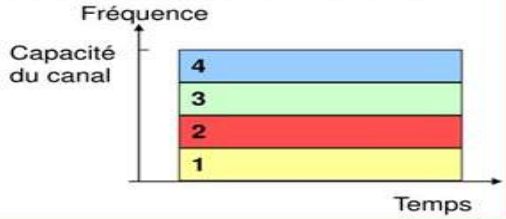
5.1 Techniques statiques (multiplexage FDMA, TDMA synchrone)

- La bande passante est répartie de façon définitive entre les stations (temporellement ou fréquentiellement).

- Accès Multiple à répartition dans le temps (AMRT) ou TDMA, *Time Division Multiple Access*



- Accès Multiple à répartition en fréquence (AMRF) ou FDMA, *Frequency Division Multiple Access*



- ☹ Mal adapté aux réseaux locaux où le retrait/ajout de stations est fréquent ce qui nécessite de redéfinir la trame fréquemment.
- ☹ Perte de la bande passante quand une station n'émet pas.

Contrôle d'accès multiple

5.2 Techniques probabilistes (aléatoires)

5.2.1 ALOHA

Mise en œuvre pour un réseau radio de diffusion de paquets reliant les îles d'Hawai.

Principe

- Attendre un acquittement au maximum pendant une durée égale à 2 fois le temps de propagation.
- Si le paquet subit une erreur ou une collision il faut une retransmission
- Au bout de n retransmissions successives du même paquet (avec échec), l'émetteur abandonne

Emission

- Accès au support pour émettre une trame
- Attendre un acquittement au maximum pendant une durée égale à deux fois le temps de propagation (slot)
- Si une réception d'acquittement est faite alors transmission OK
- Sinon ré-émission de la trame selon un algorithme de reprise

Réception

- Vérifier la trame reçue
- Si vérification est positive alors émission d'un acquittement
- Sinon rien (soit une collision s'est produite ou erreur de transmission)

5.2.2 ALOHA en tranches ou Slotted Aloha

Principe

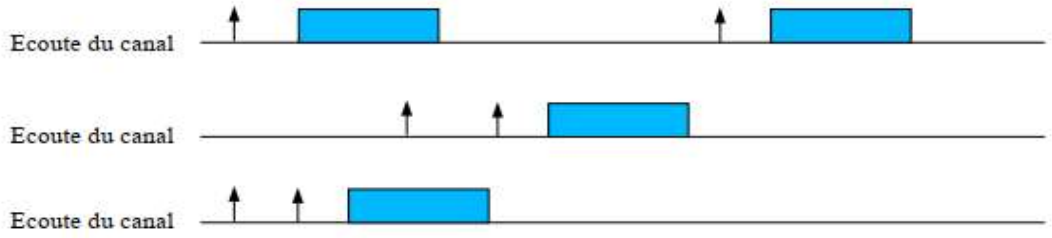
- Le temps est discrétisé : découpé en tranches de temps appelé slot
  - Les stations sont synchronisées
  - Une station transmet un paquet au début d'un slot
- Amélioration par rapport à ALOHA simple

5.2.3 Les techniques Carrier Sense Multiple Access (CSMA)

Principe

- Cette technique consiste à écouter le canal avant d'entreprendre une émission.
- Si le communicateur détecte un signal sur le canal, il diffère son émission à un moment ultérieur

Problème : il peut toujours y avoir des collisions en cours d'émission



CSMA/Collision Detection (IEEE 802.3)

Principe

- A l'écoute préalable du signal, s'ajoute l'écoute pendant la transmission et en cas de collision, la ré-émission au bout d'un temps aléatoire.

- Utilisé pour Ethernet, normalisée par l'ISO sous l'appellation 802.3

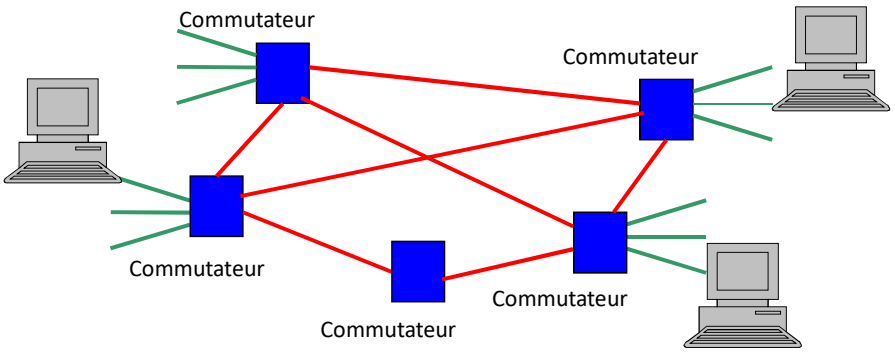
Pseudo-algorithme

- Ecouter le câble pour détecter la présence d'un signal.
- Si transmission en cours alors attendre la fin.
- Dés que le support est libre alors transmettre et rester à l'écoute pour détecter les collusions.
- Si collusion alors **E** stop l'envoi et attend un délai [0..N] puis retransmettre le signal.
- Si nouvelle collision alors **E** stop l'envoi et attend un délai [0.. 2N] puis retransmettre le signal.
- Ainsi de suite.

Avantages

- Gain d'efficacité
- Détection précoce des collisions
- Reprise après collision visant à diminuer la probabilité d'une nouvelle collision

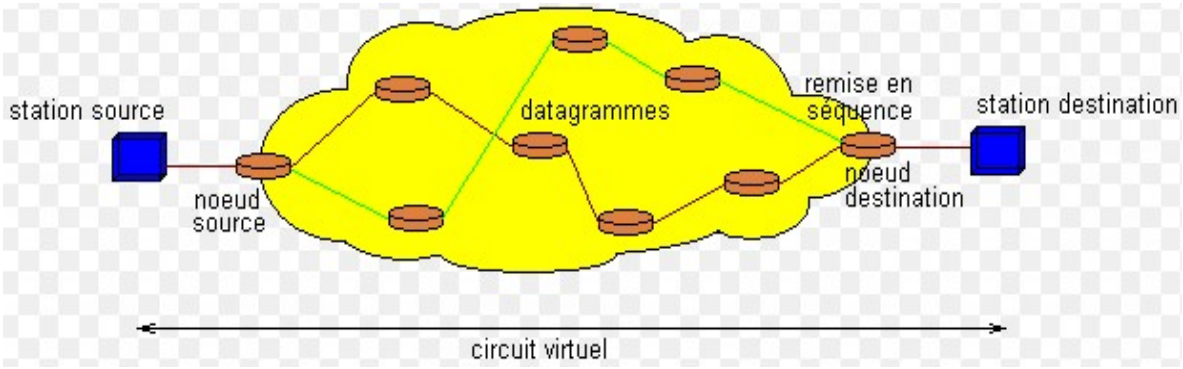
6. Commutation de circuit



Les réseaux à commutation permettent à tout équipement informatique connecté de communiquer directement avec tout autre équipement à travers un réseau de type maillée.

a. Commutation par circuit

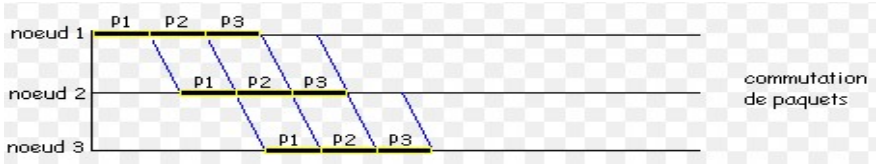
- Elle est la base de la commutation téléphonique, elle consiste à l'allocation des liaisons pour établir **le circuit virtuel** pour la durée de commutation entre l'émetteur et le récepteur avant l'échange des données.
- Le circuit établi en mode connecté, est un '*circuit privé* et sera libéré lorsque l'une des deux rompra la connexion.
- Il offre la sécurisation des données et une QdS soutenue. Les données sont acheminés sans retard car il n y a pas de délais d'attente au niveau des nœuds et dans l'ordre de leur séquencement à l'émission puisqu'ils empruntent tous le même chemin.
- Cette technique ne permet pas une gestion efficace des ressources et implique des coûts élevés pour la maintenance et la mémorisation du circuit virtuel tout au long des nœuds empruntés.
- L'établissement d'un circuit se fait en 3 temps :
  - Etablissement du circuit (appel) : ouverture de connexion
  - Phase de transfert des données
  - Fermeture du circuit (raccrochage) : fermeture de connexion



Commutation par circuit

b. Commutation par paquets

- Le message à transmettre est fragmenté par l'émetteur en blocs de longueur limitée appelés « paquets ».
- Chaque paquet est acheminé de manière indépendante dans le réseau des autres (empruntent des chemins différents) .
- Du fait de sa taille réduite, celui-ci est stocké temporairement dans les mémoires vives et non sur les disques.
- Utilise le principe ***Stock and Forward***, Une fois le message complètement reçu ,par chaque nœud , il est mémorisé, puis vérifie contre les erreurs, avant son envoi vers le commutateur suivant



Commutation par paquets

Commutation par circuit Vs Commutation par paquets

Commutation de circuit

- Circuit dédie
- Service Garanti (bande passante) -QoS
- Utilisation du support inefficace
- Chemin unique pas de redondance

Commutation par paquets

- Circuit partagé
- Messages divisés en paquets
- Utilisation efficace du support
- Redondance, plusieurs chemins possibles