

# **Technologies des réseaux de communication**

Gérard-Michel Cochard & Edoardo Berera & Michel Besson Thierry Jeandel & Gérard-Michel Cochard

**Université Virtuelle de Tunis**

2007

# Caractéristiques d'une voie de transmission

Sommaire :

[Introduction](#)

[Transmission d'une onde sinusoïdale](#)

[Signal quelconque et bande passante](#)

[Rapidité de modulation et débit binaire](#)

[Bruit et capacité](#)

[Trafic](#)

[Les supports de transmission](#)

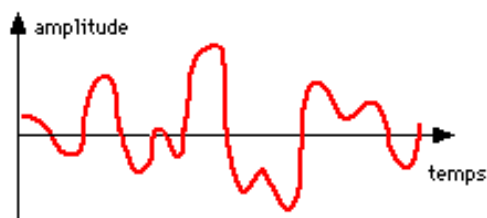
## Introduction

L'information qui transite sur les réseaux de télécommunication consiste en messages de types divers : textes, sons, images fixes ou animées, vidéo, etc.... La forme que revêt cette information est commode pour une communication directe et classique (conversation, échange sur papier, ....) lorsque les interlocuteurs sont en présence. Quand ils sont distants l'un de l'autre, l'emploi des réseaux de télécommunication est une manière moderne de résoudre la transmission d'informations. Toutefois, pour les nécessités du transport, la transmission d'un message nécessite un encodage en signaux de type électrique ou électromagnétique :

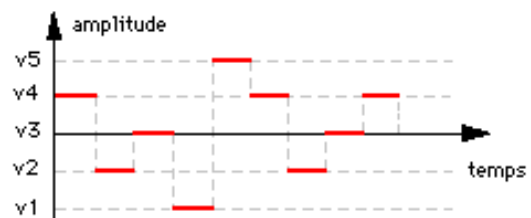


L'émetteur et le récepteur sont, de nos jours, des ordinateurs. La voie de transmission peut être une simple liaison directe entre émetteur et récepteur ou beaucoup plus complexe dans le cadre d'un ou plusieurs réseaux de télécommunications. Les signaux sont les véhicules de transport de l'information.

Les signaux peuvent être **analogiques** ou **numériques**



signaux analogiques : représentés par une grandeur physique variant de manière continue

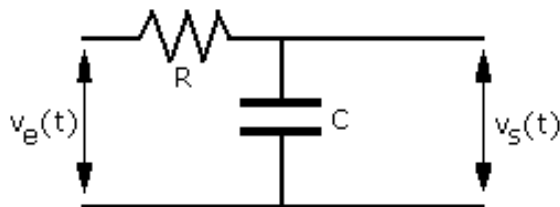


signaux numériques : représentés par une grandeur physique ne prenant qu'un certain nombre de valeurs discrètes

Exercices et tests : [QCM1](#), [QCM2](#)

# Transmission d'une onde sinusoïdale

L'onde sinusoïdale, infinie ou réduite à une période, est le plus simple des signaux en ce sens qu'elle est facilement générée, mais son intérêt réside surtout dans le fait suivant : n'importe quel signal peut être exprimé à partir d'ondes sinusoïdales. Ces faits justifient une étude particulière qui va permettre de définir quelques propriétés des voies de transmission. Considérons donc une voie de transmission, supposée point à point sans interruption ou intermédiaire et composée de deux fils métalliques. Un tronçon de voie peut alors être considérée comme un quadripôle (nous négligeons ici les effets d'induction) composé d'une résistance R et d'une capacité C.



Le signal sinusoïdal appliqué à l'entrée du quadripôle (tension entre les deux fils) est :

$$v_e(t) = V_e \sin \omega t$$

avec  $V_e$  : amplitude maximale ;  $\omega$  : pulsation ;  $f = \omega/2\pi$  : fréquence ;  $T = 2\pi/\omega = 1/f$  : période.

Le signal de sortie est

$$v_s(t) = V_s \sin (\omega t + \Phi)$$

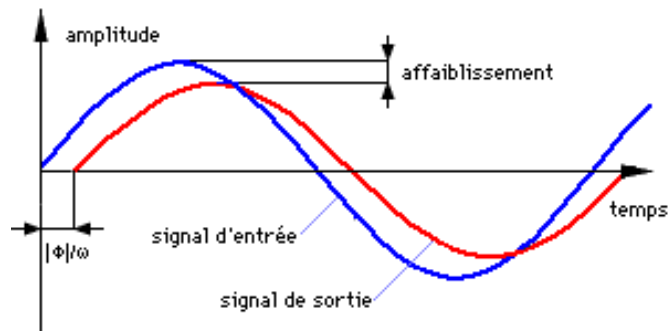
avec :  $\Phi$  : déphasage.

La tension de "sortie" dépend de la tension d'entrée mais aussi des propriétés physiques du quadripôle. Les lois de l'électromagnétisme montrent que, dans le cas simple considéré :

$$V_s/V_e = (1 + R^2 C^2 \omega^2)^{-1/2}$$

$$\Phi = \text{atan}(-RC \omega)$$

On constate donc que l'amplitude de sortie  $V_s$  est plus faible que l'amplitude d'entrée  $V_e$  : il y a **affaiblissement** et qu'il apparaît un **déphasage**  $\Phi$  entre la tension d'entrée et la tension de sortie. Si l'on superpose les deux ondes (entrée et sortie) dans un diagramme temporel, on a le résultat suivant :

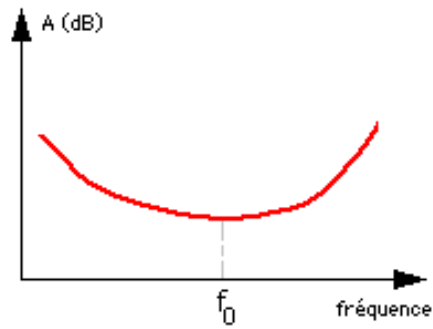


L'affaiblissement A (parfois appelé atténuation) du signal est le rapport des puissances  $P_e/P_s$  du signal émis,  $P_e$ , et du signal reçu,  $P_s$ . Chacune des puissances s'exprime en Watts. Toutefois, on préfère utiliser une échelle logarithmique basée sur la définition du décibel :

$$A(\omega) = 10 \log_{10}(P_e/P_s) \quad (\text{en décibels})$$

La figure ci-contre indique une courbe typique d'affaiblissement en fonction de la fréquence pour une voie de transmission quelconque.

On notera que la fréquence "optimale" est  $f_0$  et que, si l'on souhaite une faible atténuation d'un signal sinusoïdal envoyé, il faudra que celui-ci possède une fréquence proche de  $f_0$ .



Exercices et tests : [QCM3](#), [QCM4](#), [QCM5](#), [QCM6](#)

## Signal quelconque et bande passante

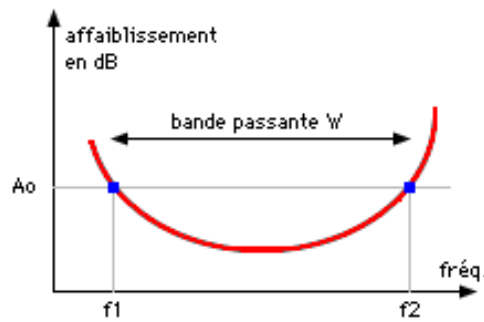
Le théorème de Fourier exprime mathématiquement le fait qu'un signal quelconque peut être considéré comme la superposition d'un nombre fini ou infini de signaux sinusoïdaux. Sans entrer dans les détails mathématiques du théorème, rappelons-en les conséquences pratiques :

- un signal quelconque  $x(t)$  est décomposable en une série de signaux sinusoïdaux
- si le signal est périodique, il peut s'exprimer sous forme d'une série de Fourier ; les termes de la série sont des signaux sinusoïdaux dont les fréquences varient comme multiples d'une fréquence de base  $f_0$
- si le signal n'est pas périodique, il peut s'exprimer sous forme d'une intégrale de Fourier (extension continue de la série de Fourier) ; les signaux sinusoïdaux constituants ont des fréquences continûment réparties

[exemple 1](#)

[exemple 2](#)

Puisqu'un signal quelconque peut être considéré comme la superposition d'une série de signaux sinusoïdaux, on peut imaginer que la transport de ce signal complexe équivaut au transport des signaux sinusoïdaux le composant. Comme leurs fréquences sont différentes, ils seront plus ou moins affaiblis et à l'arrivée, certains d'entre eux ne seront plus discernables. Si on se définit un seuil d'"audibilité"  $A_0$ , tous les signaux sinusoïdaux qui ont une fréquence inférieure à  $f_1$  seront considérés comme perdus ; de même ceux qui ont une fréquence supérieure à  $f_2$  seront aussi considérés comme perdus. Seuls seront perceptibles à l'arrivée, les signaux qui ont une fréquence comprise entre  $f_1$  et  $f_2$ . Cette plage de fréquence est appelée la **bande passante** ou **largeur de bande** de la voie.



Autrement dit, étant donné un signal complexe quelconque, ce signal sera relativement bien transmis si ses composants sinusoïdaux ont des fréquences comprises dans la largeur de bande. On peut aussi remarquer que plus la largeur de bande est grande, meilleur est le signal à l'arrivée ce qui explique pourquoi on est très intéressé à utiliser des voies de transmission avec une grande largeur de bande.

exemple : la largeur de bande de la ligne téléphonique est 3100 Hz car les fréquences vocales sont comprises entre 300 Hz et 3400 Hz.

Exercices et tests : [Exercice 2](#), [Exercice 7](#), [QCM7](#), [QCM8](#)

## Rapidité de modulation et débit binaire

Un message est constitué d'une succession de signaux (analogiques ou numériques) de durée égale  $\Delta$  (moment élémentaire). Ces signaux se propagent sur une voie de transmission à la vitesse de la lumière ( $3 \cdot 10^8$  m/s dans le vide, pratiquement la même valeur dans une fibre optique,  $2 \cdot 10^8$  m/s environ dans des voies filaires métalliques). On peut donc déjà concevoir que la vitesse de propagation n'est pas un facteur contraignant. Le facteur contraignant est la cadence avec laquelle on "met" le signal sur la ligne. Cette cadence est définie par la **rapidité de modulation** :

$$R = 1/\Delta \text{ ( en bauds).}$$

Si le message est binaire, chaque signal transporte  $n$  bits (quantité d'information). On est alors conduit à définir le **débit binaire** :

$$D = nR \text{ (en bits/s)}$$

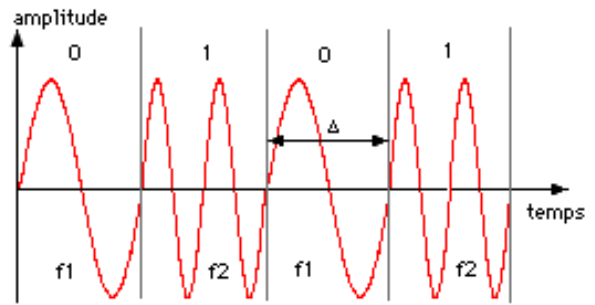
qui correspond à la cadence avec laquelle on "pose" les bits sur la ligne.

exemple : vidéotex (Minitel) :  $R = 1200$  bauds et  $D = 1200$  bits/s. Ceci signifie qu'un signal élémentaire transporte un seul bit. Un écran chargé a un volume approximatif de 2 Ko ; par suite, en négligeant le temps de propagation, le temps approximatif du transport est 13,3 secondes ce qui est important compte tenu du faible volume de l'information transportée.

Examinons quelques situations pour expliciter et illustrer les définitions relatives à la rapidité de modulation et au débit binaire.

exemple 1 : transmission de données numériques par des signaux analogiques ; on utilise deux types de signaux analogiques, chacun ayant une durée  $\Delta$ , l'un possède une fréquence  $f_1$ , l'autre une fréquence  $f_2$  (double de  $f_1$  sur le schéma) : les deux signaux sont aisément discernables. On peut convenir que le premier signal transporte un "0" et que le second transporte un "1". La cadence avec laquelle on envoie les signaux sur une voie est égale à la cadence avec laquelle on transmet les bits puisque chaque signal transporte un bit.

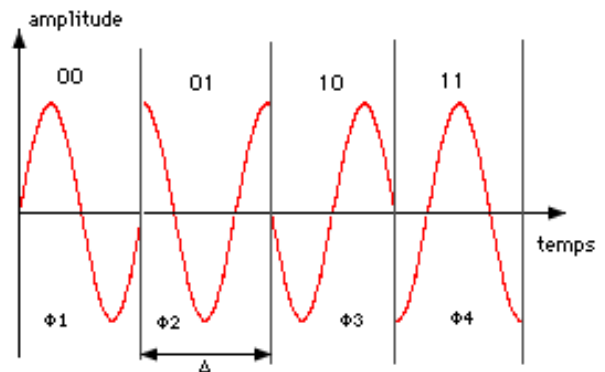
La distinction entre 0 et 1 dépend uniquement de la fréquence du signal sinusoïdal (modulation de fréquence).



$$R = 1/\Delta \quad D = R$$

exemple 2 : transmission de données numériques par des signaux analogiques ; on utilise cette fois 4 types de signaux sinusoïdaux obtenus par déphasage successif de  $\pi/4$ . Chacun des signaux peut transporter deux bits, soit 00, soit 01, soit 10, soit 11. Il en résulte que le débit binaire est le double de la rapidité de modulation.

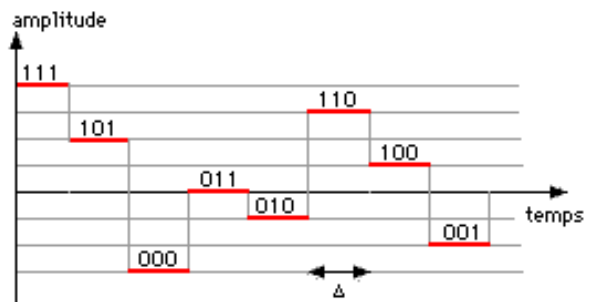
La distinction entre les signaux ne dépend que de la phase du signal sinusoïdal (modulation de phase).



$$R = 1/\Delta \quad D = 2R$$

exemple 3 : transmission de données numériques par des signaux numériques ; imaginons 8 signaux différents par leur amplitude et de même durée  $\Delta$ . Chacun des signaux peut transporter 3 bits puisqu'il existe 8 combinaisons différentes de 3 bits.

La distinction entre les signaux ne dépend que de leur amplitude (modulation d'amplitude).



$$R = 1/\Delta \quad D = 3R$$

Pour une meilleure performance dans la rapidité de transmission, on cherche à améliorer le débit binaire. Puisque  $D = nR$ , on cherchera à augmenter le débit binaire en augmentant

- soit  $n$ , mais le bruit (voir plus loin) est un frein important (difficulté à discerner les différents niveaux)
- soit  $R$ , mais on ne peut dépasser une valeur  $R_{\max}$ .

Ce dernier résultat a été démontré par Nyquist (1928) qui établit un rapport entre la rapidité maximum et la bande passante  $W$  :

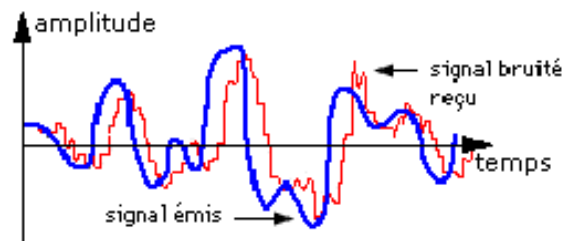
$$R_{\max} = 2W,$$

Ce résultat est théorique et, dans la pratique,  $R_{\max} = 1,25W$

Exercices et tests : [Exercice 4](#), [Exercice 8](#), [QCM9](#)

## Bruit et capacité

Le bruit consiste en signaux parasites qui se superposent au signal transporté et qui donnent, en définitive, un signal déformé;



On distingue 3 types de bruit :

- bruit déterministe (dépend des caractéristiques du support)
- bruit aléatoire (perturbations accidentelles)
- bruit blanc (agitation thermique des électrons)

Le bruit le plus gênant est évidemment le bruit aléatoire. Il peut modifier notablement le signal à certains moments et produire des confusions entre "0" et "1". Pour cette raison, il faut veiller à ce que la puissance du signal soit supérieure à celle du bruit. Le paramètre correspondant est le rapport "signal sur bruit" S/B défini en décibels par :

$$S/B(\text{en décibels}) = 10 \log_{10}(P_S(\text{Watt})/P_B(\text{Watt}))$$

où  $P_S$  et  $P_B$  désignent respectivement les puissances du signal et du bruit.

Le théorème de Shannon (1948) exprime l'importance du facteur S/B : ce facteur limite la quantité  $n$  de bits transporté par chaque signal

$$n_{\max} = \log_2 \sqrt{1 + \frac{P_S}{P_B}}$$

Par suite, en utilisant le théorème de Nyquist, on en déduit le débit maximum d'une voie :

$$C = D_{\max} = R_{\max} n_{\max} = 2W \log_2 \sqrt{1 + \frac{P_S}{P_B}} = W \log_2 \left[ 1 + \frac{P_S}{P_B} \right]$$

$C$ , débit maximum, est la capacité de la voie de transmission.

exemple : voie téléphonique de largeur  $W = 3100$  Hz et de rapport  $S/B = 20$  dB. En utilisant la formule précédente, on calcule la capacité de la voie téléphonique :  $C = 20,6$  Kbits/s environ.

**Exercices et tests :** [Exercice 1](#), [Exercice 3](#), [Exercice 5](#), [Exercice 6](#), [Exercice 9](#), [Exercice 10](#), [Exercice 14](#), [QCM10](#), [QCM11](#)

# Trafic

Le trafic est une notion liée à l'utilisation d'une voie de transmission. Le trafic permet de connaître le degré d'utilisation d'une voie et par conséquent de choisir une voie adaptée à l'utilisation que l'on veut en faire ; il ne servirait à rien, en effet, de posséder des lignes de transmission surdimensionnées, sinon à perdre de l'argent en abonnements.

Pour évaluer le trafic, on considère qu'une transmission ou communication est une session de durée moyenne  $T$  (en secondes) ; soit  $N_c$  le nombre moyen de sessions par heure. L'intensité du trafic est alors donnée par l'expression :

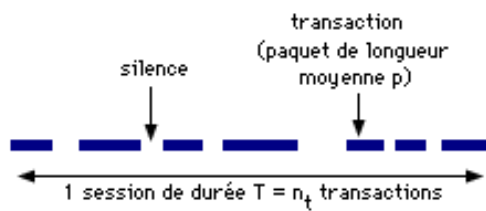
$$E = T N_c / 3600 \text{ ( en Erlangs)}$$

Autrement dit, l'intensité du trafic mesure le temps d'utilisation de la voie par heure.

En fait, une analyse plus fine est quelquefois nécessaire car une session comporte un certain nombre de "silences", notamment dans les applications conversationnelles. On peut distinguer les deux cas extrêmes suivants concernant les types de sessions :

- sessions où  $T$  est pleinement utilisé (rare)
- sessions où  $T$  comprend des "silences"

Dans ce dernier cas, l'intensité du trafic ne donne pas l'occupation réelle du canal. On décompose la session en transactions de longueur moyenne  $p$  en bits, entrecoupées par des silences. Soit  $N_t$  le nombre moyen de transactions par session.



D étant le débit nominal de la voie, le débit effectif de la voie (pour cette utilisation) est :  $d = \frac{N_t p}{T}$

et le taux d'occupation du canal est défini par le rapport :  $\theta = \frac{d}{D}$

exemple : calcul scientifique à distance : l'utilisateur dialogue avec un ordinateur central ;

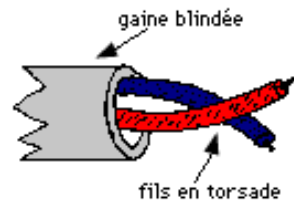
$p = 900$  bits,  $N_t = 200$ ,  $T = 2700$  s,  $N_c = 0.8$ ,  $D = 1200$  b/s d'où  $E = 0.6$  Erlangs     $\theta = 0.05$  (voie utilisée théoriquement à 60% et effectivement à 5%).

Exercices et tests : [Exercice 11](#), [Exercice 12](#), [QCM12](#), [QCM13](#)



# Les supports de transmission

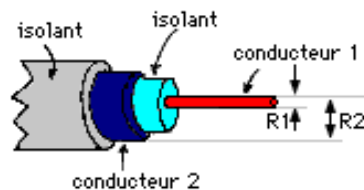
Le support le plus simple est la **paire symétrique torsadée** (UTP : Unshielded Twisted Pairs) . Il s'agit de deux conducteurs métalliques entremêlés (d'où le nom de paire torsadée). Le signal transmis correspond à la tension entre les deux fils. La paire peut se présenter emprisonnée dans une gaine blindée augmentant (comme la torsade) l'immunité contre les perturbations électromagnétiques (STP : Shielded Twisted Pairs).



Pour les paires UTP, nettement moins onéreuses que les paires STP, plusieurs catégories sont définies (de 1 à 5). Les catégories 1 et 2 correspondent à une utilisation en bande étroite, les catégories 3 à 5 (la meilleure) à une utilisation en large bande (100 MHz pour la catégorie 5).

Les deux avantages principaux de ce type de support sont son coût très bas et sa facilité d'installation. Par contre, les inconvénients sont assez nombreux : affaiblissement rapide, sensibilité aux bruits, faible largeur de bande, faible débit. Pour de faibles distances, ce support est relativement utilisé : réseaux locaux, raccordements téléphoniques, notamment.

Le **câble coaxial** constitue une amélioration de la paire torsadée. Ce support constitué de 2 conducteurs à symétrie cylindrique de même axe, l'un central de rayon  $R_1$ , l'autre périphérique de rayon  $R_2$ , séparés par un isolant.



Par rapport à la paire torsadée, le câble coaxial possède une immunité plus importante au bruit et permet d'obtenir des débits plus importants. Une version du câble coaxial, le CATV, est utilisé pour la télévision par câble.

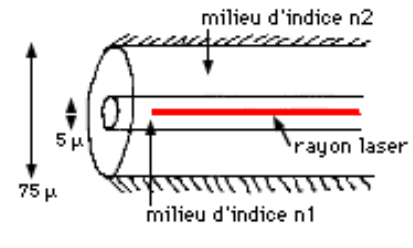
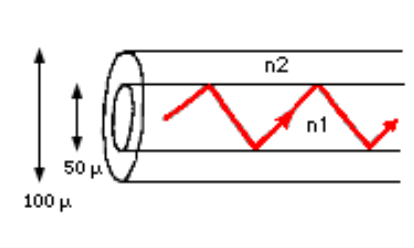
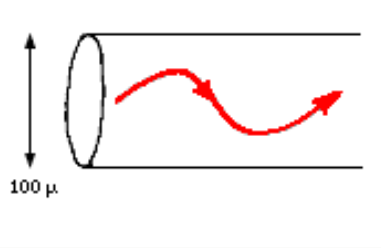
La **fibres optiques** est apparue vers 1972 (invention du laser en 1960). et constitue un domaine en plein développement du fait d'un grand nombre d'avantages :

- faible encombrement : diamètre de l'ordre du 1/10 de mm (les fibres sont en fait groupées en faisceaux)
- légèreté
- largeur de bande de l'ordre du GigaHertz pour des distances inférieures à 1 km ce qui permet un multiplexage composite (TV, HiFi, Téléphone, données informatiques,...)
- faible affaiblissement : à 140 Mbits/s, l'affaiblissement est 3 dB/km pour une longueur d'onde de 0,85 micromètre (régénération tous les 15 km) et de 0,7 dB/km pour une longueur d'onde de 1,3 micromètre (régénération tous les 50 km).
- insensibilité aux parasites électromagnétiques (taux d'erreur approchant  $10^{-12}$ )
- matériau de construction simple et peu coûteux (silice pour les fibres en verre)

Les fibres optiques véhiculent des ondes électromagnétiques lumineuses ; en fait la présence d'une onde lumineuse correspond au transport d'un "1" et son absence au transport d'un "0" ; les signaux électriques sont transformés en signaux lumineux par des émetteurs ; les signaux lumineux sont transformés en impulsions électriques par des détecteurs.. Les émetteurs de lumière sont, soit des LED (Light Emitting Diode ou Diode Electro-Luminescente) classiques, soit des diodes lasers (composants plus délicats). Les détecteurs de lumière sont, soit des photodétecteurs classiques, soit des photodétecteurs à avalanche.

La propagation des signaux lumineux s'effectuent par réflexion sur une surface ; en effet, pour une longueur d'onde donnée et une inclinaison du rayon par rapport à la normale à la surface de séparation entre deux milieux, la lumière incidente se réfléchit totalement (pas de réfraction) ce qui signifie que l'on peut "emprisonner" un ou plusieurs rayons à l'intérieur d'un

milieu tubulaire. En fait, il existe actuellement trois types de fibres optiques ; le premier type est appelé monomode (un seul rayon lumineux par transmission), les deux autres sont multimodes (plusieurs rayons transmis simultanément) :

		
<p><b>fibre monomode</b></p> <p>les indices de réfraction sont tels que <math>n2 &gt; n1</math>. Le rayon laser (longueur d'onde de 5 à 8 micromètres) est canalisé. Cette fibre permet de hauts débits mais est assez délicate à manipuler et présente des complexités de connexion.</p>	<p><b>fibre multimode à saut d'indice</b></p> <p>Les rayons lumineux se déplacent par réflexion sur la surface de séparation (<math>n2 &gt; n1</math>) et mettent plus de temps en déplacement que le rayon de la fibre monomode. L'affaiblissement est de 30 dB/km pour les fibres en verre et de 100 dB/km pour les fibres en matière plastique.</p>	<p><b>fibre multimode à gradient d'indice</b></p> <p>L'indice de réfraction croît depuis centre vers les bords du tube. La réflexion est plus "douce" de ce fait.</p>

Il est possible depuis plusieurs années de multiplexer sur une fibre plusieurs messages numériques se différenciant par la longueur d'onde ; la technologie correspondante s'appelle WDM (Wavelength Division Multiplexing).

La fibre optique possède aussi quelques inconvénients qui tendent cependant à s'amenuiser avec le développement technologique :

- matériels d'extrémité délicats et coûteux
- courbures brusques à éviter
- connexion délicate de deux fibres

Toutefois, du fait de son grand nombre d'avantages, les réseaux utilisent de plus en plus la fibre optique.

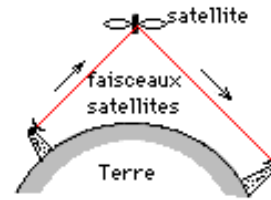
Déjà très utilisées pour la radio et la TV, les ondes électromagnétiques permettent une transmission sans supports matériels. Cette utilisation est dépendante de la fréquence de l'onde.

Pour les besoins de transmission, on peut classer les ondes en deux groupes :ondes non dirigées et ondes dirigées.

- ondes non dirigées : l'émission a lieu dans toutes les directions (inondation) : pas vraiment d'intérêt pour des communications personnalisées, sauf dans le cas de la téléphonie cellulaire (captage par relais). Par contre, pour la diffusion d'informations, l'utilisation est courante (radio, télévision, ....)
- ondes dirigées : les utilisations des ondes dirigées, c'est à dire émise dans une direction particulière, sont principalement les faisceaux hertziens terrestres, les transmission satellite et les réseaux sans fils.

<p><b>faisceaux hertziens terrestres.</b> Les ondes sont émises d'un relais à l'autre en ligne droite. La courbure de la Terre implique une distance maximum entre les relais (tours hertziennes).</p>	
--	--

**transmission satellite.** Le problème de la courbure de la Terre est résolu avec l'utilisation des satellites de télécommunication. Les satellites sont situés sur des orbites géostationnaires et sont donc considérés comme fixes par rapport à la Terre. (distance Terre-satellite : 36 000 km)



**réseaux sans fils :** ces réseaux locaux (WLAN, Wireless Local Area Networks) sont apparus récemment et permettent de s'affranchir des câbles, souvent inesthétiques et surtout peu commodes. Une première catégorie de réseau utilise des ondes dont les longueurs d'ondes sont situées dans l'infra-rouge. Le principe est bien connu puisque les télécommandes infra-rouge sont maintenant d'un usage banal ; les réseaux à infra-rouge permettent un "câblage" intérieur très fonctionnel (à condition d'éviter les obstacles). À l'extérieur, l'utilisation de l'infra-rouge est plus délicate à cause des perturbations électromagnétiques. Une autre catégorie de réseau sans fils est celle des réseaux à ondes lumineuses (laser) ; le faisceau laser est en effet suffisamment fin pour être dirigé vers un capteur ; cette technique est d'ailleurs utilisée pour relier deux bâtiments voisins sans effectuer de câblage "en dur" (émetteurs et détecteurs sur les toits par exemple). Toutefois la transmission par laser peut être affectée par les conditions météorologiques ; par ailleurs elle est encore coûteuse.

# Codage de l'information

Sommaire :

[Numérisation de l'information](#)

[Le texte](#)

[L'image fixe](#)

[Le son et la vidéo](#)

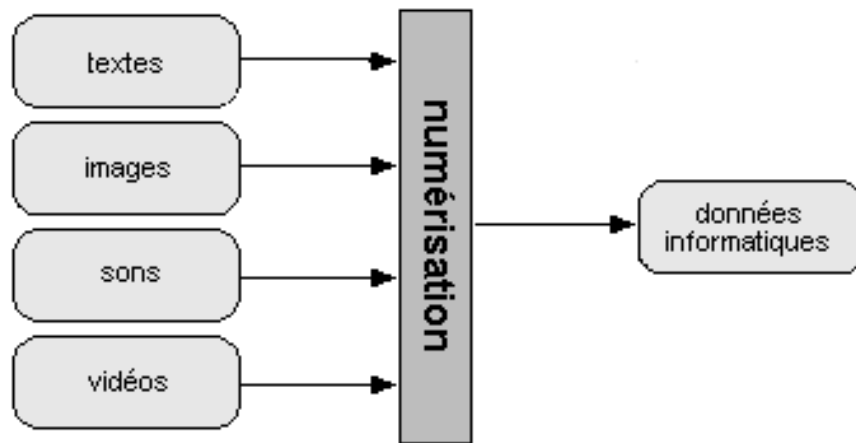
[La protection contre les erreurs](#)

## Numérisation de l'information

L'information existe sous des formes diverses. Pour la manipuler et, en particulier, la transporter, on est amené à la coder.

<b>parole :</b> système : téléphone codeur : microphone décodeur : écouteur transmission : signaux analogiques et numériques	<b>image fixe :</b> système : télécopie codeur : scanner décodeur : interpréteur de fichier transmission : signaux analogiques et numériques
<b>données informatiques :</b> système : réseaux de télé-informatique codeur : contrôleur de communication + ETCD décodeur : contrôleur de communication + ETCD transmission : signaux analogiques ou numériques	<b>télévision :</b> système : diffusion hertzienne codeur : caméra décodeur : récepteur TV + antenne transmission : signaux analogiques ( et bientôt numériques)

De nos jours, l'information est souvent présentée dans des documents composites, comme une page Web, où simultanément peuvent être présentés un texte, une image fixe, un clip vidéo,.... . L'information est, en effet, présentée sous forme multimédia. Chaque type d'information possède son système de codage, mais le résultat est le même : une suite de 0 et de 1. Le transport de l'information consiste alors à transmettre des bits, quelque soit la signification du train de bits transmis.



Dans les paragraphes qui suivent, on examinera comment il est possible de numériser chaque média.

## Le texte

Le premier code relatif au texte est certainement le code Morse, en service bien avant l'utilisation de l'ordinateur. Et pourtant, il s'agit bien d'un code binaire qui aurait pu servir à numériser les textes, puisqu'il est composé de deux symboles seulement : le point et le trait (on pourrait aussi bien dire 0 et 1).

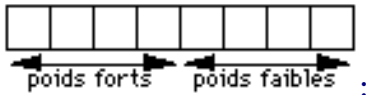
•— A	—••• B	—•—• C	—•• D	• E
••—• F	—•— G	•••• H	•• I	•—•— J
—•— K	•—•• L	—•— M	—• N	—•— O
•—•— P	—•—• Q	•—• R	•••• S	— T
••— U	•••— V	•—• W	—••— X	—•— Y
		—••• Z		

Malheureusement, il souffre de deux inconvénients majeurs :

- il est "pauvre" : peu de caractères peuvent être codés ;
- il utilise des combinaisons de traits et de points de longueur variable ce qui n'est pas commode, notamment pour la numérisation d'éléments ayant des probabilités d'apparition de même ordre.

Pour ces raisons, il n'a pas été utilisé pour le codage numérique de l'information (apparemment, on n'y a peut-être pas pensé !) ; toutefois, compte tenu de son utilisation passée, il méritait d'être mentionné.

Si on se fixe comme règle de trouver un code permettant de représenter numériquement chaque caractère de manière à obtenir un nombre de bits fixe, il est simple de comprendre qu'avec un code à  $p$  positions binaires on pourra représenter  $2^p$  caractères. Effectivement, dans le passé, on a utilisé de tels codes, généralement en les définissant par des tables, le code étant divisé en poids faibles et en poids forts :



- code à 5 positions : un de ses représentants est ATI (Alphabet Télégraphique International, utilisé par le Télec)
- code à 6 positions : ISO6 (ce code très employé sur les premiers ordinateurs est aujourd'hui abandonné)
- code à 7 positions : ASCII : la panacée universelle

		poids forts							
		000	001	010	011	100	101	110	111
poids faibles	0000	NUL	DLE	SP	0	@	P	\	p
	0001	SOH	DC1	!	1	A	Q	a	q
	0010	STX	DC2	"	2	B	R	b	r
	0011	ETX	DC3	#	3	C	S	c	s
	0100	EOT	DC4	\$	4	D	T	d	t
	0101	ENQ	NAK	%	5	E	U	e	u
	0110	ACK	SYN	&	6	F	V	f	v
	0111	BEL	ETB	,	7	G	W	g	w
	1000	BS	CAN	(	8	H	X	h	x
	1001	HT	EM	)	9	I	Y	i	y
	1010	LF	SUB	*	:	J	Z	j	z
	1011	VT	ESC	+	;	K	[	k	{
	1100	FF	FS	'	<	L	]	l	~
	1101	CR	GS	-	=	M	^	m	}
	1110	SO	RS	.	>	N	_	n	~
	1111	SI	US	/	?	O	<--	o	DEL

code ASCII

ce code fait apparaître des caractères non imprimables appelés caractères de manœuvre qui provoquent des actions sur des dispositifs informatiques ou qui transportent de l'information de service. Par exemple, FF signifie "passage à la page suivante" ce qui pour une imprimante est une information indispensable.

- code à 8 positions : ASCII étendu, EBCDIC

Le code ASCII est un code sur 7 positions ; comme les ordinateurs stockent l'information dans des mots dont la longueur est un multiple de 8 bits (octets), on complète généralement le code ASCII par un "0" en tête pour former un octet. On peut aussi utiliser ce degré de liberté supplémentaire pour définir des alphabets spéciaux ; dans ce cas, on avertit en mettant un "1" en tête à la place du "0" ce qui correspond au code ASCII étendu ; malheureusement, il y a plusieurs codes ASCII étendus car il n'y a pas encore de normalisation imposée ce qui rend difficile mais pas insurmontable le passage d'un document d'une plate-forme à une autre.

Le code EBCDIC est d'emblée un code sur 8 bits ce qui permet d'obtenir 256 caractères représentables contre 128 pour le code ASCII. Il a été utilisé par IBM pour le codage de l'information sur ses machines. Il n'a pas atteint toutefois la popularité du code ASCII.

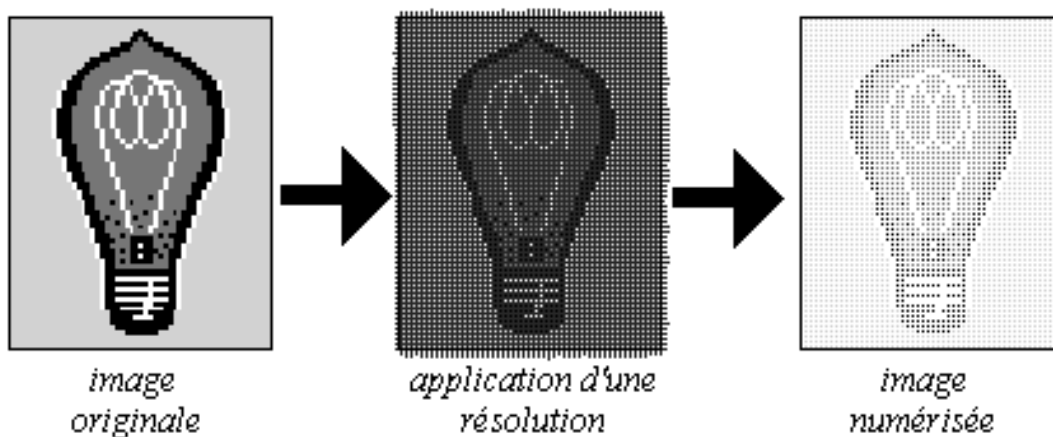
- code à 16 positions : [Unicode](#)

Ce code est récent et a été mis en oeuvre pour satisfaire tous les usagers du Web. Il incorpore presque tous les alphabets existants (Arabic, Armenian, Basic Latin, Bengali, Braille, Cherokee, etc....) ; il est compatible avec le code ASCII. Par exemple le caractère latin A est codé 0x41 en ASCII et U+0041 en Unicode ; le caractère monétaire € est codé 0x80 en ASCII étendu et U+20AC en Unicode.

*Exercices et tests : [QCM14](#)*

## L'image fixe

L'image numérique est usuellement une image décrite en termes de lignes et chaque ligne en terme de points. Une image VGA de résolution 640x480 signifie que l'image est une matrice de 480 lignes, chaque ligne comportant 640 points ou pixels. Une image est alors représentée par un fichier donnant la liste des points ligne par ligne, colonne par colonne.



Un pixel est codé suivant la qualité de l'image :

- image en noir et blanc (image binaire) : un seul bit suffit pour coder le point (0 pour noir, 1 pour blanc) ;
- image en 256 nuances de gris : chaque point est représenté par un octet (8 bits) ;
- image en couleur : on montre que la couleur peut être exprimée comme une combinaison linéaire de trois couleurs de base, par exemple Rouge(R), Vert(V), Bleu(B). Ainsi une couleur quelconque x est exprimée comme

$$x = aR + bV + cB$$

où a, b, c sont des doses de couleurs de base. Usuellement, une bonne image correspond à des doses allant de 0 à 255. Par suite une image couleur de ce type peut être représentée par 3 matrices (une par couleur de base) dont chacune d'elle possède des éléments sur 8 bits, ce qui au total fait 24 bits par pixel.

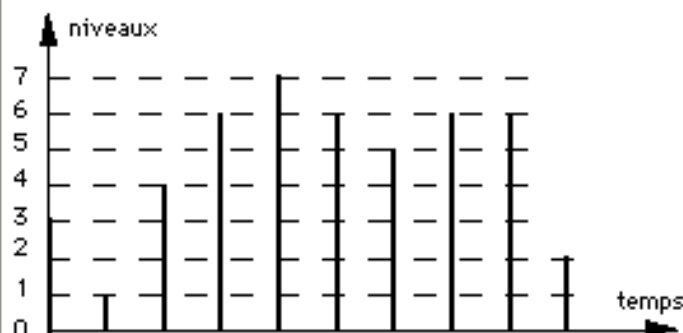
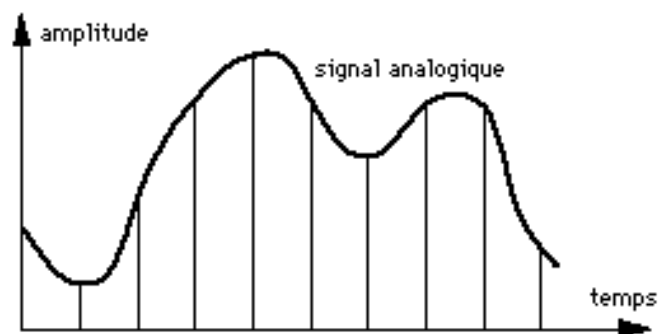
On se rend vite compte du volume atteint pour des images importantes et de bonne définition. Une image 640x480 en couleur (24 bits) occupe un volume de 921 600 octets. On est alors amené à utiliser des techniques de **compression** pour réduire la taille des fichiers d'images. Une des premières techniques est l'emploi de codes de Huffman qui emploie des mots codés de longueur variable : long pour les niveaux de couleur rares, court pour les niveaux de couleur fréquents. Ce type de codage est dit sans perte puisque la compression ne dénature pas l'information. D'autres méthodes permettent d'obtenir des résultats plus performants en terme de réduction de volume ; dans cette catégorie, dite compression avec perte, des détails peu pertinents de l'image disparaissent ; c'est notamment le cas du standard JPEG qui utilise des transformations en cosinus discrets appliquées à des sous-images.

## Le son et la vidéo

Les données de type son et vidéo sont à l'origine analogique sous forme de signaux (un ou deux signaux pour le son, 3 signaux pour la vidéo-image). Ces signaux analogiques sont numérisés de la manière suivante :

### 1 - Échantillonnage

Le signal est échantillonné : à une fréquence donnée f, on mesure la hauteur du signal. On obtient alors une séquence de mesures.



### 2 - Quantification

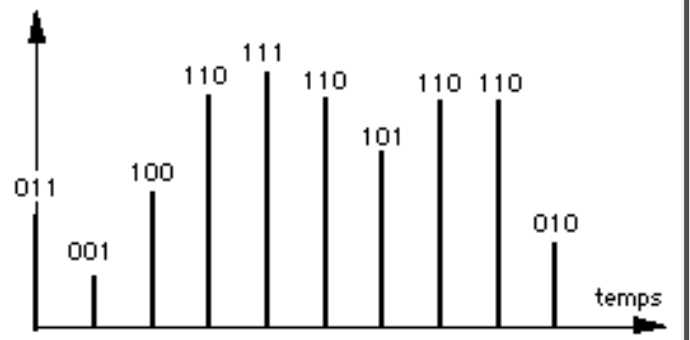
On se fixe une échelle arbitraire de valeurs (usuellement suivant une puissance de 2:  $2^p$  valeurs) et on fait correspondre chaque mesure à une valeur dans cette échelle. on est évidemment conduit à faire des approximations ce qui correspond à un bruit dit de quantification



### 3- Codage

Chaque valeur est transformée en sa combinaison binaire, la suite de ces combinaisons étant placée dans un fichier.

011001100110111110101110110010.....



Le volume des fichiers obtenus après numérisation dépend crucialement de la fréquence d'échantillonnage  $f$  et de la valeur de  $p$  (longueur du codage de chaque valeur). La fréquence d'échantillonnage, en particulier, ne peut être choisie arbitrairement. Les résultats en traitement de signal indiquent que la fréquence d'échantillonnage d'un signal doit au moins être le double de la plus grande des fréquences du signal (c'est à dire la plus grande de toutes celles des composantes sinusoïdales - développement de Fourier - composant le signal).

exemple : la parole est transmise usuellement par le réseau téléphonique. Elle correspond à des signaux analogiques dont la fréquence varie entre 300 Hz et 3400 Hz. La plus grande des fréquences est donc 3400 Hz que l'on arrondit à 4000 Hz par précaution. La fréquence d'échantillonnage doit donc être au moins de 8000 Hz. Si l'on choisit cette fréquence d'échantillonnage et si l'on décide de coder sur 8 bits chaque échantillon (cela est suffisant pour la parole), on obtient pour une seconde de parole un volume de 64 000 bits ; une transmission en temps réel de la parole nécessite donc des liaisons à un débit de 64 Kbits/s. C'est notamment le cas du RNIS français (Numéris) qui propose des canaux à 64 Kbits/s.

Comme dans le cas de l'image fixe, mais de manière extrêmement amplifiée, les volumes obtenus sont considérables et il est nécessaire, pour leur stockage comme pour leur transport, de les compresser. Les techniques diffèrent ici, suivant que l'on a un fichier son ou un fichier vidéo.

Pour le son, le système de codage explicité plus haut (codage sur  $n$  bits de chaque échantillon) est appelé PCM (Pulse Code Modulation). Il est possible de réduire le volume avec les codages suivants :

- MPCM (Delta PCM) où le codage porte sur les différences entre les valeurs successives échantillonnées,
- ADPCM (Adaptive Differential PCM) où des interpolations sont effectuées afin de diminuer le volume.

La problématique du son (et aussi de la vidéo) est une transmission en "temps réel" ; il est donc nécessaire d'utiliser des systèmes de codage ou **codecs** performants. Les codecs audio sont décrits par des normes standards de l'ITU dont voici quelques exemples :

- codec G.711 : algorithme de codage : PCM ; échantillonnage à 8 KHz, débit nécessité : 64 Kbits/s ;
- codec G.722 : algorithme de codage : ADPCM ; échantillonnage à 7 KHz ; débit nécessité : 64 Kbits/s ;
- codec G.723 : algorithmes de codage MP-MLQ (MultiPulse Maximum Likelihood Quantization) et ACELP (Algebraic Code-Excited Linear Prediction) ; échantillonnage à 8 KHz ; débit nécessité entre 5,3 et 6,3 Kbits/s ;

Pour la vidéo, divers procédés de codage sont employés dans le but de réduire le volume des fichiers. Le plus connu correspond à la série de normes MPEG. Le principe de compression s'appuie sur trois types d'images :

- les images "intra" sont des images peu compressées qui servent de repère (une image intra pour 10

images successives) ;

- les images "prédites" sont des images obtenues par codage et compression des différences avec les images intra ou prédites précédentes (une image prédite toutes les trois images) ;
- les images "interpolées" sont calculées comme images intermédiaires entre les précédentes.

L'utilisation de vidéos numériques MPEG nécessite la présence d'une carte de décompression dans le micro-ordinateur d'exploitation. Les principaux standards sont MPEG1 (débit nécessité : 1,5 Mbits/s), MPEG2 (débit nécessité : 4 à 10 Mbits/s), MPEG4 (débit nécessité : 64 Kbits/s à 2 Mbits/s).

**Exercices et tests :** [Exercice 15](#), [Exercice 21](#), [Exercice 35](#), [Exercice 40](#), [QCM21](#)

## Protection contre les erreurs

Lors de la transmission d'un train de bits, des erreurs peuvent se produire, c'est à dire qu'un "1" peut être transformé en un "0" ou réciproquement.

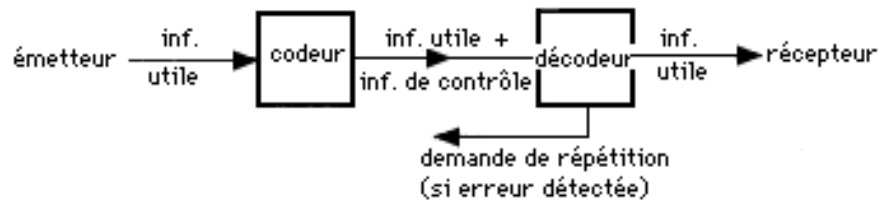
On définit le taux d'erreur par le rapport :

$$\tau = \frac{\text{nombre de bits erronés}}{\text{nombre total de bits}}$$

L'ordre de grandeur du taux d'erreur est de  $10^{-5}$  à  $10^{-8}$ . Suivant le type d'application, une erreur peut avoir des conséquences importantes et c'est pourquoi il convient souvent de mettre en oeuvre des dispositifs permettant de détecter les erreurs et si possible de les corriger. Il convient de noter à ce sujet que le taux d'erreur dépend de la qualité du support de transmission (notamment son immunité au bruit).

Les statistiques indiquent que 88% des erreurs proviennent d'un seul bit erroné, c'est à dire que ce bit erroné est entouré de bits corrects ; 10% des erreurs proviennent de deux bits adjacents erronés. On voit donc que le problème prioritaire à résoudre est la détection d'un seul bit erroné et, si possible, sa correction automatique.

Dans cet ordre d'idées, on utilise des codes détecteurs d'erreurs : l'information utile est encodée de manière à lui ajouter de l'information de contrôle ; le récepteur effectue le décodage et à l'examen de l'information de contrôle, considère que l'information est correcte ou erronée ; dans le dernier cas, une demande de répétition de la transmission est effectuée.



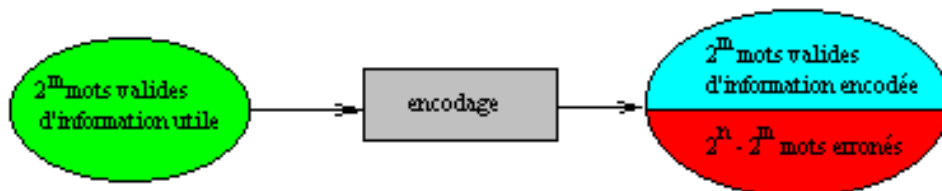
Les codes détecteurs d'erreurs se classent en 2 catégories :

- **codes en bloc** : l'information de contrôle et l'information utile forment un tout consistant. Si le bloc est composé de deux parties distinctes (information utile et information de contrôle) le code est dit **systématique**.



- **codes convolutionnels** ou **récurrents** : la détection des erreurs dans un bloc dépend des blocs précédents. Ils ne seront pas étudiés ici.

Une notion importante dans la recherche de codes détecteurs ou correcteurs est celle de distance de Hamming. Considérons une information utile constituée de mots de  $m$  bits : on peut donc construire  $2^m$  mots distincts au total. Définissons l'information de contrôle sous la forme de  $r$  bits déduits de manière unique à partir des  $m$  bits utiles. L'information "habillée" en résultant est constituée de  $n=m+r$  bits et, compte-tenu de l'unicité de la définition des bits de contrôle, on a au total  $2^m$  mots valides de  $n$  bits. Cependant, avec  $n$  bits, on peut avoir  $2^n$  mots différents. La différence  $2^n - 2^m$  indique le nombre de mots erronés.

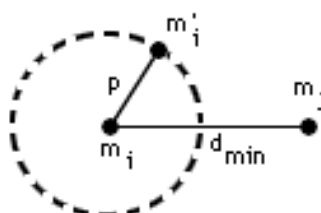


La distance de Hamming de deux mots :  $d(m_1, m_2)$  est le nombre de bits différents de même rang

exemple :  $m_1 = 10110010$        $m_2 = 10000110$        $d(m_1, m_2) = 3$

2 mots de code seront d'autant moins confondus que leur distance de Hamming sera plus grande ; on peut définir une distance minimum  $d_{\min}$  ; si  $d(m_1, m_2) < d_{\min}$ , alors  $m_2$  est une copie erronée de  $m_1$ .

$m_i$  et  $m_j$  sont des mots du code ;  $m'_i$  est une copie erronée de  $m_i$



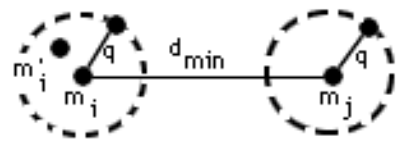
d'où la règle 1 :

Pour détecter  $p$  erreurs, il faut que  $d_{\min} > p$

exemple : détection des erreurs simples :  $d_{min} > 2$

Intéressons-nous maintenant à la correction des erreurs jusqu'à un ordre  $q$  ; chaque mot de code et ses copies "admissibles" doivent être dans des sphères non sécantes :

$m_i$  et  $m_j$  sont des mots du code ;  
 $m'_i$  est un mot erroné qui doit être assimilé à  $m_i$ .



d'où la règle 2 :

Pour corriger des erreurs jusqu'à l'ordre  $q$ , il faut que  $d_{min} > 2q$

exemple : la correction des erreurs simples nécessite  $d_{min} > 2$

### codes linéaires

Un code linéaire est un code en bloc systématique  $(n,m)$  dans lequel les  $r = n - m$  bits de contrôle dépendent linéairement des  $m$  bits d'information. Soit l'information utile représentée par le vecteur ligne  $\tilde{X} = (x_1 \ x_2 \dots \ x_m)$  ; l'information codée est représentée par le vecteur ligne  $\tilde{Y} = (y_1 \ y_2 \dots \ y_m \ y_{m+1} \dots \ y_{m+r})$  avec

$$y_1 = x_1 \quad y_2 = x_2 \quad \dots \quad y_m = x_m \quad y_{m+1} = a_1 \quad y_{m+2} = a_2 \quad \dots \quad y_{m+r} = a_r$$

où les  $a_i$  sont les bits de contrôle, donc  $\tilde{Y} = (x_1 \ x_2 \dots \ x_m \ a_1 \ a_2 \dots \ a_r)$ .

Le code est alors simplement défini par la relation matricielle  $\tilde{Y} = \tilde{X} \cdot G$  où  $G$  est la matrice génératrice du code. La forme générale de  $G$  est :

$$G = (\mathbf{1}, \mathbf{g}) = \begin{bmatrix} 1 & 0 & \dots & 0 & g_{11} & g_{12} & \dots & g_{1r} \\ 0 & 1 & \dots & 0 & g_{21} & g_{22} & \dots & g_{2r} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & g_{m1} & g_{m2} & \dots & g_{mr} \end{bmatrix}$$

$\xleftarrow{\hspace{1.5cm} n \hspace{1.5cm} \xrightarrow{\hspace{1.5cm}}$   
 $\xleftarrow{\hspace{1.5cm} m \hspace{1.5cm} \xrightarrow{\hspace{1.5cm} r \hspace{1.5cm} \xrightarrow{\hspace{1.5cm}}$   
 $\updownarrow m$

d'où les bits de contrôle :  $a_i = x_1g_{1i} + x_2g_{2i} + \dots + x_mg_{mi}$ .

exemple : code (6,3) avec

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

information utile :  $\tilde{X} = (x_1 \ x_2 \ x_3)$  donc 8 mots possibles

information codée :  $\tilde{Y} = (x_1 \ x_2 \ x_3 \ a_1 \ a_2 \ a_3)$

La relation  $\tilde{Y} = \tilde{X} \cdot G$  conduit à

$$a_1 = x_2 + x_3$$

$$a_2 = x_1 + x_3$$

$$a_3 = x_1 + x_2$$

Les mots du code sont :

000000	001110	010101	011011
100011	101101	110110	111000

On constate que  $d_{\min} = 3$  ce qui permet la correction des erreurs simples et la détection des erreurs doubles

exemple : code (8,7) avec

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

information utile :  $\tilde{X} = (x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7)$

information codée :  $\tilde{Y} = (x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ a_1)$

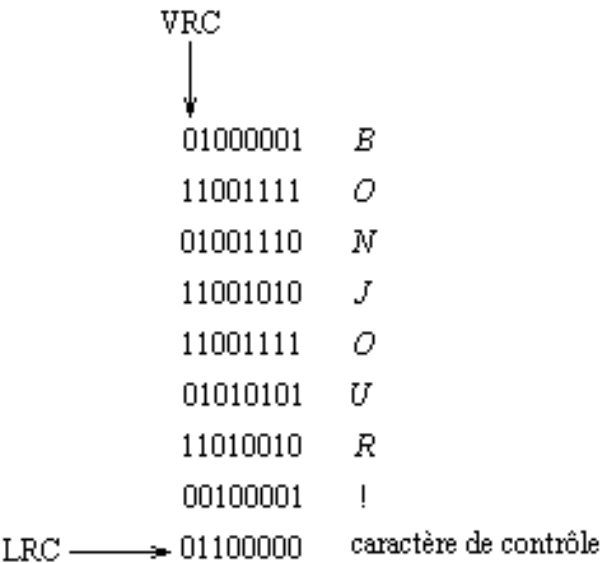
La relation  $\tilde{Y} = \tilde{X} \cdot G$  conduit à  $a_1 = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$  (modulo 2)

$a_1$  est appelé bit de parité : les mots du code ont un nombre pair de 1.

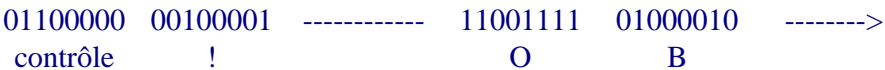
On pourra ainsi représenter des caractères sur 8 bits avec 7 bits relatifs au code ASCII et le huitième bit étant le bit de parité (que l'on peut placer, bien sûr, où l'on veut ; la coutume est de le placer en tête) :

01000001	<i>B</i>
11001111	<i>O</i>
01001110	<i>N</i>
11001010	<i>J</i>
11001111	<i>O</i>
01010101	<i>U</i>
11010010	<i>R</i>
00100001	!
↑	
bit de parité	

Avec ce système, 2 caractères différant par 1 du code ASCII diffèrent aussi par le bit de parité donc  $d_{\min} = 2$ . Ce code ne permet donc que la détection des erreurs simples. On peut améliorer la protection contre les erreurs en effectuant également un contrôle de parité "longitudinal" par opposition au contrôle de parité précédent appelé "vertical" (LRC = Longitudinal Redundancy Check ; VRC = Vertical Redundancy Check) en ajoutant un caractère de contrôle tous les b blocs :



La transmission série des blocs sera donc :



Avec ce système, deux groupes de blocs différant par 1 bit d'information utile diffèrent aussi par le bit VRC, par le bit LRC et par le bit LRC+VRC. On a donc  $d_{\min} = 4$  ce qui permet la détection des erreurs simples et doubles et la correction des erreurs simples.

• codes polynômiaux

Les codes polynômiaux sont des codes linéaires systématiques qui permettent la détection des erreurs. Ils sont très utilisés dans les procédures actuelles de transmission de données. Soit un message de m bits utiles :

$$\tilde{X} = (x_0 \ x_1 \ ..... \ x_{m-1})$$

où la numérotation des bits est quelque peu différente de celle utilisée jusqu'à présent (mais traditionnelle dans l'utilisation des codes polynômiaux).. Au message X, on associe le polynôme :

$$X(z) = x_0 + x_1 \ z + x_2 z^2 + ..... + x_{m-1} \ z^{m-1}$$

De tels polynômes peuvent être ajoutés (modulo 2) et multipliés suivant les règles booléennes. Un code polynomial est un code linéaire systématique tel que chaque mot du code est représenté par des polynômes Y(z) multiples d'un polynôme H(z) appelé polynôme générateur :

$$Y(z) = Q(z).H(z)$$

Examinons comment on passe de l'information utile (m bits) représentée par un polynôme  $X(z)$  à l'information codée (n bits) représentée par le polynôme  $Y(z)$ . On définira donc un code polynomial (n,m) et on ajoutera à l'information utile  $r = n - m$  bits de contrôle. On pose :

$$X(z) = x_0 + x_1 z + x_2 z^2 + \dots + x_{m-1} z^{m-1}$$

$$H(z) = h_0 + h_1 z + h_2 z^2 + \dots + z^r \text{ polynôme générateur de degré } r$$

Le polynôme  $z^r X(z)$  est un polynôme de degré  $m + r - 1 = n - 1$ . Il comporte n termes dont les r premiers sont nuls. Effectuons la division polynomiale de  $z^r X(z)$  par  $H(z)$  :

$$z^r X(z) = Q(z).H(z) + R(z)$$

où  $R(z)$  est un polynôme de degré  $r-1$ , reste de la division. Puisque l'addition modulo 2 est identique à la soustraction modulo 2, on a

$$Y(z) = Q(z).H(z) = z^r X(z) + R(z)$$

$Y(z)$  est le polynôme associé au mot-code. Il comporte n termes et est de degré  $n-1$ .

exemple : code polynomial (7,4), de polynôme générateur  $H(z) = 1 + z + z^3$ . Une information utile correspond au polynôme  $X(z) = x_0 + x_1 z + x_2 z^2 + x_3 z^3$ . La division de  $z^r X(z)$  par  $H(z)$  conduit aux résultats suivants :

$$Q(z) = x_3 z^3 + x_2 z^2 + (x_1 + x_3)z + x_0 + x_2 + x_3$$

$$R(z) = (x_1 + x_2 + x_3)z^2 + (x_0 + x_1 + x_2)z + x_0 + x_2 + x_3$$

$$Y(z) = x_3 z^6 + x_2 z^5 + x_1 z^4 + x_0 z^3 + (x_1 + x_2 + x_3)z^2 + (x_0 + x_1 + x_2)z + x_0 + x_2 + x_3$$

d'où le mot de code  $(x_3 \ x_2 \ x_1 \ x_0 \ a_2 \ a_1 \ a_0)$  avec

$$a_2 = x_1 + x_2 + x_3$$

$$a_1 = x_0 + x_1 + x_2$$

$$a_0 = x_0 + x_2 + x_3$$

d'où la matrice G du code

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Les principaux codes polynomiaux utilisés en téléinformatique sont :

- code CCITT V41, polynôme générateur  $H(z) = z^{16} + z^{12} + z^5 + 1$  ; utilisation dans la procédure HDLC
- code CRC 16, polynôme générateur  $H(z) = z^{16} + z^{15} + z^2 + 1$  ; utilisation dans la procédure BSC, avec codage EBCDIC
- code CRC 12, polynôme générateur  $H(z) = z^{12} + z^{11} + z^3 + z^2 + z + 1$  ; utilisation dans la procédure BSC, avec codage sur 6 bits
- code ARPA, polynôme générateur  $H(z) = z^{24} + z^{23} + z^{17} + z^{16} + z^{15} + z^{13} + z^{11} + z^{10} + z^9 + z^8 + z^5 + z^3 +$

1

- code Ethernet, polynôme générateur  $H(z) = z^{32} + z^{26} + z^{23} + z^{22} + z^{16} + z^{12} + z^{11} + z^{10} + z^8 + z^7 + z^5 + z^4 + z^2 + z + 1$

cas particulier : Un **code cyclique** est un code polynomial (n,m) tel que son polynôme générateur  $H(z)$  divise  $z^n + 1$

$$z^n + 1 = H(z)\Omega(z)$$

où  $\Omega(z)$  est un polynôme de degré n. Les codes cycliques possèdent la propriété fondamentale suivante : une permutation circulaire d'un mot du code est un mot du code.

**Exercices et tests :** [Exercice 22](#), [Exercice 23](#), [Exercice 24](#), [Exercice 25](#), [Exercice 26](#), [Exercice 27](#), [Exercice 28](#), [Exercice 29](#), [Exercice 30](#), [Exercice 31](#), [QCM24](#), [QCM25](#), [QCM26](#), [QCM27](#), [QCM28](#), [QCM29](#), [QCM30](#)



# Modes de transmission

Sommaire :

[Transmissions parallèle et série](#)

[Modes d'exploitation d'une voie de transmission](#)

[Transmissions asynchrone et synchrone](#)

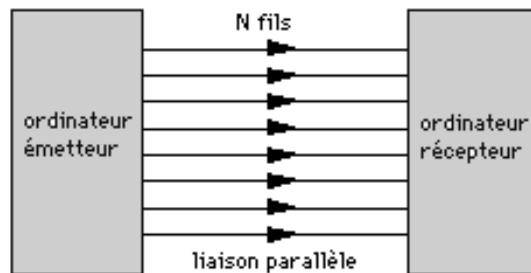
[Transmission par signaux numériques](#)

[Modulation et démodulation](#)

## Transmissions parallèle et série

- transmission parallèle

Les ordinateurs manipulent non pas des bits isolés, mais des mots de plusieurs bits aussi bien pour le calcul que pour le stockage. On est donc conduit à imaginer un système de transport dans lequel les différents bits d'un mot sont véhiculés en parallèle. Cela implique que pour des mots de N bits il faut N lignes de transmission.



Cette possibilité comporte des inconvénients évidents :

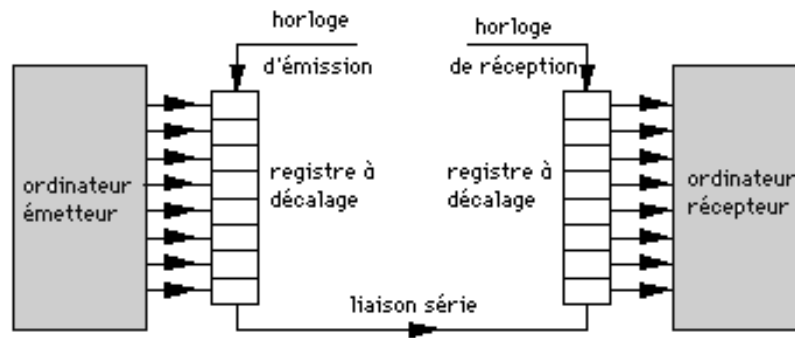
- les lignes nécessitent une masse métallique délicate à grande distance
- non synchronisation des bits transportés à grande distance

Pour ces raisons, à grande distance, la transmission parallèle n'est pas employée ; elle peut l'être, par contre, entre un ordinateur et des périphériques proches (imprimante parallèle par exemple).

Une autre possibilité, plus sophistiquée, est la transmission parallèle de signaux sur des canaux de fréquences différentes ; en fait, comme on le verra plus loin, cette possibilité correspond au multiplexage en fréquence.

- transmission série

Dans ce mode, les bits sont transmis les uns derrière les autres, ce qui nécessite une "sérialisation" effectuée par une logique de transmission dont la pièce maîtresse n'est autre qu'un registre à décalage dont le fonctionnement est rythmé par une horloge.



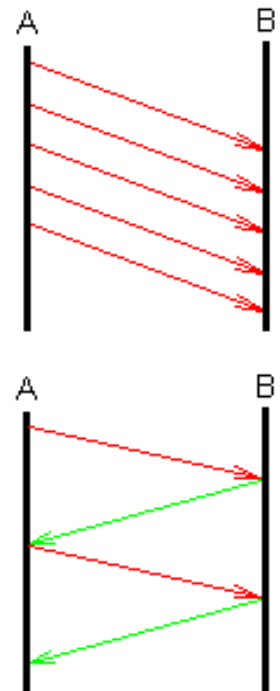
Une difficulté majeure de ce mode de transmission est liée à l'horloge ; en effet, il est nécessaire d'employer une horloge d'émission et une horloge de réception qui doivent fonctionner en synchronisme parfait.

Exercices et tests : [QCM15](#)

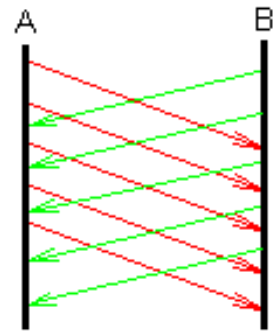
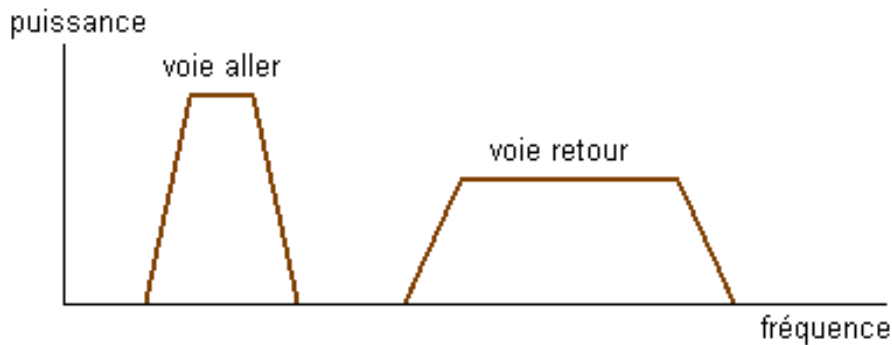
## Modes d'exploitation d'une voie de transmission

Trois modes d'exploitation peuvent être définis sur une liaison point à point reliant deux stations émettrices/réceptrices:

- mode simplex : l'une des stations émet et l'autre reçoit. La communication est donc unidirectionnelle pure.
- mode semi-duplex (half duplex ou alternatif) : la communication est unidirectionnelle, mais le sens de transmission change alternativement : une station émet, l'autre reçoit ; puis c'est la station réceptrice qui devient émettrice et réciproquement ; etc...



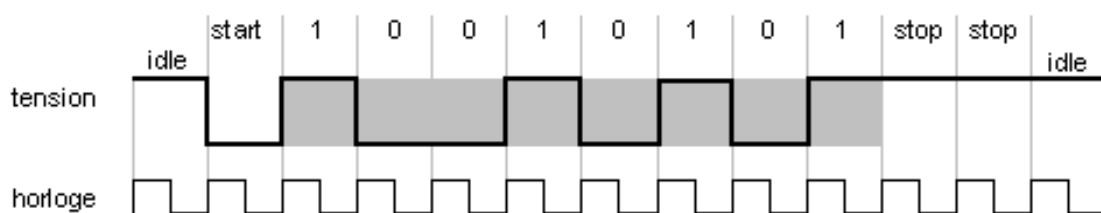
- **mode duplex (full duplex) :** les deux stations peuvent émettre et recevoir simultanément. Un moyen répandu (mais pas le seul) de permettre cette transmission à double sens est le multiplexage en fréquence : la plage de fréquence comporte deux bandes, l'une pour un sens, l'autre pour l'autre sens :



## Transmissions asynchrone et synchrone

- **transmission asynchrone**

Elle consiste en la transmission d'une succession de blocs courts de bits (1 caractère - en grisé sur la figure ci-dessous) avec une durée indéfinie entre l'envoi de deux blocs consécutifs. Un bit START annonce le début du bloc ( polarité inverse de celle de la ligne au repos - idle), un ou deux bits STOP annoncent la fin du bloc (polarité inverse de celle du bit STOP). Un bit de parité est



Pour ce type de transmission, les débits sont normalisés :

- blocs de 11 bits : 110 b/s ;
- blocs de 10 bits : 300, 600, 1200, 2400, 3600, 4800, 9600, 19200 b/s.

- **transmission synchrone**

Ce type de transmission est bien adapté aux données volumineuses et aux nécessités de transmission rapide. L'information est transmise sous la forme d'un flot continu de bits à une cadence définie par l'horloge d'émission. Le flot de bits est réparti cependant en trames qui peuvent être de longueur variable ou de longueur fixe. Les trames doivent être précédées d'un motif de bits annonçant un début de trame et, éventuellement se terminer par un motif analogue. Ce motif de bits ne doit pas évidemment être confondu avec une portion de la zone de données. On emploie à cet effet la technique du bit-stuffing que nous expliquons sur un cas particulier.

exemple : la procédure synchrone HDLC emploie des trames débutant par le drapeau 01111110 et finissant par le même drapeau. Pour éviter que ce motif ne se retrouve à l'intérieur de la trame, on convient de remplacer chaque groupe de cinq "1" successifs par 111110 ; à la lecture, chaque fois que l'on trouvera le motif 111110, on enlèvera le "0".

Comme nous l'avons déjà signalé, l'horloge de réception doit être synchrone avec l'horloge d'émission. Pour résoudre ce problème on peut envisager deux solutions :

- solution 1 (mauvaise) : transmettre sur deux canaux parallèles l'information et l'horloge ; cette solution est à rejeter car en dehors du fait qu'elle nécessite une bande passante non négligeable, sur longue distance, les signaux des deux canaux se désynchronisent.
- solution 2 (bonne) : intégrer l'horloge à l'information : emploi d'un encodage particulier comme on le verra plus loin.

**Exercices et tests :** [Exercice 18](#), [QCM16](#)

## Transmission par signaux numériques

Après numérisation de l'information, on est confronté au problème de la transmission des "0" et des "1". Une première possibilité est l'utilisation de signaux numériques ce qui paraît logique (on verra que des signaux analogiques peuvent aussi convenir).

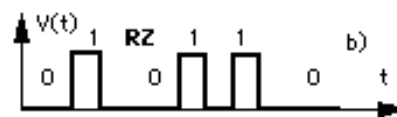
Il s'agit donc de faire correspondre un signal numérique pour le "0" et un autre signal numérique pour le "1". Il y a plusieurs manières de procéder. Nous donnons ci-dessous quelques exemples (du plus simple vers le plus compliqué).

- codes NRZ (Non Retour à Zéro), RZ (Retour à Zéro), bipolaire NRZ et RZ

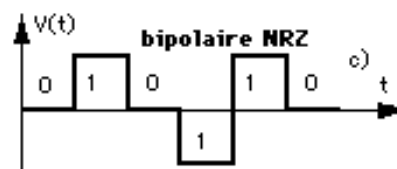
a) NRZ : le codage est simple : un niveau 0 pour le "0", un niveau  $V_0$  pour le "1"



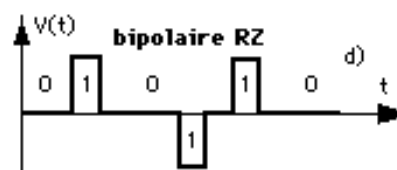
b) RZ : chaque "1" est représenté par une transition de  $V_0$  à 0.



c) bipolaire NRZ : alternativement, un "1" est codé positivement, puis négativement

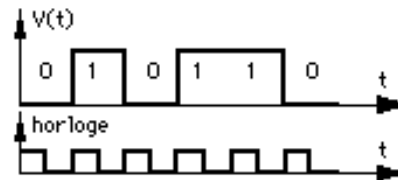


d) bipolaire RZ : même traitement que précédemment.

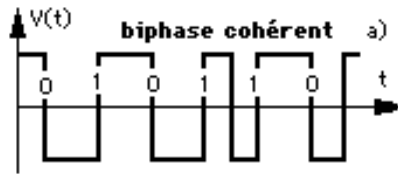


- codes biphasés : le signal d'horloge et le signal de données sont convolués.

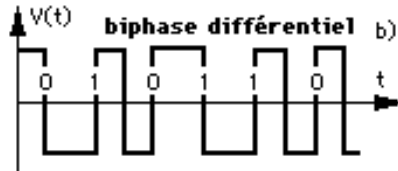
ces codes sont définis sur le schéma ci-contre par comparaison au codage NRZ



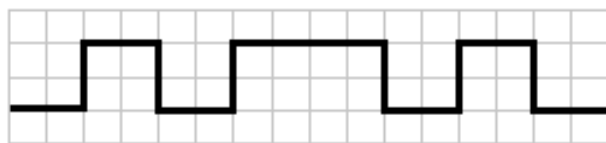
a) codage biphasé cohérent ou Manchester : le "0" est représenté par une transition positive-négative et le "1" par une transition négative-positive.



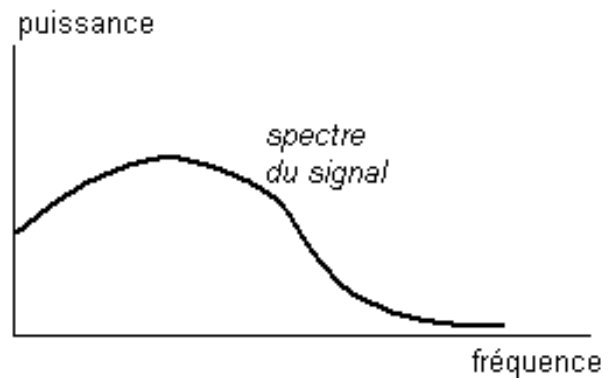
b) codage biphasé différentiel : saut de phase de 0 pour un "0" et saut de phase de  $\pi$  pour un "1"



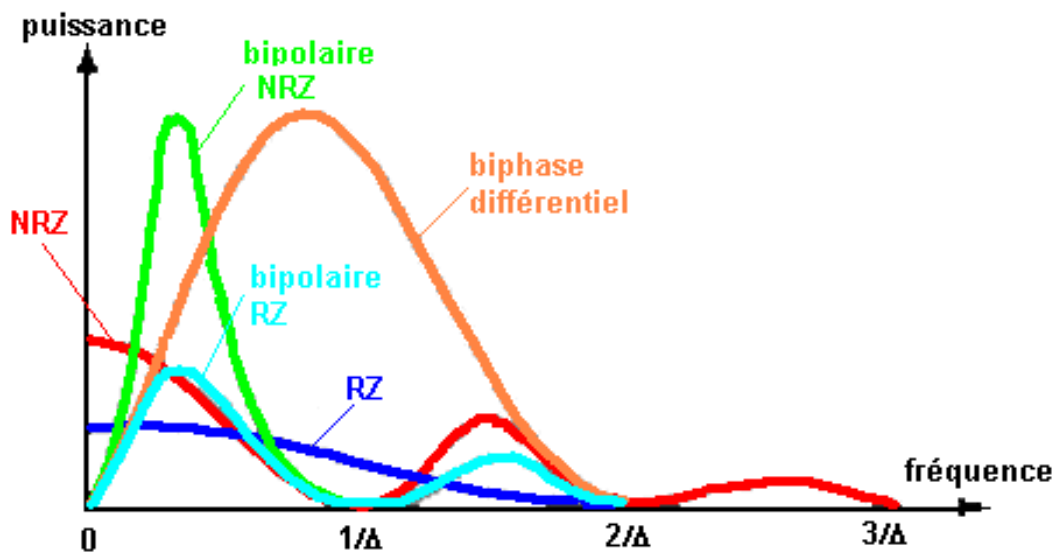
Pour ces codages, il est important de vérifier que les fréquences transportées se trouvent dans la bande passante car ils ne doivent pas subir un trop fort affaiblissement. Pour un codage donné d'une valeur binaire (un octet par exemple), le signal est décomposé en composantes sinusoïdales de Fourier et le spectre des fréquences est établi :



transformée de Fourier  
→



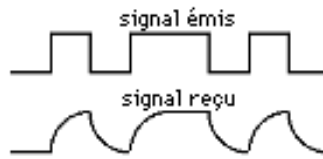
En effectuant cette opération pour toutes les valeurs possibles et en les combinant, on obtient le spectre du code. Quelques allures de ces spectres sont données ci-dessous.



Suivant les voies de transmission utilisées, il est alors possible de voir si le codage convient ou pas. En particulier, les codes NRZ et RZ possèdent l'inconvénient de posséder une harmonique non négligeable à la fréquence zéro

(composante qui passe mal au travers des équipements réseaux).

Par ailleurs et d'une manière générale, les signaux numériques possèdent un très gros inconvénient : ils se déforment à grande distance (effet capacitif des lignes) :



ce qui signifie que le transport par des signaux numériques n'est possible qu'à courte distance. Pour des longues distances, il faut employer une autre méthode : la modulation .

**Exercices et tests :** [Exercice 13](#), [Exercice 20](#), [Exercice 32](#), [Exercice 38](#), [Exercice 39](#), [QCM17](#)

## Modulation et démodulation

La modulation consiste à utiliser une onde "porteuse" sinusoïdale :

$$v(t) = V \sin(\omega t + \Phi)$$

dans laquelle on va modifier certains paramètres pour représenter les "0" et les "1" :

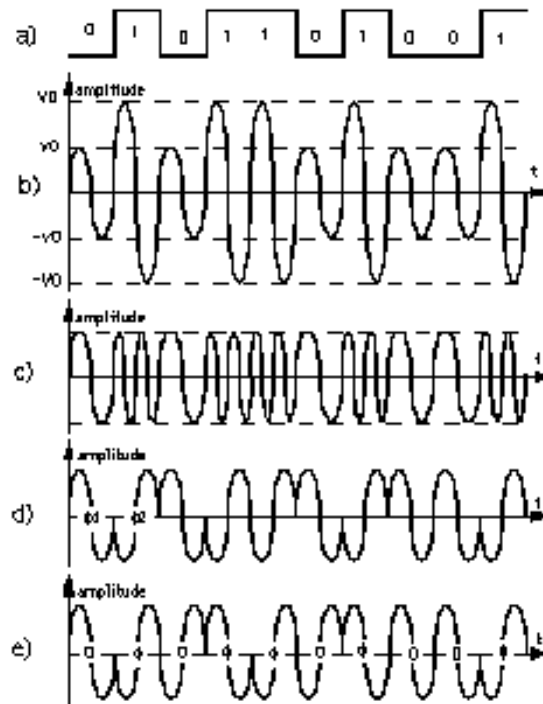
- modification de  $V$  (modulation d'amplitude)
- modification de  $\omega$  (modulation de fréquence)
- modification de  $\Phi$  (modulation de phase)

a) signal numérique à transporter en NRZ

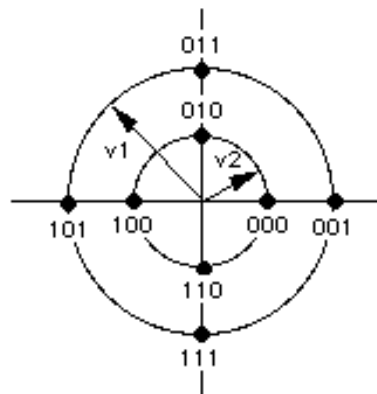
b) modulation d'amplitude

c) modulation de fréquence

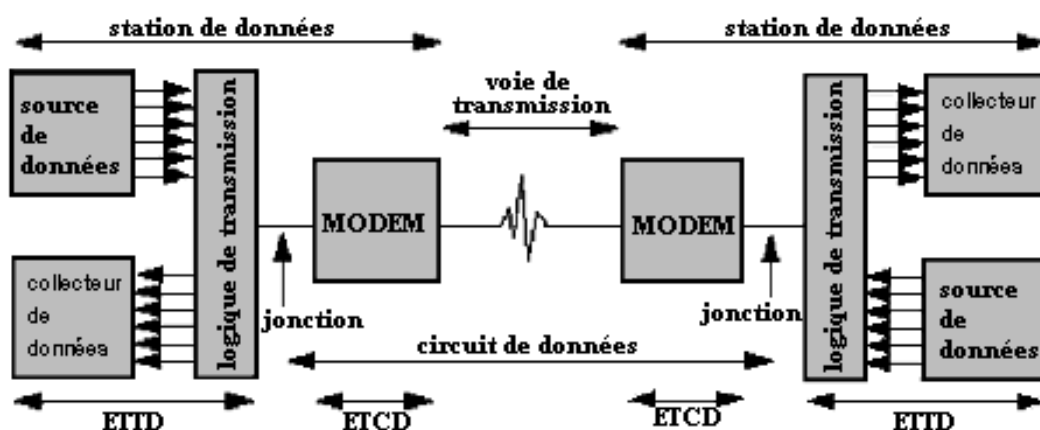
d) et e) modulation de phase



On peut aussi imaginer une combinaison des différents types de modulation, par exemple, la combinaison d'une modulation d'amplitude et d'une modulation de phase (dans la figure ci-dessous, cette combinaison permet d'avoir 8 signaux différents, chaque signal transportant chacun 3 bits) :



Pour les longues distances, la solution de la modulation est quasi-générale. Une liaison télé-informatique classique (en modulation) est représentée ci-dessous :



ETTD : Equipement Terminal de Traitement de Données  
ETCD : Equipement Terminal de Circuit de Données

**Exercices et tests :** [QCM18](#), [QCM19](#), [QCM20](#)



# Commutation et Multiplexage

Sommaire :

[Principe de la commutation](#)

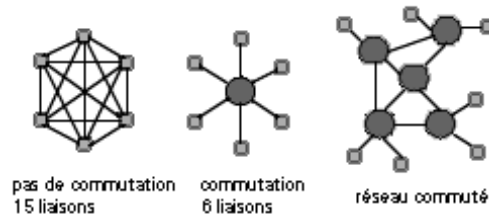
[Types de commutation](#)

[Multiplexage](#)

[Voies Numériques Multiplexées](#)

## Principes de la commutation

Pour la communication entre usagers, la commutation est essentielle. Il est en effet impensable de relier chaque usager à tous les autres. En effet, si l'on voulait relier  $n$  stations directement à chacune d'elles, il faudrait établir  $n(n-1)/2$  liaisons ce qui est impensable au niveau planétaire.

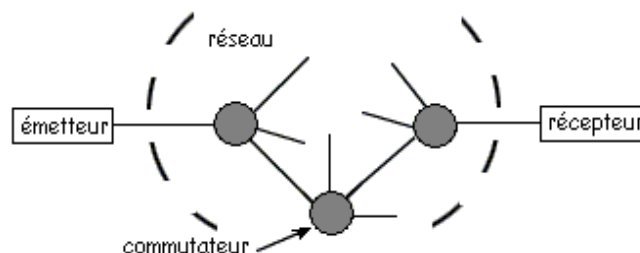


On est conduit logiquement à construire les réseaux à partir de **nœuds de commutation**. Ces nœuds de commutation sont chargés d'acheminer dans la bonne direction les informations qu'ils reçoivent. Cette fonctionnalité est appelée **routage**.

## Types de commutation

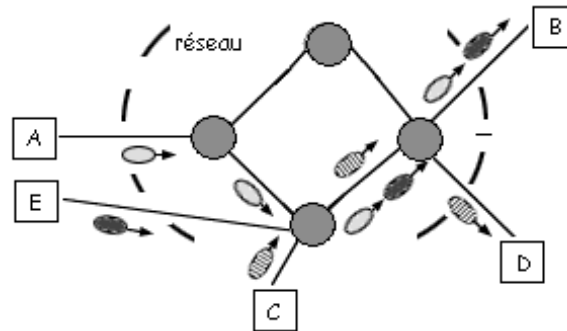
En fait, la commutation peut se concevoir de manières différentes

- **commutation de circuits** : elle consiste à réquisitionner, pour une communication, des tronçons de réseau pour assurer une liaison de bout en bout ; les tronçons sont liés les uns aux autres à chaque nœud de commutation ; la communication terminée, les tronçons sont libérés et disponibles pour une nouvelle commutation. Cette méthode est bien connue en téléphonie.



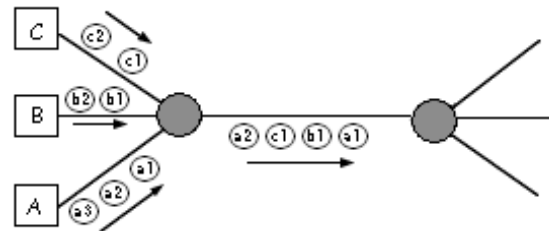
[illustration](#)

- **commutation de messages** : l'information à transmettre est découpée en messages ; les messages circulent sur le réseau à manière du transport automobile. Chaque nœud de commutation sert de routeur mais aussi d'hébergement des messages en situation d'engorgement des tronçons du réseau. Ce mode de commutation a pratiquement disparu au profit de la commutation de paquets.



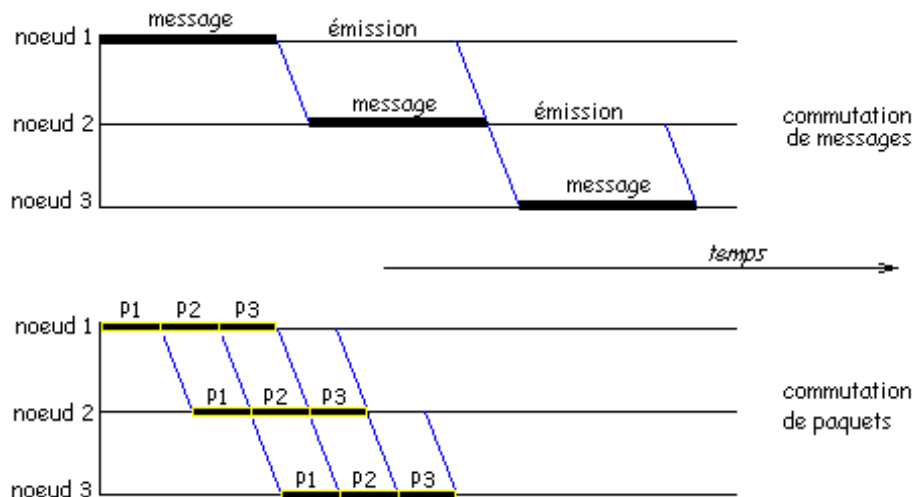
illustration

- **commutation de paquets** : chaque message est découpé en paquets de petite taille qui sont numérotés pour un ré-assemblage éventuel. Les paquets circulent dans le réseau et les nœuds de commutation en effectuent le routage et l'hébergement. Sur un tronçon, les paquets se suivent, même s'ils n'appartiennent pas au même message.



illustration

L'intérêt de la commutation de paquets sur la commutation de messages peut être rendu évident par la figure ci-dessous ; on gagne du temps par la simultanéité de réception et de transfert de paquets différents.



Il existe deux types de commutation de paquets

- le **circuit virtuel** : tous les paquets d'un même message suivent le même chemin défini pour chaque message ; la méthode est similaire à celle de la commutation de circuits.
- le **datagramme** : chaque paquet d'un message peut emprunter un chemin différent des autres ; à l'arrivée, il faut réordonner les paquets du message car des paquets peuvent aller plus vite que d'autres puisqu'empruntant des chemins différents.

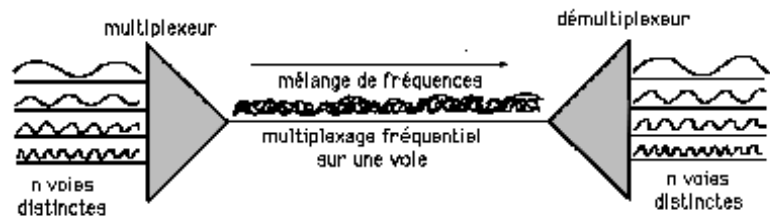
Exercices et tests : [QCM38](#), [QCM39](#)

# Multiplexage

Le multiplexage consiste à faire passer plusieurs messages sur un même tronçon de réseau. On distingue deux types de multiplexage :

- **multiplexage spatial**

La bande passante du canal est divisée en sous-bandes (canaux) chaque message correspond à une sous-bande de fréquence ; un multiplexeur mélange les différents messages ; un démultiplexeur, à l'arrivée, sépare, grâce à un filtrage en fréquence, les messages.



- **multiplexage temporel** : ce type de multiplexage est bien adapté aux réseaux à commutation de paquets. Le multiplexeur n'est autre qu'un mélangeur de paquets, le démultiplexeur est un trieur de paquets.



**exemple** : liaison à trame MIC offerte par France Télécom ; 1 trame (analogue à un train) comporte 30 IT utilisateurs et 2 IT de service (chaque IT, qui signifie "intervalle de temps", est analogue à un wagon). Chaque IT peut recevoir l'équivalent d'un paquet.

Chaque IT peut recevoir un octet ; une trame transporte donc 32 octets (256 bits ). Le débit total est de 2 Mbits/s. Si un usager utilise cette trame en mettant un paquet dans une IT précise dans chaque trame, le débit, pour cet usager, sera de 64 Kbits/s. S'il utilise deux IT par trame, il double son débit.

On constatera qu'une trame est transmise toutes les 125 microsecondes.

Exercices et tests : [Exercice 16](#), [Exercice 17](#), [Exercice 19](#), [Exercice 36](#), [QCM22](#), [QCM23](#)

# Voies numériques multiplexées

Les infrastructures de transport de l'information sont de nos jours dédiées au transport de données numériques. L'exemple du paragraphe précédent en est un exemple. Par ailleurs, destinées à transporter de volumineuses quantités de données binaires, elles utilisent la technique du multiplexage.

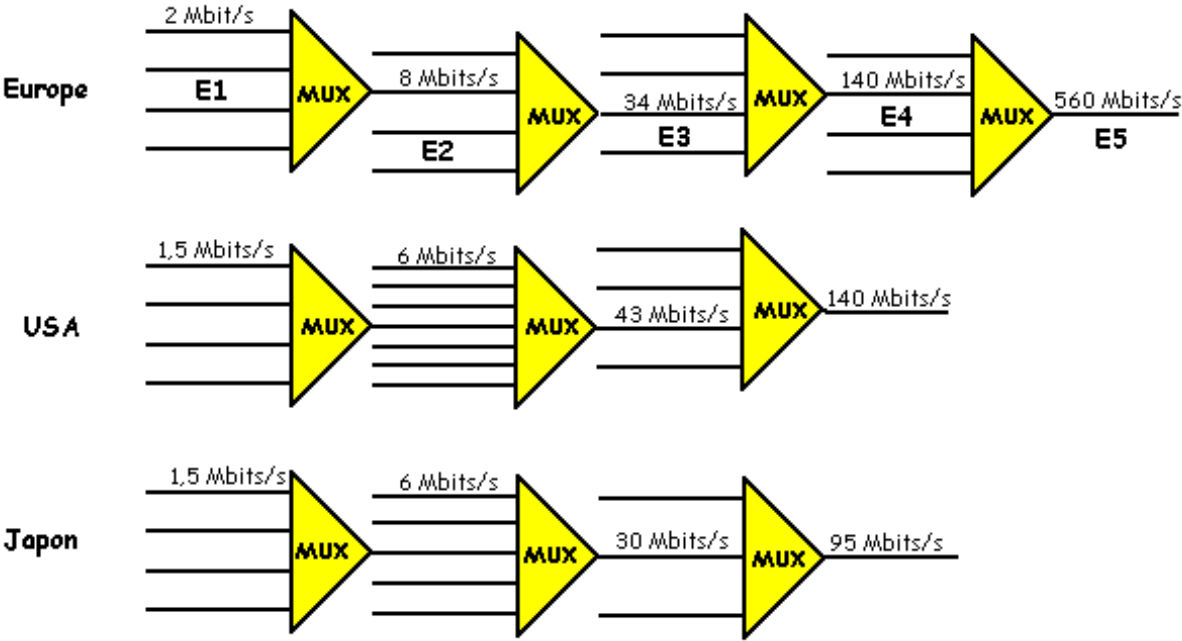
## PDH : Plesiochronous Digital Hierachy

Comme précisé précédemment, l'utilisation de trames MIC réalisent le multiplexage de 32 voies (IT) à 64 Kbits/s (l'utilisation de toute la trame correspond à 2 Mbits/s). Il faut noter que deux IT sont réservés pour le service (IT0 et IT16).

- IT0 : sert à délimiter les trames (mot de verrouillage de trame) : trame paire : 10011001 ; trame impaire : 11000000
- IT16 : informations de signalisation

Conçues à l'origine pour transporter la voie numérisée, ces trames sont multiplexées pour un transport d'un grand nombre de communications téléphoniques. Il faut aussi noter une différence de standardisation entre l'Europe (32 voies par trame) et les USA-Japon (24 voies par trame).

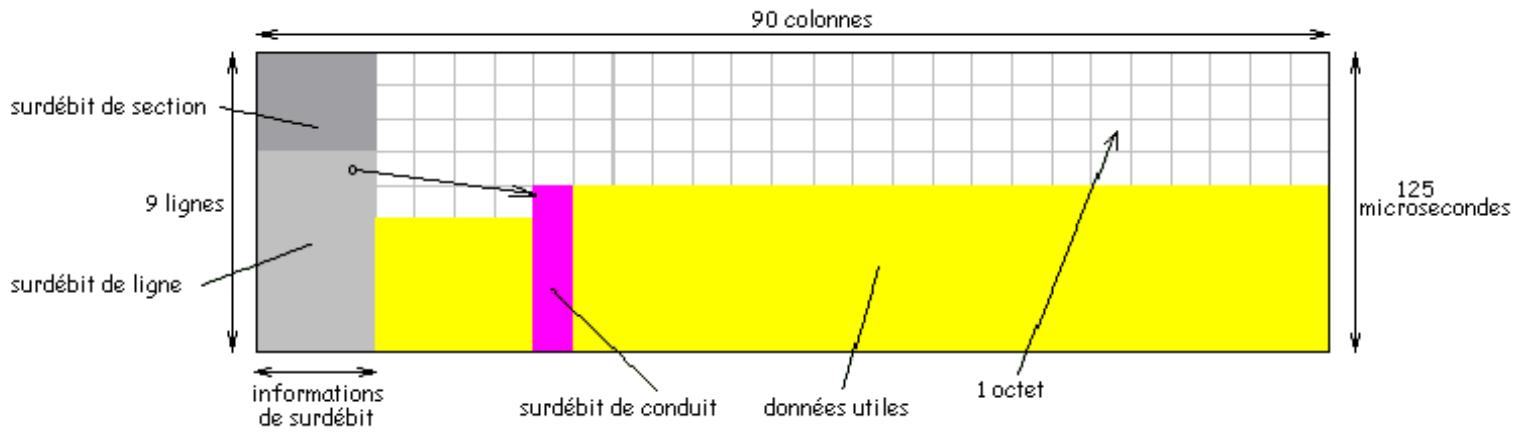
Le multiplexage successif des trames permet d'obtenir de hauts débits. De l'information de contrôle étant entrée à chaque niveau de multiplexage, le débit n'est pas exactement le débit nominal. C'est d'ailleurs pour cette raison que cette hiérarchie est appelée plésiochrone (plésio = presque).



Evidemment, comme cette technologie n'est pas vraiment synchrone, il est nécessaire de démultiplexer complètement pour accéder à une voie. c'est un inconvénient majeur qui a conduit à définir une autre hiérarchie, la hiérarchie synchrone.

## SDH : Synchronous Digital Hierachy

La hiérarchie SDH a été développée en Europe tandis qu'une hiérarchie analogue était développée aux USA : SONET (Synchronous Optical NETwork). Dans ce type de hiérarchie, la trame est plus complexe que dans le cas de PDH. Elle se reproduit 8000 fois par seconde et transporte 810 octets ce qui correspond à un débit de 51,84 Mbits/s ; cela signifie aussi qu'un octet particulier de la trame est transporté à un débit de 64 Kbits/s. La trame est présentée sous forme d'une grille de 9 lignes et 90 colonnes :



Les octets des trois premières lignes et des trois premières colonnes (surdébit de section), ainsi que le reste des trois premières colonnes (surdébit de ligne) sont utilisés pour la synchronisation. Un pointeur indique le début des données (conteneur virtuel) ; les données utiles commencent par un octet de surdébit de conduit. On peut insérer des données n'importe où dans la trame (dans les 87 colonnes suivant les trois premières).

La trame SDH est compatible avec la trame SONET, mais comporte 9 lignes de 270 colonnes (2430 octets). Elle est transmise en 125 microsecondes ce qui correspond à un débit de 155,52 Mbits/s, soit 3 fois le débit nominal de la trame SONET.

Les correspondances entre les niveaux de multiplexage de SDH et de SONET sont données dans le tableau suivant :

SDH	SONET	débit en Mbits/s
	STS1	51,84
STM1	STS3	155,52
	STS9	466,56
STM4	STS12	622,08
	STS18	933,12
	STS24	1244,16
	STS36	1866,24
STM16	STS48	2488,32

# Notion de protocole

Sommaire :

[Modélisation et protocoles](#)

[Exemples](#)

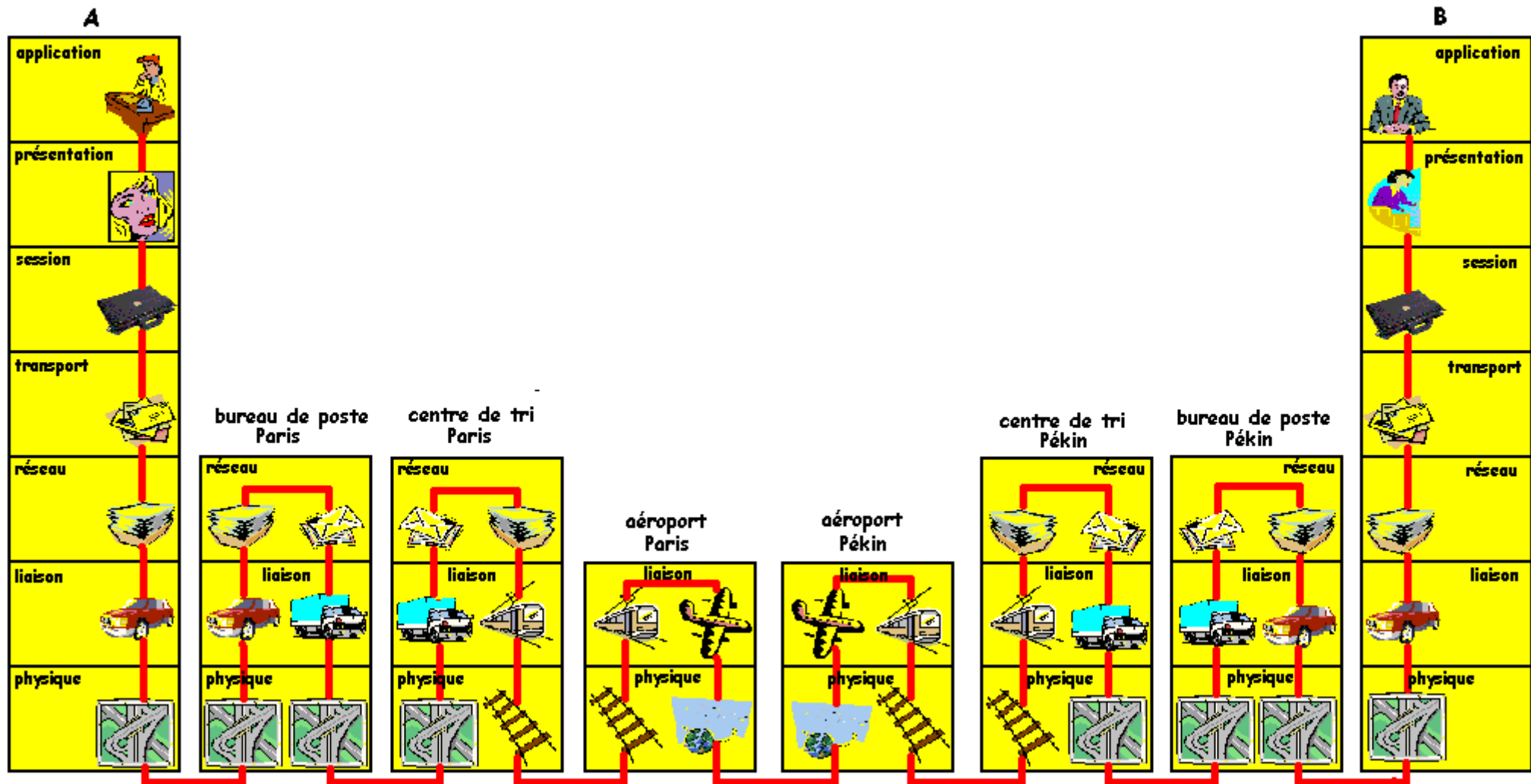
## Modélisation et protocoles

Un réseau de transmission de données est souvent exprimé sous la forme d'un modèle en **couches**. Pour faire comprendre ce concept, imaginons une modélisation de la poste internationale. Deux correspondants A, à Paris, et B, à Pékin s'envoient du courrier postal. Comme A ne parle pas le chinois et que B ne parle pas le français, la langue anglaise, supposée compréhensible par un nombre suffisant de personnes, sera choisie pour correspondre. Admettons aussi que ces deux correspondants envoient leur courrier à partir de leur lieu de travail (entreprise par exemple) : leur courrier partira donc en même temps que le courrier de leur entreprise qui est géré par un service courrier.

Imaginons alors la succession d'événements pour que A envoie une lettre à B.

- A écrit la lettre en français avec son stylo.
- A donne sa lettre à une secrétaire anglophone qui la traduit en anglais, la met dans une enveloppe et écrit l'adresse de B
- La personne chargée du ramassage du courrier passe dans le service de A pour ramasser le courrier.
- Le service courrier effectue un tri du courrier et l'affranchit avec une machine à affranchir.
- Le courrier est déposé au bureau de poste.
- Le courrier est chargé dans une voiture qui l'emmène au centre de tri
- Le courrier pour la Chine est emmené à l'aéroport de Paris par train
- Le courrier pour la Chine est transmis par avion à l'aéroport de Pékin
- Le courrier est transmis par train de l'aéroport de Pékin au centre de tri de Pékin
- Le courrier pour l'entreprise de B est transmis à l'entreprise par voiture
- Le service courrier de l'entreprise de B trie le courrier arrivé par service
- Le courrier est distribué à heure fixe aux destinataires et en particulier au service de B
- La secrétaire de B ouvre le courrier et traduit en chinois le contenu de la lettre destinée à B
- B lit la lettre que lui a envoyée A.

On peut résumer par un schéma la succession des événements afin de mettre en évidence un modèle en couches et les noeuds du réseau:



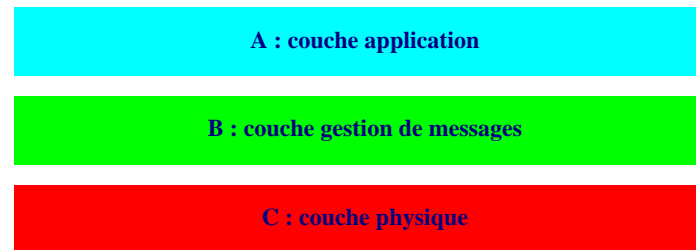
La dénomination des couches est conforme à un standard appelé OSI (Open System Interconnect) qui sera étudié plus loin. Sur cet exemple, à but uniquement pédagogique, basé sur un réseau postal (imaginaire !), explicitons les fonctionnalités de chaque couche.

- couche application : écriture/lecture de la lettre
- couche présentation : traduction, mise en forme, ouverture de lettre
- couche session : relevé/distribution du courrier dans les services
- couche transport : action du service courrier
- couche réseau : action du bureau de poste ou du centre de tri
- couche liaison : acheminement de la lettre entre deux noeuds consécutifs du réseau
- couche physique : utilisation des supports de communication

Dans cette modélisation, chaque couche est bâtie sur la couche inférieure. Par exemple, le transport routier (couche liaison) a besoin de l'infrastructure routière (couche physique).

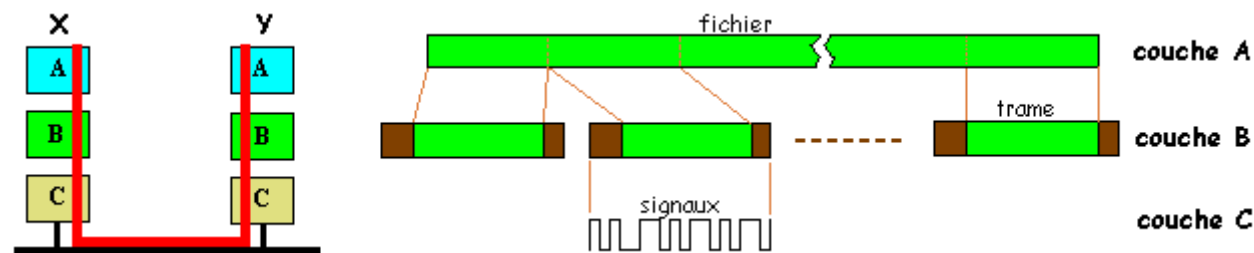
Pour chacune des couches, des fonctionnalités (ici très résumées) sont définies qui sont des services rendus aux couches supérieures. Les lignes rouges du schéma indiquent la suite de services rendus par les différentes couches. Par ailleurs, les fonctionnalités de chaque couche correspondent à des règles appelées **protocoles**.

Prenons maintenant un exemple plus "télécommunications" en envisageant un transfert de fichier entre un ordinateur X et un ordinateur Y reliés par un câble série. On peut envisager une modélisation à 3 couches :



- L'utilisateur désirant transférer un fichier fait appel à la couche A à l'aide d'une primitive du type `envoyer_fichier` (nom du fichier, destinataire).
- La couche A découpe le fichier en messages et transmet chaque message à la couche B par une primitive du type `envoyer_message` (numéro de message, destinataire).
- La couche B effectue la gestion de l'envoi de message, éventuellement en découpant le message en unités intermédiaires (trames) ; l'envoi des trames entre X et Y obéissent à des règles (protocole) : cadence d'envoi, contrôle de flux, attente d'un accusé de réception, contrôle de erreurs.
- La couche B fournit à la couche C un train de bits qui sera acheminé, indépendamment de sa signification, via une voie de transmission physique, vers le destinataire.

L'information est transmise par une voie de communication plus ou moins complexe et chemine, au niveau du destinataire dans le sens inverse de ce qui vient d'être décrit: émetteur et récepteur possède des couches identiques.



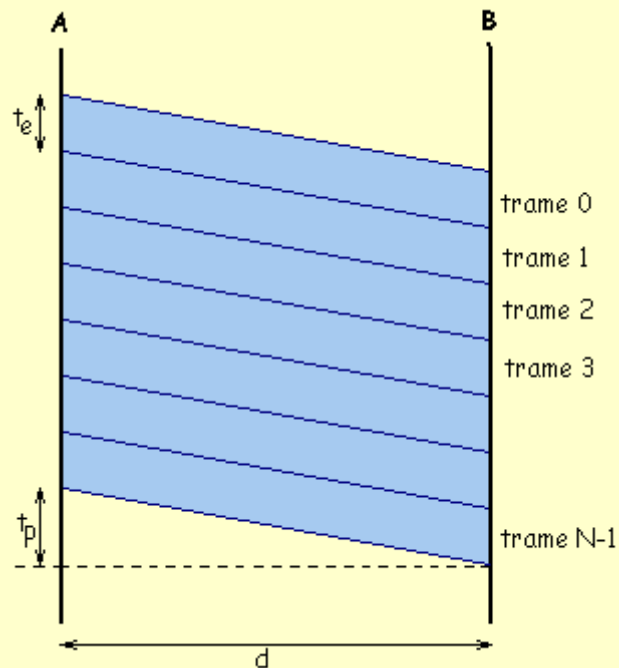
On notera aussi que les unités d'information diffèrent suivant les trois couches. Pour la couche A, l'unité est un fichier, c'est à dire une suite importante de bits. Pour la couche B, l'unité d'information est la trame qui possède une structure définie (information utile + information de service). Pour la couche C, l'unité d'information est le signal transmis sur le support physique de communication.



## Exemples

Prenons comme étude de cas l'envoi de trames sur une liaison entre 2 noeuds A et B consécutifs d'un réseau. On admettra que l'information est envoyée sous forme de blocs successifs appelés trames. On suppose que ces trames ont une longueur fixe  $L$ , que les noeuds sont distants de  $d$ , que la vitesse de propagation des signaux sur le support de communication est  $v$ , que le débit est  $D$ , que chaque signal transporte 1 bit.

**exemple 1 :** la voie de communication est parfaite et il ne peut y avoir d'erreur de transmission ; on suppose que la transmission est unidirectionnelle de A vers B ; les noeuds ont des capacités de traitement et de mémoire infinies : ils peuvent envoyer ou recevoir à tout moment. Les trames sont envoyées les unes après les autres. La chronologie des événements est indiquée ci-dessous :



Une trame est émise (et est reçue) en un temps  $t_e = L/D$ . Le temps nécessaire à l'envoi de  $N$  trames est donc  $Nt_e$ .

Mais la dernière trame étant émise, il faut laisser le temps aux signaux de se propager jusqu'à B, d'où le temps de propagation d'un bit (ou d'un signal) :  $t_p = d/v$

En définitive, le temps total de transmission de  $N$  trames est

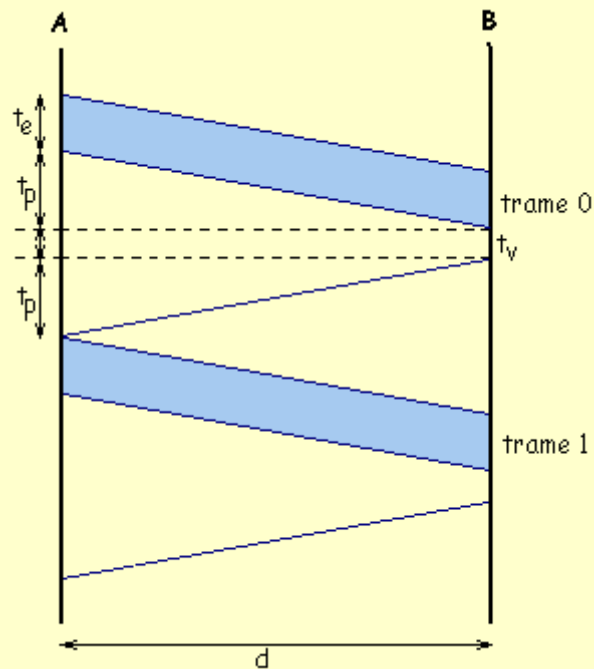
$$T = Nt_e + t_p$$

Le protocole est ici réduit à sa plus simple expression : définition de la longueur d'une trame et envoi successif des trames.

**exemple 2 :** On reprend les hypothèses de l'exemple précédent avec les modifications suivantes :

- on suppose maintenant qu'il peut y avoir des erreurs de transmission et que ces erreurs peuvent être détectées par le destinataire. Le mécanisme de détection suppose qu'un champ erreur soit incorporé à la trame.
- un acquittement est envoyé de B vers A sous forme d'un message de 1 bit (0 si la trame est correcte, 1 si la trame est erronée). A n'envoie de trame que si l'acquittement de la trame précédente a été reçu.
- si un acquittement négatif revient vers A, celui-ci doit ré-envoyer de nouveau la trame.
- les trames comporte un champ dont la valeur est le numéro de trame.
- le temps de traitement (vérification de la trame) est supposé constant et égal à  $t_v$

Le schéma chronologique est maintenant le suivant (dans l'hypothèse où il n'y a pas d'erreur).



Le temps nécessaire à l'acheminement complet d'une trame est  $t_e + 2 t_p + t_v$  où  $t_e$  et  $t_p$  ont les mêmes définitions que dans l'exemple 1. Le temps nécessaire à l'acheminement de N trames est donc

$$T = N(t_e + 2 t_p + t_v)$$

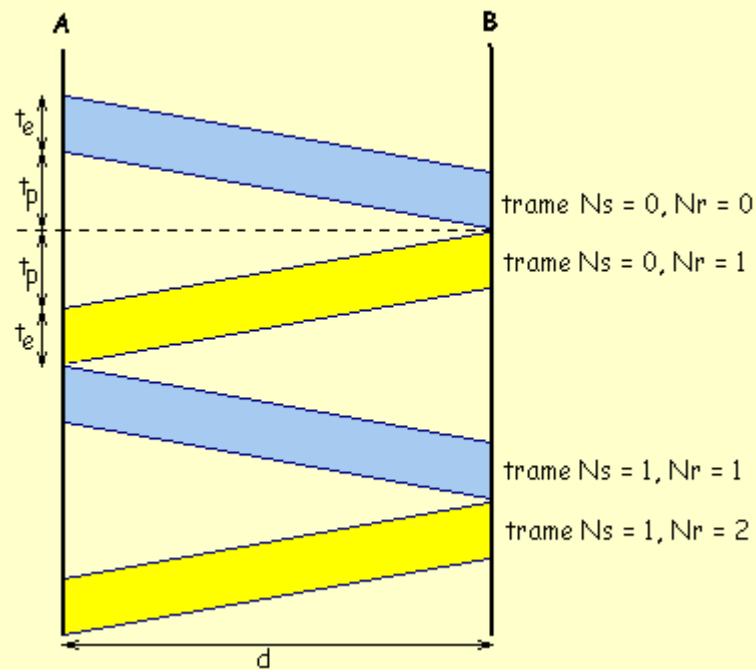
On notera que puisque l'acquittement ne comporte qu'un seul bit, le temps d'émission de cet acquittement est négligeable.

Questions : Le protocole ci-dessus possède un inconvénient majeur ; lequel ? Quelle doit être la longueur du champ relatif à la numérotation des trames ?

**exemple 3 :** On modifie maintenant les hypothèses de la manière suivante

- la transmission est bi-directionnelle ; chacun son tour X et Y envoient des trames
- chaque trame comporte, outre le champ détecteur d'erreur, un champ comportant 2 numéros : le numéro de trame  $N_s$  et le numéro de la prochaine trame attendue  $N_r$  de la part du correspondant. Si X reçoit une trame avec  $N_r = 5$ , il doit émettre la trame numéro 5 et il est sûr que la trame 4 a été reçue sans erreur (et situation analogue pour Y).
- on néglige le temps de vérification des erreurs de transmission.

Le chronologie des événements est indiquée ci-dessous dans le cas où il n'y a pas d'erreur de transmission.



Le temps nécessaire d'envoi d'une trame est  $t_e + 2 t_p$ , mais on doit attendre un temps  $t_e$  (le temps de recevoir une trame) avant d'envoyer la prochaine trame ; le temps nécessaire à l'envoi de  $N$  trames est donc

$$T = 2N(t_e + t_p)$$

Questions : Quel est l'intérêt de la double numérotation des trames ? Ce protocole possède-t-il un inconvénient ?

**Application numérique :** Examinons les performances de ces trois protocoles sous les hypothèses suivantes :

- valeurs des paramètres de base : d= 1000 m ; L = 1024 bits ; D = 64 Kbits/s et 155 Mbits/s ; v = 3.10<sup>8</sup> m/s ;
- pour l'exemple 2, les champs numérotation et erreurs ont une longueur totale de 11 octets , le temps de vérification des erreurs est de 10<sup>-5</sup> secondes; pour l'exemple 3, ces champs ont une longueur totale de 12 octets.

On s'intéressera aux critères de performance suivants : temps nécessaire à l'acheminement d'un message de longueur 1 Mo et temps d'occupation en émission par X de la voie de communication.

le tableau ci-dessous donne les résultats des calculs :

exemples	temps d'acheminement		taux d'occupation	
	D = 64 Kbits/s	D = 155 Mbits/s	D = 64 Kbits/s	D = 155 Mbits/s
exemple 1	125 s	0,05 s	1	1
exemple 2	133 s	0,28 s	0,99	0,28
exemple 3	269 s	0,17 s	0,50	0,33

Le cas de l'exemple 1 est sans intérêt car non réaliste (liaison parfaite) ; dans l'exemple 2, on a de bonnes performances pour un débit de 64 Kbits/s, par contre pour le débit de 155 Mbits/s, le taux d'occupation devient assez mauvais (ne pas oublier que les liaisons ne sont pas gratuites !) ; pour l'exemple 3, le taux d'occupation n'est pas extraordinaire, mais il faut prendre en considération que la ligne est bidirectionnelle et, en fait, le taux devrait être multiplié par 2.

**Exercices et tests :** [Exercice 33](#), [Exercice 34](#), [Exercice 37](#), [Exercice 41](#), [QCM31](#), [QCM32](#), [QCM33](#), [QCM34](#), [QCM35](#), [QCM36](#), [QCM37](#)

# Bibliographie

<b>D. BATTU</b>	<b>Télécommunications, Principes, Infrastructures et services</b>	<b>Dunod Informatiques</b>
<b>P. LECOY</b>	<b>Technologie des Télécoms</b>	<b>Hermes</b>
<b>C. SERVIN</b>	<b>Telecoms 1, de la transmission à l'architecture de réseaux</b>	<b>Dunod Informatiques</b>
<b>W. STALLINGS</b>	<b>Data and Computer Communications</b>	<b>Prentice Hall</b>
<b>G. BOUYER</b>	<b>Transmissions et réseaux de données</b>	<b>Dunod</b>
<b>M. MAIMAN</b>	<b>Télécoms et Réseaux</b>	<b>Masson</b>
<b>P. ROLLIN, G. MARTINEAU, L. TOUTAIN, A. LEROY</b>	<b>Les Réseaux, principes fondamentaux</b>	<b>Hermes</b>
<b>A. TANENBAUM</b>	<b>Réseaux</b>	<b>InterEditions</b>
<b>P-G. FONTOLLIET</b>	<b>Systèmes de télécommunications, bases de transmission</b>	<b>Dunod</b>

# Exercices et Tests

sommaire :

[Enoncés](#)

[Solutions](#)

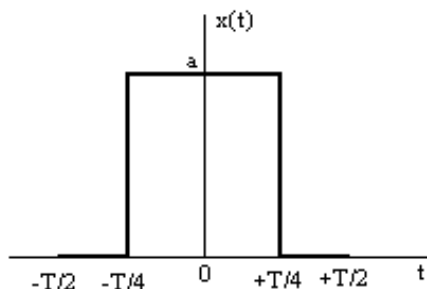
[QCM](#)

## Exercice 1

- 1) Une image TV numérisée doit être transmise à partir d'une source qui utilise une matrice d'affichage de 450x500 pixels, chacun des pixels pouvant prendre 32 valeurs d'intensité différentes. On suppose que 30 images sont envoyées par seconde. Quel est le débit  $D$  de la source ?
- 2) L'image TV est transmise sur une voie de largeur de bande 4,5 MHz et un rapport signal/bruit de 35 dB. Déterminer la capacité de la voie.

## Exercice 2

Un signal numérique de forme "créneau", de période  $T$ , est envoyé sur une voie de transmission.



- 1) Décomposer le signal en série de Fourier
- 2) La voie ayant une bande passante allant de la fréquence  $4/T$  à  $8/T$ , quel est le signal reçu en bout de ligne (en négligeant le bruit, l'amortissement et le déphasage).

## Exercice 3

Quelle est la capacité d'une ligne pour téléimprimeur de largeur de bande 300 Hz et de rapport signal/bruit de 3 dB ?

---

#### Exercice 4

Un système de transmission numérique fonctionne à un débit de 9600 bits/s.

- 1) Si un signal élémentaire permet le codage d'un mot de 4 bits, quelle est la largeur de bande minimale nécessaire de la voie ?
  - 2) Même question pour le codage d'un mot de 8 bits.
- 

#### Exercice 5

Une voie possède une capacité de 20 Mbits/s. La largeur de bande de la voie est de 3 MHz. Quel doit être le rapport signal/bruit ?

---

#### Exercice 6

Si l'affaiblissement est de 30 dB, quel est le rapport  $|V_e/V_s|$  des ondes sinusoïdales d'entrée et de sortie d'une portion de voie de transmission ?

---

#### Exercice 7

La décomposition en série de Fourier d'un signal périodique conduit à une superposition de signaux sinusoïdaux de fréquences  $f$ ,  $3f$ ,  $5f$ ,  $7f$ ,... Sachant que la bande passante est  $[5f, 25f]$ , combien de signaux sinusoïdaux élémentaires seront détectés à l'arrivée ?

---

#### Exercice 8

Une voie de transmission véhicule 8 signaux distincts ; sa rapidité de modulation est  $R = 1200$  bauds. Quel est le débit binaire de

cette ligne ?

---

### Exercice 9

Une voie de transmission véhicule 16 signaux distincts. Quelle est la quantité d'information binaire maximale pouvant être transportée par chaque signal ?

---

### Exercice 10

Le rapport signal sur bruit d'une voie de transmission est de 30 dB ; sa largeur de bande est de 2 MHz. Quelle est, approximativement, la capacité théorique de cette voie ?

---

### Exercice 11

Sur une voie de transmission, on constate que le nombre de communications par heure est de 1,5 et que chaque communication a une durée moyenne de 360 secondes. Quel est le trafic correspondant ?

---

### Exercice 12

Sachant que pour une voie de transmission, le nombre de transactions par communication est de 4000, la longueur moyenne d'une transaction est de 12000 bits, la durée moyenne d'une communication est 3600 secondes, le débit binaire est 64 Kbits/s, donner le taux d'occupation de la voie.

---

### Exercice 13

On envoie la suite de bits : 01001110.

Quels sont les signaux correspondants en NRZ, RZ, bipolaire NRZ, bipolaire RZ, biphase cohérent, biphase différentiel ?



---

### Exercice 14

On considère un signal audio dont les composantes spectrales se situent dans la bande allant de 300 à 3000 Hz. On suppose une fréquence d'échantillonnage de 7 KHz.

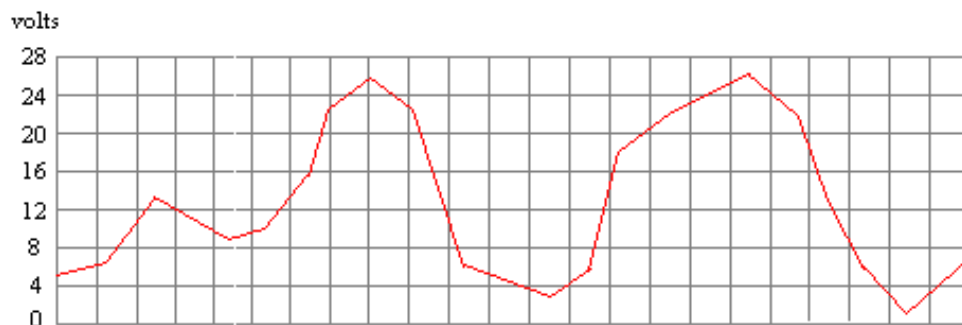
1) Pour un rapport signal sur bruit S/B de 30 dB, quel est le nombre  $n$  de niveaux de quantification nécessité ? On donne la relation :  $S/B = 6n - a$ . On prendra  $a = 0,1$ .

2) Quel est le débit nécessité ?

---

### Exercice 15

Soit le signal audio suivant :



Le codage étant effectué sur 8 niveaux et l'échantillonnage étant défini sur la figure ci-dessus, en déduire le codage binaire de ce signal.

---

### Exercice 16

4 trains d'information analogique sont multiplexés sur une ligne téléphonique de bande passante 400 - 3100 Hz. La bande passante de chaque train est de 500 Hz. Expliciter le processus de multiplexage.

---

## Exercice 17

3 lignes sont multiplexées sur une liaison à commutation de paquets de longueur 1200 bits. Chaque ligne transporte des messages de longueur respective : 3600 bits, 12000 bits, 4800 bits. Le débit de la liaison commutée est de 4800 bits/s. Décrire le processus de multiplexage.

---

## Exercice 18

Des caractères ASCII sur 8 bits sont envoyés sur une voie de transmission de débit nominal D.

- 1) On effectue la transmission en mode asynchrone avec un bit start et un bit stop. Exprimer en fonction de D le débit utile.
  - 2) On effectue la transmission en mode synchrone avec des trames comportant un drapeau de début et un drapeau de fin, chacun de 8 bits, un champ de contrôle de 48 bits et un champ d'information de 128 bits. Exprimer en fonction de D le débit utile.
  - 3) Même question que b) mais avec un champ d'information de longueur 1024 bits.
- 

## Exercice 19

Trois voies à 1200 bits/s sont multiplexées sur une voie à 2400 bits/s. Ces trois voies véhiculent des paquets de même longueur. Pour un paquet, quel est le débit apparent sur la voie multiplexée ?

---

## Exercice 20

Dans la liste suivante apparaissent des codages en bande de base ; lesquels ?

RZ ISO6 TCP HTTP NRZ RVB

---

## Exercice 21

Pour numériser un son mono analogique, on utilise une fréquence d'échantillonnage de 22 KHz et on code le un codage de valeurs sur 8 bits. Pour 1 minute de son, quel est le volume correspondant en bits (on suppose qu'il n'y a pas de compression) ?

---

**Exercice 22**

On divise le polynôme  $x^7 + x^5 + 1$  par le polynôme générateur  $x^3 + 1$ . Quel est le reste obtenu ?

---

**Exercice 23**

On considère des mots de 3 bits et un codage linéaire de matrice G. Déterminer les mots codés.

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

---

**Exercice 24**

Un code cyclique utilise la matrice H définie ci-dessous :

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Cette matrice H est l'équivalent de la matrice G et est définie par la relation  $H.Y = 0$  où Y est le vecteur "codé" comportant les bits utiles et les bits de contrôle ; la matrice H est toujours de la forme (h, 1) et possède r lignes (r étant le nombre de bits de contrôle). Quel est l'algorithme de codage ?

---

**Exercice 25**

Un code utilise le polynôme générateur  $x^2 + x + 1$ . Quel est l'encodage du message 11011 ?

## Exercice 26

On considère le code ci-dessous

<u>mots</u>	<u>mots code</u>
00	10011
01	10100
10	01001
11	01110

Ce code permet-il

- 1) de détecter toutes les erreurs doubles ?
- 2) de corriger toutes les erreurs simples ?

## Exercice 27

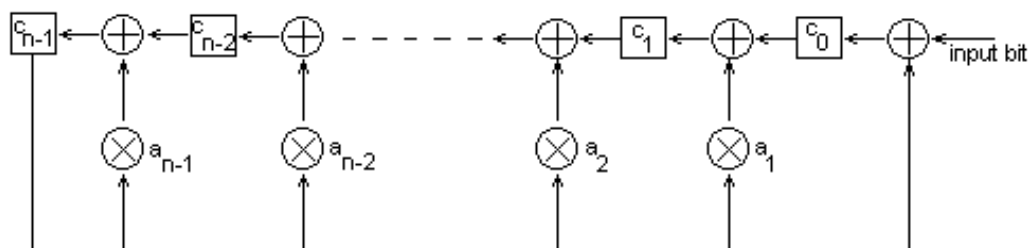
Un message de longueur 11 bits est encodé avec 4 bits de contrôle par un code polynômial basé sur l'utilisation du polynôme générateur

$$H(z) = z^4 + z^3 + 1.$$

- 1) Déterminer l'algorithme de calcul des bits de contrôle.
- 2) Soit le mot utile suivant :  $M = 10011011100$  ; encoder ce mot.

## Exercice 28

Dans le cas d'un codage polynômial, on peut automatiser le calcul des bits de contrôle avec un circuit intégré basé sur un registre à décalage et des portes XOR. L'architecture d'un tel circuit est décrite par le schéma ci-dessous.



pour un polynôme générateur du type  $H(z) = 1 + a_1z + a_2z^2 + \dots + a_{n-1}z^{n-1} + z^n$ . Les bits à encoder sont envoyés un par un à l'entrée du registre à décalage, suivis de  $n$  zéros. Ce qui reste dans le registre à décalage après cette opération est le champ de contrôle.

- 1) Imaginer la structure du circuit d'encodage pour le cas de l'exercice 6.

2) Appliquer le circuit au mot utile :  $M = 10011011100$  et en déduire le champ de contrôle.

---

## Exercice 29

Quelle est la distance de Hamming entre  $m1 = (11010101)$  et  $m2 = (10110101)$  ?

---

## Exercice 30

Soit un code linéaire (6,3) dont la matrice est

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Quelle est l'information codée correspondant à l'information utile 101 ?

---

## Exercice 31

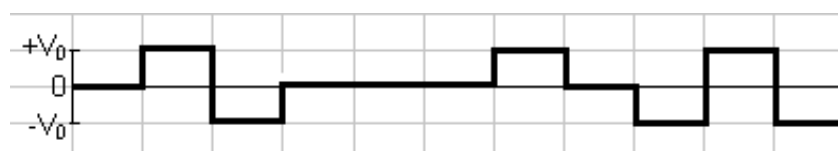
Dans le cas de l'exercice 30, quelle est la matrice G ?

---

## Exercice 32

Dans les trames normalisées E1, on utilise le code Bipolar AMI qui consiste à coder un 0 par une absence de tension électrique et un 1 par une tension alternativement positive et négative.

1) Quelle est la suite binaire codée de la figure ci-dessous ?



2) Sachant qu'une trame E1 correspond à un débit de 2 Mbits/s, quelle est la durée d'un moment élémentaire (durée d'un signal numérique) ?

---

### Exercice 33

On utilise dans la transmission de trames d'un émetteur A vers un récepteur B un protocole défini de la manière suivante.

- a) l'émetteur envoie successivement trois trames puis attend leur acquittement de la part de B.
- b) quand cet acquittement arrive, l'émetteur envoie les trois trames suivantes et attend un nouvel acquittement.
- c) les trames sont composées de 1024 bits dont 80 bits de service
- d) les acquittements sont composés de 64 bits
- e) le débit de la voie est de 2 Mbits/s et la vitesse de propagation des ondes électromagnétiques est de  $3.10^8$  m/s sur la voie de 10 km reliant A et B.

1) Quelle est la durée nécessaire à l'expédition confirmée d'une trame ?

2) Quel est le taux d'occupation de la voie ?

3) Un message de 1 Mo est envoyé de A vers B par utilisation du protocole précédent. Quelle est la durée totale de la transmission de ce message ?

---

### Exercice 34

Deux machines A et B sont reliées par un réseau utilisant le protocole de liaison HDLC. La machine A reçoit de la machine B une trame correcte portant les numéros  $N(R)=5$ ,  $N(S)=4$ . La machine A, à son tour, envoie à la machine B une trame comportant les numéros  $N(S)$  et  $N(R)$ . Quelles sont les valeurs de  $N(S)$  et  $N(R)$  ?

---

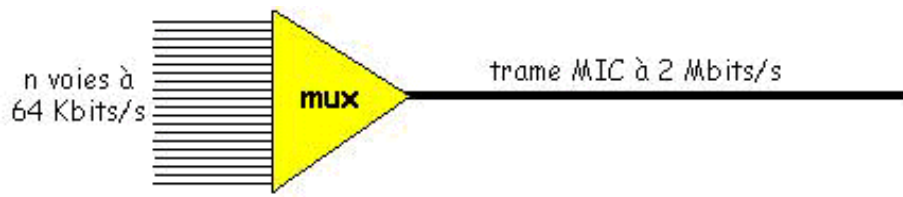
### Exercice 35

On désire transporter du son numérique sur une voie de transmission. La largeur de bande de la voix humaine est supposée bornée supérieurement à 4000 Hz. En appliquant le théorème de l'échantillonnage, le son est numérisé à 8000 Hz et codé sur 8 bits.

Quel doit être le débit de la ligne utilisée ?

## Exercice 36

La trame MIC permet de multiplexer plusieurs voies à 64 Kbits/s.



a) Sachant que la trame MIC correspond à un débit de 2 Mbits/s, combien de voies peuvent-elles ainsi être multiplexées dans une trame MIC ?

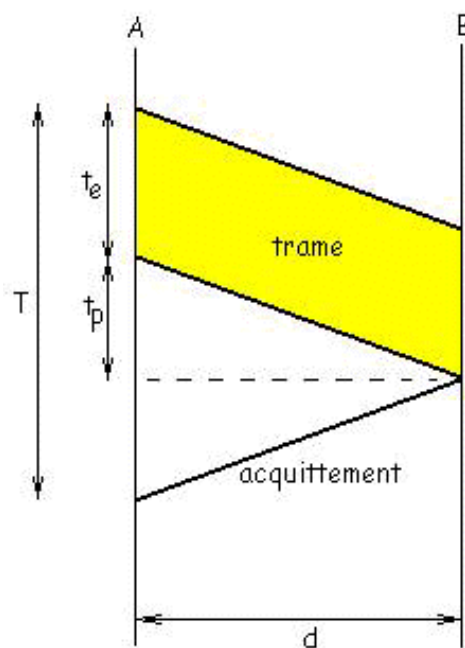
b) Une application particulière, comme la visioconférence, nécessite un débit de 192 Kbits/s. Indiquer comment, avec une trame MIC, il est possible d'atteindre ce débit.

## Exercice 37

On imagine un protocole de transmission obéissant aux règles suivantes :

- le débit est  $D$
- à la suite de l'envoi d'une trame par la station A, un acquittement est renvoyé à A par la station B destinataire de la trame. On considérera que cet acquittement peut être réduit à 1 bit.
- la longueur  $L$  de la trame est fixe

On désigne par  $d$  la distance entre les stations A et B et par  $v$  la vitesse de propagation d'un signal (correspondant ici à un bit) dans la voie reliant A et B.



a) Exprimer le temps total de transmission d'une trame  $T$  (depuis l'émission du premier bit jusqu'à la réception de l'acquittement) en fonction de  $L$ ,  $D$ ,  $d$ ,  $v$ .

b) En déduire en fonction du rapport  $a = t_p/t_e$  le taux d'occupation  $\theta$  de la voie (rapport du temps d'émission  $t_e$  d'une trame sur le temps total de transmission  $T$ ) ;  $t_p$  désigne le temps de propagation d'un bit entre A et B.

c) Application numérique : Calculer  $\theta$  pour  $L=1024$  bits ;  $D = 64$  Kbits/s ;  $d = 1000$  m ;  $v = 2.10^8$  m/s

d) Application numérique : Calculer  $\theta$  pour  $L = 53$  octets ;  $D = 155$  Mbits/s ;  $d = 1000$  m ;  $v = 2.10^8$  m/s (situation présentant des analogies avec l'ATM).

e) A partir des résultats des deux applications numériques précédentes, quelles conclusions pouvez-vous en tirer ?

## Exercice 38

On envisage plusieurs types de codage pour transmettre des données binaires par des signaux numériques. Les principaux codes sont définis par le tableau ci-dessous :

code	définition			
NRZL	0 : niveau haut ; 1 : niveau bas			
NRZI	0 : pas de transition ; 1 : transition			
Bipolar AMI	0 : pas de signal ; 1 : alternativement niveau positif ou négatif			
Pseudoternaire	0 : alternativement niveau positif ou négatif ; 1 : pas de signal			
Manchester	0 : transition haut-bas au milieu de l'intervalle ; 1 : transition bas-haut au milieu de l'intervalle			
Differential Manchester	toujours une transition au milieu de l'intervalle ; 0 : transition au début de l'intervalle ; 1 pas de transition au début de l'intervalle			
B8ZS	Comme Bipolar AMI mais toute suite de 8 zéros est remplacée par une suite comme indiqué : voltage précédent négatif : 00000000 devient 000-+0+- voltage précédent positif : 00000000 devient 000+-0-+			
HDB3	Comme Bipolar AMI mais toute suite de 4 zéros est remplacée par une suite comme suit			
	polarité du niveau précédent	nombre de 1 depuis la dernière substitution		
		impair	pair	
		négatif	000-	+00-
		positif	000+	-00-

1) Représenter la suite binaire 01001100011 dans les codes NRZL, NRZI, Bipolar AMI, Pseudoternaire, Manchester, Differential Manchester.









binaire	0	1	0	0	1	1	0	0	0	1	1	
NRZL												
NRZI												
Bipolar AMI												
Pseudoternaire												
Manchester												
Differential Manchester												

2) Représenter la suite 1100000000110000010 par les codes Bipolar AMI, B8ZS, HDB3 :

binaire	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0
Bipolar AMI																			
B8ZS																			
HDB3																			

Exercice 39

Les réseaux locaux rapides utilisent des codages spécifiques. C'est le cas du codage 4B/5B utilisé dans 100BaseX et FDDi sur fibre optique : Chaque suite de 4 bits est codée sur 5 bits suivant le schéma suivant :

mot de 4 bits	codage	signal
0000	11110	
0001	01001	
0010	10100	
0011	10101	
0100	01010	
0101	01011	
0110	01110	
0111	01111	
1000	10010	
1001	10011	
1010	10110	
1011	10111	
1100	11010	
1101	11011	
1110	11100	
1111	11101	
idle	11111	
start 1	11000	
start 2	10001	
end 1	01101	
end 2	00111	
error	00100	

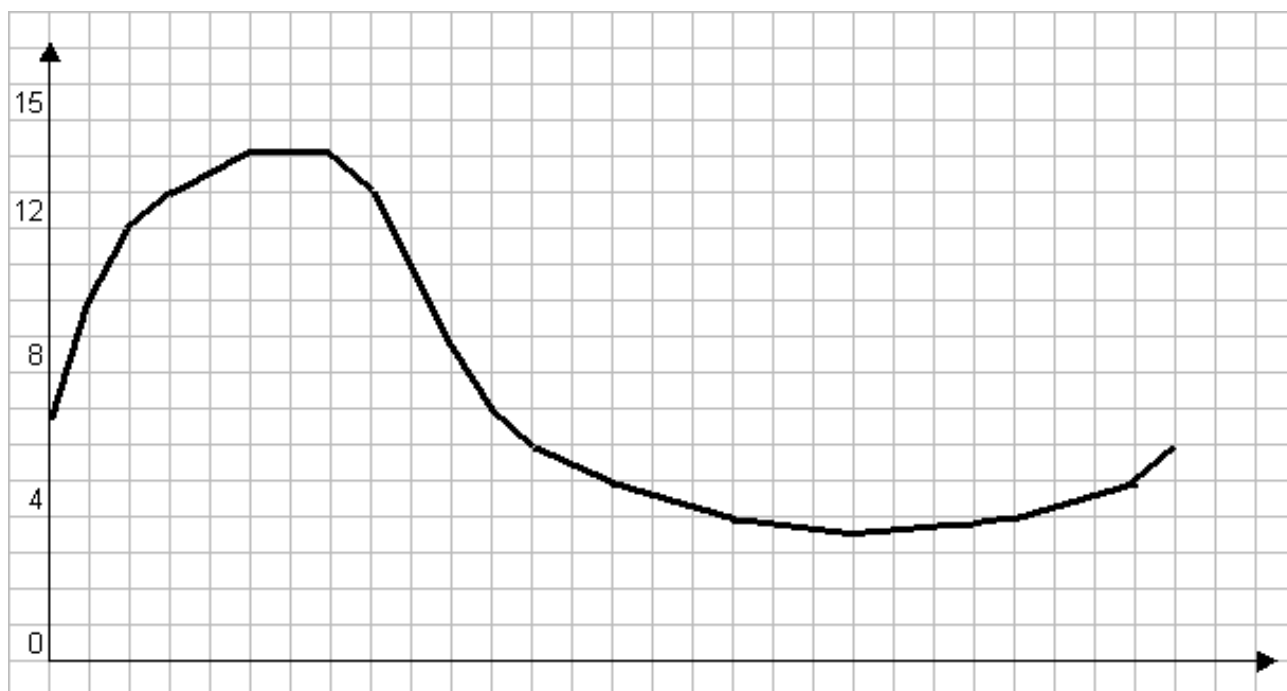
Déterminer quel est le codage en signaux utilisé et compléter le tableau ci-dessus.

Exercice 40

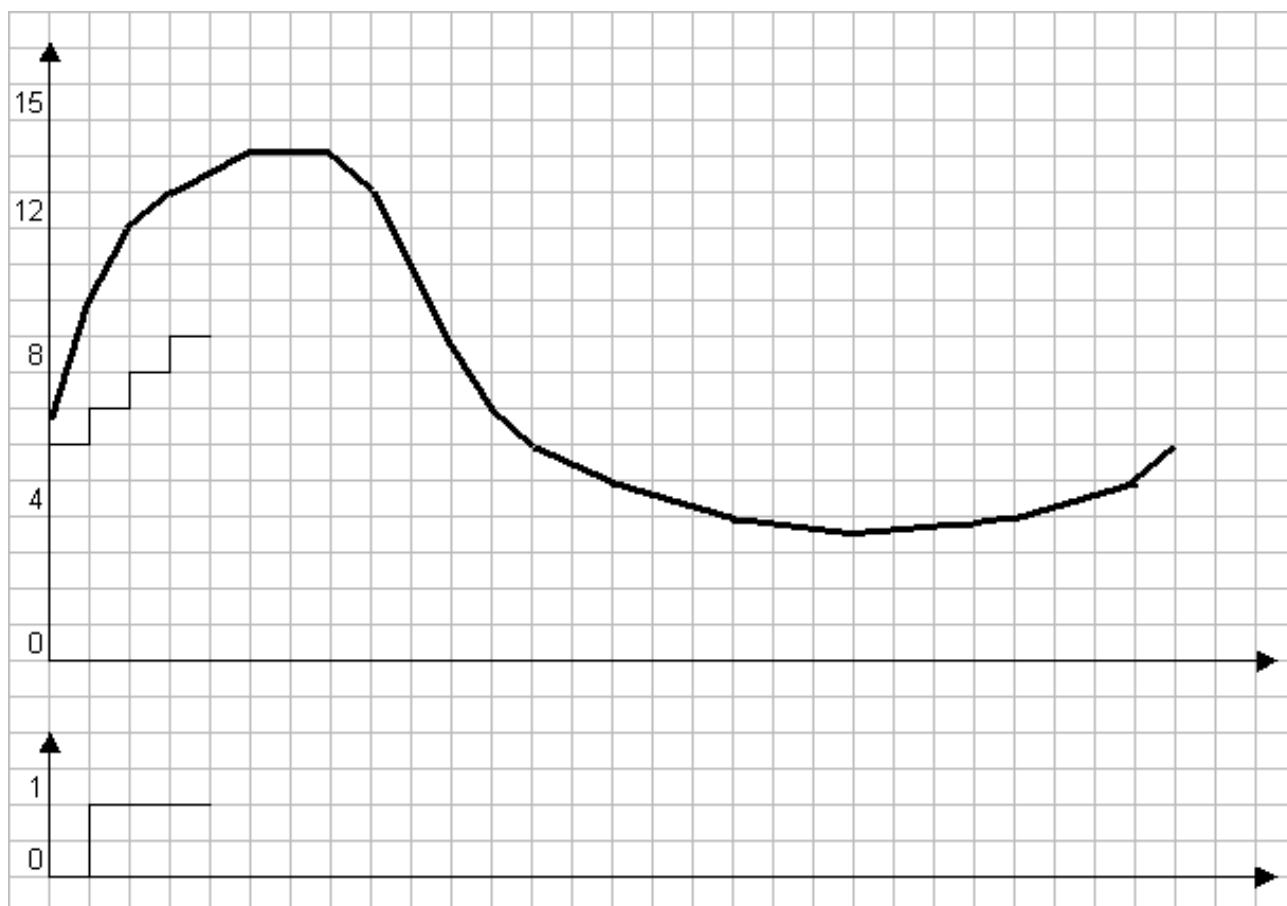
1) Dans le cadre de l'échantillonnage de données analogiques, on peut utiliser le codage ordinaire PCM (Pulse Code Modulation) qui

consiste à coder sur  $n$  bits chaque valeur mesurée de la donnée (avec approximation de quantification : on va au plus près par exemple).

Soit la donnée analogique suivante que l'on désire coder sur 4 bits (les lignes verticales indiquent les instants d'échantillonnage). En déduire le fichier binaire correspondant.



2) On peut aussi utiliser la méthode de codage appelée Modulation Delta. Cette méthode consiste à monter d'un pas de quantification à chaque échantillonnage, vers le haut si on est au-dessous de la courbe analogique, vers le bas si on est au-dessus de la courbe analogique. Le codage résultant est binaire : transition si on change de sens, pas de transition si le sens ne change pas. Le schéma ci-dessous indique le début de codage. Compléter le codage et donner le fichier binaire résultant.



## Exercice 41

1) On considère une ligne half-duplex entre deux stations  $S_1$  et  $S_2$  fonctionnant suivant le mode Stop and Wait :

$S_1$  envoie une trame  $f_1$  et attend ;  $S_2$  à la réception de  $f_1$  envoie un acquittement ;  $S_1$  reçoit l'acquittement  
 $S_1$  envoie une trame  $f_2$  et attend ;  $S_2$  à la réception de  $f_2$  envoie un acquittement ;  $S_1$  reçoit l'acquittement

-----  
 $S_1$  envoie une trame  $f_n$  et attend ;  $S_2$  à la réception de  $f_n$  envoie un acquittement ;  $S_1$  reçoit l'acquittement

a) Exprimer le temps total d'expédition d'une trame depuis l'envoi du premier bit jusqu'à la réception du dernier bit de l'acquittement. On utilisera les durées suivantes :

$t_{\text{prop}}$  : temps de propagation d'un bit entre  $S_1$  et  $S_2$   
 $t_{\text{frame}}$  : temps d'émission d'une trame  
 $t_{\text{proc}}$  : temps de traitement de données reçues  
 $t_{\text{ack}}$  : temps d'émission d'un acquittement

On considère que  $t_{\text{proc}}$  est négligeable devant les autres durées et que la taille d'un acquittement est négligeable devant la taille d'une trame de données. En déduire une approximation de la durée d'expédition de  $n$  trames.

On pose  $a = t_{\text{prop}} / t_{\text{frame}}$  Exprimer le taux d'occupation de la ligne  $\theta$  en fonction de  $a$ .

Si  $D$  est le débit binaire de la ligne,  $d$ , la distance entre les stations,  $v$  la vitesse de propagation des ondes sur la ligne,  $L$  la longueur d'une trame en bits, exprimer  $a$  en fonction des grandeurs précédentes.

b) On suppose que  $t_{\text{frame}} = 1\text{s}$ , d'où  $t_{\text{prop}} = a$ .

Suivant que  $a < 1$  ou  $a > 1$ , indiquer ce qui se passe aux instants  $t = 0, 1, a, 1+a, 1+2a$ .

c) On considère 3 types de liaisons :

c1) liaison véhiculant des cellules ATM (53 octets) ; débit 155,52 Mbits/s ; fibre optique. Calculer  $a$  pour une distance de 1000 km et  $\theta$  . Conclusion.

c2) liaison de réseau LAN ; trames de 1000 bits ; débit 10 Mbits/s ;  $v = 2.10^8$  m/s dans les conducteurs de cuivre. Calculer  $a$  pour une distance de 1 km, puis  $\theta$  . Conclusion.

c3) liaison téléphonique à 28,8 Kbits ; trames de 1000 bits ; Calculer  $a$  pour une distance de 1000 m et de 5000 km. Calculer  $\theta$  . Conclusion.

2) On envisage une méthode de fenêtre glissante. On considère que la largeur vde la fenêtre est  $N$  ( $N = 2^n - 1$  où  $n$  est le nombre de bits servant au codage du numéro de trame). Supposons que  $t_{\text{frame}} = 1$

Etudier ce qui se passe aux instants  $t = 0, a, a+1, 2a + 1$ . On envisagera les deux cas  $N > 2a+1$  et  $N < 2a + 1$

En déduire l'expression du taux d'occupation  $\theta$ .

Donner la représentation graphique de  $\theta$  en fonction de  $a$ , pour  $N=1, N=7, N=127$ .

3) On envisage un contrôle d'erreur.

a) On désigne par  $P$  la probabilité pour qu'une trame soit erronée et par  $r$  le nombre de fois où on transmet la même trame (sans erreurs  $r=1$ ). Montrer que  $r = 1/(1-P)$ .

b) Montrer que le taux d'occupation pour la méthode Stop and Wait est, dans le cas d'un contrôle d'erreur donné par  $\theta = (1-P)/(1+2a)$ .

c) On considère la méthode SR-ARQ (Selective Reject-Automatic Repeat Request) : dans une rafale de trame, seule la trame erronée est retransmise. Déterminer l'expression de  $\theta$ .

d) On considère la méthode GBN-ARQ (Go Back - Automatic Repeat Request) : dans une rafale de trames, on retransmet toutes les trames à partir de la trame erronée. Si  $K$  est le nombre de trames à retransmettre, donner l'expression de  $r$  en fonction de  $P$  et  $K$  ( $r$  est le nombre moyen de trames transmises pour transmettre avec succès une trame de la séquence). En considérant les deux cas  $N > 2a + 1$  et  $N < 2a + 1$ , quelle est la valeur de  $K$  ? En déduire l'expression de  $\theta$ .

---

### Solution de l'Exercice 1

1) Volume  $V = 33\,750\,000$  bits ; le débit  $D$  est  $D = 33,75$  Mbits/s.

2) Appliquons la relation  $C = 2W \log_2(1 + S/B)^{1/2}$ . Toutefois, il faut faire attention que dans cette relation  $S/B$  est exprimée en rapport de puissances et non en décibels. On écrira donc de préférence

$$C = 2W \log_2(1 + P_S/P_B)^{1/2}$$

$$P_S/P_B = \exp[(\ln(10)/10) \cdot S/B] = 3162 \text{ d'où } C = (9/2) \cdot (\ln(3163)/\ln(2)) = 52 \text{ Mbits/s.}$$

A noter que avec  $S/B = 30$  dB, on aurait  $C = 44,8$  Mbits/s et que avec  $S/B = 20$  dB, on aurait  $C = 29,96$  Mbits/s.

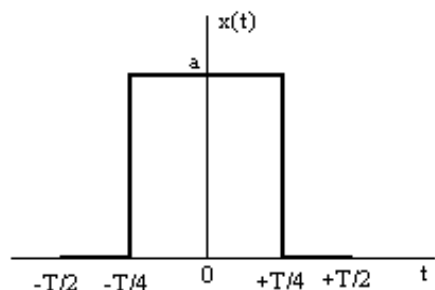
---

### Solution de l'Exercice 2

1) Le développement en série de Fourier est

$$x(f) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos n\omega_0 f + b_n \sin n\omega_0 f) \quad \text{avec } \omega_0 = \frac{2\pi}{T}$$

et comme le signal est pair, on n'a pas de termes en sinus.



$$x(f) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos n\omega_0 f)$$

Les coefficients sont

$$\frac{a_0}{2} = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} x(f) df = \frac{a}{2} \quad a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} x(f) \cos n\omega_0 f df = \frac{2a}{n\pi} \sin n \frac{\pi}{2}$$

Cette dernière relation peut encore s'écrire

$$a_{2k+1} = \frac{2a}{(2k+1)\pi} (-1)^k \quad a_{2k} = 0 \quad \text{d'où} \quad x(f) = \frac{a}{2} + \sum_{k=0}^{\infty} \frac{2a(-1)^k}{(2k+1)\pi} \cos(2k+1)\omega_0 f$$

2) Il ne reste que

$$x'(f) = \frac{2a}{5\pi} \cos 5\omega_0 f - \frac{2a}{7\pi} \cos 7\omega_0 f$$

On constatera qu'il ne reste que peu de choses à l'arrivée !

## Solution de l'Exercice 3

En reprenant les considérations de l'exercice 3, on obtient  $C = 475,5$  bits/s.

## Solution de l'Exercice 4

1) Puisqu'un signal transporte 4 bits, la rapidité de modulation est  $R = D/4 = 1200$  bauds.

La rapidité de modulation maximale est  $R_{\max} = kW$  avec  $k = 1,25$ . Donc  $R < 1,25 W$  et par suite

$W > R/1,25$  soit  $W_{\min} = 2400/1,25 = 1920$  Hz.

2) Dans ce cas un signal transporte 8 bits, donc  $W_{\min} = 1200/1,25 = 960$  Hz.

---

### Solution de l'Exercice 5

En reprenant les considérations de l'exercice 3, on a  $1 + P_S/P_B = \exp [C \cdot \ln(2)/W] = 101$ , d'où  $P_S/P_B = 100$ .

En décibels,  $S/B = 10 \log_{10}(P_S/P_B) = 20 \text{ dB}$ .

---

### Solution de l'Exercice 6

L'affaiblissement est donné par la relation  $A = 10 \log_{10}(P_e/P_s)$  où  $P_e$  et  $P_s$  désignent les puissances électriques d'entrée et de sortie ; on a  $P_e = V_e I = V_e^2/Z$  et de même  $P_s = V_s^2/Z$  d'où  $A = 20 \log_{10}(V_e/V_s)$ . On a donc dans les conditions de l'énoncé :  $V_e/V_s = 10^{3/2} = 31,62$

---

### Solution de l'Exercice 7

On voit que la superposition ne comprend que des signaux dont la fréquence est un multiple impair de  $f$  ; entre  $5f$  et  $25f$  (bornes comprises, il y a 11 valeurs, donc 11 signaux.

---

### Solution de l'Exercice 8

1 signal transporte 3 bits (8 combinaisons possibles) ; donc  $D = 3R = 3600 \text{ bits/s}$

---

### Solution de l'Exercice 9

Avec 4 bits on peut former 16 combinaisons différentes auxquelles correspondent les 16 signaux distincts. Donc la quantité d'information binaire transportée par signal est 4 bits.

---

### Solution de l'Exercice 10

$C=19,93.10^6$  bits/s.

On emploie la relation  $C = W \log_2(1+(S/B)_W)$  et la relation  $(S/B)_{dB} = 10 \log_{10}((S/B)_W)$  qui convertit le rapport des puissances en Watts S/B en son équivalent en décibels.

---

### Solution de l'Exercice 11

La relation à employer est la définition du trafic :  $E = N.T/3600 = 1,5 \times 360/3600 = 0,15$  Erlang

---

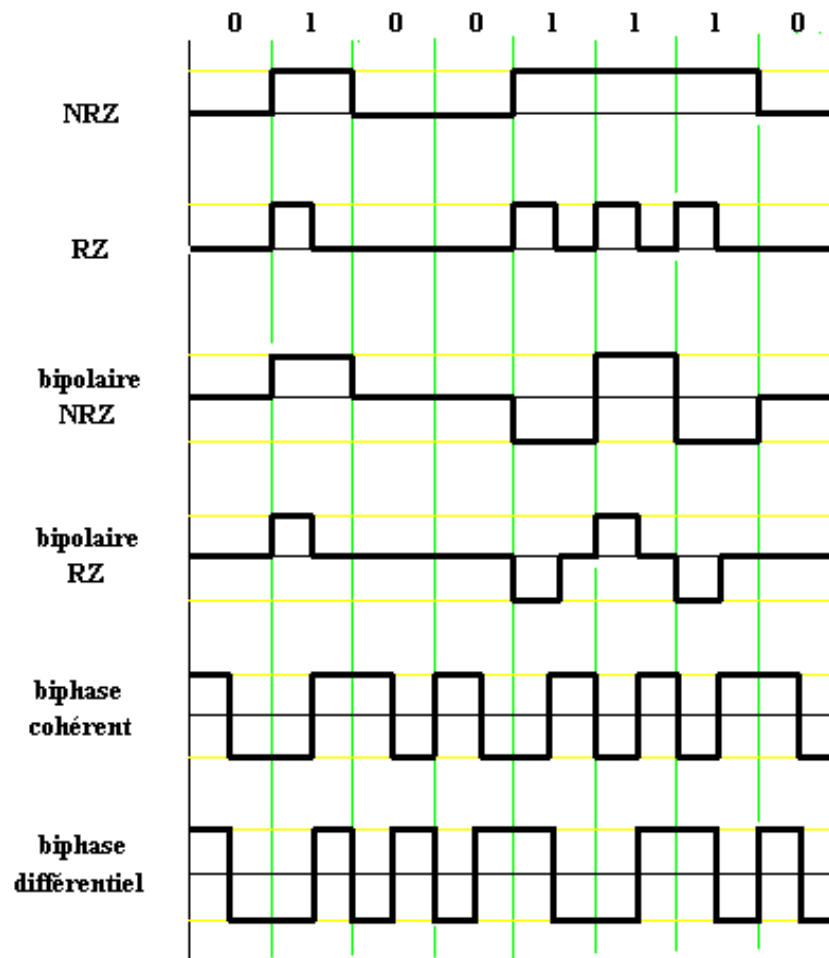
### Solution de l'Exercice 12

Le débit effectif est  $d = 4000 \times 12000/3600 = 13\,333,333$  et le taux d'occupation est le rapport  $\theta = d/D = 0,20$

---

### Solution de l'Exercice 13





## Solution de l'Exercice 14

1) Appliquons la formule de l'énoncé pour trouver le nombre de niveaux :

$n = (S/B + a)/6 = 5$  environ. La puissance de 2 la plus proche est 4. On prendra donc 4 niveaux, ce qui signifie un codage de chaque échantillon sur 2 bits.

2) A la fréquence de 7 KHz, on a 7000 échantillons par seconde, soit 14 000 bits par seconde qui est donc le débit nécessaire.

## Solution de l'Exercice 15

En redéfinissant l'échelle verticale par des graduations allant de 0 à 7 (8 niveaux), on obtient la "hauteur" de chacun des échantillons (en allant au plus près) :

1 2 3 3 2 2 3 6 6 6 3 1 1 1 4 5 6 6 6 5 2 1 1 2

soit après codage



1) Soit  $d$  la durée d'émission d'un bit. Alors  $D = 1/d$ . Un caractère correspond à 10 bits, soit une durée d'émission de  $10d$ .

Le débit utile est alors  $U = 8/10d = 0,8 D$  en supposant que les caractères sont envoyés les uns derrière les autres.

2) Une trame compte 192 bits dont 128 utiles. Le débit utile est donc  $U = 128/192d = 0,66 D$

3) Une trame compte 1088 bits dont 1024 utiles. Le débit utile est donc  $U = 1024/1088d = 0,94 D$

### Solution de l'Exercice 19

Le débit sera trois fois plus faible puisque un paquet sur trois appartient au même message.

### Solution de l'Exercice 20

Les bons sigles sont RZ et NRZ

### Solution de l'Exercice 21

1 minute = 60 secondes . Par seconde, on effectue 22 000 mesures codées chacune sur 8 bits.

On a donc un volume de  $60 \times 22\,000 \times 8 = 10\,560\,000$ .

### Solution de l'Exercice 22

$$\begin{array}{r}
 x^7 + \phantom{0x^6} + \phantom{0x^5} + \phantom{0x^4} + \phantom{0x^3} + \phantom{0x^2} + \phantom{0x} + 0 \\
 \phantom{0x^7} + x^5 + \phantom{0x^4} + \phantom{0x^3} + \phantom{0x^2} + \phantom{0x} + 0 \\
 \phantom{0x^7} + \phantom{0x^5} + x^4 + \phantom{0x^3} + \phantom{0x^2} + \phantom{0x} + 0 \\
 \phantom{0x^7} + \phantom{0x^5} + \phantom{0x^4} + x^2 + \phantom{0x} + 0 \\
 \phantom{0x^7} + \phantom{0x^5} + \phantom{0x^4} + \phantom{0x^2} + x + 0
 \end{array}
 \begin{array}{r}
 1 \\
 1 \\
 1 \\
 1 \\
 1
 \end{array}
 \left| \begin{array}{l}
 x^3 + 1 \\
 \hline
 x^4 + x^2 + x
 \end{array} \right.$$

On obtient donc :  $Q(x) = x^4 + x^2 + x$  et  $R(x) = x^2 + x + 1$ .

NB: ne pas perdre de vue qu'en addition modulo 2,  $1+1 = 1-1 = 0$ .

### Solution de l'Exercice 23

Il y a 3 bits utiles et 1 bit de contrôle, soit 4 bits pour un mot du code. La relation  $\tilde{Y} = \tilde{X} \cdot G$  permet de déterminer l'algorithme de calcul du bit de contrôle :

$$(\tilde{y}_1 \ \tilde{y}_2 \ \tilde{y}_3 \ \tilde{y}_4) = (\tilde{x}_1 \ \tilde{x}_2 \ \tilde{x}_3) \begin{vmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{vmatrix} = (\tilde{x}_1 \ \tilde{x}_2 \ \tilde{x}_3 \ \tilde{x}_1 + \tilde{x}_2 + \tilde{x}_3)$$

Donc on aura le codage suivant :

mot non codé	mot codé
000	0000
001	0011
010	0101
011	0110
100	1001
101	1010
110	1100
111	1111

### Solution de l'Exercice 24

H possède r=3 lignes et n=6 colonnes, donc il y a m=6-3=3 bits utiles.

En posant  $\tilde{X} = (\tilde{x}_1 \ \tilde{x}_2 \ \tilde{x}_3)$  et  $\tilde{Y} = (\tilde{y}_1 \ \tilde{y}_2 \ \tilde{y}_3 \ \tilde{y}_4 \ \tilde{y}_5 \ \tilde{y}_6) = (\tilde{x}_1 \ \tilde{x}_2 \ \tilde{x}_3 \ a_1 \ a_2 \ a_3)$  et en calculant  $\tilde{H} \tilde{Y}$  qui doit être égal à 0, on obtient :

$$\begin{vmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ a_1 \\ a_2 \\ a_3 \end{vmatrix} = \begin{vmatrix} \tilde{x}_1 + \tilde{x}_2 + \tilde{x}_3 + a_1 \\ \tilde{x}_2 + \tilde{x}_3 + a_2 \\ \tilde{x}_1 + \tilde{x}_2 + a_3 \end{vmatrix}$$

$$\text{donc} \quad \begin{aligned} a_1 &= \tilde{x}_1 + \tilde{x}_2 + \tilde{x}_3 \\ a_2 &= \tilde{x}_2 + \tilde{x}_3 \\ a_3 &= \tilde{x}_1 + \tilde{x}_2 \end{aligned}$$

Le tableau ci-dessous donne, avec cet algorithme, le code de tous les mots utiles :

mot non codé	mot codé
000	000000
001	001110
010	010111
011	011001
100	100101
101	101011
110	110110
111	111100

### Solution de l'Exercice 25

$H(z)=z^2+z+1$  : le degré de ce polynôme est 2, donc il y a 2 bits de contrôle. Par ailleurs le mot utile proposé comporte 5 bits, donc le code porte sur des mots utiles de  $m=5$  bits. On en déduit le nombre de bits des mots codés :  $n=7$ .

$$\begin{aligned} \text{On pose : } X(z) &= x_0 + x_1 z + x_2 z^2 + x_3 z^3 + x_4 z^4 \\ z^2 X(z) &= x_0 z^2 + x_1 z^3 + x_2 z^4 + x_3 z^5 + x_4 z^6 \end{aligned}$$

$$\begin{array}{r|l} x_4 z^6 + x_3 z^5 + x_2 z^4 + x_1 z^3 + x_0 z^2 & z^2 + z + 1 \\ 0 + (x_3+x_4)z^5 + (x_2+x_4)z^4 + x_1 z^3 + x_0 z^2 & x_4 z^4 + (x_3+x_4)z^3 + (x_2+x_3)z^2 \\ 0 + (x_2+x_3)z^4 + (x_1+x_3+x_4)z^3 + x_0 z^2 & + (x_1+x_2+x_4)z + x_0+x_1+x_3+x_4 \\ 0 + (x_1+x_2+x_4)z^3 + (x_0+x_2+x_3)z^2 & \\ 0 + (x_0+x_1+x_3+x_4)z^2 + (x_1+x_2+x_4)z & \\ 0 + (x_0+x_2+x_3)z + x_0+x_1+x_3+x_4 & \end{array}$$

On a donc :

$$Q(z) = x_4 z^4 + (x_3+x_4)z^3 + (x_2+x_3)z^2 + (x_1+x_2+x_4)z + x_0+x_1+x_3+x_4$$

$$R(z) = (x_0+x_2+x_3)z + x_0+x_1+x_3+x_4$$

$$Y(z) = z^2 X(z) + R(z) = x_4 z^6 + x_3 z^5 + x_2 z^4 + x_1 z^3 + x_0 z^2 + (x_0+x_2+x_3)z + x_0+x_1+x_3+x_4$$

$$\text{soit } a_1 = x_0+x_2+x_3 \quad a_0 = x_0+x_1+x_3+x_4$$

Ainsi le mot 11011 est codé 1101100.

## Solution de l'Exercice 26

Pour le code fourni, la distance minimale de Hamming est  $d_{\min}=3$ .

- détection de  $p=2$  erreurs : d'après la règle 1, on a  $d_{\min}>p$ , soit  $3>2$ , donc cette détection est possible.
- correction de  $q=1$  erreur : d'après la règle 2, on a  $d_{\min}>2q$ , soit  $3>2$ , donc cette correction est possible.

On peut le vérifier sur le code fourni : Dans la colonne des mots codés, les colonnes de bits 2 et 3 redonnent les bits utiles, la colonne de bits 1 est l'inverse de la colonne de bits 2 ; de même la colonne de bits 5 est l'inverse de la colonne de bits 3 ; enfin la règle de parité impaire est appliquée aux colonnes de bits 1,4,5. On a donc trois bits de contrôle ( $a_1$  pour la colonne 1,  $a_2$  pour la colonne 4,  $a_3$  pour la colonne 5) :

$$a_1 = x_1 + 1 \quad a_2 = a_1 + a_3 + 1 \quad a_3 = x_2 + 1$$

où  $x_1$  et  $x_2$  désigne les bits du mot non codé.

Ainsi Soit le mot 01 qui est codé en 10100. Supposons que dans la transmission se produise une erreur et que le mot reçu soit 00100. Avec les règles ci-dessus, il est clair que l'on peut détecter et corriger cette erreur simple.

De même si 2 erreurs se produisent, par exemple 10100 est transformé en 00000, on détecte facilement l'erreur double en utilisant  $a_1$  et  $a_3$ .

## Solution de l'Exercice 27

1) Le mot utile étant  $(x_{10}, x_9, x_8, x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0)$ , le mot à encoder sera de la forme  $(x_{10}, x_9, x_8, x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0, a_3, a_2, a_1, a_0)$ .

En utilisant la relation  $Y(z) = Q(z)H(z) + R(z)$ , on obtient :

$$a_3 = x_0 + x_1 + x_2 + x_4 + x_6 + x_7 + x_{10}$$

$$a_2 = x_2 + x_3 + x_4 + x_5 + x_7 + x_9 + x_{10}$$

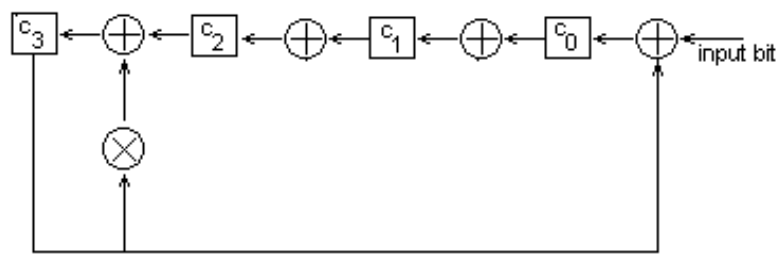
$$a_1 = x_1 + x_2 + x_3 + x_4 + x_6 + x_8 + x_9$$

$$a_0 = x_0 + x_1 + x_2 + x_3 + x_5 + x_7 + x_8$$

2) Le mot encodé sera 100110111001101

## Solution de l'Exercice 28

1) Le circuit est le suivant :



2) La suite de décalages est indiquée ci-dessous : le champ de contrôle est 1101.

c3	c2	c1	c0	entrée
0	0	0	0	
0	0	0	1	1
0	0	1	0	0
0	1	0	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	1	0
0	0	1	0	1
0	1	0	1	1
1	0	1	1	1
1	1	1	1	0
0	1	1	1	0
1	1	1	0	0
0	1	0	1	0
1	0	1	0	0
1	1	0	1	0

Solution de l'Exercice 29

La distance de Hamming est le nombre de bits de même rang qui diffèrent. Soit ici 2.

Solution de l'exercice 30

On emploie la relation  $HY=0$  :

$$H \bullet Y = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} x_2 + x_3 + a_1 \\ x_1 + x_3 + a_2 \\ x_1 + x_2 + a_3 \end{bmatrix}$$

d'où  $a_1 = x_2 + x_3$   $a_2 = x_1 + x_3$   $a_3 = x_1 + x_2$

On obtient donc 101101.

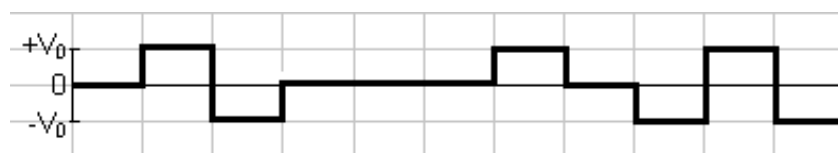
### Solution de l'Exercice 31

On emploie la relation

$$\tilde{Y} = \tilde{X} \bullet G \text{ soit } \begin{bmatrix} x_1 & x_2 & x_3 & x_2 + x_3 & x_1 + x_3 & x_1 + x_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

### Solution de l'Exercice 32

1)



bits	0	1	1	0	0	0	1	0	1	1	1
------	---	---	---	---	---	---	---	---	---	---	---

2) Un signal transporte un bit. La rapidité de modulation R et le débit D ont la même valeur . Comme  $R = 1/D$  , on a

Durée $\Delta$ :	0,5 microseconde
------------------	------------------

### Solution de l'Exercice 33



1)  $T = 3t_e + 2t_p + t_{ack}$  avec  $t_e = 1024/(2.10^6) = 0,5. 10^{-3} \text{ s}$  ;  $t_p = 10^4/(3.10^8) = 0,033.10^{-3} \text{ s}$  ;  $t_{ack} = 64/(2.10^6) = 0,032. 10^{-3} \text{ s}$

Durée :	$T = 1,598.10^{-3} \text{ s}$
---------	-------------------------------

2)  $q = 3t_e / T$

Taux d'occupation :	0,94
---------------------	------

3) nombre de trames =  $(8.10^6)/(1024 - 80) = 8475$  trames ce qui nécessite 2825 rafales, donc  $4514 \text{ s} = 75 \text{ min} = 1,25 \text{ h}$

Durée totale de transmission :	1,25 h
--------------------------------	--------

### Solution de l'Exercice 34

N(S) =	5
N(R) =	5

### Solution de l'Exercice 35

Par seconde, on a 8000 mesures du signal et chaque mesure est codée sur 8 bits ; il faut donc un débit de  $8 \times 8000 = 64\,000$  bits/s

Débit :	64 Kbits/s
---------	------------

### Solution de l'exercice 36

a) Le nombre de voies (appelées IT) est  $2 \text{ Mbits/s} / 64 \text{ Kbits/s} = 32$  voies ( en fait 2 sont utilisées pour la gestion de la liaison)

b) Il suffit de prendre 3 canaux (3 IT) de la trame MIC.

## Solution de l'Exercice 37

a) En se basant sur le dessin, on voit que  $T = t_e + 2t_p$ .

$$t_e = L/D \text{ et } t_p = d/v \text{ d'où } T = L/D + 2d/v$$

$$b) \theta = t_e/T = t_e/(t_e + 2t_p) = 1/(1 + 2a) \text{ avec } a = dD/Lv$$

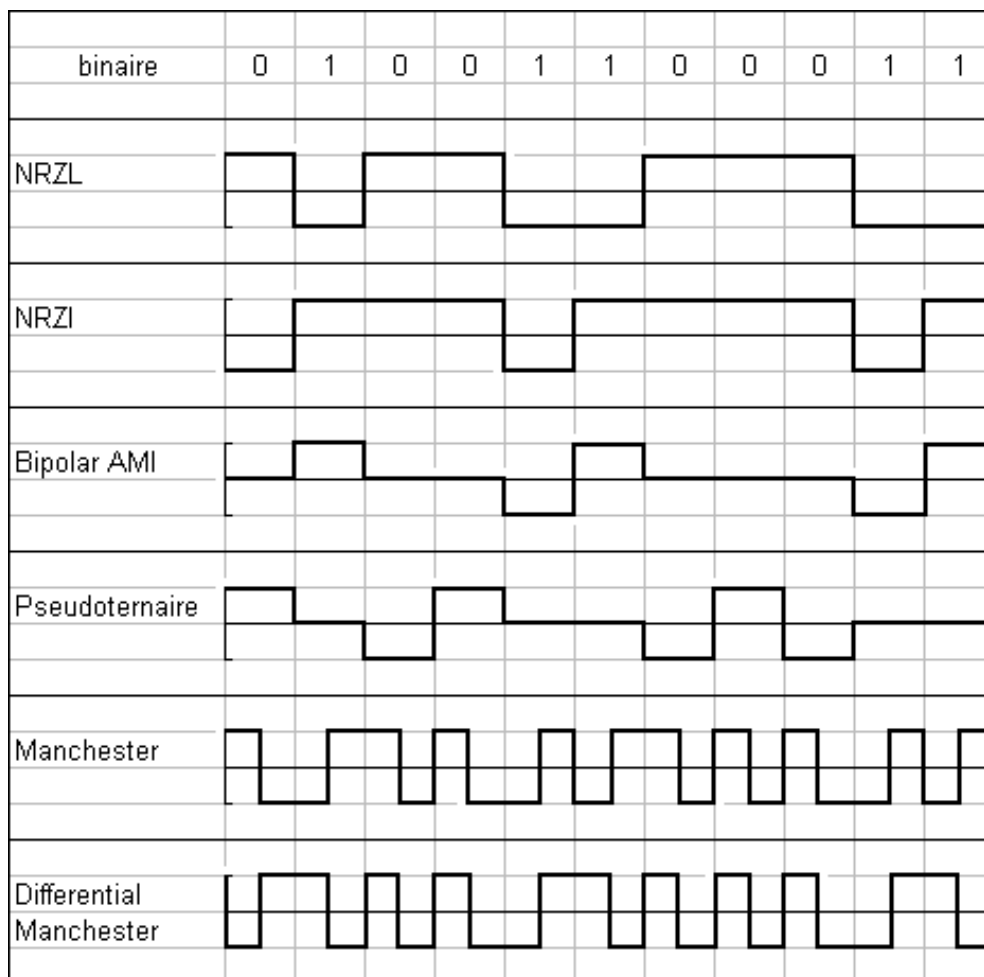
$$c) a = 10^3 \times 64 \times 1024 / (1024 \times 2 \times 10^8) = 32 \times 10^{-5} \quad \text{On en déduit que pratiquement, } \theta = 1$$

$$d) a = 10^3 \times 155 \times 10^6 / (53 \times 2 \times 10^8) = 14,6 \quad \text{On en déduit } \theta = 0,03$$

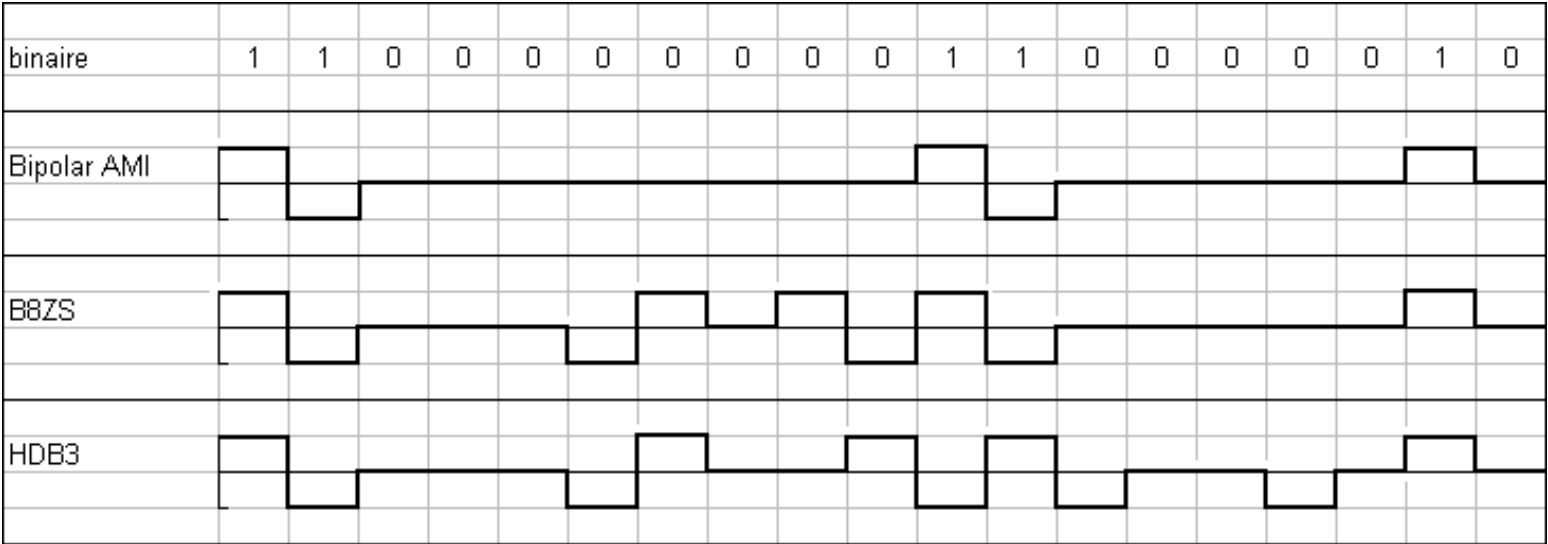
e) Pour des débits moyens, le protocole fonctionne bien ; pour des débits élevés, le protocole est quasi inutilisable.

## Solution de l'exercice 38

1)



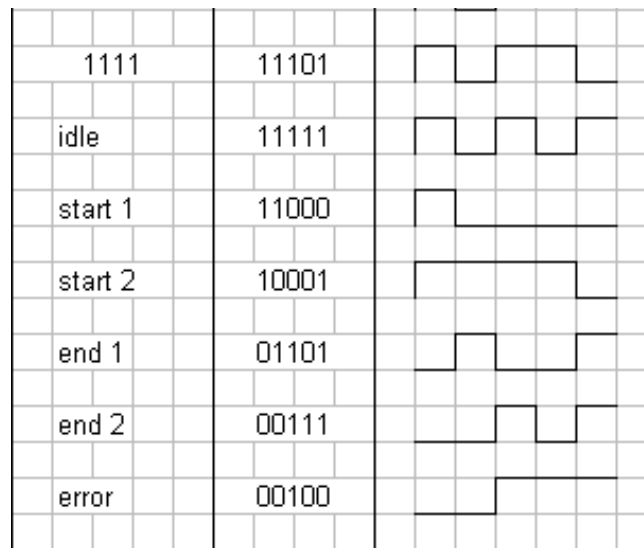
2)



Solution de l'exercice 39

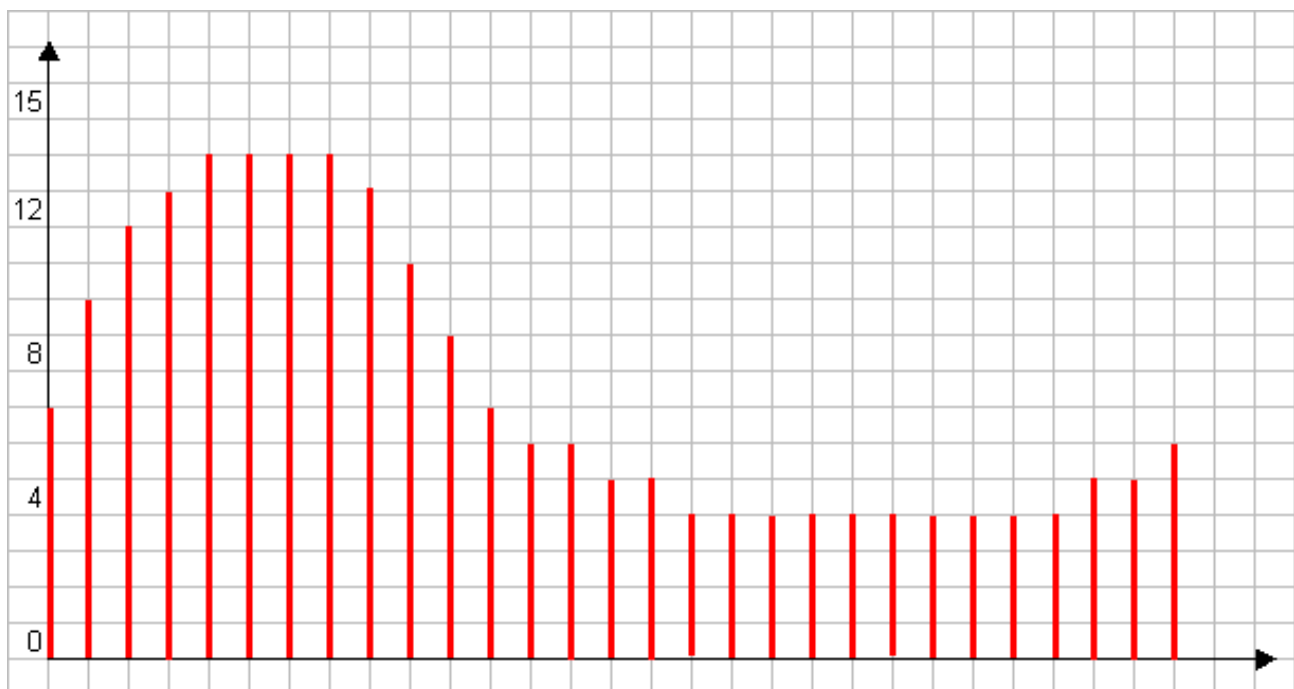
On vérifie aisément que le codage utilisé est NRZI.

mot de 4 bits	codage	signal
0000	11110	
0001	01001	
0010	10100	
0011	10101	
0100	01010	
0101	01011	
0110	01110	
0111	01111	
1000	10010	
1001	10011	
1010	10110	
1011	10111	
1100	11010	
1101	11011	
1110	11100	
1111	11101	



### Solution de l'exercice 40

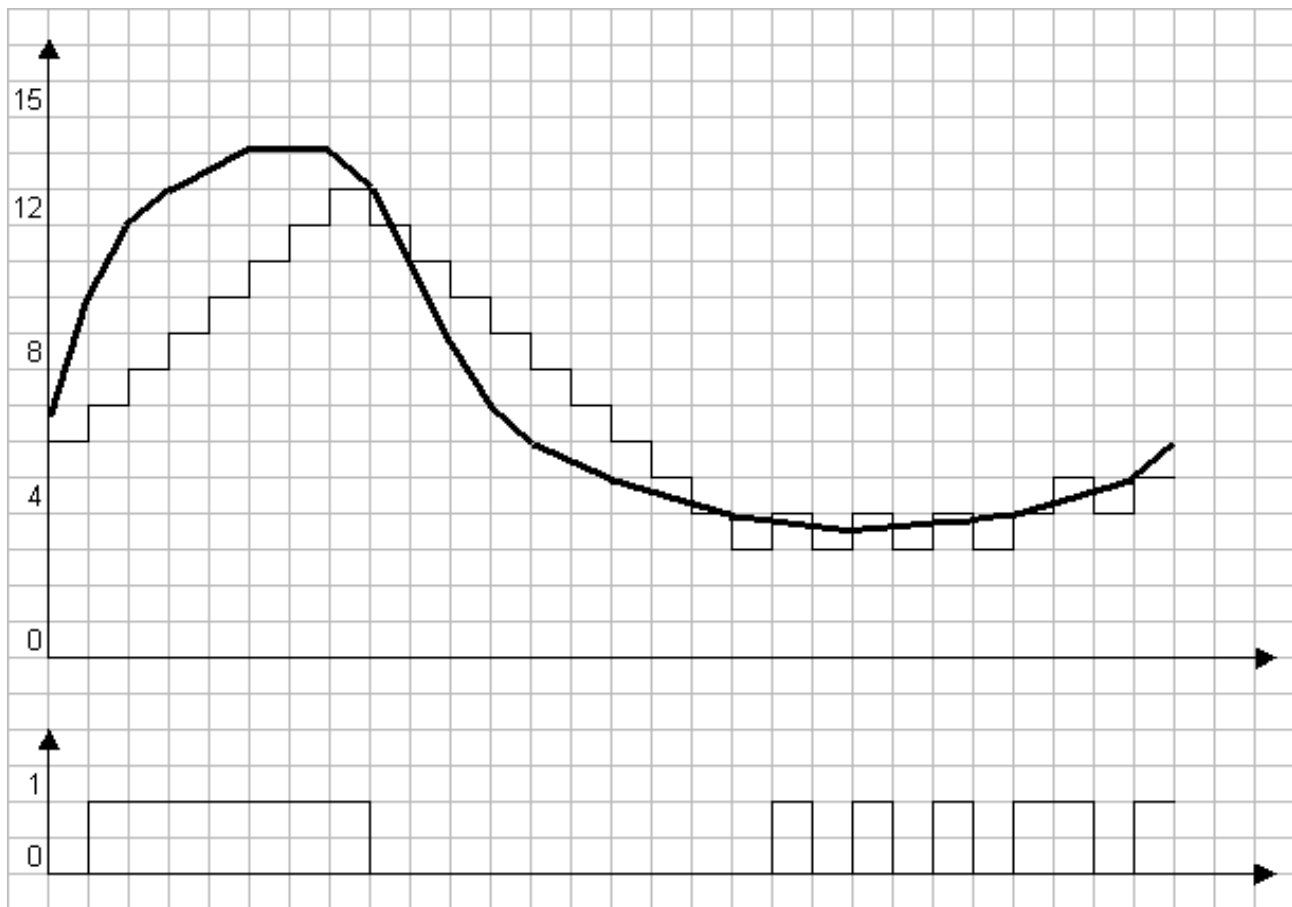
1) Après échantillonnage et quantification, on obtient une série de mesures :



d'où le codage de la donnée analogique (chaque mesure sur 4 bits) :

0111 1010 1100 1101 1110 1110 1110 1110  
1101 1011 1001 0111 0110 0110 0101 0101  
0100 0100 0100 0100 0100 0100 0100 0100  
0100 0100 0101 0101 0110

2)



Le codage est donc :

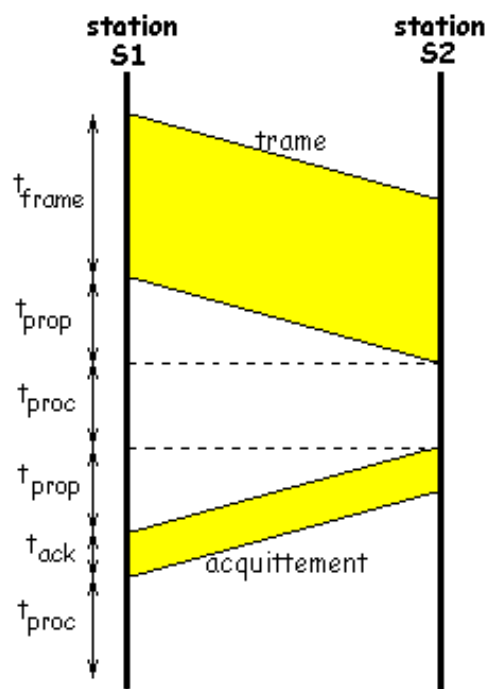
0111111100000000001010101101

## Solution de l'exercice 41

1a)

En se basant sur le schéma ci-contre, on en déduit très aisément :

$$T = t_{\text{prop}} + t_{\text{frame}} + t_{\text{proc}} + t_{\text{prop}} + t_{\text{ack}} + t_{\text{proc}}$$



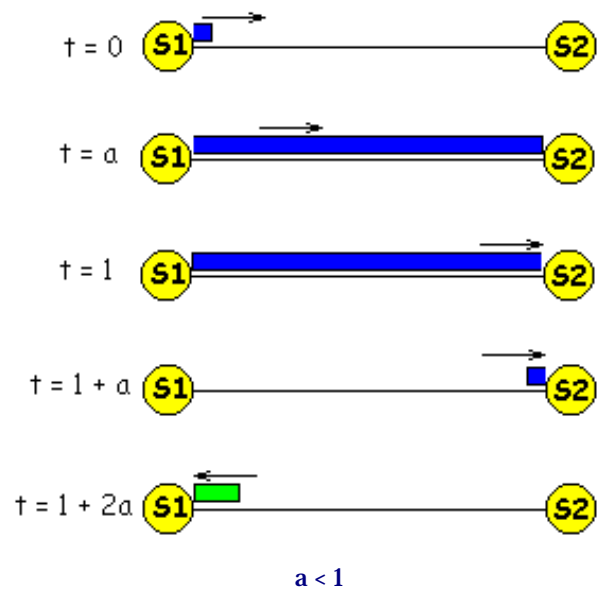
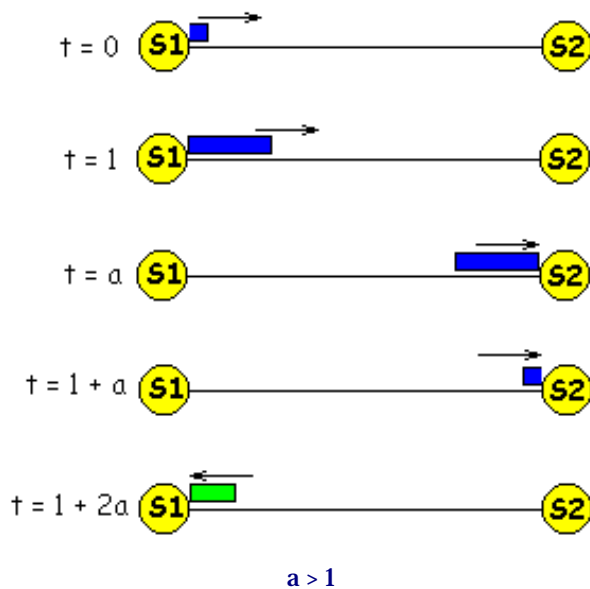
On peut négliger tous les termes de la somme précédente sauf  $t_{\text{frame}}$  et  $t_{\text{prop}}$

$$\text{d'où : } T_n = n T = n(2t_{\text{prop}} + t_{\text{frame}})$$

$$\theta = 1/(2a+1)$$

$$t_{\text{prop}} = d/v \quad t_{\text{frame}} = L/D \quad \text{d'où } a = (dD)/(vL)$$

1b)



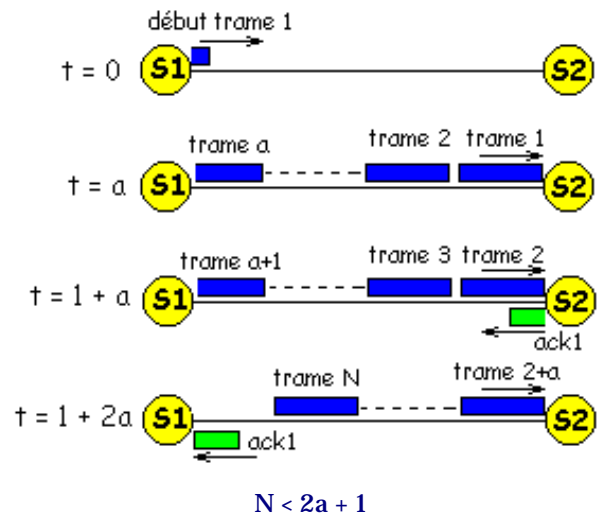
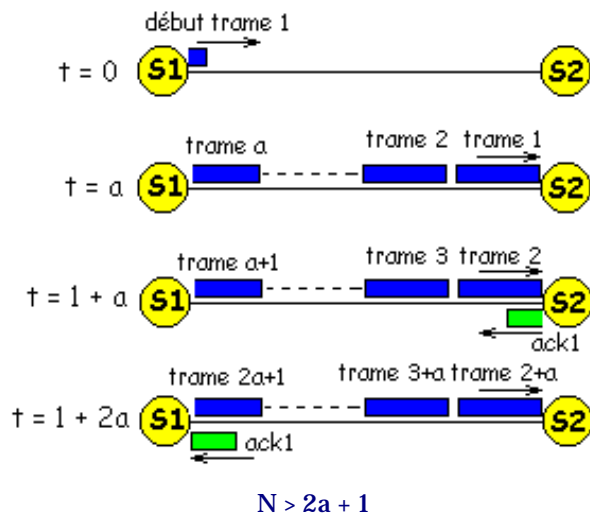
1c1)

$t_{\text{frame}} = (8 \times 53)/(155,52 \times 10^6) = 2,7$  microsecondes. Pour la fibre optique  $v = 3.10^8$  m/s et  $t_{\text{prop}} = 0,33 \times 10^{-2}$  s.  $a = 1200$  et  $\theta = 0,0004$  (désastreux).

1c2)  $t_{\text{frame}} = (1000/10^7) = 10^{-4}$  s.  $t_{\text{prop}} = 1000/(2.10^8) = 0,5 \times 10^{-5}$  s.  $a = 0,05$  et  $\theta = 0,9$  (très satisfaisant)

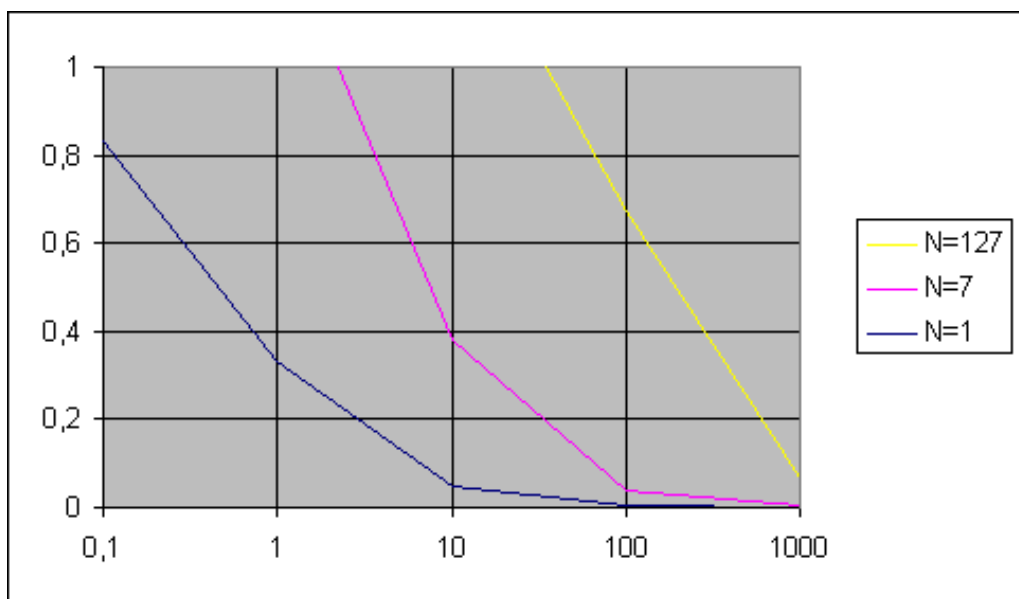
1c3)  $d = 1000$  m  $a = 1,44 \times 10^{-4}$   $\theta = 1$  (très bon)  
 $d = 5000$  km  $a = 0,72$   $\theta = 0,4$  (efficacité moyenne)

2)



Dans le cas où  $N > 2a+1$ , la ligne est toujours occupée donc  $\theta = 1$

Dans le cas où  $N < 2a+1$ , la durée d'émission est  $N$  pendant le temps  $2a+1$ , donc  $\theta = N/(2a+1)$



3a)

Imaginons que l'on effectue  $k$  tentatives pour transmettre une trame : les  $k-1$  premières sont erronées et la dernière est bonne ; la probabilité de cette situation est donc  $p_k = P^{k-1}(1-P)$ . Le nombre moyen  $r$  est donc :

$$r = \sum_{k=1}^{\infty} k p_k = \sum_{k=1}^{\infty} k P^{k-1} (1-P) = (1-P) [1 + 2P + 3P^2 + 4P^3 + \dots] = (1-P) \frac{d}{dP} [P + P^2 + P^3 + \dots] = (1-P) \frac{d}{dP} [1 + P + P^2 + P^3 + \dots - 1] \\ = (1-P) \frac{d}{dP} \left[ \frac{1}{1-P} - 1 \right] = (1-P) \frac{d}{dP} \left[ \frac{1}{1-P} - 1 \right] = (1-P) \frac{d}{dP} \left[ \frac{P}{1-P} \right] = (1-P) \cdot \frac{1}{(1-P)^2} = \frac{1}{1-P}$$

3b)

Le facteur  $1+2a$  qui représente le temps d'expédition d'une trame est à remplacer par  $r(1+2a)$ , puisqu'on effectue  $r$  tentatives. Donc, le taux d'occupation est :

$$\theta = (1-P)/(1+2a)$$

3c)

La méthode est la même ; il faut remplacer  $\theta$  par  $\theta/r$  d'où

$$\theta = 1 - P \text{ pour } N > 2a+1$$

$$\theta = N(1-P)/(1+2a) \text{ pour } N < 2a+1$$

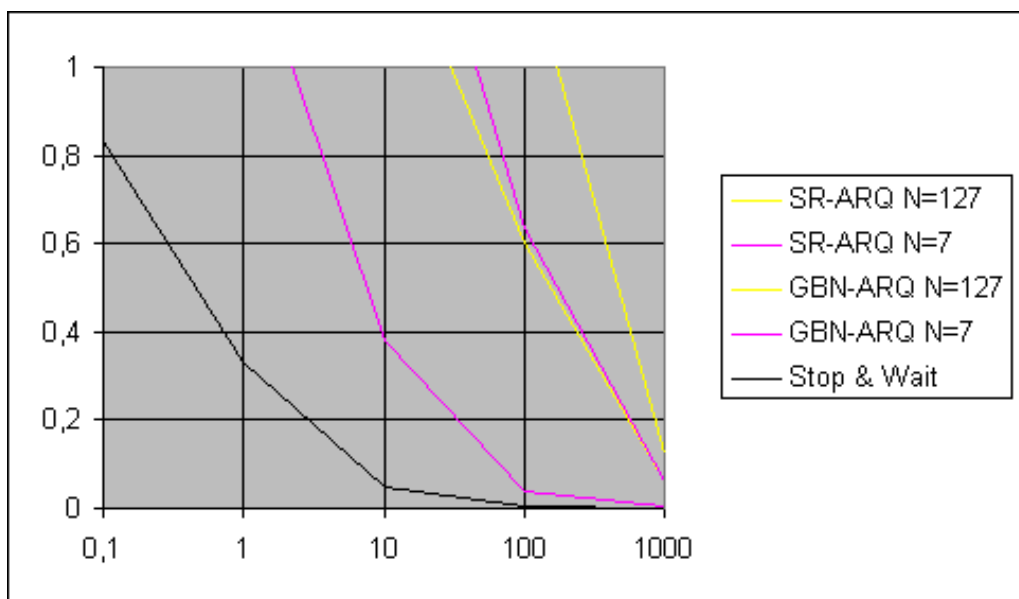
3d) Pour chaque erreur, il faut retransmettre K trames. Dans le cas de k tentatives, on a une transmission de trame puis k-1 fois la transmission de K trames ; donc en définitive, pour k tentatives, le nombre de trames à transmettre est  $1 + (k-1)K$  (au lieu de k dans le cas précédent). La valeur moyenne de r est donc :

$$r = \sum_{k=1}^{\infty} [1 + (k-1)K] P^{k-1} (1-P) = \sum_{k=1}^{\infty} [1 - K + Kk] P^{k-1} (1-P) = (1-P) \left[ (1-K) \sum_{k=1}^{\infty} P^{k-1} + K \sum_{k=1}^{\infty} k P^{k-1} \right] = (1-P) \left[ \frac{1-K}{1-P} + K \frac{1}{(1-P)^2} \right] = 1 - K + \frac{K}{1-P} = \frac{1-P+KP}{1-P}$$

En se basant sur la question 2, on considérera que l'on a  $K = 2a+1$  pour  $N > 2a+1$  et  $K=N$  pour  $N < 2a+1$ .

Donc, on a  $\theta = (1-P)/(1+2aP)$  pour  $N > 2a+1$

$$\theta = N(1-P)/((1+2a)(1-P+NP))$$



A noter que les méthodes Stop & Wait, GBN-ARQ N=1 et SR-ARQ N=1 donnent le même résultat.

## QCM

1) Un signal analogique est représenté par une grandeur physique

•



- - 
  - 
  -
- 

2) Un signal numérique est représenté par une grandeur physique

- - 
  - 
  - 
  -
- 

3) Une grandeur sinusoïdale est caractérisée par une amplitude, une fréquence et

- - 
  - 
  - 
  -
- 

4) Une voie de transmission élémentaire est un quadripôle contenant une résistance et

- - 
  - 
  - 
  -
- 

5) L'affaiblissement d'une ligne se mesure en

- - 
  - 
  - 
  -
- 

6) Si l'affaiblissement est de 20 db, le rapport  $|V_e/V_s|$  des ondes sinusoïdales d'entrée et de sortie est de

- 
- 
- 
- 
- 

---

7) La décomposition en série de Fourier d'un signal conduit à une superposition de signaux sinusoïdaux de fréquences  $f$ ,  $3f$ ,  $5f$ ,  $7f$ , .... Sachant que la bande passante est  $[5f, 15f]$ , combien de signaux sinusoïdaux élémentaires seront détectés à l'arrivée ?

- 
- 
- 
- 
- 

---

8) La ligne téléphonique pour le transport de la voix possède une largeur de bande de l'ordre de

- 
- 
- 
- 
- 

---

9) Une voie de transmission véhicule 16 types de signaux distincts ; sa rapidité de modulation est  $R = 1200$  bauds. Quel est le débit binaire de cette ligne ?

- 
- 
- 
- 
- 

---

10) Une voie de transmission véhicule 8 types de signaux distincts. Quelle est la quantité d'information binaire transportée par chaque signal ?

- 
- 
- 
- 
-

11) Le rapport signal sur bruit d'une voie de transmission est de 20 dB ; sa largeur de bande est de 3100 Hz. Quelle est, environ, la capacité théorique de cette voie ?

- 
- 
- 
- 
- 

---

12) Sur une voie de transmission, on constate que le nombre de communications par heure est 2 et que chaque communication a une durée moyenne de 3600 secondes. Quel est le trafic correspondant ?

- 
- 
- 
- 
- 

---

13) Sachant que, pour une voie de transmission, le nombre de transactions par communication est de 4200, la longueur moyenne d'une transaction est de 1200 bits, la durée moyenne d'une communication est de 3600secondes, le débit binaire est de 64 Kb/s, donner le taux d'occupation de la voie.

- 
- 
- 
- 
- 

---

14) Avec le code ASCII simple, on peut représenter

- 
- 
- 
- 
- 

---

15) Une transaction série entre deux ordinateurs nécessite

- 
- 
-

- -
- 

16) En transmission asynchrone, il s'écoule entre deux transmissions d'information

- - 
  - 
  - 
  -
- 

17) L'un des codes suivants est un code utilisé en transmission en bande de base

- - 
  - 
  - 
  -
- 

18) Pour transformer un signal numérique en un signal analogique, il faut utiliser

- - 
  - 
  - 
  -
- 

19) ETCD signifie

- - 
  - 
  - 
  -
- 

20) Dans ETTD, le dernier D signifie

- 
-

- - 
  -
- 

21) Pour numériser un son analogique, on effectue un échantillonnage, puis

- - 
  - 
  - 
  -
- 

22) Lorsqu'on partage une voie de transmission entre plusieurs communications de messages de manière à partager la bande passante entre les diverses communications, on effectue

- - 
  - 
  - 
  -
- 

23) Pour effectuer un multiplexage temporel, il faut

- - 
  - 
  - 
  -
- 

24) Les erreurs les plus fréquentes observées dans une transmission de message sont

- - 
  - 
  - 
  -
- 

25) La distance de Hamming entre  $m_1 = (10101010)$  et  $m_2 = (10111110)$  est

-

- 
- 
- 
- 

26) Pour détecter 3 erreurs il faut que la distance de Hamming minimale soit

- 
- 
- 
- 
- 

27) Pour corriger des erreurs jusqu'à l'ordre 2, il faut que la distance de Hamming minimale soit :

- 
- 
- 
- 
- 

28) Pour corriger des erreurs jusqu'à l'ordre 1 et détecter des erreurs jusqu'à l'ordre 2, il faut que la distance de Hamming minimale soit

- 
- 
- 
- 
- 

29) Soit un code linéaire (6,3) dont la matrice est

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Quelle est l'information codée correspondant à l'information utile 111 ?

- 
- 
-

- -
- 

30) Soit le polynôme générateur  $x^3 + x + 1$  du code (7,4). Donner le codage de l'information utile 1111.

- - 
  - 
  - 
  -
- 

31) Dans la procédure HDLC, un fanion a pour code

- - 
  - 
  - 
  -
- 

32) Dans HDLC, une demande de connexion s'effectue avec

- - 
  - 
  - 
  -
- 

33) Dans la procédure HDLC, le champ de détection d'erreur s'appelle

- - 
  - 
  - 
  -
- 

34) Dans la procédure HDLC, le polynôme générateur est

- 
-

- - 
  -
- 

35) Dans la procédure HDLC, une trame I comporte deux numéros, le numéro de trame et

- - 
  - 
  - 
  -
- 

36) Dans la procédure HDLC, si on émet une trame I de numéros  $N(S) = 3$  et  $N(R) = 2$ , la trame reçue de numéro 1 est- elle acquittée ?

- - 
  - 
  - 
  -
- 

37) Dans la procédure HDLC, si on émet une trame de numéros  $N(S) = 3$  et  $N(R) = 2$ , quel est le numéro de la prochaine trame attendue ?

- - 
  - 
  - 
  -
- 

38) Dans la technique du datagramme, les paquets

- - 
  - 
  - 
  -
- 

39) Dans la méthode du circuit virtuel



- 
- 
- 
- 
-

Ministère de l'Enseignement Supérieur et des recherches scientifiques  
Université Virtuelle de Tunis

Intitulé du chapitre :

**Technologies des réseaux de communication**

Nom de l'auteur :

**Gérard-Michel Cochard & Edoardo Berera  
& Michel Besson Thierry Jeandel**

Cette ressource est la propriété exclusive de l'UVT. Il est strictement interdit de la reproduire à des fins commerciales. Seul le téléchargement ou impression pour un usage personnel (1 copie par utilisateur) est permis.

# Architecture physique des réseaux

---

## Sommaire:

[Introduction](#)

[Réseaux de télécommunication](#)

[Commutation de circuits](#)

[Diffusion](#)

[Réseaux d'ordinateurs](#)

[Commutation de paquets](#)

[PAN, LAN, MAN, WAN](#)

---

## Introduction

Du point de vue de l'utilisateur deux grands réseaux s'imposent :

- le réseau téléphonique
- le réseau Internet

Ces deux exemples sont typiques de deux classes de réseaux :

- les réseaux de télécommunication
- les réseaux d'ordinateurs

On parle depuis 30 ans de convergence entre les télécommunications et l'informatique mais les deux types de réseaux, bien que basés sur les mêmes technologies opto-électroniques et malgré leur complémentarité et leur interdépendance, restent assez différents en termes de services rendus, terminaux employés et coûts d'utilisation.

## Questions

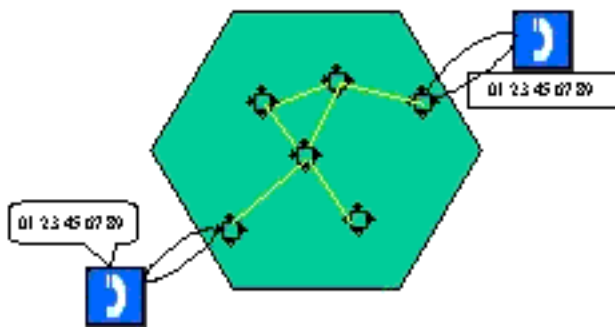
1. Connaissez-vous d'autres exemples de réseaux de télécommunications ?
2. Connaissez-vous d'autres exemples de réseaux d'ordinateurs ?

---

## Réseaux de télécommunications

### Le réseau téléphonique

# Réseau téléphonique



Principe de fonctionnement:  
commutation de circuits

- Service de base: appels locaux et longue distance
- Usages
  - voix, fax
  - accès Internet
- Tarifs
  - Durée, Distance
- Eléments
  - terminaux simples
  - commutateurs
  - boucle locale 2 fils cuivre
  - artères en fibre optique

## Services de base

Les services de base fournis par le réseau téléphonique sont les appels locaux et les appels longue distance grâce à un plan de numérotation mondial.

## Usages

L'usage principal est la communication de la voix, mais grâce à la modulation des signaux numériques avec des modems il est possible d'utiliser le réseau téléphonique aussi pour envoyer des fax ou pour accéder à l'Internet à partir de son ordinateur personnel.

## Tarifs

Les coûts d'utilisation sont typiquement basés sur la durée de l'appel et la distance entre les correspondants.

## Elements et architecture physique

Le réseau téléphonique est constitué de **terminaux simples**, téléphones, fax, modems, de **noeuds de commutation** sous la forme de commutateurs dans les centraux téléphoniques ou d'autocommutateurs, aussi appelés PABX pour Private Automatic Branch Exchange, dans les entreprises, et de **lignes de transmission**.

L'abonné est raccordé à la centrale téléphonique par une boucle locale typiquement constitué de 2 fils en cuivre, sur des distances de l'ordre de quelques kilomètres, ou d'une liaison sans fils appelée Boucle

## Locale Radio (BLR).

Les centraux téléphoniques sont raccordés entre eux par des artères longue distance typiquement en fibre optique ou par des liaisons radio appelées des faisceaux hertziens.

---

## Commutation de circuits

Le principe de fonctionnement est la commutation de circuits, c-à-d la création et le maintien d'un circuit (qui à l'origine était électrique, mais aujourd'hui seulement logique) à usage exclusif des deux correspondants pendant toute la durée de l'appel.

Les commutateurs et les lignes de transmission constituent le réseau de transport (de la voix) du réseau téléphonique. Nous verrons plus loin en parlant de réseaux logiques que les réseaux téléphoniques modernes sont en réalité basés aussi sur un autre réseau appelé le réseau de signalisation.

---

## Diffusion

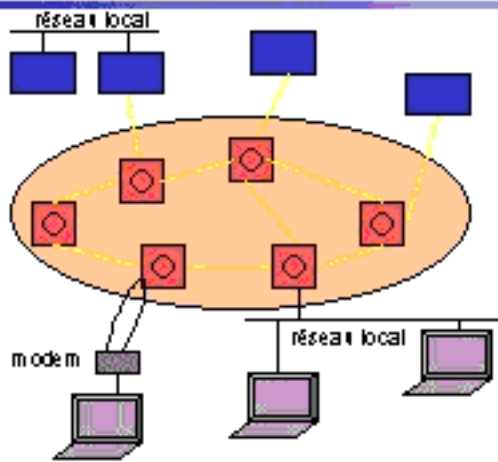
### Questions

1. Quels sont les services de base, terminaux, coûts d'utilisation et architecture des réseaux de radio et de télévision ?
- 

## Réseaux d'ordinateurs

### Le réseau global Internet

# Internet



Principe de fonctionnement:  
commutation de paquets (de données)

- INTERNET:  
INTERconnected NETworks
  - Le réseau des réseaux
- Service de base:  
interconnexion d'ordinateurs  
et de réseaux d'ordinateurs
- Eléments
  - Ordinateurs individuels
  - Serveurs
  - Routers
  - Boucles et réseaux locaux
  - Artères en fibres optiques

## Service de base

Le nom Internet vient de "interconnected networks". Le service de base est bien l'interconnexion d'ordinateurs isolés, comme l'ordinateur individuel à la maison ou un ordinateur portable doté d'un modem, et de réseaux d'ordinateurs, comme les réseaux locaux d'entreprise et de campus. Internet est le réseau des réseaux.

L'objectif d'Internet est de relier tous les objets qui peuvent avoir une adresse Internet et qui savent utiliser le protocole IP (Internet Protocol). Aujourd'hui il s'agit essentiellement des ordinateurs, mais demain les téléphones portables, les téléviseurs et même les appareils électroménagers pourraient avoir une adresse Internet (par exemple pour appeler automatiquement le service après vente en cas de panne).

## Eléments et architecture physique

Internet est constitué d'**ordinateurs serveurs** qui fournissent un ensemble de services (information, messagerie, etc.) et d'**ordinateurs clients** que les utilisateurs utilisent pour accéder aux services de l'Internet. Ces ordinateurs sont interconnectés par des **ordinateurs spécialisés** dans l'acheminement des données appelés **routeurs**.

## Commutation de paquets

Le principe de fonctionnement de l'Internet est la commutation de paquets. L'information échangée entre ordinateurs est découpée en paquets de quelques dizaines jusqu'à quelques milliers d'octets qui sont acheminés grâce à l'adresse de destination (et l'adresse de la source pour pouvoir envoyer les réponses) écrite dans l'en-tête de chaque paquet.

Les serveurs, les clients et les routeurs constituent les noeuds du réseau. Le protocole IP (Internet Protocol) utilise un plan de numérotation mondial (adresse IP sur 32 bits pour le protocole IP version 4 actuellement utilisé et adresse IP sur 128 bits pour le protocole IP version 6 qui sera déployé dans les années à venir).

Les noeuds sont interconnectés par des artères en fibres optiques ou faisceaux hertziens ou encore liaisons satellitaires à haut débit dans la partie centrale du réseau où les trafics sont agrégés, par des liaisons téléphoniques pour les ordinateurs isolés et par des câbles téléphoniques ou coaxiaux ou liaisons sans fil dans le cadre des réseaux locaux de campus et d'entreprise qui relie un ensemble d'ordinateurs dans les mêmes locaux.

### Coût d'utilisation

Coût au temps passé ou forfait pour l'accès via le réseau téléphonique. Coût indépendant de la distance entre l'ordinateur doté de navigateur et l'ordinateurs serveur d'informations (ou autre application).

---

## PAN, LAN, MAN et WAN

### Question

1. Comment peut-on classer les réseaux d'ordinateurs en fonction de leur taille ?

---

auteur : Edoardo Berera - Miage Nice

[EB](#) date de dernière modification : 22 mai 2002

# Architecture logique des réseaux

## Sommaire:

[Introduction](#)

[Réseaux de gestion \(OSS\)](#)

[Réseaux de signalisation](#)

[Réseaux de transport](#)

[Réseau intelligent \(IN\)](#)

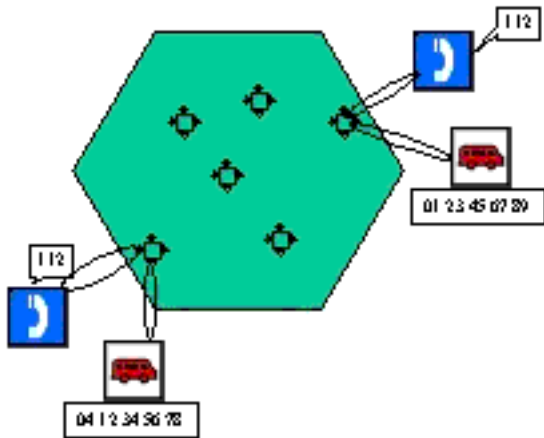
[Telecommunications Management Network \(TMN\)](#)

[Réseaux de téléphonie cellulaire](#)

## Introduction

### Exemple: services intelligents

## Services intelligents



- Appels d'urgence
  - 112 en Europe
  - 911 en USA
    - où est effectuée la traduction du numéro ?
- Où est l' "intelligence" ?
  - numéros "800" ("verts")
    - Services
      - Appels gratuits
      - Tarifs spéciaux

Pour comprendre l'architecture logique des réseaux nous allons prendre l'exemple de services très connus du réseau téléphonique: les appels d'urgence et les numéros verts (ou numéros 800).

En cas d'urgence il suffit d'appeler le numéro 112 partout en Europe ou le numéro 911 partout aux Etats-Unis pour obtenir immédiatement les services locaux d'urgence. Les numéros 112 et 911 ne correspondent pas à des numéros du plan de numérotation du réseau téléphonique mais désignent des services.



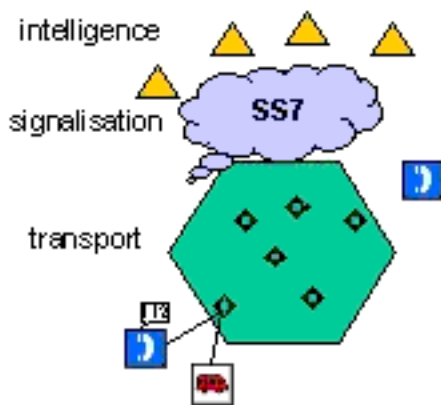
## Question

1. Où est effectuée la traduction du numéro 112 en numéro 04 12 34 56 78, par exemple à Nice, et en numéro 01 23 45 67 89, par exemple à Paris ?

Ces appels sont gratuits; vous pouvez appeler d'une cabine téléphonique sans carte ou argent ou d'un téléphone portable même si le crédit de votre carte est épuisé.

Les numéros verts se comportent de manière similaire; ils désignent souvent les centres d'appel de sociétés tout en laissant au réseau le soin de déterminer quel bureau appeler en fonction de la localisation de l'appelant, de l'heure de l'appel et de la disponibilité d'opérateurs pour répondre.

## Réseau Intelligent



- De serveurs du réseau fournissent l'"intelligence"
  - par ex.: la traduction des numéros
- Séparation entre
  - Réseau de transport de la voix
    - Commutation de circuits
  - Réseau de signalisation
    - Commutation de paquets
      - messages définis dans le protocole international SS7 Signaling System n°7

En réalité la structure d'un réseau téléphonique moderne est constituée de trois réseaux logiques:

- réseau de gestion
- réseau de signalisation
- réseau de transport

## Réseau de gestion (Operations Support Systems, OSS)

Le réseau de gestion comporte une série de serveurs qui sont utilisés par l'opérateur pour gérer le service de télécommunication sur le plan technique (configuration, surveillance, gestion des pannes, gestion des services) et sur le plan administratif (gestion des clients, facturation, marketing). En anglais ces derniers systèmes sont souvent appelés OSS pour Operations Support Systems, alors que les systèmes utilisés pour la gestion des services dits intelligents sont appelés des SCP pour Service Control Points et enfin les ordinateurs utilisés pour l'administration technique du réseau sont globalement désignés comme appartenant au TMN pour Telecommunications Management Network.

---

## Réseau de signalisation

Le réseau de signalisation est constitué de commutateurs de paquets, basés sur un protocole international défini par l'Union Internationale des Télécommunications (UIT, ou ITU en anglais) appelé SS7 pour Signaling System n°7. Ce protocole sert à transférer les données de l'appel, par exemple le numéro appelé mais aussi le numéro de l'appelant, de la centrale téléphonique de l'appelant jusqu'à la centrale téléphonique de l'appelé pour déterminer si le numéro appelé est libre ou occupé.

Ce même protocole sert aussi à transférer les données de l'appel aux serveurs de gestion des services comme les numéros verts pour obtenir la traduction du numéro logique appelé (les numéros 800) en numéro réel en fonction d'un script qui aura été défini par l'opérateur pour chaque client. En anglais les commutateurs de paquets du réseau de signalisation sont appelés STP pour Signaling Transfer Points.

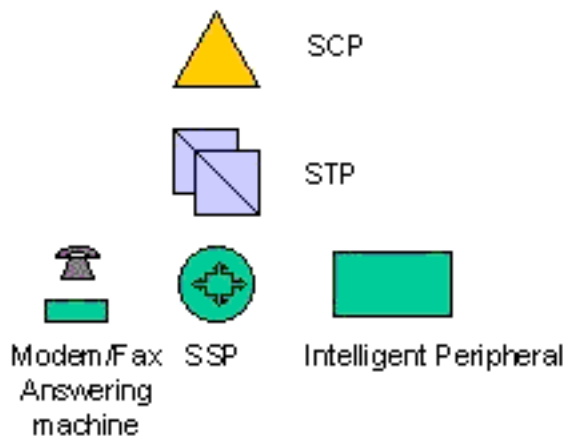
---

## Réseau de transport

Nous avons déjà parlé du réseau de transport dans la section sur l'architecture physique des réseaux. Ajoutons simplement que les commutateurs sont appelés des SSP pour Service Switching Points en anglais.

## Réseau Intelligent. IN pour Intelligent Network en anglais

### Eléments du Réseau Intelligent



- Service Control Point (SCP)
- Signaling Transfer Point (STP)
- Service Switching Point (SSP)  
normalement intégrés dans les commutateurs téléphoniques
- Intelligent Peripherals
  - Interactive Voice Response (IVR)
    - réponse vocale
  - Media Gateways
    - par ex. voix sur Internet
    - le signal voix numérisée du réseau téléphonique doit être mis dans des paquets pour pouvoir être acheminé par Internet

Les réseaux téléphoniques modernes sont un bel exemple de réseau d'ordinateurs avec l'intelligence du réseau, pour fournir les nouveaux services, située au coeur du réseau et gérée essentiellement par l'opérateur; des terminaux qui restent très simples (postes téléphoniques, fax, répondeurs, et modems) sont utilisés pour accéder aux services.

Le réseau Internet est par contre un exemple de réseau d'ordinateurs avec l'intelligence située à la périphérie du réseau dans des serveurs gérés par les très nombreux fournisseurs de services Internet et dans les ordinateurs des utilisateurs qui peuvent ultérieurement traiter localement l'information reçue.

Le coeur du réseau par contre est essentiellement constitué de routeurs dont le rôle essentiel est d'acheminer le plus rapidement possible les paquets de données, donc avec un minimum de traitement toutes les fois que cela est possible.

Il est important de bien comprendre les deux points de vue différents des "télécommunicants" (opérateurs et ingénieurs de télécommunications) et des "informaticiens" (fournisseur de services Internet et ingénieurs informaticiens) pour comprendre la dynamique de l'évolution des réseaux.

Les "télécommunicants" auront toujours tendance à ajouter des services au coeur du réseau pour garder la gestion des services et facturer les appels plus cher alors que les "informaticiens" auront toujours tendance à demander des débits élevés mais les services les plus simples possibles (la simple connectivité sans rien d'autre) pour réduire les dépenses de télécommunications sachant que

**l'intelligence des services peut mieux être fournie par une multitude de fournisseurs à l'extérieur du réseau.**

---

**auteur : Edoardo Berera - Miage Nice**

**[EB](#) date de dernière modification : 22 mai 2002**

# Architecture en couches des réseaux

## Sommaire:

[Introduction](#)

[Modèle de référence OSI \(Open Systems Interconnection\)](#)

[Notions de couche, service, protocole et interface](#)

[Couche Physique](#)

[Couche Liaison de données](#)

[Couche Réseau](#)

[Couche Transport](#)

[Couche Session](#)

[Couche Présentation](#)

[Couche Application](#)

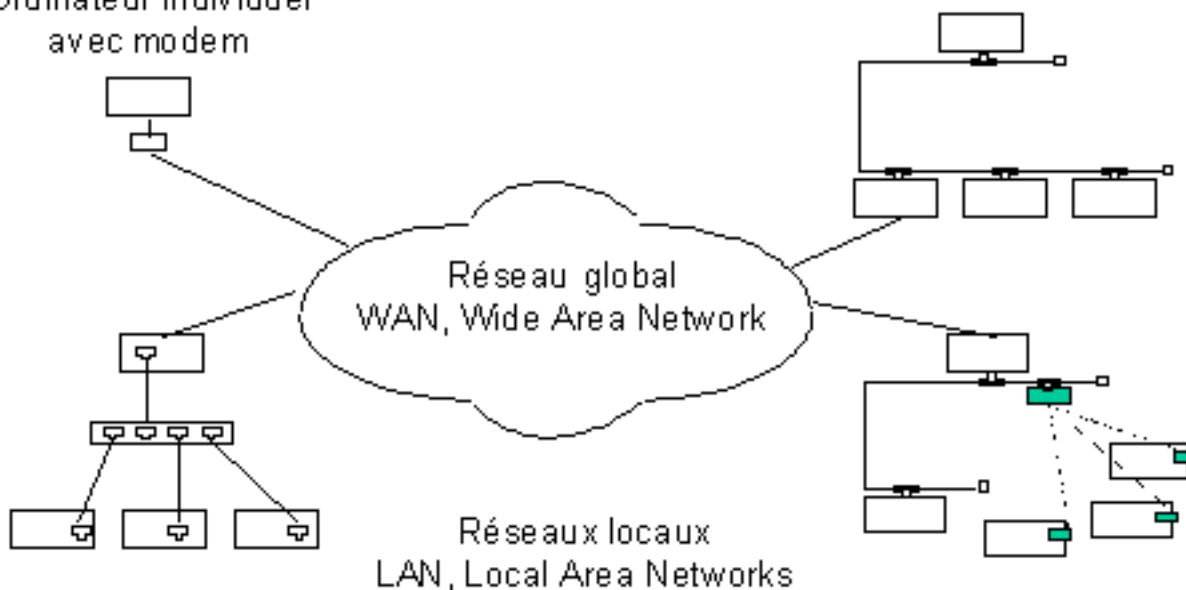
[Architecture du réseau Internet](#)

[Architectures propriétaires](#)

## Introduction

## Interconnexion de réseaux

Ordinateur individuel  
avec modem



L'interconnexion de réseaux est un problème complexe car les réseaux sont souvent hétérogènes. Pour

qu'un ordinateur puisse communiquer avec un autre, dont on connaît le nom, il faudra d'abord trouver son adresse.

### Question:

1. Quelle est l'adresse de votre ordinateur ?

Le coeur des réseaux longue distance est souvent constitué de routeurs fortement maillés, c-à-d interconnectés par des nombreux chemins différents; il faudra donc trouver le meilleur chemin.

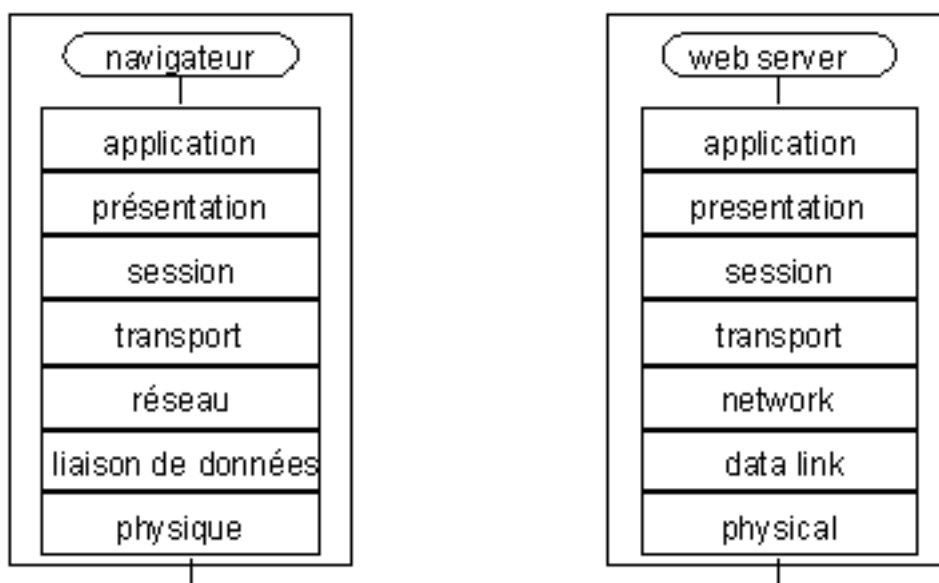
### Question:

1. Quels pourraient être les critères pour déterminer le "meilleur" chemin ?

---

## Modèle de référence OSI (Open Systems Interconnection)

### Les 7 couches du modèle OSI



Le modèle de référence OSI propose une décomposition du problème d'interconnexion des réseaux d'ordinateurs, en couches superposées à partir de la couche physique, c-à-d la couche qui met en oeuvre le logiciel de gestion de la porte de communication ou de la carte utilisées pour le raccordement au réseau, jusqu'a la couche application, c-à-d la couche qui met en oeuvre les services nécessaires aux applications qui utilisent le réseau, comme, par exemple, les logiciels de courrier électronique ou de navigation sur l'Internet.

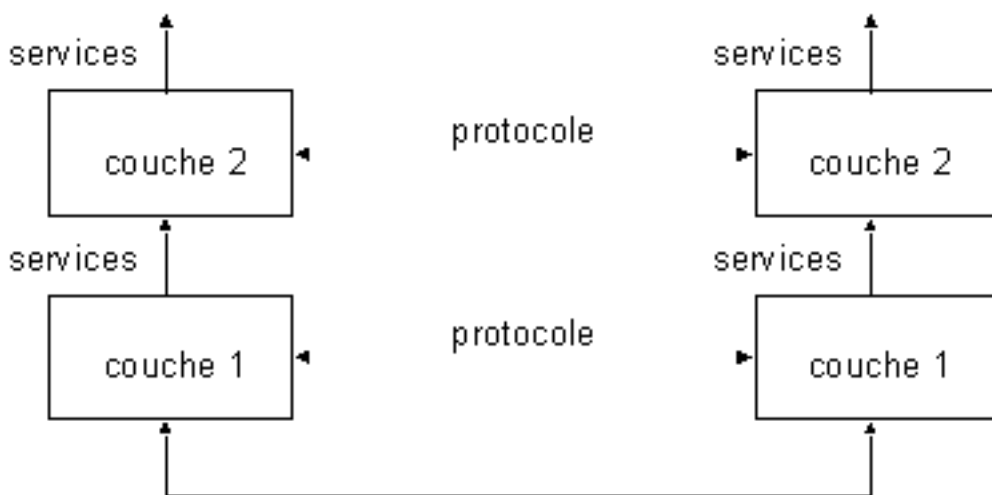
Ce modèle est devenu une norme internationale sous la double référence ISO 7498 et ITU-T X.200 Series et une norme française sous la référence AFNOR NF Z 70-001.

---

## Notions de couche, service, protocole et interface

### Services et Protocoles

---



---

Les logiciels mis en oeuvre dans chaque couche d'un ordinateur résolvent une partie des problèmes de communication, en utilisant des protocoles de communication, c-à-d des ensembles de règles, procédures et messages définis et standardisés, pour communiquer avec la couche homologue de l'ordinateur distant.

Chaque couche offre des services à la couche de niveau supérieur. Cette méthode simplifie l'écriture des logiciels de la couche car ça lui permet de traiter une autre partie des problèmes de communication en s'appuyant sur les services fournis par la couche de niveau inférieur.

Il suffit de parcourir les sept couches du modèle de référence OSI pour s'en convaincre.

---

## Couche Physique

### Couche physique

---

application
présentation
session
transport
réseau
liaison de données
physique

- Fonctions
  - Emission et réception des signaux (radio) électriques (bits)
  - Sérialisation: octets  $\longleftrightarrow$  bits
- Exemples
  - Cartes réseau, connecteurs, câbles, modems, concentrateurs (hubs)

---

La fonction principale de la couche physique est de matérialiser l'interface entre l'ordinateur et le réseau pour pouvoir émettre et recevoir des signaux de communication. Ces signaux peuvent être de nature électrique, électromagnétique (radio) ou optique. La définition de connecteurs, des câbles ou antennes font partie de cette couche. En général on considère que les cartes réseau, les modems et les concentrateurs (hubs) en font aussi partie.

Une autre fonction de cette couche est de sérialiser l'information, c-à-d transformer les octets en éléments binaires (bits) ou vice versa pour pouvoir émettre ou recevoir sur les canaux de



communication. Cette transformation doit être effectué à un rythme qui est imposé par la vitesse (débit binaire) de l'interface.

Beaucoup d'autres fonctions peuvent être réalisées par cette couche; la détection de l'existence d'une communication en cours (Carrier Sense) ou d'une collision (Collision Detect) sur un réseau local Ethernet en sont deux exemples.

## Questions

1. Quels sont les débits binaires des modems ?
2. Et ceux des cartes Ethernet ?

---

## Couche Liaison de données

### Couche liaison de données

---

application
présentation
session
transport
réseau
liaison de données
physique

- Fonctions
  - Envoi et réception de messages (trames) à son proche (sur un lien direct)
  - Contrôle d'erreurs de transmission
- Exemples
  - PPP Point to Point Protocol
    - raccordement d'un ordinateur avec modem à un fournisseur d'accès Internet
  - Protocole Ethernet (IEEE802.3, IEEE802.11 b)
    - liaison avec ou sans fil en réseau local

---

La fonction de la couche liaison de données est l'envoi et la réception de messages, souvent appelés trames à ce niveau, à son proche, c-à-d à un ordinateur qui se trouve sur un lien direct (sans faire appel

à des systèmes intermédiaires, les fameux routeurs). Ce lien direct peut être permanent comme dans le cas le plus simple des réseaux locaux où les ordinateurs sont tous raccordés au même câble (ou au même concentrateur, qui peut être vue comme une prise multiple de réseau!) ou bien peut avoir été créé au préalable, par exemple, par une commutation de circuit sur le réseau téléphonique en appelant un fournisseur d'accès à Internet. Dans ce dernier cas le lien direct est temporaire.

Cette couche peut aussi faire un contrôle d'erreurs de transmission, en utilisant, par exemple, dans le cas des trames Ethernet les derniers quatre octets de la trame appelés Frame Check Sequence (FCS).

Deux protocoles très utilisés à ce niveau sont:

- Point to Point Protocol (PPP) pour la communication d'un ordinateur avec modem à un fournisseur d'accès Internet (en utilisant le réseau téléphonique)
- IEEE802.3, IEEE802.11b (protocoles Ethernet) pour le raccordement en réseau local avec ou sans fils

---

## Couche Réseau

# Couche réseau

application
présentation
session
transport
<b>réseau</b>
liaison de données
physique

- Fonctions
  - Acheminer les messages (paquets) de proche en proche en fonction de leur adresse de destination (routage)
  - Fragmenter les messages en paquets
- Exemples
  - IP Internet Protocol
  - **Inter**connected **Net**works
  - IPv4, version 4, version actuelle
  - IPv6, version 6, la prochaine version

La fonction de la couche réseau est d'acheminer les messages, souvent appelés soit paquets, soit datagrammes, de proche en proche jusqu'à destination en fonction de leur adresse. Cette fonction est appelée le routage; elle fait typiquement appel à des ordinateurs spécialisés, appelés routeurs, qui sont des systèmes intermédiaires sur la route qui va de la source à la destination.

## Question:

1. Quel est le chemin, c-à-d la liste des systèmes intermédiaires, entre votre ordinateur et le serveur de l'Education Nationale, [www.education.gouv.fr](http://www.education.gouv.fr) ?

Pour réaliser l'interconnexion de tous les réseaux d'ordinateurs à travers le monde entier il faut que ce protocole soit unique. Aujourd'hui il s'agit bien du protocole Internet IP (Internet Protocol). Ce protocole est dans sa version 4, caractérisée par des adresses sur 32 bits. L'évolution de l'Internet requiert le passage à la version 6 (la version 5 a été définie, mais n'a pas été adoptée), qui est caractérisée par des adresses beaucoup plus longues, représentées sur 128 bits.

## Questions

1. Un espace d'adressage qui utilise des adresses représentées sur 32 bits permet de définir combien d'adresses différentes ?
2. Et si les adresses sont représentées sur 128 bits ?

## Couche Transport

### Couche transport

application
présentation
session
transport
réseau
liaison de données
physique

- Fonctions
  - Envoyer et recevoir les messages de bout en bout, c-à-d de la source jusqu'à destination
  - Retransmettre, éventuellement, les messages non reçus
- Exemples
  - TCP Transmission Control Protocol
    - transport avec garanties
  - UDP User Datagram Protocol
    - transport sans garantie ("best effort"), donc sans retransmission

Le rôle du service de transport est de transporter les messages de bout en bout, c-à-d de la source jusqu'à la destination, donc d'un bout à l'autre du réseau, sans se préoccuper du chemin à suivre car ce problème a déjà été traité par la couche inférieure de réseau.

Il y a plusieurs exemples de protocoles de transport. Dans le monde Internet les plus connus sont:

- TCP Transmission Control Protocol
- UDP User Datagram Protocol
- RTP Realtime Transport Protocol

Le choix dépend du type d'application et des services demandés.

Les applications de transfert de fichiers, de courrier électronique et de navigation sur le web requièrent des garanties de transmission sans erreurs et de retransmission en cas d'erreur. Dans le cas

de messages longs, le fait de découper un message en paquets plus courts peut donner lieu à la remise des paquets à l'ordinateur de destination dans le désordre. Le protocole TCP s'occupe de résoudre ces problèmes, au prix d'une certaine complexité du protocole.

D'autres applications comme les requêtes aux annuaires électroniques ( pour obtenir la correspondance entre un nom d'ordinateur et son adresse) ou les applications de gestion de réseau préfèrent utiliser un protocole plus léger mais plus rapide car les messages sont typiquement très courts et en cas d'erreurs ou d'absence de réponse, ils peuvent être répétés sans problèmes. Le protocole UDP est typiquement utilisé dans ces cas.

D'autres applications encore comme la téléphonie et la vidéoconférence sur Internet ont des contraintes de temps réel. La transmission de la voix et de la vidéo ne peuvent pas tolérer les variations de délais, appelées gigue, dans l'acheminement des paquets car les accélérations et ralentissements qui en résulteraient dans la restitution de la voix ou de l'images nuiraient gravement à la qualité de la transmission. Le protocole RTP, qui est utilisé en complément du protocole UDP, traite ces problèmes.

---

## Couche Session

## Couche session

---

application
présentation
session
transport
réseau
liaison de données
physique

- Fonctions
    - Maintenir un contexte de communication (début/identification, fin, reprise en cas d'interruption) entre source et destination
    - Pas toujours nécessaire
  - Exemples
    - Login / Logout entre machines en réseau
    - Cette fonction est souvent intégrée directement dans les logiciels d'application qui utilisent des protocoles spécifiques
- 

La fonction de la couche session est de négocier et de maintenir un contexte de communication entre la source et la destination. En début de communication il s'agit de définir le mode de communication (half duplex ou full duplex) et les règles de la communication. En cas de problème de communication, par exemple d'interruption momentanée, les services de points de reprise devraient permettre de reprendre la conversation là où elle avait été interrompue.

En pratique ces fonctions sont souvent intégrées directement dans les logiciels d'application qui utilisent des protocoles spécifiques adaptés à chaque application particulière.

---

## Couche Présentation

# Couche présentation

application
présentation
session
transport
réseau
liaison de données
physique

- Fonctions
  - Représenter les données
- Exemples
  - ASCII
    - American Standard Code for Information Interchange
  - ISO 8859
    - ASCII plus caractères avec accents
  - ASN.1 Abstract Syntax Notation 1
    - Langage de description des données et règles de représentation (utilisé par ex. par les applications de gestion des réseaux)

Le rôle de cette couche est d'aider les différentes applications à représenter les données de manière indépendante des plates-formes/systèmes d'exploitation (Macintosh/Mac OS, Intel/Windows, etc.).

Il existe plusieurs standards pour représenter les données (caractères, chiffres, booléens, mais aussi des données plus complexes construites à partir de données simples, comme les dates, les énumérations (par exemple, lundi, mardi, etc.), jusqu'aux données d'applications spécifiques comme une feuille de calcul, une présentation, un document incluant texte, tables et images).

Certaines applications se limitent à l'utilisation du standard ASCII pour représenter les caractères sans accents. D'autres applications peuvent utiliser le standard international ISO 8859 pour pouvoir représenter les caractères avec accents.

D'autres applications encore peuvent utiliser un véritable langage de description de données (simples et complexes) avec des règles de représentation des données pour le transfert entre applications en réseau. Le standard ISO ASN.1 est un exemple utilisé dans le cadre des application de gestion de réseau.

La couche de présentation pourrait aussi fournir des services de cryptage de l'information.

Mais encore une fois cette couche est souvent intégrée directement dans les logiciels d'application.

---

## Couche Application

---

### Couche application

---

application
présentation
session
transport
réseau
liaison de données
physique

- Fonctions
  - Transfert de fichiers, courrier électronique, navigation Internet (requêtes/réponses), voix et vidéo sur Internet, gestion de réseau, etc.
- Exemples
  - FTP File Transfer Protocol
  - SMTP Simple Message Transfer Protocol
  - HTTP HyperText Transfer Protocol
  - RTP Real-time Transport Protocol
  - RTSP Real Time Streaming Protocol

---

Le rôle de la couche application est de fournir les services et les protocoles nécessaires aux applications qui souhaitent s'ouvrir sur le réseau. Il faut noter que les applications elles mêmes ne font pas partie de la couche application.

Les exemples de protocoles que nous pouvons classer dans cette couche sont très nombreux car les applications sont nombreuses et ne cessent de se développer.

Les protocoles les plus connus sont HTTP, FTP et SMTP pour naviguer sur le web, transférer des fichiers ou envoyer des messages électroniques.

Le protocole RTP (Realtime Transport Protocol) dont nous avons parlé à-propos de la couche transport peut aussi être classé dans la couche application (voir architecture Internet).



---

## Architecture du réseau Internet

# Internet

---

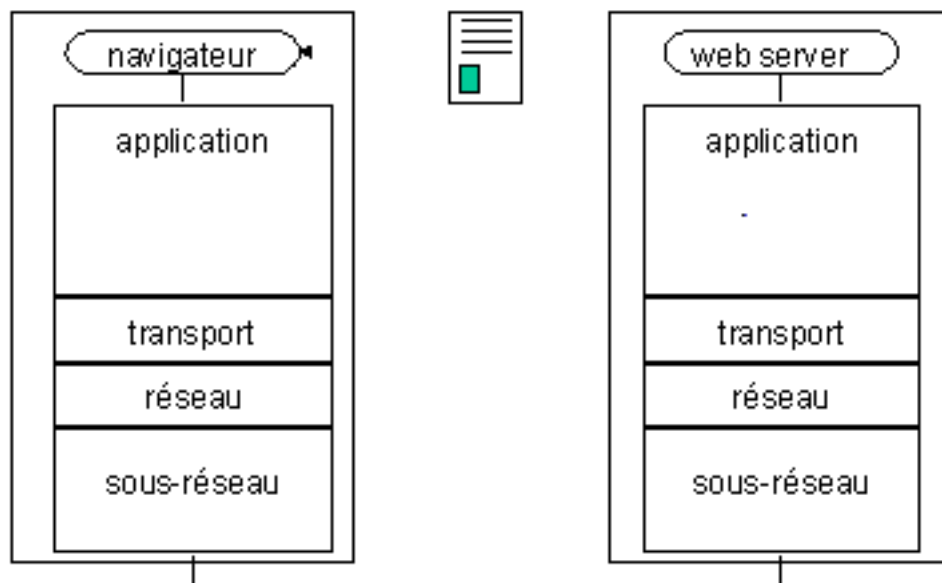
Modèle de référence OSI	Architecture Internet	Exemples de protocoles
application	application	FTP SMTP HTTP
présentation		
session		
transport	transport	TCP, UDP
réseau	réseau	IPv4, IPv6
liaison de données	sous-réseau	PPP, Ethernet
physique		

---

L'architecture des applications Internet a tendance à mettre en oeuvre les fonctions des couches session, présentation et application dans des protocoles qui intègrent les différentes fonctions en un seul protocole.

Cette approche contredit en partie la notion de couches mais se justifie pour des raisons de performance et de pragmatisme. Le modèle ISO et ses couches servent de référence pour poser les problèmes, alors que au moment de l'implémentation il est peut être plus simple et performant d'intégrer les fonctions nécessaires à une application dans un seul protocole.

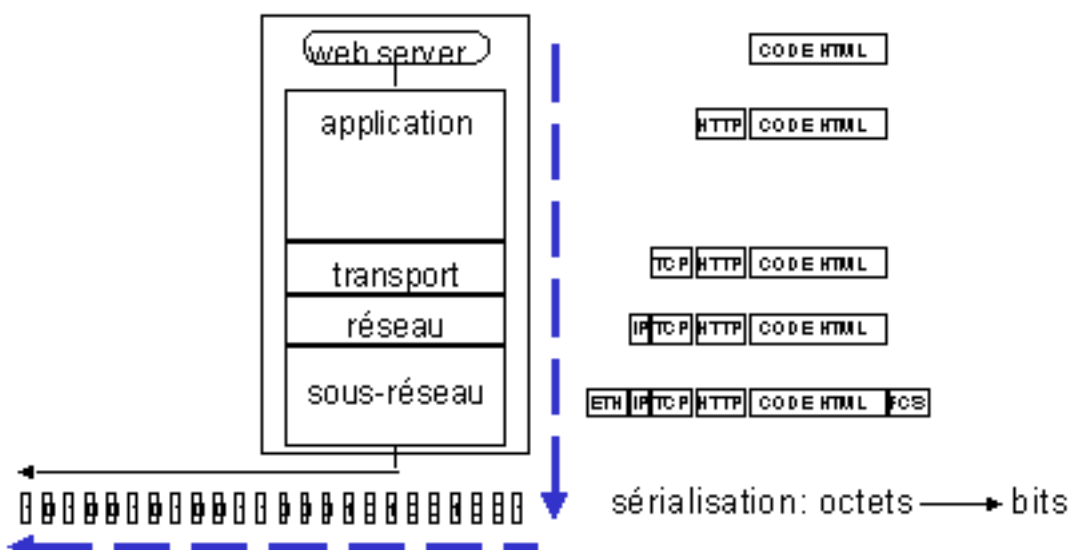
## Exemple transmission page HTML



### Exemples

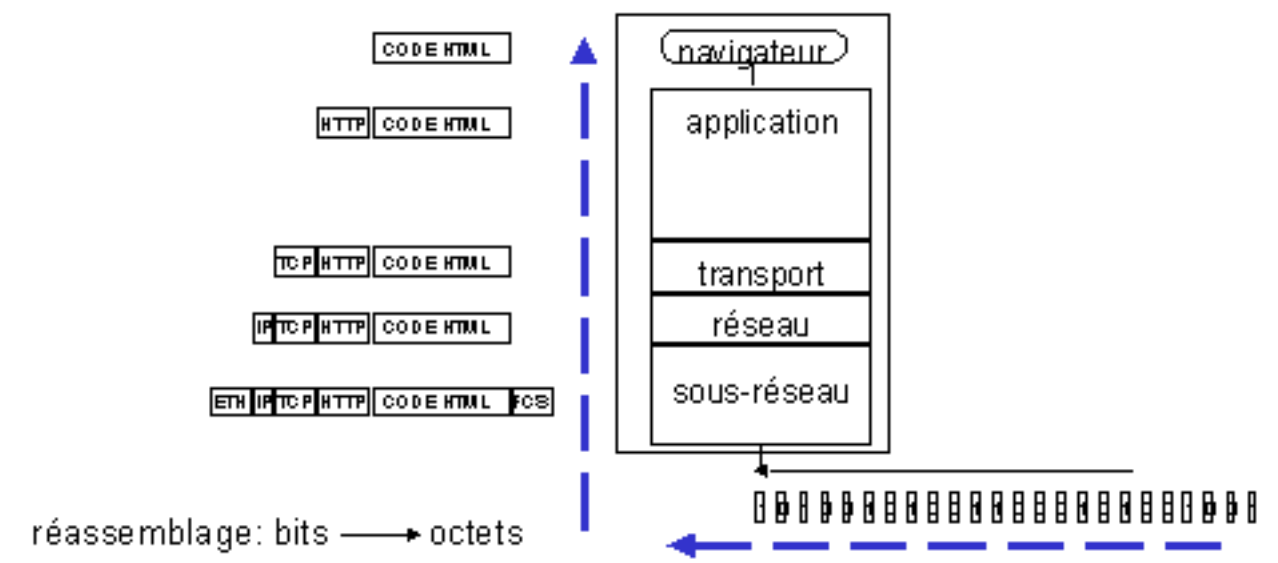
Prenons l'exemple de transmission d'une page HTML d'un serveur Web à un navigateur.

## Exemple transmission page HTML



Pour que le code HTML de la page puisse être transmis au navigateur qui en a fait la demande, l'application serveur Web fait appel au protocole HyperText Transfer Protocol qui ajoute ses propres informations destinées à son homologue, le protocole HTTP de l'ordinateur distant. Le logiciel de la couche application fait ensuite appel aux services de la couche transport pour obtenir un service de transmission fiable à destination de l'ordinateur distant. Le protocole TCP est choisi; celui-ci ajoute des informations à destination de son homologue. Les logiciels de la couche transport font appel aux services de la couche réseau pour acheminer les données. Le protocole IP est imposé par le réseau Internet. Les logiciels de la couche réseau n'ont plus qu'à choisir le lien et les moyens physiques pour envoyer les données. Les logiciels de la couche physique sérialisent les octets de la trame constituée du code HTML de la page à transmettre plus toutes les informations ajoutées par les différents protocoles utilisés.

## Exemple réception page HTML



Les éléments binaires reçus par l'ordinateur de destination sont réassemblés en octets pour reconstituer la trame avant de la passer au logiciel de la couche sous-réseau qui utilise le protocole Ethernet pour vérifier si la trame est valide. Si oui la couche sous-réseau peut passer les données à la couche réseau qui utilise les informations du protocole IP pour vérifier la validité du paquet reçu et savoir à quel protocole de la couche supérieure il faut passer les données et ainsi de suite jusqu'au logiciel du navigateur. En cas de problème le logiciel de la couche qui a détecté le problème essaie d'en informer son homologue et éventuellement les logiciel de la couche supérieure.

Cette description est simple et sommaire. Pour plus de précisions il faut étudier en détail l'opération de chaque protocole.

---

## Architectures propriétaires

Les réseaux d'ordinateurs se sont développés à partir des années 1970s. Pendant vingt ans les architectures propres à chaque constructeur ont dominé le paysage des réseaux. Les plus importantes de ces architectures sont:

1. SNA Systems Network Architecture de IBM
2. DSA Distributed Systems Architecture de Bull
3. DNA/DECNET DEC (Digital Equipment Corporation) Network Architecture

Dans le domaine des réseaux locaux d'entreprise l'architecture de Novell a dominé pendant une dizaine d'années.

Avec l'arrivée d'Internet en début des années 1990s (bien que sa conception remonte à l'année 1969 dans les laboratoires des universités américaines) toutes les architectures se sont ouvertes sur Internet soit en intégrant ses protocoles, soit en créant au moins des passerelles qui permettent aux données de passer d'un réseau qui utilise une architecture propriétaire aux réseaux qui utilisent l'architecture Internet.

---

auteur : Edoardo Berera - Miage Nice

[EB](#) date de dernière modification : 22 mai 2002

# Bibliographie

---

Simon Nora et Alain Minc, "La télématique. Rapport...", editions..., 197x

Guy Pujolle, "Les Réseaux", Eyrolles

Andrew Tanenbaum, "Computer Networks", Prentice Hall

Z 70-001, Norme expérimentale, Systèmes de traitement de l'information, Modèle le référence de base pour l'interconnexion de systèmes ouverts, AFNOR, 1982

---

**date de dernière modification : 25-08-02**

# Exercices

---

## Sous Linux

1. Installer un modem
2. Installer une carte de réseau Ethernet
3. Configurer un accès à un Fournisseur d'Accès Internet

## Sous Windows

1. Installer un modem
2. Installer une carte de réseau Ethernet
3. Configurer un accès à un Fournisseur d'Accès Internet

## Sous Linux et Windows

1. Vérifier la configuration réseau de votre machine (ipconfig)
  2. Tester la connectivité avec un autre ordinateur (ping)
  3. Visualiser la route entre votre ordinateur et celui avec lequel vous venez de faire un test de connectivité (tracroute, tracert)
-

# Glossaire

---

**AFNOR**

Association Française de Normalisation

**DNA/DECNET**

DEC (Digital Equipment Corporation) Network Architecture

**DSA**

Distributed Systems Architecture (Bull)

**Full duplex****FTP**

File Transfer Protocol

**Half duplex****HTTP**

HyperText Transfer Protocol

**IN**

Intelligent Network

**IP**

Internet Protocol

**ISO**

International Standards Organisation

**ITU- T**

International Telecommunication Union, Telecom sector

**LAN**

Local Area Network

**MAN**

Metropolitan Area Network

**OSI**

Open Systems Interconnection

**OSS**

Operations Support Systems

**PAN**

Personal Area Network

**RSTP**

Realtime Streaming Transport Protocol

**RTP**

Realtime Transport Protocol

**SCP**

Service Control Point

**SMTP**

Simple Mail Transfer Protocol

**SNA**

Systems Network Architecture (IBM)

**SNMP**

Simple Network Management Protocol

**SS7**

Signaling System #7

**SSP**

Switching Service Point

**STP**

Switching Transfer Point

<b>TCP</b>	Transmission Control Protocol
<b>TMN</b>	Telecommunications Management Network
<b>UDP</b>	User Datagram Protocol
<b>UIT</b>	Union Internationale des Télécommunications
<b>WAN</b>	Wide Area Network

---



Ministère de l'Enseignement Supérieur et des recherches scientifiques  
Université Virtuelle de Tunis

Intitulé du chapitre :

**Technologies des réseaux de communication**

Nom de l'auteur :

**Gérard-Michel Cochard & Edoardo Berera  
& Michel Besson Thierry Jeandel**

Cette ressource est la propriété exclusive de l'UVT. Il est strictement interdit de la reproduire à des fins commerciales. Seul le téléchargement ou impression pour un usage personnel (1 copie par utilisateur) est permis.

# Le niveau physique : mécanismes et protocoles

Sommaire :

[Principes de fonctionnement de cette couche](#)

[Sérialisation, codages et décodages en numérique](#)

[Transmission](#)

[Transmission dans les réseaux locaux](#)

[Transmission numérique et analogique](#)

[La modulation du signal](#)

[Codages et modulation](#)

[Equipements actifs et passifs de niveau liaison et physique](#)

[Composantes de la couche physique](#)

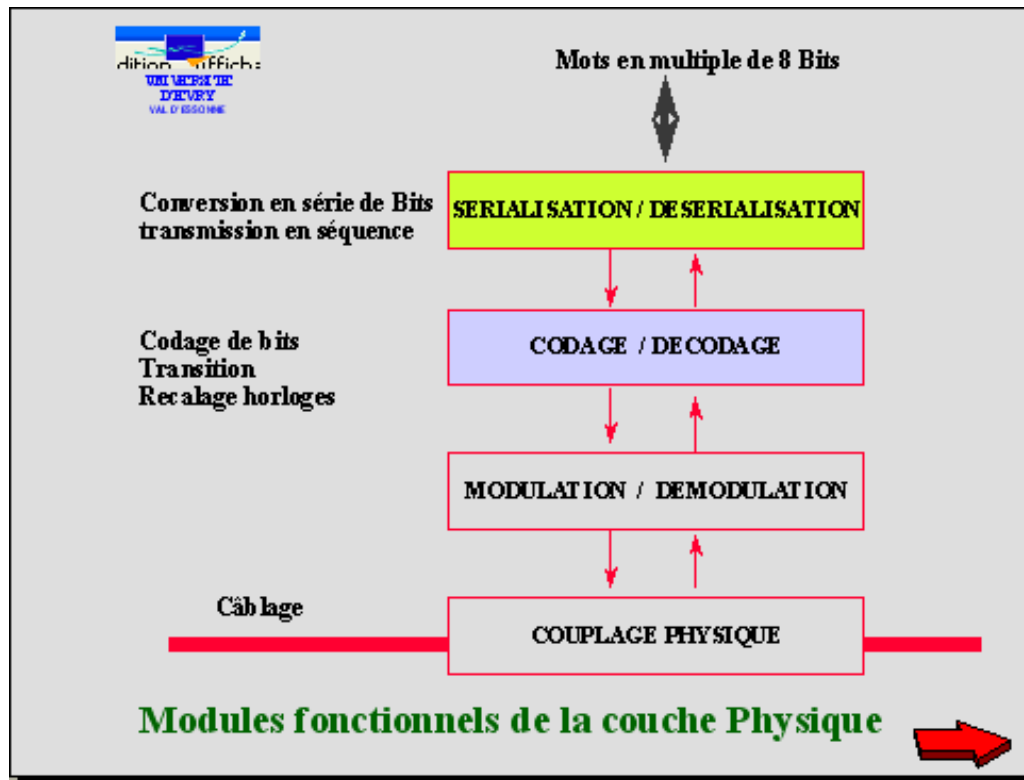
[Interactions et primitives](#)

## Principes de fonctionnement de cette couche

Le niveau Physique établit la manière dont sont transportés, sur un support physique, les bits composant le message transmis entre un émetteur et un destinataire. Ce message, avant d'être modulé sur le support, doit être initialement codé sous la forme d'une suite de 0 et de 1.

## Sérialisation, codages et décodages

[Sérialisation et détection des erreurs](#) - [Codage de caractère](#) - [Codes souvent utilisés](#)



## Sérialisation et détection des erreurs

La conversion de mots de multiples de 8 bits, en séries de bits transmis séquentiellement et inversement est appelée : **sérialisation**.

## Codage de caractère

Tout échange de message nécessite un langage ou du code qui doit être compris par l'émetteur ou le récepteur

Les deux codes les plus souvent utilisés sont ASCII et EBCDIC.

- **ASCII** ( American Standard Code for Information Exchange) publié par l'ANSI ANSI 3.4

Code à 7 bits (spécifie les 7 premiers bits d'un caractère à huit bits ) utilise tous les codages de 0000000 à 1111111 soit 128 valeurs, le 8e Bit permet d'avoir 128 caractères supplémentaires ou une parité, la parité peut être paire ou impaire.

- **EBCDIC**

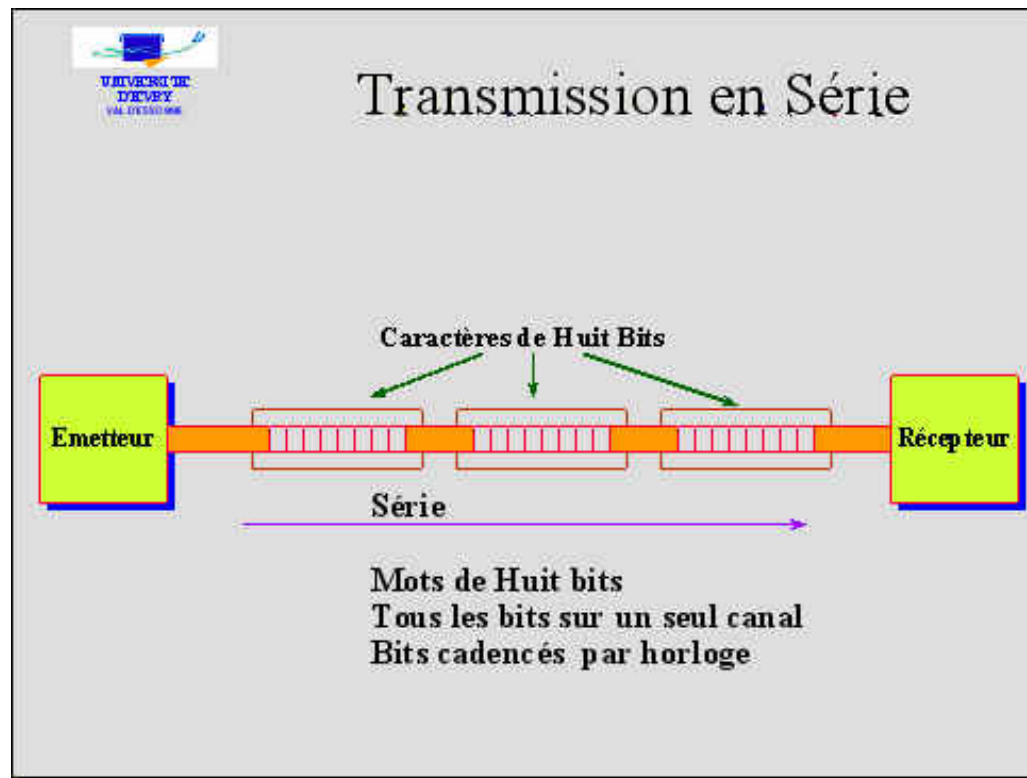
Adopté par IBM, Dérivé du BCD ( Binary Decimal Code ), il représente 256 caractères par la combinaison des 8 bits ; pas de possibilité de parité. IBM utilise le checksum pour détecter les erreurs.

## Transmission

[Transmission série/parallèle](#) - [Les capacités : Bits, Baud, Hertz et Débit binaire](#) - [Modes et caractéristiques appliquées aux transmissions](#)

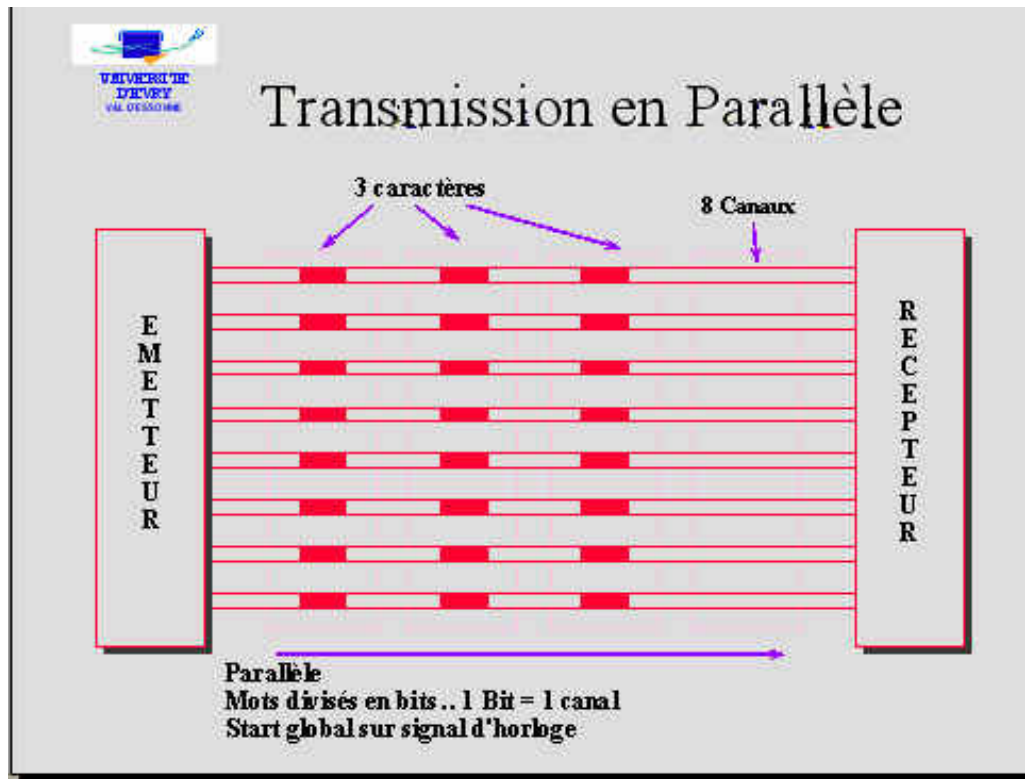
### Transmission série/parallèle

La transmission est l'étape qui suit le codage, deux modes sont possibles lors de ce déroulement d'échange : mode parallèle ou mode série



#### Série

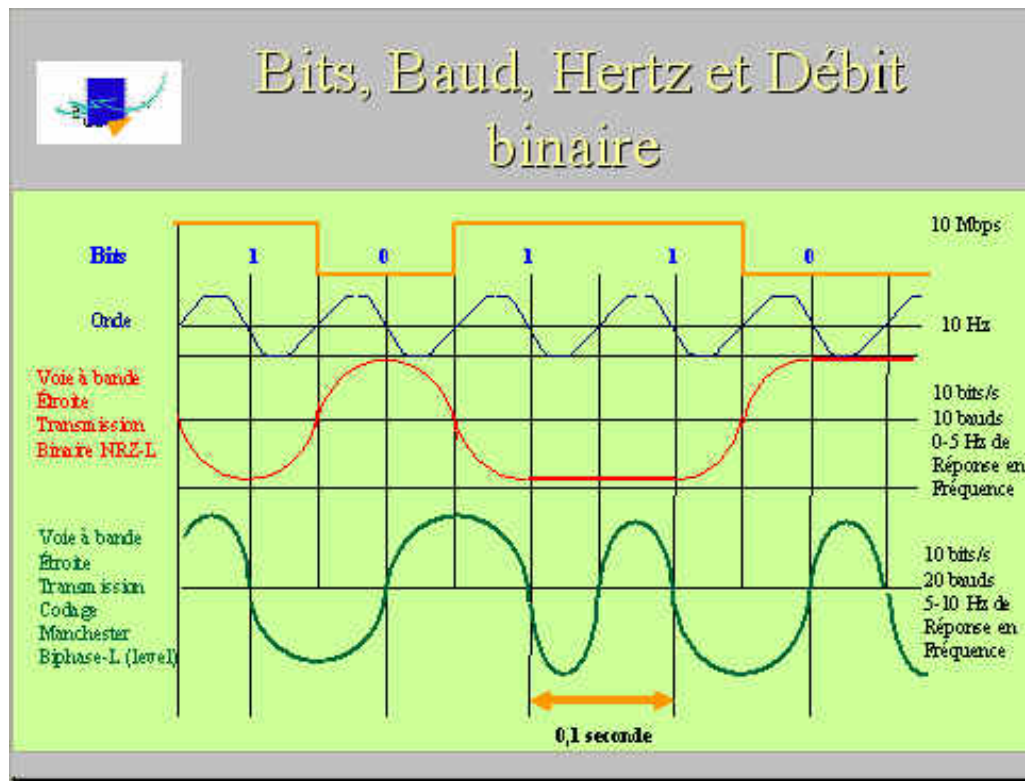
Dans ce cas les bits se suivent les uns après les autres, cependant il y aura deux manières de les faire transiter : soit par le mode synchrone, soit par le mode asynchrone.



### Parallèle

Dans ce cas chaque caractère est envoyé sur un fil et tous les caractères arrivent simultanément à leur destination.

**Les capacités : Bits, Baud, Hertz et Débit binaire**



La Rapidité de modulation est le nombre de symboles transmis.

### Débit en bauds

Le débit en bauds est le nombre de changements d'états par seconde.

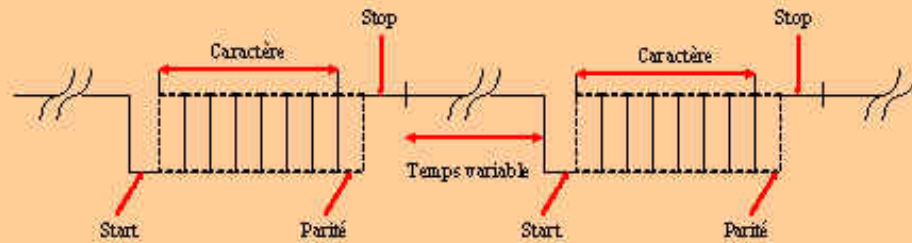
### Débit binaire

Le débit binaire est le nombre de bits écoulés par seconde.

## Modes et caractéristiques appliquées aux transmissions



# Transmission Asynchrone



**Synchro préalable inexistante**

## Mode synchrone

Il s'agit de transmettre dans un intervalle constant sur lequel l'émetteur et le récepteur se sont accordés. La répétition de cet intervalle est continue.

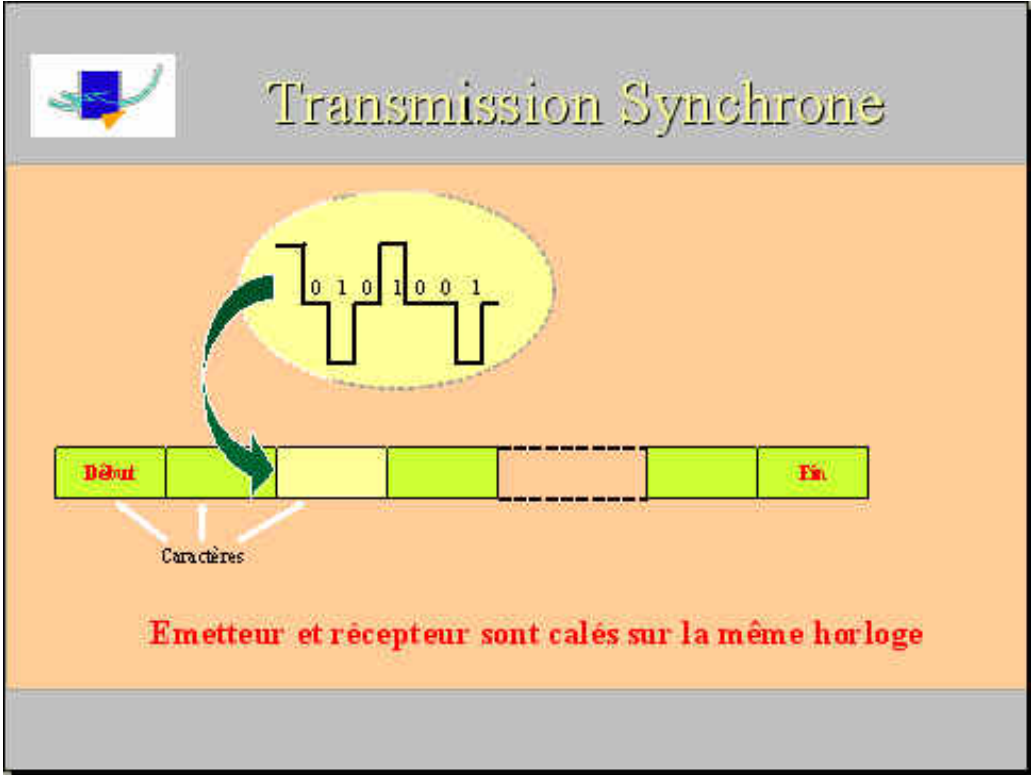
Les caractères sont envoyés de manière séquentielle, sans séparateurs. Ce mode convient aux débits importants.

Le signal est émis en étant synchronisé sur une horloge au moment de l'envoi d'un bit.

Le débit en bauds de la ligne est fonction de la cadence de l'horloge (nombre de tops d'horloge par seconde), 60 bauds équivalent à 60 intervalles de temps basiques dans une seconde.

Plusieurs types de signaux peuvent être éventuellement transmis simultanément, un signal a une valence de  $n$  si le nombre de niveaux transportés dans un intervalle de temps est de  $2^n$ .

La capacité de transmission du lien, en bits par secondes, est égal à  $n$  multiplié par la vitesse en bauds. Donc un lien à 50 bauds de valence  $n=2$  aura un débit de 100 bits par seconde.



**Mode asynchrone**

- Pas de relation établies à l'avance entre émetteur et récepteur.
- Deux signaux, les bits start et stop, encadrent les bits de chaque caractère. Une transmission débute à un instant quelconque.

**Modems**

Sur une ligne similaire voici quelques débits en fonction du niveau de performance de différents modems :

V32bis	14,4 Kbps
V34	28,8 Kbps
V 34bis	33,6 Kbps
V90	56 Kbps

**Sens de transmission de point à point**

- Liaisons unidirectionnelles :

Elles sont aussi appelées simplex, elles vont toujours dans le sens émetteur vers récepteur.>

- Liaisons bidirectionnelles :



Appelées aussi à l'alternat, semi duplex ou bien encore half-duplex ; dans ce cas l'émetteur peut devenir récepteur et inversement.

- Liaisons bidirectionnelles simultanées :

Appelées aussi duplex ou full-duplex ; la transmission est simultanée dans les deux sens de l'échange.

## Transmission dans les Réseaux Locaux

[Traitement des messages en réseau locaux](#) - [Déphasage entre horloge et signal](#) - [Contrôle d'erreurs](#)

### Traitement des messages en réseau locaux

Les informations sont transmises sur une liaison PHY en série

Chaque bit est représenté par une durée DELTA ou base de temps, l'émetteur et le récepteur reconnaissent cette base de temps avec leur horloge. Cette base delta définit la durée d'un bit

- Débit = nombre de Bits/sec émis en série sur le support

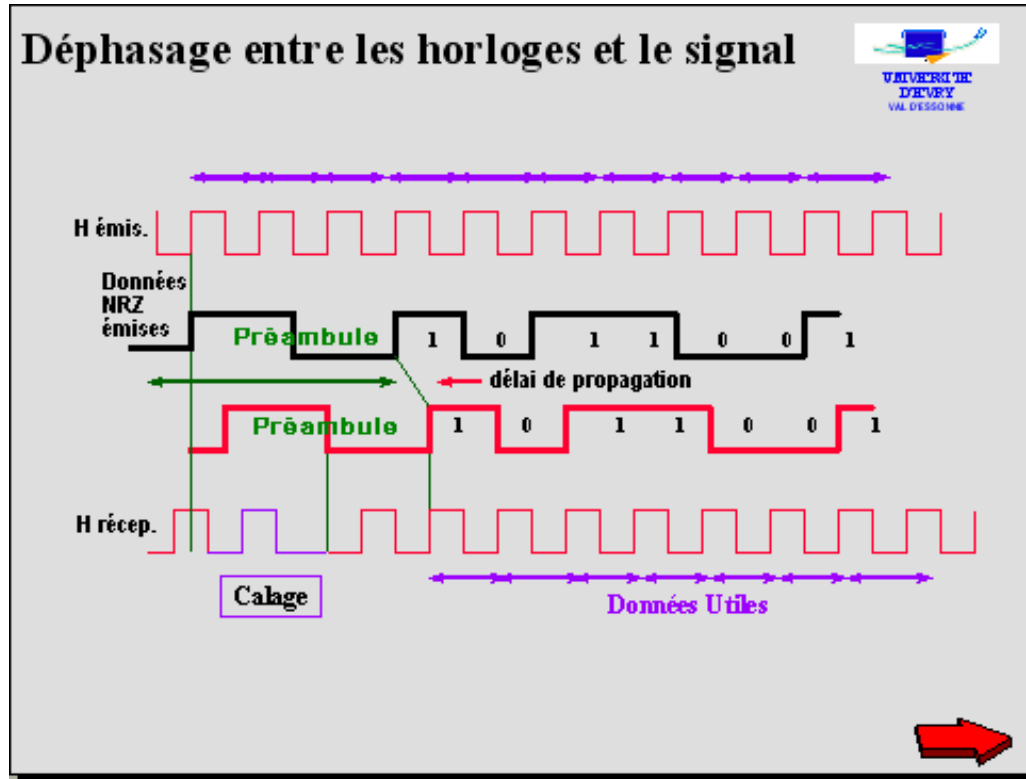
Les bits sont envoyés par BLOCS séparés en transmission asynchrone, le signal d'horloge n'est pas transmis, il y a risque de déphasage entre horloge et signal d'où un besoin de synchronisation en 2 fonctions :

- Calage
- Verrouillage

Les codages permettent de maintenir la synchronisation.

Le codage des Bits garantit une transition pour recaler l'horloge avant risque de découpe de bits en + ou en -.

## Déphasage entre horloge et signal



### Calage (Ex : CSMA CD)

L'action de calage consiste à synchroniser l'horloge sur le début de DELTA :

Entre 2 blocs le Réseau est inactif (= dérive des horloges).

Le délai de propagation est différent entre source et récepteur.

On emploiera donc un préambule ou bits start durant quelques temps bits.

## Contrôle d'erreurs

### Le bit de parité

A tout groupe de bits est ajouté un bit de résultat du ou exclusif des bits précédents.

Défaut : non détection si deux ou un nombre pair de bits sont défectueux, car non localisation de l'erreur ou non décompte d'erreurs.

### Le CRC

Ajout d'une séquence de longueur constante à la fin de trame.

Cette séquence est élaborée à partir d'un polynôme générateur et des données transmises, la même séquence est exécutée à l'arrivée, si le résultat est négatif la trame est mise au rebut.

## Transmission Numérique et Analogique

[Généralités](#) - [Affaiblissement du Signal](#) - [Affaiblissement du RTC](#) - [Puissance du signal](#) - [Perte de signal](#) - [Qualité d'un signal Analogique](#) - [Lignes Numériques](#) - [Avantage des Lignes Numériques](#) - [Analogique / Numérique](#) - [Échelle dynamique de la voix](#) - [Companding](#) - [Codecs](#) - [Codage de la voix et Modulation](#) - [Numérisation de signaux vidéo](#)

### Généralités

Lorsqu'un signal est transmis des bruits externes peuvent venir affecter la transmission. Connaître le niveau du bruit permet de calculer la capacité maximum de la ligne en bits/s.

Les interfaces, les ondes électromagnétiques et le support lui-même peuvent participer à ce bruit. Le rapport signal / bruit est l'une des caractéristiques majeures pour estimer la capacité d'un canal.

Si l'on procède à une estimation de ce signal/bruit durant un intervalle de temps, on pourra exprimer sa valeur en décibels (dB).

Limites théoriques du débit binaire d'un canal soumis à un bruit

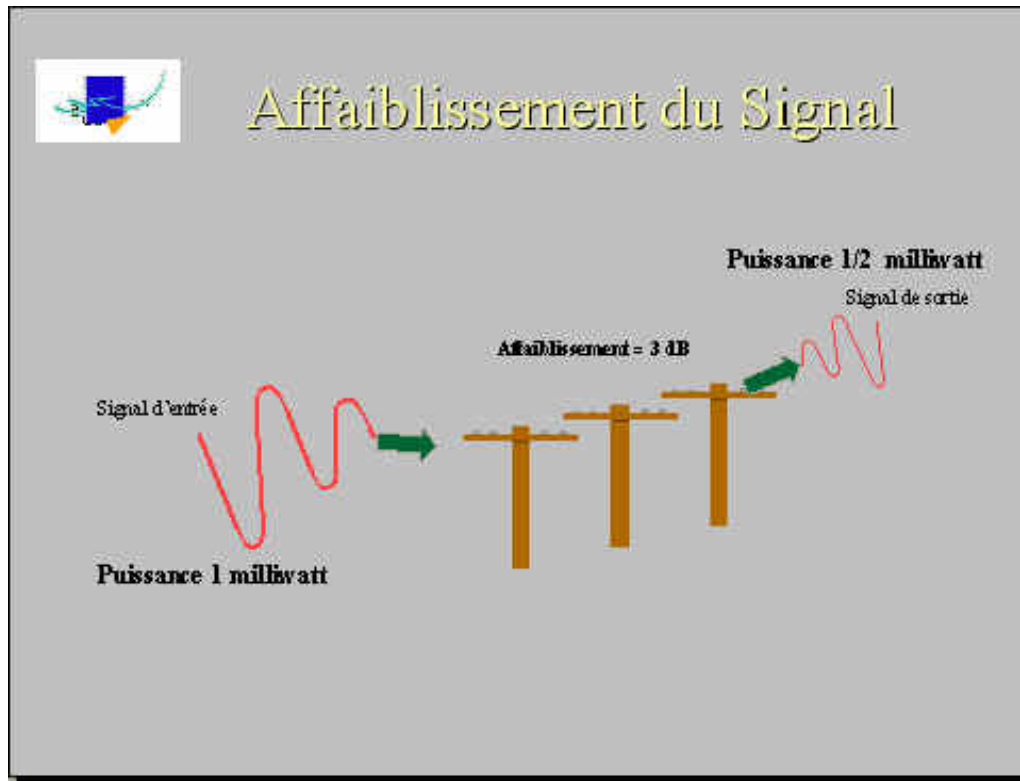
- La loi de Shannon permet de l'énoncer ainsi :

$$\text{Capacité (bits/s)} = \text{Bande Passante (Hz)} \cdot \log_2 (1 + \text{Signal / Bruit})$$

- Transmission via modems sur RTC (à faible performance)

BP :	3 kHz
Puissance du Signal :	-20 dBm = 10μW
Puissance du Bruit :	- 50 dBm = 10nW
Capacité (bits/s):	3000 Log <sub>2</sub> (1+1000)=30Kbps

### Affaiblissement du Signal



### Perte dans les canaux logiques

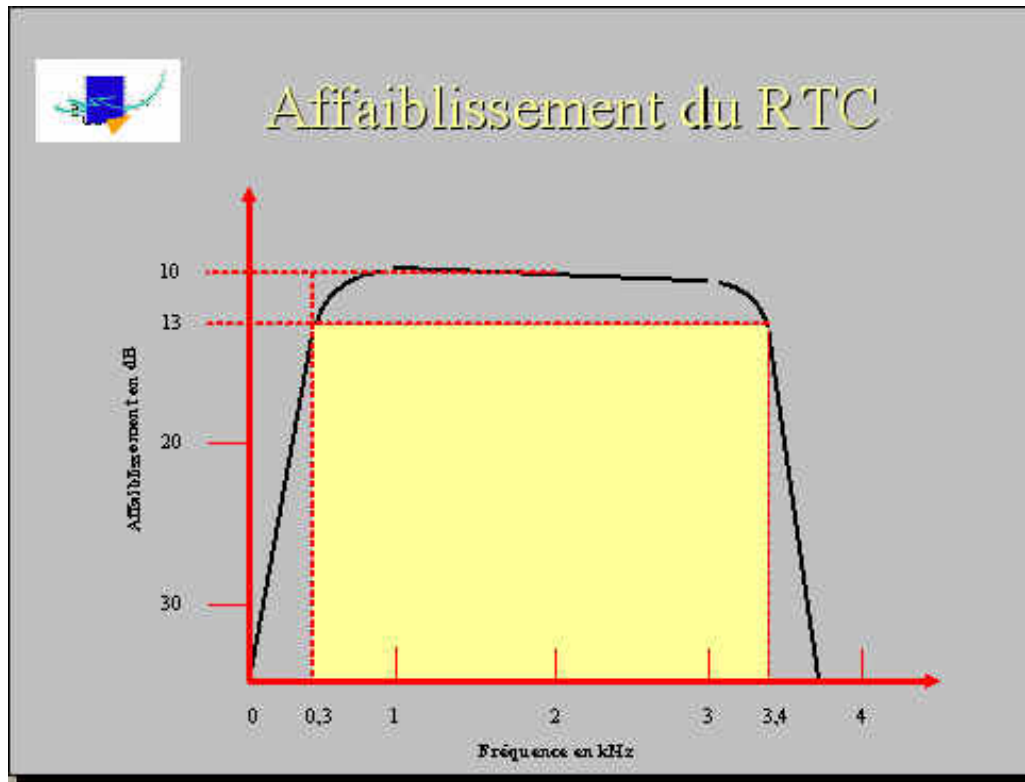
Les signaux traversant une ligne téléphonique sont affaiblis.

### Affaiblissement

Il est mesuré en décibel (dB)

$\text{dB} = 10 \log_{10} (\text{puissance d'entrée} / \text{puissance de sortie})$ .

### Affaiblissement du RTC



### Réponse en fréquence pour une ligne analogique

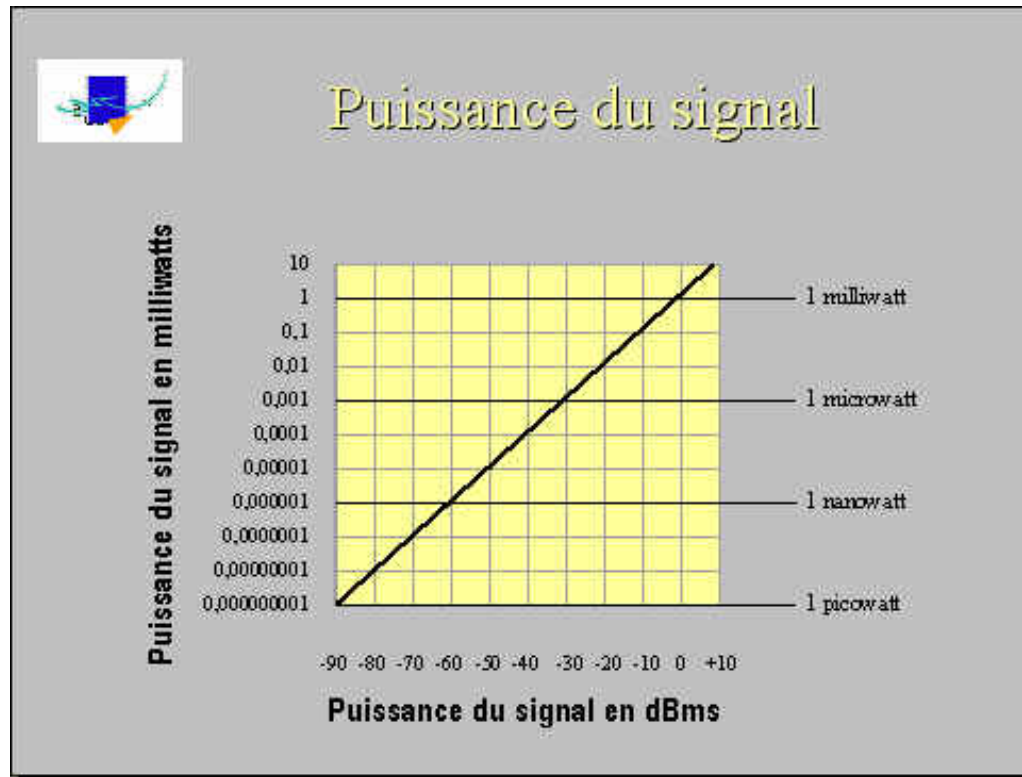
Bande passante de 0.3 à 3.4 kHz

$$BP = 3,1 \text{ kHz}$$

### Filtre numérique

La perte de signaux supérieures au 3,4 kHz est due au filtre numérique.  
L'énergie nécessaire à l'échantillonnage au dessus de 4 kHz est négligeable.  
Les basses fréquences sont filtrées pour élimination de parasites.

### Puissance du signal



La puissance est généralement référencée à un milliwatt (son faible pour un téléphone)

### L'unité

L'unité est le décibel référencé à un milliwatt (dBm)

### Calcul

La puissance en dBm =  $\log_{10}(\text{puissance}(\text{milliwatt}) / 1\text{milliwatt})$

## Perte de signal

### Perte dans les communications

Elle est caractérisée par la différence de puissance entre signal émis et signal reçu.

### Perte dans un guide d'onde (coax, TP, fibre)

Elle est liée aux propriétés du média.

### Perte en Sans Fil

Elle est due à la liaison en air libre.

### Exemple :

Sur un câble coaxial RG 58 de 30m de long, la perte est de 12 dB à 900 MHz.

Sur une distance et fréquence identiques en air libre la perte est de 61 dB.

## Qualité d'un signal Analogique

### Principale mesure

Première mesure de qualité : le rapport Signal/Bruit, mesure exprimée en dB.

$S/B = \log_{10}$  (puissance du signal/puissance du bruit).

### Valeurs typiques de S/B

Conversation téléphonique : 35 dB environ.

Signal vidéo TV : 45 dB environ.

Compact disque audio : 92 dB environ.

## Lignes Numériques

### Caractéristique

La principale caractéristique de ces lignes est le débit binaire (en bits/s).

### Mesure de qualité principale

Le taux d'erreur sur les bits BER (Bit Error Ratio ou Rate).

BER= Nombre de bits erronés / nombre de bits reçus.

### Valeurs classiques de BER

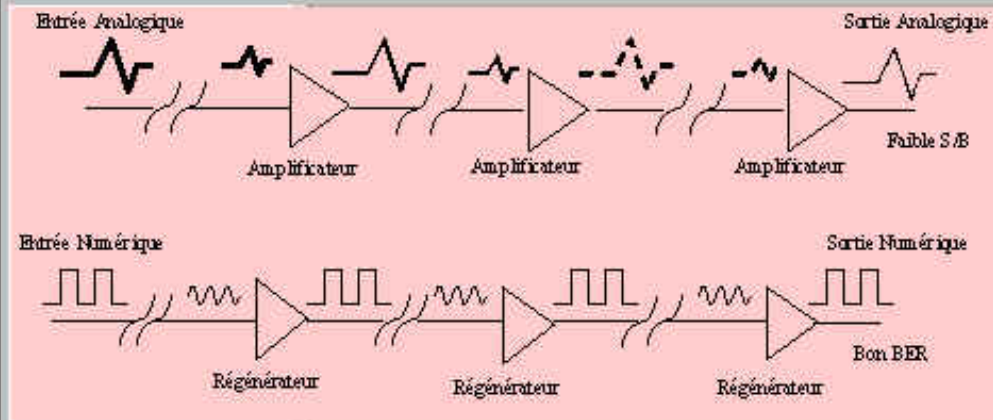
Ligne téléphonique avec modem:  $10^{-5}$  (1 pour 100 000)

Ligne fibre optique:  $10^{-12}$  (1 pour 1 000 000 000 000).

## Avantage des Lignes Numériques



# Lignes Numériques



## En Analogique

Le bruit s'accumule à chaque étage d'amplification, donc on constate un mauvais rapport S/B sur la distance.

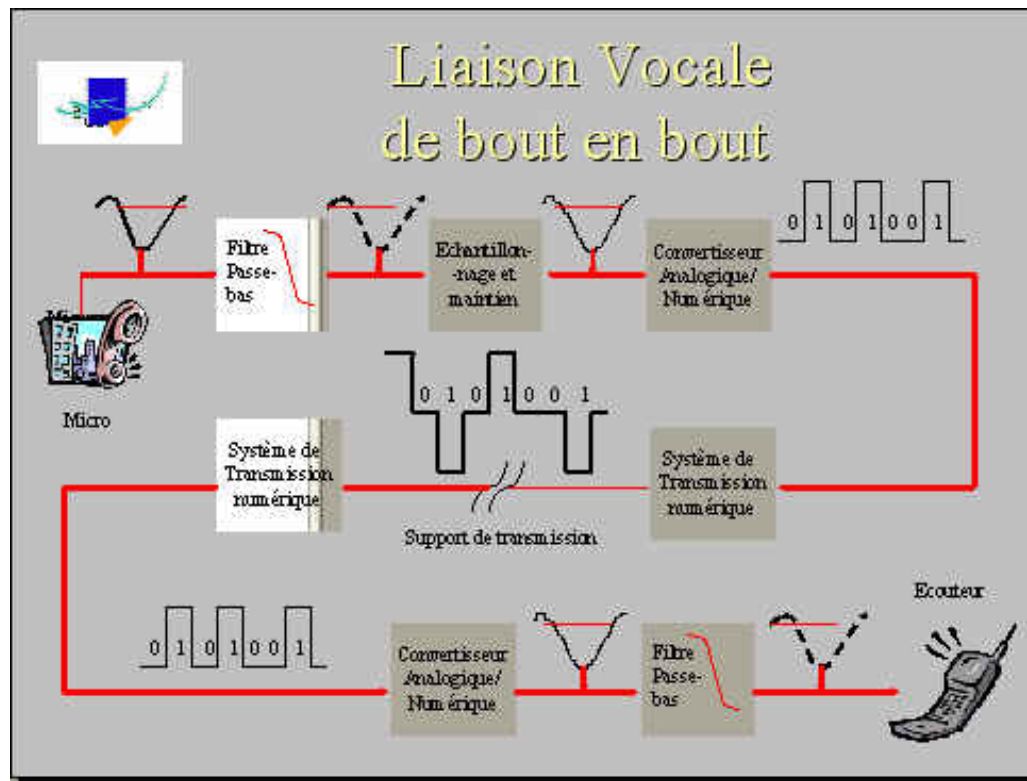
## En Numérique

Les erreurs aléatoires des répéteurs ont peu d'effet sur le BER.

La commutation des signaux numériques est simple (BER variant faiblement).

**Analogique / Numérique**





### Principes

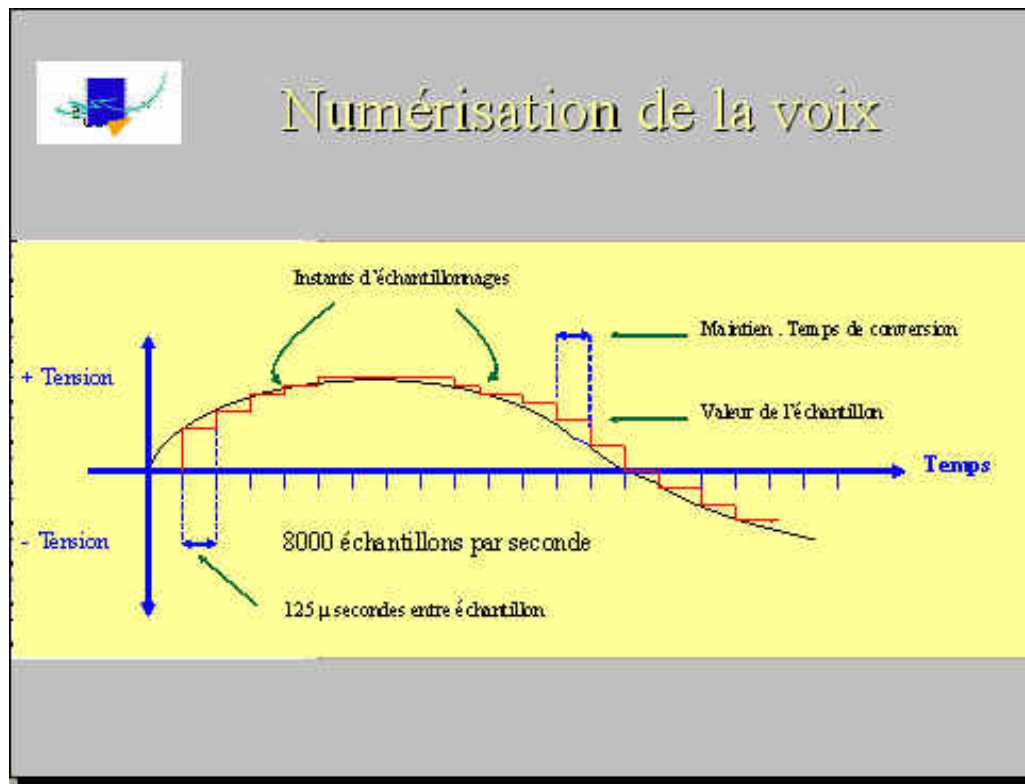
Les signaux analogiques sont prédisposés aux bruits.

Les canaux de transmission sans fil sont généralement bruités.

Les signaux numériques sont immunisés (bruit) à condition que le S/B soit au-dessus d'un certain seuil.

### Numérisation de la voix

Le courant modulé de conversation peut inclure une énergie significative pour



toute fréquence au-dessus de 4 kHz.

Les filtres bas limitent la Bande Passante à 4 kHz.

### Échantillonnage

Le circuit d'échantillonnage et de maintien prend une valeur du signal analogique à un instant  $t$  et maintient cette valeur jusqu'à  $t+1$ .

### Quantification

On représentera un échantillon par une valeur numérique au moyen d'une loi de correspondance.

### Conversion (Codage)

Le convertisseur analogique - numérique change les valeurs analogiques des échantillons en une suite de nombres numériques (codage : 8 bits par échantillon).

### Système de transmission numérique

Il positionne les échantillons (codés sur 8 bits) en une transmission série sur la ligne  
Le récepteur retrouve les échantillons et les envoie au convertisseur numérique /analogique.

### Convertisseur Analogique / Numérique

Il change les 8 bits en une simple tension et maintient cette valeur jusqu'au prochain échantillon.

## Filtre passe-bas

La courbe en escalier restituée traverse un filtre passe-bas pour être lissée et être conforme au signal d'entrée. Le filtre de sortie pilote l'écouteur.

## Mode de transmission

Ce mode est appelé MIC Modulation par Impulsions Codées.

## Échelle dynamique de la voix

### Plage acoustique de l'oreille humaine

Elle est sensible à une plage importante, la puissance minimum détectable à puissance maximum utilisable va de  $10^{12}$  à 1 (120 dB).

### Variation de puissance

Il est difficile de discerner un doublement de puissance (3 dB).

### Convertisseurs A/N ou N/A

Les convertisseurs A/N ou N/A se servent de ces particularités de l'écoute humaine pour les communications téléphoniques vocales pour les transmettre à faible débit, mais avec qualité

8 bits/échantillons X 8000 échantillons/s = 64 000 bits/s.

Le débit de 64 Kbps est généralisé.

Les deux convertisseurs (A/N et N/A) sont différents et incompatibles.

## Bruit de quantification

La différence entre le signal initial et son approximation numérique s'appelle le **bruit de quantification**. Un convertisseur 4 bits (16 niveaux) a un bruit de quantification élevé.

## Companding

### Echelle non Linéaire

Elle permet d'obtenir plus de pas pour les signaux faibles et moins pour les signaux forts.

### Quantificateur non Linéaire

Le quantificateur non linéaire compresse les 12 bits en une équivalence A/N de 8 bits et le convertisseur N/A expande les 8 bits reçus à une équivalence de 8 bits.

## Codecs

Le Codec convertit la voix analogique en codage numérique.

Exemple d'algorithmes : la loi PCM en Europe (Pulse Coded Modulation) et la loi  $\mu$ PCM (Amérique du Nord).

### Loi A et $\mu$

Pour obtenir une correspondance entre la valeur de l'échantillon et le nombre le représentant, on utilisera deux lois :

- Loi A en Europe
- Loi  $\mu$  en Amérique du Nord

Ces lois sont semi-logarithmiques, la précision étant garantie de manière pratiquement constante.

Ils utilisent instantanément la compression extension, ils ont un S/B constant :

- Grands pas pour les signaux forts
- Petits pas pour les signaux faibles

Ils travaillent sur 8 bits par échantillon. Performance aussi satisfaisante et équivalente pour les signaux faibles qu'une conversion linéaire sur 12 bits (A) ou 13 bits ( $\mu$ ),

On obtient un équilibre entre le S/B et la bande dynamique. Les Codecs sont spécifiés par le G711 de ITU-T.

## Codage de la voix et Modulation

### Codage classique sur RNIS (Réseau Numérique à Intégration de Service)

Le codage classique sur RNIS est PCM (Pulse Coded Modulation)

Le débit classique de 64 Kbps est trop élevé pour les liens radio. Ce signal bien que simple contient de multiples redondances.

### Codage en GSM

Le codage GSM est le RPE-LPC (Regular Pulse Exited - Linear Predictive Coder avec un Long Term Predictor Loop).

L'information des échantillons précédents, qui ne change pas rapidement est utilisée pour prédire l'échantillon courant.

La voix est divisée en échantillons de 20 ms, chacun encodé sur 260 bits pour un débit résultant de 13 Kbps. Ce qui correspond au débit plein.

Sur cette base, après tests, on constate que plusieurs bits du bloc sont plus important pour la perception de la qualité de la voix; trois classes sont proposées:

- Class Ia 50 bits très sensible aux erreurs de bits
- Class Ib 132 bits moyennement sensible
- Class II 78 bits la moins sensible

La classe **Ia** dispose d'un **CRC de 3 bits** ajoutés pour la détection des erreurs, ils permettent le rejet de la trame si elle est jugée trop endommagée pour être compréhensible.

Elle est remplacée dans ce cas par une version légèrement atténuée de la précédente trame reçue.

### **Modulation en GSM**

En GSM la modulation est de type MSK avec pré filtrage gaussien. GSMK (Gaussian-filtered Minimum Shift Keying ). Le débit de modulation est de 270,83 kb/s.

## **Numérisation de signaux vidéo**

### **Fréquence d'échantillonnage**

Virtuellement les systèmes de numérisation des signaux vocaux et vidéo sont identiques. Cependant la fréquence d'échantillonnage est supérieure en vidéo.

### **Convertisseurs**

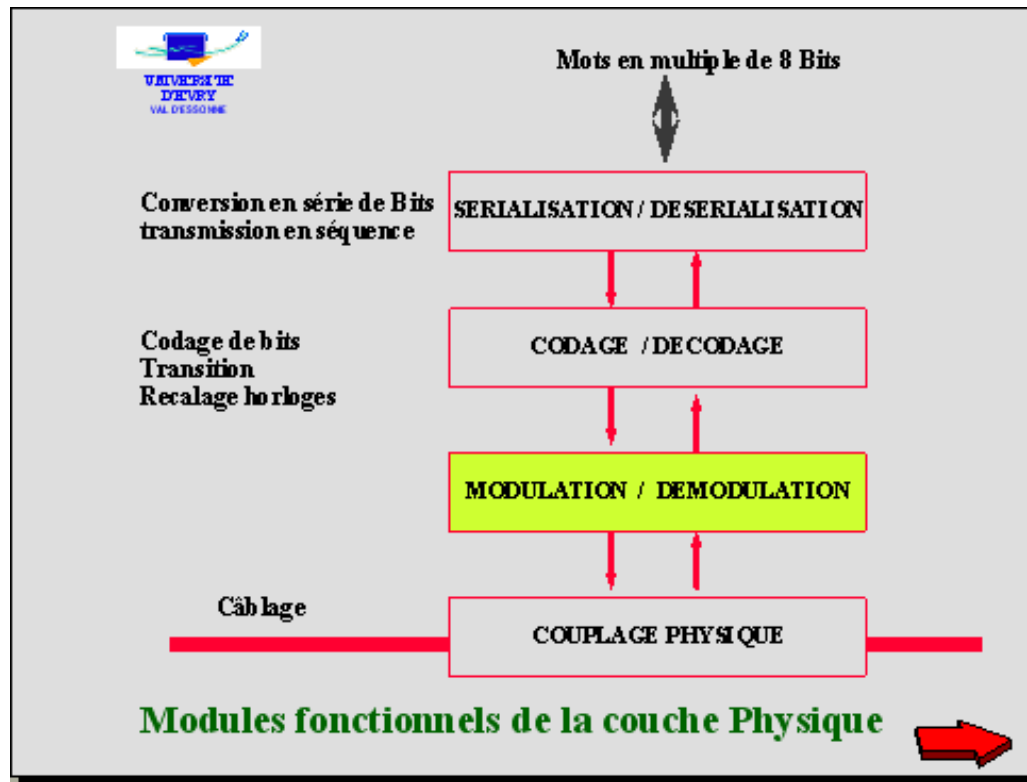
En vidéo, des convertisseurs linéaires uniformes sur huit bits sont utilisés. En vidéo couleur, un convertisseur huit bits est utilisé pour le rouge, un pour le vert et un pour le bleu.

### **Débits et vidéo compressée**

Les débits nécessaires (importants) ont conduit au développement des techniques de compression de la vidéo.

## La Modulation du signal

### Modulation du Signal - Types de données et signaux



### Modulation du Signal

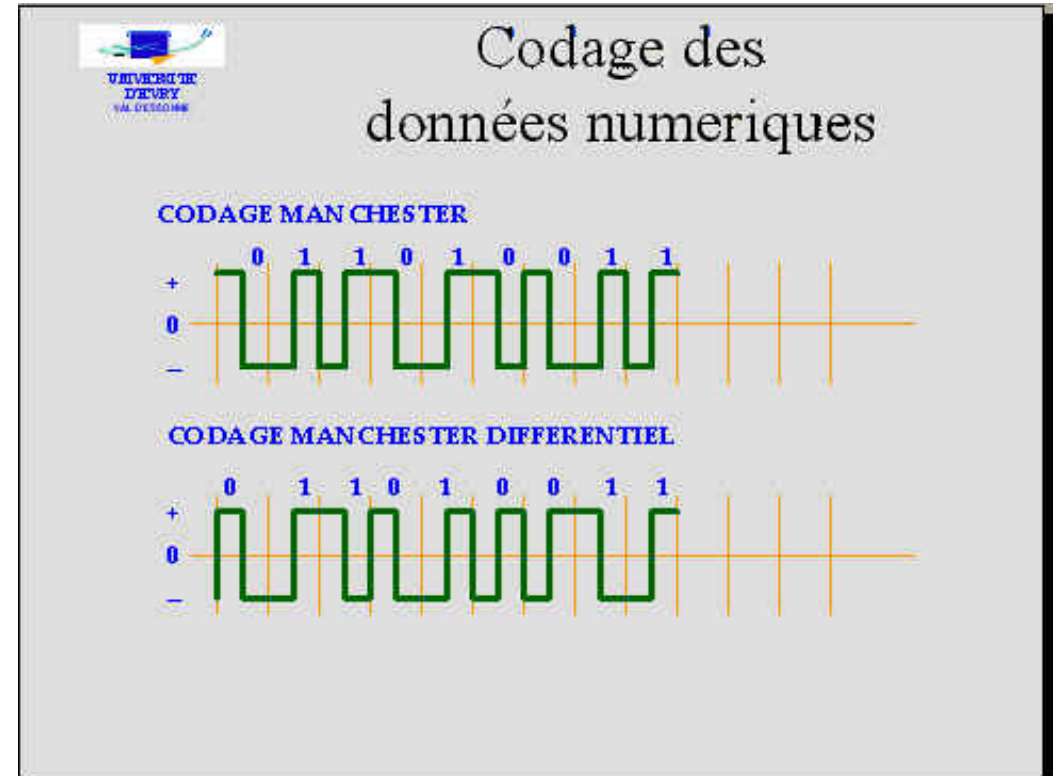
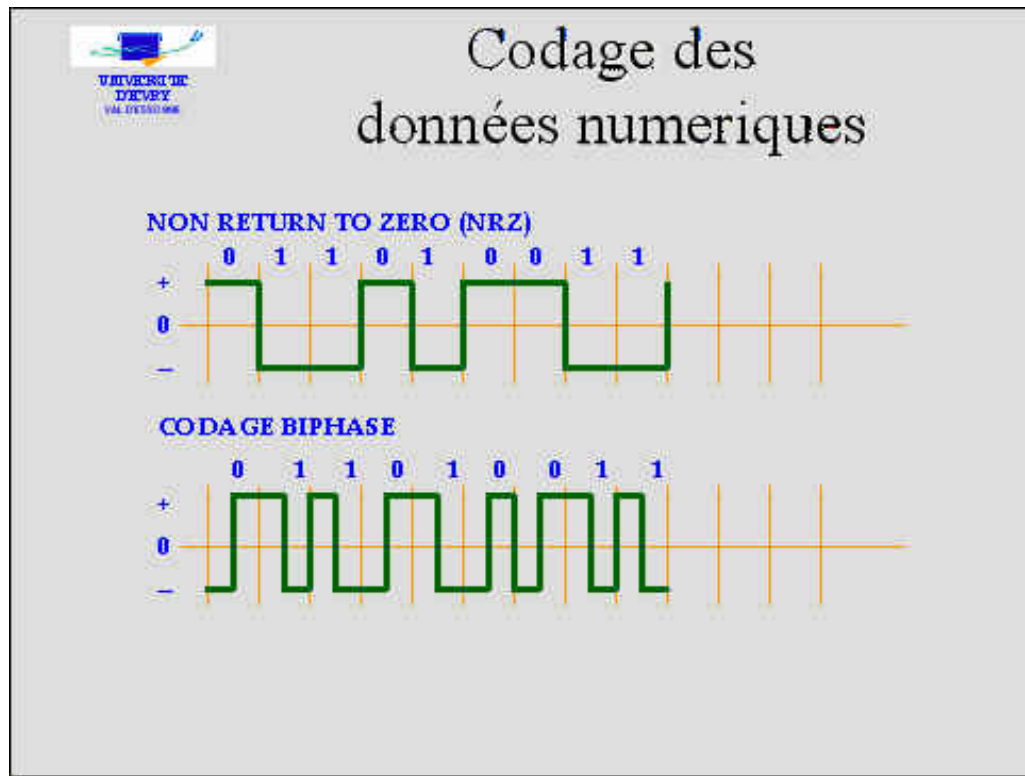
Deux termes sont couramment opposés : large bande et bande de base

#### Bande de base

L'émetteur est 1 générateur de courant, un seul signal est porté

Il est limitée par la bande passante du canal et par le rapport signal- bruit de celui-ci.

Le transport de l'information en bande base est la technique la plus simple, il n'est pas nécessaire de passer par une modulation. Des changements discret sur les signaux représentant l'information binaire vont permettre la transmission de suites binaires. Dès que la distance devient importante il est nécessaire de moduler le signal en bande de base (via un modem).



Des courants variables représentent les bits à émettre en bande de base.

### Large bande

L'émetteur est une source de fréquence basée sur trois caractéristiques, pourront varier

- AMPLITUDE,
- FRÉQUENCE,
- PHASE.

Moduler un signal (porteuse) consiste à modifier l'une des trois caractéristiques au rythme d'un autre dit «modulant», ici le signal numérique.

Le signal résultant a un spectre limité, centré autour de la porteuse, résolvant ainsi les problèmes de partage du support..

Il suffit d'attribuer à chaque communication une porteuse différente en fréquence pour les faire cohabiter sur un même support.

Le nombre maximal d'impulsions qu'un canal peut transmettre, ou sa rapidité de modulation exprimée en bauds, est égal au double de la bande passante (Nyquist).

Si elle ne peut prendre que deux valeurs 0 ou 1 : le débit est égal à la rapidité de modulation.

## Types de données et signaux

### Données

#### Données Analogiques

Elles prennent n'importe quelle valeur durant un intervalle.

#### Données Numériques

Ensembles discrets, ils ne peuvent prendre qu'un nombre réduit de valeurs.

### Signaux

#### Signaux Analogiques

Ils prennent une valeur quelconque dans leur échelle, passent lentement d'une valeur à une autre, ils sont définis par 3 paramètres:

- L'amplitude
- La fréquence
- La phase

Ils ont des avantages :

- Offrent une grande bande passante
- Supportent des réseaux étendus et complexes
- Sont peu sujet à l'atténuation sur de longues distances
- 

#### Signaux Numériques

Ils ont deux propriétés principales :

Ils ne peuvent prendre qu'un nombre limité de valeurs discrètes, parfois deux . Les transitions de valeur sont presque instantanées d'un état à un autre. Ils nécessitent une horloge et une synchronisation entre l'émetteur et le récepteur.

Les signaux analogiques ou numériques peuvent transporter des données analogiques ou numériques. Quelques exemples :



**Signaux analogiques transportant des données numériques**

Signaux de composition du téléphone.

Modems entre ordinateurs.

**Signaux analogiques transportant des données analogiques**

Stations de radio ( voix et musique sur des signaux analogiques utilisant la modulation de fréquence ou modulation d'amplitude).

**Signaux numériques transportant des données numériques**

Lignes entre un terminal et un ordinateur central.

**Signaux numériques transportant des données analogiques**

Compacts disques codant en valeur numérique l'amplitude et la fréquence du son détectés à chaque instant par des micros.

Les lecteurs de CD inversent ce processus et recréent une musique analogique.

## Codages et Modulations

### Codage de données numériques en signaux analogiques - Codage de données numériques en signaux numériques

#### Codage de données numériques en signaux analogiques

Les données numériques peuvent être transmises à l'aide de porteuses analogiques en modulant l'une des trois caractéristiques suivantes :

- Amplitude
- Fréquence
- Phase

#### Modulation d'Amplitude ASK

Elle code les données numériques en modulant l'amplitude d'une porteuse entre deux niveaux ou plus, pas de fiabilité sur les longues distances, déformation par des interférences, des atténuations et des amplifications.

#### Modulation de Fréquence - FSK

Elle code les données numériques en modulant la fréquence d'une porteuse entre deux valeurs ou plus, plus fiables sur les longues distances, peu utilisées sur les lignes téléphoniques au delà de 1200 Bauds.

### **Modulation de Phase**

Elle code les signaux numériques en décalant la phase de la porteuse d'une certaine valeur, très fiables et résistantes aux erreurs, elle contient des changements d'états qui peuvent être utilisés pour synchroniser les horloges de l'émetteur et du récepteur.

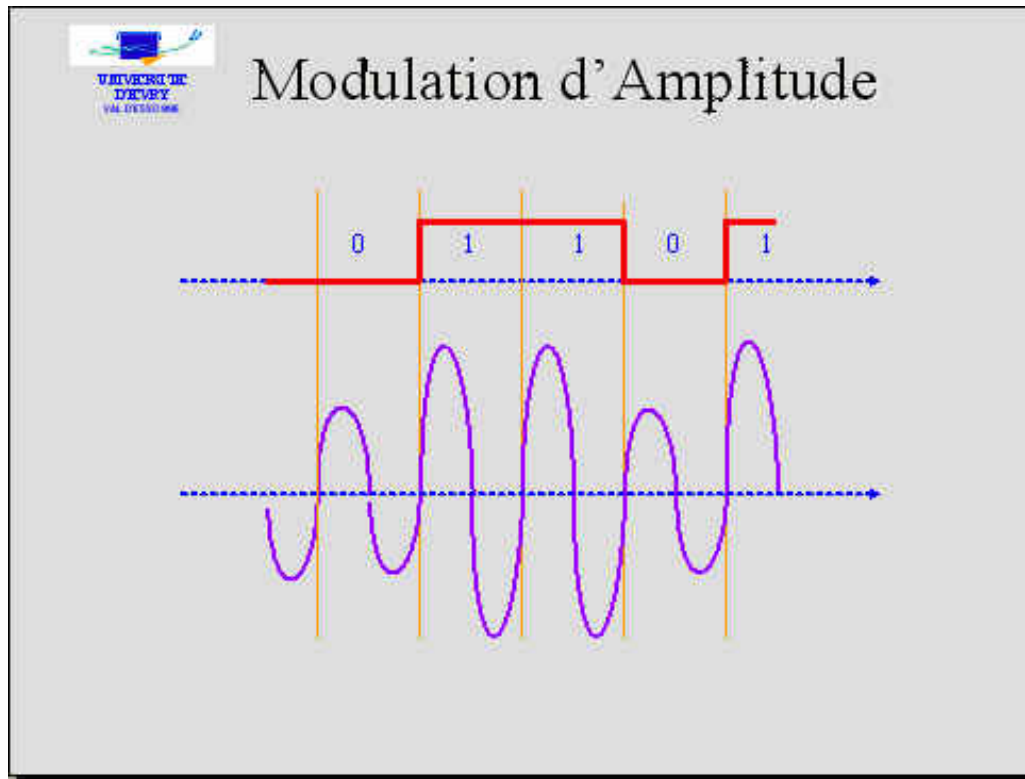
### **Codage de données numériques en signaux numériques**

La plupart des signaux transmettent des données numériques à l'aide de signaux numériques. La mesure des signaux est facilitée par des horloges qui permettent à l'émetteur et au récepteur de se mettre d'accord sur le début d'un bit.

Dans tous les systèmes électroniques, le 0 représente un niveau de référence pour les signaux, une tension nulle est considérée comme le potentiel électrique de la terre, généralement mesurée à l'aide d'un bon conducteur enfoui dans la terre.

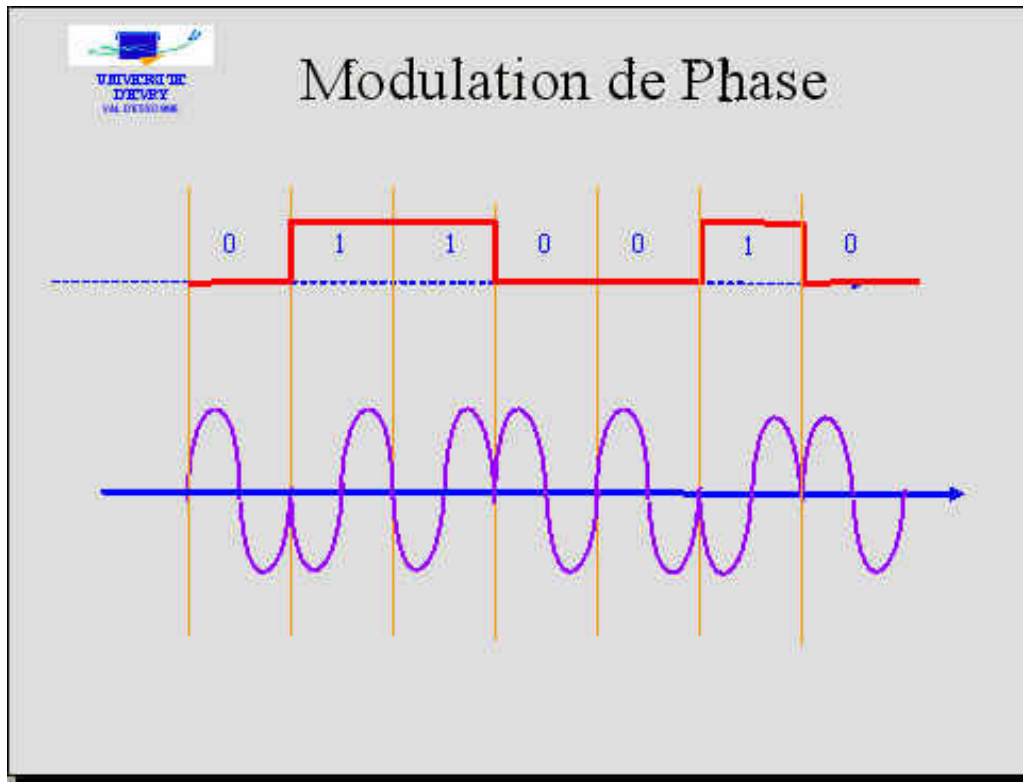
Grandeurs physiques, Modulation du signal

- Modulation d'Amplitude
- Modulation de Phase
- Modulation de Fréquence
- Modulation de Phase à 4 moments



## Modulation d'Amplitude

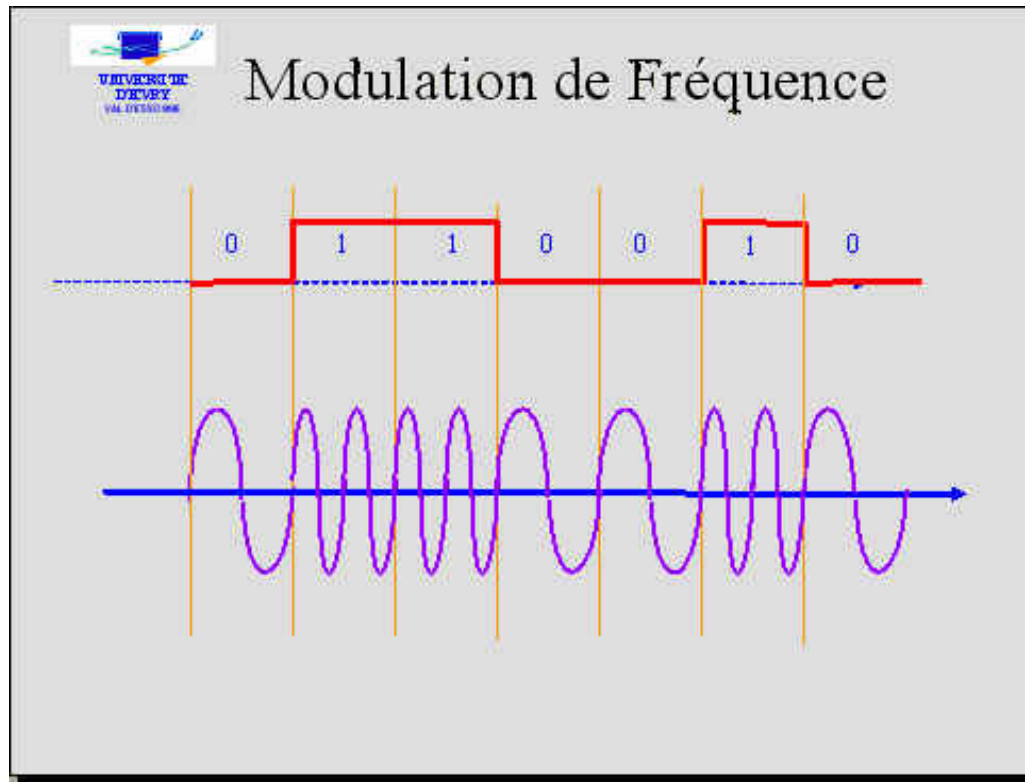
La distinction entre 0 et 1 est obtenue en faisant varier l'amplitude du signal.



## Modulation de Phase

La distinction entre 0 et 1 est obtenue par un signal qui commence à des emplacements différents de la sinusoïde (Phase).

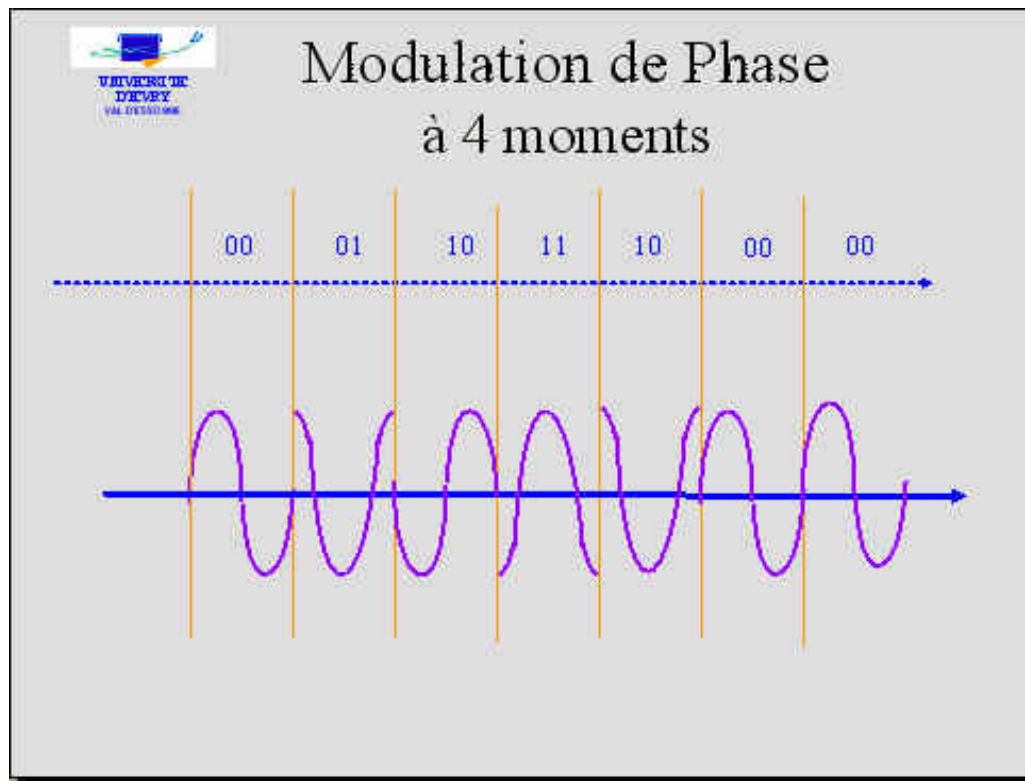
Les 0 et 1 sont représentés par des phases de  $0^\circ$  et  $180^\circ$ .



## Modulation de Fréquence

L'émetteur a la possibilité de changer la fréquence d'envoi des signaux pour distinguer entre 0 et 1.

## Modulation de phase à 4 moments



### Limité à deux états

Dans les exemples précédents la grandeur physique utilisée ne représente que deux états possibles.

Si on émet et détecte à l'arrivée plus de deux états de la même grandeur, on peut donner à chaque état une signification permettant de coder 2 ou plusieurs bits.

En utilisant 4 phases, fréquences ou amplitudes, on peut coder 2 bits à chaque état.

Fig = 2 bits par modulation de phase.

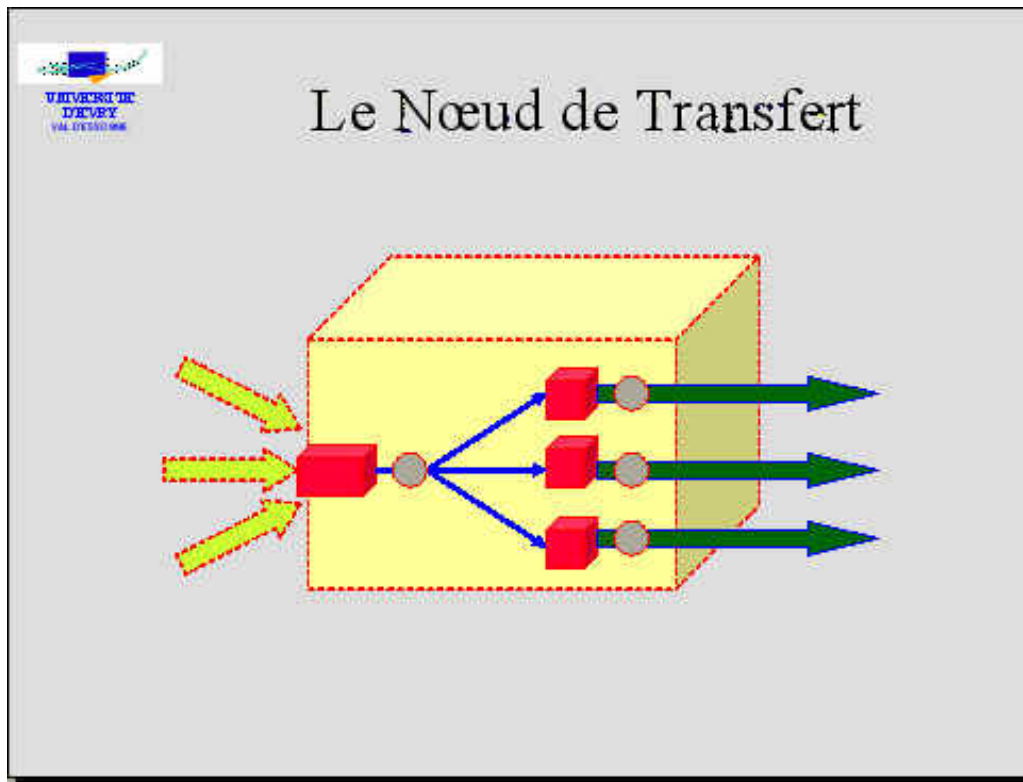
# Equipements actifs et passifs de niveau Physique et Liaison

[Les terminaux et noeuds de transfert](#) - [Multiplexages](#) - [Multiplexage temporel](#) - [Multiplexage statistique](#) - [Multiplexeur Temporel Statistique](#) - [Le multiplexage traditionnel](#) - [L'accès multiple](#) - [Interconnexion de réseau par une fibre optique](#) - [Répéteurs](#) - [Hubs et Base T](#) - [Ponts](#)

## Les terminaux et noeuds de transfert

Dans un réseau, les nœuds de transfert sont des interfaces intermédiaires ou relais vers les destinataires. Ils reçoivent des messages sur une ligne d'entrée et la retransmettent sur une ligne de sortie. Entre deux ils mémorisent le message en buffer ou tampon.

En réalité, s'ils ne traitent pas la trame directement, ils récupèrent la NPDU ; puis à partir des informations dont ils disposent ils retransmettent la trame ou le paquet dans une file de sortie, ils auront au passage modifié l'en-tête afin de permettre au paquet ou à la trame de continuer son parcours vers un autre nœud.



De chaque côté du lien, des équipements sont connectés.

L'ETTD Equipement Terminal de Transmission de Données est une des machines du réseau point de départ des informations à transmettre.

L'ETCD est placé aux extrémités des réseaux interconnectés ou support de transmission, il aura pour objet de réaliser l'adaptation du signal aux caractéristiques du support afin d'autoriser la transmission.

## Paramètres de l'ETCD

Codage (bande base ou modulation)  
Rapidité de modulation en bauds

Débit en bits/s  
Mode et sens de transmission  
Interface avec l'ETTD

## Quatre catégories d'ETTD

Terminaux lourds : (mini ordinateurs) capacités de transmission importante sur liens synchrone haut débit.

Terminaux légers : clavier et écran, utilisent des liens asynchrone bas débit.

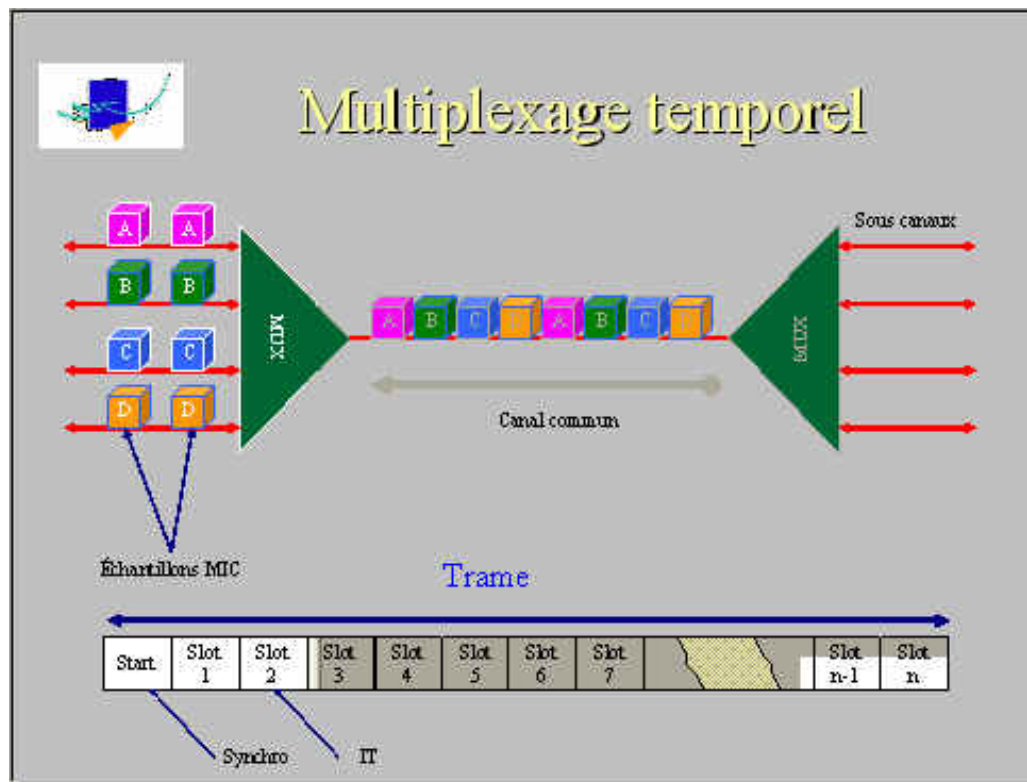
Terminaux intelligents : traite en local les données et n'échangent que ce qui est indispensable.

Les postes de travail : doté de processeurs puissants, ils sont entre les terminaux lourds et les terminaux intelligents.

## Multiplexages

Si l'on considère une ligne de communication de point à point, il est quelquefois avantageux de partager les moyens de transfert entre plusieurs utilisateurs, l'infrastructure devient alors commune. Le Multiplexeur ou MUX recevra les données de multiples sources, souvent à vitesse lente, pour les transmettre sur un lien à haut débit. Le démultiplexage (opération inverse) interviendra sur le MUX opposé, ce qui permettra après extraction, d'acheminer les données de chaque source initiale vers le destinataire correspondant ne fonction du débit de sa ligne.

## Multiplexage temporel



## Multiplexage temporel

### Fonctionnalités du MUX

Le MUX permet le partage d'un lien numérique. Les échantillons vocaux numérisés à chaque communication utilisent un canal propre.

Il permet à de nombreux circuits vocaux d'être transportés sur une seule liaison physique.

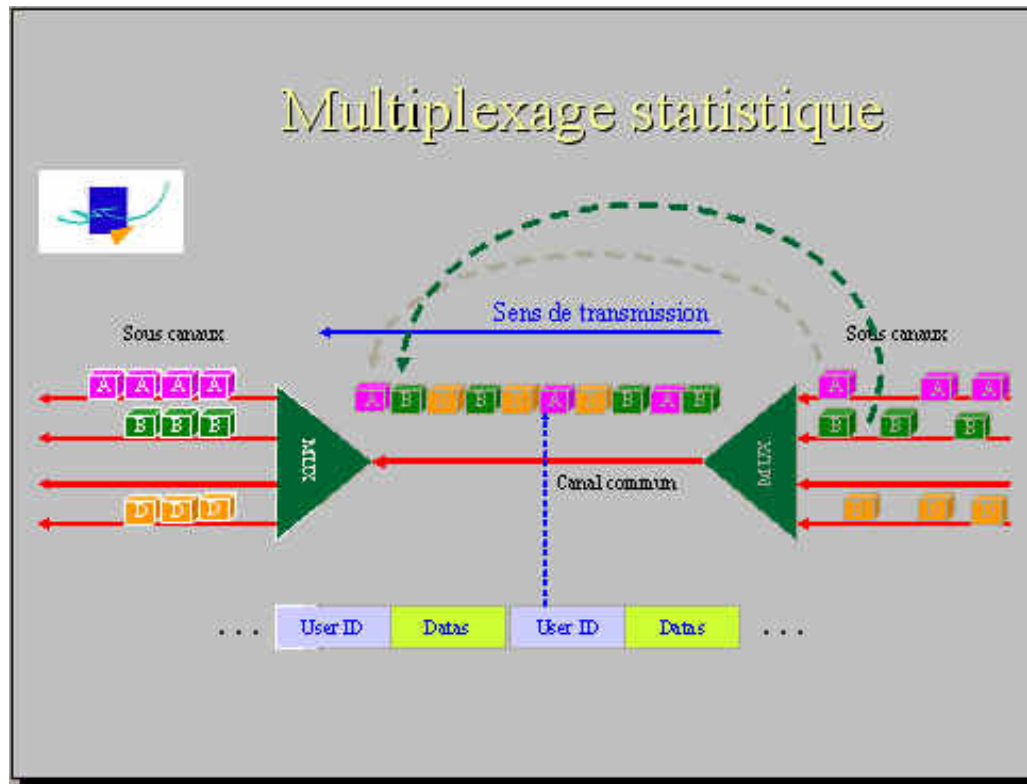


Appelé système à gain de paires, plusieurs canaux sur une seule paire torsadée.

### Mux en téléphonie

Un canal unique transporte toutes les trames divisées en IT, 8000 trames /sec, une toutes les 125 microsecondes. Chaque trame débute par une synchro suivie de n IT. Chaque IT contient un échantillon vocal codé sur 8 bits.

### Multiplexage statistique



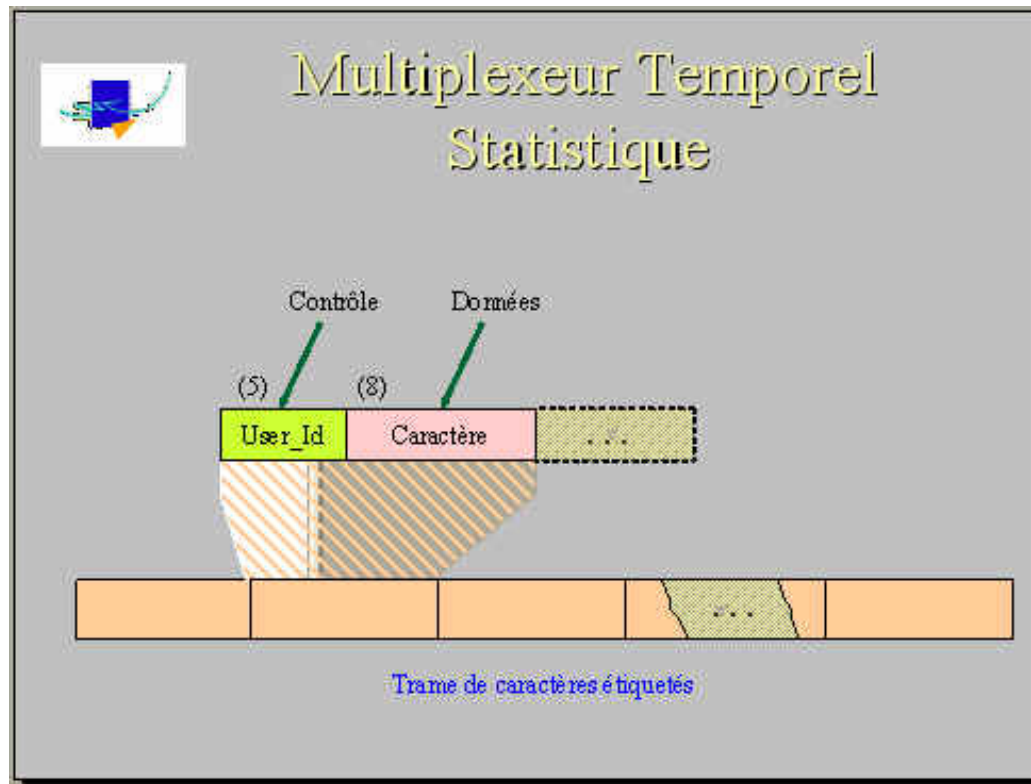
### Multiplexage statistique

#### Intervalle de temps disponible

Les données sont mémorisées jusqu'à ce que le MUX ait un intervalle de temps disponible .

Les données sont transmises dans l'ordre de réception.

### Multiplexeur Temporel Statistique



### Multiplexeur Temporel Statistique

#### IT libres

Quelques utilisateurs remplissent les IT qui leurs sont affectés. Des IT restent donc libres.

#### Principe du MUX temporel Statistique

Un caractère utilisateur est étiqueté avec un User ID. Le Multiplexage Temporel Statistique peut être utilisé pour la commutation de paquets.

On obtient un meilleur contrôle des information et datas.

### Le multiplexage traditionnel

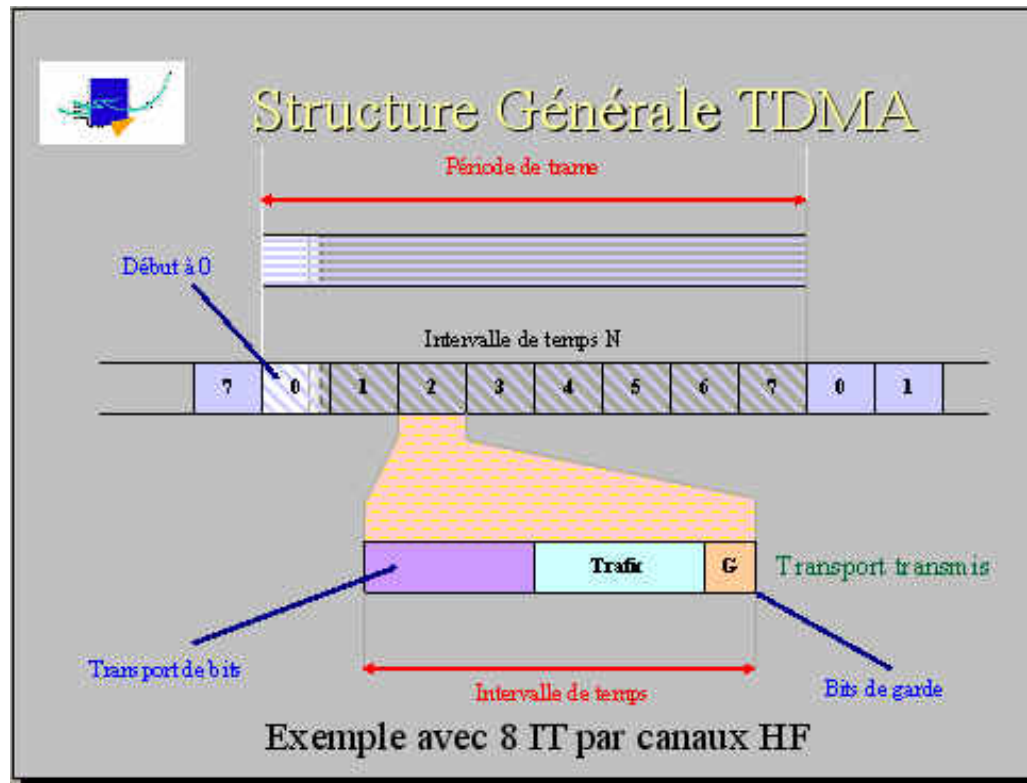
#### Schémas d'accès multiples

Limites du multiplexage traditionnel : Il alloue un utilisateur par canal, cette allocation est inefficace. Le trafic voix et donnée nécessite l'accès au canal pour une durée limitée.

#### Avantages de l'accès multiple

Il résout le problème des canaux sous utilisés, il permet l'accès au réseau à plus d'utilisateurs que de canaux disponibles (ou IT).

## L'accès multiple



### Time Division Multiple Access

#### Communications Numériques

Le TDMA est utilisé pour les communications numériques sur réseaux cellulaires et par satellites.

### Le SBS Satellite Business System

Il a été mis en place en 1980 et est utilisé pour la voix compressée, les données à hautes vitesses, la télécopie rapide, la messagerie électronique, la visio conférence.

Le système SBS permet le mixage de voix et des données.

- **Synchronisations et allocations**

La station de référence contrôle la synchronisation, les allocations, Elle établit une fenêtre de demande (15 ms au total), elle reçoit les demandes de trafic des stations durant la fenêtre de demandes, elle envoie dans le même temps des affectations d'IT aux stations et la synchronisation.

- **Multiplexage voix et données.**

Les stations multiplexent voix et données dans les messages et chaque transpondeur du satellite a un débit de 48 Mbps.

## Interconnexion de réseau par une fibre optique

### Trois spécifications :

- 10BASEFP (P=Passive Star)

La spécification définit une topologie en étoile passive qui fusionne les fibres optiques s'y raccordant et permet l'éclatement du signal lumineux. Le cœur de l'étoile ne comporte aucun dispositif électronique, son temps de traversée est quasiment nul, la longueur entre transmetteur et étoile ne doit pas être supérieure à 500 m.

- 10BASEFL (L=link)

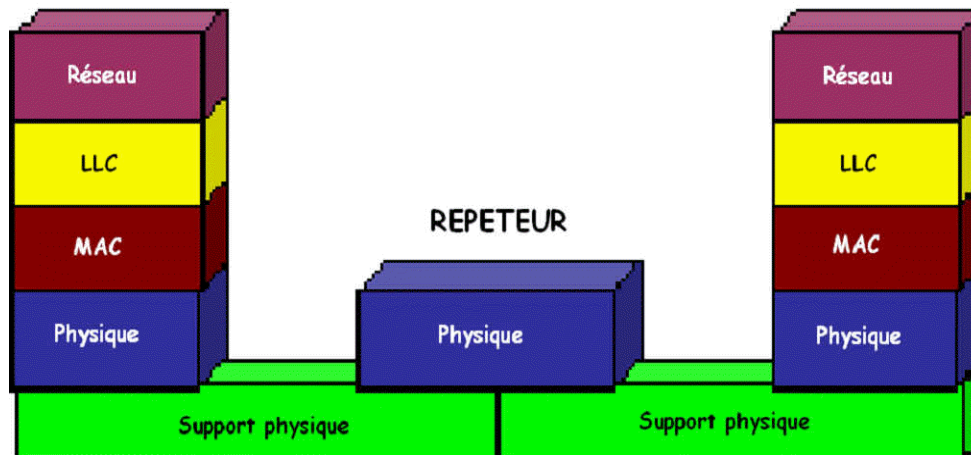
Désigne les répéteurs de type FOIRL (Fiber Optic Inter Repeater Link) qui permettent l'interconnexion à distance de deux segment de câble ou de fibre optique. Elle peut permettre de construire une topologie en étoile autour d'un multiport, les distances sont fonction des spécifications 1 Km et plus récemment 2 Km.

- 10BASEFB (B=Backbone)

Ce support définit une topologie en étoile active permettant la mise en place d'un réseau fédérateur.

Les transmetteurs sont des FOMAU (Fiber Optic Medium Access Unit), la distance étoile / transmetteur peut aller jusqu'à 2 Km.

### Répéteurs



Objet :

Les répéteurs sont destinés à permettre la propagation d'un signal au-delà de la limite fixée par la propagation du signal sur un simple support physique ou média de réseau.

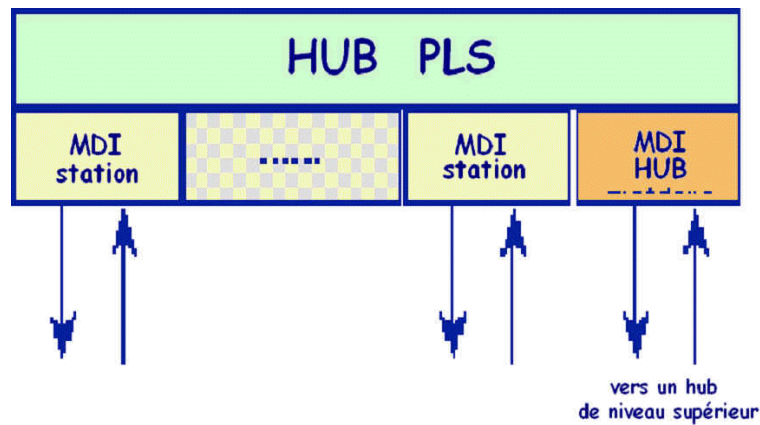
**Exemple :** en 10 Mbps un câble coaxial blindé est limité à 500 m, au-delà le taux d'erreur constatée ne permettrait pas aux équipements en communication de recevoir les données transmises. Le diamètre de ce réseau sera porté au maximum à 2500 m grâce à l'interposition de 4 répéteurs au plus entre l'émetteur et le récepteur.

Le répéteur permet également de marier des supports physiques différents (sous réserve de conservation de la structure de trame initiale). Exemple : on pourra prolonger du Cuivre par de la Fibre Optique avec un répéteur approprié.

## Hubs et Base T

### Généralités

Ils se présentent sous forme de modules en châssis (image ci-dessous) ou en boîtiers empilables. Les stations sont raccordées en étoile au HUB (niveau physique).



### L'architecture du hub comporte deux niveaux

MDI (Medium Dependant Interface) avec une entité par port d'accès.

PLS (Physical Layer Signalling) pour la communication entre les ports.

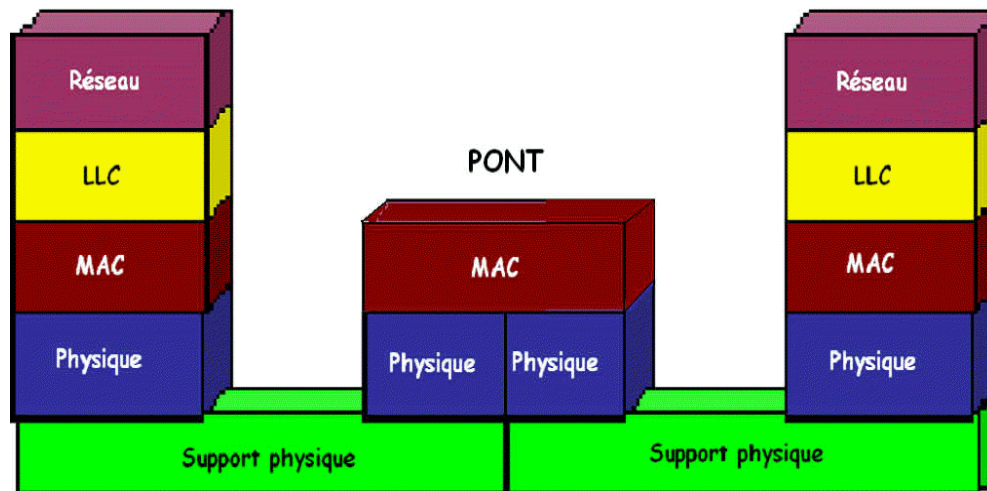
### Mode de connexion

Chaque station est connectée en un point (port RJ45) au hub grâce à deux paires torsadées (une pour chaque sens de transmission). Le câble utilisé couramment en 10 BASE T est de type UTP de classe 5 (qualité données). L'utilisation de câble STP est possible (bien que non définie par la spécification).

Afin d'autoriser des cascades le hub offre, en outre, un port pour AUI vers une MAU sur lien 10 BASE 5 ou un port de type BNC pour lien 10 BASE 2 (Ethernet fin).

## Ponts

Ils permettent d'étendre la portée géographique d'un réseau, de décharger un segment du réseau. Ils assurent un routage transparent pour la couche LIAISON



### Synthèse

Ils sont transparents pour les utilisateurs de la Couche MAC et donc des Couches supérieures

Ils ne sont pas adressés pour effectuer des fonctions de routages

Ils filtrent les messages qu'ils transmettent. Ce routage n'interprète pas les adresses

Ils décident du maintien ou non d'AD dans le réseau d'origine  
 Ils réalisent une adaptation de vitesse entre réseaux Technique. store and forward  
 Ils assurent la transmission de messages avec adresse de groupe  
 Les topologies avec pont seront arborescentes  
 Les boucles seront interdites pour éviter le retour par un autre pont  
 Ils ne modifient ni interprètent aucune information de la partie donnée de la trame MAC  
 Ils permettent d'utiliser plusieurs médiums différents dans chaque réseau qu'il relie  
 Les délais ne sont pas garantis de bout en bout à cause des files d'attente

Le pont enregistre dans des tables internes les adresses de toutes les stations.

### Les tables

Les ponts exécutent une lecture de l'adresse source (émetteur), si absente de table de sortie, elles en font l'ajout. A chaque entrée est associée une durée de vie. Si la durée maximum est atteinte, l'adresse source est retirée de la table de sortie.

Si l'adresse de destination est vue dans la table de sortie locale: contention du message dans le réseau d'origine, sinon il y aura copie dans file de sortie vers l'autre réseau.

### La Technique store and forward entraîne

- Des délais ( tables saturées et temps de mémo du message.)
- Des pertes éventuelles (débit des réseaux très différents)

Il est possible de définir un format et des champs dans les adresses pour en réduire la portée dans le cas de diffusions. Exemple : IBM définit un champ anneau destinataire

dans les 48 bits, cette technique existe en FDDI, ce champ indique si le message doit être retransmis ou pas.

## Composantes de la couche Physique

### La couche Physique

#### La couche Physique

##### Trois parties majeures

- Sous couche PLS (Physical Layer Signalling)

Gère l'interface avec MAC, permet de générer les signaux électriques pour les bits issus de MAC. Les signaux sont véhiculés sur le support physique, la surveillance des signaux est assurée et une génération de signal de détection de collision est prévue.

Inversement code les signaux physiques du support en signaux logiques pour MAC réceptrice.

- AUI (Attachment Unit Interface)

La MAU peut être embarquée sur la carte, dans ce cas il n'y a pas d'AUI, elle permet à la station d'être éloignée du support (cas spécifiques),

##### Composition:

Deux circuits de données (data\_in et data\_out).

Deux circuits de contrôle (control\_in et control\_out) commande de la MAU.

Un circuit d'alimentation.

- MAU (Medium Attachment Unit)

Gère les fonctions du niveau physique, diffère selon le support de transmission employé.

**Fonctions:**

Transmission d'un signal sur le support.  
Réception d'un signal provenant du support.  
Reconnaissance de la présence d'un signal sur le support.  
Reconnaissance d'une collision.  
Interruption automatique d'une trame anormalement longue.

**Composition:**

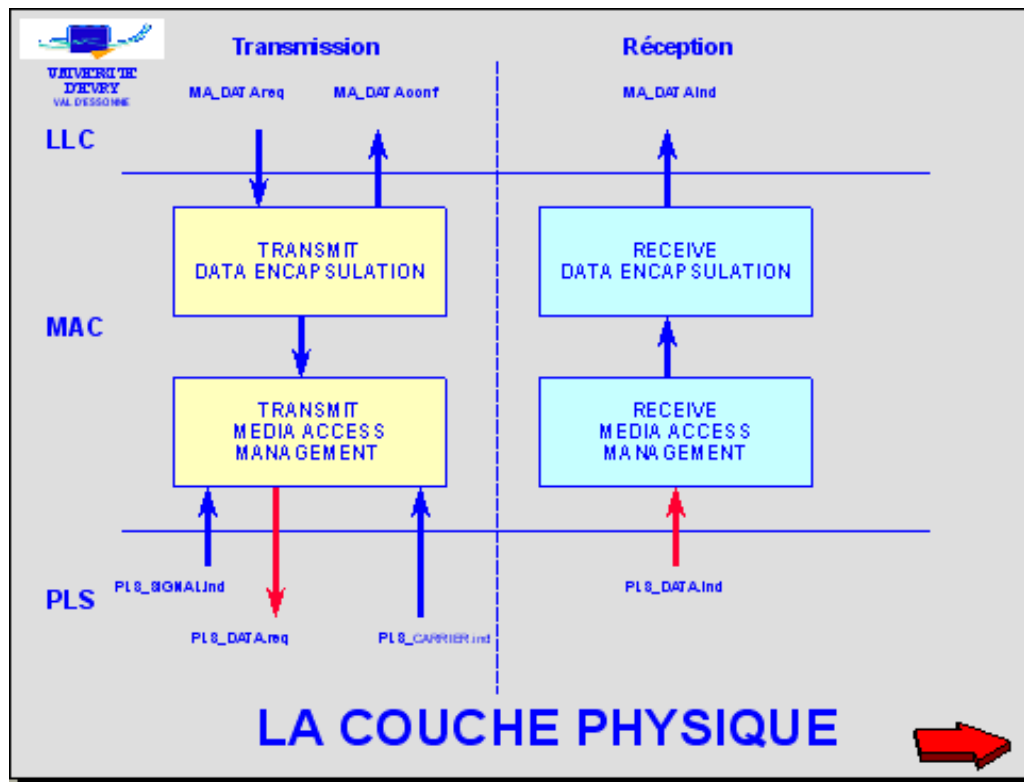
Boîtier d'accès attaché au câble (transceiver).  
Circuit sur une carte interface en fond de panier connectée au bus interne de la machine.

## Interactions et primitives

### Interactions PLS/MAC/LLC - Primitives, paramètres et valeurs

#### Interactions PLS/MAC/LLC





## La Couche Physique

Primitive	Paramètre	Valeurs
PLS_DATA.Request	output_unit	1,0 : valeur binaire de la donnée DATA_COMPLETE Transm. finie
PLS_DATA.Indication	input_unit	1,0 : valeur binaire de la donnée
PLS_CARRIER	carrier_status	CARRIER_ON signal détecté par la MAU CARRIER_OFF aucun porteur détecté par la MAU
PLS_SIGNAL	signal_status	SIGNAL_ERROR la MAU a détecté une collision NO_SIGNAL_ERROR aucune collision détectée

➔

### Les Primitives de service

#### Deux aspects :

- Pour transfert de données entre MAC et le prestataire PLS :

PLS\_DATA.Request (output\_unit)  
PLS\_DATA.Indication (input\_unit)

- Pour effet local à l'interface MAC-PLS

PLS\_CARRIER.indication (carrier\_status)  
PLS\_SIGNAL.indication (signal\_status)

### Primitives, paramètres et valeurs

## Génération et Effets

- Primitive PLS\_DATA.request

Cette primitive est générée par MAC afin de demander à PHY de transmettre un bit de donnée sur le support, ou d'arrêter la transmission. A réception PLS encode et transmet le bit ou indique la fin de transmission selon le cas.

- Primitive PLS\_DATA.indication

Elle est générée par la sous couche PLS à destination de toutes les entités MAC du réseau à la suite d'une requête (voir ci-dessus).

- Primitive PLS\_CARRIER

Elle rend compte de l'activité sur le support à la couche MAC. Elle est générée à chaque changement du paramètre Carrier\_status

- Primitive PLS\_SIGNAL

Elle indique l'état de la couche physique. Elle est générée à chaque changement du paramètre Signal\_status

# Le niveau MAC : 802 et CSMA/CD

Sommaire :

[Principes](#)

[Protocole CSMA/CD](#)

## Principes

[Norme IEEE 802](#) - [L'adressage](#) - [Autres formats d'adresse](#) - [Le délai de propagation](#) - [Notion de tranche Canal \(bus\)](#) - [La détection d'interférences](#) - [Délai de propagation sur Boucle](#)

## Norme IEEE 802

La norme **802.1** décrit les fonctions de gestion du réseau, elle comporte des fonctions spécifiques au type de MAC choisi et la norme **802.2** décrit la couche LLC .

Elles sont communes aux trois normes MAC :

- 802.3 protocole CSMA/CD
- 802.4 protocole jeton sur Bus
- 802.5 protocole jeton sur Anneau

Plusieurs normes leurs sont ajoutées pour la couche PHYSIQUE

## L'adressage

Il est réalisé sur une liaison à laquelle les abonnés sont raccordés, Ceux-ci reconnaissent leurs adresses au passage.

## FORMAT des ADRESSES

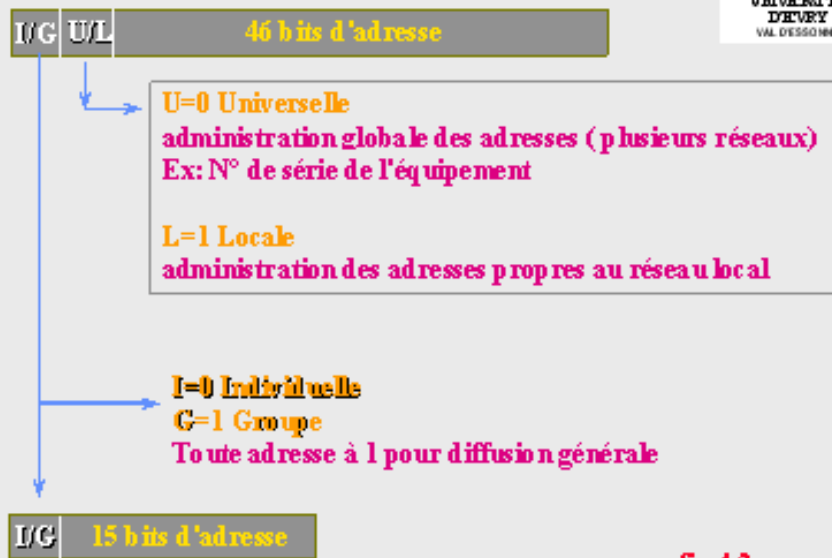


fig 4.2

Deux formats :

- court 16bits HDLC
- long 48 bits

Le coupleur à une ADRESSE physique unique.

Notion d'adresse de groupe:

- Une adresse de **groupe** est commune à plusieurs coupleurs elle permet la diffusion

En format étendu une adresse est dite:

- **universelle** si gérée par un organisme, dépendent de IEEE
- **locale** si gérée par administrateur

Autres formats d'adresse

Adresses dans les Boucles :

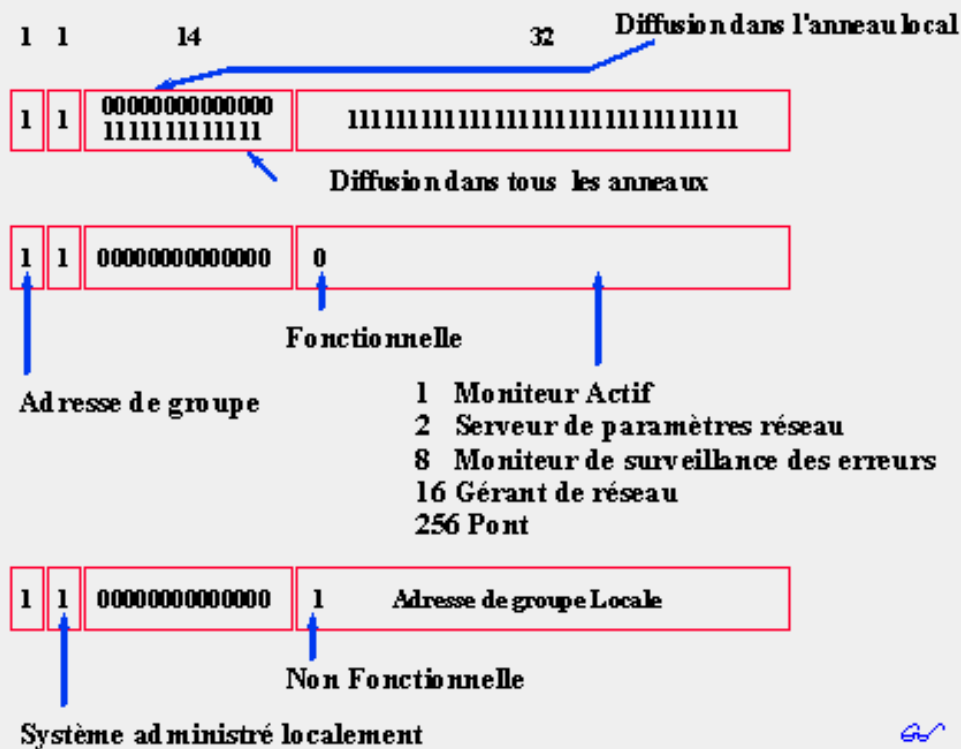
- anneau à jeton IBM (Token Ring)
- FDDI

Le champ des 48 bits ou 16 bits a été décomposé :

- deux 1er octet si Adresse longue.
- 1er octet si Adresse courte
- Ils servent à

Michel Besson

## Diffusion des adresses dans les boucles



désigner  
l'anneau :

- les 1er bits conservent leur signification.

**Décomposition Adresse longue :**

- 14 bits = No anneau
- 0 dans le champ = anneau local
- 1 dans le champ = tous les anneaux
- 32 bits = No station

**Notion d'Adresse fonctionnelle :**

Elle est ajoutée dans la partie Adresse station :

- 1er bit=1..... adresse normale ...ou
- 1er bit=0 .....adresse fonctionnelle qui désigne :
  - le moniteur actif
  - le moniteur surveillance erreur
  - le gérant du réseau
  - un pont

**Le délai de propagation**

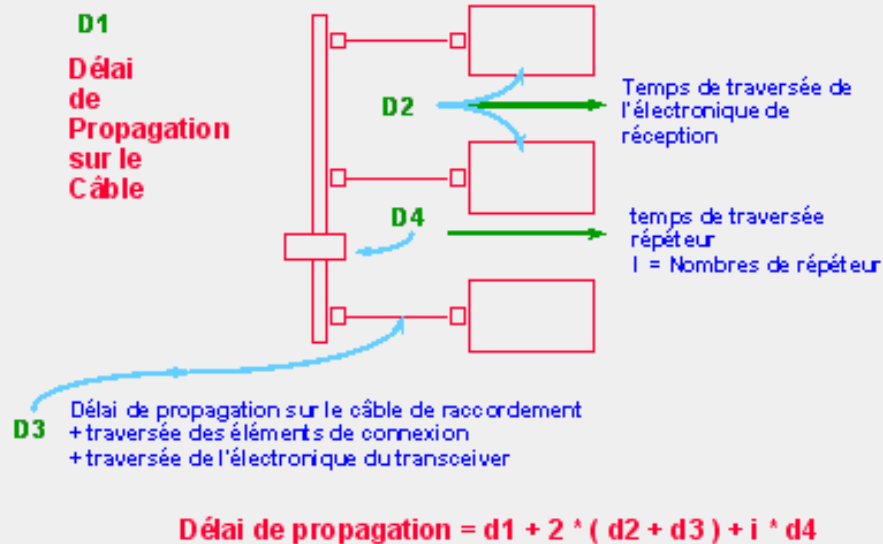


Figure 4.3 : Délai de propagation

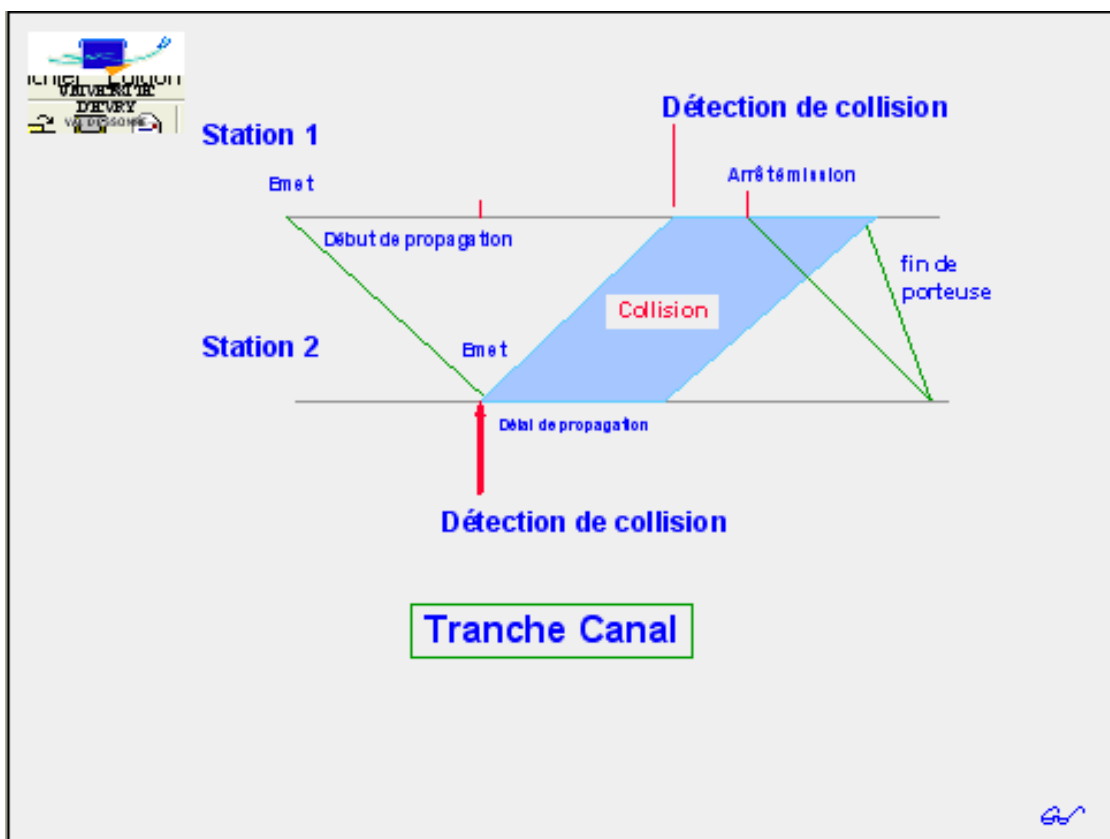


Délai de Propagation sur Bus

somme de 4 délais :

$$p = d1 + 2 * (d2 + d3) + i * d4$$

## Notion de tranche Canal (bus)



Notion de tranche Canal (bus)

Permet de définir la durée depuis le 1er bit émis et l'instant où l'émetteur est sûr qu'aucun signal n'a perturbé son émission

$$TC = 2p$$

## La détection d'interférences

Scénario du mécanismes CSMA/CD :

**Comment à un instant  $t$ , la station 1 voit-elle le canal libre ?**

- la **station 1** émet, mais à  $T + \Delta$  la station 2 voit aussi canal libre (  $\Delta < p$  ).
- la **station 2** émet ;
- la **station 2** cesse d'émettre peu après avoir observé la collision qui s'en suit.
- la **station 1** verra la collision à  $t + \Delta + p$  au plus tard et cessera d'émettre aussi.
- la **station 2** verra le canal libre à nouveau à  $t + \Delta + 2p$ .

**Autres brouillages possibles :**

Sur le bus on peut observer des phénomènes d'échos, ces brouillages seront vus comme des collisions ; pour s'assurer de leur disparition on respecte une attente (silence inter message),

Le délai d'attente =  $d_1 + i \cdot d_4$

**Délai de propagation sur Boucle**

Il dépend de la longueur du câble et du temps de traversée stations ( station = répéteur ).

Ce délai est pour N stations :  $Nd_4 + d_1$  ; le délai est donc dépendant du nombre stations connectées.

## Protocole CSMA / CD

[Caractéristiques](#) - [Détection de conflits](#) - [Acquisition / Ajournement](#) - [Résolution des conflits](#) - [Principes retenus pour le CSMA/CD](#) - [Trame 802.3](#) - [Paramètres du CSMA/CD](#) - [Description algorithmes en émission/réception](#) - [Les primitives de services MAC](#) - [Etat et processus des échanges en couche MAC](#)

### Caractéristiques

L'accès au canal est **aléatoire**

Les messages en **conflit** sont **perdus**

On procède à un **retardement** du message si le canal est **occupé**

Il y a **arrêt** de transmission si le message émis **n'est pas entendu**

**3 aspects techniques de CSMA doivent être considérés :**

- Détection de conflits
- Acquisition ajournement
- Résolution de conflits
- 

### Détection de conflits

## Méthode : la détection de collisions

Une station qui émet écoute simultanément.. au maxi 2p

Compare le message émis au message écouté, s'il est brouillé : arrêt et retransmission ultérieure suivant algorithme de résolution de conflit utilisé.

**On distingue 2 types d'interférences :**

### a) Détection sans forçage

Toutes les stations qui émettent détectent le conflit.

### b) Détection avec forçage

Toutes les stations qui émettent détectent le conflit, sauf une qui réussit à écouler son message sans interférence; dans ce cas on peut mettre en oeuvre un système de priorité.

## Acquisition / Ajournement

**Méthode :**

Si message à émettre ( ancien ou nouveau ) l'**acquisition** correspondra à une tentative si canal libre. L'**ajournement** est l'arrêt de la tentative si canal occupé.

**Types d'ajournements :**

### 1 non persistant :

Si le canal est occupé, les messages sont ajournés comme s'il y avait conflit

### 1-persistant :

On attend la libération du canal, si canal libre il y aura tentative immédiate d'émission avec une probabilité de conflit=1 en supposant que plusieurs stations soient également en attente.

## Résolution des conflits

**3 politiques peuvent être observées :**

### 1 Réémission non adaptative :

La station retarde la prochaine émission pendant une durée aléatoire (tirée d'une distribution constante en temps).

### 2 Réémission adaptative :

La loi de distribution des délais est variable .

Le contrôle est local ou global.

But : adapter la fréquence de tentative à la charge du canal.

## Principes retenus pour le CSMA/CD

**Base de l'algorithme**

- **Détection des conflits** par Détection d'interférence.



- **Ajournement 1-** persistant pour acquisition/ajournement.
- **Résolution des conflits** sur Délai adaptatif de réémission par contrôle local.

Ce délai tiré aléatoirement de la loi exponentielle binaire dont la moyenne est fixée pour la 1ere réémission ; ensuite pour chaque conflit la moyenne est multipliée par 2

## Trame 802.3

### Structure des trames

Préambule	Synchro horloge du récepteur, pare-chocs
SFD	Doit être intégralement reconnu ; 1-1 consécutifs, ils marquent le début de la trame MAC
Adresse destinataire	Utilisée par couche Physique qui prend ou non copie si reconnu
Adresse source	Remplie par Couche Physique à l'émission.
Length	Longueur des données LLC, le reste est le PAD
Datas LLC	Données utiles LLC
PAD	Bourrage effectuée par PHY pour atteindre la taille minimale.
CRC	Correct si valeur finale = 0 à l'instant du dernier bit

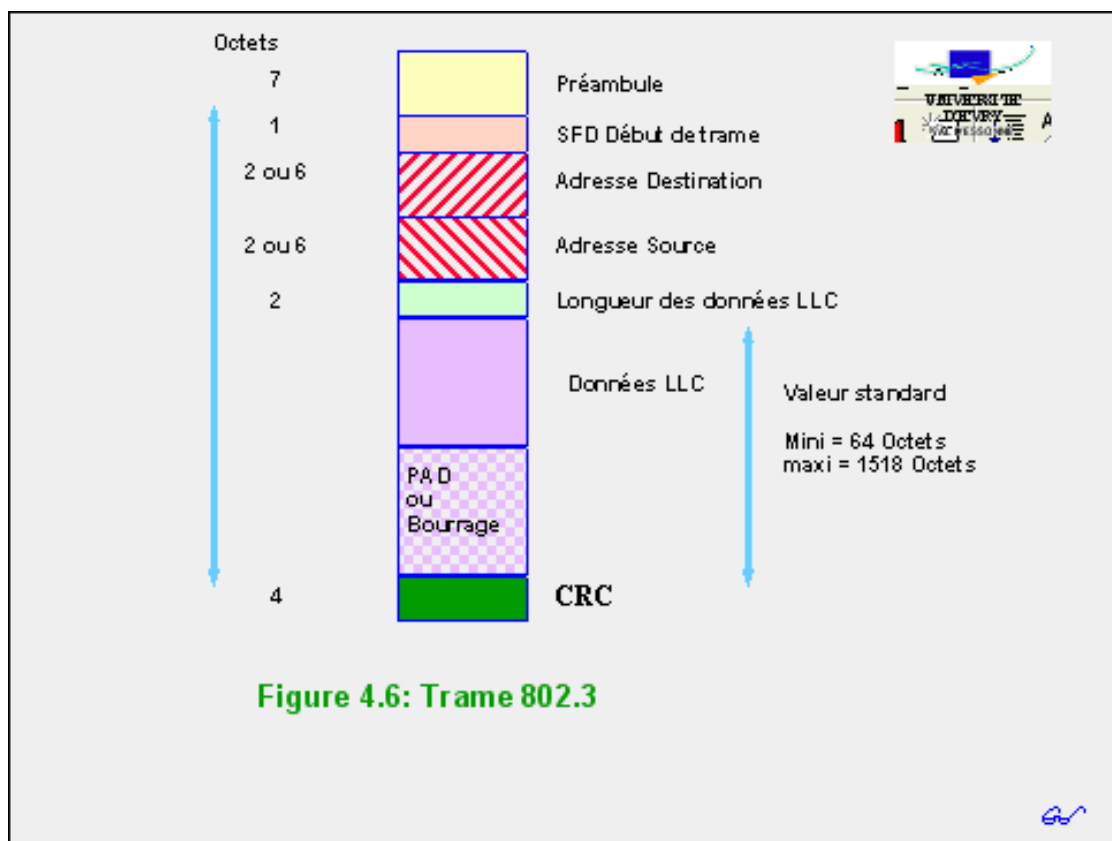


Figure 4.6: Trame 802.3

### Longueur minimum de trame

Elle est fixée à 64 octets par durée d'émission  $D = > \text{tranche canal} (TC=2p)$ .

### Longueur maxi de trame

Elle est fixée à 1518 octets pour limiter l'occupation canal.

## Paramètres du CSMA / CD



### PARAMETRES DU CSMA/CD

Paramètre	Signification	Valeur
SLOT TIME	fenêtre de collision	512 temps-bit
INTER FRAME GAP	attente entre deux transmissions	9,6 microsecondes
ATTEMP LIMIT	nombre maximal de retransmissions	16
BACKOFF LIMIT	limite de l'intervalle de tirage	10
JAM SIZE	taille de la séquence de bouillage	4 octets
MAX FRAME SIZE	longueur maximale de la trame	1518 octets
MIN FRAME SIZE	longueur minimale de la trame	64 octets
ADRESS SIZE	longueur du champ d'adresse	48 bits

### Caractéristiques Physiques normalisée

La longueur maxi d'un segment est définie:

- par la norme
- par le choix  
BRO ou BAS ou  
OPT

Le débit est défini par un triplet

- 1 BAS 1, 10  
BAS 2
- 10 BAS 5, 10  
BAS T
- 10 BAS F

Ces éléments sont interopérables

### Description algorithme en émission / réception

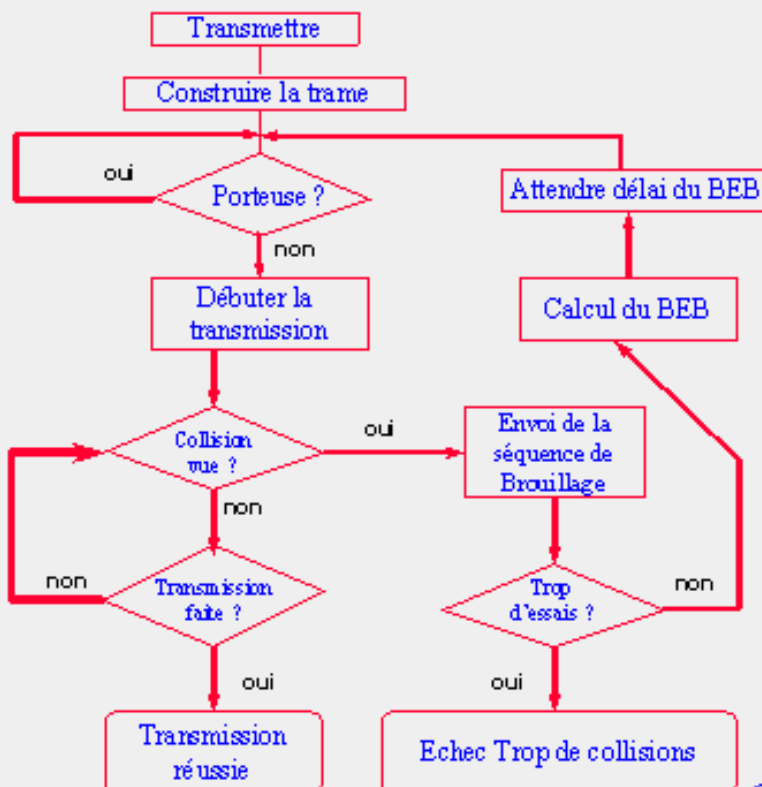
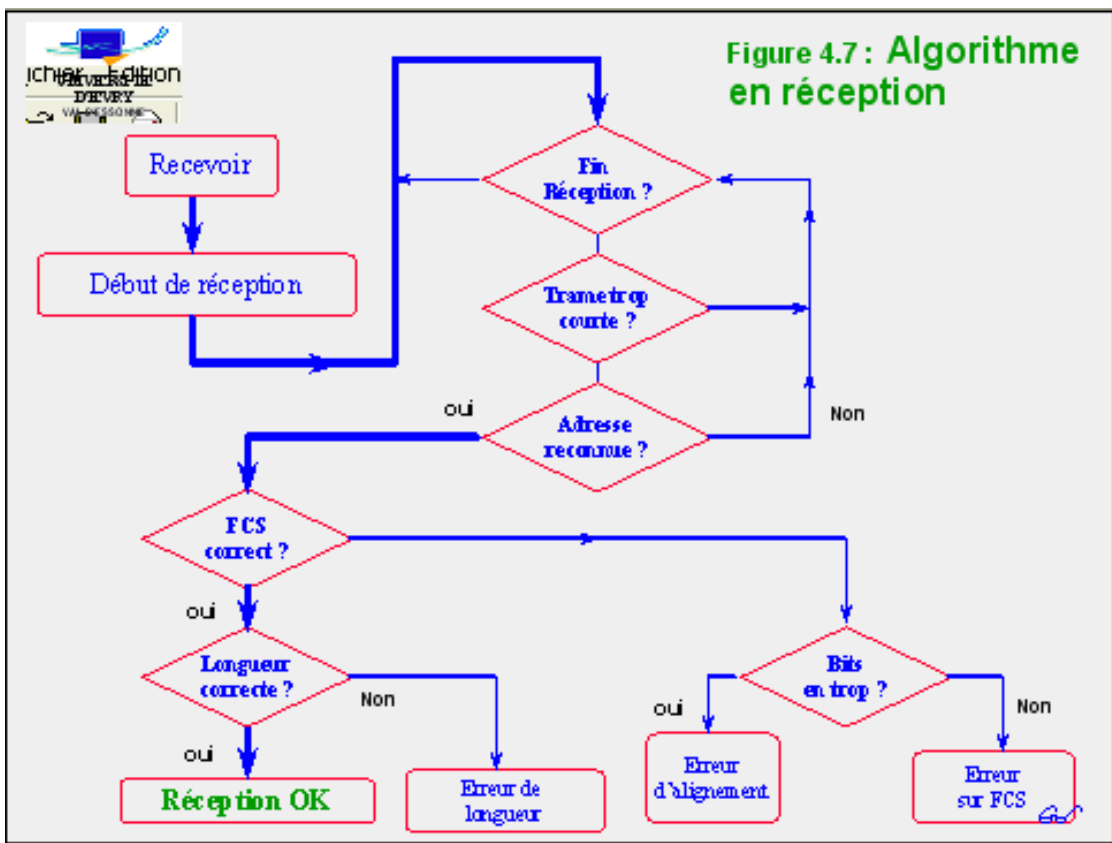


Figure 4.5 :  
Algorithme  
d'émission

### Algorithme en émission

Le BEB Binary  
Exponential Backoff

- calcule le délai  
aléatoire  
d'attente
- le nombre  
d'essai maxi =  
16



### Détection erreurs

Fin correcte : si val=0 détectée à l'instant du dernier bit.

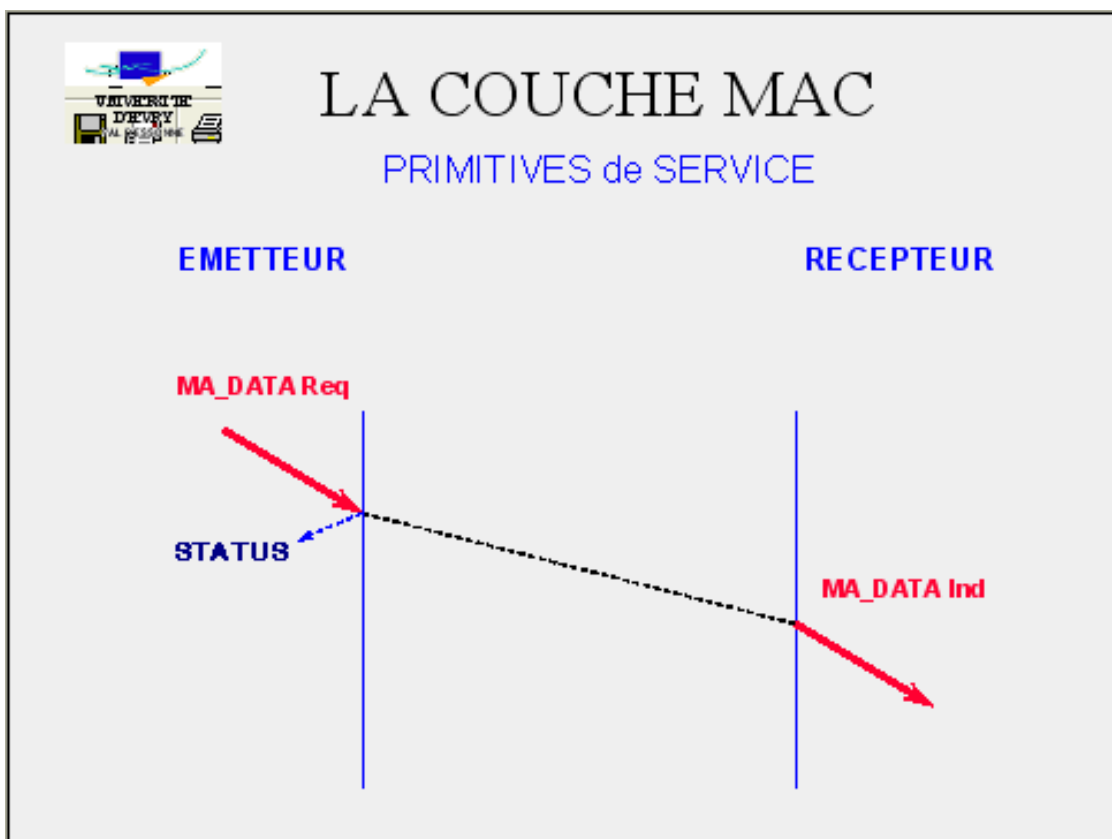
Erreur d'alignement : une trame doit contenir 1 nombre entier d'octet.

Parasite sur canal : il est vu comme collision

L'insert ou le retrait station n'a aucune incidence (contrôle local).

Nota : Pour combler le défaut relatif du BEB (probabiliste) un protocole 802.3D ou CRCD apportant une garantie de transmission a été étudié.

### Les primitives de services MAC



Elles sont utilisées par la couche LLC pour permettre l'échange de données entre MAC et LLC.

Le protocole CSMA/CD étant en mode non connecté, seul le transfert de données est pris en compte.

Il y a 2 primitives MAC :

- MA\_DATA request
- MA\_DATA indication

## Primitive MA\_DATA.Request

Elle permet le transfert de données d'une entité LLC émettrice vers une ou plusieurs (adressage de groupe) entités réceptrices.

A l'arrivée de cette primitive au niveau MAC, une trame est constituée à partir de cette primitive et des valeurs propres au niveau MAC (Ad source, longueur des données et séquence de contrôle)

## Sémantique de la primitive MA\_DATA.Request

Paramètres: dest\_address, lengh\_data, m\_sdu, service\_class, transmit\_status (fonction de compte rendu local)

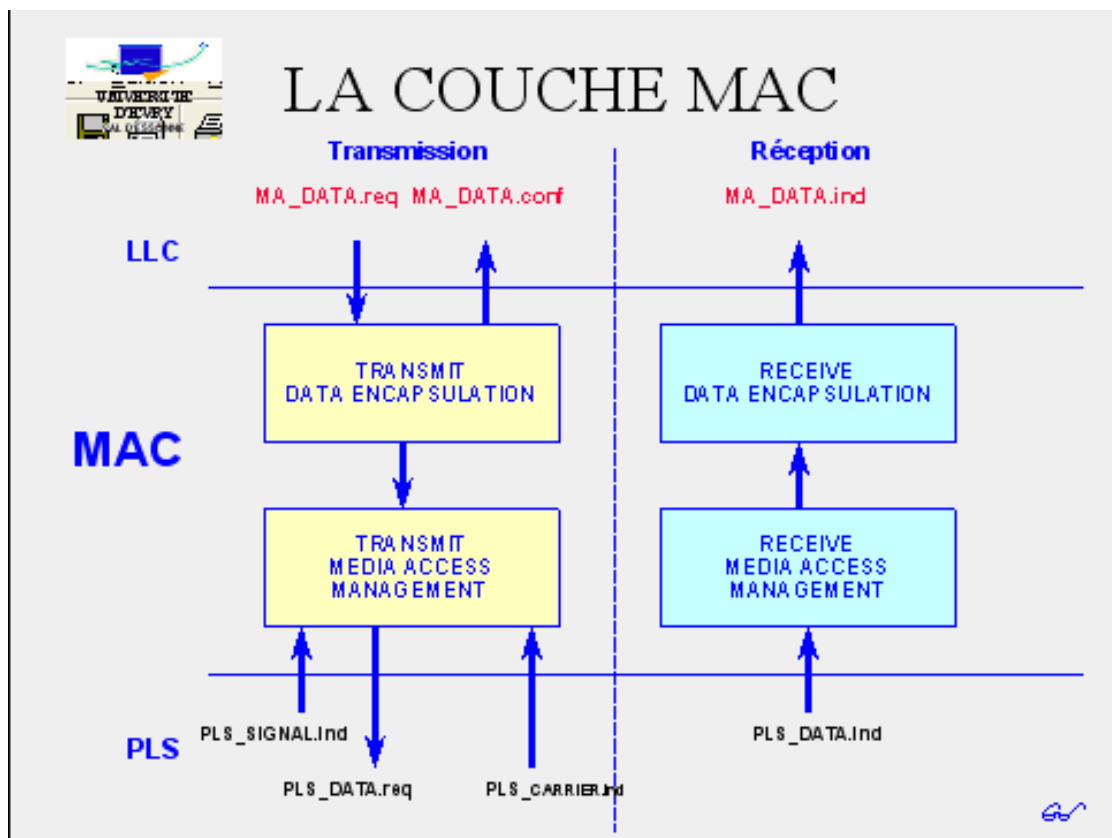
- dest\_address: adresse simple ou adresse de groupe
- lengh\_data: longueur des données LLC
- m\_sdu: unité de données de MAC, soit les données LLC
- service-class : qualité de service demandée par la couche LLC ou un niveau supérieur. (non utilisé dans CSMA/CD)
- Transmit\_Status est le résultat d'une fonction retournant un compte-rendu sur le déroulement de la transmission
  - Valeur 1: transmit OK
  - Valeur 2 : excessive\_collision\_error (abandon de transmission)

## Etat et processus des échanges en couche MAC

### Transmit Data Encapsulation

### Réception des données de la sous couche LLC

- construire la trame



### Transmit Media Access Management

Présentation d'une série de bits à la couche PHY pour transmission sur le support

- Attente si support occupé
- Ajout FCS aux trames sortantes
- Attente fin silence inter message
- Activation du processus de gestion de collision si constat

## **Receive Media Access Management**

Réception d'une série de bits depuis la couche PHY

- vérification du FCS des entrantes
- destruction des trames trop courtes

## **Receive Data Encapsulation**

Présentation des trames reçues à LLC avec adresses de diffusion ou de station

- éliminer les trames ne portant pas d'adresses de station
- extraction de la partie données des trames reçues

# Le niveau MAC : Hauts débits et VLANs

Sommaire :

[Les technologies du 100 Mbits](#)

[Le Gigabit Ethernet](#)

[Ethernet et la commutation](#)

[Les Virtual LANs](#)

## Les technologies du 100 Mbits/s

[Généralités](#) - [Principes généraux](#) - [Protocole 100 Base TX](#) - [Protocole 100 Base FX](#) - [Protocole 100 Base T4](#) - [Round Trip Colision Delay](#) - [Le Path Delay Value en 100 Base T](#)

### Généralités

Egalement appelé Fast Ethernet, il est l'extension du réseau Ethernet à 100 Mbits/s. Trois sous normes sont proposées pour le 100 Mbits/s : 100 Base TX, 100 Base FX, 100 Base T4.

### Principes généraux

#### Méthode d'accès CSMA / CD

La méthode est conservée avec ses qualités : efficacité, rapidité, mais aussi son défaut : non déterministe.

#### Format de trame

Il est identique à celui du 10 Base T.

#### Longueur des trames

Elle est identique à celle du 10 Base T soit au minimum 64 octets, ce qui représente donc un temps de transit de 5,12µs. La distance qui peut être parcourue durant cette période n'excède pas 1000 m. Un réseau Ethernet aura donc une couverture maximum de 500 m. Cette distance sera réduite à 210 m en raison du temps non négligeable consacré à la traversée des Hubs. Le silence entre trame est réduit à 0,96 µs.

### Protocole 100 Base TX

Il reprend la couche TP-PMD ( Twisted pair - Physical Médium Dependent ) de FDDI..

Il utilise une conversion 4B/5B ( débit binaire : 125 Mbps) suivi du codage MLT-3 ( 3 Levels Multiline Transmission) = réduction de la fréquence du signal principal à 31,25 MHz.

### Compatibilité 10 Base T

Elle est identique.

Fonctionnement sur 2 paires torsadées UTP Cat 5 ou sur 2 paires torsadées blindées STP.

Longueur maxi 100 mètres et couverture maximum du réseau 450 m.

## Interconnexion

L'interconnexion peut être mise en oeuvre par un Hub-Commutateur.

## Evolutions

Commutation dynamique de paquets ( 100 Mbps par ports ).  
Fonctionnement en Full duplex ( 2x100 Mbps).

## Protocole 100 Base FX

Technologie identique au 100 Base T appliquée à la Fibre Optique multimode.  
Couverture maximum 450 m.

## Protocole 100 Base T4

### Différences avec le 100 base T

Le Cabling System : il supporte tous câbles de 4 paires torsadées non blindés.  
Longueur inchangée 100 m.

**Conversion** : 8B / 6T

### Mode de transmission

Les trois premières paires sont utilisées pour la transmission ( 25 MHz sur chaque) dans chaque sens, la quatrième paire pour la détection de collision.

## Round Trip Colision Delay

Il permet le calcul du domaine de collision. Autrement dit, il définit l'éloignement maximum entre les deux partenaires d'une communication, et ce de manière à garantir le bon fonctionnement des algorithmes prévus pour la détection de collision lors d'une transmission.

Cette limitation est fonction, entre autre, du type de répéteur utilisé ; l'IEEE 100BaseT définit deux types de répéteurs :

### Répéteurs de Classe I

- Temps de latence: 0,7 microsecondes maxi
- Un seul hop permis au maximum

### Répéteurs de Classe II

- Temps de latence: 0,46 microsecondes maxi
- Un ou deux hops permis au maximum



# Round Trip Collision Delay

## Technologies 100 Mbps Règles

Taille des domaines de collision

	Cuivre	Mixte cuivre / fibre multimode	Fibre multimode
DTE/DTE ou Switch - Switch	100 mètres		412 mètres 2000 Full Duplex
Un Répéteur de classe I	200 mètres	260 mètres	272 mètres
Un Répéteur de classe II	200 mètres	308 mètres	320 mètres
Deux Répéteurs de classe II	205 mètres	216 mètres	228 mètres

## Le Path Delay Value en 100 Base T

### Pourquoi calculer ce délai ?

- Afin de rester dans la norme CSMA/CD soit 512 Temps Bits en termes de fenêtre de collision et valider les configurations avant la mise en place physique des matériels.
- Afin de limiter les collisions et les erreurs de CRC

### Quelles valeurs doit on calculer ?

- LSDV (Link Segment Delay Value) :

C'est la valeur affectée par la longueur du lien entre 2 DTE.

DTE Data Terminal Equipments : tout équipement positionné en fin de segment (station, pont, router, switch à l'exception d'un répéteur).

- RDV Repeater Delay Value :

Soit le délai de traversée de tout équipement faisant office de répéteur du signal (class1)

- DTE Délai Value :

C'est le délai de transit (DTE DV) considéré entre un couple de DTE

- Safety margin value : une marge de sécurité 0,5 bt





# Round Trip Collision Delay

## Délai des Composants du Réseau

Composant	Round Trip Delay en Bit Time par mètre	Maximum Round Trip Delay en Bit Time
Deux TX/FX DTE Devices	---	100
Deux T4 DTE Devices	---	138
Un T4 DTE Device et un TX/FX DTE Device	---	127
Segment de câble CAT 3	1,14	114 (100m)
Segment de câble CAT 4	1,14	114 (100m)
Segment de câble CAT 5	1,112	111,2 (100m)
Segment de câble STP	1,112	111,2 (100m)
Segment de câble Fibre Optique	1,0	412 (100m)
Répéteur de classe I	---	140
Répéteur de classe I avec tout ports TX/FX	---	92
Répéteur de classe I avec port(s) T4	---	67

TX /FX /T4 = Repeater 100 base TX /FX /T4

### Formule du Path Delay Value

$PDV = \text{somme des LSDV} + \text{somme des RDV} + \text{DTE DV} + \text{Marge de sécurité}$

PDV doit être inférieur à 512 Bit Times (bt)

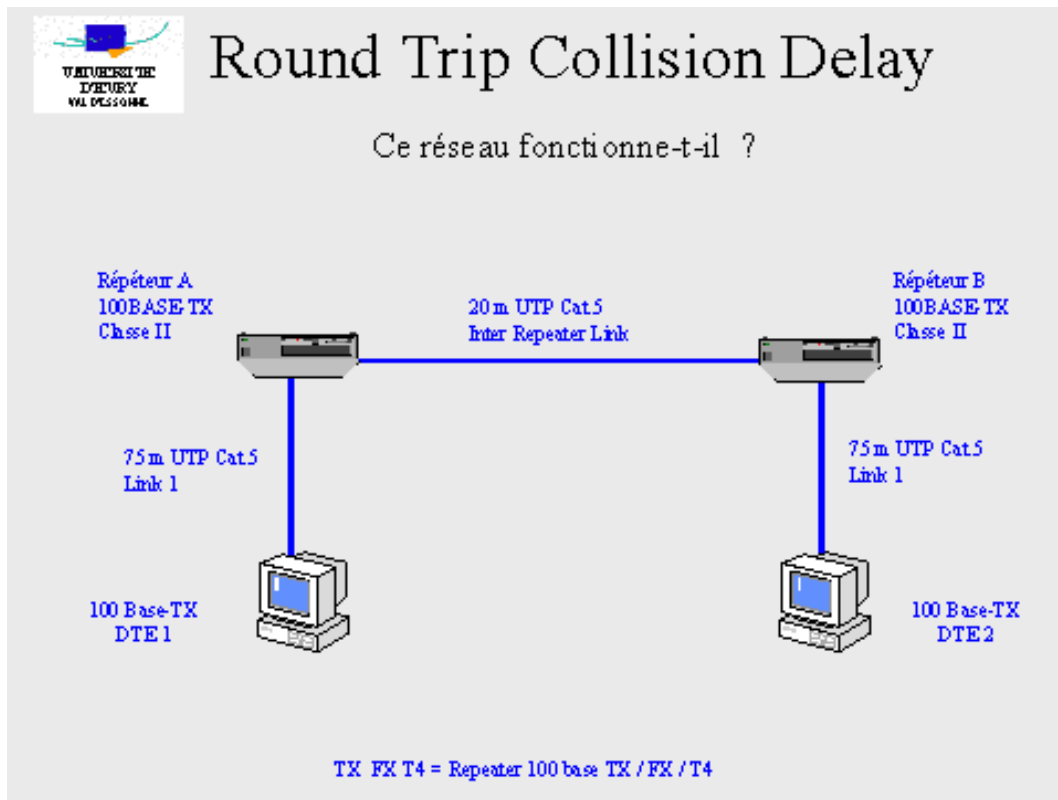
### La limitation est fonction du type de répéteur utilisé

Les deux répéteurs de Classe II sont séparés de 20 m au lieu des 5 m basiques.

Considérons que le DTE 1 débute sa transmission par une trame de taille minimum soit 64 octets (soit 512 bits).

Le DTE2 manque de justesse l'écoute du signal du DTE1 et débute sa transmission également.

La collision surviendra du côté droit du réseau et doit le traverser en sens inverse pour atteindre le DTE, cet événement doit se produire dans les 512 bt sinon le DTE1 ayant terminé sa transmission aura arrêté de transmettre quand il sera informé d'une collision et ne déduira pas que c'est sa trame qui a été endommagée par la collision.



## Le Gigabit Ethernet

Généralités - Gestion des transmissions en modes full et half duplex - GMII (Gigabit Media Independent Interface) - Solutions normalisées - Répéteurs et Hubs - Routage en Gigabit - Le mode commuté

### Généralités

C'est une évolution du standard Ethernet. Des améliorations ont été apportées par rapport au Fast Ethernet à 100 Mbps.

#### Backbone de second niveau

Il peut concurrencer la technologie ATM sur certains segments de marché (ATM sur LAN).

#### Les secteurs d'application du Gigabit:

L'imagerie, l'édition vidéo, le multimédia.

#### Points forts

- Il fonctionne sur du cuivre ou de la fibre optique
- Il est une extension à la technologie Ethernet 802.3 à 10 ou 100 Mbps.
- Il restera compatible avec des millions de noeuds Ethernet installés. Soit 120 Millions de noeuds sont

installés avec une progression de 30 millions d'unités par an (étude IDC).

- Son fonctionnement sera Half ou Full Duplex

## Une reconnaissance de multiple schémas d'encodages

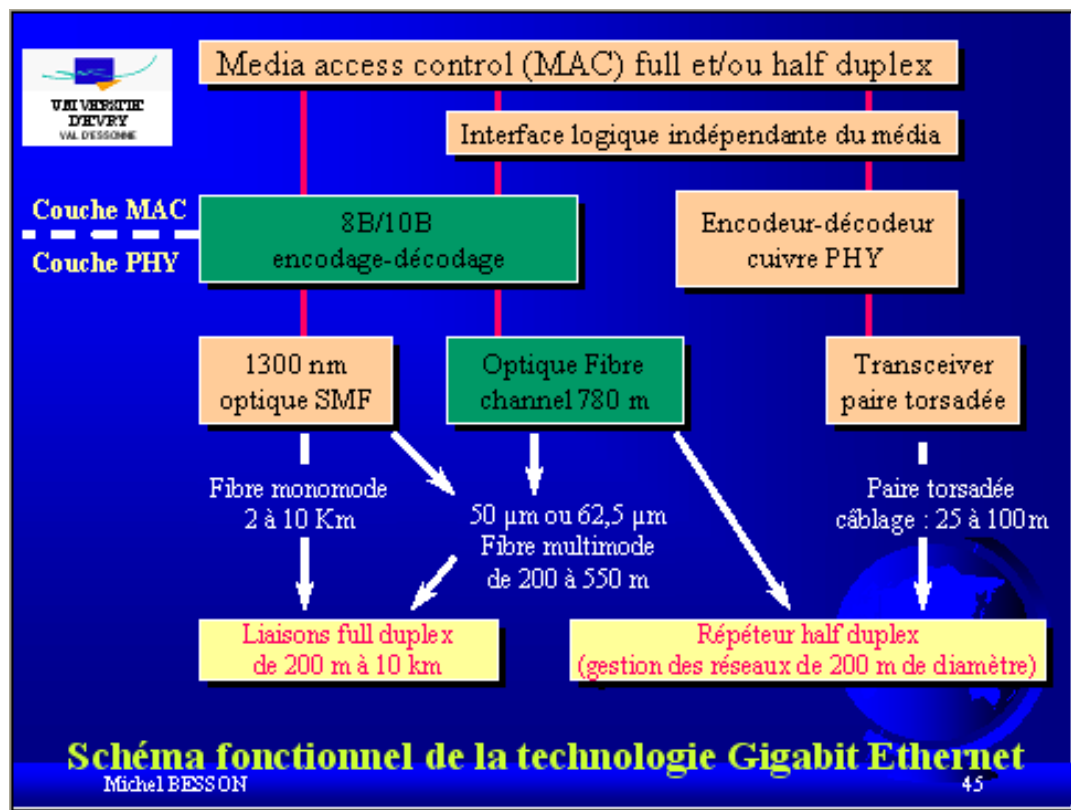
Couche FC-0 de l'ANSI X.3T11 du Fibre channel.

Cette couche physique définit les caractéristiques du média et de l'interface.

**Sont concernés :**

câbles, connecteurs, drivers, transmetteurs et récepteurs.

vitesse et distance pour chaque média.



Au dessus se situe le « **serialiser/deserialiser** »

de multiples schémas d'encodages sont reconnus dont le 8B/10B (spec. du Fibre Channel).

un serialiser/deserialiser fournira également un mécanisme de fonctionnement pour la paire torsadée adaptée à la couche physique du Gigabit.

### Encodage

La couche d'encodage 8B/10B reprend la couche FC-1 du Fibre Channel.

Elle décrit la synchronisation des octets et le schéma d'encodage et de décodage 8B/10B, un octet est transmis comme un groupe codé de 10 bits, (diminution du prix des composants).

## Gestion des transmissions en modes full et half duplex

**Au niveau de la couche MAC**

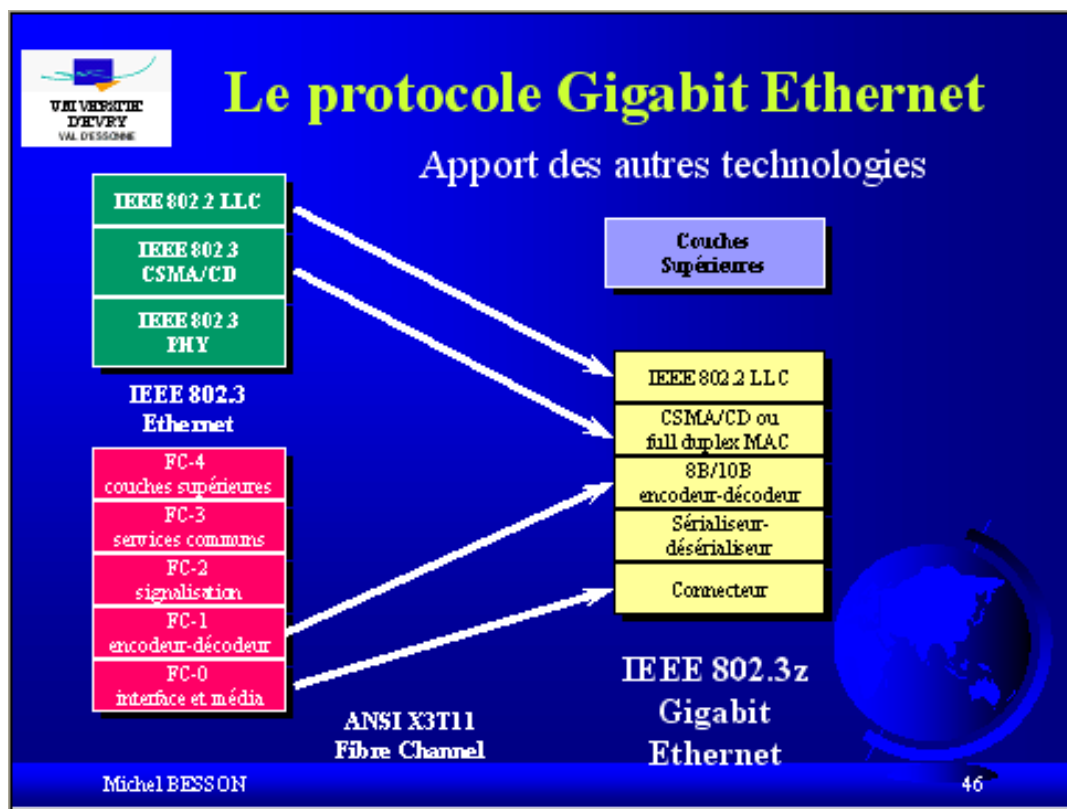
Le Gigabit gère les transmissions en half et full duplex

Full duplex

La couche MAC s'appuie sur la spécification IEEE 802.3x en incluant le contrôle de flux de trames. La focalisation actuelle de la normalisation porte sur cette méthode. Le débit sera donc de 2 Gbps (deux directions simultanées sur une même connexion)

Half duplex

La couche MAC reprendra la méthode CSMA/CD dans sa version IEEE 802.3 classique.

**Au dessus de la couche MAC**

Au dessus de MAC le Gigabit respecte tout sans changement. Il reste compatible à la norme IEEE 802.2 LLC et Ethernet. Les protocoles IPX/SPX, TCP/IP sont donc admis.

**Full Duplex**

Le Full Duplex limité aux connexions point à point. La méthode CSMA/CD est écartée dans ce cas. Le full duplex sera mis en œuvre entre un poste de travail et un commutateur, deux commutateurs, deux postes de travail. Hubs et répéteurs utilisant des ports partagés ne seront pas concernés par le Full Duplex.

**Le mécanisme de contrôle de flux IEEE 802.3z**

Il sera disponible pour des transmissions en Full Duplex, il fonctionne comme le XON/XOFF. Le récepteur peut envoyer un paquet à la station émettrice pour obtenir l'arrêt des émissions durant un temps donné, celle-ci attendra la fin de la période fixée ou la réception d'un paquet avec temps=0, à ce moment elle reprendra sa transmission.

**Technique d'accès CSMA CD**

Le 802.3z ; la technique CSMA/CD est modifiée ; en effet pour être compatible avec les déclinaisons

d'Ethernet la taille d'une trame doit se situer entre 64 et 1518 octets.

Les 64 octets, qui correspondent à 512 bits lesquels sont émis en 512 ns. Ce temps, s'il est respecté, permet à la station émettrice de ne pas se déconnecter avant de recevoir un éventuel signal de collision. La distance équivalente à ce temps est de 100 m, cependant en l'absence de hub elle est réduite à 50 m.

## Carrier Extension

### Half Duplex et CSMA/CD

Le mécanisme CSMA/CD sera mis en œuvre, les signaux ne voyageant que dans un seul sens à un instant donné. Les segments Ethernet pourront être alors partagés. Plus de deux stations partageant le même poste.

## Technique Carrier extension

Si l'on considère la réalité : avec un seul hub et les câbles établis de celui-ci aux coupleurs, la distance serait de quelques m seulement ; afin d'éviter ce problème la taille de trame a été artificiellement portée à 512 octets, l'émetteur ajoute un PAD qui est ensuite retiré par le coupleur du récepteur. On observe que même en passant de 64 à 512 octets le débit dans ce cas reste faible si la majorité des trames est en taille minimum (1/8<sup>e</sup> de la bande est utilisé).

Le full duplex reste une technique possible en Gigabit.

## Un mode optionnel

Half et Full Duplex peuvent être sélectionnés au niveau du commutateur

Ceci permet la migration de segment partagé vers des segments point à point en Full Duplex.

Néanmoins le port d'un commutateur peut être partagé en le faisant précéder par le port d'un répéteur.

## GMII (Gigabit Media Independant Interface)

- L'interface comporte un chemin de donnée sur 8 bit
- Les émetteurs récepteur doivent travailler à une fréquence de 125 MHz
- Le Fiber channel fournit le codage
- Un seul type de répéteur
- Les technique d'auto-négociation sont conformes à la FO

## Solutions normalisées

1000baseCX	avec 2 TP (Twisted Pair) de 150 Ohms
1000baseLX	une paire de FO (longueur d'onde élevée)
1000baseSX	une paire de FO (longueur d'onde courte)
1000baseT	avec 4 TP (Twisted Pair) Cat 5 UTP

## Répéteurs et Hubs

Il sont utilisables pour assurer la couverture d'un réseau en étoile. De manière classique le message entrant est recopié sur toutes les lignes de sortie. Les différentes solution offerte par la norme peuvent être interconnectées par le biais d'un hub ou d'un répéteur.

## Routage en Gigabit

Les routeur Gigabit existent (sous IP par exemple) le paquet IP extrait de la trame Ethernet est intercepté par le routeur avant de retransmettre de nouveau sur une trame Ethernet.

## Le mode commuté

Le mode commuté peut être utilisé en Gigabit ; il utilise une configuration dite Full Duplex. Ce système autorise une généralisation des interconnexions qui accepte aussi bien le Gigabit, le Fast Ethernet que l'Ethernet classique.

## Ethernet et la Commutation

### Full Duplex Switched Ethernet

#### Full Duplex Switched Ethernet

Historique : La première solution a été de découper les réseaux Ethernet en sous réseaux en utilisant des pont afin de les relier entre eux. Il s'agissait de réaliser de la contention en confinant au maximum le trafic dans un espace local. Le pont agit comme un commutateur en réalisant du store and forward chaque fois que le destinataire n'appartient pas au réseau d'origine de la trame.

#### Problématique

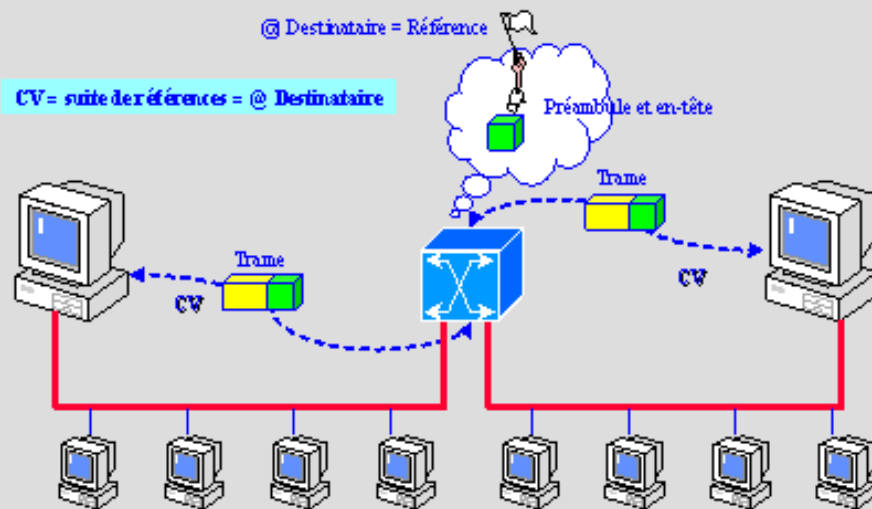
En commutation Ethernet chaque commutateur est le point central de jonction d'une carte coupleur Ethernet. Le commutateur aura pour rôle d'acheminer les trames dans la direction correcte. La notion de paquet d'ouverture de route balisant celle-ci par des références (Ex : MPLS) n'existe pas, donc la commutation via un commutateur qui exige une référence est, à priori, impossible en Ethernet.

La solution de commutation existe néanmoins si l'on considère que l'adresse du destinataire sur 6 octets peut être interprétée comme une référence. Le CV est basée sur une suite de ce type de référence, chaque commutateur devra interpréter cette référence pour trouver le bon chemin ou lien de sortie conduisant les trames depuis l'émetteur jusqu'au destinataire.

#### Obligations

- Gérer les congestions internes au commutateur
- Gérer les adresses de l'ensemble des coupleurs
- Utiliser des

# La Commutation Dynamique



**Difficultés :** Gestion des @ de tous les coupleurs raccordés au réseau  
Gestion des congestions possibles au sein du commutateur  
Liaison obligatoire de bout en bout pour assurer la continuité dans la prise en compte des références

© M. BESSON

techniques de contrôle spécifiques inter commutateurs

- Utiliser des techniques de contrôle spécifiques au trafics d'origine coupleur.

## Améliorations

- Inutilité de technique de contrôle de collision
- Disparition des limites de distance

## Avantages et inconvénients de l'Ethernet Commuté

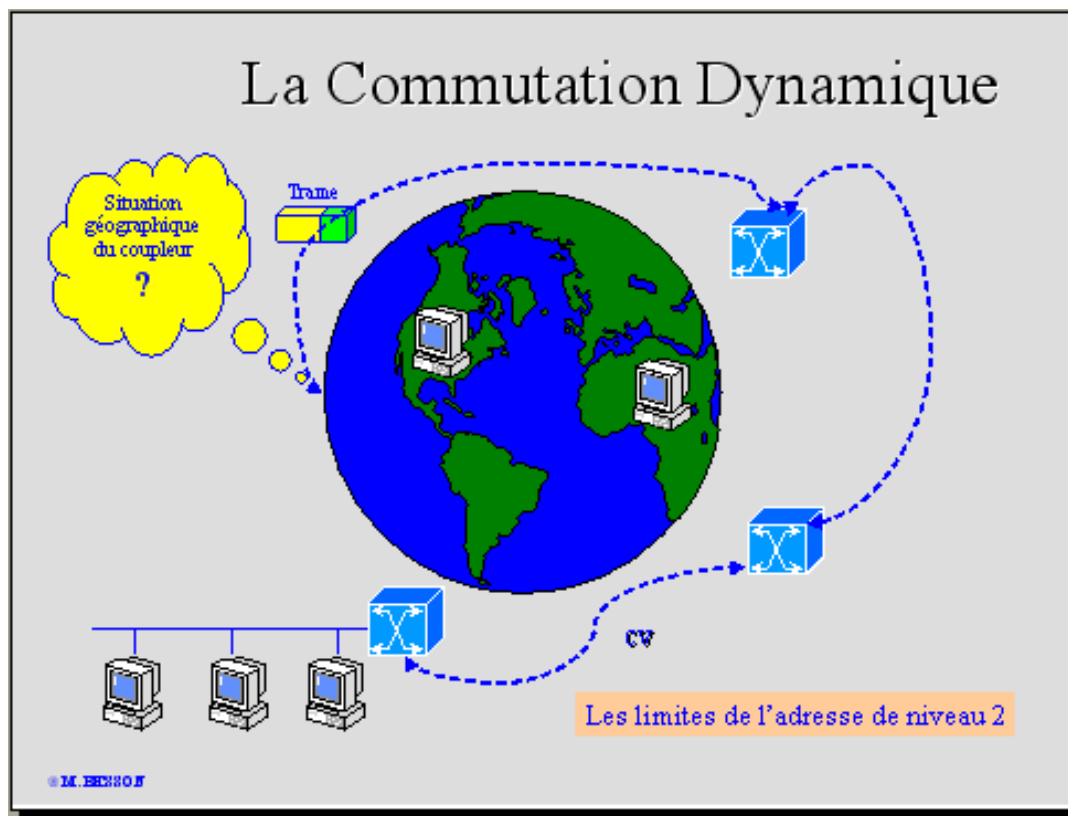
La simplicité de mise en œuvre est flagrante s'il s'agit de réseaux de taille raisonnable.

L'Ethernet commuté intègre donc facilement, grâce à une compatibilité totale, de tous les environnements Ethernet. Les trames Ethernet encapsulent les paquets ou datagrammes d'autres protocoles de niveaux supérieurs ce qui permettrait de faciliter le transfert de messages entre réseaux.

Cependant l'adressage de niveau trame n'a rien de hiérarchique (c'est un adressage dit Plat), Pour cette raison, la mise à jour de tables de routage est quasiment impossible dans un inter-réseau doté d'un nombre important de machines.

Le partage d'un même lien physique ou segment par

Michel Besson



tous les coupleurs et la baisse de performance qui en résulte, peut être contournée par une augmentation de débit (chaque machine communicante disposant d'un débit pouvant atteindre 1 Gbps), mais la difficulté liée aux congestions dans le réseau de commutation et à la connaissance de l'adresse de chaque coupleur connecté demeure.

## Deux solutions pour l'entreprise

Si le réseau est trop important pour autoriser une répartition entre réseaux partagés et réseau commutés, deux solutions peuvent alors être utilisées : les VLANs et la commutation de niveau 3.

## Les modes de commutations

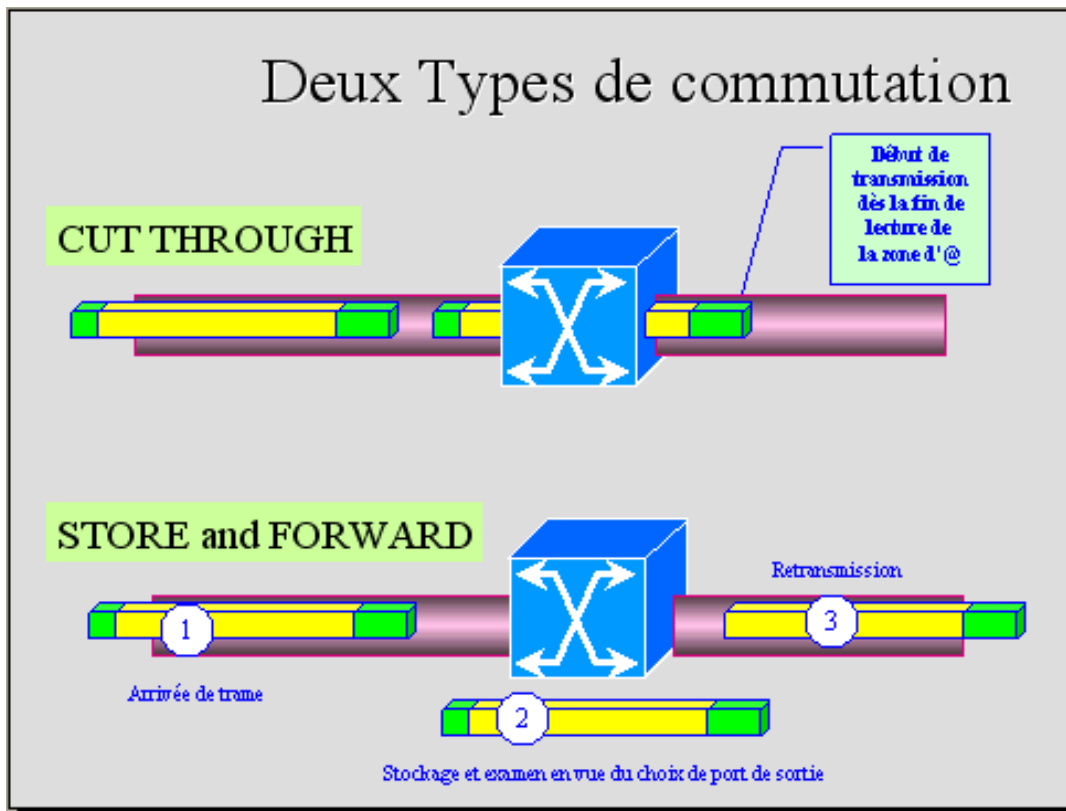
### Modes de raccordement

- La commutation par port : les coupleurs
- La commutation par segment : des segments de réseaux entiers sont directement raccordés au commutateur.

### Modes de traitement des paquets dans le commutateur

- Le Store and Forward : le paquet est stocké en mémoire, examiné et retransmis par un port de sortie.
- Le Cut Through (Fast





Forward) : Le paquet commence, sans stockage préalable, à être retransmis dès que l'adresse de destination est lue. Le paquet peut, cependant, être abîmé par exemple suite à une collision sur le réseau d'origine.

- L'Adaptative Error Free : Afin de contourner l'éventuel envoi de trame erronées on emploiera une technique appelée Adaptative Error Free ; les trames sont commutées en Cut Through avec vérification au vol de la zone de CRC. En cas de détection d'erreurs successives le commutateur repasse en mode Store and Forward.

## Les Virtual LANs

[Le concept de VLAN](#) - [Les types de VLAN](#) - [Le contrôle de flux](#)

### Le concept de VLAN

Objectif : faciliter la configuration et l'administration de réseaux très étendus et segmentés par des ponts.

### Les Stratégies

Les utilisateurs peuvent être le critère principal de la stratégie appliquée, mais la situation géographique du réseau peut également l'être. Le VLAN peut être considéré comme un domaine de broadcast, toutes les machines pouvant dans cet espace être sollicitées par la diffusion. S'il existe plusieurs VLAN on pourra mettre en place une politique de sécurité afin de filtrer les communication transitant entre eux.

## Les types de VLAN

### Les VLAN de niveau physique

Ces VLAN de niveau 1 regroupent toutes les machines appartenant à des réseaux physiques identiques ou bien différents, sous réserve d'une gestion commune des adresses.

### Les VLAN de niveau MAC

Ces VLAN de niveau 2 sont fondés sur des adresses MAC des machines, Ces machines peuvent être physiquement réparties dans des lieux différents, une machine peut appartenir simultanément à plusieurs VLAN.

### Les VLAN de niveau Paquet

Ces VLAN de niveau 3 sont à base de machines regroupées en fonction de leur adresses de niveau 3 (IP ou masque IP). ARP sert de lien entre l'adresse MAC et celle-ci.

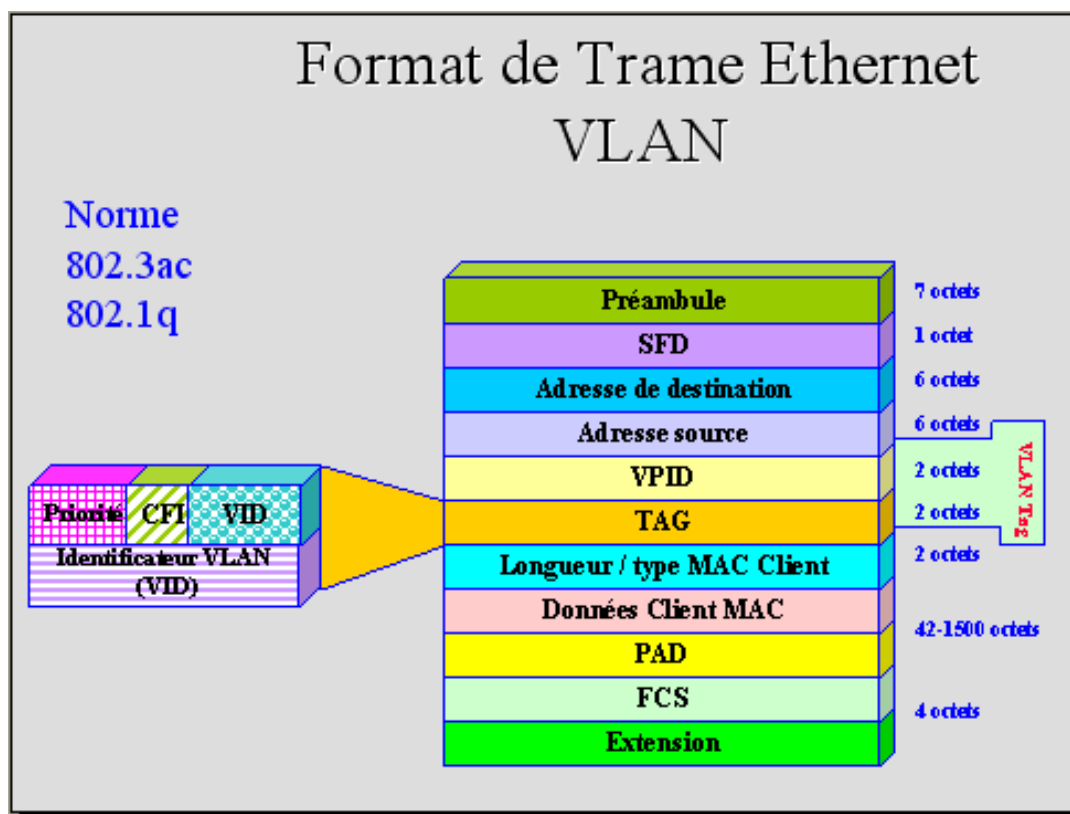
#### Une adresse complétée

L'adresse de la machine qu'elle soit de niveau MAC ou autre, doit être complétée afin de cibler son ou ses VLAN d'appartenance.

### La norme VLAN Tagging IEEE 802.1q

Un identificateur de VLAN existe donc, il se présente sur 4 octets, il est positionné entre le champs Length et la Source Address dans une trame MAC Ethernet. Cette insertion fait passer la longueur de trame de 1518 à 1522 octets. . Le format est décrit dans la norme 802.3ac et 802.1q, il présente les champs suivants:

**VPID** (VLAN Protocol Identifier) : la valeur 0x81-00 indique la présence du champ TCI.



**TCI** (Tag Control Information) : ce champ se décompose lui-même en trois autres champs :

- **Priorité** sur 3 bits : 8 niveaux de priorité, particulièrement utilisé en multimédia, il est décrit par la norme 802.1p
- **CFI** (Canonical Format Indicator) sur 1 bit : utilisé (valeur 1) dans les encapsulations de trames token ring.

**VID** (VLAN Identifier) sur 12 bits : indique l'adresse du VLAN

## Le contrôle de flux

Afin d'empêcher une accumulation des paquets de niveau MAC dans les commutateurs, un contrôle de flux sera mis en place. Une trame PAUSE à été prévue pour assurer ce contrôle de type Back Pressure. L'alerte de congestion remontera de nœud en nœud jusqu'à l'origine du flux. La requête concerne une demande d'arrêt d'émission pendant un temps plus ou moins long en fonction de la gravité du problème. Selon la durée d'interruption précisée dans la requête le nœud décidera ou non de propager celle-ci vers les nœuds en amont.

# Le niveau MAC : le jeton sur anneau (Token Ring)

Sommaire :

[Introduction](#)

[Principes](#)

[Mécanismes du 802.5](#)

[Format des trames du 802.5](#)

[Caractéristiques des adresses](#)

[Gestion des priorités et du jeton 802.5](#)

[Scénario](#)

[Temporisateurs en 802.5](#)

[Drapeaux en 802.5](#)

[Automate de transmission](#)

[Protocole SMT](#)

[Service MAC en 802.5 mis à disposition de LLC](#)

[Service MAC pour l'entité SMT](#)

[Composants de la couche physique](#)

[Traitement des fautes](#)

## Introduction

Le but est de régler les conflits qui se produiraient si on laissait tous les abonnés accéder en même temps

## Principe

Cette technique est utilisée dans anneau physique :

- Le chemin suivi par le jeton est un anneau physique
- Une station est reliée à une autre par une voie point à point monodirectionnelle


Cette technique est connue sous le nom de TOKEN RING

Elle est retenue à l'origine par IBM, le réseau est en fait étoilé ( par concentrateurs MAU ou Médium Access Unit

).

**Le réseau est doté d'un noeud central assurant :**

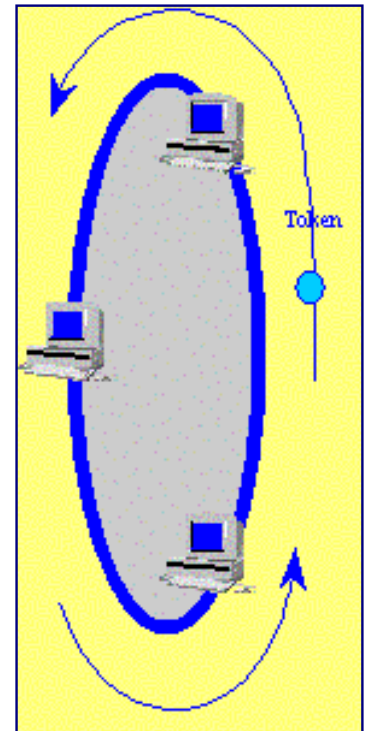
- La surveillance des stations (présence ou absence)
- La fermeture auto du circuit si absence constatée (continuité)
- La détection message ayant + 1 tour
- La régénération du jeton



## Evolution d'un Réseau Token Ring

	IBM Token Ring	IEEE 802.5
Topologie	Etoile	Nonspécifié
Longueur maximum du segment en mètres	En fonction du type de câble, nombre de MAUs, etc.	En fonction du type de câble, nombre de MAUs, etc.
Nombre maximum d'attachements par segments	260 en STP, 72 en UTP	250
Diamètre maximum du réseau	En fonction du type de câble, nombre de MAUs, etc.	En fonction du type de câble, nombre de MAUs, etc.

Voir Tableaux du cours sur la couche 1 Physique



Standardisé initialement par IBM, Token Ring a été normalisé par l'IEEE : Norme IEEE 802.5

Ce tableau présente les quelques caractéristiques principales à prendre en compte.

Voir normes physique

dessin selon dispo 5

# Mécanismes du 802.5

## Une station a le jeton

- Elle émet vers successeur

Si le successeur est destinataire = prise de copie

- Le successeur transmet a son suivant et ainsi de suite
- La station origine lit son message (1 tour a été réalisé)

Elle retire celui ci de la propagation  
Elle passe le jeton

## Des priorités peuvent être affectées aux stations

Si peu de priorité : une stations ne pourra émettre que tous les "n" passage du jeton  
Le jeton comporte donc une indication de priorité

## Principes :

- Si la stations a la priorité correcte:

Elle émet  
Sinon elle passe le jeton

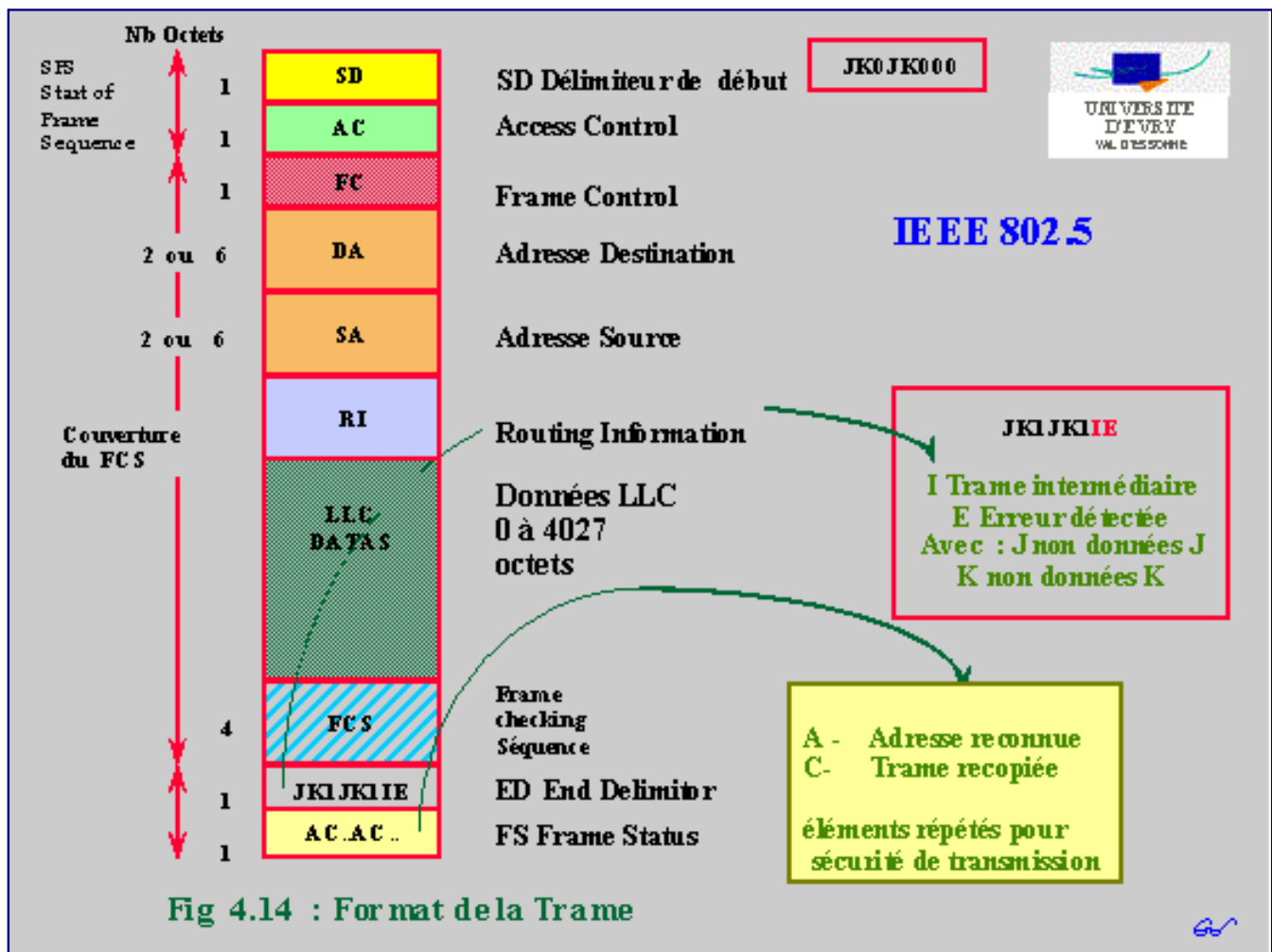
- Si le message est urgent

La station relaie la priorité qu'elle désire, place la valeur au dessus de la priorité courante, le prochain jeton aura cette priorité.

## Autres caractéristiques :

- Toutes les trames portent le jeton
- L'anneau est actif en permanence
- Il n'y a pas de silence inter message
- En absence de trames: des signaux sont produits en permanence. Il n'y a pas de préambule, cette émission continue joue le rôle d'amortisseur

# Format des trames 802.5



## Descriptif des champs de la trame

- **SD** Starting Delimitor

Il permet la reconnaissance du début de Trame (Cche Physique)  
Des codes JK sont utilisés ( voir Manchester) soit: 2 temps bit sans transit.

- **AC** Access Control

gestion du jeton  
gestion des priorités

- **FC** Frame Control

Il définit le type de trame : les deux 1er bits caractérisent une trame de contrôle (trame MAC) ou bien une trame LLC (données)

- S'il s'agit d'une Trame . LLC :

3 bits sont nuls  
3 bits PPP caractérisent les priorités

- S'il s'agit d'une Trame. MAC:

Les bits servent à coder les fonctions anneau

- DA Adresses du destinataire
- SA Adresses source
- FCS Frame Checking Sequence : Contrôle d'intégrité
- ED End Delimitor : permet à la couche PHY de reconnaître la fin d'une trame

comporte des codes non data : JK

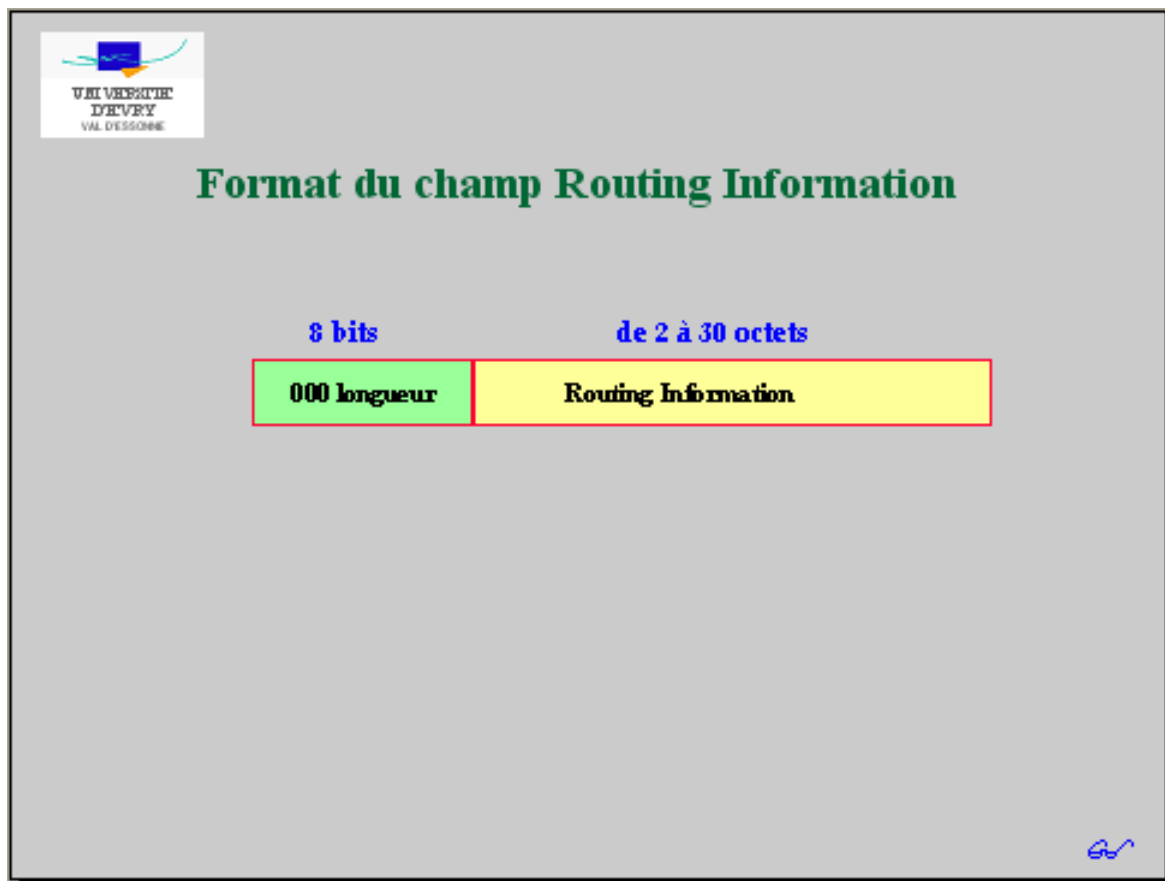
bit I = trame suivie --> par trame de source identique  
 bit E = 1, si erreur CRC détectée  
 1 = élément binaire 1

- FS Frame Status : il comporte 2 quadruplets ; un récepteur peut positionner:

A=1 si la station reconnaît son adresse  
 C permet signaler une copie correcte de la trame par le destinataire  
 Les deux derniers sont non utilisés

- Nota : Un émetteur peut arrêter son Emission par une séquence d'abandon:

2 octets SD-ED ... Start et End Delimiters



**Champ d'information de routage (RI - Routing Information)**

Absent des spécifications de 1985

Objectif

Permettre le routage de la trame par la station source, notamment lorsque la trame est appelée à traverser une succession de ponts.

### Longueur de l'information de routage

Sa longueur est variable ( comprise entre 2 et 30 octets), Cinq des bits du premier octet permettent de la représenter.





## FORMAT DU CHAMP ACCESS CONTROL

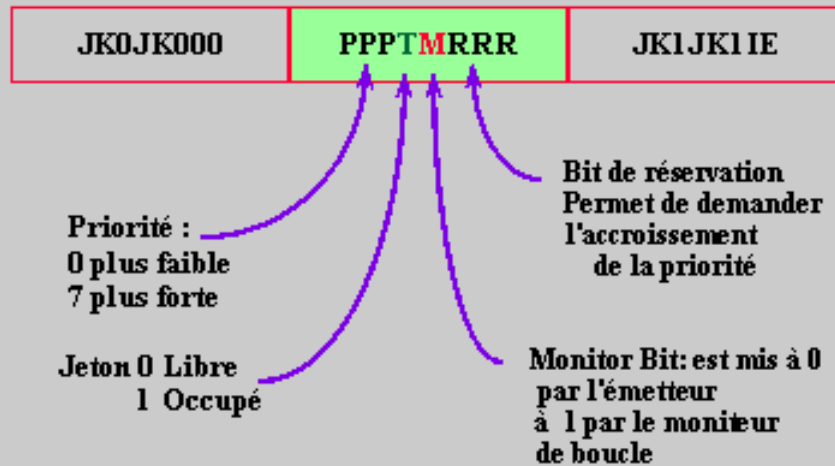


Fig 4.15 Format du Champ Contrôle d'accès et du Jeton libre

### Détail du champ AC Access Control

bit T=Token.  
le bit M est mis  
à 0 par  
l'émetteur  
le bit M est  
passé de 0-->1  
par le moniteur,  
la Trame sera  
mise au rebut  
dès qu'il la  
reverra.

bits P = priorité  
en cours  
bits R =  
réservation de  
priorité  
P=R lors de la  
régénération

### bits FF du champ FC

## Format du champ Frame Control

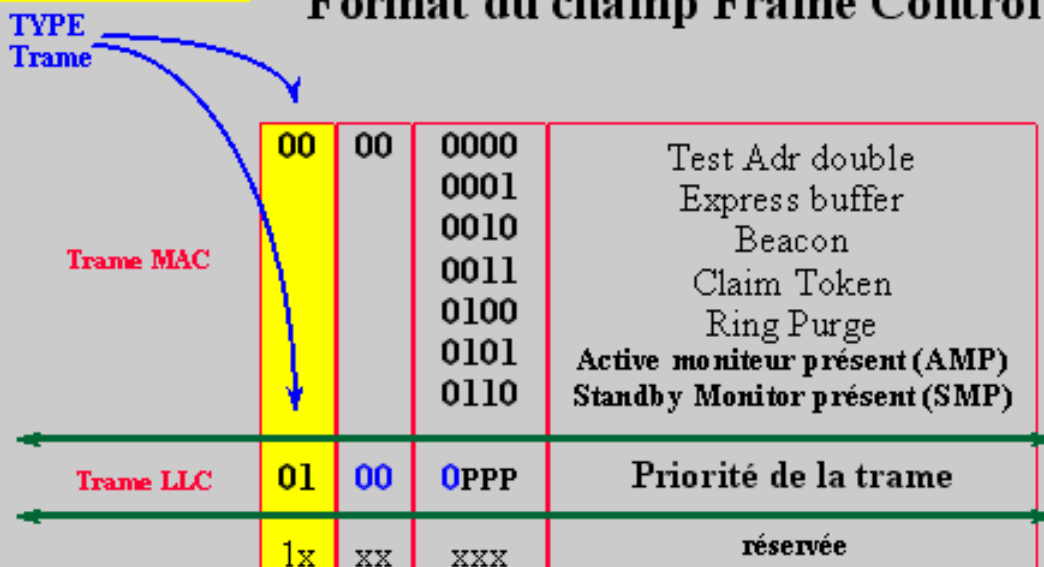


Figure 4.16 : Format du Champ FRAME CONTROL

### Détail du champ FC Frame Control

Il définit le type  
de trame :

Les deux 1er bits  
signalent trame  
contrôle (MAC)  
ou une trame LLC  
(data).

Si Trame . LLC :

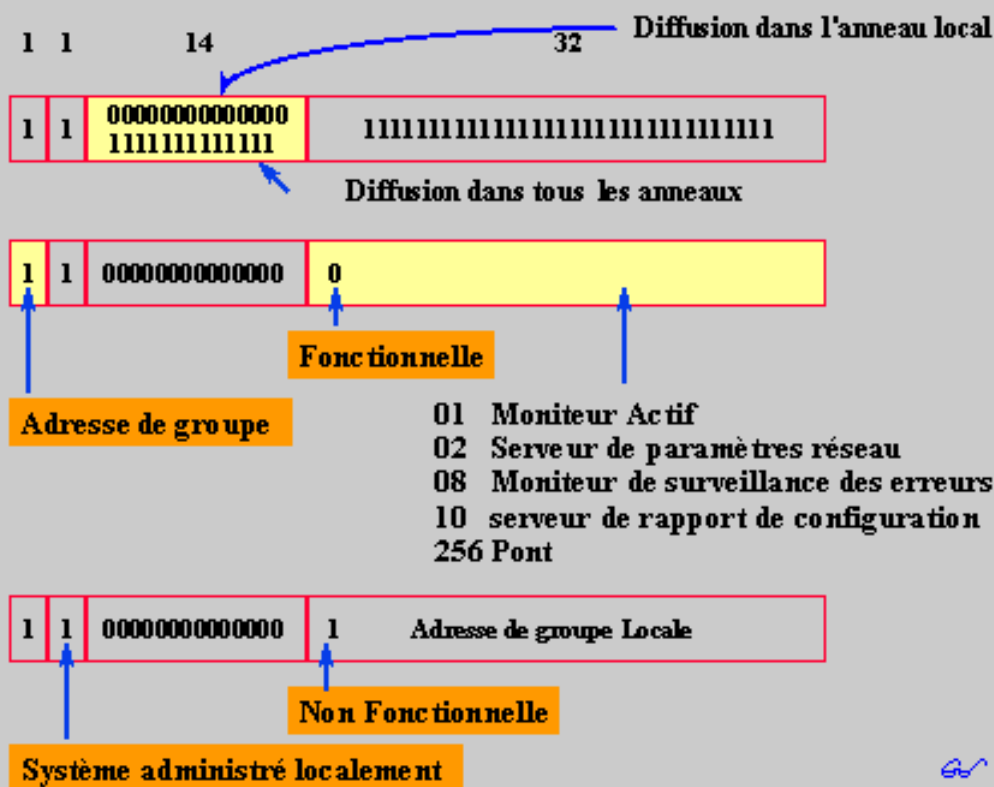
3 bits sont nuls  
et 3 bits  
désignent la  
priorité PPP

Si Trame. MAC:

Les bits servent  
à coder les  
fonctions de  
l'anneau.

## Caractéristiques des adresses

### F 4.9 Diffusion des adresses dans les boucles



**Boucle : anneau à jeton IBM**

Le champ des 48 bits ou 16 bits a été décomposé:

- Les deux 1er octets si adresse longue

- Le 1er octet si adresse courte

Désignation de l'anneau

- les 1er bits conservent leur signification.

### Décomposition d'une adresse longue :

14 bits : N° anneau ;

- si des 0 dans le champ la diffusion est restreinte à anneau local
- si des 1 dans le champ la diffusion est permise dans tous les anneaux.

32 bits = Adresse station

Une notion d'adresse fonctionnelle peut être ajoutée dans la partie adresse station

- 1er bit=1..... adresse normale
- 1er bit=0 .....adresse fonctionnelle

Désignation en hexa différentes adresses fonctionnelles :

- 01 Moniteur Actif
- 02 Serveur de paramètres réseau

08 Moniteur de surveillance des erreurs  
10 serveur de rapport de configuration  
256 Pont

## Gestion des priorités et du Jeton 802.5

### Principe

Une trame passe Jeton libre, une station la capture

Entre la capture et l'émission une durée  $>$  temps bit s'écoule, la station maintient son émission.

### Complication par mécanisme de Priorité

Le champ de réservation "Priority Réserve":PR

Une Station qui désire un jeton de Priorité = AP tente de positionner le champ PR

Si  $AP > PR$  elle se positionne sinon elle abandonne.

Sachant que PI est la priorité indiquée trois cas peuvent se présenter :

**Cas 1 :** jeton libre , si  $PI \leq AP$  la station capture le jeton, dès transmission de sa trame le jeton aura priorité = AP ; PI sera mémorisé

**Cas 2 :** jeton libre ou non si  $PI \geq AP$ , la station passe le jeton

**Cas 3 :** jeton libre ou non si  $PR < AP$  alors on fait  $PR = AP$ , on indique ainsi la priorité souhaitée pour le prochain jeton.

### Corollaire

La station qui a le jeton libère celui ci si  $AP$  de sa prochaine Trame  $<$  PI courant

Pour Mémoire : PI= AP de la Trame émise avant. La PI retransmise sera = à la valeur maxi: soit de la PR reçue, soit de la PI avant capture (mémorisée).

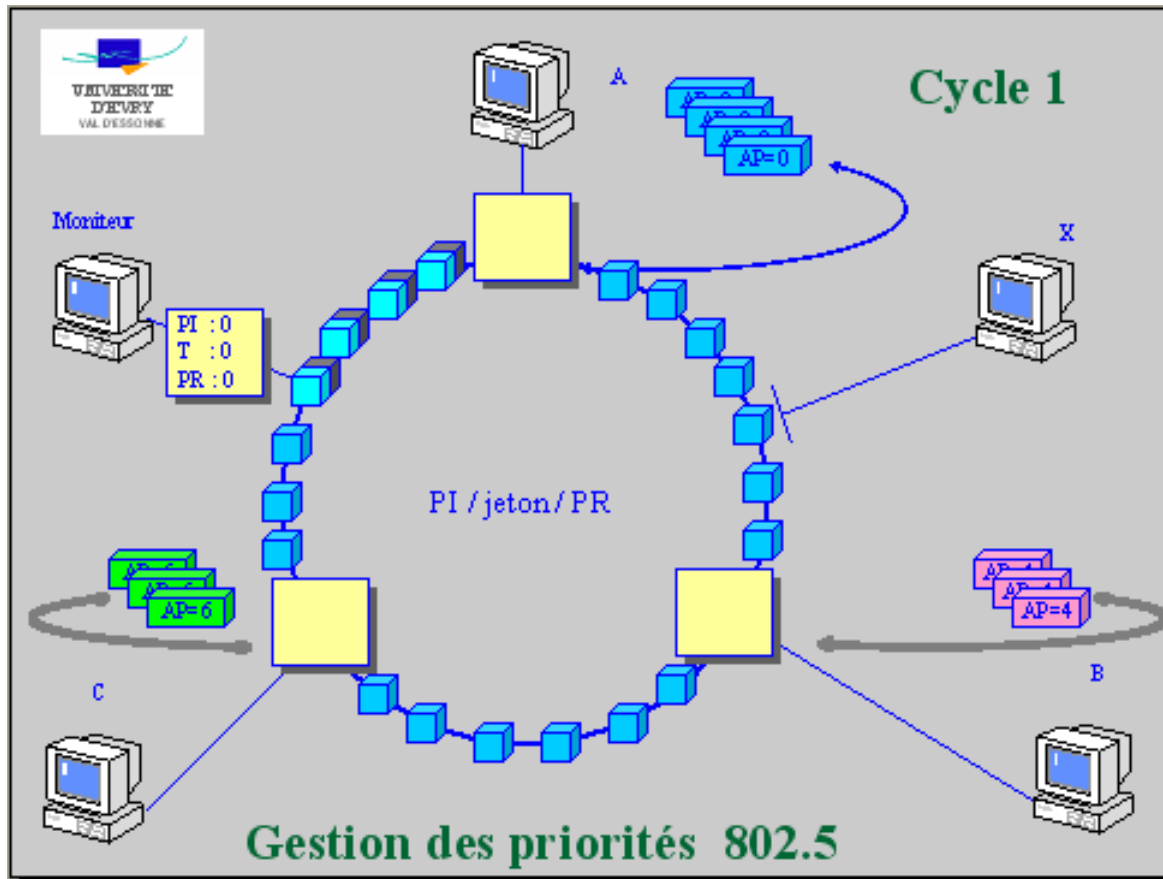
# Scénario

Notation: **PI** (priorité indiquée)/ **jeton** / **PR** (priorité de réservation).

pour mémoire : jeton libre=0 jeton occupé=1

**Début :**

- Arrivée en A d'un jeton libre 0/0/0
- A transmet son message.
- Arrivée en B jeton occupé 0/1/0
- B passe avec PR = 4 inscrit,
- Arrivée en C,
- C passe avec PR = 6 inscrit,
- A retire son jeton , en génère un avec PI=6, mémorise la PI,
- Arrivée en B qui ne peut l'utiliser, mais remet la PR = 4
- C prend le jeton et commence à émettre, A passe, B passe.
- Fin des transmissions de C, il libère jeton avec PI = 6 la PR de B est conservée.
- A voit jeton dont PI = 6 = PI mémo, il fait le mise à jour PI = PR soit PI = 4, la PI sera égale au maximum constaté entre l'ancienne priorité = 0 et la PR = 4.
- Il génère 1 jeton=0 avec PR=0 et passe.
- La trame arrive à B qui la voit , la capture et émet.....un tour complet est fait.
- B libère dès transmissions un jeton de priorité PI=4.
- A voit la priorité



qu'il avait émise la retire et régénère un jeton PI= 0.

Nota : Seule une station qui a Monté le jeton en PRIORITE est habilitée à le Descendre

But : Permettre à une priorité donnée de faire un tour complet, si la station capable de descendre la priorité est absente un mécanisme de purge de valeur est prévu.

## Temporisateurs en 802.5

### TRR (Timer Return to Repeat)

Pour s'assurer que la station peut retourner à l'état de répétition  
Est supérieur au temps de latence maximum (délai de propagation sur l'anneau + temps de latence de chaque station), par défaut = 4 ms

### THT (Timer Holding Token)

Pour contrôler le temps de transmission maximal de la station détenant le jeton  
condition : la durée des trames prévues à transmettre doit être < THT, par défaut = 8,9 ms

### TQP (Timer Queue PDU)

Détermine le délai avant envoi d'une trame SMP après réception d'une trame AMP ou SMP  
Par défaut = 20 ms

### TVX (Timer Valid Transmission)

Utilisé par le moniteur actif pour détecter l'absence de transmission valide  
Par défaut = 10 ms

### TNT (Timer No Token)

Utilisé pour détecter la perte du jeton  
Par défaut = 2,6 s

### TAM (Timer Active Monitor)

Utilisé par le moniteur actif pour déterminer la période d'envoi d'une trame AMP

Par défaut = 7s

### **TSM (Timer Standby Moniteur)**

Utilisé par les moniteurs en veille pour vérifier la présence d'un moniteur actif et détecter si un jeton circule en continu. Par défaut = 15s

### **TER (Timer Error Repeat)**

sert à reporter les valeurs des compteurs d'erreurs dans les trames Report Error transmises au serveur d'erreurs. Par défaut = 2s

### **TBT (Timer Beacon Transmit)**

Définit le temps pendant lequel une station émet des trames Beacon avant de passer en état By-Pass. Par défaut = 16s.

### **TBR (Timer Beacon Receive)**

Définit le temps pendant lequel une station peut recevoir des trames Beacon de son voisin aval avant de passer en état By-Pass. Par défaut = 160ms

## **Drapeaux en 802.5**

### **I- FLAG**

positionné sur réception d'un champ ED avec le bit I mis à 0.

### **SFS- FLAG**

positionné sur réception d'une séquence SFS (Start of Frame Sequence SD+AC).

### **MA- FLAG**

positionné sur réception d'un champ SA égal à l'adresse de la station.

### **SMP- FLAG**

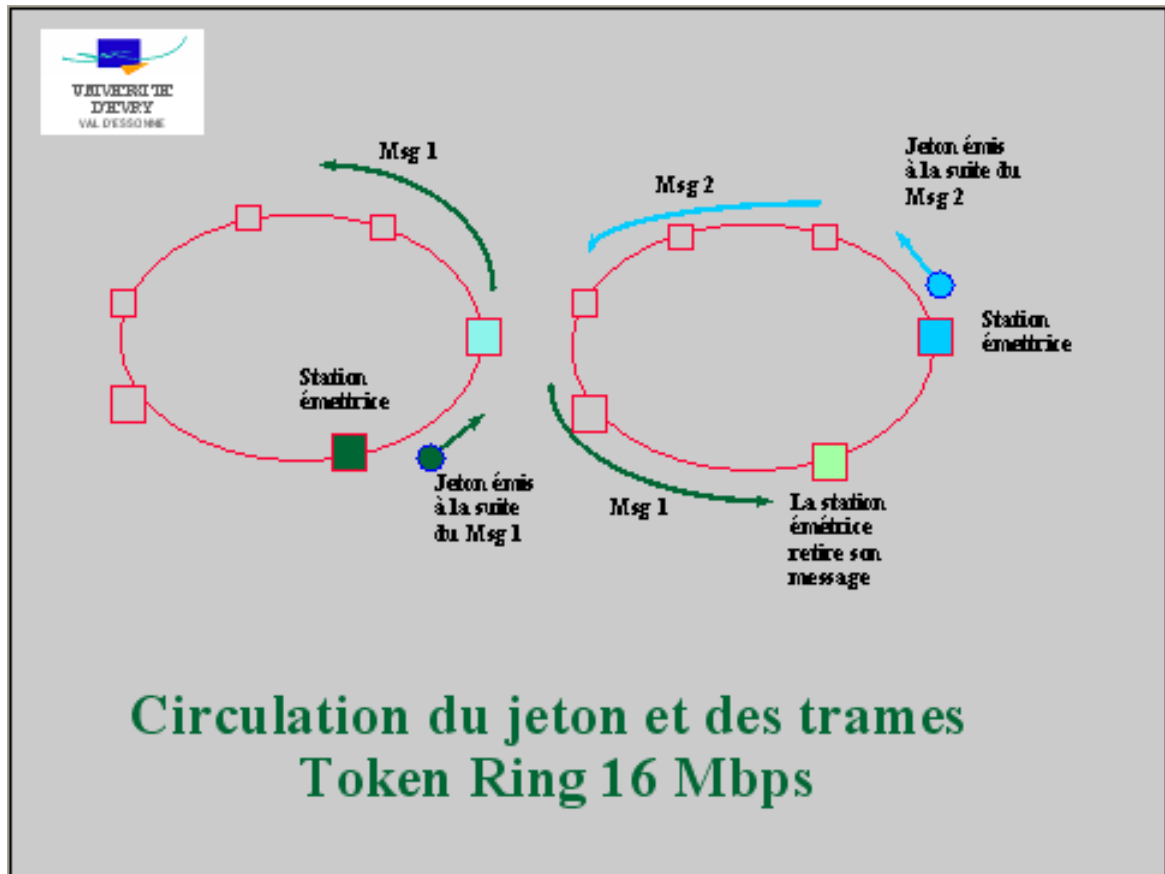
positionné par les moniteurs en veille sur réception d'une trame SMP ou AMP avec les bits A et C à 0 (process de notification complet).

## NN- FLAG

positionné par le moniteur actif sur réception d'une trame SMP ou AMP avec les bits A et C à 0 (process de notification complet).

## BR- FLAG

positionné sur réception d'une trame Beacon et remis à zéro sur réception de toute autre trame.

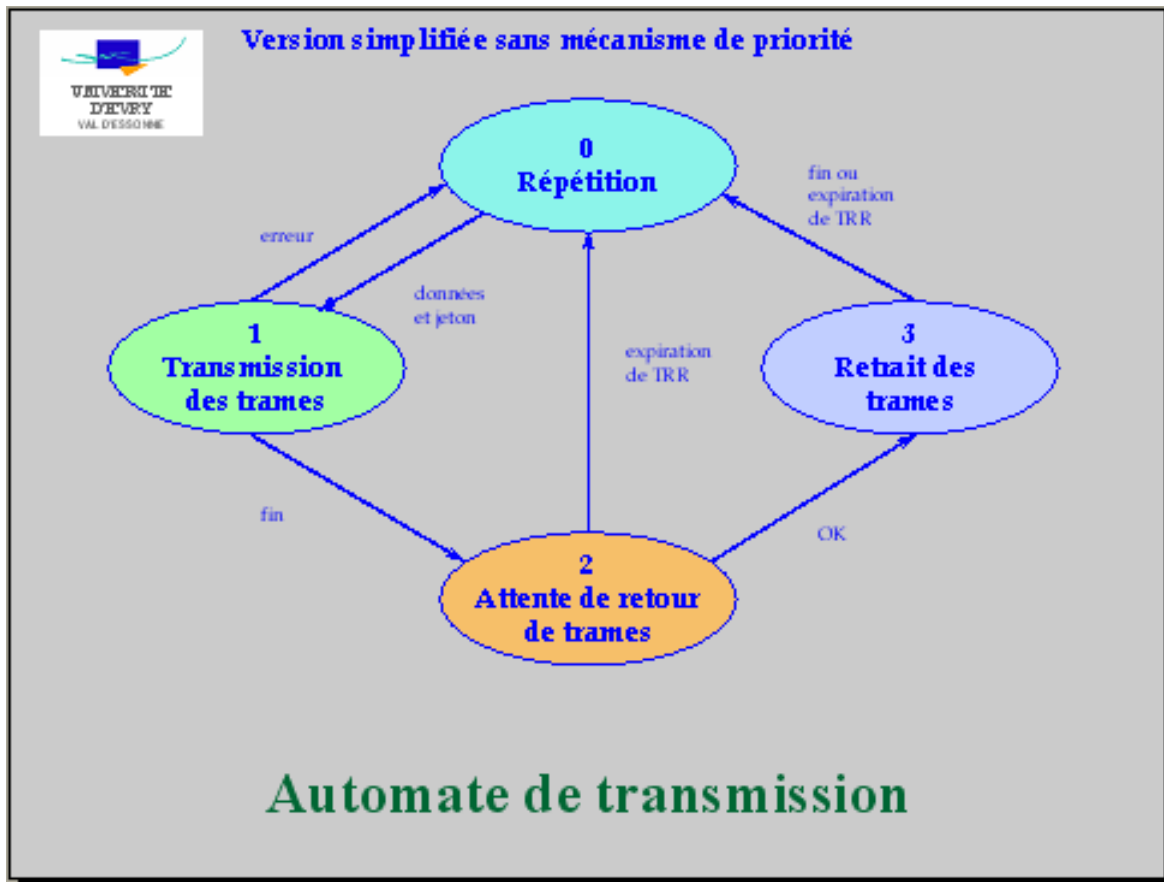


Circulation du Jeton en 16 Mbps

Dans ce cas l'émetteur n'attend pas de revoir sa tête de message pour libérer le jeton.

Celui-ci est libéré par la station 1 dès la fin de son envoi, une seconde station pourra alors le capturer et transmettre à son tour.

## Automate de transmission



Trois automates existent :

Automate de fonctionnement (operational machine)

Automate du moniteur en veille (standby monitor machine)

Automate du moniteur actif (active monitor machine)

## Les états

- Etat 0 Répétition

Pas de trame à transmettre (répétition des bits entrants), si réception d'une requête de transmission d'une trame de données ou d'une trame SMT, elle cherche à détecter le jeton.. Sur réception du jeton elle arme THT et passe à état 1

- Etat 1 Transmission de trames

La station émet toutes les trames de priorité égales ou supérieures à celles du jeton,

elle cesse à la fin des PDU prévues ou si THT expiré, elle émet ensuite une séquence de fin de trame ED/FS et passe à 2.

Après détection du jeton ( $T=0$ ), la station peut détecter des situations d'erreurs: ED absent, Trame de



reset d'anneau. Si anomalie elle retourne à l'état 0

- **Etat 2 Attente de retour de trames**

La station attend le retour de sa propre trame.

Si réception SA = son adresse, le jeton est réémis et la station passe en état 3

Si le TRR expire sans retour de trame portant son adresse

le compteur de trame perdues est incrémenté

la station retourne en 0

- **Etat 3 Retrait de trames**

la station retire les trames qu'elle avait émises jusqu'à bit I=0 (dernière) et retourne à l'état 0. Si la dernière trame n'est pas reçue, elle s'arrête lorsque le TRR expire et retourne à l'état 0.

- **Réception de trames par la station**

Répétition et vérification : en répétant les bits entrants la station vérifie si la trame contient DA=son adresse, si oui elle la copie et les bits AC sont passés à 1.

Deux types de trames peuvent lui être destinées :

- des trames de contrôle
- des trames de données

Si détection d'erreur : le bit E = 1 (champ ED) erreur détectée.

## **Protocole SMT (Station ManagemenT)**

### **Stations de gestion**

Le moniteur actif (Active Monitor) centralise la gestion.

Les autres stations sont en veille (Standby Monitor), elles sont capables de détecter à tout moment la défaillance

de l' Active Monitor. Elles prennent alors la relève du contrôle.

### Moniteur Actif (Processus d'élection et rôle)

Cette station a gagné le processus d'appel du jeton lors de l'initialisation de l'anneau

#### Processus:

- Chaque station arme son TVX, s'il expire, elles arment leur TNT (Timer No Token), s'il expire, elles peuvent transmettre une trame "Claim Token".
- Si la Claim Token fait un tour sans que la station ait reçu une "claim token" d'une autre station : la station devient le **moniteur actif (MA)**

#### Action du moniteur actif

- Le moniteur actif génère alors un nouveau jeton ; les autres seront les moniteurs en veille.
- Le moniteur actif reprend les erreurs portant sur tout ce qui circule (dont jeton), il positionne à 1 le bit M monitor sur toutes les trames. Il Réarme TVX à chaque passage de trame ou de jeton.
- L'absence de jeton est détectée par TVX (expiration), le compteur de trame en erreur est alors incrémenté.
- Une purge de l'anneau est alors effectuée. Les trames non valides (moins de 3 octets) et les trames orphelines (bit monitor déjà à 1) sont retirées de l'anneau et donnent lieu à une purge durant TRR.
- Un jeton est réémis avec priorité plus faible, pour mémoire un jeton circule de façon persistante avec une priorité donnée.

## Rôle des moniteurs en veille

#### A l'initialisation de l'anneau

- Ils vérifient l'unicité de leur adresse et que celle-ci est connue de leur voisin direct en amont.

#### A l'état de veille

- Ils vérifient qu'il y a un moniteur actif présent.
- Ils signalent leurs présence à leur voisin (Neighbor Notification).
- Ils vérifient qu'il y a un moniteur actif présent.
- Ils signalent leurs présence à leur voisin (Neighbor Notification).

## Processus Neighbor Notification

- Le moniteur actif diffuse une AMP (Active Monitor Présent).
- La station en aval effectue les opérations suivantes:
  - Armement de TSM (Timer Standby Moniteur).
  - Copie de la trame AMP et stockage de l'Ad de cette station (Amont).
  - Positionnement des bits A et C de la trame AMP (Active Monitor Present).
  - Armement du temporisateur TQP (Timer Queue PDU) et transmission de sa trame SMP (Standby Monitor Present).
  - La station voisine effectue le même travail sur réception de la trame SMP et ainsi de suite.

## Si TSM expire

- Un moniteur en veille commence à émettre une Claim Token.

## Si TNT (Timer No Token) expire

- La station transmet une trame Beacon (signale d'une panne grave).
- Le réseau sera ensuite réinitialisé.

# Service MAC en 802.5 mis à disposition de LLC

## Trois primitives de service (Norme ISO 10039)

- MA\_DATA.request

### Paramètres :

Contrôle de trame : donne la valeur de FC à utiliser pour la trame MAC.

Adresse destination : individuelle ou de groupe.

m\_sdu : données LLC à émettre.

Classe de service : priorité à utiliser pour le transfert.

- MA\_DATA.indication

### Paramètres :

Contrôle de trame : donne la valeur de FC utilisée pour la trame MAC reçue.

Adresse destination : individuelle ou de groupe.

Adresse source : identifie l'émetteur.

m\_sdu : données LLC reçues et délivrées.

Etat de réception : C.Rendu FR\_GOOD, FR.WITH.ERROR. avec raison : Valeur bit E (0,1,invalid) valeur bits A.C (00,11,10,invalid).

Classe de service fournie: priorité effectivement utilisée pour le Transfert.

- MA\_DATA.confirmation

(signification locale: indique le succès ou l'échec de l'émission)

Paramètres :

Etat de transmission : FR\_GOOD, FR\_WITH\_ERROR avec raison : valeur bit. E (0,1,invalid) valeur bits A.C (00,11,10,invalid).

Classe de service fournie : priorité effectivement utilisée pour le Transfert..

## Service MAC pour l'entité SMT

L'interface entre MAC et SMT est totalement locale à la station.

Elle est utilisée par le moniteur pour contrôle des opérations MAC d'une ou plusieurs stations .

### Primitives

MA\_INITIALIZE\_PROTOCOL.request ,MA\_INITIALIZE\_PROTOCOL.confirmation

MA\_CONTROL.request , MA\_STATUS.indication, MA\_NMT\_DATA.request

MA\_NMT\_DATA.indication, MA\_NMT\_DATA.confirmation

MA\_NMT\_DATA.request, MA\_NMT\_DATA.indication, MA\_NMT\_DATA.confirmation

Elles ont des rôles identiques aux MA\_DATA\_xxxxx, mais portent des trames de contrôle MAC et non LCC (données). Elles permettent l'envoi et la réception de trames de contrôle MAC.

## La sous couche PHY assume l'interface avec la couche MAC



Il permet de transférer le signal codé en Manchester (Bande de base)

### TCU (Trunk Coupling Unit)

Elle relie la station au support à l'aide d'un connecteur (Medium Interface Connector), celui-ci peut être éventuellement éloigné de la station

### Rôle de la TCU (ou répéteur)

Elle connecte la station au support. Ses fonctions principales sont :

- Répéter les bits entrants
- Insérer des données sur l'anneau
- Recevoir des données depuis l'anneau
- Détruire des données
- Amplifier les signaux

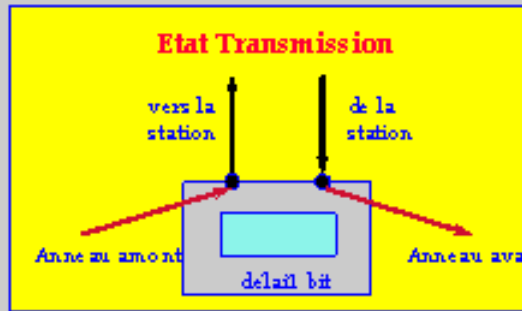
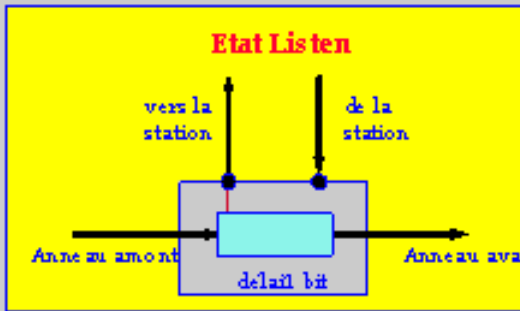
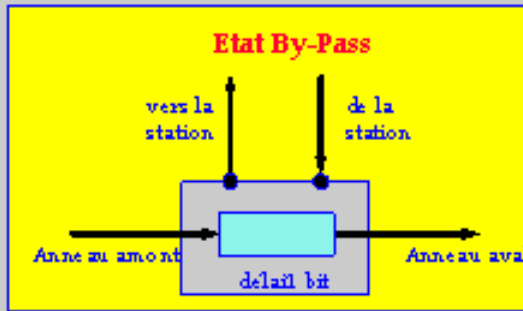
## Etats de la TCU

- **By- Pass :**

La station est inactive, la TCU répète les bits entrants de Michel Besson



# Couche Physique en 802.5



## Fonctionnement de la TCU

l'amont vers l'aval.

- **Listen :**

La station est active, la TCU copie les bits entrants vers la station et retransmet vers l'aval en simultané. Les bits peuvent être modifiés au vol.

- **Transmission**

La station transmet une trame et peut recevoir des bits depuis l'amont  
Si ces derniers appartiennent à une autre transmission : la station les stocke, le temps de sa transmission avant de les retransmettre

- **Le tampon de latence**

Il est utilisé par le moniteur actif pour deux objectifs:

### Compensation du délai de retour du jeton

Si toutes les stations sont en état de répétition le jeton ne doit pas revenir trop vite. L'anneau doit avoir une certaine latence (en temps-bit).

Un tampon de latence est inséré sur l'anneau (niveau moniteur actif) les bits entrants sont mémorisés temporairement (val moy 24 bits).

### Compensation de la gigue (variation de propagation du signal)

La synchro des stations est basée sur ce signal, selon avance ou retard. le moniteur insère ou enlève des bits de l'anneau.

# Traitement des fautes

Aperçu technique : on constate 2 types d'erreurs

## Type 1 : Erreurs Matérielles Stables

### Défaut matériel

- Emission continue

Insertion d'un flot de bits par une station, donc écrasement des trames en transit

Deux origines :

- Répétition mal effectuée
- Entrée d'une station qui croit avoir le jeton

- Perte du signal

Origine : rupture de l'anneau

### Panne de concentrateur

- Erreur de fréquence

Origine : dérives importantes des horloges de l'émetteur et du récepteur

- Panne de coupleur

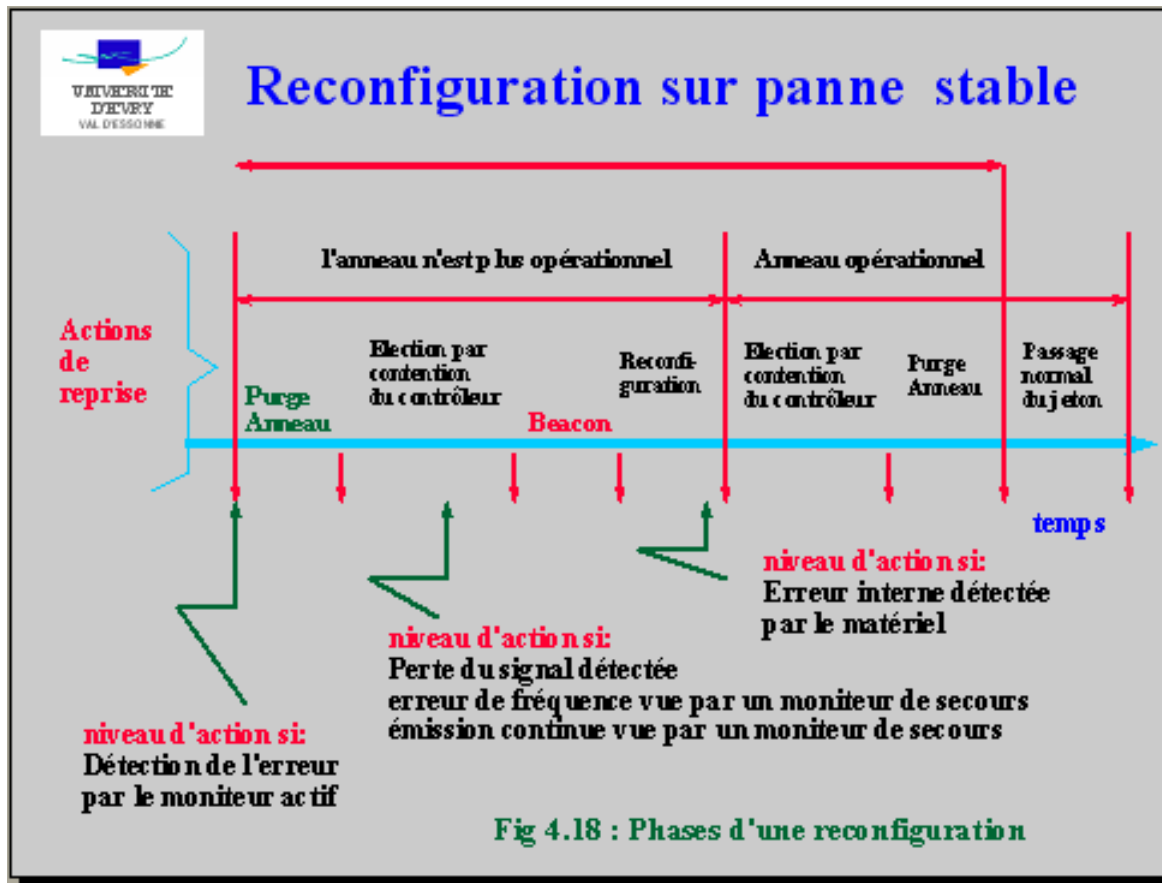
Origine : détectable par le site Hôte

### Action

- Court circuiter ou retirer l'élément fautif
- Procéder à la reconfiguration de l'anneau :

D'abord en automatique  
Sinon via maintenance

### Phases de reconfigurations sur panne stable



### Sur détection de l'erreur :

1 - Le moniteur tente purge anneau

2 - Il y a forcément échec car panne stable (non récupération)

3 - Les moniteurs de secours ( autres équipements ) :

- Déduisent présence d'une faute (en raison de la non récupération)
- Tentent la réélection d'un nouveau moniteur.

4 - Puis en raison de la présence continue d'une faute :

- Enchaînent par une séquence BEACON, déconnexion physique des stations
- Ensuite il y a test local et individuel de chaque station
- Si positif = réinsertion de la station Si négatif = appel maintenance

5 - Si fin de test positif = il y a tentative de reconfiguration :

Si réussite de la reconfiguration :

- 1 - réélection d'un contrôleur
- 2 - celui-ci purge l'anneau
- 3 - et réémet un jeton

Si échec de la reconfiguration :

Fin de tentative automatique; appel à la maintenance.

Note sur la procédure de réélection du contrôleur :

Il s'agit d'un mécanisme conflictuel, il n'y a pas de collision réelle sur anneau.

Le fait pour un émetteur de recevoir quelque chose que l'on n'a pas émis est considéré comme collision, quelque chose que l'on n'a pas émis est considéré comme collision



## Type 2 : Erreurs Transitoires Matérielles ou Logicielles

### Récupérables par les protocoles de gestion de l'anneau

- Erreur de Transmissions
  - émission continue
  - erreurs de fréquence (parasites)
  - perte temporaire du signal
  - erreurs dues à une faute de transmission
- Perte de trames
- Perte ou modification du jeton
- Création de multiples moniteurs
- Non reconnaissance d'un délimiteur de trame

### Traitement des Erreurs Transitoires

La différence est dans la durée, fin constatée lors de la phase de reconfiguration

#### Origine des erreurs transitoires

- Panne vue par le moniteur actif

Elle entraîne la purge de l'anneau

Echec possible :

la commande n'est pas perçue par toutes stations

Ex : station moniteur de secours en cours de tentative de réélection

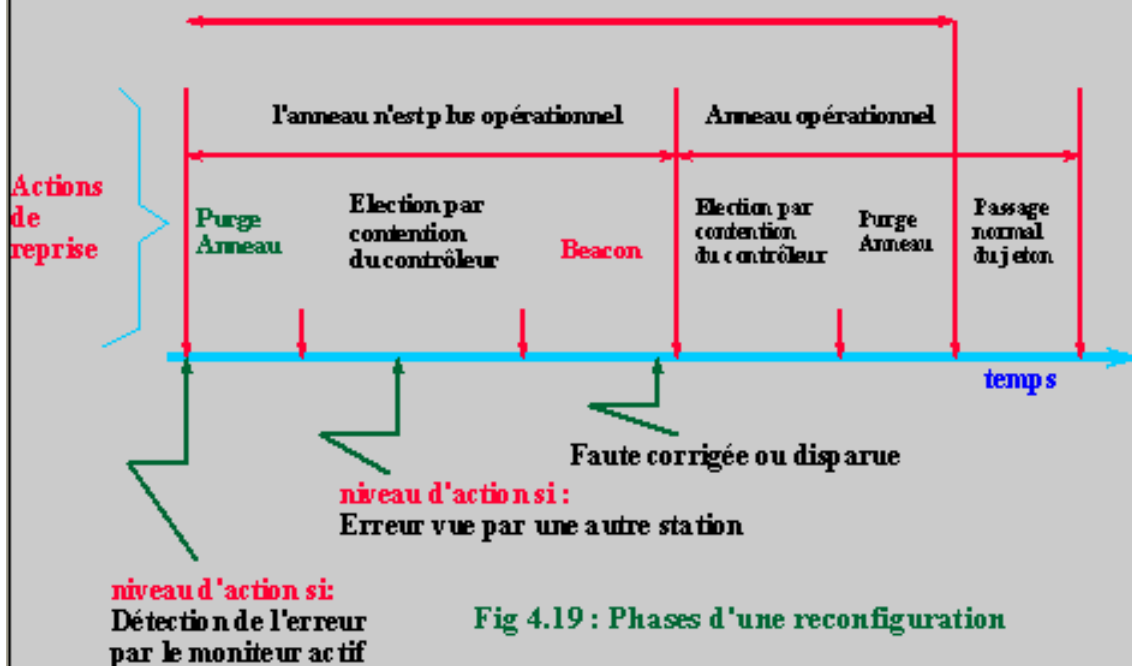
#### Phases de reconfigurations (dans ce cas)

- Panne vue par moniteur actif

Purge de l'anneau :

- Echec possible, la commande n'est pas perçue par toutes stations.
- Echec possible, la station moniteur de secours est en cours de tentative de réélection par exemple.

## Reconfiguration sur panne Transitoire



- 1er cas :

La panne cesse pendant la purge  
Le système entre en phase de réélection directe  
Le nouveau moniteur règle le problème  
L'anneau est de nouveau opérationnel

- 2eme cas:

Le processus ne trouve pas sa finalité à temps, situation d'erreur trop longue  
traitement identique aux pannes stables.

### Conclusion:

Il est à noter qu'un simple parasitage peut entraîner au minimum le séquentiel de purge.

# Le niveau LLC

Sommaire :

[Généralités sur LLC](#)

[Protocoles de la norme 802.2](#)

[Primitives et scénarii d'interactions réseau/LLC](#)

[Primitives hors connexion L Data](#)

[Primitives sur connexion](#)

[Primitives sur service datagramme acquitté](#)

[Scénarios en LLC de type 2](#)

[Principes de retransmission](#)

[Problèmes liés à la numérotation des trames](#)

[Scénarii d'échanges](#)

## Généralités sur LLC

### Définitions

Le niveau liaison ( logical Link Control) gère des liaisons de points à points :

- Il permet la réalisation des émissions et des réceptions des messages de la couche physique.
- Il rend la couche MAC transparente aux utilisateurs.
- Il permet à la couche Réseau de soumettre des paquets à transmettre.

### Définitions

**Que sont les datagrammes ?**

Ils se singularisent par le fait de pouvoir envoyer des paquets à un ou plusieurs utilisateurs de façon isolée. C'est est le mode basic retenu par les protocoles de cette couche , donc pas de relation de séquençement.

### Caractéristiques des protocoles LLC

Ils sont basés sur HDLC (High Level Data Link Control)

Ils sont appelés LAP (Link Access Protocol) dans le monde X25 ( LAP B,C,D )

Ils travaillent sur le champ Données des trames MAC .

# Protocoles de la norme 802.2

[Les 3 classes de services proposées](#) - [Définitions des PDUs et SAPs](#) - [Format des LPDUs](#)

## Les 3 classes de services proposées

### Le service sans connexion

**But: du IEEE LLC type 1:**

Fournir une garantie de livraison des messages appelés LSDU  
Permettre la détection et la reprise sur erreur  
Fournir un service sans connexion ni acquittement

### Le service sur connexion

**But du IEEE LLC type 2 :**

Créer et gérer des échanges sur connexions  
Acquitter les données  
Vérifier leur ordre  
Détecter les erreurs ou doublons  
Contrôler le flux

### Moyens employés :

La Numérotation ( le Protocole utilisé est identique au X25 LAP B)

L'identification de connexion unique se fait avec :

- Le couple SSAP/DSAP + le couple DA/SA (niveau. MAC).
- Une référence logique créée spécialement.

### Corollaire :

Les connexions. le sont entre 2 correspondants, pas de multipoint

### Le service de Datagramme acquitté

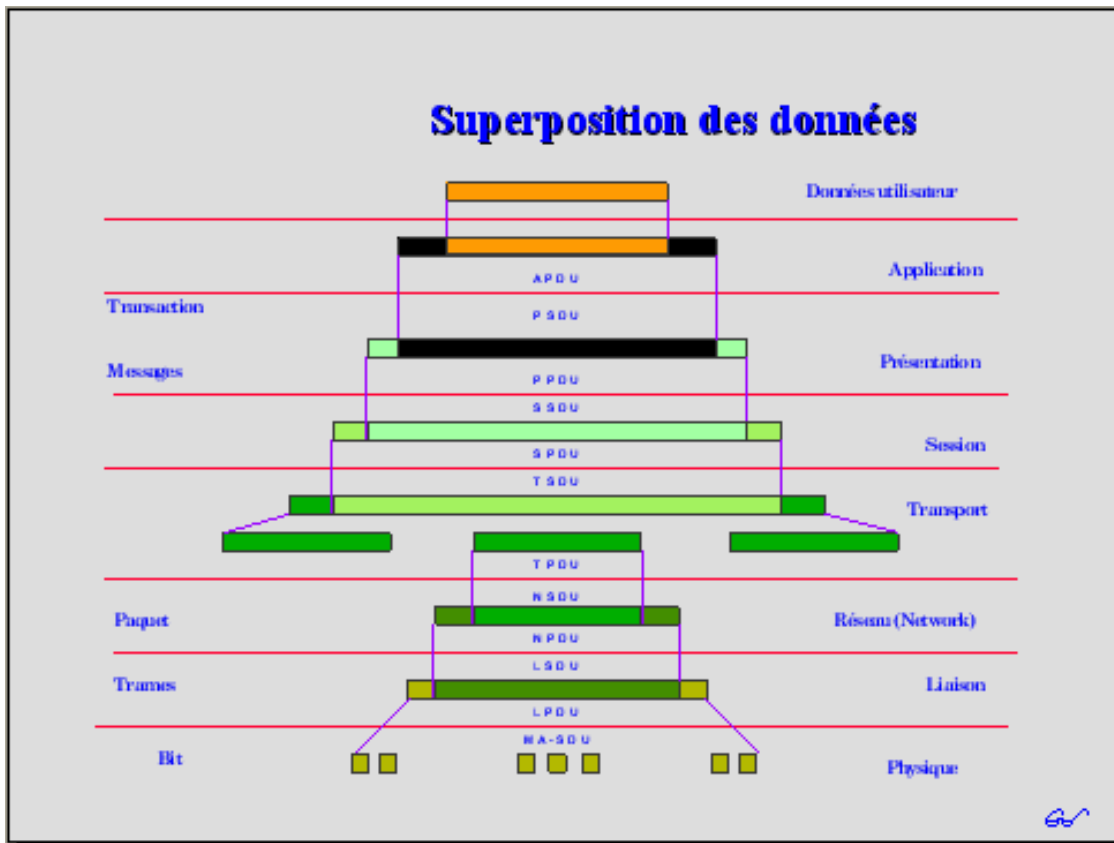
**But: du IEEE LLC type 3:**

Améliorer la fiabilité des échanges  
Offrir néanmoins une gestion facile :

- Pas de reprise si non-acquittement.
- Prévue pour le temps réel

Bornage supérieur du timer possible si MAC le prévoit.

## Définitions des PDUs et SAPs



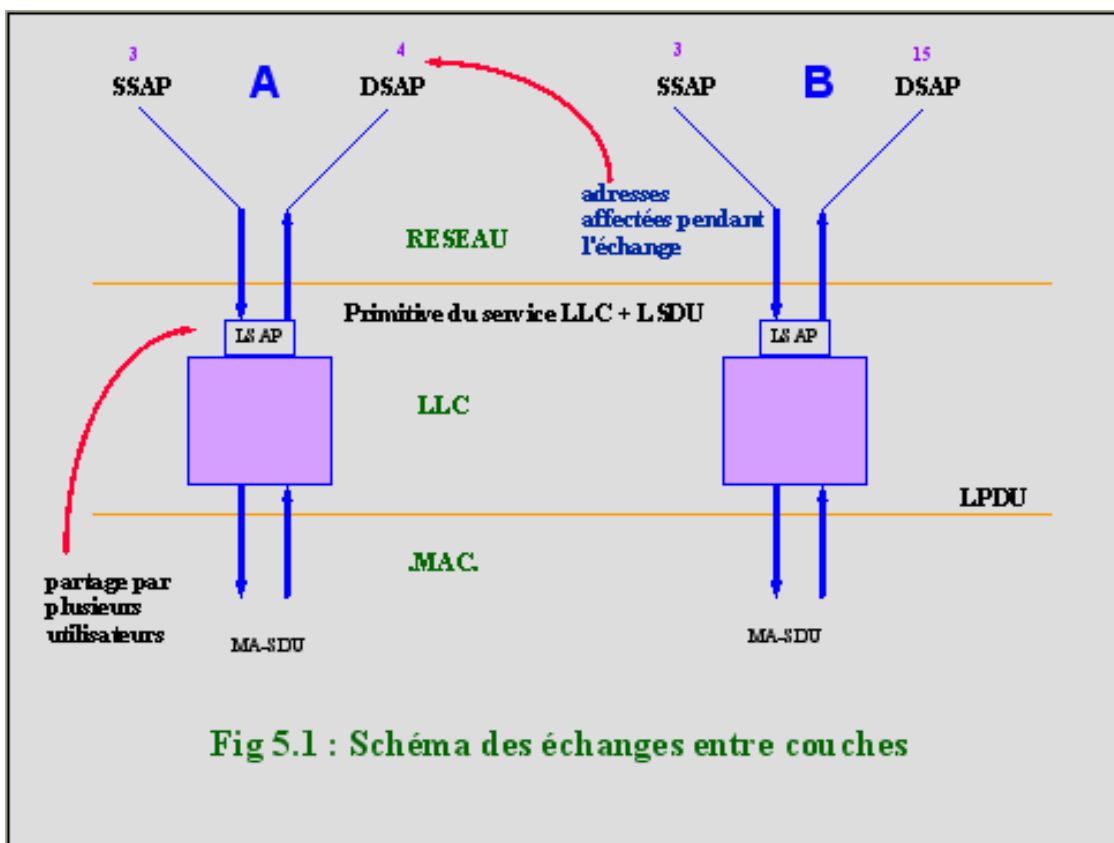
On observe que :

Une PDU de couche  $i$  devient une SDU dans la couche  $i-1$ .

Chaque enveloppe est utilisée pour la gestion du protocole de la couche qui l'insère.

Les enveloppes s'ajoutent .... avant transmission sur support PHYSIQUE.

Les enveloppes se retirent ..... lors de la livraison.



Définition des LPDU:

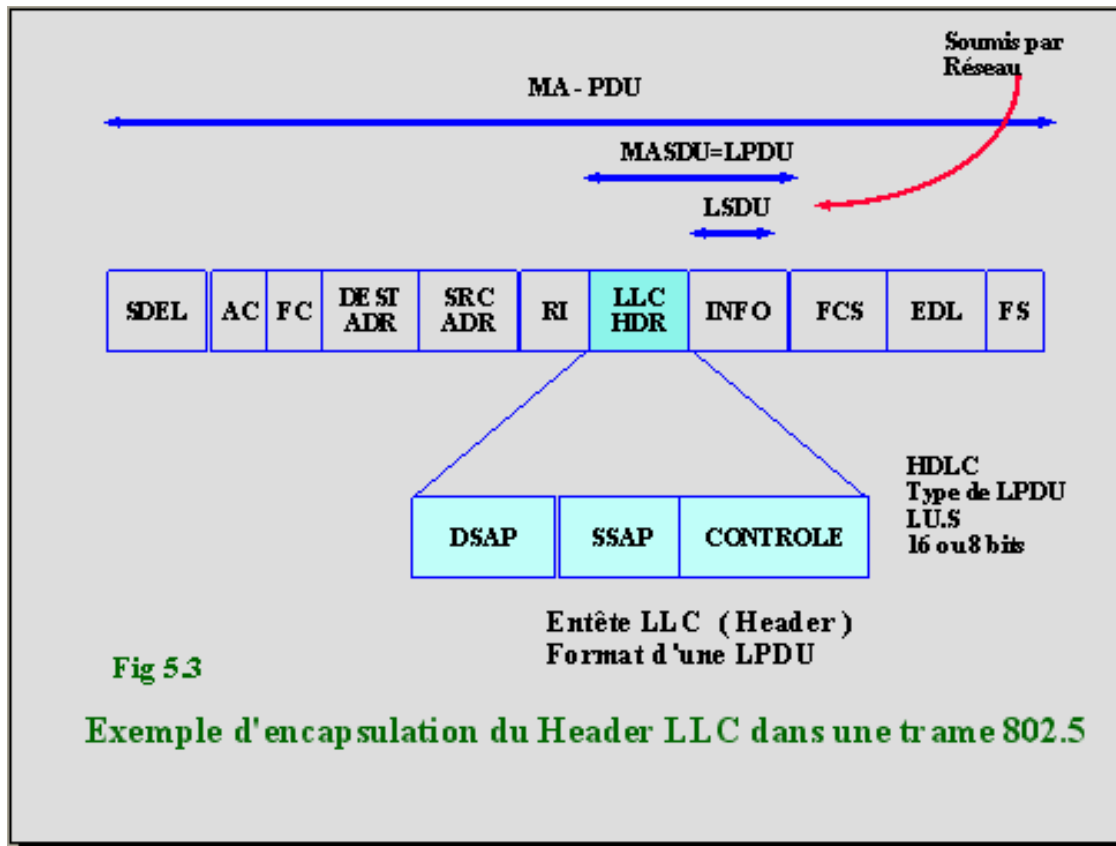
Ce sont des Entités de données émises entre deux LLC en communication.

La soumission et réception de ces données par les Utilisateurs se fait sous forme de LSDU à travers les Primitives

Définition des LSAP :

Ce sont les entrées locales du service Liaison, elles peuvent être utilisées par de multiples utilisateurs simultanément. A chaque utilisateur on affecte une DSAP et une SSAP pour les différencier

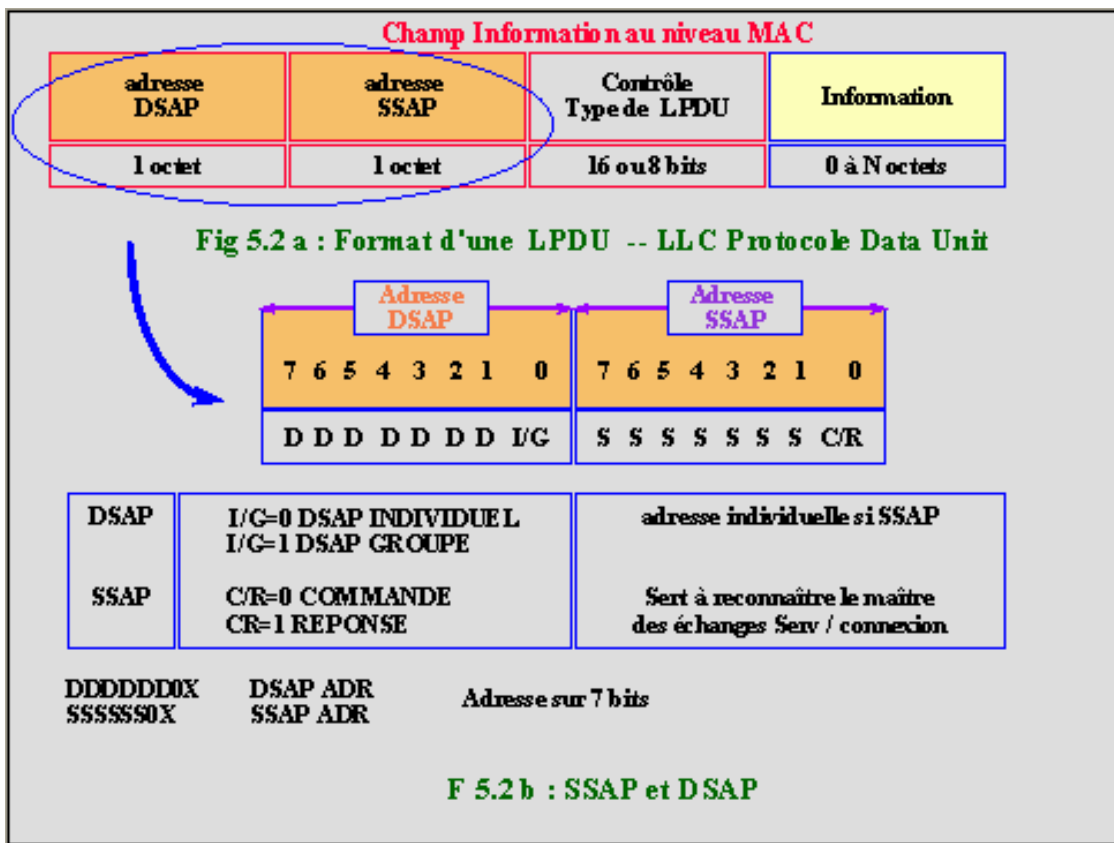
### Position des données LLC



### Données LLC

La figure ci-contre présente le schéma d'encapsulation des données LLC dans une trame Token Ring de niveau MAC.

### Format des LPDUs



Les octets décrits sont le Champ DONNEE des trames MAC sans la MA-SDU soit la LPDU.

### Format des champs d'adresse :

SSAP et DSAP désignent 1 ou plusieurs SAP locaux aux LLC impliqués.

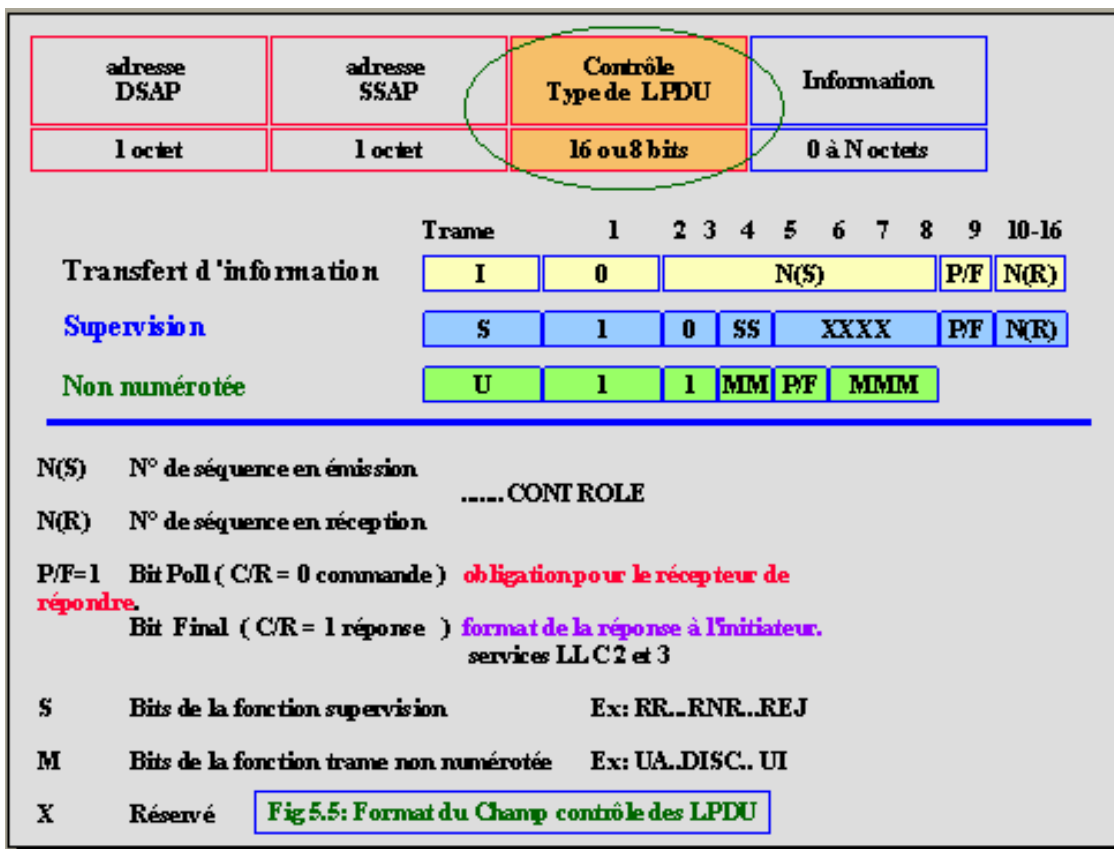
- Il permettent l'identification de l'origine de l'échange avec l'adresse.:

Association au SSAP de la SA trouvée dans la trame MAC

Adresse DASP ou SSAP = 1 Octet : 7 bits pour l'adresse + 1 bit spécial ... I/G, C/R

- Caractéristique à observer : Dans le champ DSAP voir I/G, dans le champ SSAP voir C/R, pour mémoire : le bit C/R sert à reconnaître le maître de l'esclave.

Trois type de LPDU sont présentées :



I :  
information  
S : supervision  
U : non  
numérotée  
(Unnumbered)

Observer la présence du bit  
P/F dans toutes les trames

## Le champ de contrôle d'une LPDU

Il est conforme au format étendu de HDLC et Il définit 3 Types de LPDU :

### U Les LPDU non numérotées (voir tableau) incluent

Le bit P/F qui est décrit ci-après  
 Un champ Contrôle sur 1 octet  
 Le bit M qui code une fonction

### S Les LPDU de supervision

Gèrent en service de type 2 : le contrôle de flux et les retransmissions RR, RNR, REJ  
 Seul le No de Séquence en Réception est utilisé, ceci permet à l'Emetteur d'indiquer quel No de trame il s'attend à recevoir ( voir plus loin).

### I Les LPDU d'information incluent

Un seul type pour Commande ou Réponse  
 Le champ N(S) =No Séquence en Emissions = numérote les trames  
 Le champ N(R) =No Séquence en Réception = acquitte toutes les trames déjà reçues (voir mécanisme)

## Fonctions du Bit P/F

Dans les types U, S, I ... le bit P/F fonctionne en conjonction avec le bit C/R du champ SSAP (fig. précédente)



# LPDU Synthèse et champs

## □ Fonctions du Bit P/F

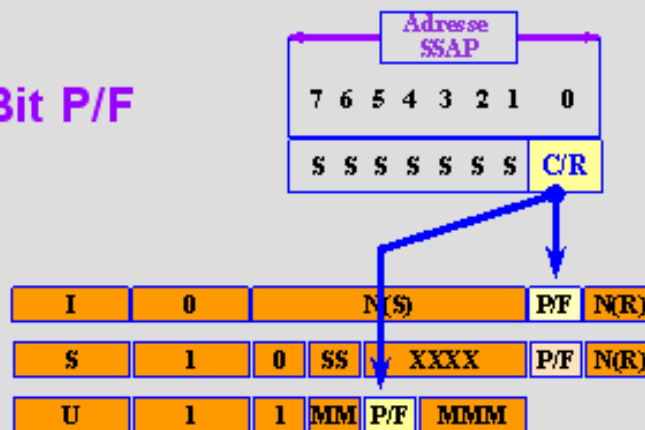
C/R=0

C/R=1

Transfert d'information

Supervision

Non numérotée



**BUT: Imposer une réponse à un LLC silencieux**

**C/R=0 : Trame de commande...** si P/F=1 (POOL)

sollicite une réponse du LLC adressé  
la réponse vient dans trame C/R=1

**C/R=1 : Trame de réponse...** si P/F=1 (FINAL)

indique que LLC distant répond à la sollicitation

## Objectif du bit P/F :

Imposer une réponse à un LLC silencieux.

Il est utilisé dans une LPDU type 2 et 3. Dans type 2 permet de résoudre des cas de dysfonctionnement et de reprise après ERREURS

## Codes des LPDUs de Supervision

Les LPDUs de Supervision :

Codes des LPDU de Supervision utilisées dans la gestion des protocoles sur connexion en Type 2 IEEE 802.2

Supervision

S	1	0	SS	XXXX	PF	N(R)
---	---	---	----	------	----	------

Code SS	Commande	Réponse	Sémantique
00	RR	RR	Receveur prêt
10	RNR	RNR	Receveur non prêt
01	REJ	REJ	Rejet

Champ contrôle LPDU I

LLC 2

Tableau 5.2 : Codes des LPDU de Supervision utilisées dans la gestion des protocoles sur connexion  
Type 2 IEEE 802.2

Tableau des LPDUs non numérotées utilisées

Champ contrôle LPDU U											
Non numérotée											
<table border="1"> <tr> <td>U</td> <td>1</td> <td>1</td> <td>MM</td> <td>PF</td> <td>MMM</td> </tr> </table>						U	1	1	MM	PF	MMM
U	1	1	MM	PF	MMM						
type LLC	Code (héxa)	LPDU de									
		Commande	Réponse	sémantique							
1	0 1D 07	UI XID TEST		Information non numérotées Echange d'identification Test							
2	1E 06 02 18 11	SABME  DISC	UA  DM FRMR	Mise en mode asynchrone équilibré étendu Acquittement non numéroté Déconnexion Mode déconnecté Rejet de trame							
3	0 06 02	UI	UA FRMR	Information non numérotée Acquittement non numéroté Rejet de trame							

Table 5.1 : LPDU non numérotées utilisées

# Primitives et Scénarios d'interactions Réseau / LLC

## Les primitives

### Les primitives

#### A quoi servent les primitives ?

Elles servent à réaliser les Protocoles. Chaque couche met à disposition de sa voisine des primitives pour réaliser des services

Voir les illustration par diagramme temporel en scénario

#### Trois types de primitives utilisées en LLC

##### REQUETE :

- La couche réseau soumet une Primitive pour l'exécuter.

##### INDICATION :

- LLC indique à Réseau (destinataire) soit :

L'arrivée d'une SDU depuis LLC distant  
Une ouverture de connexion  
Une fermeture de connexion

##### CONFIRMATION :

- LLC signale à la couche Réseau (origine) la fin de l'exécution de sa précédente requête

**Fig 5.6** Diagramme temporel des interactions

**Fig 5.6 bis Diagramme temporel des interactions**

## Primitives hors connexion L\_DATA

- Elles permettent d' EMETTRE une trame simple hors connexion de type 1

### Généralités

La requête donne en Paramètre les adresses LSAP source et destination

Les données sont dans le champ LSDU

La classe de service donne une indication de priorité elle peut être utilisée si MAC est de type à jeton :

- LDSAP va entrer dans la construction de la trame MAC
- L'adresse distante contient :

le DSAP

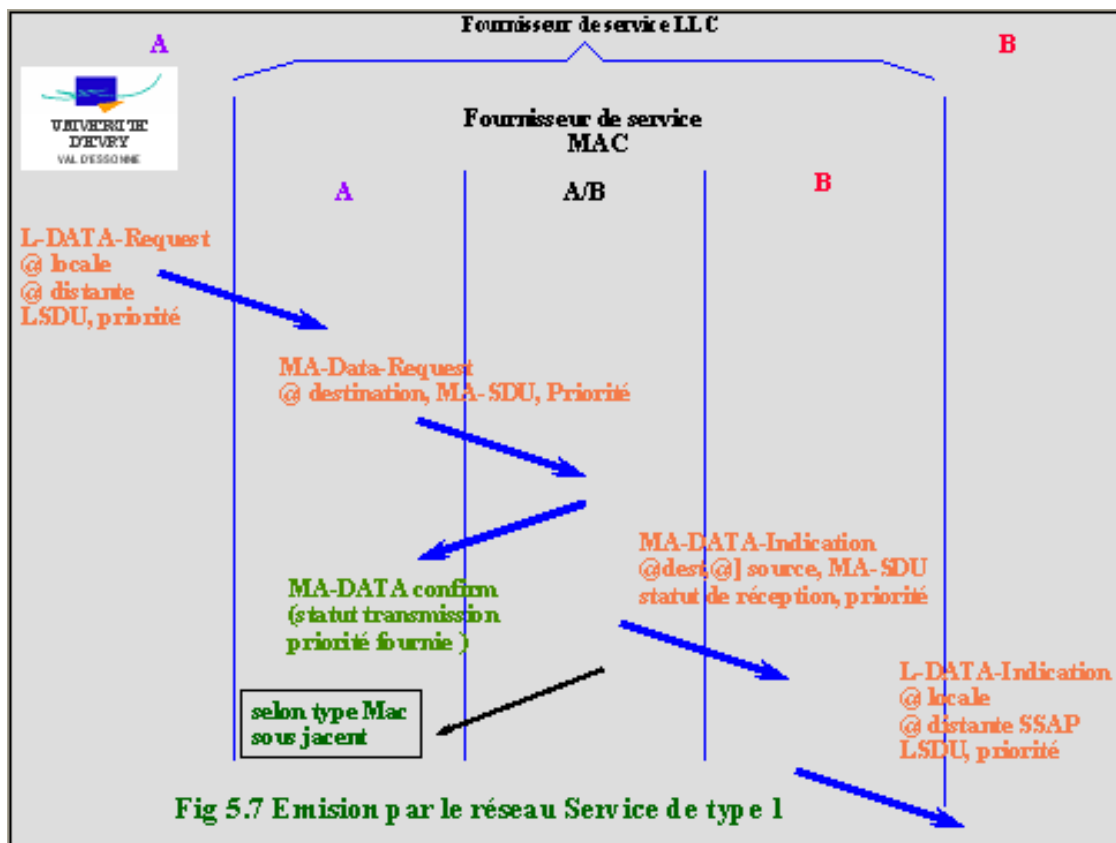
le DA (MAC) à utiliser

### Interaction Réseau - LLC - MAC

#### Requête du réseau

LLC soumet MA-DATA request à MAC (voir paramètres fournis)

MAC confirme avec une MA-DATA confirmation en indiquant dans le champ Statut :



- comment s'est passé la Transmission
- quelle est la priorité retenue

**Nota:**

La confirmation ne garantit pas la réception du message ( voir MAC utilisé seule la norme 802.5 contient : un bit indiquant ..adresse reconnue et un bit indiquant ..trame copiée.

Si la communication est réussie chaque LLC reçoit une MA- DATA indication avec comme paramètres :

- adresse origine
- compte rendu de transmission.
- + longueur message
- + priorité

### Action de LLC

- Il lit champ contrôle voit une LPDU type U
- Il lit l'adresse DSAP, sait à quel service RESEAU il doit fournir la LSDU correspondante
- Il le fait avec une L-DATA indication qui comportera les indication SSAP et SA

## Primitives sur Connexion

### Primitives utilisées

Elles permettent de gérer une trame sur connexion type 2

Une connexion nécessite 3 phases :

- ouverture
- transfert
- fermeture

### Primitives utilisées

Elles sont au nombre de 5

L- CONNECT

Ouverture

L- DATA CONNECT

Transfert de données

L- RESET

Purge

- Perte de toutes LSDU
- Retour connexions état post ouverture
- Reprise pour les correspondants après confirmation. et indic.

L- DATA FLOW CONTROL

Modification de débit

- Ajuste quantité de données admise sans risque de perte.
- Le paramètre montant la spécifie
- A titre local entre le LLC et une auto génération de contrôle de flux peut être émis
- Est indépendant sur chaque coté de la connexion.

## L- DATA DISCONNECT

### Fermeture

- peut être généré indifféremment par chaque correspondant

Primitive de service	Paramètres		
	Requête	Confirmation	Indication
L-CONNECT ouverture	@ locale @ distante Classe de service	@ locale @ distante Statut Classe de service	@ locale @ distante Statut Classe de service
L-DATA CONNECT transfert	@ locale @ distante LSDU	@ locale @ distante Statut	@ locale @ distante LSDU
L-DISCONNECT fermeture	@ locale @ distante	@ locale @ distante Statut	@ locale @ distante Cause
L-RESET purge	@ locale @ distante	@ locale @ distante Statut	@ locale @ distante Cause
L-CONNECT-FLOW-CONTROL modif de débit	@ locale @ distante Montant		@ locale @ distante montant

**Table 5.3 : Primitive du service LLC sur connexion ( Type 2 )**

## Rappel des services types 2

Séquencement

Reconnaissance des pertes

Retransmission

Elimination des doublons

Full duplex pour transmission des données

## Rappel des principes

- Deux correspondants maximum
- Adresse complète du correspondant à fournir :
  - Soit la DSAP ( une référence peut être substituée à l'adresse après connexion)

La DA et son adresse SSAP, ce sont les seuls paramètres à fournir.

Le LLC distant peut refuser si :

- DSAP inconnu
- Manque de ressource
- Ne peut réaliser la classe de service demandée

## Primitives sur Service Datagramme Acquitté

### Rappel des principes

### Rappel des principes

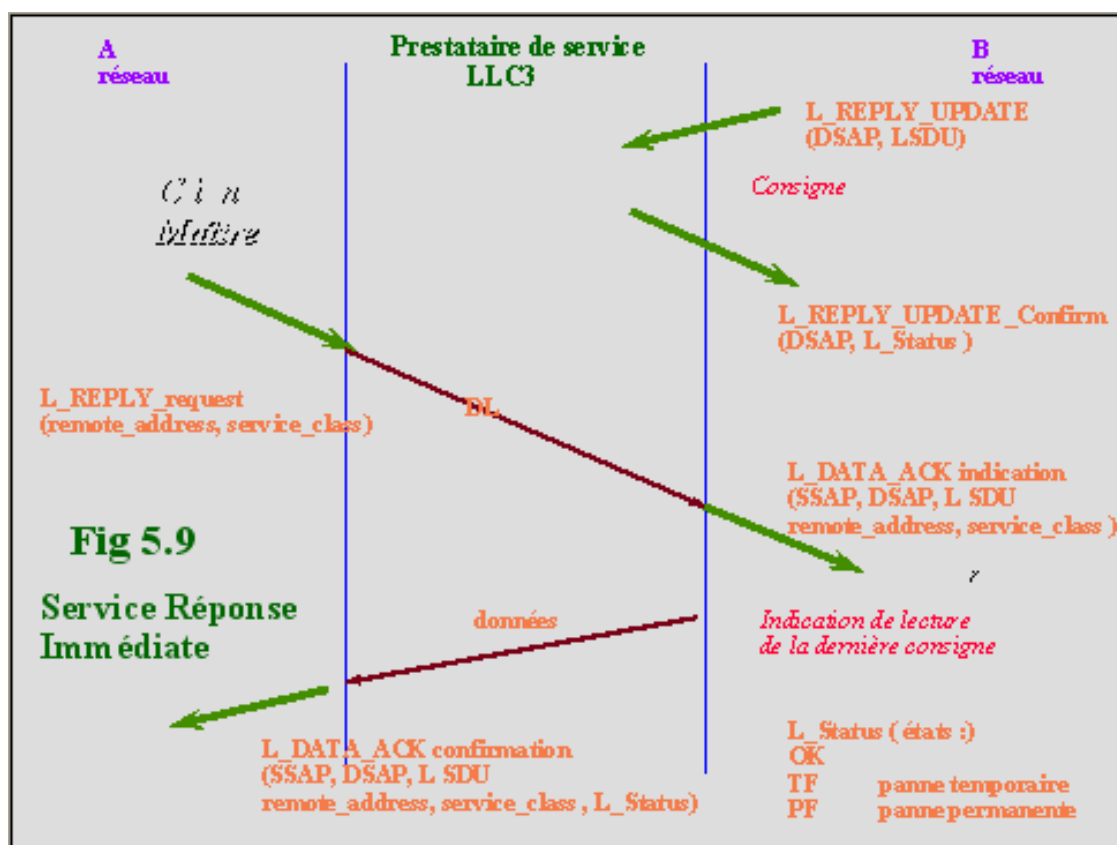
Elles permettent de gérer une trame sans connexion type 3.

- Il s'agit de trouver un moyen terme entre le type 1 qui ne garantit rien et le type 2 sécurisé mais lourd à gérer.
- L'acquittement confirme l'arrivée du message néanmoins la retransmission n'est pas garantie si l'acquittement arrive hors délai
- Une borne supérieure peut exister si MAC le permet.

### Contraintes

- Elles sont définies uniquement pour transmissions de point à point
- Le nombre de LSDU soumis avant confirmation est limité
- Elles sont destinées aux applications à temps réel contraint

### Types de primitives et services

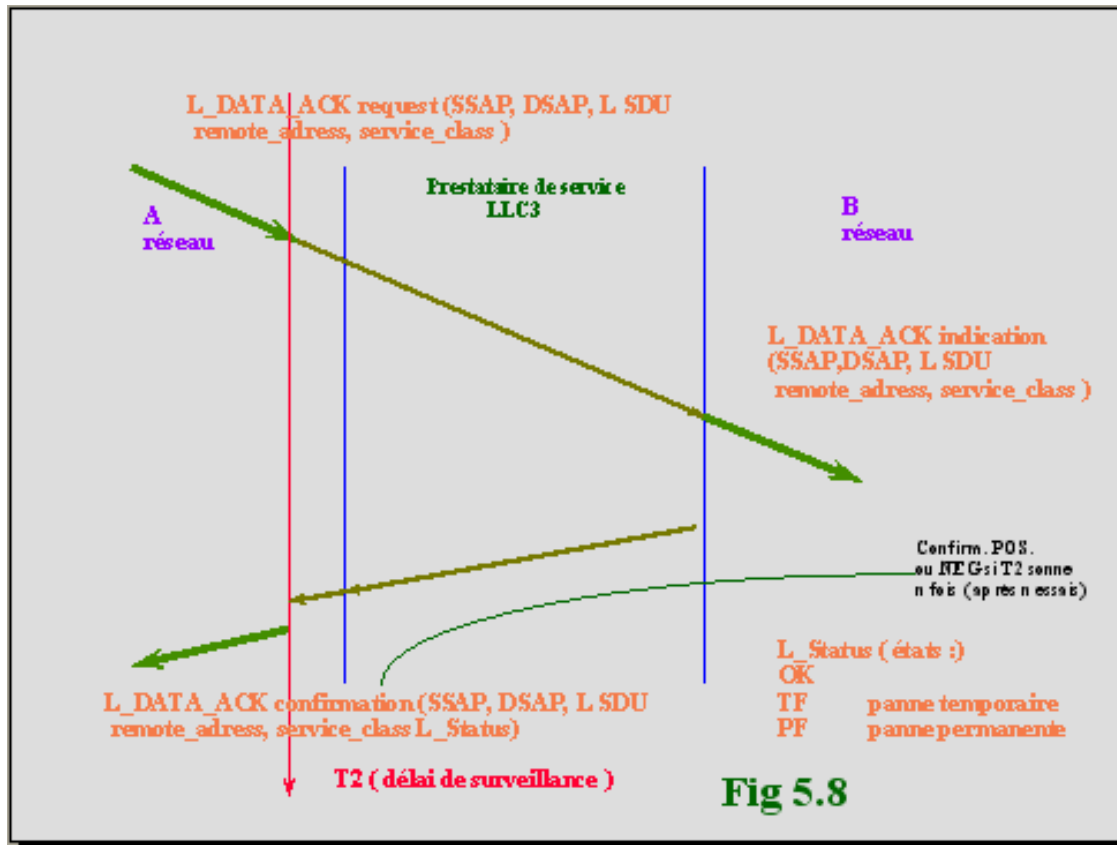


L-DATA ACK service  
Données acquittées  
L-DATA ACK service  
Réponse immédiate  
L-REPLY\_UPDATE  
L-REPLY

Prévues pour équipements peu intelligents

Formalise le principe de polling à scrutation, Conçues dans le cadre de MAP (Manufacturing Automation Protocols)





**Requête :** la LPDU est envoyée au destinataire

Un timer **T2** est armé pour surveiller le non retour d'acquittement.

Le destinataire acquitte la LPDU et délivre la LSDU au DSAP indiqué

- Si T2 sonne les données sont réexpédiées.
- Si après n répétitions de tentatives aucun ACK ne revient, une confirmation NEG est rendu au demandeur.

## Scénarii en LLC de type 2

Séquence d'ouverture de connexion - Scénario d'ouverture impossible - Autres scénarii - Transfert de données

### Séquence d'ouverture de connexion

Réseau A -> L-CONNECT request

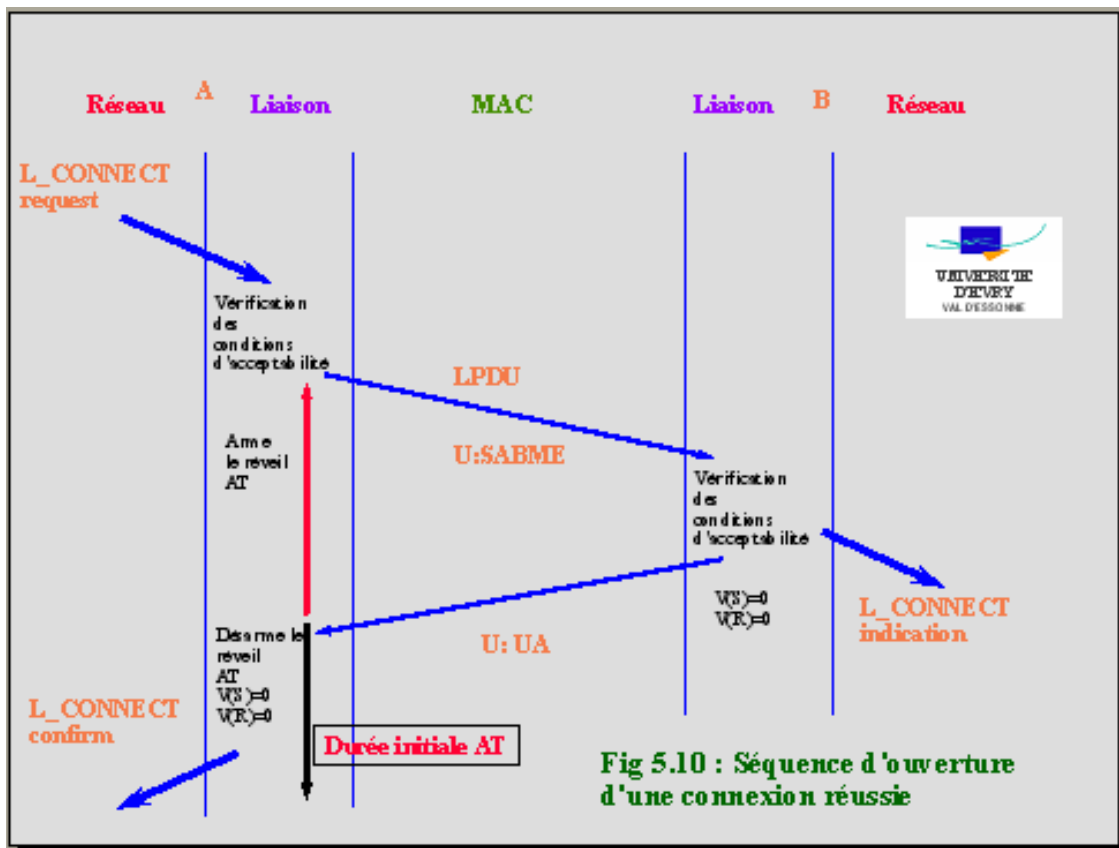
- LLC vérifie si correct. (ressources, adresses, etc..)

LLC A -> LPDU U :SABME -> LLC B

- arme AT pour attente de réponse

LLC B <- U :SABME

- vérifie correct. note que la DSAP reçue est locale



LLC B -> LPDU U:UA. envoyée

- met V(R) et V(S) = 0 initialisation pour décompte

LLC B -> L-CONNECT indication à -> Réseau B

- connexion ouverte de ce coté

LLC A <- LPDU U:UA. reçue

- désarme AT
- met V(R) et V(S) = 0 initialisation pour décompte

LLC A -> L-CONNECT confirmation au demandeur Réseau A

### Scénario d'ouverture impossible



LLC A -> LPDU SABME LLC B

LLC B -> LPDU SABME LLC A

**Collision** : si les correspondants acceptaient ..le résultat serait : 2 connexions

LLC A -> LPDU UA

LLC B -> LPDU UA

Chacun vérifie si une connexion de sa propre origine n'est pas déjà ouverte avant de fournir une indication au DSAP.

## Fermeture de connexion

Elle peut être faite par l'un quelconque des correspondants

LLC A <- L-DISCONNECT request

- Les données en transfert de A->B peuvent être perdues et en transfert de B->A sont perdues

LLC A -> LPDU DISC

- Libère ressources tampons , compteurs, désarme les timers et Arme AT

LLC B <- LPDU DISC

- Libère et désarme

LLC B -> LPDU UA

LLC A <- LPDU UA

- Désarme AT, aucune trace n'est gardée

## Transfert de données

### Principes de la Phase Transfert

Dès émission ou réception de LPDU UA la phase transfert commence.

Des LPDU de type I (information) et S ( supervision) sont utilisées pendant cette phase pour :

- gérer les échanges
- garantir le séquençement
- récupérer les pertes

### Principes généraux des techniques

Ils sont utilisables dans les trois protocoles et comportent :

- No de séquence
- Acquittements

Chaque trame peut être acquittée positivement et individuellement par une trame en retour :ACK

L'ACK peut être incorporé dans une trame de donnée en retour (Piggy Aking )

Autre stratégie utiliser des NAK pour d'acquittement NEG: signal d'erreur

Des réveils sont armés pour éviter les DEAD LOCKS

### **Retransmissions automatiques**

- La politique consiste à garder en tampons tout ce qui n'est pas ACK acquitté positivement

## **Principes de retransmission**

STOP AND WAIT - GO BACK N

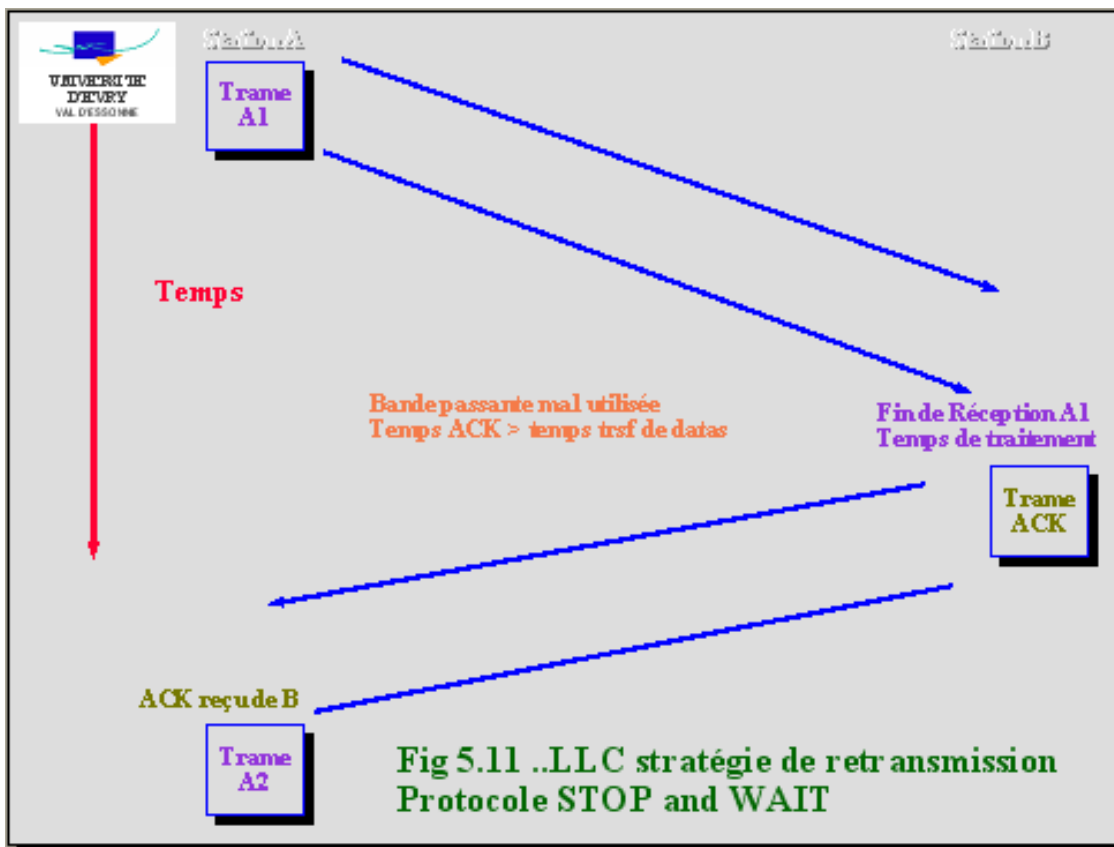
### **STOP AND WAIT**

**Principes :**

- Une seule trame est émise à la fois

- On attend l'ACK

Michel Besson



ou NAK avec réveil armé

### Conséquences :

Mauvaise utilisation de la bande passante, générer un ACK prend plus de temps que transmettre des données

## GO BACK N

### Principe sur NAK:

Toutes les trames sont transmises en continu

- Sur réception d'un NAK ( fig 5.12a NAK3 ) on reprend la transmission à N trame en arrière, soit au niveau du 1er NAK reçu.
- Toutes celles qui suivent sont réémises

Ce principe n'est pas adapté au réseau local

Il est difficile de renvoyer un NAK si le correspondant n'est pas lisible (CRC erroné )

- Convient bien au point à point

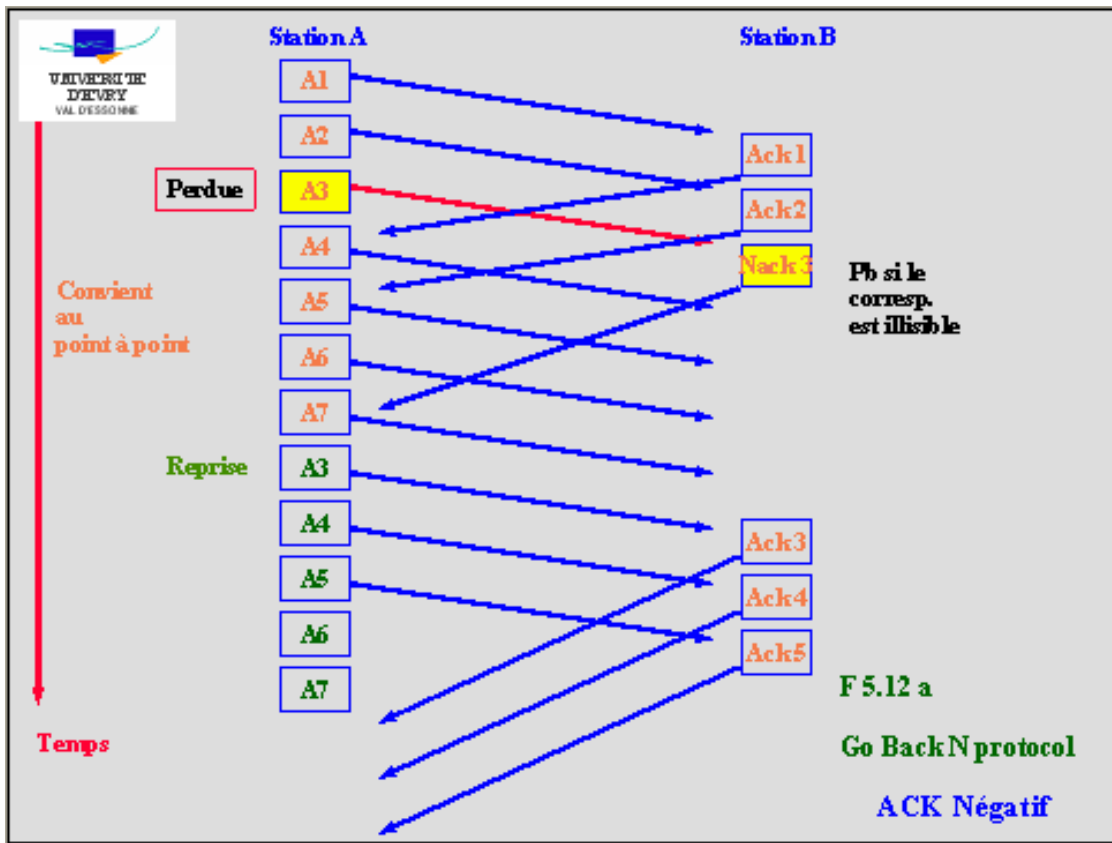
### Principe GO BACK N stratégie basée sur les NAK

## Utilisation de NAK en conjonction avec des Numéros de Séquence

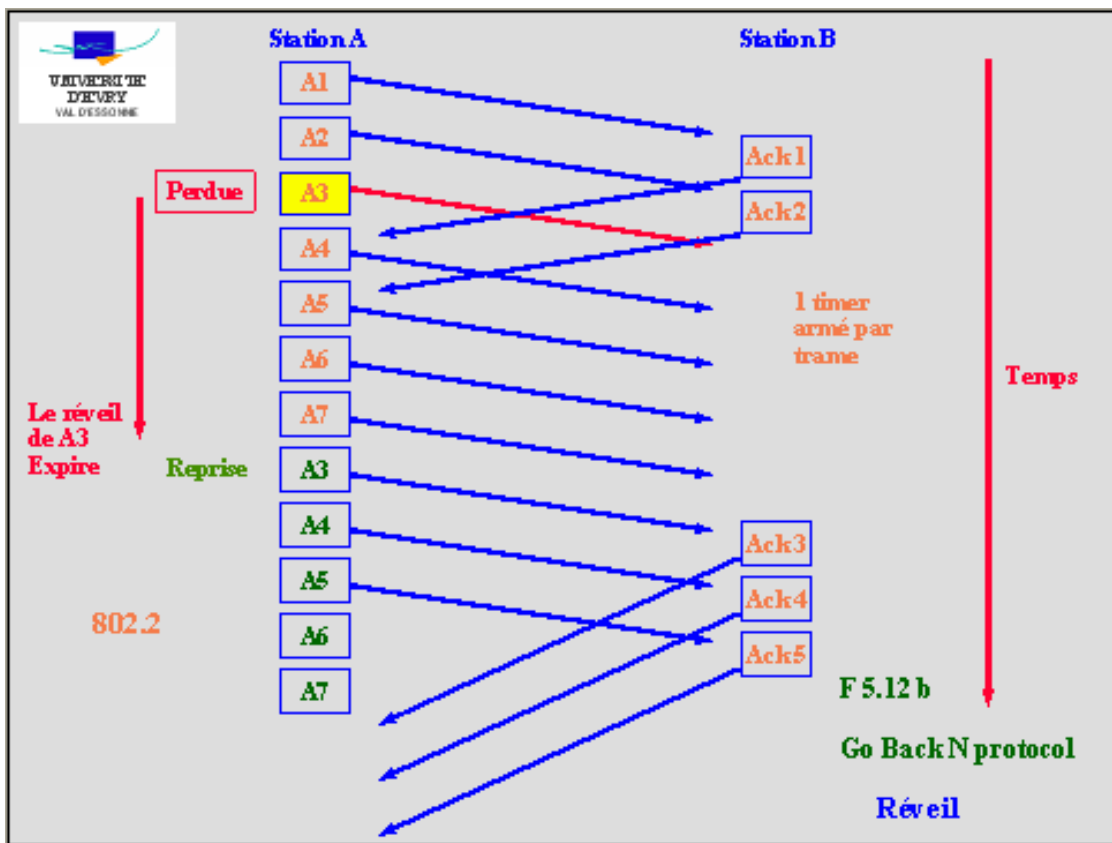
- Seule la trame dotée d'un NAK ou bien dont le réveil sonne est renvoyée.
- Le récepteur doit réordonnancer ses tampons pour respecter l'ordre
- Technique avantageuse dans les réseaux satellites aux délais de propagation longs.

Peu utilisée dans les réseaux locaux

- Nombre de tampons important
- Implémentation lourde



## GO BACK N Stratégie basée sur le réveil



## Principe sur Expiration du réveil:

- Un réveil est armé pour chaque trame émise
- Un réveil armé pour la 1ère trame non acquittée, en réalité on l'arme sur la 1ère trame non acquittée au moment la réception de l'ACK de la précédente
- Celui de A3 expire (trame non acquittée) et provoque une réaction de même type.
- 802.2 utilise cette

[anticipation \(décrite plus loin\)](#)

## Problèmes liés à la numérotation des trames

### Fenêtre d'anticipation - Séquencement

#### Fenêtre d'anticipation

Les compteurs N(S) et N(R) sont sur 7 bits donc modulo 128

Si l'ACK n'arrive pas avant le No 128 = fin de transmission tampons épuisés.

**En pratique:** La limite est K tampons dont dispose le LLC pour émettre soit  $\leq 128$  )

**La fenêtre d'anticipation :** c'est le nombre de LPDU que peut envoyer un LLC avant blocage.

- Le récepteur devra avoir aussi K tampons disponibles.
- Cette valeur K est fixée à l'ouverture de la connexion pour toute la durée de sa vie, de même que la taille des trames qui influera sur la taille des tampons.

#### Séquencement

**Rappel types de compteurs, ils définissent soit :**

Un évènement terminé





Trame	1	2	3	4	5	6	7	8	9	10-16
Transfert d'information	I	0	N(S)						P/F	N(R)
Supervision	S	1	0	SS	XXXX				P/F	N(R)
Non numérotée	U	1	1	MM	P/F	MMM				

Avec :

- N(S) N° de séquence en émission ..... CONTROLE
- N(R) N° de séquence en réception
- S Bits de la fonction supervision Ex: RR..RNR..REJ
- M Bits de la fonction trame non numérotée Ex: UA..DISC..UI
- X Réserve
- P/F=1 Bit Poll ( C/R = 0 commande ) obligation pour le récepteur de répondre.  
 Bit Final ( C/R = 1 réponse ) format de la réponse à l'initiateur.  
 services LLC 2 et 3

**Fig 5.5: Format du Champ contrôle des LPDU**

N(S) est le No de trame Emise  
 N(R) est le No de trame Attendue

Un évènement prévu

V(S) est le No de trame à Emettre  
 V(R) est le No de trame Attendue  
 bit Pool=1 C/R=0 commande  
 bit Final=1 C/R=1 réponse

## Scénarii d'échanges

Scénario 1 Envoi de LPDU I - Le contrôle de Flux - Cas typiques lies au Bit Pool - Phénomènes d'asynchronismes et REJ - Gestion de la fenêtre d'anticipation

### Scénario 1 Envoi de LPDU I

LLC reçoit une commande L-DATA\_ CONNECT

LLC crée une LPDU ..I correspondante

- met N(S) = V(S)
- met N(R)= V(R)
- met bits Polling C/R=0

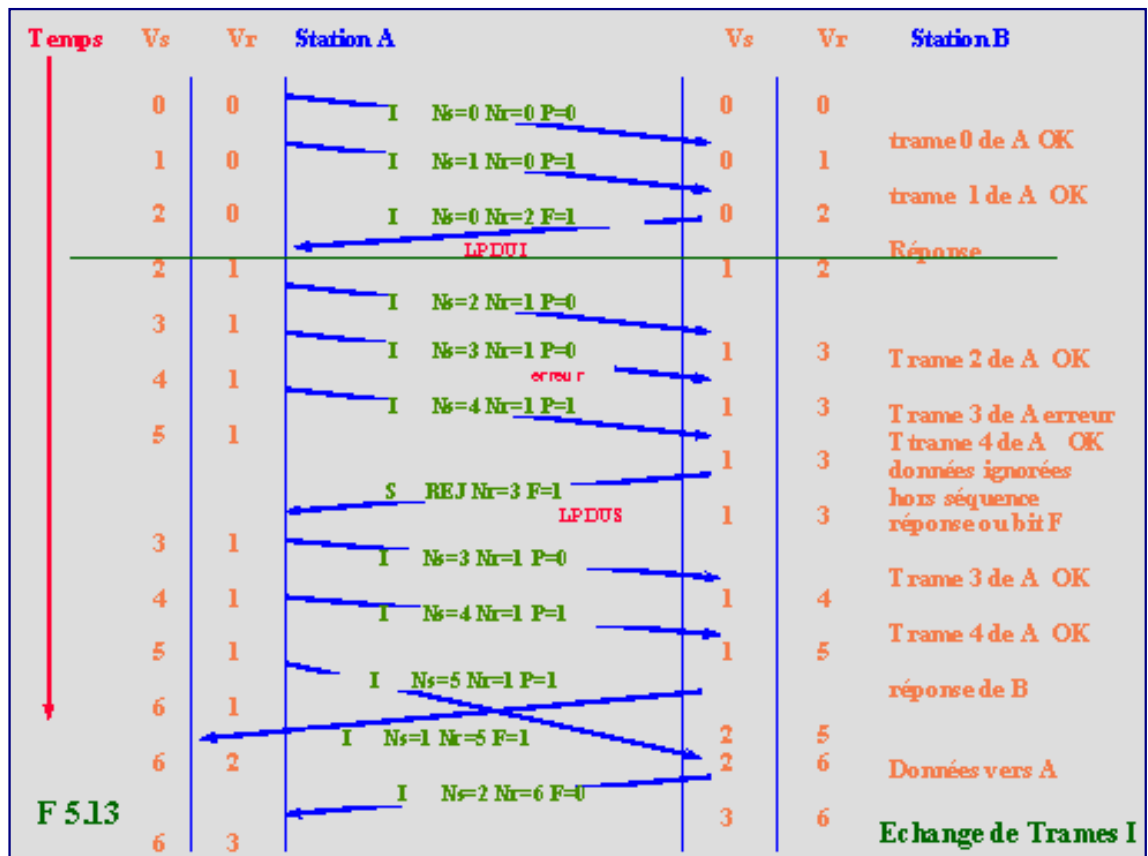
## Première partie du S1 sans erreur

A -&gt; LPDU

- $V(S)+1=1$ , AT armé

B &lt;- reçoit la 0

- accepte car  $N(S)=\text{son } V(R)$
- $P=0$  pas d'obligation de réponse  $V(R)=V(R)+1=1$
- si attente de timing  $>T_A$  ... A doit réémettre
- si pas de donnée à re-expédier la réponse peut être : RR



B -&gt; LPDU suite à demande. état de la 1ere : P=1

- TA cours toujours sur la 0

B &lt;- reçoit la 1

- accepte car  $N(S)=\text{son } V(R)$
- sait qu'il faut répondre, vu  $P=1$

Suite scénario 1

B -&gt; LPDU ..I

- en profite pour répondre
- met  $C/R=1, F=1$
- met  $N(S)=V(R)=0$
- met  $N(R)=V(S)=2$
- fait  $V(S)=V(S)+1=1$

A <- la reçoit

- sait que 0 et 1 ont été reçues
- libère ses tampons

A -> L-DATA CONNECT

- confirmation ...x 2 à son commanditaire
- une par LSDU transmises
- désarme TA de la 1ere

## 2ème partie du S1 : l'erreur est détectée

A -> LPDU 2

- Le réveil TA la suit

-> LPDU 3

-> LPDU 4

- La LPDU 4 comporte Communication de polling

B <- LPDU 2

- OK car  $N(S)=V(R)$

A -> LPDU 3

- Trame  $N(S)3$  en erreur
- Pas de conclusion

B <- LPDU 4 - Hors séquence  $N(S)$  différence au  $V(R)$

- Ignore la partie LSDU, ne la délivre pas à l'Ut.
- Voit origine, voit bit  $P=1$ , doit répondre

B -> LPDU ..S : REJ Car pas de données à transmettre.

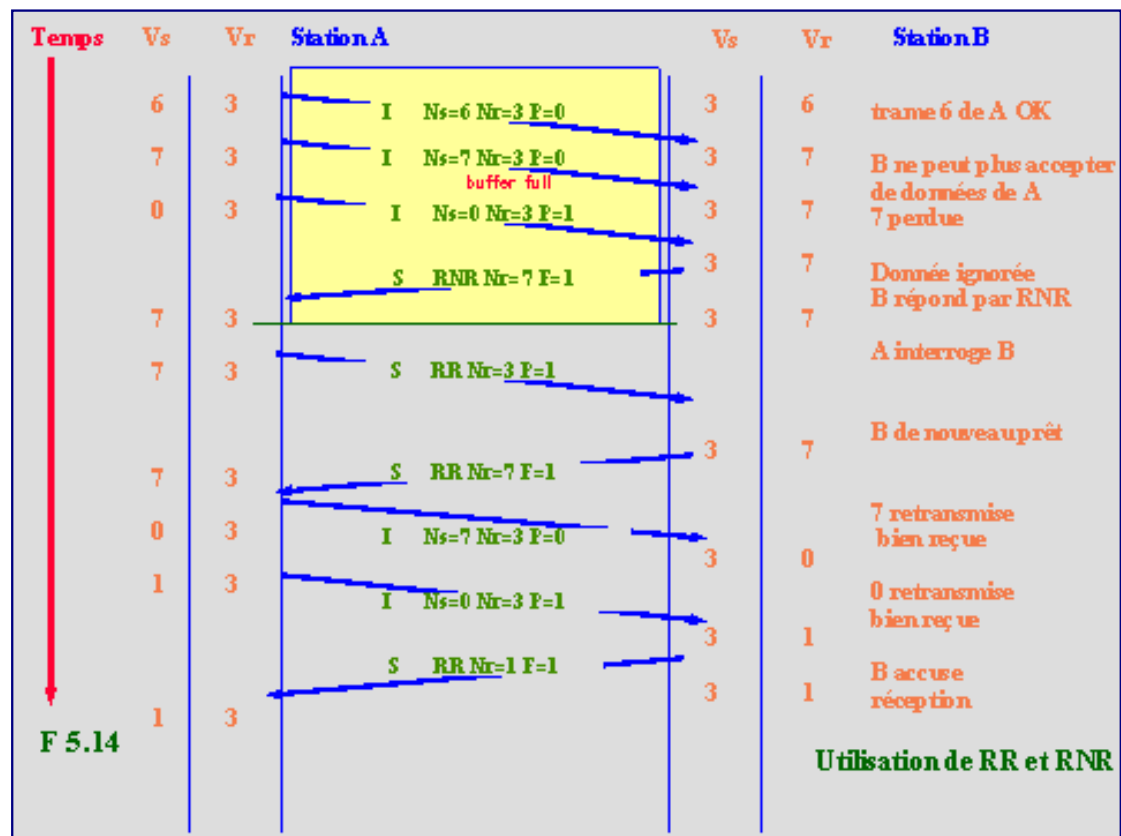
- Il précise par  $N(R)=V(R)=3$  le No attendu en Séquence.
- $V(S)$  n'est pas incrémenté ..supervision = hors séquence
- Un réveil particulier est armé pour surveiller ce REJ
- S'il sonne n tentatives seront tentées avant de décider la fermeture finale.
- Les autres LPDU arrivées hors délai seront ignorées.

A <- LPDU ..S : REJ

- Sait qu'il doit renvoyer la 3
- Remet  $V(S) = 3$  et reprend a ce niveau ( Go Back N )

## Le contrôle de Flux

## Utilisation de RR et RNR



## Contrôle de Flux

### Etat initial du scénario

Celui-ci fait suite à la dernière séquence, de la fig. 5.13, nous supposons qu'après réception de la LPDU 6 ... B à utilisé tous ses tampons.

Cause possible l'utilisateur ne retire pas ses données

B <- LPDU 6 sans problème

B <- LPDU 7 bit Pool=0

- Aucun avis, bien qu'il soit en situation de blocage

B <- LPDU 0 bit Pool=1

B -> LPDU S:RNR N(R)=7

A <- LPDU S:RNR

- A voit la situation, mais ne peut qu'attendre le déblocage. de B met son V(S)=7

A -> LPDU S:RR

- Interrogation de A à B bit P=1 pour forcer B à répondre, ce qui permet de vérifier l'état de B ou d'être fixé sur les Acquittements.

B -> LPDU S:RR bit F=1 mode réponse

- Ok retour à la normale

A <- LPDU S:RR sait qu'il peut renvoyer des ..I

- A ne connaît pas la valeur de fenêtre disponible pour autant

A -> LPDU...I ..N=7

A -> LPDU...I ..N=0

- Celle-ci comportera une interrogation P=1

B <- les reçoit

B -> LPDU S:RR

- B répondra par RR car aucune données à transmettre, acquitte les 2 reçues au passage.

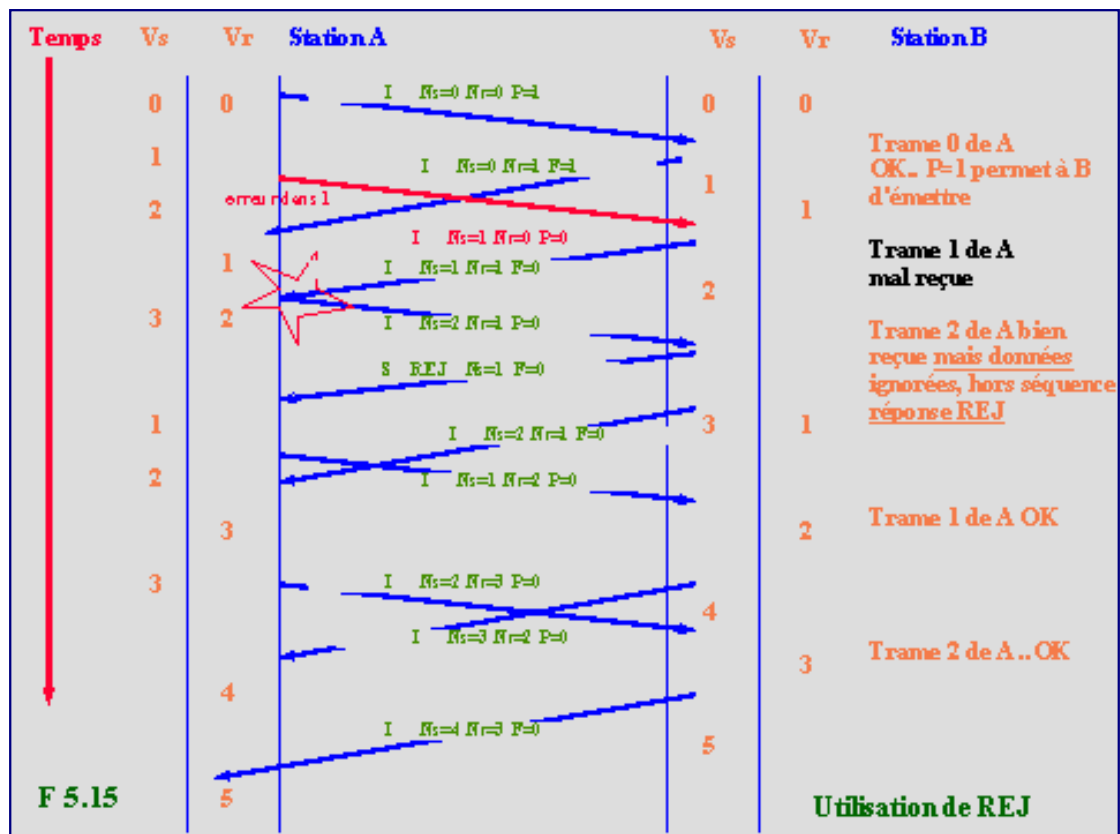
## Cas typiques liés au Bit Pool

Un réveil de type P-Bit plus court que TA existe, il est armé lors de l'envoi d'une trame de commande .. Bit Pool affirmé.

En cas de non réponse la LPDU de commande est renvoyée n fois ( P-Bit réarmé n fois ).

La connexion est alors considérée comme rompue.

## Phénomènes d'asynchronismes et REJ



## Phénomènes d'asynchronismes et REJ

La figure montre un exemple où les trames arrivent pendant une émission.

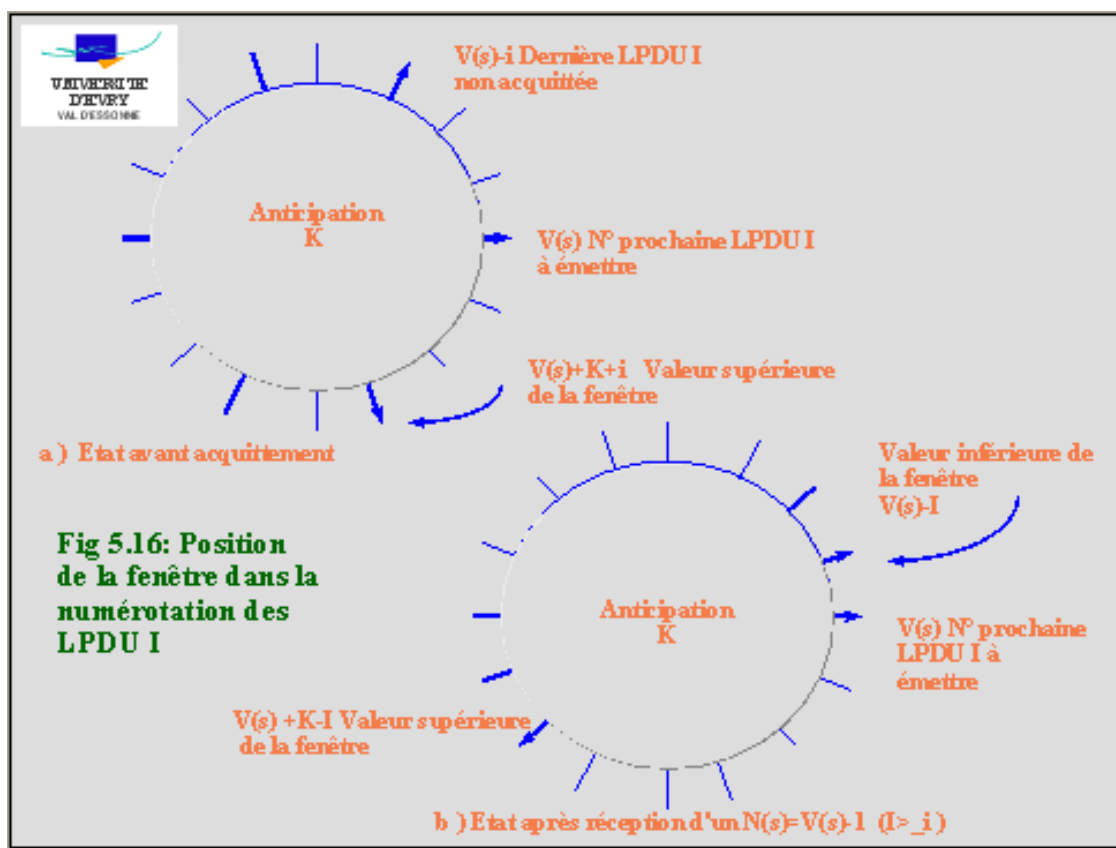
### Conclusion:

La norme décrit sous forme d'automate d'état finis le fonctionnement du protocole.  
La norme précise les réactions que le récepteur de LPDU doit avoir dans tous les cas.

## Gestion de la fenêtre d'anticipation

### Principe

Cette fenêtre indique à l'émetteur le nombre de LPDU de type INFORMATION qu'il peut soumettre avant d'exiger un acquittement



Chaque LPDU I émise = Val. fenêtre locale - 1

Chaque ACK positif Reçu = Val. fenêtre locale + 1

Ceci est insuffisant pour garantir le contrôle de flux

dans le cas où l'utilisateur ne retire pas ses données celles-ci sont conservées

L'échange des valeurs locales n'est pas permise

Ministère de l'Enseignement Supérieur et des recherches scientifiques  
Université Virtuelle de Tunis

Intitulé du chapitre :

**Technologies des réseaux de communication**

Nom de l'auteur :

**Gérard-Michel Cochard & Edoardo Berera  
& Michel Besson Thierry Jeandel**

Cette ressource est la propriété exclusive de l'UVT. Il est strictement interdit de la reproduire à des fins commerciales. Seul le téléchargement ou impression pour un usage personnel (1 copie par utilisateur) est permis.



## Module 214

---

# Réseaux et Protocoles

---

## La famille des protocoles TCP/IP

[Introduction](#)

[Le protocole IP](#)

[Les protocoles de transport](#)

[Le routage sous IP](#)

[La couche Application](#)

[Exercices](#)

[Bibliographie](#)

Thierry Jeandel

[jeandel@univ-nancy2.fr](mailto:jeandel@univ-nancy2.fr)



# Introduction

Sommaire :

[TCP/IP, un "vieux" protocole](#)

[Les couches de TCP/IP](#)

[Historique de TCP/IP](#)

[Les organismes liés au développement de TCP/IP](#)

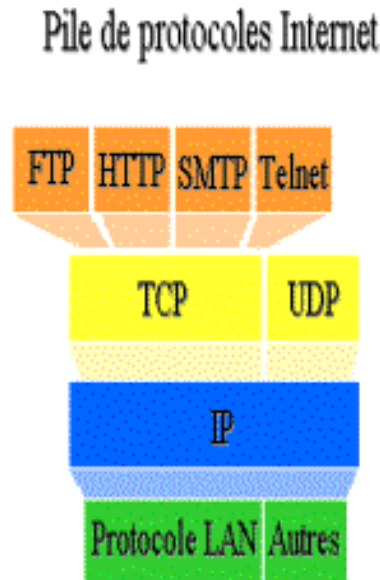
## TCP/IP, un "vieux" protocole...

25 ans ! Une éternité en informatique, c'est pourtant l'âge du protocole TCP/IP, ou plutôt des protocoles de la famille TCP/IP. En effet, TCP et IP sont les deux briques d'une famille de protocoles qui s'est enrichie au fil des années et qui comprend désormais une bonne douzaine de protocoles utilisant les services des couches IP ou TCP.

Bien sûr, en 25 ans, TCP/IP a beaucoup évolué et évolue encore en fonction des innovations technologiques et des besoins mais voir un protocole, d'abord dédié aux réseaux étendus, devenir quasiment le protocole le plus adopté par les réseaux d'entreprise est un fait suffisamment unique pour être noté et mériter quelques explications.

Une des raisons principales du succès de TCP/IP est qu'il est le **protocole du réseau internet**, il a donc ainsi profité de sa popularité, mais le protocole TCP/IP a d'autres caractéristiques fondamentales qui ont contribué à son succès :

- C'est un protocole **ouvert**. Le terme **ouvert** s'oppose à celui de **propriétaire** qui indiquerait que le protocole est lié à un constructeur, or ce n'est pas le cas de TCP/IP qui dans sa définition même n'est lié à aucun type de matériel.  
A l'origine, TCP/IP a été créé pour un système Unix, cela a d'ailleurs contribué à la popularité de ce protocole, désormais on trouve une implémentation de TCP/IP sur pratiquement tous les systèmes d'exploitation et pour pratiquement tous les types de matériel.
- C'est une famille de protocoles structurée en **couches**. L'ensemble des fonctionnalités nécessaires au bon fonctionnement d'un réseau et de toutes les applications qui s'y rapportent est hiérarchisé en un ensemble de couches dont le rôle est défini de façon précise. TCP/IP se rapproche en cela du modèle OSI de l'ISO, à la différence que TCP/IP est organisé en seulement 4 couches au lieu des 7 du modèle ISO.



- C'est un protocole **routable**, c'est à dire que des mécanismes peuvent être mise en oeuvre pour déterminer le chemin qu'un message doit prendre pour arriver à son destinataire.

## Les couches de TCP/IP

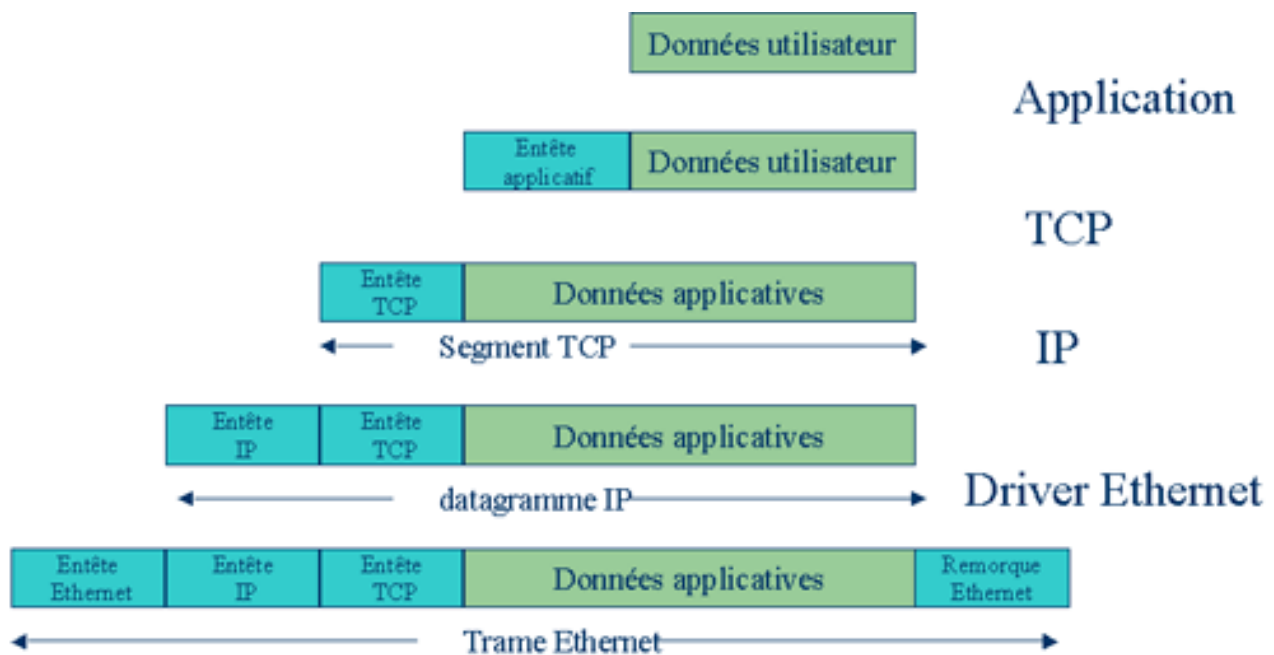
La famille de protocoles TCP/IP est ce que l'on appelle un **modèle en couche** comme il est défini dans le modèle **OSI** (Open System Interconnexion) édicté par l'**ISO** (International Standard Organisation). Mais à la différence du modèle OSI qui comprend **7 couches**, le modèle en couche de TCP/IP, qu'on appelle parfois **modèle DoD** pour se souvenir que ce modèle a été conçu pour le Department Of Defense des Etats Unis, ne comprend que **4 couches** qu'on peut définir de la façon suivante (en partant des couches les plus basses):

- La couche "**Accès au réseau**". Cette couche concerne la connexion physique proprement dite et est directement liée au type de réseau utilisé : Ethernet, réseau à jeton, etc... Cette couche peut être considérée comme la fusion des couches **Liaison de Données** et **Physique** du modèle OSI. A ce niveau on parle de **trame** d'information.
- La couche **Internet (IP)**. Cette couche est responsable de l'**adressage logique** du réseau, de l'acheminement de l'information d'un noeud du réseau à un autre. Les unités logiques d'informations véhiculées par cette couche sont appelées des **datagrammes**.
- La couche **Transport**. Cette couche, parfois appelée également **couche hôte à hôte** ou **Service Provide Layer** où l'on trouve 2 protocoles **TCP** et **UDP**, est responsable du service de transmission fiable de données. Le terme **segment** est utilisé pour désigner les paquets d'informations.
- La couche **Application**. Cette couche regroupe un ensemble d'applications liées aux réseaux TCP/IP.

On peut citer **HTTP**, le protocole du Web, mais aussi **FTP** le protocole de transfert de fichiers, **Telnet** l'émulation de terminal, etc...

Cette couche regroupe les 3 couches hautes du modèle OSI: **Application**, **Présentation** et **Session**. Les unités d'information sont appelées **messages**.

Chacune des couches intermédiaires fournit aux couches supérieures des **services** et utilisent les services de la couche inférieure, on assiste donc au niveau du format des données circulant sur le réseau à une **encapsulation** des données.



## Historique de TCP/IP

### Un enjeu stratégique

Dans les années 60, les responsables de la **DARPA** (Defense Advanced Research Projects Agency) se sont rendus compte que le parc d'ordinateurs utilisés dans le domaine militaire étant composé de machines de constructeurs différents, seuls les ordinateurs de même marque pouvaient communiquer entre eux ! De plus le système était très centralisé donc très vulnérable en cas de destruction d'un des sites or la "guerre

froide" entre USA et URSS est à son paroxysme.

Le ministère de la défense américaine (**DOD** : Departement Of Defense) demanda donc aux ingénieurs de la DARPA de mettre au point un protocole de communication qui serait indépendant du matériel.

Ce protocole devait permettre non seulement à des machines hétérogènes de dialoguer entre elles mais également de permettre de construire un réseau **non centralisé** dans lequel l'information pouvait être **distribuée**. Les informations envoyées devaient parvenir **sans perte** au destinataire, quelles que soient les pannes et les incidents rencontrés en cours de route.

## Premières expérimentations

Pour réaliser ce protocole les chercheurs utiliseront une théorie avancée par **Paul Baran** et **Donald Davis** qui avaient imaginé un protocole basé sur la "**commutation de paquets**" : le message à envoyer est découpé en paquets, paquets qui empruntent des routes différentes sur le réseau et sont reclassés à l'arrivée pour reconstituer le message initial.

La première expérimentation eu lieu en 1969, elle permis de relier 4 sites de l'ouest américain (Université de Californie à Los Angeles, Université de Californie à Santa Barbara, Université de l'Utah à Salt Lake City et le SRI International dans la Silicon Valley).

Cette expérience a marqué le début du projet **ARPAnet** (Advanced Research Projects Agency network).

Ensuite le nombre de sites connectés a rapidement augmenté, en 1972 une autre démonstration met en oeuvre 50 **noeuds** et vingt **hôtes**.

A cette époque les bases de ce protocole qu'on appelle désormais TCP/IP (Transmission Control Protocol / Internet Protocol) sont jetées par Vinton Cerf et Robert Kahn.

## Un projet universitaire

Dans les années 70, l'infrastructure d'Arpanet est mise à disposition des universités américaines par le biais de la **National Science Foundation**, le réseau s'appelle désormais **NFSNet**, il relie les principaux centres de recherches américains, les universités et quelques laboratoires de constructeurs informatiques mais le réseau est toujours **limité au territoire américain**. Le réseau permet des échanges de fichiers, de courrier électronique mais également de se connecter à distance sur d'autres ordinateurs distants.

Ce réseau va ensuite être **prolongé en Europe** puis dans la plupart des pays industrialisés. Le nom d'**INTERNET** apparaît pour la première fois en 1982, et en 1988, il devient un réseau mondial essentiellement consacré à la recherche civile.

Parallèlement au développement d'internet, le système d'exploitation Unix d'abord réservé au monde universitaire puis à celui de quelques réseaux de mainframes va connaître un essor considérable grâce à l'apparition de **Linux**, un Unix "léger", gratuit et ouvert, portable sur un micro ordinateur. Bien évidemment, Linux intègre le protocole TCP/IP.

## La naissance du Web

En 1989, un chercheur du **CERN** (Centre Européen de Recherche du Nucléaire) de Genève, **Tim Berners-Lee** crée, en 1989, le concept de **World Wide Web**. S'appuyant sur la technologie internet, le Web est un univers d'informations reliées par des liens dits **hypertextes**. L'hypertexte permet, grâce à un simple clic

de souris, de passer d'un texte à un autre, de faire apparaître une image, entendre un document sonore, de visionner une vidéo, etc.

Ensuite c'est le début de la démocratisation d'internet avec l'apparition des premiers navigateurs Mosaic d'abord, ensuite Netscape et Internet Explorer.

## La consécration de TCP/IP

Depuis le milieu des années 90, l'essor d'internet et le besoin de connecter la plupart des réseaux locaux à ce réseau mondial font que le protocole TCP/IP est en train de s'imposer comme un standard de fait. Pour satisfaire à de nouveaux besoins : nombre croissant d'ordinateurs connectés, nature des données qui circulent sur les réseaux (voix, image, etc...), gestion des priorités, etc, le protocole IP est en train de subir des modifications profondes concrétisées par la sortie du protocole **IP version 6**.

## Les organismes liés au développement de TCP/IP

### L'IAB

Les règles de fonctionnement de l'internet et des protocoles qui le régissent n'appartiennent à aucune société privée, ils sont accessibles à tous. L'organisme chargé de superviser le développement des protocoles de l'internet s'appelle l'**IAB** (Internet Architecture Board). L'IAB fait partie d'une instance beaucoup plus large qui s'appelle l'[Internet Society](#). L'[IAB](#) comprend deux groupes dont les rôles sont distincts :

- l'IRTF (Internet Research Task Force) qui est dédié à la recherche à long terme sur le devenir du réseau
- l'IETF (Internet Engineering Task Force) le groupe de développement chargé de mettre au point les standards

A côté de ces 2 instances, existent 2 "Steering Group" qui supervisent les travaux des 2 "Task Force" cités précédemment :

- l'IRSG (Internet Research Steering Group) qui supervise l'IRTF
- l'IESG (Internet Engineering Steering Group) qui supervise l'IETF

### Les RFC

Ces organismes ont un mode de fonctionnement assez original, il repose sur un principe très démocratique et un refus de tout ce qui ressemble à des monopoles. Chacun publie sa solution et celle qui marche le mieux se développe sous l'effet du ralliement des utilisateurs. Le processus d'établissement des standards

repose donc sur les **RFC (Request For Comment)**. Cette technique et cette idée de dire "SVP commentez cette proposition et dites nous ce que vous en pensez" est au cœur de l'idéologie et de la synergie qui ont fait d'Internet ce qu'il est.

Les protocoles qui peuvent devenir des standards passent donc par une série d'états qui sont la **proposition**, le **brouillon** puis enfin le **standard**. Au fur et à mesure de l'évolution, certains protocoles sont classés **historique**.

La liste complète des RFC peut être obtenue sur le site <http://www.rfc-editor.org/>

# Le Protocole IP

Sommaire :

[Adressage IP](#)  
[Structure du datagramme IP](#)  
[Sous-réseau et sur-réseau](#)  
[Protocole de résolution d'adresse](#)  
[IPv6](#)

## Adressage IP

[Généralités](#) - [Adresses spéciales](#) - [Affectation des adresses](#)

Cette partie concerne uniquement le format d'adressage de la version 4 du protocole IP : IPv4. Prochainement et progressivement va se mettre en place une autre version IP : IPv6 avec un système d'adressage complètement différent..

### Généralités sur l'adressage

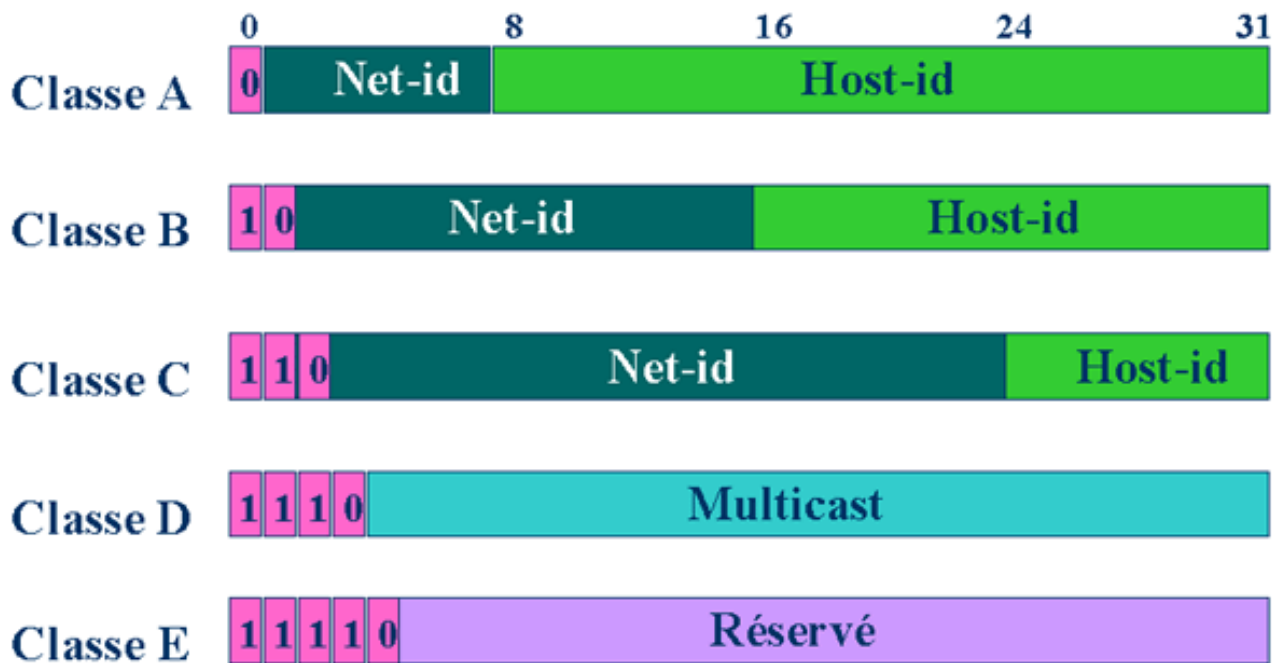
L'adressage IP est un adressage **logique** totalement indépendant des adresses de la couche physique comme les adresses MAC par exemple, cette indépendance permet à un réseau IP d'interconnecter des équipements hétérogènes. Une opération de conversion entre les adresses physiques et les adresses logiques est donc indispensable, cette opération est généralement désigné par le terme **mapping**.

Un adresse IP est une séquence de **32 bits**, ce qui devrait en principe nous donner  $2^{32}$  connexions possibles c'est à dire un peu plus de **4 milliards d'adresses** (4 294 967 296 pour être exact), en fait, certaines adresses sont exclues ou réservées ce qui fait que le nombre maximal effectif de connexions est moindre .

L'adressage IP reflète, de par sa structure, la distinction entre les différents réseaux logiques. En effet un certain nombre de bits de l'adresse IP identifie le réseau lui même (**netid**), l'autre partie identifie l'hôte dans ce réseau (**hostid**). Ce découpage netid - hostid constitue donc un **plan d'adressage hiérarchique** pour un réseau IP, ce qui permet une meilleure gestion des routeurs qui n'ont besoin que de mémoriser des adresses de réseaux et non des adresses d'hôtes. Il va sans dire que des réseaux interconnectés doivent avoir des netids distincts.

Cette structuration est différente selon la classe du réseau. On distingue **5 classes de réseaux** codée de A à E. La distinction de classe de réseaux se fait sur la valeur des premiers bits. Pour les classes A, B et C, la taille de la partie d'adresse réservée au net-id varie, elle est de 1 octet pour la classe A, 2 pour la classe B et 3 pour la classe C.

La classe D est réservée à la multidiffusion (multicast), technique utilisée par des protocoles spéciaux pour transmettre simultanément des messages à un groupe donné de noeuds différents, de la diffusion de vidéo par exemple. La classe E était réservé à un usage ultérieur.



### Exercices et tests : [Exercice 1](#)

Pour des raisons de commodités, les adresses IP sont rarement exprimées en binaire mais en **notation décimale pointée**. Chaque octet est traduit en sa valeur décimale et les 4 nombres sont séparés par des points.

**Exemple :** 137.65.4.1. correspond à l'adresse 10001001 01000001 00000100 00000001

### Les adresses spéciales

Il existe un certain nombre d'adresses IP réservées :

- **hostid = 0** désigne le **réseau** lui-même

L'hostid égal à 0 ne sera jamais affecté à un hôte mais il désigne le réseau lui-même.

Exemple : 192.145.56.0 est un réseau de classe C dont l'hostid est à 0 donc cette adresse désigne le réseau lui-même.

- **0.0.0.0** désigne l'**hôte** lui-même

Lorsque tous les bits d'une adresse IP sont à 0, cela signifie "cet hôte-ci sur ce réseau". Cette adresse spéciale est par exemple utilisée par un hôte afin d'obtenir une adresse IP de manière dynamique dans le cas du protocole BOOTP.

- Tous les bits de l'**hostid** = 1 indique une **diffusion dirigée**

Lorsque tous les bits de l'hostid sont égaux à 1, on est en présence non pas d'une adresse d'hôte mais d'une adresse de **diffusion dirigée (direct broadcast)** c'est à dire un message destiné à tous les hôtes d'un réseau sans exception.

Exemple : 192.145.56.255 est une adresse de classe C dont la partie réservée à l'hostid est égale à 255 donc pour laquelle tous les bits sont à 1, on est donc en présence d'un message destiné à l'ensemble des hôtes du réseau 192.145.56.0.

- **255.255.255.255** = **diffusion limitée**



Une **diffusion limitée (limited broadcast)** est un message qui est envoyé à tous les hôtes du réseau dont fait partie l'expéditeur. La diffusion limitée est représentée par l'adresse spéciale 255.255.255.255.

Exemple : L'adresse de destination 255.255.255.255 indique que le message doit être envoyé à tous les hôtes du réseau dont fait partie l'expéditeur.

- **netid = 0** indique que l'hôte fait partie du réseau

Lorsque que la partie netid est égale à 0 et que la partie hostid est non nulle, cela signifie qu'on est en présence d'un message issu du même réseau.

Exemple : Si un hôte d'adresse 192.14.25.56 reçoit un paquet à destination de 0.0.0.56, il considérera que ce paquet lui est bien destiné.

- **127.x.x.x** = adresse de **bouclage**

Le netid 127.0.0.0 qui devrait normalement faire partie de la classe A est en fait utilisé pour désigner l'**adresse de bouclage (loopback)**, peut importe le hostid utilisé. Un paquet envoyé à cette adresse ne passe pas par les interfaces réseau mais est déposé directement sur le tampon de réception de la machine elle même. Cette adresse de bouclage permet de vérifier la configuration de la couche logicielle TCP/IP d'une machine.

Exemple : 127.0.0.1 désigne l'adresse de bouclage sur la machine elle même.

**Exercices et tests :** [Exercice 2](#) , [Exercice 3](#)

## Affectation des adresses IP

L'affectation d'une adresse IP à un réseau ne peut pas se faire n'importe comment car pour que le système fonctionne il ne faut absolument pas que 2 hôtes puissent avoir une adresse IP identique sinon c'est tout le système d'adressage qui s'écroule.

L'attribution d'un netid à un réseau est donc soumis à une autorité centrale. C'est l'[Address Supporting Organization](#) (ASO) qui est responsable de l'allocation des adresses IP (et de tous les identifiants uniques) dans l'Internet. Cet organisme a délégué cette responsabilité à des organismes régionaux comme [ARIN](#) aux USA, le [RIPE NCC](#) en Europe, et l'[APNIC](#) en Asie.

Ces organismes délèguent à nouveau à d'autres organismes : les "**local registries**". Il existe trois types de "local registries" en Europe définis par le RIPE NCC:

- Les "**provider local registries**"  
Ce sont des "local registries" qui allouent des adresses IP pour les clients d'un fournisseur de service particulier. C'est typiquement le cas des fournisseurs de service. Ces "local registries" se voient allouer des plages d'adresses qu'il peuvent ensuite "redistribuer" à leur clients.
- Les "**enterprise registries**"  
Ce sont des "local registries" qui allouent des adresses IP à l'intérieur d'une entreprise donnée.
- Les "**last resort registries**"  
Ce sont des "local registries" qui allouent des adresses IP si le demandeur ne peut être servi par l'un des deux types de "local registries" ci-dessus. Ces "local registries" sont actuellement en voie de disparition car l'allocation d'adresses par eux ne permet pas d'obtenir une bonne agrégation des tables de routages indispensable au bon fonctionnement de l'Internet.

Les adresses de classe C ont été divisées en 8 blocs qui correspondent à peu près à des zones géographiques :

192.0.0 - 193.255.255	Plusieurs régions, ces adresses ont été allouées avant la répartition régionale
194.0.0 - 195.255.255	Europe
196.0.0 - 197.255.255	Utilisées lorsqu'il est nécessaire d'affecter des adresses IP qui ne sont pas basées sur la région
198.0.0 - 199.255.255	Amérique du Nord
200.0.0 - 201.255.255	Amérique centrale et Amérique du Sud
202.0.0 - 203.255.255	Zone Pacifique
204.0.0 - 205.255.255	Utilisées lorsqu'il est nécessaire d'affecter des adresses IP qui ne sont pas basées sur la région

206.0.0 - 207.255.255	Utilisées lorsqu'il est nécessaire d'affecter des adresses IP qui ne sont pas basées sur la région
208.0.0 - 223.255.255	Disponible

Mais on peut être amené à utiliser le protocole TCP/IP sans être connecté à Internet ou en étant connecté à Internet via une passerelle applicative (un serveur **Proxy** ou un **Firewall** par exemple), on parle alors de **réseaux privés**. Pour ces réseaux privés, il est prévu d'utiliser des plages d'adresses spécifiques (qui par ailleurs ne sont jamais affectés par l'ASO) :

- 10.0.0.0 à 10.255.255.255 pour un réseau de classe A
- 172.16.0.0 à 173.31.255.255 pour des réseaux de classe B
- 192.168.0.0 à 192.168.255.255 pour des réseaux de classe C

Remarque : Le **RIPE NCC** donne accès à sa base de donnée (par l'intermédiaire du site <http://www.ripe.net/perl/whois?> ) afin d'associer à une adresse IP des informations comme les coordonnées de l'instance responsable de cette adresse IP, l'intervalle d'adresses allouée, etc...

Remarque : Cela ne fonctionne qu'avec les adresses IP gérée par le RIPE NCC donc uniquement en Europe:

Donner une  
adresse IP :

## Structure du datagramme IP

### Généralités - Datagramme IP - Fragmentation des datagrammes

#### Généralités

La couche IP est une **couche intermédiaire** dans la vision globale d'un réseau. En effet si on se réfère au modèle OSI, la couche IP correspond à la **couche "réseau"** (couche 3) de la pile de protocoles. Cette position intermédiaire signifie donc que le protocole IP va, à la fois fournir des services aux couches supérieures (TCP et UDP) et utiliser des services des couches inférieures (Ethernet, token Ring, X25...).

Le protocole IP est un **protocole sans connexion** ce qui signifie que chaque **datagramme** va être étiqueté avec l'**adresse de l'expéditeur** et du **destinataire**. Dans ce datagramme on va également trouver un certain nombre d'informations concernant l'acheminement du datagramme comme le **nombre de routeurs** traversés par exemple. La version 4 de IP prévoit même la possibilité d'insérer dans les datagrammes des informations de "**qualité de service**" mais elles sont en fait peu utilisés, c'est dans la version 6 que cette technique est réellement développée.

Le réseau IP est un réseau **abstrait** qui fait abstraction des problèmes matériels gérés par les couches inférieures, chaque datagramme pourra donc éventuellement être **routé par des chemins différents**, ce qui ne garanti pas qu'à l'arrivée les datagrammes soient dans le même ordre qu'au départ, c'est pourquoi un mécanisme de **numérotation de datagramme** est mis en oeuvre. La remise en ordre des différents datagrammes n'incombe pas au protocole IP mais aux protocoles supérieures (TCP, UDP).

## Le Datagramme IP

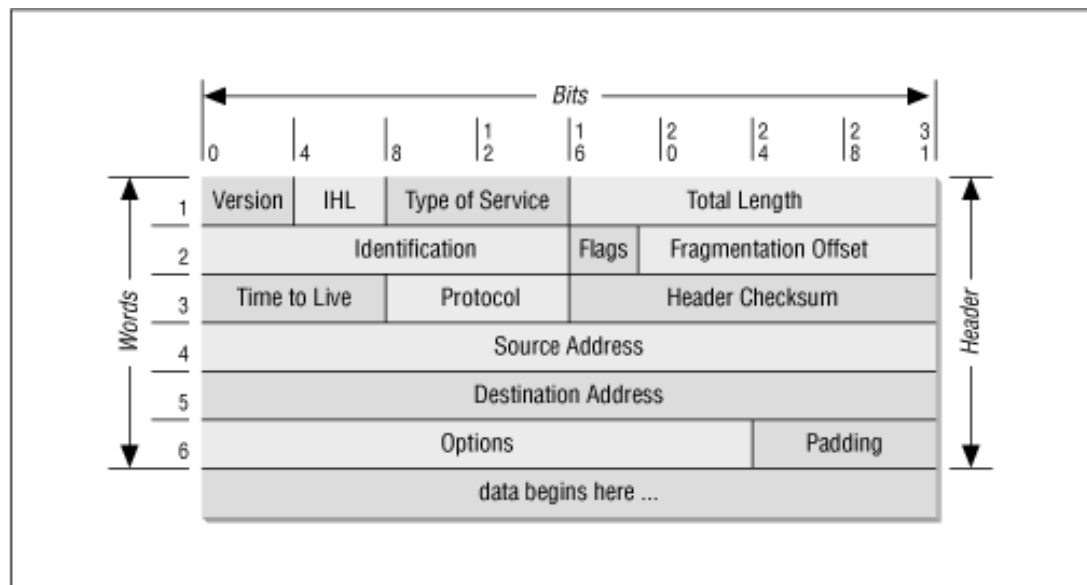
Le datagramme IP est divisé en 2 parties :

- **L'entête IP**  
Cette partie va contenir des informations essentielles comme les adresses des destinataire et expéditeur, des informations sur la taille du datagramme, des informations sur l'éventuelle fragmentation du datagramme, la qualité de service attendue par les couches supérieures, etc...
- **Les Données IP**  
Ces données sont celles qui ont été transmises par les couches supérieures.

### L'entête IP

L'entête IP a une longueur minimale de 20 octets mais il peut être plus grand. On va trouver dans cet entête un certain nombre d'informations essentielles (structurées en champ) pour l'acheminement de l'information :

- **Version** : n° de version du protocole
- **IHL** (Internet Header Length) : longueur de l'entête
- **TOS** (Type of Service) : qualité de service désirée
- **Total Length** : Longueur totale du datagramme
- **Identification** : N° du datagramme
- **Flags** : Drapeau de fragmentation
- **Fragment Offset** : Décalage du fragment
- **TTL** (Time To Live) : Durée de Vie du datagramme
- **Protocol** : protocole supérieur
- **Header Checksum** : Contrôle d'erreur de l'entête
- **IP source et destination** : Adresses expéditeur et destinataire
- **Options** : Options du protocole
- **Padding** : Remplissage



*Cliquer sur une zone pour obtenir le descriptif du champ.*

#### Version (4 bits)

Le champ version est utilisé pour indiquer le **numéro de version de protocole** utilisé (4, 6 etc...). Le logiciel Ip commencera par vérifier la valeur de ce champ pour s'assurer que le datagramme peut être traité. Les valeurs possibles vont de 0 à 15 avec 4 pour IPv4, 6 pour IPv6 et d'autres valeurs pour des versions moins connues comme TP/IX ou TUBA.

[Retour au schéma de l'entête](#)

#### IHL Internet Header Length (4 bits)

**Longueur de l'entête** en mots de 32 bits : Tous les champs de l'entête sont de longueur fixe sauf le champ Options qui lui est de longueur variable, la longueur de l'entête doit donc être précisée, c'est à cela que sert ce champ.

[Retour au schéma de l'entête](#)

### TOS Type Of Service (8 bits)

Ce champ permet d'indiquer des informations liées à la **qualité de service** désiré en terme de **priorité** (préséance), de **délais**, de **débit**, de **fiabilité** et de **coût**.

Le bit de préséance lié à la priorité était à l'origine destiné aux applications du ministère de la défense américain, les valeurs possibles sont étiquetées de la manière suivante :

- Flash : priorité maximale sur tous les circuits
- Immediate : Dans les 4 heures
- Priority : Le même jour
- Routine : D'ici au lendemain

Les 4 drapeaux délai, débit, fiabilité et coût peuvent prendre alternativement les valeurs Normal (0) ou Elevé (1). Exemple la suite de bit 0010 correspond à "fiabilité maximale".

La plupart des applications ignorent la valeur de ce champ TOS mais il semble que ce champ puisse jouer un rôle important dans certains protocoles de routage. Par exemple si un routeur connaît 2 chemins possibles pour acheminer un datagramme : l'un à haut débit mais délai important comme une connexion satellite par exemple et l'autre une connexion à délai faible et débit faible également (ligne spécialisée). Si l'application nécessite un fort volume de transfert de données, la liaison satellite sera choisie, si, par contre, c'est une application interactive qui nécessite des saisies de touches utilisateur, c'est la ligne spécialisée qui doit être utilisée.

[Retour au schéma de l'entête](#)

### Total Length (16 bits)

Ce champ contient la **longueur totale du datagramme**, le datagramme peut donc faire au maximum  $2^{16}$  octets - 1 c'est à dire 65 535 octets de long, ce qui est rarement le cas car les couches inférieures ne sauraient pas traiter des paquets de données aussi importants. La longueur minimale est de 576 octets (512 octets de données + 64 octets d'entête).

[Retour au schéma de l'entête](#)

### Identification (16 bits)

Ce champ correspond au **numéro de datagramme**. Chaque datagramme est numéroté par l'expéditeur en partant d'une certaine valeur initiale. La codification sur 16 bits permet de numéroté les datagrammes jusque 65535.

[Retour au schéma de l'entête](#)

### Flags (3 bits)

On va trouver 2 **drapeaux** sur ces 3 bits (le premier bit étant toujours à 0) :

- DF (Flag de fragmentation) : Ce drapeau s'il est à 1 indique que le datagramme ne doit pas être fragmenté.
- MF (More fragment) : Si ce drapeau est à 1 cela signifie que le datagramme est fragmenté et que d'autres fragments vont arriver, si ce flag est à 0 cela signifie que c'est l'unique ou le dernier fragment.

[Retour au schéma de l'entête](#)

### Fragment Offset (13 bits)

Ce champ indique la **position des données du fragment** par rapport au début du datagramme originel.

[Retour au schéma de l'entête](#)

### TTL Time To Live ( 8 bits)

Ce champ "Durée de Vie" représente la **durée de vie maximale d'un datagramme** sur le réseau, cette durée est exprimée en secondes. Le principe est d'éviter que des datagrammes errent indéfiniment sur le réseau. Cette "durée de vie" est décrémentée à chaque routeur de la durée nécessaire à son traitement, mais en fait comme l'évaluation de cette durée nécessiterait un traitement supplémentaire, les routeurs se contentent de décrémenter de 1 le TTL, on considère que ce TTL correspond finalement à un **compteur de sauts (hops)**.

Lorsque le TTL tombe à 0, le datagramme est détruit par le routeur.

[Retour au schéma de l'entête](#)

### Protocol (8 bits)

Ce champ indique à quel **protocole de couche supérieure** ce datagramme est destiné. Par exemple la valeur de Protocol est 6 pour TCP, 17 pour UDP, 1 pour ICMP, etc...

[Retour au schéma de l'entête](#)

### Header Checksum (16 bits)

**Champ de contrôle de l'entête.** Le complément à 1 de chaque valeur de 16 bits de l'entête est ajouté (sauf le champ Checksum Header). On prend alors le complément à 1 de cette somme que l'on code dans ce champ. Ce champ est donc **recalculé** à chaque routeur puisque le TTL est décrémenté.

[Retour au schéma de l'entête](#)

### IP source et destination (2 x 32 bits)

Le réseau IP est un **réseau sans connexion** donc sur chaque datagramme doivent figurer les adresses source et destination.

[Retour au schéma de l'entête](#)

### Options IP ( taille variable, multiple de 32 bits)

Ce champ correspond à des informations qui sont assez peu utilisées dans le protocole IP et qui concernent la sécurité, l'enregistrement du chemin emprunté au travers des routeurs, l'obligation d'emprunter une certaine route, etc...

[Retour au schéma de l'entête](#)

### Padding

Champ de **remplissage** lié au champ Options IP qui permet d'avoir un multiple de 32 bits.

[Retour au schéma de l'entête](#)

Exercices et tests : [Exercice 4](#)

## La fragmentation des datagrammes

Le protocole IP peut être utilisé en réseau hétérogène, c'est à dire qu'un datagramme peut traverser des réseaux de nature différentes (Ethernet, Token Ring, X25..), or ces réseaux ont des caractéristiques totalement différentes notamment ne ce qui concerne leur MTU, c'est à dire la taille de l'unité de transmission maximale.

Type de réseau	MTU en octets

Ethernet	1500
Token Ring	4 440 à 17 940
FDDI	4 352
X 25	1 007

### Quelques valeurs de MTU de réseaux courants

Le problème survient donc si la taille d'un datagramme IP est plus grand que le MTU du réseau emprunté, dans ce cas le datagramme doit être découpé en **fragments**. C'est le **routeur qui va effectuer cette opération** et non l'émetteur du datagramme initial puisque ce dernier ignore la structure des réseaux traversés. Chaque fragment va être traité comme un datagramme IP mais à l'arrivée les différents fragments devront être réassemblés.

## Sous-réseaux et sur-réseaux

[Généralités](#) - [Pourquoi des sous-réseaux](#) - [Principe du subnetting](#) - [Masques de sous-réseau](#) - [Sur-réseaux](#)

### Généralités

Lorsque le protocole IP a été mis au point, les nombres de réseaux et d'hôtes potentiellement adressables par le système d'adressage mis en place semblaient surdimensionnés par rapport aux besoins de l'époque, désormais après le développement fulgurant d'Internet ce système a montré ses limites. En effet le système d'adressage divisé en classes a pour effet de générer un gâchis considérable d'adresses.

C'est essentiellement pour remédier à ce gaspillage d'adresses que dès **1985** un mécanisme de sous-réseau est apparu, le principe est de diviser un réseau en plusieurs **sous-réseaux** interconnectés entre eux.

Plus récemment le concept de **surréseau** que l'on désigne également par **CIDR (Classless Internet Domain Routing)** a vu le jour. Le principe est de combiner plusieurs numéros de réseaux (généralement de classe C) pour former un réseau plus important.

### Sous-réseaux

#### Pourquoi des sous-réseaux ?

Dans le mécanisme d'adressage IP on distingue principalement 3 classes de réseaux (A,B et C) pour lesquelles chaque réseau disposera d'un nombre maximum d'adresses (cf Adressage IP). Par exemple pour un réseau de classe C, c'est 254 adresses qui seront utilisables puisque l'adresse de l'hôte est codée sur un octet ce qui donne donc 256 adresses possibles auxquelles on soustrait les adresses spéciales (255 et 0) qui ne peuvent être allouées à un hôte.

Mais il arrive fréquemment qu'on veuille fragmenter un réseau pour différentes raisons :

- La première motivation de fragmentation d'un réseau peut être la **simplification de l'administration**. En effet, l'installation de routeurs et le partitionnement en sous-réseaux d'un réseau va permettre d'administrer les sous-

réseaux de façon indépendante. Par exemple, le plan d'adressage de chaque sous-réseau pourra se faire de façon totalement indépendante.

- Une seconde motivation peut être également de faire une **économie d'adresses**. Prenons par exemple le cas d'une entreprise qui dispose d'un réseau de classe B, ce qui lui offre un potentiel de 65534 adresses d'hôtes. Il y beaucoup de chances qu'une proportion assez faible de ces adresses soit réellement employée. Si cette même entreprise désire installer un second réseau séparé physiquement du premier par un routeur elle a le choix entre faire l'acquisition d'un nouveau numéro de réseau ce qui est dommage puisqu'il lui reste un potentiel d'adresses non utilisées. Dans ce cas de figure, la solution du partitionnement en sous-réseau est bien meilleure.
- Enfin, le dernier avantage de la fragmentation en sous-réseaux est l'aspect **sécurité**. En effet, il sera plus facile d'isoler certains hôtes du réseau s'ils se trouvent dans un sous-réseau relié par un routeur au reste du réseau que si toutes les machines se trouvent sur un même réseau.

### Principe du subnetting

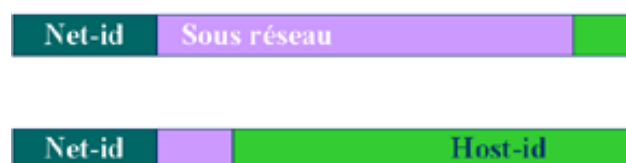
Le subnetting est le terme anglo-saxon qui désigne la **fragmentation en sous-réseau**. Le principe est d'introduire en plus des notions de netid et d'hostid, la notion de **sous-réseau**. Or une adresse IP a une taille fixe de 32 bits donc la technique consiste à "prendre" les bits nécessaires au codage du sous-réseau sur la partie réservée à l'hostid.



Il sera donc possible d'utiliser un nombre variable de bits pour caractériser le sous-réseau.

Par exemple pour un réseau de classe C d'adresse 192.47.56.0, on peut décider d'utiliser 3 bits pour caractériser le sous-réseau, ce qui laissera seulement 5 bits pour adresser les hôtes dans chacun des sous-réseaux. En respectant les règles d'adressage vues précédemment cela donne donc  $2^5 - 2 = 30$  adresses d'hôtes possibles par sous-réseau. Par contre l'utilisation de 3 bits par sous-réseau permet en théorie l'utilisation de  $2^3 = 8$  sous-réseaux, mais comme pour les netid les numéros de sous-réseau composés uniquement de 1 ou uniquement de 0 ne sont pas autorisés donc cela fait seulement 6 sous-réseaux possibles.

Lors de la structuration en sous-réseaux, il faudra donc trouver un **équilibre** entre un grand nombre de sous-réseaux avec peu d'adresses possibles par sous-réseau d'une part et peu de sous-réseaux et un nombre d'adresses important.



Enfin, on peut remarquer que la mise en sous-réseau a pour inconvénient de diminuer l'espace d'adressage d'un réseau puisque dans notre exemple on est passé pour ce réseau de classe C de 254 adresses d'hôtes possibles à seulement  $6 \times 30 = 180$  adresses potentielles.

Exercices et tests : [Exercice 5](#), [Exercice 6](#)

### Les masques de sous-réseaux

Un **masque** de sous-réseau est une suite de 32 bits (comme pour les adresses IP) qui va permettre à un hôte (un ordinateur, un routeur) de déterminer si un hôte dont on connaît l'adresse IP est accessible directement (car l'hôte se trouve dans le même sous-réseau ou le même réseau si aucun découpage n'a été effectué) ou bien s'il faut passer par un routeur (qui se nomme souvent **passerelle par défaut** ou **default gateway** dans les logiciels).

**Tout hôte IP a donc besoin d'un masque**, que le réseau soit découpé en sous réseau ou non. Ce masque est construit de la façon suivante : Les **bits du masque à 1 correspondent à la partie d'adresse qui doit être identique** pour faire partie du même sous-réseau ou réseau si celui-ci n'est pas segmenté, les bits à 0 correspondent à la partie d'adresse qui varie d'un hôte à l'autre.

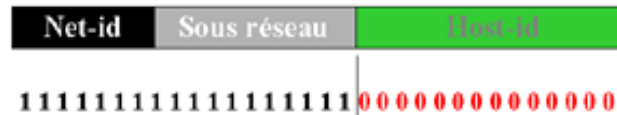
Pour un réseau qui n'est pas segmenté en sous-réseau, le masque standard se déduit aisément puisque les bits à 1 correspondent



à la partie réservée au net-id :

Classe de réseau	Masque standard
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

Pour les hôtes se trouvant dans un sous-réseau, le masque va dépendre du nombre de bits réservés au sous-réseau. Ce nombre de bits devra être forcément supérieur à 1, car sur un bit on ne pourrait coder que 2 sous-réseau de n° 1 et 0 ce qui est déconseillé par les différentes RFCs.



### Exemple :

Soit un réseau de classe C d'adresse 192.47.56.0, dans lequel on a décidé de laisser 3 bits pour le sous-réseau, le masque aura la valeur suivante : 11111111 11111111 11111111 11100000, ce qui donne en notation décimale pointée 255.255.255.224. Dans ce réseau, on pourra donc disposer de 6 sous-réseaux : 001, 010, 011, 100, 101 et 110.

Si on considère la machine 192.47.56.57, cette machine se trouve dans le sous-réseau 001 car 57 en binaire donne **0011 1001**, prenons l'adresse 192.47.56.37, d'après le masque on en déduit qu'elle se trouve dans le même sous-réseau que notre hôte car 37 en binaire donne **0010 0101**, par contre les adresses suivantes ne sont pas accessibles directement :

- 192.47.56.66 car pas dans le même sous-réseau mais dans le sous-réseau **010**, car 66 = **0100 0010**, **010** est différent de **001**
- 192.47.41.24 car pas dans le même réseau (41 différent de 56)
- 192.47.56.12 car adresse illégale (le sous réseau **000** ne peut être attribué), car 12=**0000 1100**

### Exercices et tests : [Exercice 7](#)

### Sur-réseaux

Il y a depuis pas mal de temps pénurie de réseaux de classe A et de classe B, par contre il reste encore un grand nombre de réseaux de classe C disponibles. Mais l'inconvénient majeur de ces réseaux de classe C est qu'ils ne peuvent prendre en charge que 254 hôtes. Les organisations de grande taille se voient donc obligées d'utiliser plusieurs réseaux de classes C.

Le principe du **sur-réseau** est donc d'utiliser un ensemble d'adresses **de réseaux de classe C contiguës** au lieu d'un seul réseau de classe B. Par exemple, une entreprise a besoin d'un réseau de 8 000 hôtes, il est donc possible d'utiliser 32 réseaux de classe C, car  $32 \times 254 = 8128$ . Ce mécanisme est très utilisé par les fournisseurs d'accès internet qui ont besoin d'un volant important d'adresses IP.

Si on voit bien l'avantage d'utiliser plusieurs réseaux de classe C à la place d'un réseau de classe B, un problème va résider au niveau des routeurs. En effet, pour avoir la même capacité d'adressage qu'un réseau de classe B, il faut 256 adresses de réseaux de classe C. La place nécessaire au stockage de ces adresses dans la mémoire d'un routeur se trouve donc multiplié par 256 !.

La technique **Classless Internet Domain Routing (CIDR)** permet de n'utiliser qu'une seule entrée pour un bloc d'adresses de réseau de classe C. Le bloc de classe C sera décrit en donnant la plus basse adresse du bloc suivie d'un **masque de sur-réseau**. Le masque de sur-réseau est construit en mettant des 1 pour la partie commune à tous les sous-réseau et des 0 pour la partie variable. On peut remarquer qu'à l'inverse des sous-réseaux, les bits à 1 sont pris sur la partie netid de l'adresse.

**Exemple :** (198.24.32.0, 255.255.224.0) désigne un bloc de sous réseaux de classe C. Afin de déterminer lequel, il faut exprimer ces 2 données en binaire :

$$\begin{aligned}
 198.24.32.0 &= 11000110\ 00011000\ 00100000\ 00000000 \\
 255.255.224.0 &= 11111111\ 11111111\ 11100000\ 00000000
 \end{aligned}$$



La partie commune aux différents réseaux est indiquée en rouge sur la figure précédente, on peut donc déterminer les adresses de réseaux faisant partie de ce bloc :

Première adresse : **11000110 00011000 00100000 00000000** = 198.24.32.0

Dernière adresse : **11000110 00011000 00111111 00000000** = 198.24.63.0

On utilise également une autre notation pour désigner ces blocs CIDR : **plus basse adresse du bloc / nombre de bits de préfixe commun** ce qui donne pour l'exemple précédent : 198.24.32.0/19

## Protocoles de résolution d'adresses

[Généralités](#) - [ARP](#) - [RARP](#)

### Généralités

L'adressage IP est un adressage logique qui utilise dans la version 4 de ce protocole une adresse sur 32 bits. Néanmoins la transmission des informations va se faire à l'aide des couches basses, les datagrammes IP seront donc encapsulés dans des trames de la couche 2 du modèle OSI telle que les trames Ethernet, Token ring, etc...

Ces réseaux physiques utilisent un système d'adressage physique, il faudra donc faire la correspondance entre les adresses du protocole IP et les adresses physiques (adresse **MAC : Medium Access Control**) des équipements correspondants, c'est le rôle du protocole **ARP**.

Plus rarement, il sera nécessaire de faire la correspondance entre une adresse physique MAC et une adresse logique IP, le protocole **RARP** assure ce rôle.

### Le protocole ARP

ARP signifie **Address Resolution Protocol**, son rôle est de mettre en place un mécanisme de restitution d'une adresse MAC à partir d'une adresse IP dans le cadre d'un réseau où le support est partagé comme le réseau Ethernet par exemple.

Le principe de fonctionnement du protocole ARP est le suivant :

Supposons que la machine A de numéro IP 193.54.41.38 et d'adresse MAC 00:b0:d0:65:dd:19 désire connaître l'adresse MAC de l'hôte B d'adresse 193.54.41.42 afin de lui envoyer un message.

1. La machine A va émettre sur le support une trame MAC de diffusion appelée **trame ARP Request**. Dans cette trame figurent les adresses IP et MAC de l'émetteur A et l'adresse IP de l'hôte B.

```
■ Frame 31 (60 on wire, 60 captured)
  Arrival Time: Nov  5, 2002 14:46:57.535711000
  Time delta from previous packet: 0.000244000 seconds
  Time relative to first packet: 3.695690000 seconds
  Frame Number: 31
  Packet Length: 60 bytes
  Capture Length: 60 bytes
  Ethernet II
    Destination: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
    Source: 00:b0:d0:65:dd:19 (rgmi205dp.plg.univ-nancy2.fr)
    Type: ARP (0x0806)
    Trailer: 00000000000000000000000000000000...
  Address Resolution Protocol (request)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (0x0001)
    Sender MAC address: 00:b0:d0:65:dd:19 (rgmi205dp.plg.univ-nancy2.fr)
    Sender IP address: rgmi205dp.plg.univ-nancy2.fr (193.54.41.38)
    Target MAC address: 00:00:00:00:00:00 (rgmi205dc.plg.univ-nancy2.fr)
    Target IP address: rgmi208tj.plg.univ-nancy2.fr (193.54.41.42)
```

**Trame ARP Request :** L'adresse Ethernet de destination est à ff:ff:... ce qui indique une diffusion, le champ Target MAC Adress est vide.

**2.** Tous les noeuds du réseau physique reçoivent cette trame ARP. Chacun des noeuds compare l'adresse IP destinataire figurant dans cette trame avec la sienne.

**3.** La machine B a reconnu son adresse IP dans le champ destinataire, elle répond donc directement à A (car elle connaît son adresse MAC) en encodant sa propre adresse MAC dans une **trame ARP Reply**.

```
■ Frame 32 (42 on wire, 42 captured)
  Arrival Time: Nov  5, 2002 14:46:57.535750000
  Time delta from previous packet: 0.000039000 seconds
  Time relative to first packet: 3.695729000 seconds
  Frame Number: 32
  Packet Length: 42 bytes
  Capture Length: 42 bytes
  Ethernet II
    Destination: 00:b0:d0:65:dd:19 (rgmi205dp.plg.univ-nancy2.fr)
    Source: 00:c0:4f:0c:71:8c (rgmi208tj.plg.univ-nancy2.fr)
    Type: ARP (0x0806)
  Address Resolution Protocol (reply)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (0x0002)
    Sender MAC address: 00:c0:4f:0c:71:8c (rgmi208tj.plg.univ-nancy2.fr)
    Sender IP address: rgmi208tj.plg.univ-nancy2.fr (193.54.41.42)
    Target MAC address: 00:b0:d0:65:dd:19 (rgmi205dp.plg.univ-nancy2.fr)
    Target IP address: rgmi205dp.plg.univ-nancy2.fr (193.54.41.38)
```

Trame ARP Reply - L'adresse MAC demandée se trouve dans le champ Sender Mac adress.

4. A reçoit donc la réponse à sa requête, il place donc le couple adresse IP/adresse MAC de B dans sa table de cache ARP
5. A connaissant l'adresse MAC de B peut lui envoyer le message.

La **table de cache ARP** est destinée à conserver un certain temps en mémoire (généralement 15 minutes) les adresses IP résolues en adresse MAC afin d'économiser du temps et du trafic réseau lors des échanges.

```
G:\>arp -a

Interface : 193.54.41.42 on Interface 2
Adresse Internet      Adresse physique      Type
193.54.41.33          00-a0-c9-4c-f6-e8     dynamique
193.54.41.43          00-c0-4f-7d-38-9e     dynamique
193.54.41.53          00-c0-4f-a4-f1-0a     dynamique
193.54.41.252         00-60-97-91-68-8f     dynamique
193.54.41.254         00-d0-d3-33-8f-8c     dynamique
```

Table cache ARP

## Le protocole RARP

Généralement une machine peut connaître facilement sa propre adresse IP car elle est stockée dans un endroit de son disque local, mais certains ordinateurs ne disposent pas de disque local (diskless station), ou tout du moins n'utilisent pas ce disque local pour stocker des paramètres de configuration comme l'adresse IP.

Pour obtenir son adresse IP, l'ordinateur en question devra utiliser le protocole RARP (**R**everse **A**dress **R**esolution **P**rotocol), le principe de ce protocole est le suivant :

1. La machine A (diskless station) envoie en diffusion une requête RARP dans laquelle figure son adresse MAC
2. Tous les noeuds du réseau reçoivent cette requête mais seuls le ou les serveurs RARP vont la traiter
3. Le serveur RARP tient à jour une liste des adresses IP des noeuds du réseau avec la correspondance avec l'adresse physique MAC
4. Si l'adresse MAC figurant dans la trame RARP existe dans la table de correspondance du serveur RARP, ce dernier renvoie une réponse à la machine A avec son adresse IP

## IPv6

A l'heure actuelle le "vieux" protocole IPv4 comporte de nombreux inconvénients :

- Le **potentiel d'adressage** est insuffisant : 4 milliards d'adresses alors que le nombre d'ordinateurs mais également de matériels en tout genre intégrés à des réseaux ne cesse d'augmenter de façon significative.
- La **qualité de service** est très mal assuré
- Des **problèmes de sécurité** demeurent
- Les réseaux et les routeurs sont **encombrés** par les nombreux contrôles effectués à différents niveaux

Les principaux objectifs du nouveau protocole IPv6 dont les premiers travaux remontent à 1993 (le protocole IPv5 existe déjà mais est destiné au streaming) sont les suivants :

- **Augmenter le potentiel d'adressage**
- Réduire la taille des tables de routage afin d'**alléger le travail des routeurs**
- **Simplifier** le protocole, pour permettre aux routeurs de router les datagrammes plus rapidement,
- Fournir une **sécurité** (authentification et confidentialité) satisfaisante
- Améliorer la **qualité de service** pour prendre en compte les applications "temps réel" comme la vidéo par exemple
- Donner la possibilité à un ordinateur de se **déplacer** sans changer son adresse

Il n'est pas envisageable de basculer du jour au lendemain de IPv4 à IPv6, donc le changement se fait petit à petit, pour l'instant il existe des "niches" IPv6 qui dialoguent entre elles ou avec le reste d'internet en encapsulant les trames IPv6 dans des trames IPv4, progressivement le nombre de ces niches va augmenter et seul resteront des niches IPv4 au milieu d'un monde IPv6.

La trame IPv6 est considérablement simplifiée puisqu'on n'y trouve plus que les éléments suivants :

- Le champ **Version** (4 bits) :  
Version d'IP utilisée.
- Le champ **Traffic Class** (8 bits) :  
Contrôle de flux . Des priorités de 0 à 7 sont affectées aux sources capables de ralentir leur débit en cas de congestion, les valeurs 8 à 15 sont assignées au trafic temps réel (audio, vidéo, etc...).
- Le champ **Flow Label** (20 bits) :  
contient un numéro unique choisi par la source qui a pour but de faciliter le travail des routeurs et de permettre la mise en oeuvre les fonctions de qualité de services. Cet indicateur peut être considéré comme une marque pour un contexte dans le routeur. Le routeur peut alors faire un traitement particulier : choix d'une route, traitement en "temps-réel" de l'information, ...  
Le champ identificateur de flux peut être rempli avec une valeur aléatoire qui servira à référencer le contexte. La source gardera cette valeur pour tous les paquets qu'elle émettra pour cette application et cette destination. Le traitement est optimisé puisque le routeur n'a plus à consulter que cinq champs pour déterminer l'appartenance d'un paquet. De plus, si une extension de confidentialité est utilisée, les informations concernant les numéros de port sont masquées aux routeurs intermédiaires.
- Le champ **Payload Length** (2 octets) :  
Taille des données utiles.
- Le champ **Next Header** (8 bits) :  
Protocole de niveau supérieur ICMP, UDP, TCP.
- Le champ **Hop Limit** (8 bits) :  
Remplace le champ "TTL" (Time-to-Live) de IPv4. Sa valeur est décrémentée à chaque noeud traversé.  
Si cette valeur atteint 0 alors que le paquet IPv6 traverse un routeur, il sera rejeté avec l'émission d'un message ICMPv6 d'erreur. Le but est d'empêcher les datagrammes de circuler indéfiniment.
- Champs **Source Address** et **Destination Address** sur 128 bits (16 octets chacun) :  
Adresse de l'émetteur et du destinataire. 128 bits au lieu de 32 bits pour IPv4, ce qui offre un potentiel d'adressage de  $3,4 \times 10^{38}$  ce qui devrait suffire pendant quelques temps...

<b>Version</b> == 6	<b>8 bits</b> <b>Traffic Class</b>	<b>20 bits</b> <b>Flow Label</b>	
<b>16 bits</b> <b>Payload Length</b>		<b>8 bits</b> <b>Next Header</b>	<b>8 bits</b> <b>Hop Limit</b>
<b>128 bits</b> <b>Source Address</b>			
<b>128 bits</b> <b>Destination Length</b>			

Datagramme IPv6



# Les protocoles de transport

Sommaire :

[Le protocole TCP](#)

[Le protocole UDP](#)

## Le protocole TCP

[Généralités](#) - [Segmentation et séquençement des données](#) - [Mécanisme d'acquittement](#) - [Etablissement d'une connexion](#) - [Contrôle du flux de données](#) - [Structure des segments TCP](#) - [Numéros de ports usuels](#)

### Généralités

TCP est un protocole de la couche Transport au sens du modèle OSI. Il s'exécute au dessus du [protocole IP](#) qui lui fournit un service de datagrammes sans connexion entre deux machines. TCP est un protocole **orienté connexion** qui garantit que les données sont remises de façon **fiable**. TCP s'oppose à UDP qui est moins robuste mais plus efficace dans certaines situations.

Le protocole IP est sans connexion et ne garantit absolument pas que le datagramme envoyé a été remis, TCP s'appuie donc sur l'hypothèse que IP n'est pas fiable et qu'il faut au niveau Transport mettre en place un certain nombre de contrôle.

Les fonctionnalités de TCP sont donc principalement :

- **Etablissement d'une connexion**
- Transmission fiable des données en effectuant un **contrôle des données** et en effectuant un **réémission** pour les données qui n'ont pas pu être transférées correctement.
- **Réordonnement** des informations transférées. En effet, les informations seront en fait transmises dans des datagrammes IP qui peuvent éventuellement emprunter des chemins différents donc ne pas arriver dans l'ordre d'émission.
- Gérer le **multiplexage**, c'est à dire que plusieurs applications peuvent utiliser simultanément les services du protocole TCP, exemple : un client courrier qui s'exécute en même temps qu'une navigation sur le web et un téléchargement de fichier.

### Segmentation et séquençement des données

Lorsque de l'information doit être envoyée d'un émetteur vers un récepteur par le protocole TCP, cette information est découpée en **segments** qui peuvent être de **taille variable**. Mais pour des raisons de fiabilité chaque octet d'un segment va être numéroté avec un **numéro de séquence**.

Lors de l'envoi de l'information, dans l'entête de chaque segment, seul le n° de séquence du premier octet sera mentionné. Les autres numéros (en fait c'est le numéro du dernier qui est intéressant) seront calculés en ajoutant ce numéro au nombre d'octets présents dans la partie "données" du segment.

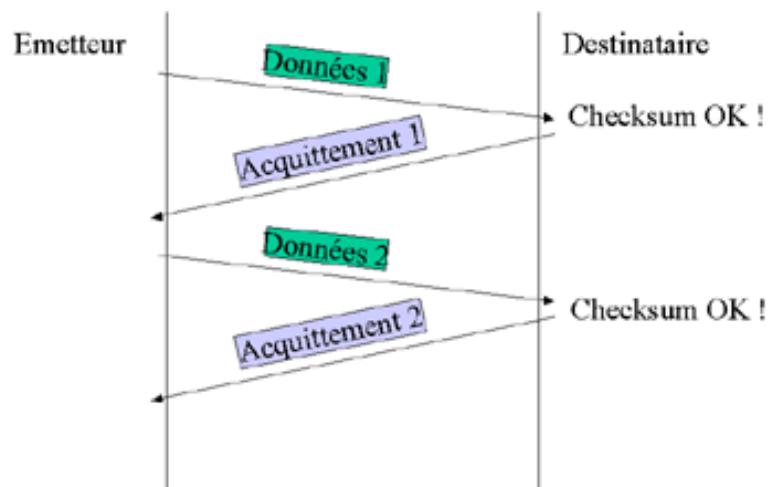
Ce numéro de segment est codé sur 32 bits ce qui permet de numéroté les segments jusqu'à la valeur  $2^{32} = 4\,294\,967\,296$ , au delà il faudra repartir à partir de la valeur 0.

## Mécanisme d'acquittement

La transmission doit être fiable, TCP utilise donc le mécanisme classique d'**acquittement** mais dans une version dite **cumulative**.

### Le principe de l'acquittement

Le principe général est assez simple : Lorsqu'un segment est reçu par le destinataire, celui-ci vérifie que les données contenues dans ce segment sont correctes (consultation du checksum) et si c'est le cas envoie un message d'**acquittement positif** (ACK) vers l'expéditeur.



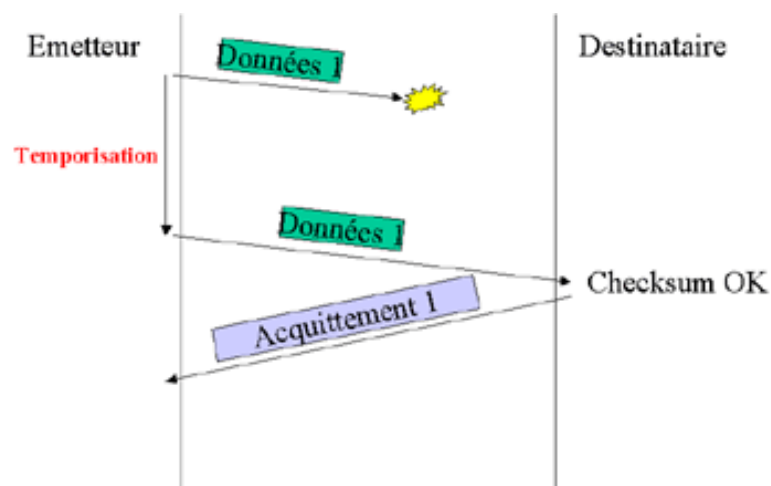
Deux types de problèmes peuvent se produire :

- Les données du segment sont **endommagées**.
- Le segment **n'arrive jamais** à destination.

Pour détecter ce type de problème, chaque fois qu'il envoie un segment l'expéditeur effectue 2 opérations :

- Il stocke dans un buffer une **copie** du segment qu'il vient d'envoyer
- Il arme une **temporisation**

Si au bout d'un certain délai aucun acquittement positif n'a été reçu du destinataire le segment est renvoyé en utilisant la copie présente dans le buffer, si par contre un acquittement est reçu pour ce segment, la copie est supprimée du buffer.

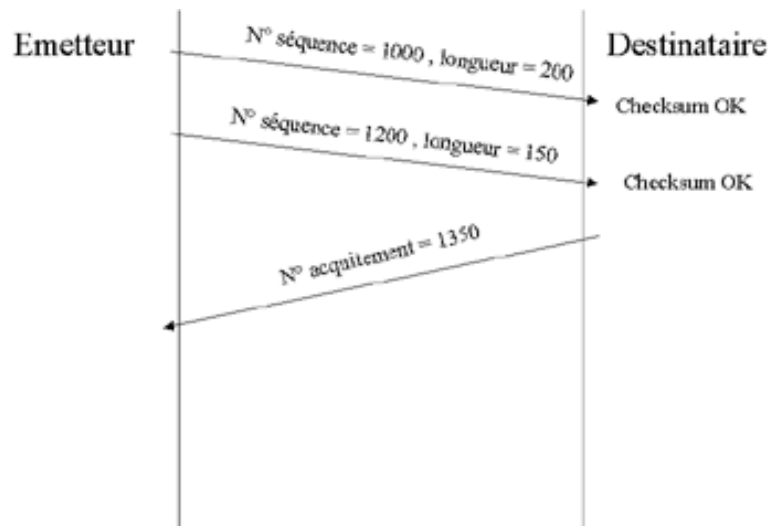


## L'acquittement cumulatif

Le protocole TCP utilise le principe de l'**acquittement cumulatif**, c'est à dire que comme les données sont envoyées par segments de taille variable mais comportant le n° de séquence du premier octet du segment (cf paragraphe sur la segmentation et le séquençement des données), si le contrôle du checksum est satisfaisant, le récepteur déduit à partir du n° de séquence du premier octet et du nombre d'octets reçus le n° de séquence du dernier octet et accuse réception pour cet octet, ce qui implique de façon implicite que tous les octets dont le n° de séquence est inférieur à ce n° de séquence ont été bien reçus.

Il se peut même que pour des raisons d'optimisation, le récepteur attende la réception de plusieurs segments avant d'envoyer un acquittement, ceci a pour but de diminuer le nombre de segments d'acquittements circulant sur le réseau.

Cette technique d'acquittement cumulé a pour principal avantage d'éviter la retransmission de données si un paquet d'acquittement s'est perdu.



## Etablissement d'une connexion

TCP est un **protocole orienté connexion**, cela signifie qu'il va établir et **maintenir** une connexion entre deux machines et **surveiller** l'état de cette connexion pendant toute la durée du transfert. Pour établir une connexion, il faut évidemment identifier les extrémités (end points) de cette connexion, cela se fait avec le couple (n° IP, n° de [port](#)). Le n° IP est celui de l'interface réseau par laquelle les données vont transiter, le n° de port est un numéro associé à l'application.

Exemple de connexion : (192.14.56.17,1400) <-> (192.14.56.26,21)

Cette connexion est établie entre les 2 machines 192.14.56.17 et 192.14.56.26 en utilisant respectivement les ports 1400 et 21.

TCP fonctionne en **full duplex**, c'est à dire que lorsqu'une connexion est établie les données vont pouvoir transiter simultanément dans un sens et dans l'autre.

La demande de connexion peut s'effectuer de 2 manières :

- **passive** (Passive Open), ceci signifie que la machine accepte une connexion entrante. C'est le cas par exemple d'un serveur FTP par exemple qui va se mettre en attente de demande d'établissement de connexion de la part d'un client FTP.
- **active** (Active Open) pour demander l'établissement de la connexion.

L'initialisation d'une connexion se fait toujours par ce qui s'appelle une "**Poignée de main à 3 voies**" qui est la traduction littérale de "**Three Way Handshake**", cette initialisation se déroule donc en 3 étapes. Ces 3 étapes ont pour but essentiel de synchroniser les numéros de séquence des 2 machines :



- La machine A envoie un segment de type "ouverture de connexion" avec le n° de séquence X (dans ce segment ne figure aucune donnée)
- La machine B renvoie un segment de type "ouverture de connexion" avec le n° de séquence Y et en acquittant la séquence X envoyée par A
- La machine A renvoie un acquittement à B du segment n° Y

De cette façon chaque machine connaît le n° de séquence de l'autre et l'échange d'information peut débuter.

Exemple :

Dans l'exemple qui suit la machine A (193.54.41.42) va effectuer une connexion telnet sur le serveur S (193.50.231.10).

**1ère étape :**

```
Frame 43 (58 on wire, 58 captured)
Ethernet II
Internet Protocol, Src Addr: rgmi208tj.plg.univ-nancy2.fr (193.54.41.42), Dst Addr: saphir.plg.univ-nancy2.fr (193.50.231.10)
Transmission Control Protocol, Src Port: 3454 (3454), Dst Port: telnet (23), Seq: 4578763, Ack: 0
  Source port: 3454 (3454)
  Destination port: telnet (23)
  Sequence number: 4578763
  Header length: 24 bytes
  Flags: 0x0002 (SYN)
    0... .... = Congestion window reduced (CWR): Not set
    .0.. .... = ECN-Echo: Not set
    ..0. .... = Urgent: Not set
    ...0 .... = Acknowledgment: Not set
    .... 0... = Push: Not set
    .... .0.. = Reset: Not set
    .... ..1. = Syn: Set
    .... ...0 = Fin: Not set
  Window size: 8192
  Checksum: 0xf9e2 (correct)
  Options: (4 bytes)
```

La machine A envoie un segment TCP vers S pour ouvrir la connexion (SYN). Le n° de séquence est fixé par A à 4578763, le champ ACK est à 0 car il n'y a encore aucun segment en provenance de S à valider.

**2ème étape :**

```
Frame 44 (60 on wire, 60 captured)
Ethernet II
Internet Protocol, Src Addr: saphir.plg.univ-nancy2.fr (193.50.231.10), Dst Addr: rgmi208tj.plg.univ-nancy2.fr (193.54.41.42)
Transmission Control Protocol, Src Port: telnet (23), Dst Port: 3454 (3454), Seq: 1718545887, Ack: 4578764
  Source port: telnet (23)
  Destination port: 3454 (3454)
  Sequence number: 1718545887
  Acknowledgement number: 4578764
  Header length: 24 bytes
  Flags: 0x0012 (SYN, ACK)
    0... .... = Congestion window reduced (CWR): Not set
    .0.. .... = ECN-Echo: Not set
    ..0. .... = Urgent: Not set
    ...1 .... = Acknowledgment: Set
    .... 0... = Push: Not set
    .... .0.. = Reset: Not set
    .... ..1. = Syn: Set
    .... ...0 = Fin: Not set
  Window size: 8760
  Checksum: 0xa34b (correct)
```

La machine S renvoie un segment TCP vers A afin d'acquitter la demande de connexion (ACK=4578764 indique le n° du prochain octet attendu ce qui implicitement acquitte le précédent), le n° de séquence de S est 1718545887.

**3ème étape :**

```

Frame 45 (54 on wire, 54 captured)
Ethernet II
Internet Protocol, Src Addr: rgm1208tj.plg.univ-nancy2.fr (193.54.41.42), Dst Addr: saphir.plg.univ-nancy2.fr (193.50.231.1)
Transmission Control Protocol, Src Port: 3454 (3454), Dst Port: telnet (23), Seq: 4578764, Ack: 1718545888
  Source port: 3454 (3454)
  Destination port: telnet (23)
  Sequence number: 4578764
  Acknowledgement number: 1718545888
  Header length: 20 bytes
  Flags: 0x0010 (ACK)
    0... .. = Congestion Window Reduced (CWR): Not set
    .0.. .. = ECN-Echo: Not set
    ..0. .... = Urgent: Not set
    ...1 .... = Acknowledgment: Set
    .... 0... = Push: Not set
    .... .0.. = Reset: Not set
    .... ..0. = Syn: Not set
    .... ...0 = Fin: Not set
  window size: 8760
  checksum: 0xbb08 (correct)

```

La machine A acquitte le précédent segment en provenance de S (ACK=1718545888) ce qui termine la phase de connexion.

## Contrôle du flux de données

Les machines qui émettent et qui reçoivent des données sont sans doute différentes, elles ne sont sans doute pas en mesure de travailler au même rythme. Il se peut donc que l'émetteur envoie ses données avec un **débit trop important** par rapport au débit de traitement du récepteur. En fait le récepteur stocke les données reçues dans un **buffer** et c'est la place encore libre dans ce buffer qui détermine la quantité de données que cette machine peut encore recevoir.

C'est pour cette raison que le protocole TCP prévoit un mécanisme de **contrôle de flux de données** basé sur la technique dite de la "**fenêtre glissante**". Le principe est le suivant : A chaque acquittement, le récepteur renvoie une valeur (taille de la fenêtre) qui correspond au nombre d'octets que l'émetteur peut envoyer avant le prochain retour d'acquittement. La taille de la fenêtre est fixée par le récepteur à chaque émission d'acquittement.

**Exemple :**



Dans cet exemple la fenêtre est de largeur 8. Les octets 1 à 3 ont été envoyés et acquittés, par contre les octets 4 à 8 non pas encore été acquittés il reste donc une incertitude sur leur bonne réception par le récepteur, comme la taille de la fenêtre est de 8, l'émetteur peut encore émettre sans attendre d'acquittement les octets 9 à 11 mais pas au delà.

Supposons qu'à cet instant l'émetteur reçoit un acquittement pour les octets 4 à 6 et une taille de fenêtre de 10, on obtient alors la situation suivante :



Lorsque le récepteur est trop sollicité et qu'il voit l'espace libre dans son buffer diminuer, il va avoir tendance à réduire la taille de la fenêtre jusqu'à éventuellement atteindre 0 ce qui signifie que le récepteur n'est plus apte pour l'instant à recevoir des données. Il y a un cas très embêtant c'est le cas où le destinataire impose une taille de fenêtre de 1, c'est que

l'on appelle le **syndrome de la fenêtre stupide** (SWS pour Silly Window Syndrom).

```

Frame 45 (54 on wire, 54 captured)
Ethernet II
Internet Protocol, Src Addr: rgm1208tj.plg.univ-nancy2.fr (193.54.41.42), Dst Addr: saphir.plg.univ-nancy2.fr (193.50.231.1)
Transmission Control Protocol, Src Port: 3454 (3454), Dst Port: telnet (23), Seq: 4578764, Ack: 1718545888
  Source port: 3454 (3454)
  Destination port: telnet (23)
  Sequence number: 4578764
  Acknowledgement number: 1718545888
  Header length: 20 bytes
  Flags: 0x0010 (ACK)
    0... .. = Congestion Window Reduced (CWR): Not set
    .0.. .. = ECN-Echo: Not set
    ..0. .. = Urgent: Not set
    ...1 ... = Acknowledgment: Set
    ....0... = Push: Not set
    ....0.. = Reset: Not set
    ......0. = Syn: Not set
    .......0 = Fin: Not set
  Window size: 8760
  Checksum: 0xbb08 (correct)

```

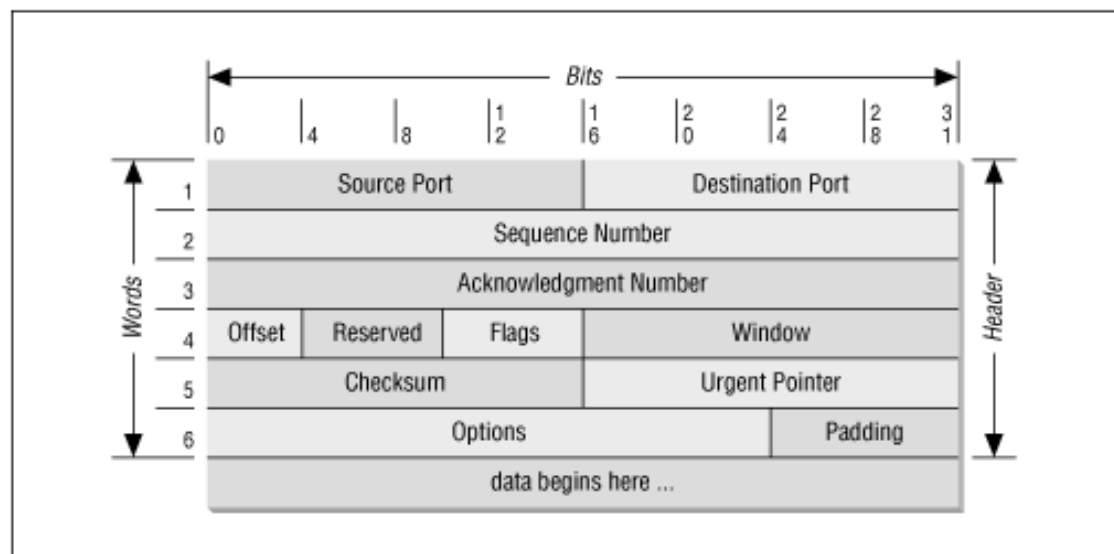
Dans ce segment TCP, la largeur de fenêtre est de 8760

## Structure des segments TCP

Le segment TCP c'est l'unité de transfert du protocole TCP, il est utilisé indifféremment pour établir les connexions, transférer les données, émettre des acquittements, fermer les connexions.

De façon classique, la structure d'un segment TCP comprend un **entête** de taille variable qui utilise un format en mot de 32 bits suivi d'une zone de données.

- [Source Port et Destination Port](#) : N° de port source et destination
- [Sequence Number](#) : N° séquence du premier octet de données
- [Acknowledgment Number](#) : N° d'acquiescement
- [Offset](#) : Nb de mots de l'entête
- [Flags](#) : Drapeaux
- [Window](#) : Taille de la fenêtre
- [Checksum](#) : Contrôle d'erreurs
- [Urgent Pointer](#) : Données urgentes
- [Options](#) : Options du protocole
- [Padding](#) : Remplissage



Cliquer sur une zone pour obtenir le descriptif du champ.

### Source Port et Destination Port (2 x 16 bits)

Ces deux champs de 16 bits chacun contiennent les **numéros de port** de la source et de la destination. Certains [numéros de ports](#) sont dédiés à un protocole particulier (par exemple le port 80 est dédié à http).

[Retour au descriptif du segment TCP](#)

### Sequence Number (32 bits)

Ce n° sur 32 bits correspond au **numéro de séquence du premier octet** de données de ce segment de données, en effet le protocole TCP numérote chaque octet envoyé. Si le [drapeau SYN](#) vaut 1, ce champ définit le numéro de séquence initial (ISN).

[Retour au descriptif du segment TCP](#)

### Acknowledgment Number (32 bits)

Ce champ sert lorsque le segment est un **segment d'acquittement** (le drapeau ACK du champ [Flags](#) est à 1), il indique le numéro de séquence du prochain octet attendu (c'est à dire le n° de séquence du dernier octet reçu + 1), tous les octets précédents cumulés sont implicitement acquittés.

[Retour au descriptif du segment TCP](#)

### Offset (4 bits)

Le champ [Options](#) ayant une largeur variable, ce champ donne la **taille en mots de 32 bits de l'entête du segment**. Si le champ Options est vide, cette taille est égale à 5 (entête de 20 octets).

[Retour au descriptif du segment TCP](#)

### Flags (6 bits)

Ce champ comprend 6 drapeaux qui indique le rôle du segment TCP :

- ACK : Indique un segment d'acquittement
- SYN : Ouverture de la connexion
- FIN : Fermeture de la connexion
- RST : Réinitialisation de la connexion pour cause d'erreurs non récupérables
- PSH : Demande de remise immédiate des données au processus de la couche supérieure
- URG : Données urgentes

[Retour au descriptif du segment TCP](#)

### Window (16 bits)

**Taille de la fenêtre**, c'est à dire le nombre d'octets que le récepteur est en mesure d'accepter à partir du numéro d'acquittement.. Voir le paragraphe sur le [contrôle de flux](#).

[Retour au descriptif du segment TCP](#)

### Checksum (16 bits)

Le Checksum permet de contrôler si le paquet TCP n'a pas été modifié lors de son transport.

[Retour au descriptif du segment TCP](#)

### Urgent Pointer (16 bits)

Donne la position d'une **donnée urgente** en donnant son décalage par rapport au numéro de séquence. Ce champ n'est utilisé que si le [drapeau URG](#) est positionné. Les données urgentes devront passer devant la file d'attente du récepteur, c'est par exemple avec ce mécanisme qu'il est possible d'envoyer des commandes d'interruption au programme Telnet.

[Retour au descriptif du segment TCP](#)

### Options (variable)

Utilisé à des fonctions de test.

[Retour au descriptif du segment TCP](#)

### Padding (variable)

Octets de bourrage qui permettent de terminer l'en-tête TCP.

[Retour au descriptif du segment TCP](#)

## Numéros de port usuels

No port	Mot- clé	Description
20	FTP-DATA	File Transfer [Default Data]
21	FTP	File Transfer [Control]
23	TELNET	Telnet
25	SMTP	Simple Mail Transfer
37	TIME	Time
42	NAMESERVER	Host Name Server
43	NICNAME	Who Is
53	DOMAIN	Domain Name Server
79	FINGER	Finger
80	HTTP	WWW
110	POP3	Post Office Protocol - Version 3
111	SUNRPC	SUN Remote Procedure Call

## Le protocole UDP

[Généralités](#) - [Structure du paquet UDP](#)

### Généralités

Le protocole UDP est une alternative au protocole [TCP](#). Comme TCP, il intervient au dessus de la couche IP, au niveau **Transport** au sens des couches ISO.

Les caractéristiques du protocole UDP sont les suivantes :

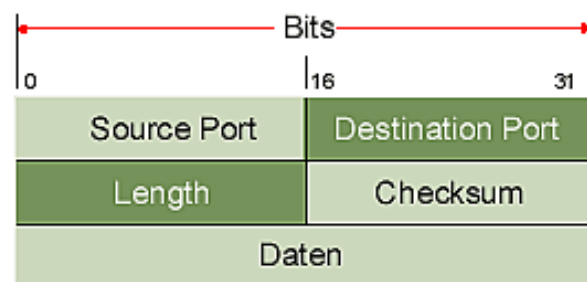
- identifie les processus d'application à l'aide de **numéros de ports UDP** (distincts des numéros de port TCP)
- possède un **contrôle d'erreurs assez rudimentaire**, il est donc destiné aux réseaux fiables
- UDP est un protocole qui n'est **pas orienté connexion**
- peut éventuellement vérifier l'**intégrité** des données transportées
- Les données ne sont **pas séquencées** donc rien ne permet de vérifier que l'ordre d'arrivée des données et le même que celui d'émission. Ceci le destine plutôt aux réseaux locaux où le mode d'acheminement des informations ne risque pas d'inverser l'ordre des données mais également aux applications qui véhiculent des informations de petites tailles qui peuvent tenir en un seul datagramme.
- De par sa structure UDP est **plus rapide** que TCP, mais **moins robuste**

UDP est donc un **protocole orienté commande/réponse**. UDP peut être utile pour les applications qui nécessitent une **diffusion** d'informations car dans ce cas il serait pénalisant d'utiliser un protocole comme TCP orienté connexion qui devrait gérer (ouvrir et fermer) autant de connexion que de noeuds auxquels l'information est destinée.

En conclusion, UDP est utilisé par les applications **TFTP** (Trivial File Transfer Protocol), **DNS** (Domain Name System), **NFS** (Network File System), **SNMP** (Simple Network Management Protocol), **RIP** (Routing Information Protocol) ainsi que de nombreux services qui envoient des données en diffusion comme **WHOD** (Who Daemon pour serveurs Unix) par exemple.

## Structure du paquet UDP

- **Source Port et Destination Port** :  
port source et destination
- **Length** :  
Longueur du paquet UDP
- **Checksum** :  
champ de contrôle des données.



© bacChannel

# Le routage sous IP

**Voir chapitre 6 : Interconnexion de réseaux**

# La couche Application

Sommaire :

[La résolution de noms](#)

[Allocation dynamique d'adresses](#)

[HTTP](#)

[Transfert de fichiers](#)

[Autres applications utilisant TCP/IP](#)

## La résolution de noms

[Généralités](#) - [Fichiers HOSTS](#) - [DNS](#) - [NSLookup](#)

### Généralités

Sur un réseau TCP/IP les différents noeuds du réseaux (ordinateurs, routeurs, etc..) sont identifiés par une [adresse Ip](#) dont le format dans la version IPv4 est une suite binaire de 32 bits que l'on note généralement sous la forme de 4 entiers inférieurs à 256 séparés par des points, exemple : 192.14.25.251 (cf chapitre sur l'adressage).

Dès les débuts des réseaux TCP/IP, les utilisateurs ont rapidement pris conscience que ce type d'adressage était pratiquement impossible à mémoriser pour des individus normalement constitués. C'est la raison pour laquelle est apparue le système de noms de machine appelé **FQDN : Full Qualified Domain Name**. Par exemple :  
mamachine.service.masociete.fr

Remarque : Il ne faut pas confondre FQDN et **URL (Uniform Ressource Locator)**. L'URL est la méthode d'accès à un document distant, un lien hypertexte par exemple, avec une syntaxe de la forme: <Type de connexion>://<FQDN>/[<sous-répertoire>]/.../<nom du document>. Exemple :  
<http://mamachine.service.masociete.fr/cours/chapitre1.htm>

Puisque le protocole TCP/IP ne connaît que des adresses IP, il faut donc disposer d'un mécanisme qui permette de traduire une adresse de type FQDN en adresse IP. Cette traduction est appelée **résolution de noms**. Elle peut se faire de 2 façons :

- Manuellement, par le biais des [fichiers hosts](#)
- En utilisant les services du [DNS](#) : **Domain Name System**

### Les Fichier HOSTS



A l'origine, les réseaux IP étaient peu étendus et le nombre de machines relativement faible, c'est pourquoi la première solution pour faire correspondre un FQDN avec une adresse IP fut d'utiliser sur chacune des machines du réseau, un fichier qui porte généralement le nom `HOSTS` et qui répertorie les paires (FQDN, adresse IP) connues et utiles.

```
102.54.94.97 rhino.acme.com # source server
38.25.63.10 x.acme.com # x client host
127.0.0.1 localhost
```

### Extrait d'un fichier hosts

Il va sans dire que ce système trouve tout de suite ses limites :

- Nécessité d'une mise à jour permanente impossible à réaliser dans le cadre d'un réseau comme internet
- Non prise en compte de la structure hiérarchique de l'adressage IP

Cette solution n'est donc plus utilisée que pour des cas très particuliers de résolution de noms mais la méthode universellement utilisée est celle du serveur de noms ou DNS.

Il faut tout de même remarquer que lorsqu'un nom doit être traduit en adresse FQDN, c'est tout d'abord le fichier `hosts` qui est consulté, si le nom n'est pas trouvé il est fait appel au serveur de noms, ce qui signifie que la méthode du fichier `hosts` peut permettre :

- D'accélérer la résolution de noms pour des noms utilisés fréquemment
- Bloquer l'accès à certains sites internet indésirables (ceux qui diffusent les bandeaux publicitaires par exemple) en redirigeant le nom sur l'adresse de loopback 127.0.0.1

## Le DNS

DNS signifie **Domain Name System**, autrement dit **Système de Noms de Domaine**. Ce système repose sur une base de données **hiérarchique**, **distribuée** sur un grand nombre de serveurs appelés **serveurs de noms**.

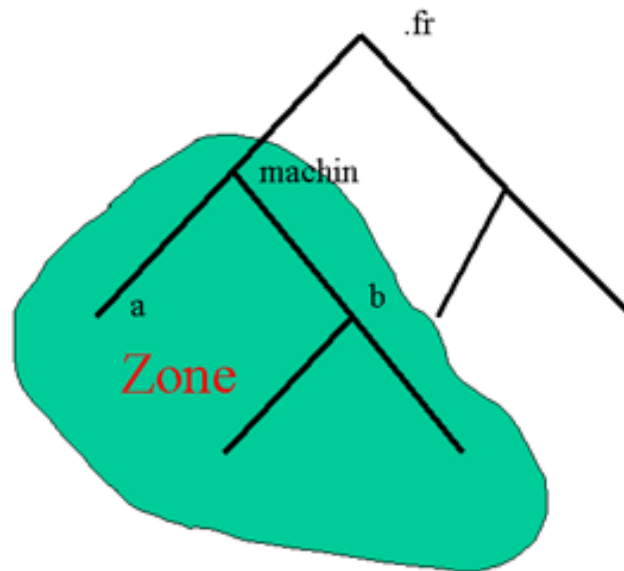
En haut de la hiérarchie on trouve les "**serveurs de racine**" (**DNS Root Servers**). Les serveurs racine connaissent les serveurs de nom ayant **autorité** sur tous les **domaines racine** (c.a.d .com, .edu, .fr, etc...). Ces serveurs racine représentent la pierre angulaire du système DNS : si les serveurs racine sont inopérants, il n'y a plus de communication sur l'Internet, d'où multiplicité des serveurs racines (actuellement il y en a 13). Chaque serveur racine reçoit environ 100000 requêtes /heure. Ces serveurs racine ont d'ailleurs fait l'objet d'une attaque de pirates informatiques en octobre 2002.



DNS Root Servers (source ICANN)

En dessous de ces "serveurs racine", on trouve des "**zones**" qui correspondent à des sous-ensembles de

l'arborescence, ces zones sont gérées par une même **autorité**.

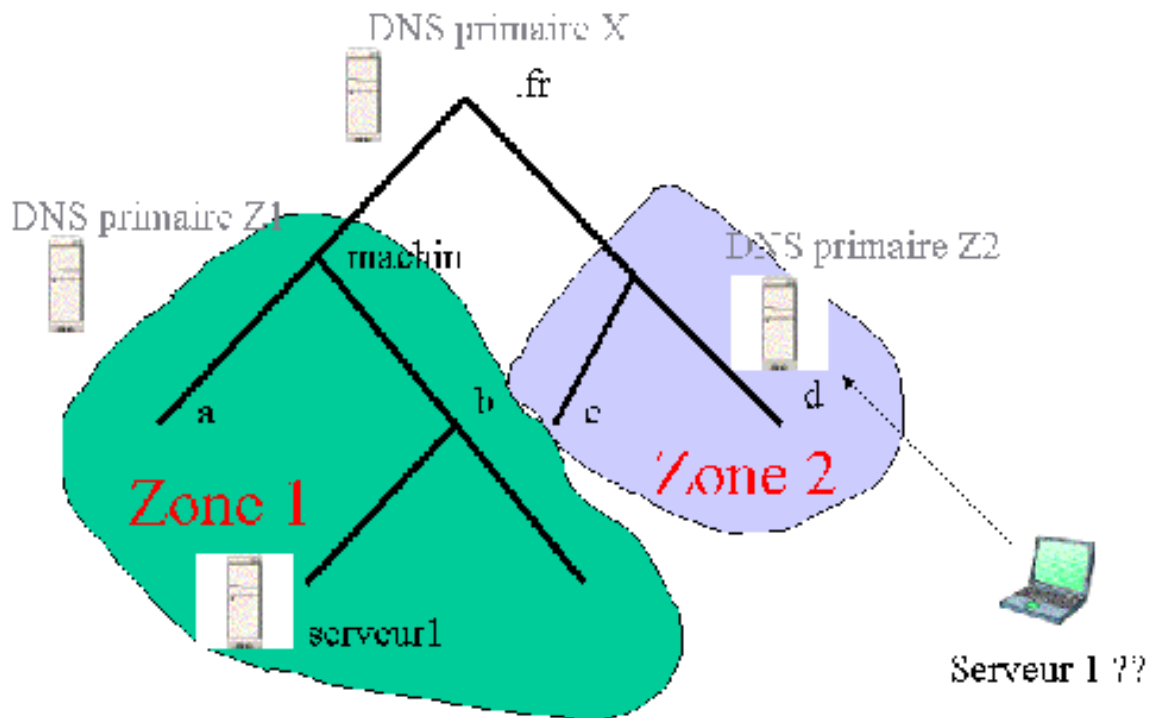


Dans ces zones on trouve 3 types de serveurs de noms :

- **Le serveur primaire**  
C'est le serveur qui a autorité sur sa zone, il tient à jour **toutes** les correspondances entre noms et adresse Ip de sa zone. Il n'y a qu'un seul serveur primaire par zone.
- **Les serveurs secondaires**  
Ce sont des serveurs qui reçoivent leurs informations directement du serveur primaire par une opération qu'on appelle "transfert de zone". Ils sont capables de remplacer ce dernier en cas de panne.
- **Les serveurs caches**  
Ce type de serveur ne constitue sa base d'information qu'à partir des réponses d'autres serveurs de noms. Ces informations sont stockées avec une durée de vie limitée (TTL). Il est capable de répondre aux requêtes des clients DNS.

Pour que le système fonctionne, il faut bien sûr qu'un serveur de noms qui ne peut pas répondre à une demande puisse interroger un autre serveur de noms, c'est pour cela que chaque serveur de noms dispose d'un certain nombre de **pointeurs** vers des serveurs de noms capables de résoudre le problème.

Algorithme de résolution de noms (exemple) :



1. L'ordinateur portable du schéma est configuré pour utiliser le serveur DNS Z2. Il émet une requête pour connaître l'adresse du serveur1.
2. Si le DNS Z2 ne possède pas cette information en cache, il va se référer au serveur de domaine X
3. X n'a pas l'information mais il connaît l'adresse du serveur DNS Z1 qui gère la Zone 1, il renvoie donc cette adresse à Z2
4. Z2 fait donc une requête vers Z1
5. Z1 connaît forcément l'adresse de serveur1, il la communique donc à Z2 qui la recopie dans son cache et la transmet au portable demandeur

## NSLookup

La commande NSLookup présente sur la plupart des systèmes d'exploitation permet d'interroger les DNS.

Par défaut, cette commande interroge le DNS pour lequel la machine a été configurée.

```
C:\>nslookup
Serveur par défaut : isis.univ-nancy2.fr
Address: 194.214.218.110
```

Il est possible ensuite de rechercher un nom de machine (ici la machine rmi208tj) du domaine, on obtiendra alors son nom complet (FQDN) et son adresse IP :

```
> rgmi208tj
Serveur: isis.univ-nancy2.fr
Address: 194.214.218.110

Nom : rgmi208tj.plg.univ-nancy2.fr
Address: 193.54.41.42
```

On peut également interroger ce DNS pour connaître l'adresse IP d'un serveur web quelconque situé n'importe où sur la toile. On obtient le vrai nom de la ressource (ici skirandopc1.skirando.ch) et on apprend que notre DNS a obtenu cette information en contactant d'autre(s) DNS (réponse de source secondaire..) :

```
> www.skirando.ch
Serveur: isis.univ-nancy2.fr
Address: 194.214.218.110
```

```
Réponse de source secondaire :
Nom : skirandopc1.skirando.ch
Address: 128.179.66.2
Aliases: www.skirando.ch
```

On peut également interroger un autre DNS, un serveur "racine" par exemple (la liste des serveurs racines peut être consultée à l'adresse : <ftp://ftp.rs.internic.net/domain/named.root>) . :

```
C:\>nslookup
Serveur par défaut : isis.univ-nancy2.fr
Address: 194.214.218.110
```

```
> server b.root-servers.net
Serveur par défaut : b.root-servers.net
Address: 128.9.0.107
```

On se trouve donc à la racine de l'arbre, on peut donc "remonter" la branche qui nous intéresse par exemple .fr  
On obtient ainsi la liste des serveurs de noms que l'on peut interroger sur ce domaine :

```
> fr
Serveur: b.root-servers.net
Address: 128.9.0.107
```

```
Nom : fr
Served by:
- DNS.CS.WISC.EDU
128.105.2.10
fr
- NS1.NIC.fr
192.93.0.1
fr
- NS3.NIC.fr
192.134.0.49
fr
- DNS.INRIA.fr
193.51.208.13
fr
- NS2.NIC.fr
192.93.0.4
fr
- DNS.PRINCETON.EDU
128.112.129.15
fr
- NS-EXT.VIX.COM
```

```
204.152.184.64
fr
- NS3.DOMAIN-REGISTRY.NL
193.176.144.6
fr
```

On peut choisir l'un d'eux et rechercher un nom de domaine (univ-nancy2.fr par exemple). On obtient ainsi l'adresse des serveurs qui peuvent nous renseigner sur des machines de ce domaine (ici il y en a 2).

```
> univ-nancy2.fr
Serveur: dns.inria.fr
Address: 193.51.208.13
```

```
Nom : univ-nancy2.fr.plg.univ-nancy2.fr
Served by:
- isis.univ-nancy2.fr
194.214.218.110
univ-nancy2.fr
- arcturus.ciril.fr
193.50.27.66
univ-nancy2.fr
```

## L'allocation dynamique d'adresses

### Généralités - DHCP

#### Généralités

Dans un réseau utilisant le protocole TCP/IP, **chaque machine doit disposer d'une adresse IP**. Cette adresse IP est généralement stockée (avec d'autres paramètres relatifs au protocole comme le masque, l'adresse de la passerelle par défaut) sur le disque dur de la machine. On parle dans ce cas d'**adressage statique**.

L'adressage IP statique présente un certain nombre d'inconvénients :

- La configuration doit se faire **manuellement**. Cette opération, même si elle est simple à réaliser, a non seulement un coût en terme de temps passé à réaliser cette opération mais comporte également un certain risque d'erreurs : mauvaise adresse, adresse dupliquée, erreur dans le masque, etc...
- Le **gaspillage** d'adresses. Sur un parc de  $n$  machines, il n'est pas toujours certain que ces  $n$  machines vont avoir besoin simultanément d'utiliser le protocole IP. C'est typiquement le problème des fournisseurs d'accès à internet qui possèdent un grand nombre de clients mais, à un instant donné, seule une proportion assez faible utilise internet.

- Une **configuration rigide**. La structure d'un réseau peut évoluer : ajout de nouvelles machines, interconnexion de réseaux, etc... Ceci est d'autant plus vrai si on utilise du matériel très mobile comme les ordinateurs portables. La configuration effectuée manuellement va dans ce cas pénaliser fortement la flexibilité du réseau et son adaptation rapide aux nouvelles contraintes.

L'adressage statique sera donc bien adapté à des réseaux de petite taille et qui évoluent peu, par contre pour les autres il sera plus intéressant de se tourner vers un mode d'**allocation d'adresse dynamique**.

Il existe 2 protocoles qui gèrent l'allocation dynamique d'adresses :

- **BOOTP** : C'est le plus ancien, il a été développé à l'origine pour le démarrage des stations sans disque.
- **DHCP** : C'est une extension du protocole BOOTP, plus souple dans sa gestion des adresses IP, c'est lui qui est utilisé à l'heure actuelle sur la plupart des configurations où une allocation dynamique d'adresses est mise en oeuvre.

## DHCP

### Le principe

DHCP signifie **Dynamic Host Configuration Protocol**. C'est un protocole de la couche Application qui utilise [UDP](#) et [IP](#). Le principe de configuration dynamique repose sur un principe de **Client / Serveur**.

Le **client DHCP**, pour obtenir une adresse IP va effectuer auprès d'un **serveur DHCP** un certain nombre de requêtes. Les adresses obtenues ne le sont généralement pas de façon définitive mais le sont pour une durée (**durée de bail**) prévues au niveau du serveur. Cette procédure s'effectue de la façon suivante :

- Le client en quête de configuration émet un datagramme spécifique (destiné au port 67 du serveur) via le protocole [UDP](#) destiné à **tout serveur DHCP présent** sur le réseau. Ce datagramme (DHCP DISCOVER) est de type diffusion c'est à dire qu'il est envoyé à l'adresse 255.255.255.255 (voir chapitre relatif à [l'adressage IP](#)). Dans ce datagramme figure l'adresse physique (adresse MAC) du client.
- Le client passe dans un **état d'attente** pendant lequel il attend qu'un serveur DHCP et lui propose une adresse IP sur son port 68. Si un serveur DHCP est actif est dispose d'une adresse IP libre il va la proposer au client qui a fait la demande et lui faire une proposition sous la forme d'une trame DHCP OFFER. Cette trame peut être envoyée soit directement à l'adresse physique du client soit en mode broadcast.
- Lorsqu'il reçoit une adresse IP, le client peut éventuellement vérifier que cette adresse IP n'est pas déjà attribuée sur le réseau. Cette vérification peut se faire en envoyant une requête ARP à l'adresse indiquée (cf chapitre [Résolution d'adresse](#)). Ensuite le client peut accepter cette adresse (c'est généralement ce qu'il fait...) ou la refuser. Si le client accepte cette adresse, il va diffuser un datagramme d'acceptation de cette adresse (DHCP REQUEST), ce qui a pour effet de réserver cette adresse et de prévenir les éventuels autres serveurs DHCP qui auraient répondu à la demande de ne pas donner suite. S'il existe plusieurs serveurs DHCP sur le réseau, c'est la première proposition valide d'adresse qui sera retenue.
- Le serveur DHCP constate que son offre a été retenue, il envoie donc un datagramme de confirmation (DHCP ACK) au client. Dans ce datagramme on trouve l'adresse IP bien sûr mais également le masque indispensable à la configuration, de manière optionnelle on peut également trouver l'adresse de la **passerelle par défaut** et la ou les adresses des [DNS](#). A la réception de ces paramètres, le client va configurer sa pile IP avec les paramètres reçus et dans le même temps va armer 3 temporisateurs qui vont servir à la [gestion du bail](#).

### La gestion du bail

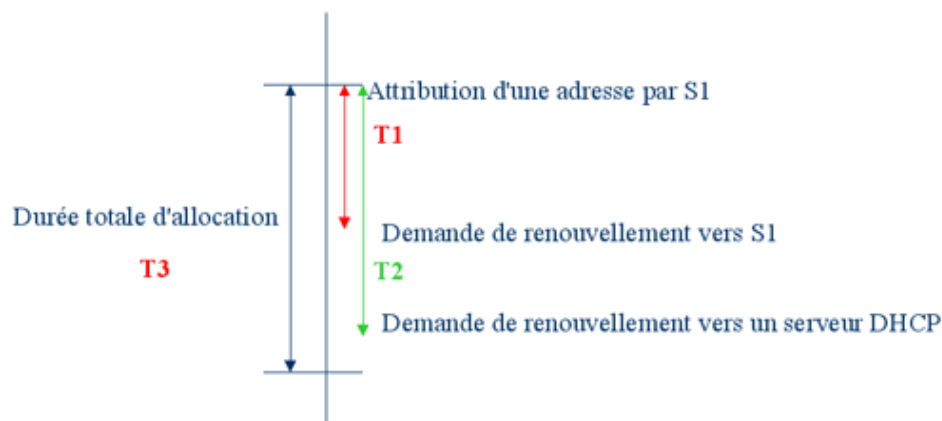
La plupart du temps, la gestion des adresses IP par un serveur DHCP est faite en incluant la notion de **bail**, c'est à dire qu'une adresse IP sera allouée à un client pour une **durée finie**.

Un client qui voit son bail arriver à terme peut demander au serveur un **renouvellement du bail**. De même, lorsque le serveur verra un bail arrivé à terme, il émettra un paquet pour demander au client s'il veut prolonger son bail. Si le

serveur ne reçoit pas de réponse valide, il rend disponible l'adresse IP précédemment allouée.

Le bail est caractérisé par 3 temporisateurs :

- **T3 : Durée totale du bail.** Cette durée est fixée par le serveur et est transmise au client par l'intermédiaire du datagramme DHCP ACK.
- **T1 : Temporisateur de renouvellement du bail.** Cette durée est généralement égale à la moitié de T3, c'est la durée limite au terme de laquelle il faudra que le client ait effectué une demande de renouvellement de bail auprès du serveur.
- **T2 : Temporisateur de rebinding.** Il se peut que le client ait procédé à une demande de renouvellement de bail mais que le serveur n'ait pas répondu à cette demande de renouvellement pour des problèmes de dysfonctionnement de ce serveur ou des connexions avec celui-ci. A l'expiration de T2, le client va donc diffuser sur le réseau une demande de renouvellement de son adresse à l'intention de tous les serveurs DHCP éventuellement présent sur le réseau. Généralement T2 équivaut à peu près à 80 % de T3.



Toute la problématique dans la gestion d'un serveur DHCP vient justement du **choix de la durée du bail**. Un bail trop court va imposer un renouvellement fréquent avec toutes les mobilisations de ressources côté serveur et client que cela engendre ainsi que les messages véhiculés sur le réseau, d'autant plus que ces messages sont de type diffusion. A l'opposé, un bail trop long ne va pas permettre une gestion économe des adresses IP surtout dans un contexte où le réseau évolue beaucoup (ordinateurs portables par exemple).

## Le protocole HTTP

[Généralités](#) - [Les URL](#) - [Les requêtes-réponses](#) - [HTTPS](#)

### Généralités

**HTTP** signifie **HyperText Transport Protocol**, c'est un protocole **léger** et **rapide** utilisé pour délivrer des fichiers **multimédia** et **hypertextes**, appelés plus généralement "ressources", en utilisant **internet**. Ces ressources sont identifiées par un **URL** : **Uniform Ressource Locator**. Une ressource peut être un fichier (fichier texte codé en HTML par exemple ou image de type jpeg, gif, etc...) ou du texte **HTML** (**Hypertext Transfer Protocol**) généré dynamiquement par un script **CGI** (**Common Gateway Interchange**).



HTTP se base sur un **système requête/réponse** entre un **client HTTP** (un **navigateur** en fait) et un **serveur HTTP**. Par défaut, le n° de port utilisé par le serveur HTTP est le **port 80** mais dans la pratique n'importe quel port peut être utilisé.

## Les URL

Un URL permet de localiser des ressources sur le Web.

Un URL se compose de plusieurs éléments :

Exemple (en cliquant sur un élément, on peut se déplacer vers la signification de celui-ci) :

**<http://www.monsite.com:8000/presentation/index.htm?user=prof&type=1>**

- Le **protocole** (ou **classification**): **http** ou **https** (http sécurisé)
- Le **nom d'hôte** : Le nom du serveur ou éventuellement son adresse IP
- Le **numéro de port** (facultatif) : Par défaut ce n° de port est 80 pour HTTP et 443 pour HTTPS, mais il est possible d'utiliser un autre n° de port (précédé par :)
- Le **chemin** : C'est l'emplacement de la ressource demandée, cela n'a pas forcément de lien avec un chemin disque dur car les serveurs gèrent des **alias** qui permettent de remplacer une portion de chemin par un littéral.
- La **chaîne de requête** (facultatif) : Permet de passer des paramètres supplémentaires aux scripts. Cette chaîne lorsqu'elle existe est constituée d'un ensemble de paires **nom=valeur** séparées par le symbole &, la chaîne elle même débute par le caractère ?
- L'**identificateur de fragment** (facultatif) : Désigne une section spécifique d'une ressource, un début de paragraphe dans une page web par exemple. Cet identificateur n'est utilisé que par les navigateurs afin de décaler sa fenêtre de visualisation en conséquence.

Dans la pratique, il existe en fait plusieurs classes d'URL :

- Les URL **absolues** : Elles contiennent le nom d'hôte comme spécifié dans la [définition précédente](#).
- Les URL **relatives** : Ces URL ne comportent pas les parties protocoles, nom d'hôte et numéro de port. Le navigateur devra alors supposé que la ressource se trouve sur la même machine. Ces URL relatives peuvent commencer par un / , il s'agira alors d'un **chemin complet**, dans le cas contraire, il s'agira d'un **chemin relatif** décrit par rapport à l'emplacement de la ressource où ce chemin est spécifié.

Exemple : l'URL ../images/schema.gif est un URL relatif avec un chemin relatif, c'est à dire que pour trouver la ressource schema.gif il faut remonter au répertoire (ou dossier) supérieur et ouvrir le répertoire images.

## L'encodage des URL :

De nombreux caractères apparaissant dans les URL posent problème, c'est le cas des caractères espace, #, ?, @, +, etc.. qui soit on une signification dans la structure d'un URL, soit ont une signification pour d'autres protocoles. Ce codage s'effectue en utilisant % suivi de la **valeur hexadécimale** du caractère sur 2 chiffres (exemple : %20 correspond à l'espace).

## Les Requetes- Réponses

Lorsqu'un navigateur demande une page (l'utilisateur a cliqué sur un lien ou a tapé un URL dans la fenêtre de navigation par exemple), celui-ci envoie au serveur une **requête** HTTP. Le serveur lui répondra par une **réponse** HTTP qui généralement (mais pas toujours..) contient la ressource demandée par le serveur. Le protocole HTTP jusqu'à la version 1.0 était un protocole qui fonctionne en mode **non connecté** (stateless) c'est à dire qu'après l'échange requête-réponse la connexion n'est pas maintenue, depuis la version 1.1, la connexion peut désormais être



maintenue.

La structure des requêtes et des réponses est la même, elle est constituée de :

- un entête
- un corps

**Structure de l'entête : une ligne initiale + couples (champ d'entête : valeur)**

La **ligne initiale** : de forme différente si l'on est dans une requête ou dans une réponse.

Dans une requête, la ligne initiale est composée de 3 parties :

- La **méthode de requête** (exemple : **GET, POST, HEAD, ...**)

Quelques méthodes de requête :

- GET : demande au serveur la ressource indiquée
- HEAD : ne demande que les entêtes mais pas la ressource
- POST : demande au serveur de modifier les informations qu'il stocke
- PUT : demande au serveur de créer ou de remplacer une de ses ressources
- DELETE : demande au serveur de supprimer une de ses ressources
- CONNECT : utilisé pour permettre aux connexions SSL de passer dans des connexions HTTP
- TRACE : demande au serveur de renvoyer les entêtes de la requête tels qu'il les a reçus

- L'**URL** de la ressource concernée
- Le **protocole** utilisé HTTP/x.x (exemple HTTP/1.1)

Exemple : GET /index.htm HTTP/1.1

Dans une réponse, la ligne initiale appelée **ligne status**, est composée de 3 parties :

- Le **protocole** utilisé HTTP/x.x (exemple HTTP/1.1)
- Le **code d'état** a 3 chiffres
- La **version textuelle** en langue anglaise correspondant à l'état

Exemple : HTTP/1.0 404 Not Found

**Les principaux codes d'état :**

Il existe 5 classes de codes d'état :

- 1xx : utilisés à bas niveau lors des transaction HTTP
- 2xx : la requête s'est bien passée (200 : OK)
- 3xx : la requête est correcte mais la ressource n'est plus là ou le serveur ne veut pas l'envoyer comme c'est le cas pour le code 304 qui signifie que la ressource n'a pas été modifiée depuis le dernier envoi donc il est inutile de la renvoyer.
- 4xx : la requête est incorrecte (exemple 404 : la ressource n'existe plus)
- 5xx : erreur du serveur (exemple script côté serveur de syntaxe incorrecte)

**Les champs d'entête :**

Les champs d'entête sont de la forme **champ d'entête : valeur**. Il en existe 16 différents pour HTTP 1.0 et 46

pour HTTP 1.1. Le seul requis pour le protocole HTTP 1.1 est le champ **HOST**.

<b>Accept- Language</b>	Langages acceptés par le navigateur
<b>Authorization</b>	Nom et mot de passe de l'utilisateur demandant la ressource
<b>Content- Base</b>	URL de base pour la résolution de toutes les URL relatives du document
<b>Content- Length</b>	Longueur du contenu de la requête. Utile pour la méthode POST dans les requêtes et pour les réponses en général (c'est notamment ce qui permet au navigateur d'afficher le pourcentage déjà chargé)
<b>Content- Type</b>	Nécessaire lorsque la méthode POST est utilisée. Valeur la plus courante : multipart/form-data
<b>Cookie</b>	Introduit par Netscape. Renvoie des couples de nom/valeur configurée par le serveur lors d'une réponse antérieure
<b>ETag</b>	Identificateur unique attribué à une ressource utilisé par les caches pour vérifier qu'une copie est obsolète ou pas.
<b>Host</b>	Nom de l'hôte cible
<b>If- Modified- Since</b>	Sert à faire des requêtes conditionnelles sur la date de dernière modification de la ressource
<b>If- None- Match</b>	Sert à faire des requêtes conditionnelles sur la valeur de l'Identity Tag (ETag)
<b>Keep- Alive</b>	Permet l'établissement de connexions permanentes. On trouve les paramètres timeout qui indique la durée au bout de laquelle la connexion sera libérée et max qui est le nombre maximum de requêtes traitées par connexion.
<b>Last- Modified</b>	Date et heure de la dernière modification de la ressource demandée
<b>Location</b>	Nouvel emplacement de la ressource
<b>Referer</b>	URL de la dernière page visitée, généralement celle d'où vient la requête
<b>Set- Cookie</b>	Demande la mémorisation par le navigateur d'une paire nom/valeur
<b>Transfer- Encoding</b>	Lorsque ce champ à la valeur <b>chunked</b> , cela indique que les données sont envoyées par blocs (chaque bloc sera précédé de la taille en octets)
<b>User- Agent</b>	Client (navigateur) utilisé

**La structure du corps :**

Pour une requête, le corps contiendra des couples de nom-valeur dans le cas de la méthode POST, ou le contenu d'un fichier dans le cas d'un upload de fichier vers le serveur. Pour une réponse, ce corps contient généralement une ressource destinée au navigateur comme du code HTML par exemple.

## HTTPS

Dans le sigle HTTPS, le S ajouté à HTTP signifie SECURE. Cela indique que le protocole HTTP utilise le protocole **SSL** (**Secure Socket Layer**) développé par Netscape et qui a été conçu pour assurer la sécurité des transactions sur internet. Ce protocole est intégré dans les navigateurs depuis 1994. Le protocole SSL en est déjà à sa version 3. On peut considérer que le protocole SSL s'insère entre le protocole HTTP et le protocole TCP.

SSL permet d'assurer les services de sécurité suivants :

- confidentialité obtenue par l'utilisation d'algorithmes à chiffrement symétrique de blocs.
- intégrité. L'intégrité des données est assurée par l'utilisation de MAC (Message Authentication Code) basés sur des fonctions de hachage.
- authentification des extrémités (client et/ou serveur)

Le déroulement d'une session SSL respecte le schéma suivant :

- Le client envoie un "client\_hello" avec une valeur générée aléatoirement et la liste des algorithmes de chiffrement supportés
- Le serveur répond avec un "server\_hello" avec également une valeur générée aléatoirement et l'algorithme de chiffrement choisi ainsi que son certificat, un message d'échange de clés
- Le client envoie un "premaster secret" qui sera utilisé pour le chiffrement
- Le client et le serveur calcul à partir de ce "premaster secret" leur "master secret" qui est une clé de 48 bits
- Le dialogue peut alors commencer, les données sont découpées en blocs, compressées et chiffrées en utilisant la fonction de hachage augmentée du "master secret"

## Transfert de fichiers

### FTP - TFTP

#### FTP

**FTP** signifie **File Transfer Protocol**, c'est un protocole de transfert de fichier qui utilise TCP. Le mode de fonctionnement est de type client-serveur. FTP permet les opérations suivantes :

- transfert de fichiers du serveur vers le client (download)
- transfert de fichiers du client vers le serveur (upload)
- renommage et suppression depuis le client de fichiers stockés sur le serveur
- listage depuis le client de répertoire situés sur le serveur

Le mode opératoire standard de FTP est le suivant :

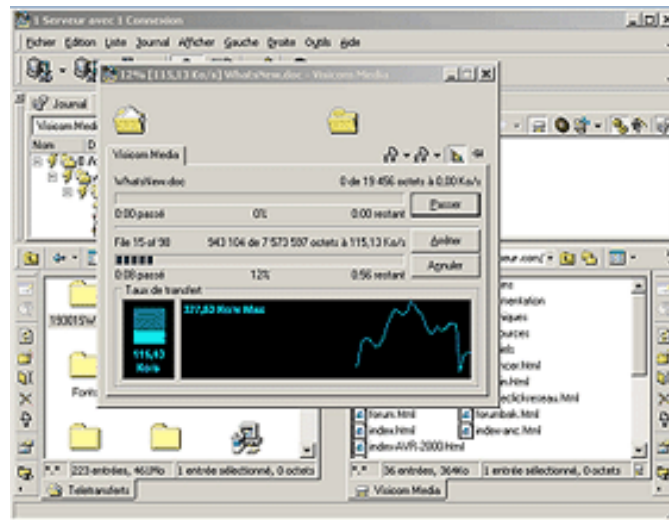
- Le client FTP initie une connexion avec le serveur FTP. Il doit s'authentifier auprès de ce serveur c'est à dire fournir un identifiant et un mot de passe.
- Le serveur FTP étant en écoute permanente de demande de connexion, il reçoit donc cette demande de

connexion . Le serveur initie donc une connexion TCP dite de connexion de contrôle qui servira à transférer les commandes TCP

- Chaque fois que le client FTP exécute une commande de transfert de données, il envoie au serveur FTP cette demande accompagnée du n° de port local utilisé
- Si le serveur TCP reçoit une commande de transfert de données, il initie une connexion dite connexion de données.

Généralement un transfert de fichier par ftp se fait en utilisant côté serveur les ports 21 pour les opérations de contrôle et le port 20 pour les données.

La commande FTP existe sur la plupart des systèmes d'exploitation mais il existe désormais des clients FTP qui permettent une utilisation simplifiée sans connaître la syntaxe des commandes.



Client FTP

## TFTP

**TFTP** signifie **Trivial File Transfer Protocol**, c'est un autre protocole de transfert de fichier mais qui utilise **UDP** comme protocole de transfert. Le protocole TFTP est plus simple à implanter que FTP mais il ne permet pas l'utilisation d'un répertoire utilisateur sur le serveur, ni celle d'un mot de passe garantissant une la protection des données. On utilise le protocole TFTP notamment pour amorcer des stations de travail sans disque dur.

## Quelques autres applications utilisant TCP/IP

[Telnet](#) - [SMTP](#) - [SNMP](#) - [NFS](#)

## TELNET

Ce protocole est utilisé pour émuler une connexion de terminal à un hôte distant. Le but de ce protocole est donc de transmettre les informations du clavier du client vers l'hôte distant et, dans l'autre sens, d'afficher les informations en retour sur l'écran du client. Telnet utilise **TCP** comme protocole de transport. Le mode de fonctionnement est de type client-serveur. Généralement côté serveur c'est le port 23 qui est utilisé.

Une connexion TELNET débute toujours par une phase de **négociation** qui a pour but de déterminer la configuration du client utilisé comme par exemple la façon dont les données vont être groupées avant d'être envoyées (ligne par ligne ou caractère par caractère).

Telnet utilise le concept de terminal virtuel qui permet de s'affranchir de la multiplicité des terminaux. Un terminal virtuel consiste à se doter d'une base de communication standard comprenant le codage des caractères ASCII et de quelques caractères de contrôle.

## SMTP

SMTP signifie **Simple Mail Transfer Protocol**, c'est le protocole standard permettant de transférer du courrier d'un serveur (on parle de serveur SMTP) à un autre. Le protocole SMTP fonctionne en **mode connecté**, le déroulement d'une connexion comprend toujours les étapes suivantes :

- L'ouverture de la session symbolisé par la commande HELO ou EHLO sur les versions plus récentes
- Un envoi MAIL FROM qui indique qui est l'expéditeur du message
- Un envoi RCPT TO qui indique le destinataire (s'il y a plusieurs destinataires, cette commande est répétée autant de fois que nécessaire)
- Un envoi DATA qui correspond au corps du message

Le port utilisé par défaut est le port 25.

La spécification de base du protocole SMTP indique que tous les caractères transmis dans un mail soient codés en ASCII sur 7 bits. Afin de pouvoir envoyer des caractères de la table ASCII étendue (accents, etc...) il est donc nécessaire d'utiliser des système de transcodage (spécifications **MIME : Multipurpose Internet Mail Extensions**) comme **base64** par exemple qui est utilisé pour coder les fichiers attachés ou **quoted- printable** pour les caractères spéciaux contenus dans le corps du message.

Il existe d'autres protocoles liés à la messagerie électronique :

- **POP (Post Office Protocol)**  
Ce protocole permet d'aller récupérer son courrier sur un serveur distant. Ce protocole est indispensable pour les personnes qui ne sont pas connectés directement à l'internet, afin de rapatrier leurs mails sur leur machine. La version POP3 gère l'authentification par nom d'utilisateur + mot de passe, par contre le cryptage n'est pas utilisé.
- **IMAP (Internet Mail Access Protocol)**  
Ce protocole considéré comme une alternative à POP offre des possibilités supplémentaires. La principale est que le client de courrier peut demander les en-têtes des messages au serveur, ou les corps de certains messages, ou la recherche de messages répondant à certains critères ce qui permet une plus grande souplesse d'utilisation.

## SNMP

SNMP signifie **Simple Network Management Protocol**, c'est donc un protocole utilisé pour contrôler les différents éléments d'un réseau. Le principe de fonctionnement de SNMP est encore une fois basé sur un modèle client-serveur.

Les différents éléments de SNMP sont :

- Une **station de gestion (Network Management Station)**. Cette station interroge les différents composants du réseau.

- Des composants de réseau (station, routeur, ponts, etc...) avec des **agents** qui sont en fait des programmes qui s'exécute sur les différents éléments du réseau.
- Des tables **MIB (Management Information Base)** qui sont en fait des bases de données gérées par chaque agent et qui contiennent des informations liées à la transmission d'informations au sein du réseau.

Le principe de fonctionnement de SNMP repose sur les **requêtes** que peut formuler la station de gestion et les réponses renvoyées par les différents agents interrogés. Pour répondre les agents utilisent bien sûr les données stockées dans leurs tables MIB. Il est prévu également que l'agent d'un élément de réseau qui passerait dans un état anormal prévienne la station de gestion par un message appelé **trap SNMP**.

SNMP s'appuie sur des services du protocole **UDP**.

## NFS

**NFS** signifie **Network File System**, c'est un protocole développé à l'origine par **Sun Microsystems**. Ce protocole permet à un serveur NFS d'exporter son **système de fichiers** vers des clients NFS. Chaque client NFS aura donc accès à l'arborescence de fichiers exportée par le serveur NFS comme s'il faisait partie de son propre système de fichiers. Les versions originales de NFS utilisaient le protocole **UDP**, les plus récentes s'appuient sur le protocole **TCP** plus adapté aux réseaux étendus. Au niveau de la **couche session** du modèle OSI, NFS utilise le protocole **RPC (Remote Procedure Call)** qui permet d'effectuer des **appels de procédure à distance**.

Pour l'utilisateur d'une partition NFS, il dispose de toutes les opérations de manipulation de fichiers ou de répertoires classique avec en plus des protocoles particuliers comme **Mount** qui implémente la **procédure de montage** de l'arborescence NFS.

# Bibliographie

## Les livres :

- **TCP/IP, Karanjit S.Siyan, CampusPress**  
Les protocoles IP et TCP sont très bien détaillés
- **TCP/IP, Joe Casad, Campus Press**  
Moins complet et technique que le précédent mais les protocoles de la couche Application sont présentés de façon assez complète

## Sur le Net :

- Les RFC de TCP/IP :

# Exercices et Tests

sommaire :

[Enoncés](#)

[Solutions](#)

## Exercice 1

Pour chacune des classes de réseaux IP, A, B et C, calculer le nombre théorique de réseaux possibles et le nombre théorique d'hôtes par réseau.

---

## Exercice 2

Que désignent les adresses suivantes (Classe de réseau, @ de la machine, @ du réseau, etc...) ?

- 137.14.0.0 : Classe B, réseau 137.14, cette adresse désigne ce réseau
- 137.14.0.255 : Classe B, réseau 137.14, hôte 0.255
- 137.14.255.255 : Classe B, réseau 137.14, diffusion
- 127.0.0.1 : Bouclage
- 0.0.146.245 : désigne l'hôte 146.245 dans le même réseau (classe B)
- 137.256.0.1 : adresse impossible ( $256 > 255$ )

---

## Exercice 3

Reprendre l'exercice 1 mais en tenant compte de la contrainte liée aux adresses spéciales. Donner pour chaque classe d'adresse l'intervalle des adresses de réseau. Evaluer le nombre théorique d'hôtes

Thierry Jeandel



adressables.

## Exercice 4

Décoder le datagramme IP ci-dessous et remplir le tableau suivant :

Version IP	
Longueur de l'entête (mots de 32 bits)	
Longueur totale du datagramme	
N° du datagramme	
Datagramme fragmenté ?	
TTL	
Protocole de niveau supérieur	
@IP source	
@IP destination	

Datagramme IP (codage hexadécimal) :

```
45 00 00 28
7d e5 40 00
80 06 a8 32
c1 36 29 2a
c1 36 29 21
0c e8 00 8b
02 95 0d 6a
03 83 12 35
50 10 1c b2
8b 40 00 00
```

## Exercice 5

Pour un réseau de classe C donner les différentes possibilités en matière de découpage en sous réseau en précisant pour chaque solution le nombre de sous réseau possible et le nombre d'hôtes par sous réseau ainsi que le nombre d'hôtes total.

---

## Exercice 6

Quel est le nombre maximum d'hôte d'un sous-réseau de classe B ? Quel est le nombre maximum de sous-réseaux que l'on peut créer à partir d'un réseau de classe A ?

---

## Exercice 7

Une entreprise a obtenu l'adresse de réseau 144.25.0.0, on veut structurer ce réseau en 20 sous-réseaux. Proposer un masque de sous-réseau et indiquer combien d'adresses seront utilisables dans chacun des sous-réseaux.

---

## Exercice 8

On a placé un analyseur de trames sur un ordinateur connecté à un réseau local. Le document suivant présente uniquement les trames concernant le protocole http. Reconstituer les étapes du dialogue.

```
GET /Deust/exo5.php HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint,
application/vnd.ms-excel, application/msword, */*
Accept-Language: fr,ru;q=0.5
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host: spmi2:8000
Connection: Keep-Alive
```

```
HTTP/1.1 200 OK
Date: Thu, 21 Nov 2002 14:36:53 GMT
Server: Apache/1.3.24 (Win32) PHP/4.2.0
X-Powered-By: PHP/4.2.0
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html
53
<html>
<head>
<title>Catalogue de photos</title>
</head>
....

GET /Deust/images/miniatures/s_image001.jpg HTTP/1.1
Accept: */*
Referer: http://spmi2:8000/Deust/exo5.php
Accept-Language: fr,ru;q=0.5
Accept-Encoding: gzip, deflate
If-Modified-Since: Wed, 07 Aug 2002 20:40:52 GMT
If-None-Match: "0-1b36-3d5185d4"
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host: spmi2:8000
Connection: Keep-Alive
```

```
HTTP/1.1 304 Not Modified
Date: Thu, 21 Nov 2002 14:36:53 GMT
Server: Apache/1.3.24 (Win32) PHP/4.2.0
Connection: Keep-Alive
Keep-Alive: timeout=15, max=99
ETag: "0-1b36-3d5185d4"
```

```
GET /Deust/images/miniatures/s_fdfd.jpg HTTP/1.1
Accept: */*
Referer: http://spmi2:8000/Deust/exo5.php
Accept-Language: fr,ru;q=0.5
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host: spmi2:8000
Connection: Keep-Alive
```

```
HTTP/1.1 404 Not Found
Date: Thu, 21 Nov 2002 14:36:54 GMT
Server: Apache/1.3.24 (Win32) PHP/4.2.0
Keep-Alive: timeout=15, max=77
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1
```

```
156
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>404 Not Found</TITLE>
</HEAD><BODY>
<H1>Not Found</H1>
The requested URL /Deust/images/miniatures/s_fdfd.jpg was not found on this server.<P>
<HR>
<ADDRESS>Apache/1.3.24 Server at <A HREF="mailto:didier.croutz@univ-nancy2.fr">Spmi2</A> Port
8000</ADDRESS>
</BODY></HTML>0
```

```
POST /Deust/exo5.php HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint,
```

Thierry Jeandel

```
application/vnd.ms-excel, application/msword, */*
Referer: http://spmi2:8000/Deust/exo5.php
Accept-Language: fr,ru;q=0.5
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host: spmi2:8000
Content-Length: 9
Connection: Keep-Alive
coupure=6

HTTP/1.1 200 OK
Date: Thu, 21 Nov 2002 14:36:58 GMT
Server: Apache/1.3.24 (Win32) PHP/4.2.0
X-Powered-By: PHP/4.2.0
Keep-Alive: timeout=15, max=75
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html
53
<html>
<head>
<title>Catalogue de photos</title>
</head>
<body>
...

GET /Deust/exo5_a.php?numphot=4 HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint,
application/vnd.ms-excel, application/msword, */*
Referer: http://spmi2:8000/Deust/exo5.php
Accept-Language: fr,ru;q=0.5
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host: spmi2:8000
Connection: Keep-Alive
```

---

## Solution de l'Exercice 1

Classe A :

Nombre de réseaux : De 000 0001 à 111 1111 , ce qui donne 127 réseaux.

Nombre d'hôtes:  $2^{24} = 16,7$  millions d'hôtes / réseau

Classe B:

Nombre de réseaux : , ce qui donne réseaux.

Nombre d'hôtes:  $2^{16} = 65\,538$  hôtes / réseau

Classe C:

Nombre de réseaux : De , ce qui donne réseaux.

Nombre d'hôtes:  $2^8 = 256$  hôtes / réseau

---

## Solution de l'Exercice 2

- 137.14.0.0 : Classe A
- 137.14.0.255
- 137.14.255.255
- 127.0.0.1
- 0.0.146.245
- 137.256.0.1

---

## Solution de l'Exercice 3

Classe A : 126 réseaux et 16,7 Millions d'hôtes/réseaux. De 1.x.x.x à 126.x.x.x

Classe B : 16384 réseaux et 65 534 hôtes/réseaux. De 128.x.x.x à 191.x.x.x

Classe C : 2 097 152 réseaux et 254 hôtes/réseaux. De 192.x.x.x à 223.x.x.x

Ce qui donne un espace d'adressage théorique de 3,7 Milliards d'hôtes environ.

---

## Solution de l'Exercice 4

Longueur de l'entête (mots de 32 bits)	5
Longueur totale du datagramme	40 octets
N° du datagramme	7d e5
Datagramme fragmenté ?	Non
TTL	128
Protocole de niveau supérieur	TCP
@IP source	193.54.41.42
@IP destination	193.54.41.33

## Solution de l'Exercice 5

Dans un sous réseau de classe C, 8 bits sont alloués à l'adresse de l'hôte.  
 Dans le cadre d'un découpage en sous réseau, on a donc les possibilités suivantes :

Nb de bits réservé au sous réseau	Nb de sous réseaux	Nb d'hôtes par sous réseau	Nb d'hôtes total
1	Impossible	Impossible	Impossible
2	2	62	124
3	6	30	180
4	14	14	56
5	30	6	180
6	62	2	124
7	Impossible	Impossible	Impossible
8	Impossible	Impossible	Impossible

---

## Solution de l'Exercice 6

Pour un réseau de classe B, l'host-id tient sur 2 octets donc 16 bits, lors d'un découpage en sous-réseaux, il faut prendre au minimum 2 bits pour le sous-réseau, ce qui en laisse 14 pour le n° d'hôte, sur 14 bits on code  $2^{14} - 2$  hôtes, c'est à dire 16 382 hôtes.

Pour un réseau de classe A, l'host-id tient sur un 3 octets donc 24 bits, lors d'un découpage en sous-réseaux, on peut prendre au maximum 22 bits (car il faut disposer d'au moins 2 bits pour coder les hôtes des sous-réseaux), sur 22 bits on peut coder  $2^{22} - 2$  sous-réseaux c'est à dire 4 194 302 sous-réseaux mais dans chacun de ces sous-réseaux on ne disposera que de 2 adresses !!

---

## Solution de l'Exercice 7

Le réseau 144.25.0.0 est un réseau de classe B, le masque par défaut est 255.255.0.0. Pour obtenir 20 sous-réseaux, il faut "prendre" sur la partie hostid 5 bits, car sur 5 bits on peut coder  $2^5$  valeurs donc  $32 - 2 = 30$  sous-réseaux (Rappel : sur 4 bits, on ne peut coder que  $2^4 - 2 = 14$  sous réseaux).

Le masque à utiliser sera donc : 255.255.248.0 (248 est codé 1111 1000).

Dans chaque sous réseau, on dispose de  $2^{11} - 2 = 2046$  adresses.

---

## Solution de l'Exercice 8

```
GET /Deust/exo5.php HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint,
application/vnd.ms-excel, application/msword, */*
Accept-Language: fr,ru;q=0.5
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host: spmi2:8000
```

Connection: Keep-Alive

**Il s'agit de la demande de la page `exo5.php` (dans le dossier ou l'alias `Deust`) sur l'hôte `spmi2` en utilisant le port `8000`. C'est une connexion permanente qui est demandée. Le navigateur est Internet Explorer 5.5, il accepte les langages français et russe ainsi que différents formats d'images et d'applications comme Word, Excel et PowerPoint.**

```
HTTP/1.1 200 OK
Date: Thu, 21 Nov 2002 14:36:53 GMT
Server: Apache/1.3.24 (Win32) PHP/4.2.0
X-Powered-By: PHP/4.2.0
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html
53
<html>
<head>
<title>Catalogue de photos</title>
</head>
....
```

**C'est la réponse du serveur. Le serveur est un serveur Apache installé sous Windows. Le code envoyé est de l'HTML généré par un script PHP. La connexion permanente est acceptée avec un timeout de 15 secondes et un nombre maxi de requêtes de 100. Les données seront découpées en blocs et envoyées par plusieurs segment TCP.**

```
GET /Deust/images/miniatures/s_image001.jpg HTTP/1.1
Accept: */*
Referer: http://spmi2:8000/Deust/exo5.php
Accept-Language: fr,ru;q=0.5
Accept-Encoding: gzip, deflate
If-Modified-Since: Wed, 07 Aug 2002 20:40:52 GMT
If-None-Match: "0-1b36-3d5185d4"
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host: spmi2:8000
Connection: Keep-Alive
```

**Le navigateur demande le chargement d'une image `s_image001.jpg`. Ce lien se trouve sur la page `exo5.php`. Si la date de dernière modification de la ressource est antérieure au 7 Août 2002, il est inutile de renvoyer cette image. De même il est inutile de renvoyer l'image si l'ETag est `"0-1b36-3d5185d4"`.**

```
HTTP/1.1 304 Not Modified
Date: Thu, 21 Nov 2002 14:36:53 GMT
Server: Apache/1.3.24 (Win32) PHP/4.2.0
Connection: Keep-Alive
Keep-Alive: timeout=15, max=99
ETag: "0-1b36-3d5185d4"
```

**Réponse du serveur : La ressource ne sera pas renvoyée puisqu'elle n'a pas été modifiée et que le ETag est toujours le même.**

```
GET /Deust/images/miniatures/s_fdfd.jpg HTTP/1.1
Accept: */*
Referer: http://spmi2:8000/Deust/exo5.php
Accept-Language: fr,ru;q=0.5
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host: spmi2:8000
Connection: Keep-Alive
```



**Le navigateur demande le chargement d'une image s\_fdfd.jpg. Cette image ne se trouve pas dans le cache du navigateur.**

```
HTTP/1.1 404 Not Found
Date: Thu, 21 Nov 2002 14:36:54 GMT
Server: Apache/1.3.24 (Win32) PHP/4.2.0
Keep-Alive: timeout=15, max=77
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1
```

```
156
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>404 Not Found</TITLE>
</HEAD><BODY>
<H1>Not Found</H1>
The requested URL /Deust/images/miniatures/s_fdfd.jpg was not found on this server.<P>
<HR>
<ADDRESS>Apache/1.3.24 Server at <A HREF="mailto:didier.croutz@univ-nancy2.fr">Spmi2</A> Port
8000</ADDRESS>
</BODY></HTML>0
```

**Réponse du serveur : l'image est introuvable. un message d'erreur en HTML est envoyé avec cette réponse.**

```
POST /Deust/exo5.php HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint,
application/vnd.ms-excel, application/msword, */*
Referer: http://spmi2:8000/Deust/exo5.php
Accept-Language: fr,ru;q=0.5
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host: spmi2:8000
Content-Length: 9
Connection: Keep-Alive
```

```
coupure=6
```

**Le navigateur envoie un formulaire (en méthode POST), le script qui traite la page est exe5.php. Contenu des données du formulaire : "coupure=6"**

```
HTTP/1.1 200 OK
Date: Thu, 21 Nov 2002 14:36:58 GMT
Server: Apache/1.3.24 (Win32) PHP/4.2.0
X-Powered-By: PHP/4.2.0
Keep-Alive: timeout=15, max=75
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html
53
<html>
<head>
<title>Catalogue de photos</title>
</head>
<body>
...
```

**Le serveur répond en générant la page correspondante (la valeur coupure=6 a sans doute été utilisée par le script php qui a généré cette page).**

**On remarque que le nombre maximum de requête dans le champ Keep-Alive diminue au fur et à mesure.**

Thierry Jeandel

```
GET /Deust/exo5_a.php?numphot=4 HTTP/1.1
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint,
application/vnd.ms-excel, application/msword, */*
Referer: http://spmi2:8000/Deust/exo5.php
Accept-Language: fr,ru;q=0.5
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host: spmi2:8000
Connection: Keep-Alive
```

**Le navigateur envoie une requête en méthode GET avec le paramètre "numphot" qui est égal à 4.**

Ministère de l'Enseignement Supérieur et des recherches scientifiques  
Université Virtuelle de Tunis

Intitulé du chapitre :

**Technologies des réseaux de communication**

Nom de l'auteur :

**Gérard-Michel Cochard & Edoardo Berera  
& Michel Besson Thierry Jeandel**

Cette ressource est la propriété exclusive de l'UVT. Il est strictement interdit de la reproduire à des fins commerciales. Seul le téléchargement ou impression pour un usage personnel (1 copie par utilisateur) est permis.

# Commutation de circuits, commutation de paquets

Sommaire :

[Introduction](#)

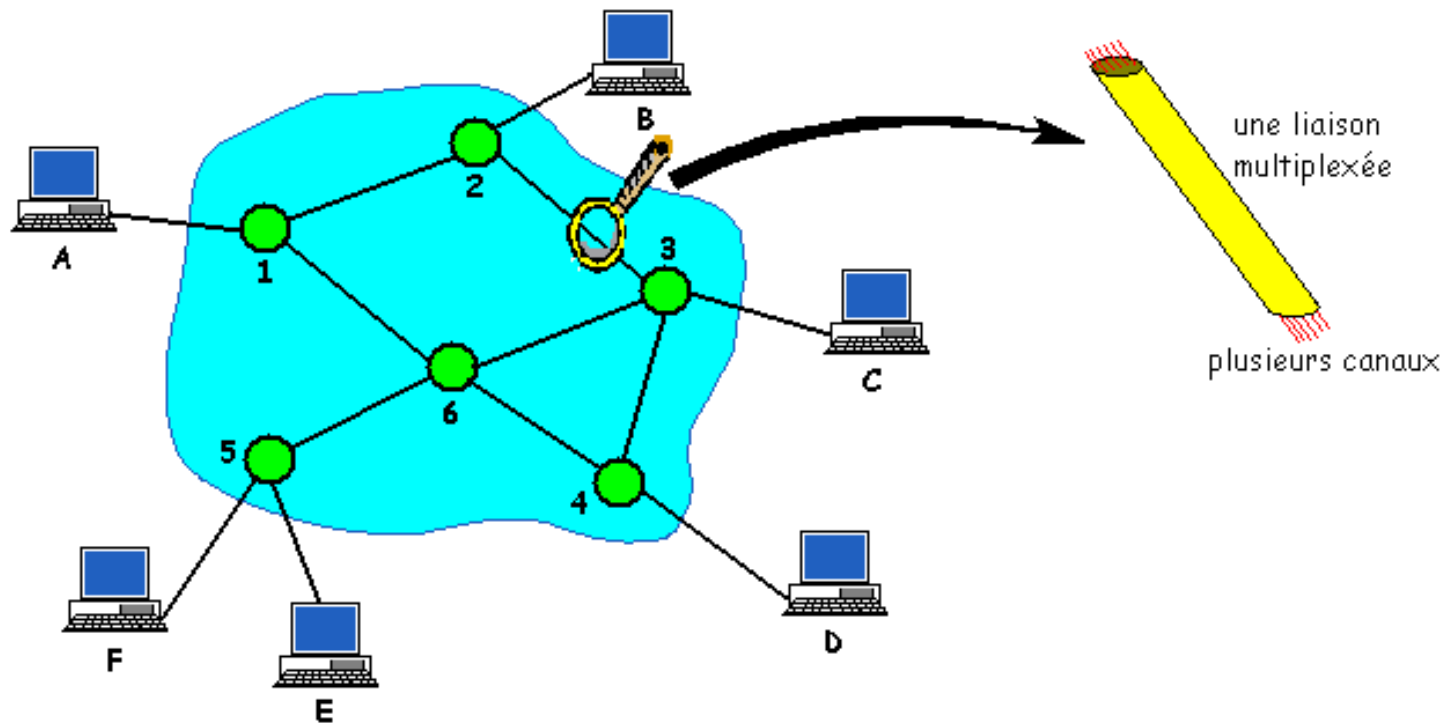
[Commutation de circuits](#)

[Commutation de paquets](#)

[X25](#)

## Introduction

Pour transmettre des informations au-delà d'un réseau local, il est nécessaire d'utiliser un réseau commuté qui est un réseau partiellement maillé, comportant des noeuds de commutation. Les stations qui échangent des informations doivent être reliées chacune à un noeud de commutation. Il existe deux grands types de réseaux commutés : les réseaux à commutation de circuits et les réseaux à commutation de paquets. Ils sont étudiés en détail plus loin



Pour transmettre des informations au-delà d'un réseau local, il est nécessaire d'utiliser un réseau commuté qui est un réseau partiellement maillé, comportant des noeuds de commutation. Les stations qui échangent des informations doivent être reliées chacune à un noeud de commutation. Il existe deux grands types de réseaux commutés : les réseaux à commutation de circuits et les réseaux à commutation de paquets. Ils sont étudiés en détail plus loin.

Le maillage d'un réseau commuté n'est pas total ce qui serait irréaliste (chaque noeud n'est pas relié directement à tous les autres noeuds). Il existe donc pour chaque noeud quelques liaisons directes avec d'autres noeuds, appelés noeuds voisins. Le choix des liaisons résulte d'une analyse en coûts, en charges et en sécurité.

Les noeuds ont pour vocation essentielle de recevoir des informations par une liaison et de les diriger vers un autre noeud par une autre liaison de manière à les acheminer au destinataire (fonction routage). Les informations vont donc passer de noeud en noeud pour arriver à destination. Certains noeuds sont des noeuds d'entrée-sortie du réseau : des stations leur sont attachées. Ces noeuds ont donc une fonction supplémentaire de réception/délivrance de données.

Les liaisons entre noeuds sont généralement optimisées : elles sont multiplexées, soit de manière spatiale (FDM : Frequency Division Multiplexing), soit de manière temporelle (Time Division Multiplexing).

## Commutation de circuits

Dans ce mode de commutation, un chemin (circuit, channel) est construit entre l'émetteur et le récepteur à partir des liaisons du réseau commuté. Ce circuit est "temporaire" dans la mesure où il n'a d'existence que sur la durée de la communication entre émetteur et récepteur. Il est ensuite libéré de manière à ce que les liaisons puissent être utilisées dans le cadre d'une autre communication.

Une communication, via un réseau à commutation de circuits nécessite donc 3 phases :

- la connexion : construction du circuit

Il faut au préalable construire un circuit entre les deux stations à faire communiquer. La station émettrice envoie une demande de connexion au noeud le plus proche. Celui-ci réceptionne cette demande, l'analyse et suivant les règles de routage choisit un canal (et le réserve) vers le noeud voisin le plus adéquat vers lequel la demande de connexion est transmise. le processus de poursuit ainsi jusqu'au noeud de rattachement de la station réceptrice, et donc jusqu'à cette station (on vérifie aussi que cette station est prête à accepter la connexion).

- le transfert des données

Le circuit de bout en bout étant défini et construit, les données peuvent être échangées entre les deux stations (le circuit est généralement full duplex) comme si ces stations étaient reliées directement.

- la déconnexion : libération des liaisons du circuit virtuel

A la fin du transfert de données, l'une des stations peut prendre l'initiative de libérer le circuit. L'avis de déconnexion est transmis de noeud en noeud et les différents canaux mobilisés pour la communication sont libérés.

Plusieurs remarques doivent être faites sur ce mode de commutation :

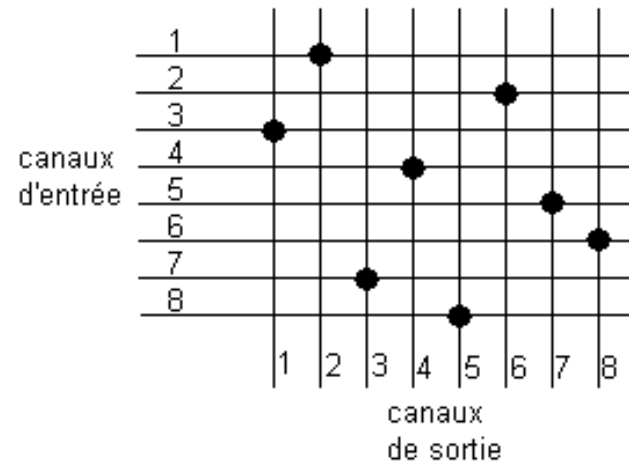
- Pour un réseau "chargé", il doit y avoir suffisamment de canaux sur les liaisons entre les noeuds pour pouvoir satisfaire les demandes de connexion et donc la construction de circuits.
- Une bonne rentabilité du réseau suppose que le circuit soit pleinement utilisé durant la communication. Ce n'est généralement pas le cas lors d'une application conversationnelle (il y a beaucoup de "blancs") ; c'est par contre le cas pour le transfert de la voix.
- La demande de connexion et l'avis de déconnexion demandent un délai supplémentaire à celui du transfert de données.

Le plus connu et le plus ancien des réseaux à commutation de circuits est le réseau téléphonique (RTC ou Réseau Téléphonique Commuté) qui, par la suite a été également utilisé pour la transmission de données.

Examinons maintenant les aspects technologiques liés à la commutation de circuit : il y en a trois principaux : la structure d'un commutateur, le routage et la signalisation.

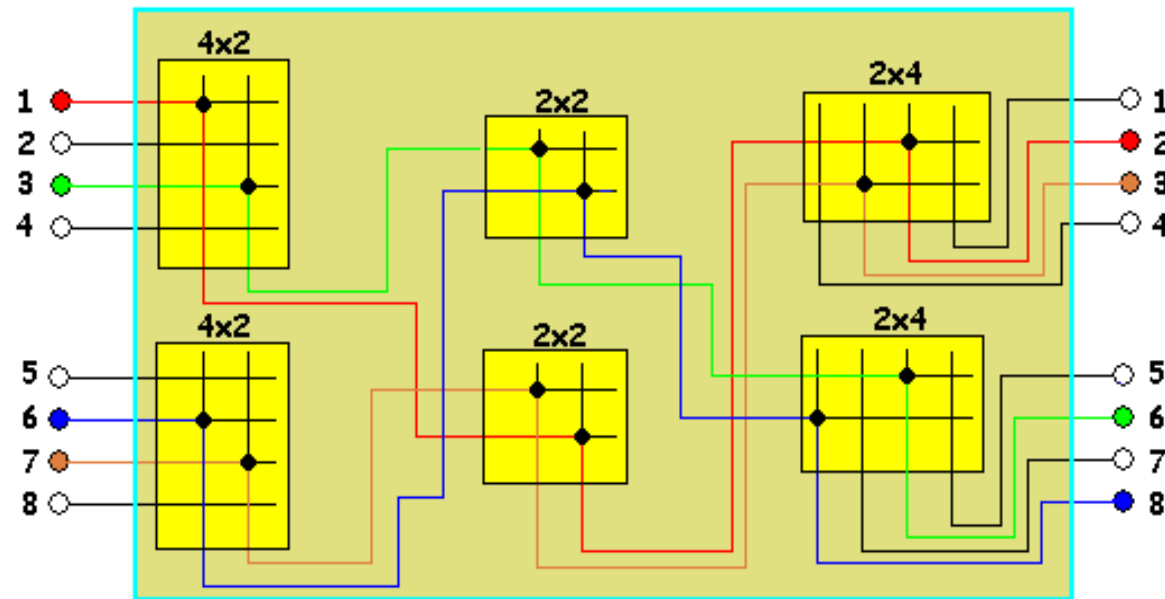
Un commutateur possède des lignes "entrée" et des lignes "sorties". A chaque ligne "entrée", le commutateur a pour rôle de faire correspondre une ligne "sortie". La structure la plus simple est une matrice d'interconnexion (crossbar matrix).

La figure ci-dessous montre un commutateur 8x8 8 lignes en entrée et 8 lignes en sortie. Les points de commutation sont actifs ou passifs suivant les besoins de commutation.



On notera que la possibilité de connexion d'une ligne d'entrée à une libre de sortie non occupée est toujours possible et que le nombre de points de connexion est  $n^2$  si le nombre de lignes d'entrée ou de sortie est égal à  $n$ . Ceci peut poser problème lorsque le nombre de lignes entrée/sortie devient grand.

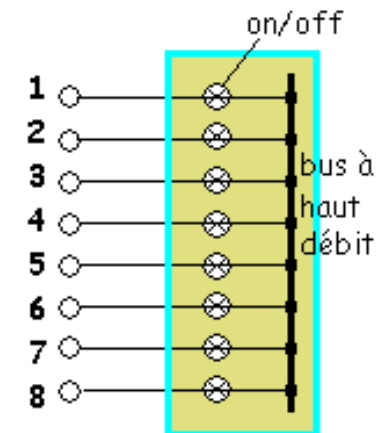
Une manière de réduire le nombre de points de connexion est de prévoir plusieurs "étages" de commutation. La figure ci-dessous en donne un exemple pour le cas de 8 lignes d'entrée et de 8 lignes de sortie.



Le nombre de points de commutation est réduit à 40 au lieu de 64. Le prix à payer est cependant un certain blocage de lignes puisque toutes les liaisons ne sont plus possibles (par exemple la station 5 ne peut être reliée à aucune autre station).

Dans les commutateurs précédents, il y a une communication par ligne interne. On peut cependant utiliser les "bienfaits" de la numérisation pour proposer des commutateurs basés sur TDM, c'est à dire sur un multiplexage temporel. Imaginons  $n$  lignes d'entrée/sortie bidirectionnelles à un débit  $d$  chacune. Le commutateur utilise un bus à haut débit. Chaque ligne d'entrée est dotée d'un "slot" de temps pour envoyer des données sous forme d'un bloc de bits sur le bus. Ces slots sont multiplexés avec TDM sur le bus en une trame (comportant donc  $n$  slots pleins ou vides). De la même manière, une ligne d'entrée retire le contenu du slot qui lui est assigné.

Bien entendu le débit du bus doit être supérieur à  $n \times d$ .

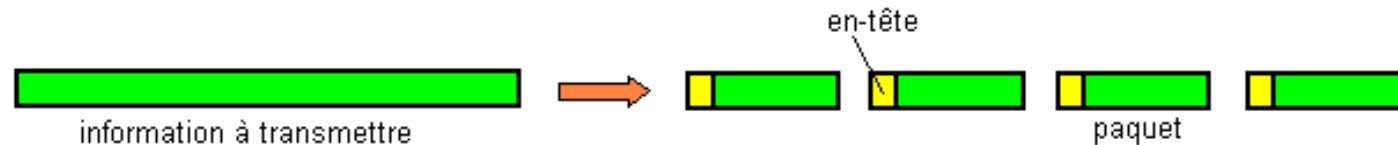




## Commutation de paquets

La commutation de paquets est apparue vers 1970 pour résoudre le problème de la transmission de données numériques sur de longues distances. La commutation de circuits était en effet relativement inadaptée (mais parfaitement adaptée pour la voix) à la transmission de données numériques ; d'une part, la communication entre systèmes informatiques comporte de nombreux "silences" et la voie de transmission, si elle est réservée en totalité à cette communication, n'est donc pas utilisée à 100% ; par ailleurs, la commutation de circuits s'effectue à débit constant ce qui contraint énormément les équipements (serveurs, stations) qui possèdent des possibilités différentes en débit.

Dans la commutation de paquets, un bloc d'information à transmettre est découpé en paquets. Un paquet comporte donc une fraction de l'information à transmettre mais aussi un champ de contrôle, généralement placé en début de paquet (en-tête).

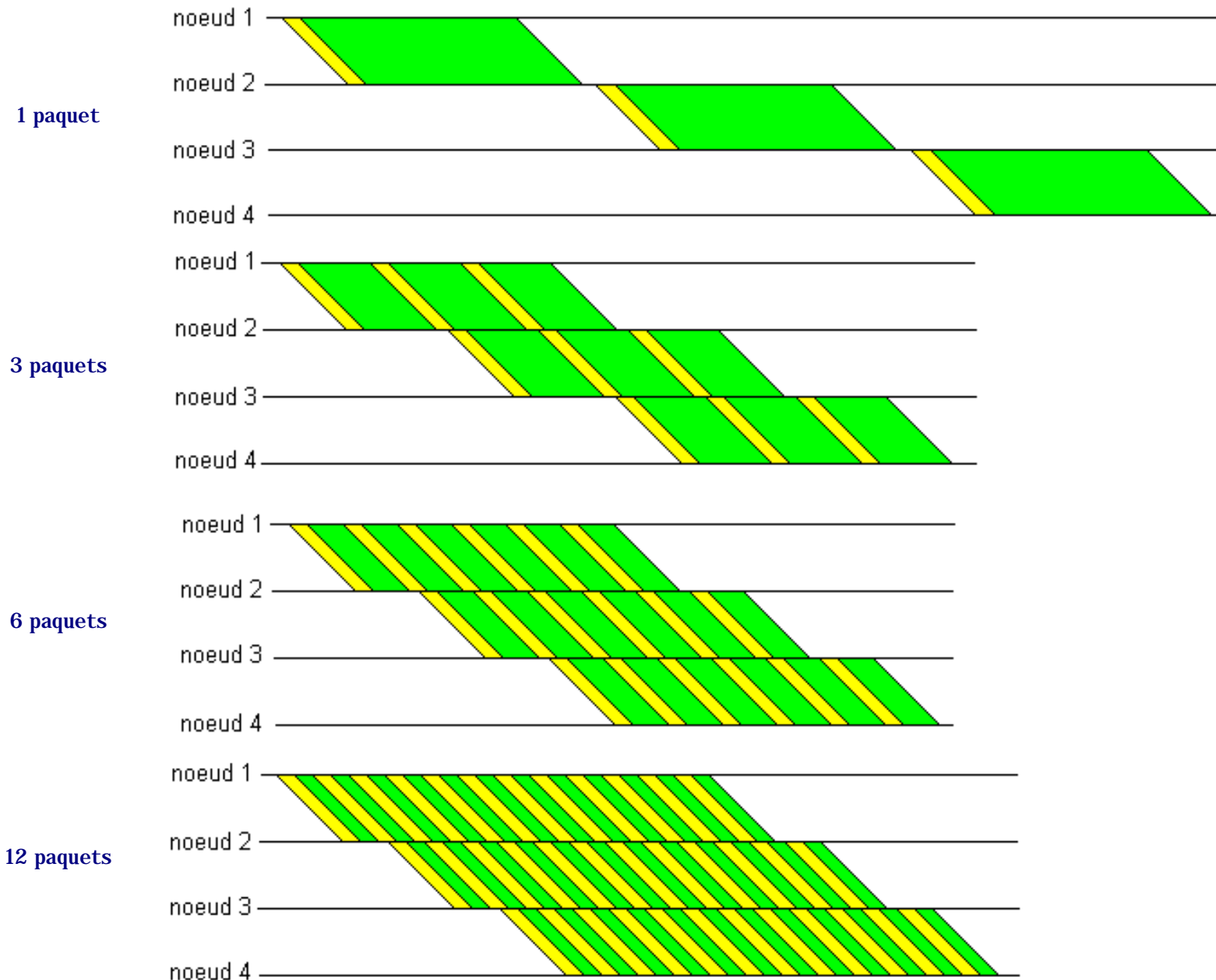


Dans un réseau à commutation de paquets, un noeud de commutation a pour rôle de recevoir les paquets entrants, d'examiner les en-têtes et les destinations, de choisir une voie de sortie pour chaque paquet, de mettre les paquets reçus dans les files d'attente adéquates pour leur acheminement. On notera, en particulier que des paquets provenant de messages différents peuvent être multiplexés (multiplexage temporel) sur une même liaison et que les débits des différentes liaisons peuvent être différents.

Rappelons que les paquets peuvent être acheminés suivant deux modes différents :

- le mode "circuit virtuel" : un chemin entre le noeud entrant et le noeud destination est construit (établissement du circuit virtuel), puis tous les paquets d'un même message suivent ce chemin ; ils arrivent donc dans l'ordre où ils ont été émis (acheminement en séquence).
- le mode "datagramme" : chaque paquet est traité indépendamment des autres ; les paquets n'arrivent donc pas nécessairement dans le même ordre que celui de l'émission ; ils doivent être remis en séquence pour délivrance au destinataire.

Le diagramme ci-dessous montre l'intérêt de la commutation de paquets par rapport à la commutation de circuits en ce qui concerne le délai de transmission. L'exemple est basé sur un message de 36 octets ; l'en-tête des paquets est supposé être de 3 octets. On considère plusieurs possibilités : 1 seul paquet de 36 octets utiles ; 3 paquets de 12 octets utiles ; 6 paquets de 6 octets utiles ; 12 paquets de 3 octets utiles. Pour la transmission de chaque paquet on doit prendre en considération le délai d'acheminement (en principe à la vitesse de la lumière) et le temps de traitement de chaque paquet (examen de l'en-tête et stockage dans un buffer) ; il faudrait y ajouter le temps d'attente avant ré-émission (le paquet est placé dans une file d'attente) ; le temps d'attente est négligé dans le diagramme.



Ce diagramme nous apprend que la taille du paquet doit être choisie de manière optimale : nous voyons, en effet, que la taille de 6 octets utiles correspond ici au meilleur délai d'acheminement. Si la taille est plus petite, l'information de contrôle (en-tête) est plus importante car plus fréquente.

L'acheminement des paquets nécessitent une fonction routage au niveau des noeuds de commutation. Cette fonction permet de diriger un paquet entrant vers le

noeud voisin le plus approprié à l'acheminement du paquet vers sa destination finale. Plusieurs méthodes de routage peuvent être envisagées. Nous passons en revue brièvement les principales méthodes.

- routage par inondation

Un paquet entrant dans un noeud de commutation est identifié (on garde trace de son identification) puis des copies sont envoyées sur toutes les autres voies possibles (autres que la voie entrante). Le paquet va ainsi se multiplier sur le réseau et transiter par toutes les liaisons du réseau. Il finira donc par arriver au destinataire. Deux mécanismes permettent d'éviter une prolifération trop grande des paquets :

- quand un paquet déjà identifié revient sur un noeud, celui-ci le reconnaît et le re-transmet pas.
- un paquet est doté d'un compteur de hops (passage sur un noeud de commutation) ; ce compteur est fixé à une valeur initiale au niveau du noeud source du réseau ; chaque fois que le paquet traverse un noeud, le compteur est décrémenté ; lorsque le compteur arrive à zéro, le paquet est détruit.

Bien que surprenant au premier abord, le routage par inondation est très efficace ; il peut être utilisé pour certaines applications militaires.

- routage statique

Une étude globale du réseau est supposée avoir été effectuée et, par emploi d'algorithmes de "moindre coût", les routes optimales entre les divers noeuds ont été établies. On sait ainsi, pour une destination donnée, diriger en chaque noeud un paquet vers le noeud voisin adéquat, ce qui signifie que chaque noeud est doté d'une table de routage (destination, noeud voisin). La méthode est simple ; toutefois elle ne prend pas en compte les modifications possibles du réseau, à savoir la rupture d'une liaison ou l'engorgement d'une liaison.

- routage aléatoire

Pour chaque noeud, on peut envisager de choisir la voie de sortie "au hasard". En fait, on utilise des probabilités permettant une certaine optimalité. Les débits des voies n'étant pas nécessairement les mêmes, on peut définir les probabilités suivantes :  $p_i = D_i / \sum D_k$  où  $D_i$  est le débit d'une voie sortante et où la somme des débits  $D_k$  porte sur toutes les voies sortantes (autres que la voie d'entrée du paquet).

- routage adaptatif

Il s'agit d'une forme plus avancée permettant une adaptation du routage à l'état du réseau. Le routage adaptatif se base sur l'échange d'information entre les noeuds du réseau. Ces informations permettent de mettre à jour les tables de routage. Bien entendu, l'échange d'information doit être assez fréquent pour parer à toute éventualité ; a contrario, l'échange d'information contribue de manière non négligeable à augmenter la charge du réseau.

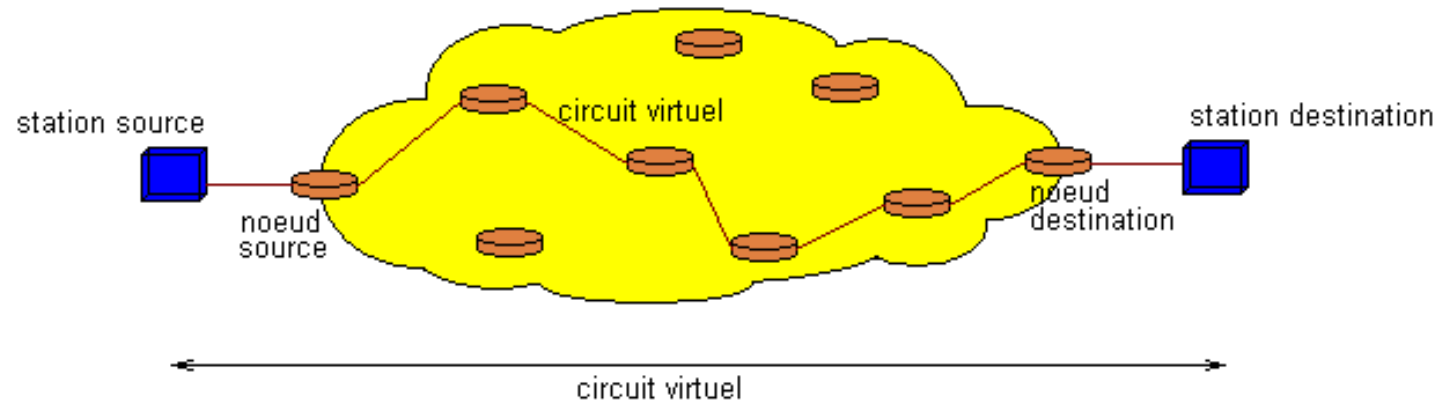
Le contrôle de congestion est une fonction du réseau qui permet d'éviter les problèmes d'engorgement de certaines artères du réseau. Les mécanismes employés sont divers. On peut déjà considérer que le routage adaptatif est une manière de limiter les circonstances de congestion car l'envoi d'informations permet de déterminer les points noirs du réseau et, donc, de prendre les décisions adéquates (par exemple diminution de débit) ; une autre façon de procéder est l'envoi d'un paquet de contrôle par un noeud "congestionné" à tous les autres noeuds ; cette méthode a le gros inconvénient d'augmenter aussi la charge du réseau. Une

variante plus intéressante est l'incorporation d'information de congestion dans les paquets d'information ; on peut ainsi faire remonter des informations de congestion vers l'amont ou l'aval d'un noeud congestionné.

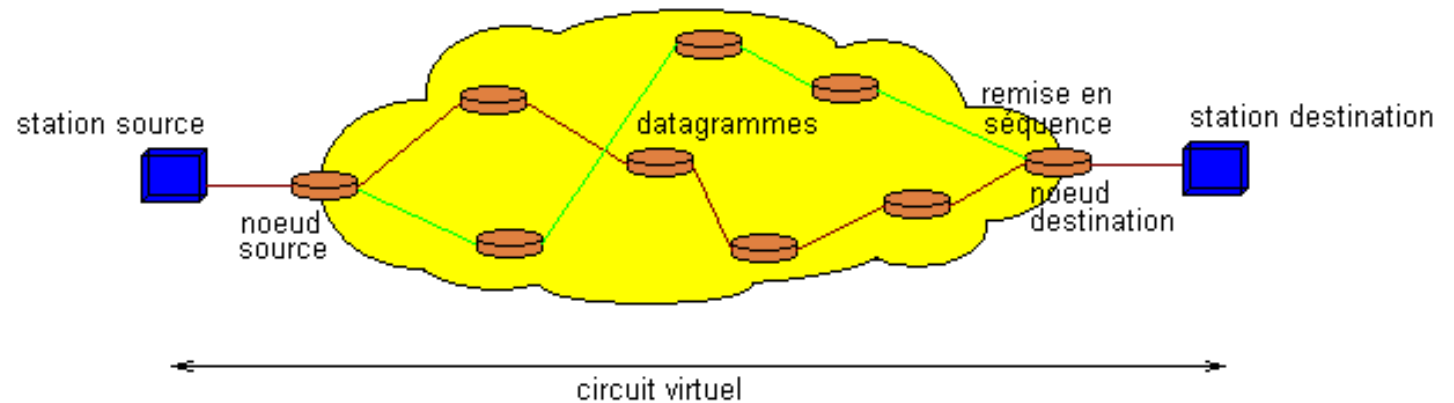
Dans la transmission de paquets, on définit deux modes de service : le service **orienté connexion** et le service **non connecté**. Dans un service orienté connexion, les paquets suivent une connexion logique entre une station source et une station destination ; les paquets transitent donc "en séquence" sur cette connexion logique (appelée circuit virtuel). Dans un service non connecté, les paquets émis d'une station source sont transmis de manière indépendante à la station destination ; ils n'arrivent donc pas nécessairement en séquence.

Bien que la terminologie des services soit analogue à celle des modes de transmission de paquets (circuit virtuel et datagramme), il faut bien comprendre que les deux types de services sont indépendants des types de réseau utilisés. On peut en effet trouver les quatre situations suivantes :

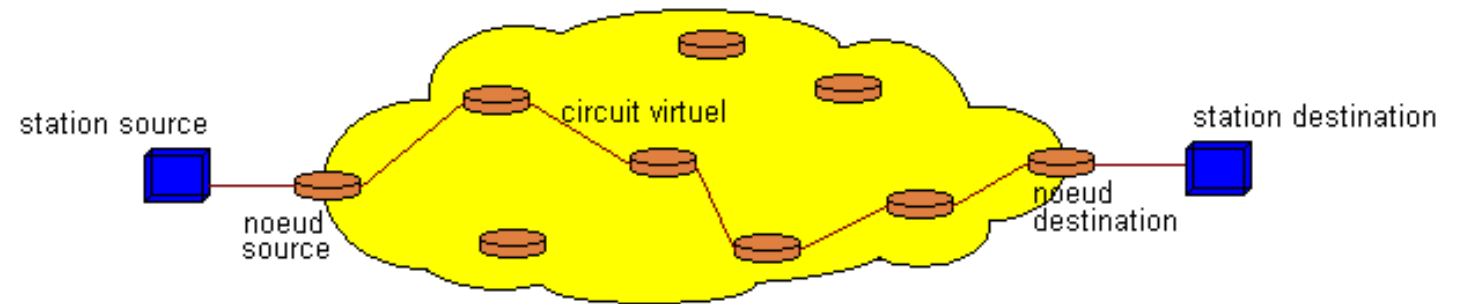
service orienté connexion via un réseau à  
commutation de paquets de type circuit virtuel



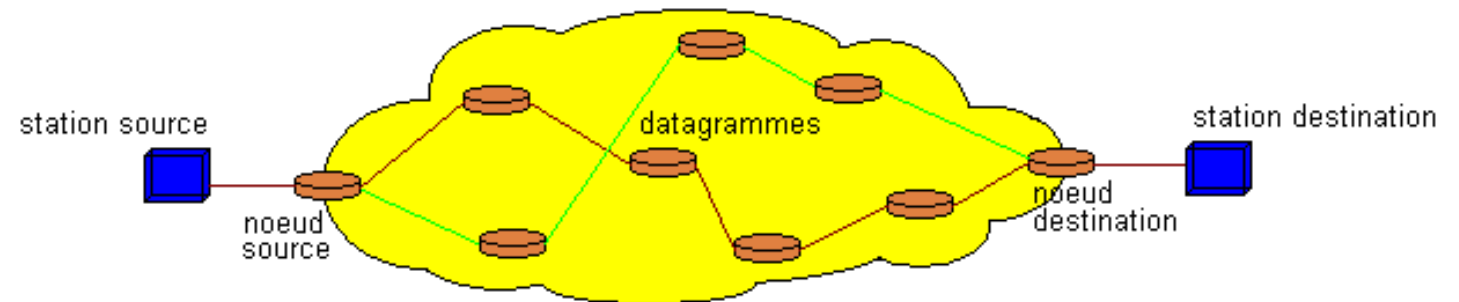
service orienté connexion via un réseau à  
commutation de paquets de type datagramme



service non connecté via un réseau à  
commutation de paquets de type circuit virtuel



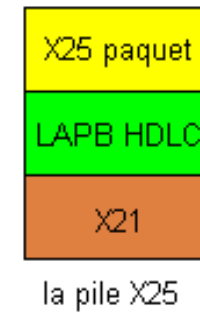
service non connecté via un réseau à  
commutation de paquets de type datagramme



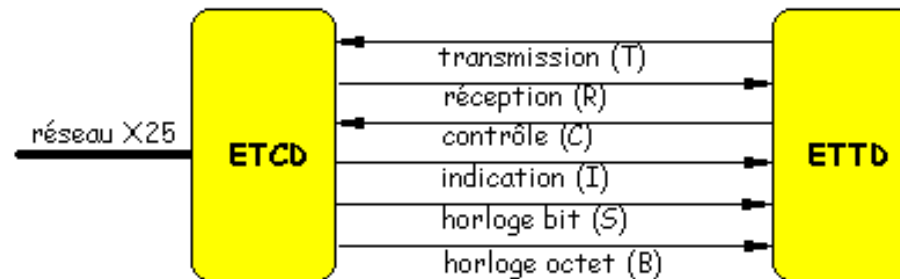
## X25

Le protocole X25 est utilisé dans plusieurs réseaux à commutation de paquets, notamment le réseau français TRANSPAC. Il s'agit en fait d'une pile de protocoles X25 résultant d'une normalisation par le CCITT (1976) et qui concerne les trois couches basses de l'ISO/OSI :

- couche physique : X21,
- couche liaison : variante LAPB (Link Access Protocol - Balanced) de HDLC,
- couche réseau : X25.paquet.

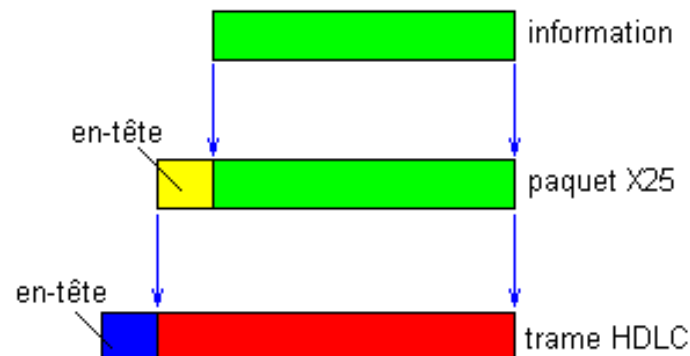


Le protocole X21 définit la jonction entre un ETTD (Equipement Terminal de Traitement de Données = ordinateur) et un point d'entrée sur un réseau X25, constitué d'un ETCd (Equipement Terminal de Circuit de Données). Le schéma ci-dessous définit l'interface X21 :

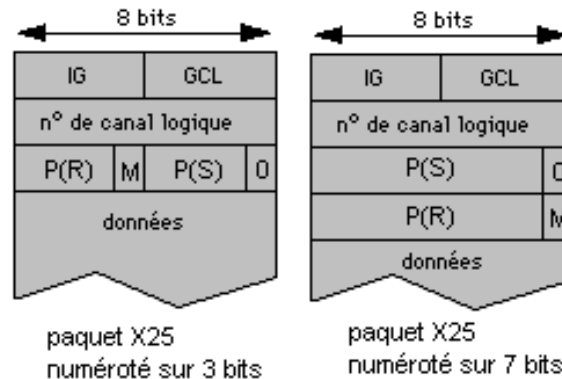


Le protocole HDLC, de la couche liaison, a été décrit au début de ce cours. Nous y renvoyons le lecteur.

Au niveau de la couche réseau, les paquets X25 sont constitués à partir de l'information provenant de la couche supérieure. Une en-tête de paquet est ajoutée. Le paquet X25 est transmis à la couche liaison et est transformé en une trame HDLC :



X25 utilise le concept de circuit virtuel : 16 groupes de 256 canaux logiques peuvent être multiplexés entre un ETTD et un ETCD. Le format général d'un paquet de données X25 est décrit ci-dessous :



- IG est l'identificateur général : il correspond généralement aux deux combinaisons suivantes :

0001 : les paquets sont numérotés de 0 à 7

0010 : les paquets sont numérotés de 0 à 127

- GCL est le numéro du groupe de canaux logiques ( de 0 à 15); le numéro de canal logique va de 0 à 255.
- P(S) et P(R) sont des numéros de séquence : P(S) est le numéro, de paquet, P(S) est le numéro du prochain paquet attendu.
- M (More Data) est un bit fixé à 1 si le paquet possède une suite (d'autres paquets appartenant au même message), sinon il est à 0.

En addition aux paquets de données, des paquets de contrôle permettent de véhiculer de l'information de service (on se limite ici à la numérotation des paquets modulo 8) :

paquet d'appel : il est caractérisé par le code 00001111 ; il comporte les adresses réseau de la source et de la destination (ces adresses de longueur variable implique l'indication de ces longueurs) ; des services complémentaires peuvent être définis ; l'utilisateur peut ajouter jusqu'à 64 octets d'informations supplémentaires.

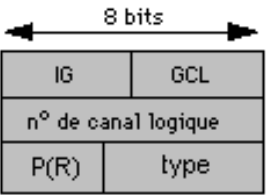


paquet de fermeture : il est caractérisé par le code 00010011 ; il comporte des indications analogues à celles du paquet d'ouverture ; il comporte aussi la raison de la fermeture du circuit virtuel.



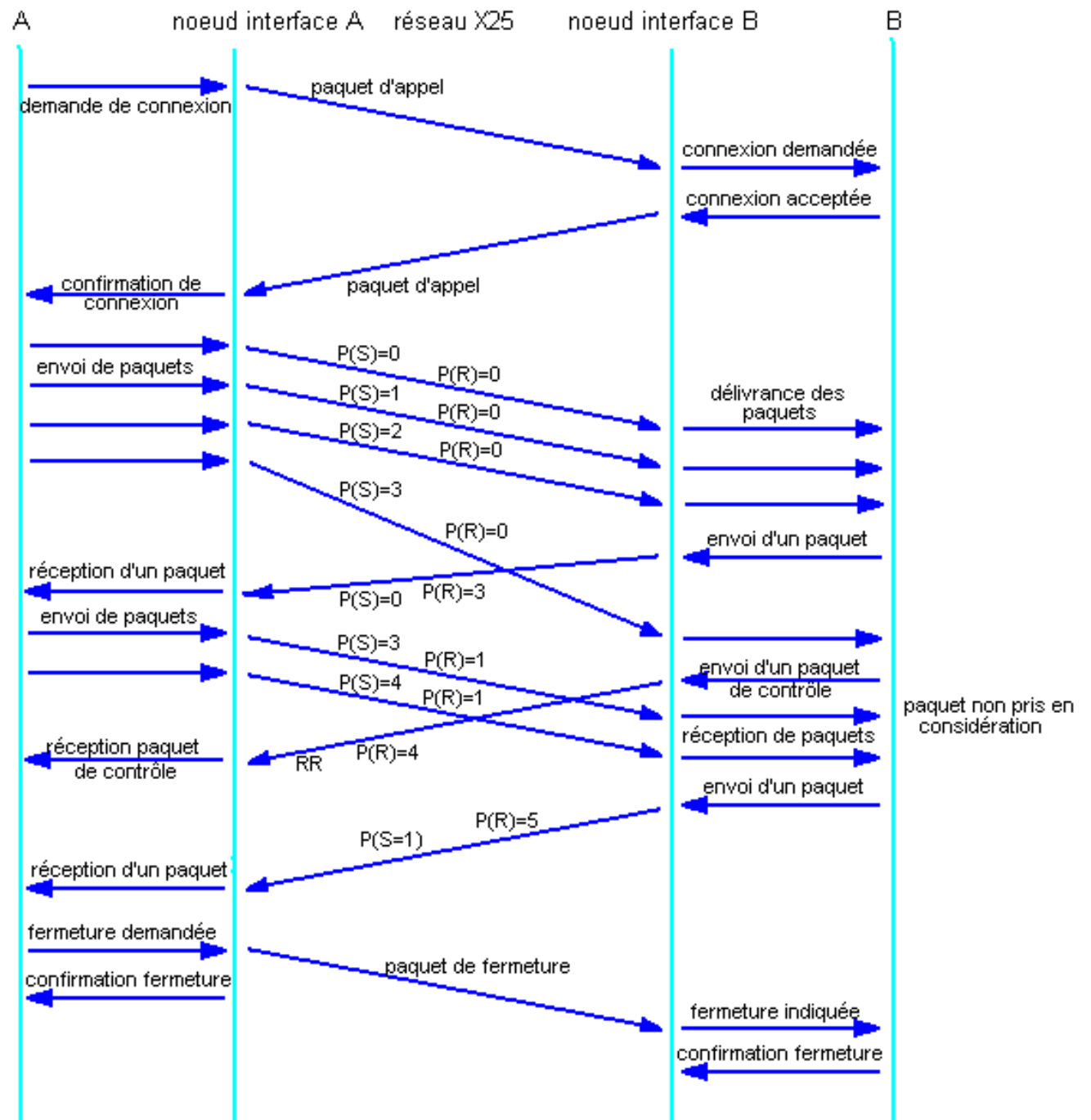
paquet de contrôle : 3 paquets de contrôle sont principalement utilisés :

- RR (Receive Ready) : acquittement des paquets de numéros antérieurs à P(R) ; type : 00001
- RNR (Receive Not Ready) : le récepteur ne peut recevoir de paquets ; il faut reprendre à partir du paquet de numéro P(R) ; type : 00101. Ce paquet est utilisé pour le contrôle de flux ;
- REJ (Reject) : le paquet reçu n'est pas accepté ; il faut reprendre à partir du paquet de numéro P(R) ; type : 01001. Ce paquet est utilisé pour le contrôle d'erreur.



La vie d'un circuit virtuel comporte trois phases : ouverture, transfert de données, fermeture . Dans la phase d'ouverture, un paquet spécial définit le chemin (circuit virtuel) que prendront les paquets suivants; à chaque noeud du réseau, il réserve les ressources nécessaires; la phase de transfert de données donne lieu à la circulation de paquets en séquence le long du circuit virtuel précédemment définit; enfin, un paquet de fermeture libère les ressources mobilisées. Le schéma ci-dessous décrit un échange typique de paquets X25 entre deux stations A et B :





On notera que l'acquittement peut s'effectuer soit via un paquet de données, soit par un paquet de contrôle si aucun paquet de données n'est à émettre. Gérard-Michel Cochard

Pour assurer la connexion de terminaux asynchrones (mode caractère), comme le Minitel par exemple, il est nécessaire de prévoir une interface dont le rôle est de convertir les caractères en paquets et réciproquement ; il s'agit d'un PAD (Paquet Assembleur Désassembleur).

## Exercices

[Exercice 1 \(questions a,b,c\)](#) ; [QCM1](#) ; [QCM2](#) ; [QCM3](#) ; [QCM4](#) ; [QCM5](#)

# Relais de trames

Sommaire :	<a href="#">Introduction</a>
	<a href="#">Architecture globale du Relais de Trames</a>
	<a href="#">Trame FR et Circuits virtuels</a>
	<a href="#">Contrôle de Congestion</a>

## Introduction

A l'origine conçu pour le RNIS (Réseau Numérique à Intégration de Services, en anglais ISDN), le relais de trame (en anglais Frame Relay ; on dit aussi, en français, "relayage de trames") est une amélioration notable de [X25](#) (X25 a donné lieu aux premiers réseaux numériques comme Transpac en France).

Le relais de trame ne concerne que les couches basses du modèle OSI : couche physique (1) et couche liaison (2) . Il est bien adapté aux classes de réseaux fiables (c'est à dire avec un taux d'erreur faible) car des "économies" sont faites sur le contrôle d'erreur. Les débits atteignent 2 Mbits/s et peuvent aller jusqu'à 45 Mbits/s (Transpac atteignait 48 000 bits/s).

Avant d'exposer en quoi consiste le relais de trame, il est bon de faire quelques rappels sur X25. La commutation de paquets X25 est basé sur trois avis : X25 paquet pour la couche réseau, [HDLC](#) pour la couche liaison et X11 pour la couche physique. Le multiplexage des paquets est effectué par la couche réseau.

Les deux couches, liaison (pour les trames) et réseau (pour les paquets) effectuent un contrôle d'erreur et un contrôle de flux ce qui génère un trafic important. Imaginons par exemple le transport d'un paquet unique depuis un ordinateur émetteur vers un ordinateur récepteur en supposant que le réseau utilisé est basé sur X25 et se compose de 5 tronçons (liaisons) et de 4 noeuds de communication. La paquet sera transmis au récepteur qui, après vérification du champ de contrôle d'erreur, renverra un acquittement (en supposant qu'il n'y ait pas d'erreur). Toutefois, le paquet est encapsulé dans des trames au niveau de chaque liaison, et pour chaque trame un contrôle d'erreur est effectué (on suppose qu'il n'y a qu'une seule trame) avec retour d'acquiescement. L'animation ci-dessous explicite le trafic généré par l'envoi de ce simple paquet et de son acquiescement.

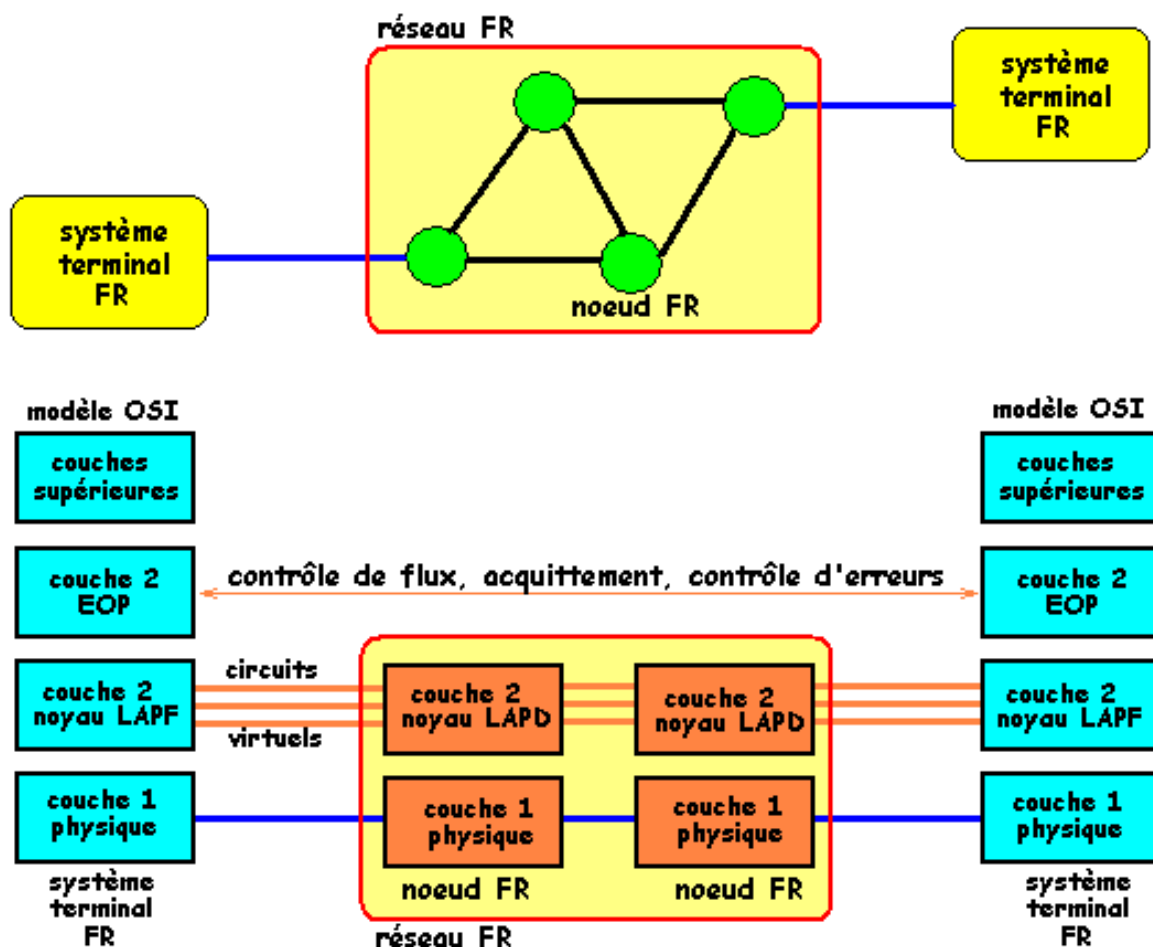
On constate que pour ce seul envoi, il est nécessaire de faire 20 transactions.

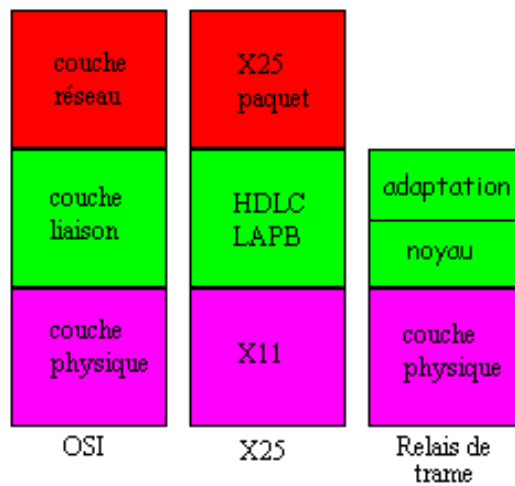
## Architecture globale du relais de trames

Dans le cas du relais de trames, la couche liaison est séparée en deux sous-couches :

- la sous-couche supérieure (couche d'adaptation ou LAPF control - LAPF = Link Access Procedure for Frame-mode bearer services, qui est une amélioration de LAPD) ne concerne que les éléments terminaux et non pas les noeuds du réseau : elle assure les fonctionnalités de contrôle de flux, de contrôle d'erreurs et d'acquittement. On l'appelle quelquefois la sous-couche EOP (Element Of Procedure).
- la sous-couche inférieure (appelée noyau ou LAPF core) a pour fonctionnalités essentielles la commutation des trames, la détection des erreurs (mais pas leur traitement qui est du ressort de la sous couche adaptation ou des couches supérieures), l'indication de congestion, le multiplexage des trames. Les trames non valides sont simplement éliminées.

La couche physique n'est concernée que par la signalisation relative au "drapeau" de trame. Rappelons qu'une trame HDLC est délimitée par des drapeaux dont le code est "01111110". La couche physique remplacera systématiquement toute suite de "11111" par "11110" pour éviter de confondre un morceau d'information avec un drapeau.



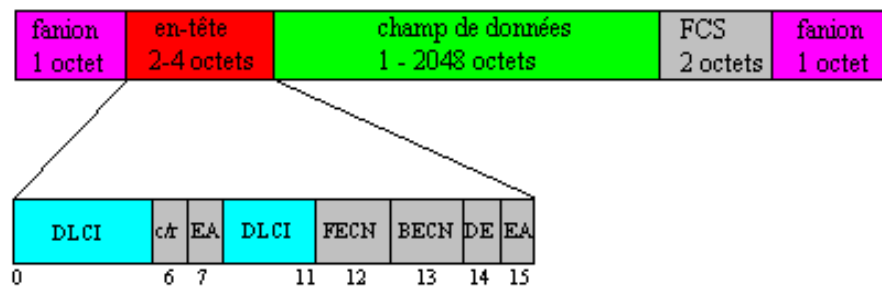


On peut constater sur l'animation ci-dessous que le transfert de données en Relais de Trames est plus simple que dans le cas de X25 (on se place dans les mêmes conditions que l'animation précédente) :

## Trame FR et Circuits virtuels

La trame employée est analogue à celle de [HDLC](#) avec cependant les différences suivantes :

- Il n'y a qu'un seul type de trame, la trame d'information ; il n'y a pas de trames de supervision, ni de trame non numérotées. La connexion et la déconnexion sont effectuées sur un canal spécial (le DLCI 0).
- Il n'est pas possible d'effectuer des contrôles de flux, ni des traitements d'erreurs : pas de champs de numéro de séquence.



Le fanion ou drapeau est la suite binaire "01111110".

Le champ en-tête peut prendre 2 (comme sur la figure ci-dessus) , 3 ou 4 octets ; il contient essentiellement le numéro de DLCI (Digital Link Connection Identifier), c'est à dire le numéro local du circuit virtuel emprunté (n'a de sens que sur une liaison). Les champs FECN, BECN et DE sont explicités plus loin.

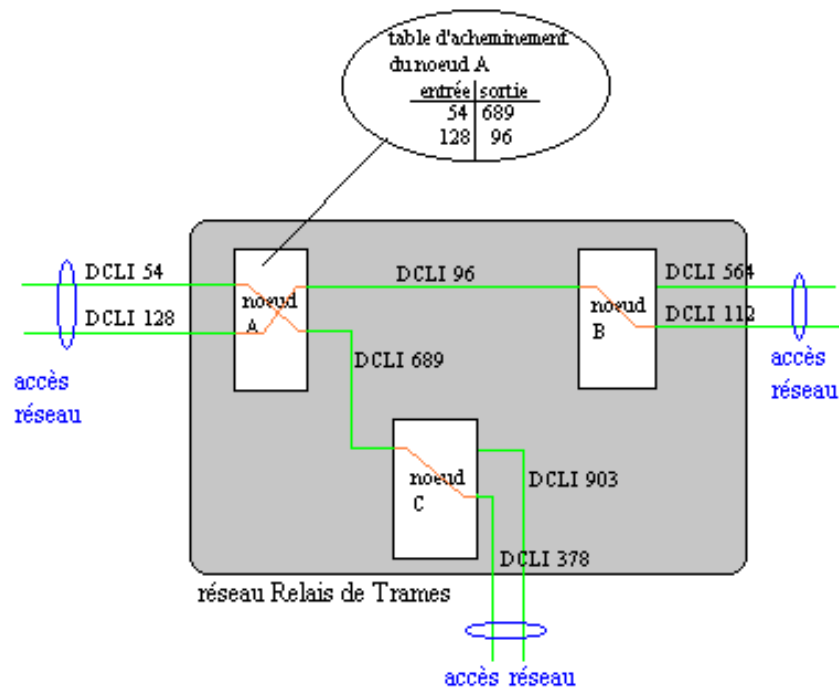
FCS (Frame Check Sequence) sert à la détection (uniquement) des erreurs. Si une erreur est détectée, la trame est éliminée par le noeud concerné.

Le champ de données est variable, sa taille maximale est définie par la taille maximale de la trame FR : 4 Ko.

DLCI tient sur deux blocs de bits, le premier de 6, le second de 4, ce qui fait 10 bits en tout ; les valeurs de DLCI peuvent donc aller théoriquement de 0 à 1023 ce qui correspond à des numéros de liaisons virtuelles. En fait, il existe des numéros réservés (par exemple, 0 est réservé à la demande de connexion, 1023 est réservé à la signalisation de congestion, ....) et il n'est possible d'utiliser que 992 numéros.

DLCI	Utilisation
0	signalisation (appel)
1 à 15	réservés
16 à 1007	DLCI pour utilisateurs
1008 à 1018	réservés
1019 à 1022	Multicast
1023	signalisation de la congestion

Au passage à un noeud de communication, une table d'acheminement établit la connexion entre la liaison virtuelle entrante et la liaison virtuelle sortante ; ces tables d'acheminement sont mises à jour à chaque demande de connexion.



## Le contrôle de congestion

Le suivi du trafic et le contrôle de congestion est basé sur l'autodiscipline des utilisateurs. Chaque utilisateur souscrit un contrat pour un débit donné, le CIR (Committed Information Rate) ; le débit effectif est mesuré pendant une certaine période T et le CIR correspond au nombre de bits Bc (Committed Burst Size):

$$\text{CIR} = \text{Bc}/T$$

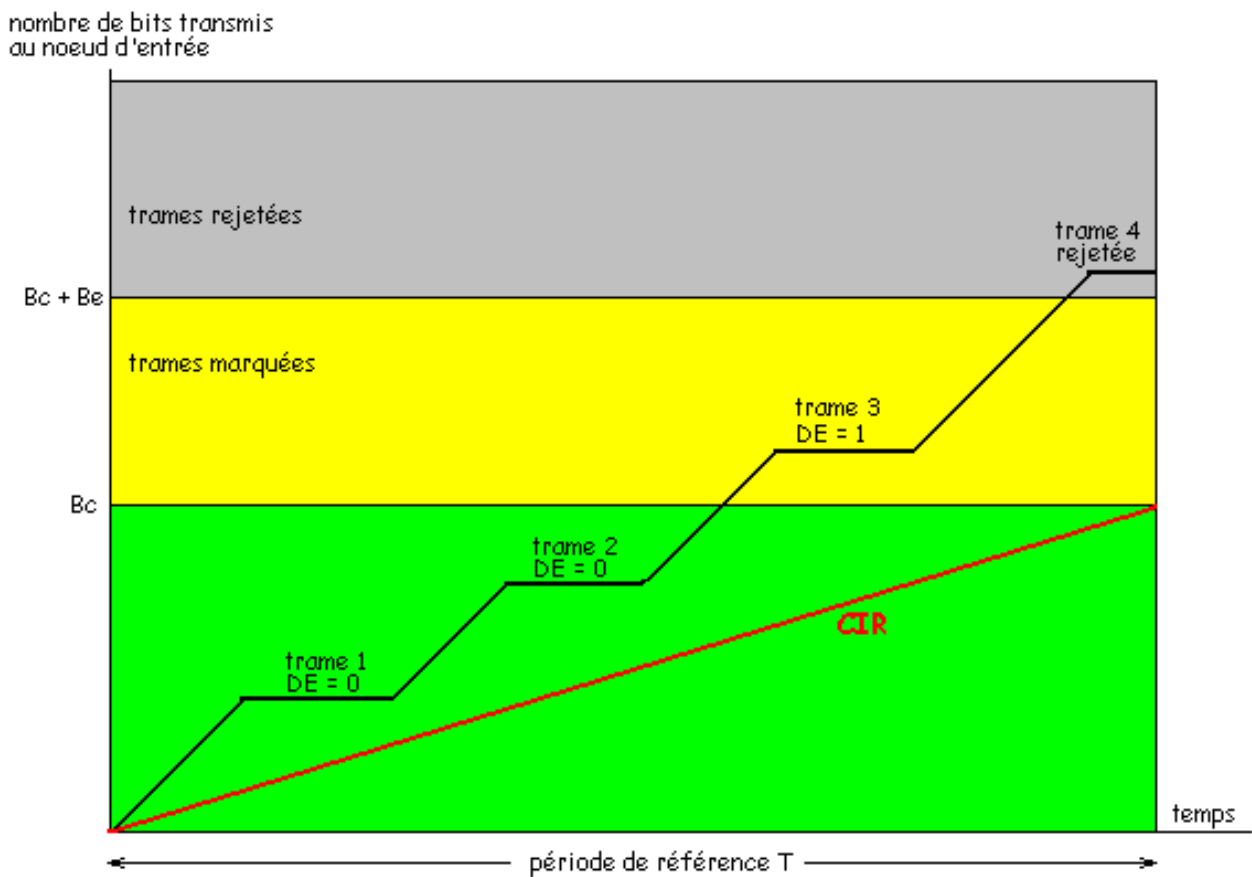
Il est toutefois permis aux utilisateurs de dépasser d'une certaine quantité ce débit, ou, pour raisonner sur la période T, la quantité Bc. On désigne par Be (Excess Burst Size) le surplus de bits permis ; le débit maximal autorisé est alors :

$$\text{Dmax} = (\text{Be} + \text{Bc})/T$$

Le suivi du trafic, pour un utilisateur donné, est effectué par le noeud d'entrée. Ce suivi utilise le bit DE (Discard Eligibility) de la trame. On mesure pendant la période T, le nombre cumulé de bits émis. Trois cas sont à considérer :

- $N < \text{Bc}$  : tout se passe bien et DE est mis à zéro.
- $\text{Bc} < N < \text{Bc} + \text{Be}$  : le débit souscrit est dépassé mais reste dans les limites permises. DE est mis à 1, mais la trame est néanmoins transmise ; cependant si on rencontre un noeud congestionné, la trame avec DE=1 est détruite.
- $N > \text{Bc} + \text{Be}$  : la trame est rejetée d'office.

La figure ci-dessous décrit l'envoi successif de trames d'un usager.



En cas de congestion, il est utile d'avertir les utilisateurs, ce qui peut se faire de trois manières différentes :

- mode FECN (Forward Explicit Congestion Notification) : un noeud qui est en congestion met à 1 le bit FECN des trames qui le traversent ce qui permet d'avertir le destinataire.
- mode BECN (Backward Explicit Congestion Notification) : ce mode est utilisé dans le cas d'une transmission bi-directionnelle (deux circuits virtuels sont alors utilisés) ; si une trame remonte vers l'amont, le noeud congestionné met le bit BECN des trames à 1 ce qui permet d'avertir l'émetteur qui devra alors réduire son débit.
- mode CLLM (Consolidated Link Layer Management) : dans la pratique, il n'y a pas symétrie dans une communication bi-directionnelle, le trafic de retour étant notablement plus faible que le trafic aller. De ce fait, la signalisation à l'émetteur d'une congestion est lente. Pour résoudre ce problème, CLLM prévoit une signalisation spéciale utilisant le canal DLCI 1023 prévu uniquement pour signaler les congestion par l'envoi d'un message spécial.

## Exercices

[Exercice 2](#) ; [QCM6](#) ; [QCM7](#) ; [QCM8](#)





# ATM

Sommaire :

[Généralités](#)

[La couche ATM](#)

[La couche AAL](#)

[Qualité de service et Contrôle de Congestion](#)

[Intégration de réseaux existants](#)

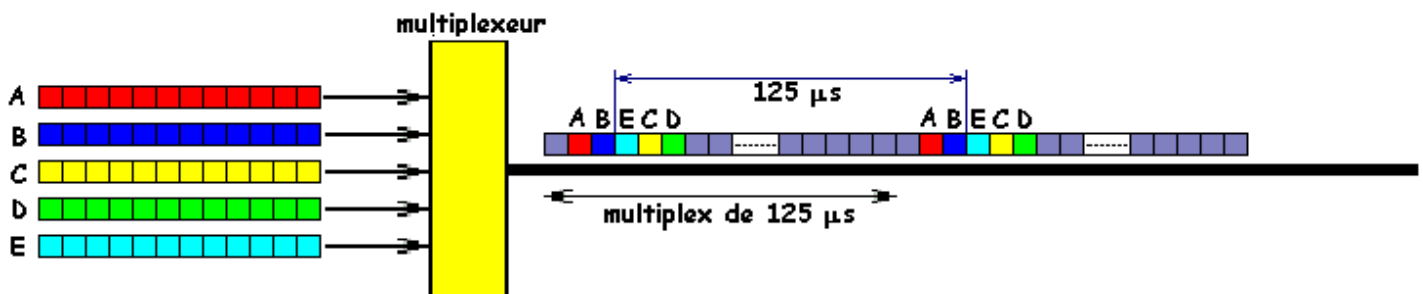
## Généralités

Pourquoi une nouvelle technologie ? Les besoins en hauts débits nécessités pour le transport de l'information multimédia nous l'impose :

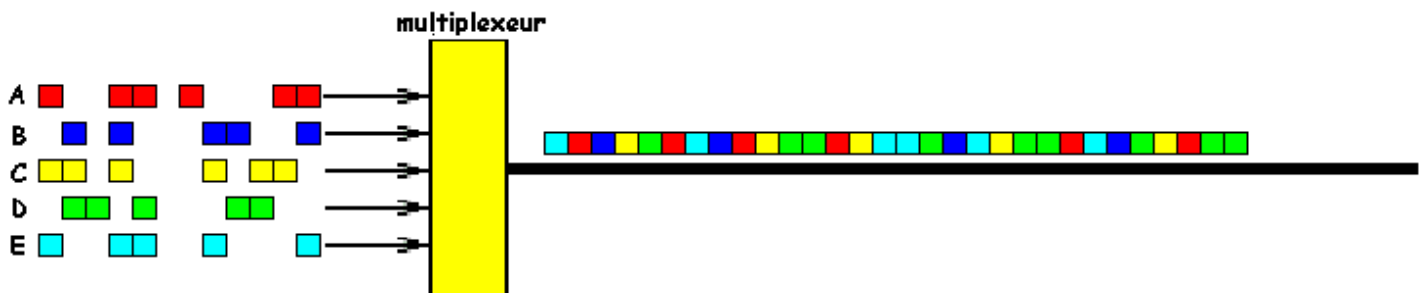
- vidéoconférence : de 128 Kbits/s à 5 Mbits/s
- TV : au moins 50 Mbits/s
- MPEG2 : 10 Mbits/s

Il s'agit donc de proposer un type de réseau permettant une large bande passante. Ce réseau est appelé RNIS-LB (Réseau Numérique à Intégration de Service Large Bande). On peut considérer ATM (Asynchronous Transfer Mode) comme un composant de ce réseau.

Quelques explications sur l'emploi du mot "asynchrone" sont nécessaires. Il est relatif au type de multiplexage employé. Le Mode Synchrone (STM) est bien connu pour le transport de la voix numérisée (à 64 Kbits/s). La trame MIC employée est un multiplex périodique divisé en intervalles de temps (32 en Europe) ; chaque intervalle de temps (IT) peut transporter un octet de voix numérisée. Un même IT revient donc périodiquement toutes les 125 microsecondes. Lors de l'établissement d'une liaison, un IT (toujours le même) est affectée à une voie entrante :



Le dispositif est simple mais peut être gaspilleur de bande passante. En effet, les IT non remplis sont "perdus". Le Mode Asynchrone (ATM) vise, au contraire, à utiliser au mieux la bande passante. L'information est structurée en blocs de longueur fixe appelées cellules ; les cellules peuvent arriver de manière irrégulière sur les voies entrantes. Elles sont placées les unes derrière les autres sur la voie multiplexée :



Chaque cellule comporte une en-tête indiquant sa destination.

La petitesse et la taille fixe des cellules ATM permettent un multiplexage performant et l'utilisation de dispositifs hardware plutôt que software, d'où un gain de rapidité permettant d'atteindre les débits ci-dessus.

Les cellules ATM sont de 53 octets comprenant une en-tête de 5 octets et une partie "données" de 48 octets. Ce nombre bizarre provient d'un compromis entre Américains (souhaitant 64 octets) et Européens (souhaitant 32 octets).

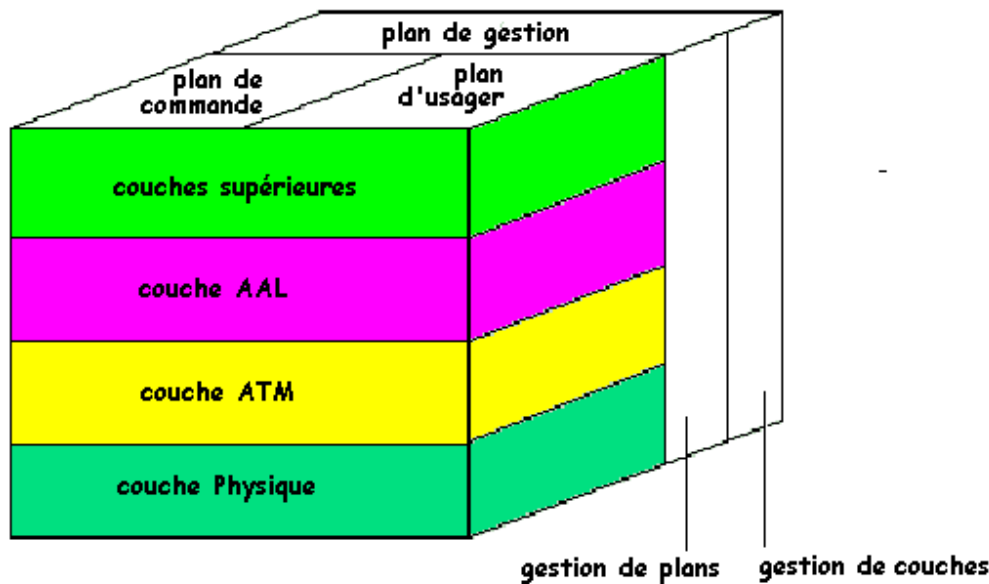
La technologie ATM est basée sur une bonne qualité des supports de transmission (fibre optique) et des équipements permettant d'éviter

- le contrôle de flux (après acceptation de la communication)
- le contrôle d'erreur

On retrouve ici quelques caractéristiques du relais de trames.

ATM est orienté connexion : établissement d'une voie de communication basée sur la notion de circuits virtuels. Une amélioration de cette technique, notamment pour le routage, repose sur la notion de groupement de circuits virtuels. Les chemins virtuels (VCC = Virtual Channel Connection) font partie de faisceaux appelés conduits virtuels (VPC = Virtual Path Connection) qu'il est possible de router globalement.

Du point de vue du modèle en couche, ATM se définit suivant le schéma ci-dessous :



Comme usuellement, la couche Physique traite des supports de transmission et de l'encodage des données et ne fait pas partie, stricto sensu, de la technologie ATM

ATM se réfère aux couches AAL et ATM

AAL (ATM Adaptation Layer) est une couche d'adaptation aux protocoles des couches supérieures tandis que ATM est une couche définissant la transmission des données dans les cellules et l'utilisation des connexions logiques.

En outre ATM distingue 3 plans :

- le plan utilisateur, relatif aux couches supérieures, qui traitera les questions de contrôle de flux et d'erreur
- le plan de commande relatif aux fonctions de connexion
- le plan de gestion comportant deux composants :
  - la gestion des plans (coordination entre les plans)
  - la gestion de couches (fonctions de gestion des ressources et des paramètres des protocoles)

La correspondance avec le modèle "standard" OSI n'est pas évidente. une tentative de correspondance est donnée ci-dessous :

modèle OSI	modèle ATM		fonctionnalités
couche transport	couche AAL	CS	interfaçage universel
		SAR	segmentation et assemblage
couche réseau	couche ATM		contrôle de flux ; gestion des circuits virtuels ; traitement des en-têtes de cellules ; multiplexage des cellules
couche liaison	couche physique	TC	génération des cellules ; génération des trames
couche physique		PMD	accès physique au réseau

CS : Convergence Sublayer  
 SAR : Segmentation And Reassembly  
 TC : Transmission Convergence  
 PMD : Physical Medium

Depuis les premiers travaux su ATM dans les années 80, un effort de normalisation a été effectué. Toutes les recommandations sont rassemblées dans la série I de l'UIT. Un groupe particulièrement actif, l'ATM Forum, rassemble près de 500 membres qui proposent des spécifications.

## La couche ATM

La couche ATM est chargée des 3 fonctions principales suivantes :

- commutation des cellules
- multiplexage des cellules
- génération/extraction de l'en-tête des cellules

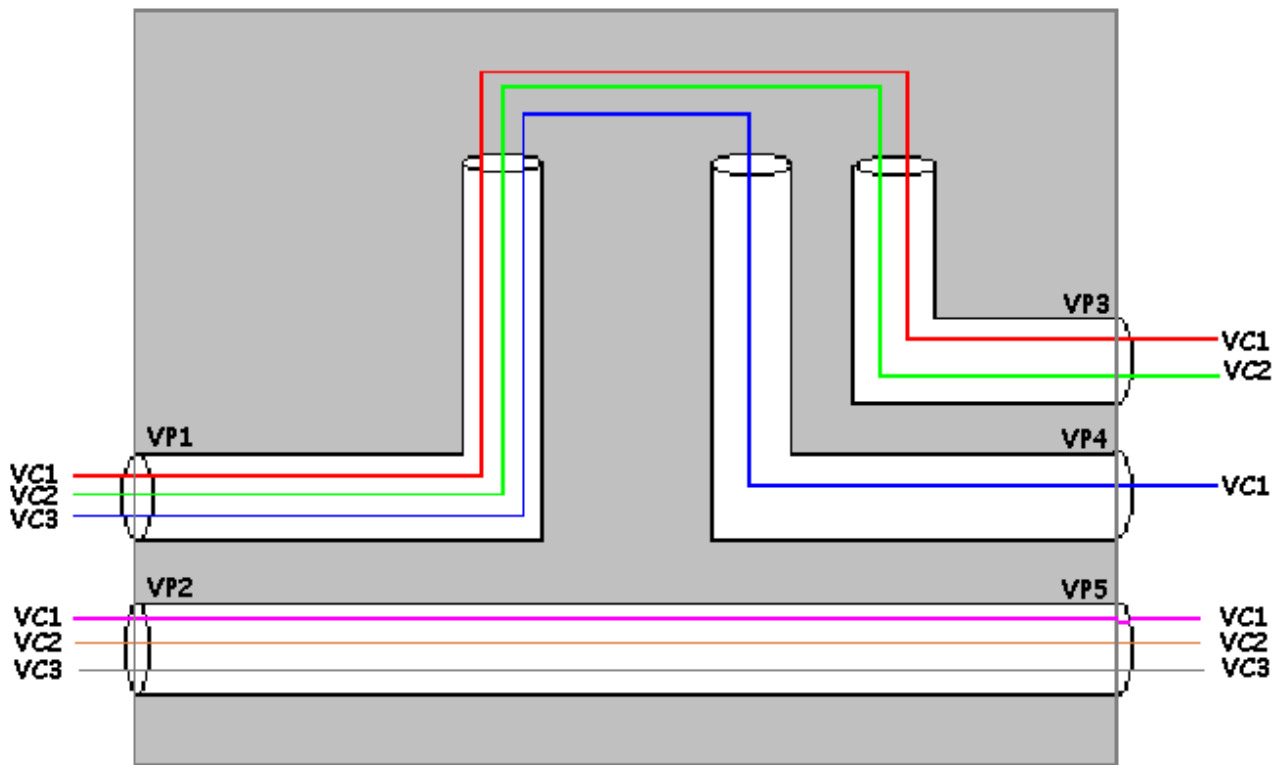
### A- Circuits virtuels et conduits virtuels

Un conduit virtuel (VPC = Virtual Path Connection) est un faisceau de circuits virtuels (VCC = Virtual Channel Connection). Par rapport au relais de trames, c'est ici une nouveauté qui permet

- de faciliter l'acheminement, celui-ci étant fait principalement pour les conduits virtuels
- de constituer des réseaux "privés" basés sur les VPC.

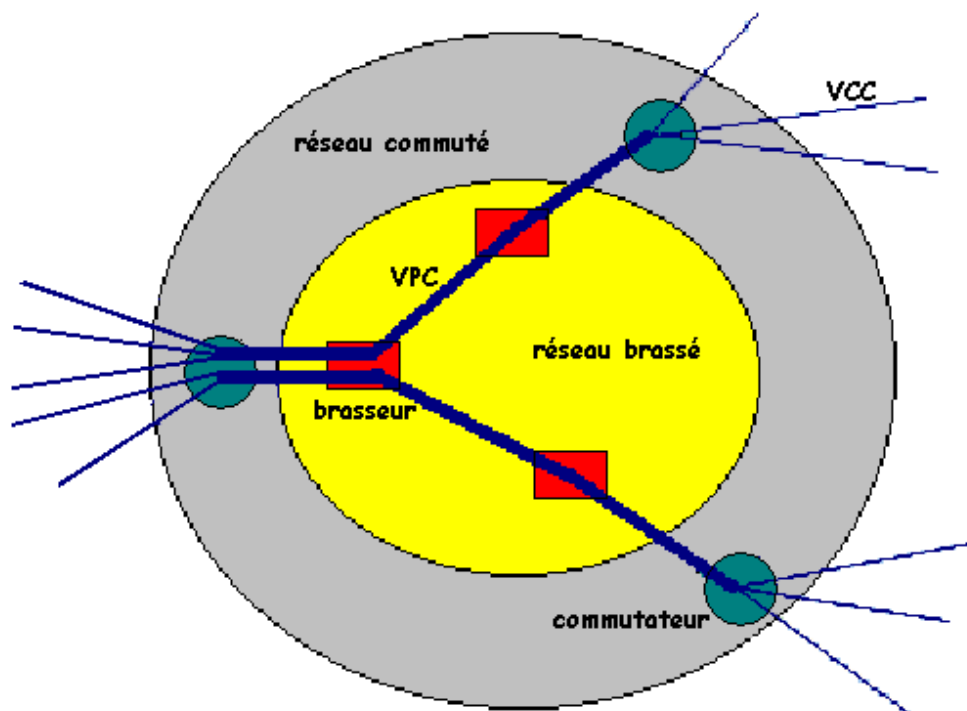


Chaque VPC et chaque VCC possède un numéro (VPI et VCI) :



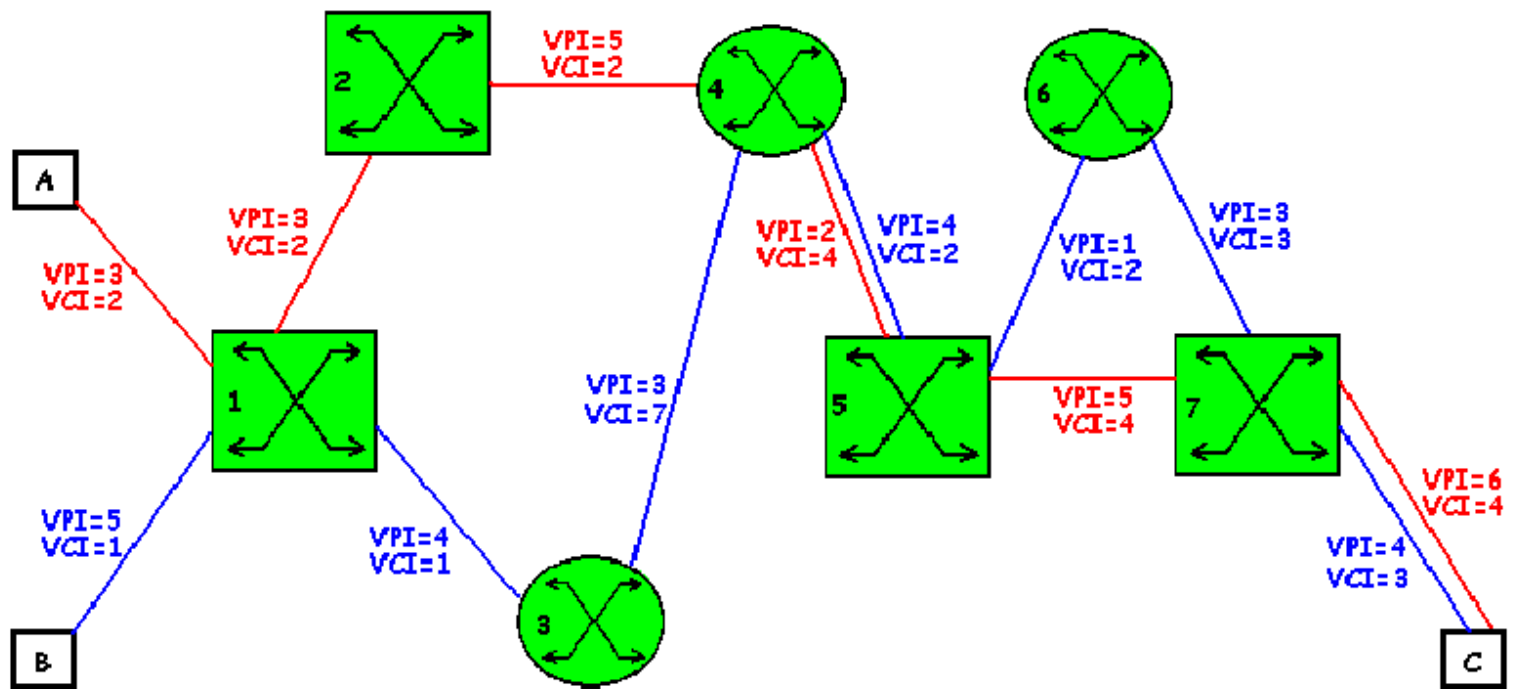
Sur la figure ci-dessus, on constate que le conduit VP2 est commuté au conduit VP5. Les circuits VC1, VC2, VC3 qu'ils contiennent sont donc automatiquement commutés. Ce type de commutation est effectué par un **brasseur**. La table de commutation est réduite puisqu'elle ne contient que les numéros de VPI.

Par contre les circuits VC1, VC2, VC3 du conduit VP1 sont commutés vers respectivement VP3/VC1, VP3/VC2, VP4/VC1. Les tables de commutation doivent donc comporter les deux numéros (conduit et circuit).



D'une manière usuelle, le coeur d'un réseau ATM ne contient que des noeuds de commutation de type brasseur : ce sont des faisceaux de circuits virtuels qui sont commutés. Cette situation simplifie considérablement la commutation de circuit (tables moins volumineuses, donc temps de commutation plus court). Cette partie du réseau est appelée **réseau brassé**. La commutation de circuits virtuels n'a lieu généralement que sur le réseau périphérique (réseau commuté) ; les tables de commutation sont ici plus complexes.

**exemple :**



où les rectangles figurent des brassards et les disques des commutateurs simples. Les tables de routage sont :

noeud 1		noeud 2		noeud 3		noeud 4		noeud 5		noeud 6		noeud 7	
in	out	in	out	in	out	in	out	in	out	in	out	in	out
3	3	3	5	4,1	3,7	5,2	2,4	2	5	1,2	3,3	5	6
5	4					3,7	4,2	4	1			3	4

Dans ATM, (comme dans X25), il faut construire de bout en bout des routes pour l'acheminement des cellules avant d'effectuer le transport d'informations. Il y a 4 types de flux de "signalisation" à considérer :

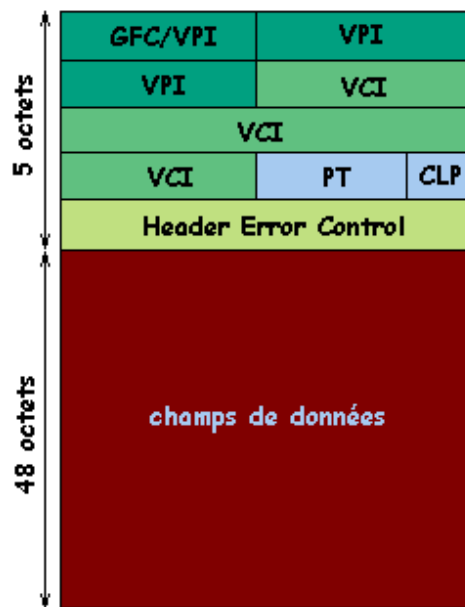
- Il peut s'agir de VCC semi-permanents, c'est à dire établis, une fois pour toutes, de manière durable, comme dans le cas de liaisons spécialisées. Dans ce cas aucun mécanisme n'est à prévoir (pas de flux de signalisation)
- un VCC (permanent) est prévu pour effectuer la réservation de VCC permettant le transport des signaux nécessaires à l'établissement et la libération de voies pour les informations à transmettre. Il porte le numéro VPI=x, VCI=1 (x=0 pour la liaison utilisateur-commutateur local) et la méthode s'appelle "méta-signalisation" puisque le canal correspondant sert à définir les canaux de signalisation qui serviront à l'établissement des liaisons utiles.
- un VCC doit être réservé pour la signalisation d'établissement/libération point à point. Il porte le numéro VPI=x, VCI=5 (x=0 pour la liaison utilisateur-commutateur local).
- il y aussi des canaux de signalisation de diffusion. Ces canaux portent le numéro VPI=x, VCI=2 (x=0 pour la liaison utilisateur-commutateur local).

## B - La cellule ATM

Comme déjà signalé, la cellule ATM comporte 53 octets : 5 octets d'en-tête et 48 octets de données :

L'en-tête comprend les champs suivants :

- identificateur VPC : numéro VPI de conduit virtuel. S'il s'agit d'une cellule reliant deux commutateurs ATM (interface NNI : Network-Network Interface), le champ est de 12 bits (il peut donc y avoir 4096 conduit virtuel distincts). S'il s'agit d'une cellule reliant l'utilisateur au réseau ATM (interface UNI : User-Network Interface), les



quatre premiers bits sont réservés au champ GFC (Generic Flow Control) qui, comme son nom l'indique, est utilisé pour le contrôle de flux et le numéro VPI prend les 8 bits suivants (dans ce cas il y a au maximum 256 conduits virtuels).

- identificateur VCC (16 bits) : numéro VCI de circuit virtuel (il peut donc y en avoir 65536).
- PT : Payload Type (3 bits) : ce code indique le type d'information transportée par la cellule (voir ci-dessous)
- CLP (Cell-Loss Priority) : bit utilisé en contrôle de congestion (voir plus loin).
- HEC (Header Error Control) (8 bits) : est un champ correcteur/détecteur d'erreur calculé à partir du polynôme générateur  $z^8 + z^2 + z + 1$  ; la détection/correction d'erreur ne porte que sur les 32 premiers bits de l'en-tête. La correction ne porte que sur les erreurs simples.

La taille de la cellule est choisie petite pour deux raisons principales :

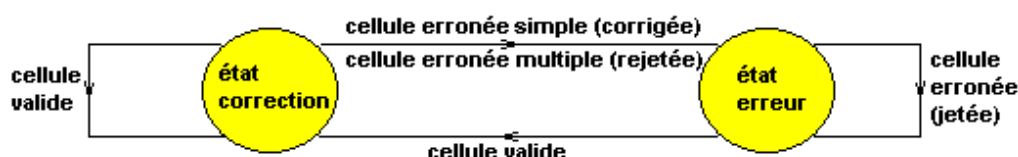
- d'une part il ne faut pas oublier que les applications téléphoniques sont prépondérantes et si une cellule doit transporter du son numérisé, il ne faut pas que la numérisation porte sur de gros volumes qu'il faudrait ensuite transporter ; au contraire, une numérisation d'un petit volume suivi de son transport immédiat est souhaitable. La voix étant numérisée à raison de 8000 échantillons par seconde et chaque échantillon étant codé sur 1 octet, il s'ensuit que la numérisation de 8 échantillons correspond à 1 ms ce qui signifie qu'une cellule ATM transporte 6 ms de son ( $6 \times 8 = 48$ ). La numérisation, le temps de transmission et la restitution du son correspondent à des délais acceptables.
- d'autre part, pour des applications temps réel (comme la voix téléphonique), il ne faut pas que le multiplexage soit pénalisant et donc il est nécessaire de faire en sorte que l'attente d'envoi à chaque multiplexeur soit la plus courte possible ce qui est possible avec des cellules de petites tailles.

Le champ PT indique sur 3 bits le type de cellules, en particulier s'il s'agit d'une cellule utilisateur ou d'une cellule de gestion :

champ PT	type de cellule
000	cellule utilisateur, pas de congestion
001	cellule utilisateur, pas de congestion
010	cellule utilisateur, congestion
011	cellule utilisateur, congestion
100	cellule de gestion F5-OAM
101	cellule de gestion F5-OAM
110	cellule FRM
111	réservé

Le bit CLP affecte une priorité qui peut être utilisée en cas de congestion : les cellules avec CLP=1 peuvent être rejetées par un commutateur engorgé tandis que les cellules avec CLP=0 doivent être transmises.

Il est à noter que le contrôle d'erreur (sur l'en-tête) ne s'effectue pas au niveau de la couche ATM mais de la couche physique. Il s'effectue de la manière suivante, sachant que deux états sont à considérer : l'état "correction" et l'état "erreur". Rappelons que HEC ne permet que la correction des erreurs simples. Si la cellule arrivant est dans l'état "correction" et si la cellule est correcte, alors la cellule est transmise à la couche ATM et l'état reste "correction" ; si au contraire, la cellule est entachée d'une erreur simple, la cellule est corrigée et transmise à la couche ATM mais on passe à l'état "erreur" ; si une nouvelle cellule arrive avec une erreur, elle est rejetée. Supposons que l'état soit "correction" et qu'une cellule arrive avec une erreur multiple : la cellule est rejetée et l'état devient "erreur". En résumé, dans l'état "erreur", une nouvelle erreur provoque le rejet de la cellule, sinon, si la cellule est correcte, l'état redevient "correction".



Dans le cas d'une cellule passée par la couche ATM à la couche physique, le HEC est calculé. On notera que le calcul doit être refait au passage à chaque commutateur puisque les numéros VPI,VCI changent.

## La couche AAL

Selon la recommandation I362, la couche AAL (ATM Adaptation Layer) possède les fonctionnalités suivantes :

- gestion des erreurs de transmission
- assemblage/désassemblage de l'information en cellules
- gestion des cellules perdues
- contrôle de flux et de synchronisation

### A - Classes d'applications

4 classes d'applications ont été définies par l'UIT :

- classe A : vidéo à débit constant, transport de la voix
- classe B : vidéo et audio à débit variable
- classe C : transfert de données en mode connecté
- classe D : transfert de données en mode non connecté

Ces quatre classes sont caractérisées par les paramètres suivants : la synchronisation entre l'émetteur et le récepteur, le mode temps réel ou temps différé, la constance ou la variabilité du débit, le mode connecté ou non connecté :

classe	A	B	C	D
synchronisation	oui		non	
temps	réel		différé	
débit	constant	variable		
mode	connecté			non connecté

Au départ, 4 protocoles AAL1, AAL2, AAL3, AAL4 avaient été prévus pour chacune des classes. Aujourd'hui, la situation est un peu plus confuse et en fait, AAL3 et AAL4 ont fusionné en AAL3/4 tandis qu'un cinquième protocole a fait son apparition, AAL5. La correspondance entre les classes d'applications et les protocoles est maintenant plutôt celle-ci :

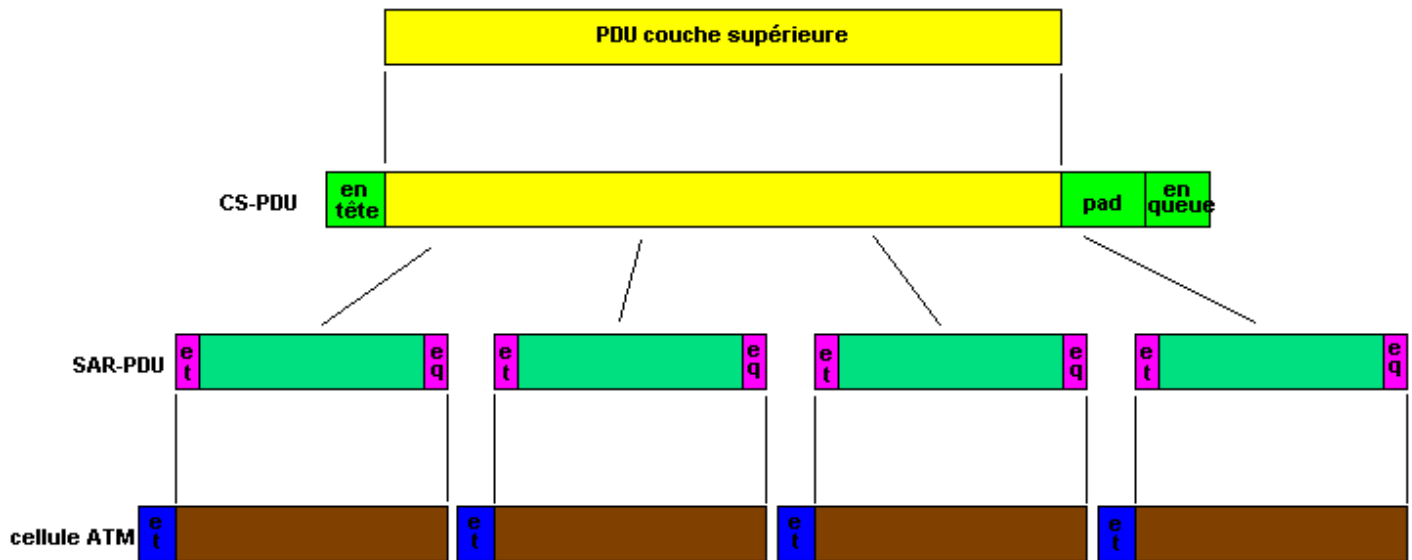
classe	A	B	C	D
protocole	AAL1	AAL2	AAL3/4, AAL5	AAL5

### B - CS-PDU, SAR-PDU et cellule ATM

Rappelons que la couche AAL est décomposée en deux sous-couches :

- CS (Convergence Sublayer) : orientée service, ses fonctions dépendent du type de service traité (voix, vidéo,...) ; au niveau de cette sous-couche, l'information provenant des couches supérieures est encapsulée dans des CS-PDU (Protocol Data Unit).
- SAR (Segmentation And Reassembly Sublayer) : fabrication des cellules (découpage des blocs CS-PDU en SAR-PDU, puis en cellules ATM) ou reconstitution des données.





## C - Protocoles AAL

### Protocole AAL1

Il est adapté à la classe A : temps réel, débit constant, synchronisation, mode connecté ce qui est le cas, par exemple de la voix ou de la vidéo sans compression. Les pertes de données sont signalées sans plus.



Alors que la CS-PDU ne possède pas de protocole particulier, la SAR-PDU possède une en-tête. Les SAR-PDU "P" ont le premier bit à 1 et possèdent un champ pointeur qui indique la position du message suivant. Les SAR-PDU "non P" ont le premier bit à 0 et ne possèdent pas de champ pointeur. Le champ SN correspond à un numéro de séquence (ce qui permet de détecter les cellules manquantes) ; SNP est un champ de contrôle sur le numéro de séquence (auto-correction des erreurs simples). P est un bit de parité (encore un contrôle d'erreur sur le numéro de séquence). La longueur totale de la SAR-PDU est de 48 octets.

### Protocole AAL2

Ce protocole est adapté aux flux audio et vidéo avec compression et débit variable. La CS-PDU n'a pas de structure particulière. La SAR-PDU possède une en-tête et une en-queue.



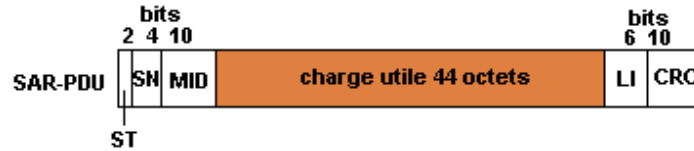
Comme précédemment, SN est le numéro de séquence ; le champ IT indique si l'on est au début, au milieu ou en fin de message ; LI indique la taille de la charge utile (inférieure ou égale à 45 octets) ; CRC est un champ détecteur d'erreur sur l'ensemble de la SAR-PDU.

### Protocole AAL3/4

Ce protocole correspond à un transport de données sous forme de messages qui peuvent être multiplexés sur le même circuit virtuel. La couche CS possède un protocole propre :



Le CPI indique le type de message ; Btag et Etag signalent le début et la fin d'un message ; Length et B size indiquent la taille de la charge utile et du tampon à attribuer au message. Des octets de bourrage sont prévus pour faire en sorte que le nombre d'octets au total soit un multiple de 4.



Le champ ST indique si on est au début (10), à la fin (01), au milieu (00) d'un message ou si le message tient dans une seule cellule (11). Le champ SN est un numéro de séquence. Le champ MID indique à quelle session se rapporte le message (multiplexage) ; LI donne la taille de la charge utile (inférieur ou égal à 44 octets) ; CRC est un champ détecteur d'erreurs.

## Protocole AAL5

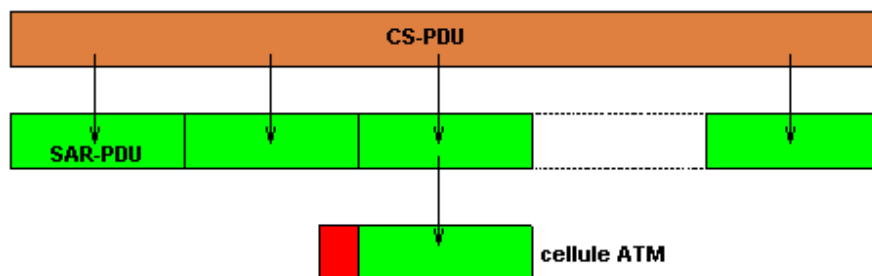
Le protocole AAL3/4 étant relativement compliqué, une proposition de simplification fut faite sous le nom de SEAL (Simple Efficient Adaptation Layer !) qui devint AAL5.

La couche CS possède un protocole visant à ajouter un en-queue à un message. La longueur totale doit être un multiple de 48 octets (on est amené à rajouter dans la charge utile des caractères de bourrage).



UU n'est pas utilisé, Length indique la longueur de la charge utile (caractères de bourrage non compris) ; CRC est un champ détecteur d'erreur.

Le passage à la couche SAR s'effectue de manière très simple : CS-PDU est découpée en blocs de 48 octets qui constituent les SAR-PDU.



## Qualité de service et contrôle de congestion

Un réseau comme ATM se doit de rendre le "service" demandé. Par exemple, si le réseau délivre du temps réel, il ne faut pas qu'il y ait de retard qui impliquerait la perte de cellules. Par suite, avant d'établir une connexion sur un circuit virtuel, une négociation de contrat doit avoir lieu pour préciser les conditions du trafic. Ce contrat peut spécifier des critères différents suivant le sens de transmission. La négociation

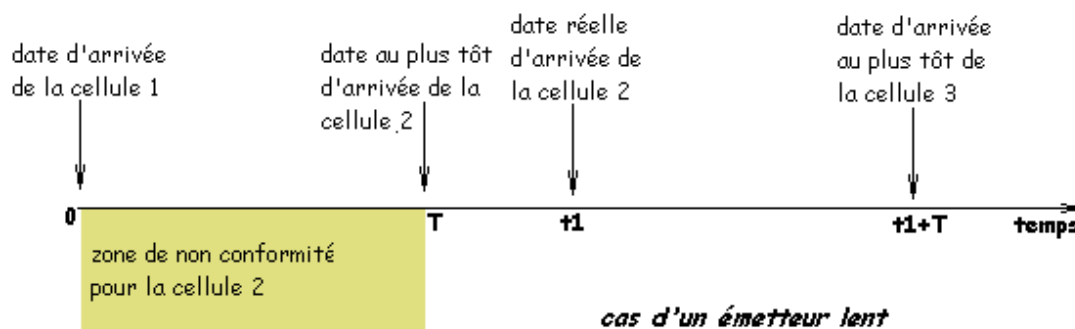
Gérard-Michel Cochar

porte sur un certain nombre de paramètres définissant la qualité de service :

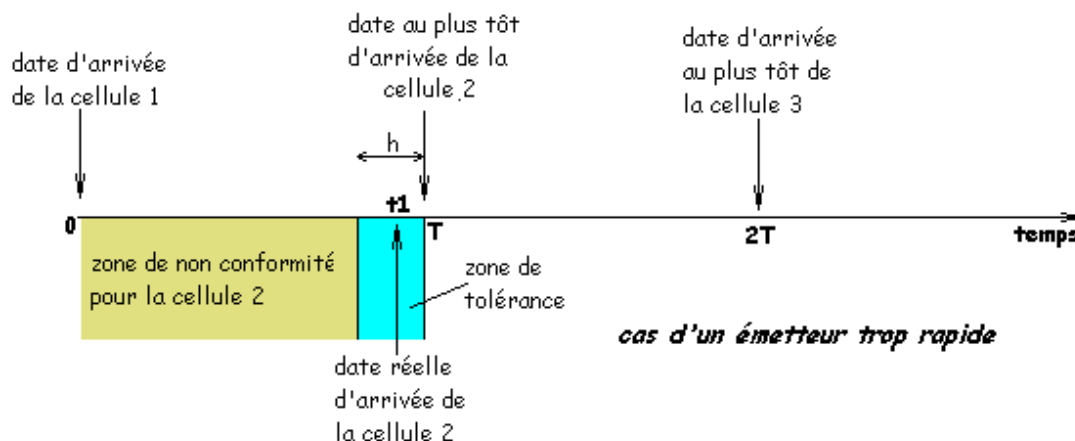
paramètres	définition
PCR (Peak Cell Rate)	valeur maximum du débit
SCR (Sustained Cell Rate)	valeur moyenne du débit envisagé
MCR (Minimum Cell Rate)	débit minimum vital
CVDT (Cell Variation Delay Tolerance)	variation maximale du délai de transmission des cellules
CLR (Cell Loss Ratio)	taux de perte des cellules
CTD (Cell Transfer Delay)	temps moyen d'acheminement d'une cellule
CDV (Cell Delay Variation)	variation admissible du temps d'acheminement des cellules
CER (Cell Error Ratio)	taux de cellules erronées
SECBR (Severely Errored Cell Block Ratio)	taux de blocs de cellules contenant des erreurs
CMR (Cell Misinsertion Rate)	Taux de cellules mal adressées

Un des mécanismes de contrôle de la qualité de service est basé sur l'algorithme Generic Cell Rate Algorithm (GCRA), qui est une version de l'algorithme bien connu, dit du "seau percé". Cet algorithme est assez proche de celui utilisé dans le relais de trames.

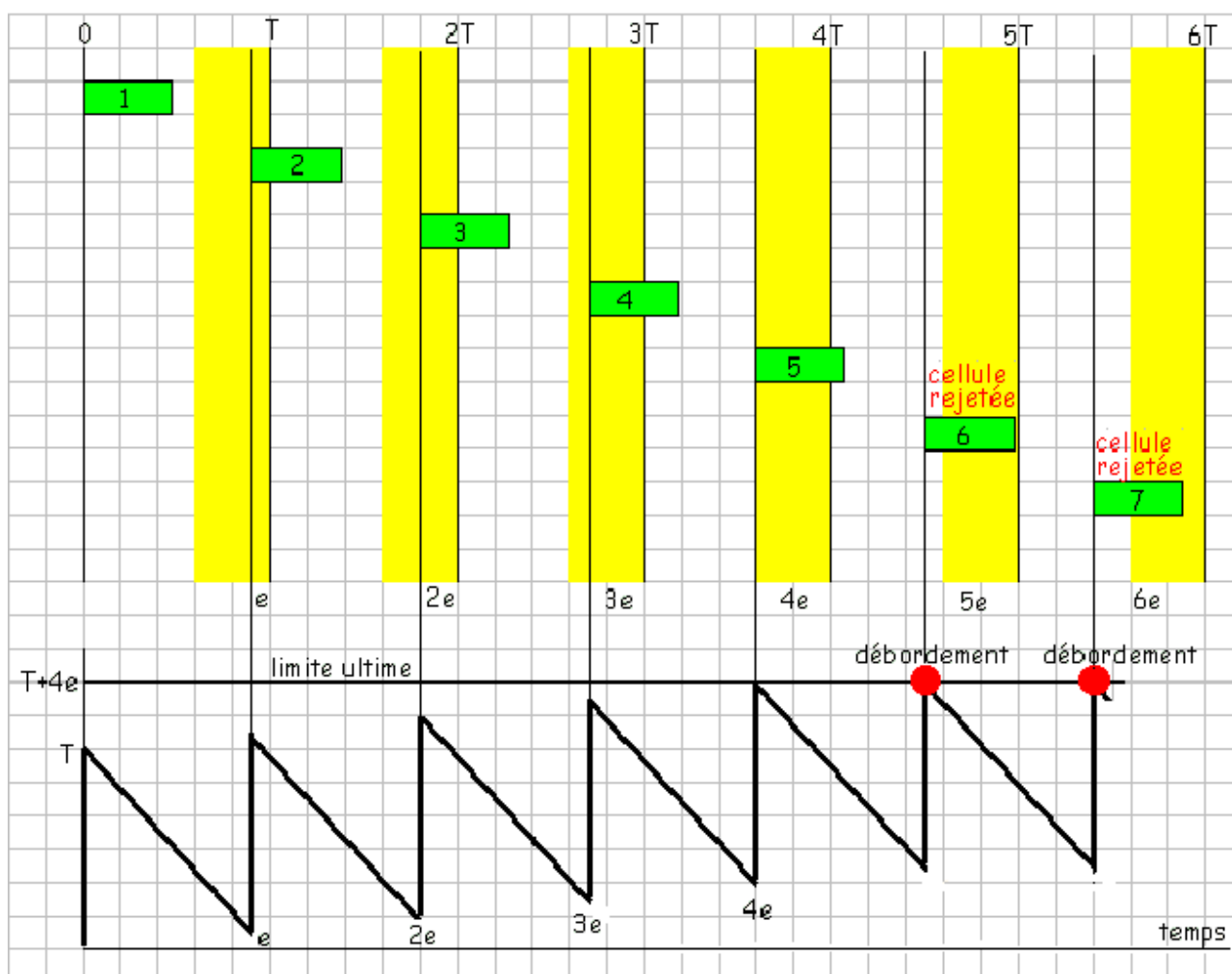
Imaginons que le contrat porte sur un débit  $D_0 = \text{PCR} = 125\,000$  cellules/s. Le temps écoulé entre deux envois successifs de cellules est donc  $T = 1/D_0$ . Une cellule ne doit donc pas arriver en un temps plus court que  $T$  après la réception de la cellule précédente. Elle peut par contre arriver dans un intervalle de temps plus grand que  $T$ . Une cellule qui se conforme à cette règle est "conforme". Si la cellule 1 arrive au temps 0 et si la cellule 2 arrive au temps  $t_1 > T$ , la cellule 3 devra arriver au plus tôt au temps  $t_1 + T$ .



Bien entendu, il se pose un problème lorsque l'émetteur "triche" en accentuant sa cadence : les cellules deviennent alors non conformes au contrat. On peut toutefois accepter une tolérance de  $h$  microsecondes (qui correspond au paramètre CVDT). On pourra considérer que la cellule est encore conforme si elle arrive au temps  $T-h$  après la cellule précédente. Mais il faudra que la cellule suivante arrive au plus tôt au temps  $2T$ .



Imaginons maintenant que l'émetteur trop rapide conserve sa cadence d'envoi. Les cellules suivantes vont s'enfoncer dans la zone de tolérance et au bout d'un moment avoir une date d'arrivée dans la zone stricte de non conformité. La cellule 2 va arriver trop tôt, la cellule 3 va arriver 2e trop tôt, la cellule 4 va arriver 3e trop tôt, etc... Le schéma ci-dessous montre une telle situation avec  $h = 4e$ . Il est clair que la cellule 6 est en totale "effraction". Elle pourra être détruite par les composants réseau.



Le bas du schéma précédent explique pourquoi l'algorithme est qualifié de "seau percé". La cellule est considérée comme un récipient apportant une quantité T litres de liquide. A l'arrivée, ce liquide est intégralement versé dans un seau de capacité T+h litres qui possède un trou et fuit à la cadence de T litres/seconde. Si les cellules respectent le contrat, elles trouveront toujours le seau vide. Si au contraire, elles arrivent en avance, elles trouveront un seau non vide ; le niveau de liquide du seau s'élèvera donc à chaque arrivée d'une cellule jusqu'au moment où le seau débordera.

On voit que temporairement, on peut dépasser le débit maximum, mais que seulement N cellules seront acceptées. Le calcul de N est assez simple si l'on se base sur le tableau ci-dessous :

cellule	date d'arrivée au plus tôt	date réelle d'arrivée
1	0	0
2	T	T-e
3	2T	2(T-e)
4	3T	3(T-e)
---	---	---
N	(N-1)T	(N-1)(T-e)

Pour la dernière cellule N, on aura atteint la limite de tolérance, ce qui correspond à la date (N-1)T-h, donc  $(N-1)(T-e) = (N-1)T - h$ , d'où la valeur de N :

$$N = 1 + \frac{h}{e}$$

e mesure l'intervalle de temps d'avance à chaque arrivée de cellules. Si le débit souscrit est  $D_0$  et si le débit réel est  $D > D_0$  (les débits étant mesurés en cellules par seconde), le temps d'arrivée au plus tôt de la cellule 2 est  $T = 1/D_0$  et son temps d'arrivée réel est  $T' = 1/D$ , d'où

$$e = T - T' = \frac{D - D_0}{D_0}$$

**exemple** : Supposons que le débit souscrit est  $D_0 = 125\,000$  cellules/s d'où  $T = 8\,\mu\text{s}$ . Imaginons que le récepteur envoie des cellules au débit  $D = 200\,000$  cellules/s. Sachant que la tolérance est  $h = 24\,\mu\text{s}$ , combien de cellules pourront être acceptées ?

La valeur de  $e$  est  $T - T' = 8 - 5 = 3\,\mu\text{s}$  ; on en déduit  $N = 9$  cellules.

La méthode précédente permet de canaliser le trafic, mais ne peut éviter la congestion d'un commutateur ATM. Il faut d'ailleurs distinguer entre la congestion due à des arrivées nombreuses en rafales de cellules pendant un temps court et la congestion due à un trafic moyen, donc calculé sur une longue période, supérieur aux possibilités du réseau.

Les stratégies de contrôle de congestion principalement utilisées sont les suivantes :

-- stratégie de prévention : comme il n'est généralement pas possible de réduire le débit des émetteurs sans dénaturer l'information transmise (notamment en temps réel), il est par contre possible d'interdire l'accès au réseau si on se rend compte d'un danger de congestion par admission d'une nouvelle transmission. Ceci intervient lorsque aucune possibilité d'admission n'est trouvée sans affecter les connexions existantes.

-- stratégie de réservation : lorsqu'une nouvelle connexion est prévue, les ressources nécessaires à cette nouvelle connexion sont réservées ; ceci s'effectue au moyen d'un message spécial SETUP dont l'objet est de mobiliser, si possible, les ressources (bande passante) nécessaires.

-- stratégie basée sur la bande passante : trois modes sont définis :

- mode EFCI (Explicit Forward Congestion Indication) : un commutateur congestionné positionnera à 1 le bit EFCI de l'en-tête d'une cellule, ce qui permettra d'avertir le destinataire (amis ne règle pas le problème).
- mode RR (Relative Rate) : un commutateur congestionné envoie une cellule spéciale RM (Resource Management) vers l'émetteur pour l'inviter à réduire son débit (ceci n'est pas possible pour toutes les applications)
- mode ER (Explicite Rate) : une cellule RM est envoyée comme dans le cas précédent mais sert à indiquer à l'émetteur qu'il ne peut augmenter son débit sans accord du réseau.

## Intégration des réseaux existants

Nous examinons dans ce paragraphe l'interconnexion de réseaux existants avec ATM, notamment

- les réseaux locaux traditionnels (legacy LAN) comme Ethernet ou Token Ring
- le protocole Internet

Plusieurs solutions sont proposés. Nous en faisons une courte revue.

### Classical IP (CLIP)

Cette solution est issue des travaux du groupe de travail IPOA (IP Over ATM). Le protocole CLIP considère un ensemble de noeuds d'un réseau ATM comme un sous-réseau IP (ce qui implique évidemment que ces noeuds possèdent la pile TCP/IP et des adresses IP). Un tel sous-réseau est appelé LIS (Logical IP Subnet).

CLIP définit deux fonctionnalités :

- la résolution d'adresses
- l'encapsulation des paquets IP

La résolution d'adresse est relativement analogue à celle que l'on emploie au passage de la couche IP à une couche MAC d'un LAN ; dans ce cas le protocole ARP/RARP. Dans le cas de CLIP on emploie ATMARP et InATMAR (Inverse ATMARP). Un serveur ATMARP est nécessaire : il maintient les tables de conversion adresse ATM-adresse IP. L'adresse du serveur ATMARP est connue de toute station connectée au réseau (configuration à l'installation).

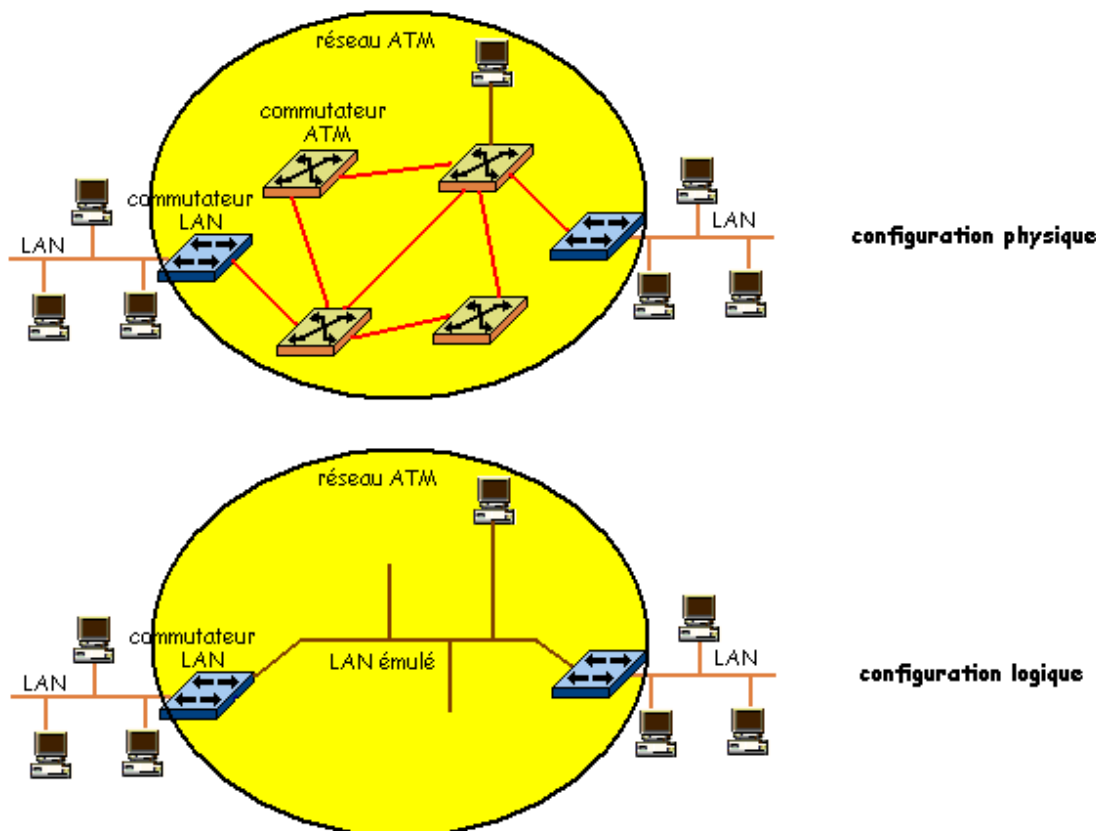
Lorsqu'une station configurée (adresse IP, adresse ATM, connaissance de l'adresse du serveur ATMARP) se connecte pour la première fois, elle se met en communication avec le serveur ATMARP par l'intermédiaire d'un circuit virtuel. Le serveur ATMARP demande à la station son adresse IP et met à jour sa table de conversion des adresses.

Si cette station veut communiquer avec un serveur, par exemple, du réseau, elle doit connaître son adresse ATM connaissant son adresse IP. Pour cela elle s'adresse à nouveau au serveur ATMARP qui lui fournit l'adresse ATM du serveur. La station enregistre cette information dans une table (ce qui évite d'avoir à recontacter le serveur ATMARP en cas d'une nouvelle connexion avec le serveur). Par suite un circuit virtuel est établi entre la station et le serveur.

En ce qui concerne l'encapsulation des paquets IP, ceux-ci sont incorporés dans une PDU de type AAL5 avec un champ indiquant le type de protocole encapsulé (ici IP, mais on peut imaginer le procédé pour d'autres protocoles). La PDU est ensuite divisée en cellules ATM.

### LAN Emulation (LANE)

Il s'agit ici de considérer un réseau ATM comme un réseau LAN classique (Ethernet ou Token Ring) :



Un LAN émulé s'appelle un ELAN (Emulated LAN) ; c'est un réseau virtuel de type Ethernet ou Token Ring. Il possède les propriétés des LAN réels : adresses MAC, broadcast, multicast et aussi des propriétés spécifiques : par exemple, pas de collision dans le cas d'un ELAN Ethernet. Les stations reliés à l'ELAN sont des LEC (Lan Emulation Client) et il doit exister deux serveurs (éventuellement confondus) : le LES (Lan Emulation Server) et le BUS (Broadcast Unknown Server). Lorsque plusieurs ELAN existent sur un même réseau ATM, un serveur LECS (Lan Emulation Configuration Server) est nécessaire.

Le serveur LECS contient les adresses ATM des LES des différents ELAN sur le réseau ATM. Le LES gère la table de conversion des adresses ATM et MAC. Le BUS est un serveur de multidiffusion (multicast). Examinons le fonctionnement du protocole LANE.

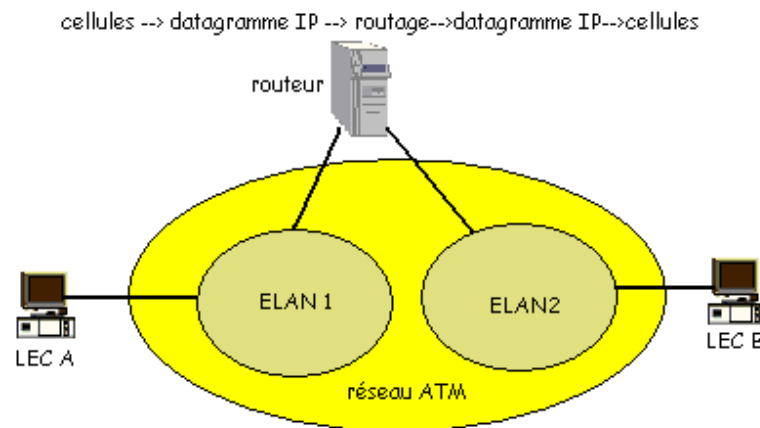
Imaginons qu'une station LEC A veuille communiquer avec une autre station LEC B. La station A commence par se connecter (par circuit virtuel) au LECS (possédant par exemple une adresse ATM réservée) pour obtenir l'adresse ATM du LES de l'ELAN concerné. La station A se connecte alors au LES (par un circuit virtuel appelé Control Direct VCC) qui met à jour sa table de conversion d'adresses ATM - MAC. La station A demande et obtient du LES l'adresse ATM de la station B. Il est alors simple pour la station A de se connecter via un circuit virtuel à la station B.

Imaginons maintenant que la station LEC A souhaite envoyer un message multicast. Pour cela, elle contacte le LES pour obtenir l'adresse ATM du serveur BUS. L'ayant obtenue elle se connecte au BUS qui se charge de la diffusion multicast du message.

### Multi-Protocol Over ATM (MPOA)

MPOA est une solution pour utiliser ATM sous divers protocoles (en fait actuellement seulement sous IP) et d'exploiter les classes de service correspondantes. Dans le protocole LANE, un réseau ATM peut être constitué de plusieurs ELAN. Le passage d'un ELAN à un autre nécessite une routeur ce qui signifie une décapsulation suivie d'une encapsulation au passage sur le routeur d'où une perte de temps faisant perdre le

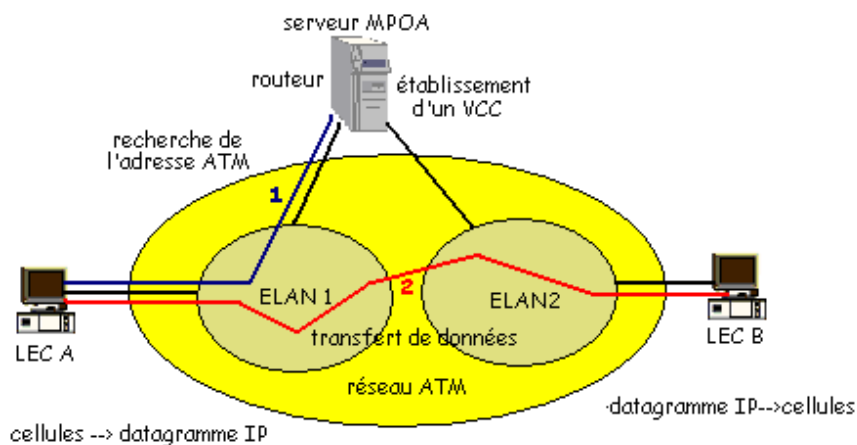
## bénéfice du réseau ATM.



MPOA remplace les routeurs par des serveurs MPOA et vise à effectuer une liaison directe entre les stations à connecter. Ceci signifie qu'au lieu de faire des passages couche 2 - couche 3 - couche 2, on effectue une connexion de bout en bout en couche 3. Le fonctionnement est le suivant. Le routeur et les clients sont dotés du logiciel adéquat du protocole MPOA.

1) phase d'appel : le client LEC A s'adresse au routeur (qui est serveur MPOA) pour obtenir l'adresse ATM du destinataire LEC B. Le serveur MPOA possède des tables de conversion. Si l'adresse n'est pas trouvée, il s'adresse au serveur MPOA suivant (en utilisant un protocole appelé NHRP (Next Hop Routing Protocol)). Si l'adresse est trouvée, il établit un circuit virtuel entre LEC A et LEC B.

2) phase de transfert : le transfert de données se fait directement via le réseau ATM entre LEC A et LEC B. Bien entendu, les paquets IP sont encapsulés et découpés en cellules (le protocole correspondant s'appelle VC Based Multiplexing).



## Exercices

[Exercice 1d](#) ; [Exercice 3](#) ; [Exercice 4](#) ; [Exercice 5](#) ; [QCM9](#) ; [QCM10](#)





# Réseau Numérique à Intégration de Services

## Sommaire :

[Généralités](#)

[Connexion et interfaces](#)

[Canaux](#)

[Signalisation](#)

## Généralités

Le réseau téléphonique est, sans conteste, le plus grand réseau mondial. Toutefois, il n'a été conçu que pour le transport de la voix et sa modernisation incluant la transmission du son numérisé devait aussi étendre le champ des médias transportés, voix mais aussi données informatiques, images fixes et animées, vidéo numériques, documents multimédias,... Telles sont les raisons qui ont influé en faveur d'un réseau numérique "universel", le Réseau Numérique à Intégration de Services (RNIS en français, ISDN = Integrated Services Digital Network en anglais). L'objectif était aussi de simplifier la vie des usagers en lui proposant un mode de connexion unique pour le téléphone, l'ordinateur et la télévision.

Ces objectifs n'ont pas été atteints, mais un RNIS a vu le jour et s'est développé, particulièrement en France, sous l'impulsion de France Télécom (le nom commercial du RNIS français est Numéris).

L'intégration de services est aussi une idée neuve qui a été surtout mise en oeuvre pour les besoins de la téléphonie seulement. Les services proposés, outre les services de transports (offre d'une infrastructure de transport de données), sont des "téléservices" comme l'audioconférence, la visioconférence, la téléalarme, ou des "compléments de service" comme les fonctions téléphoniques (indication de coûts, double appel et va-et-vient, identification d'appel, transfert d'appel, mini-message, sélection directe à l'arrivée,...)

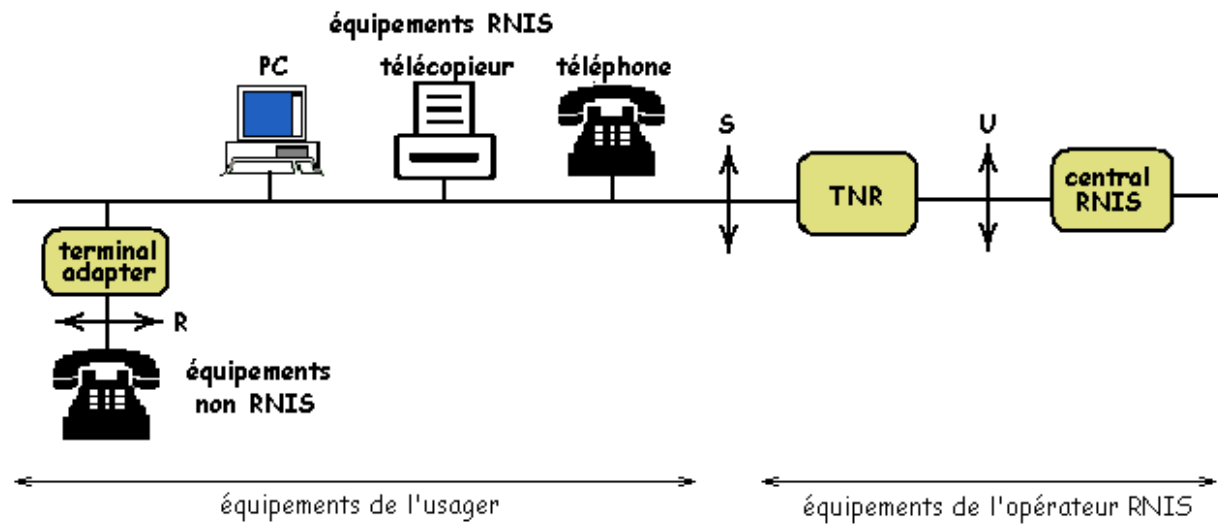
Le mot "intégration" a aussi une signification précise ; il correspond à une volonté de présenter plusieurs services simultanés sur un même support réseau ; par exemple on peut utiliser simultanément le téléphone, le fax, la transmission de fichiers.

Le réseau mis en place sous le nom de Numéris en France correspond au RNIS-BE (RNIS Bande Etroite) car les débits offerts au public ne sont pas très importants. Le RNIS-LB (RNIS Large Bande) est, quant à lui, basé sur ATM. Nous ne discutons ici que le RNIS BE et les descriptions qui suivent sont relatives à Numéris, le RNIS français.

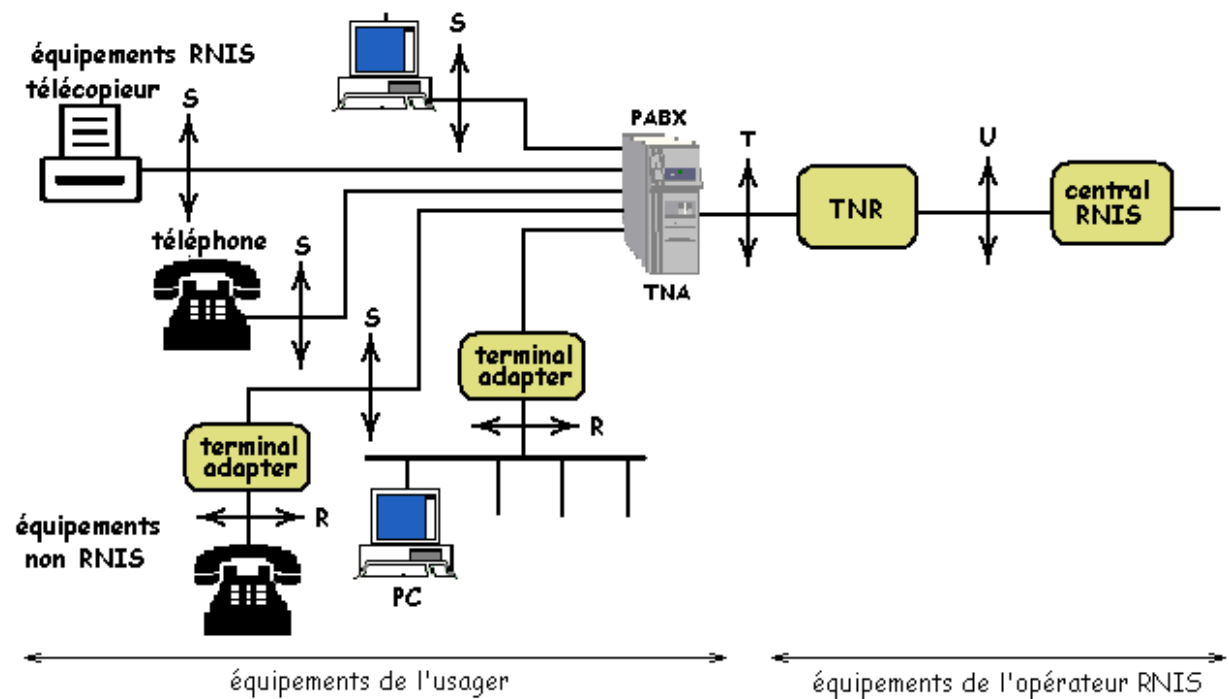
## Connexion et interfaces

En ce qui concerne les modes de connexion au RNIS, deux possibilités sont à considérer : le simple usager (à son domicile par exemple) et l'entreprise.

L'utilisateur peut posséder des équipements non compatibles RNIS et des équipements compatibles RNIS. Dans les deux cas, l'équipement de raccordement est une TNR (Terminaison Numérique de Réseau) qui permet d'accéder au réseau par une interface U. Du côté de l'abonné les équipements sont disposés sur un "bus RNIS" (on peut aller jusqu'à 8 équipements reliés) connecté au TNR par une interface S. Les équipements non RNIS peuvent aussi être reliés au bus par un boîtier spécifique TA (Terminal Adapter). Une interface R permet de connecter les équipements non RNIS au TA.



Pour une entreprise, les équipements de communication sont usuellement reliés à un PABX. La partie RNIS de ce PABX est appelée TNA (Terminaison Numérique d'Abonné). Le PABX est interfacé avec le TNR par une interface T lui-même relié au réseau par une interface U. Comme dans le cas de l'utilisateur individuel, les équipements sont reliés au TNA par des interfaces S ou un couple R/TA.



## Canaux

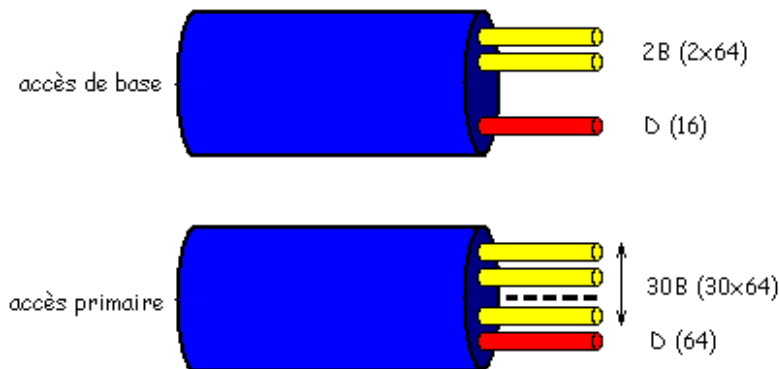
L'abonnement au RNIS comprend deux types d'accès :

- l'accès de base (BRI : Basic Rate Interface) composé de 2 canaux B et d'un canal D
- l'accès primaire (PRI : Primary Rate Interface) composé de 30 canaux B maximum et d'un canal D

accès	canaux	débits	débit total
de base	2B + D	B = 64 Kbits/s ; D = 16 Kbits/s	144 Kbits/s

primaire	30B + D	B = 64 Kbits/s ; D = 64 Kbits/s	1984 Kbits/s
----------	---------	---------------------------------	--------------

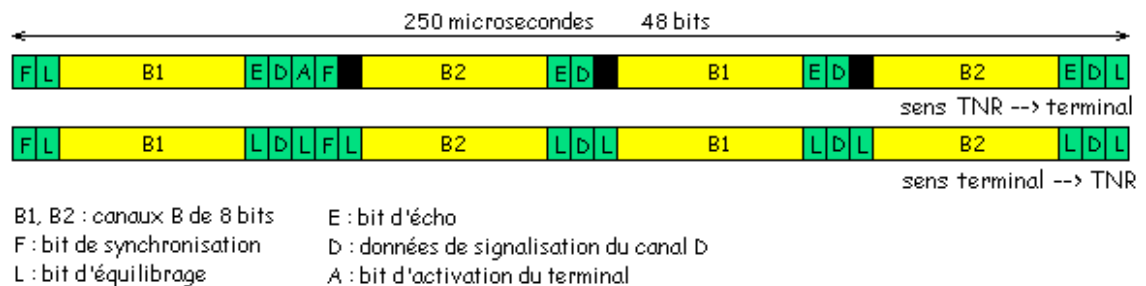
Les canaux B sont destinés au transport de données tandis que les canaux D (canaux "sémaphores") sont principalement dédiés à la signalisation. Le schéma ci-dessous donne une vue logique des accès RNIS :



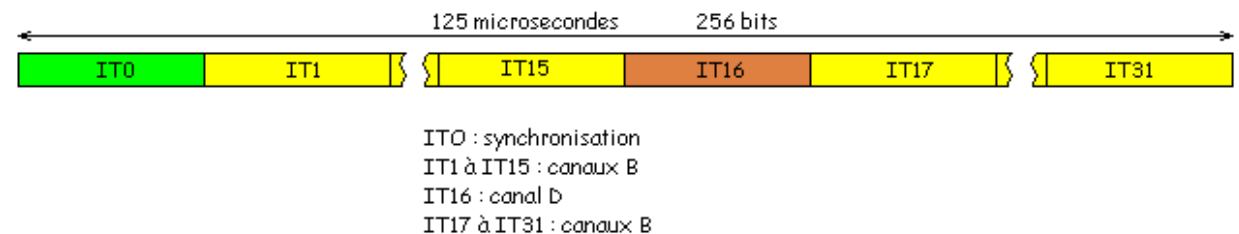
Il est également prévu des canaux H à 384, 1536, 1920 Kbits/s (non commercialisés par Numéris).

Les canaux sont en fait multiplexés sur le même support par des trames dont la structure diffère suivant le type d'accès.

- Pour l'accès de base, la trame est d'une durée 250 microsecondes et correspond à la transmission de 48 bits.

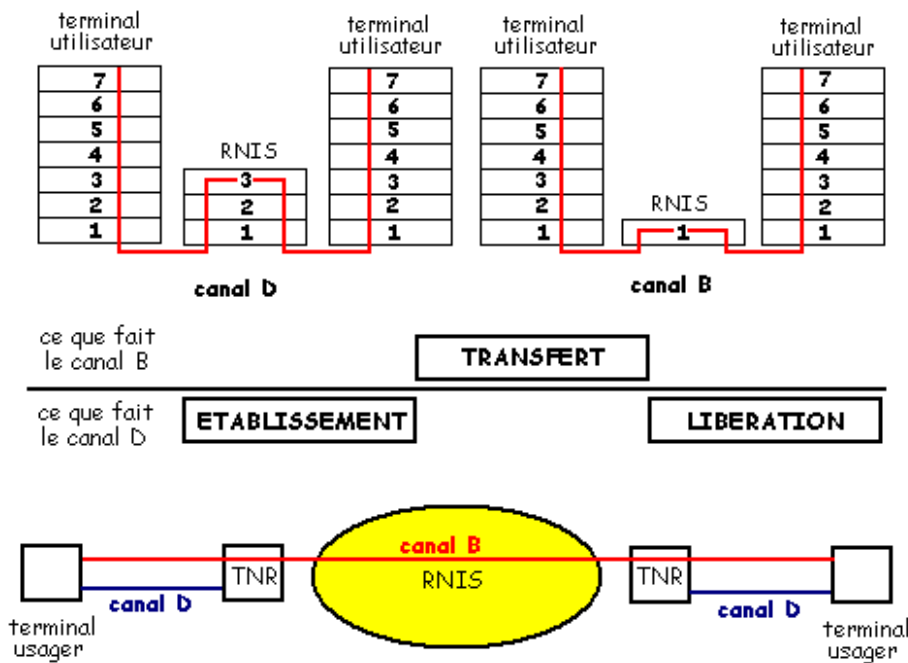


- Pour l'accès primaire, la trame est du type MIC : durée 125 microsecondes et transmission de 256 bits.



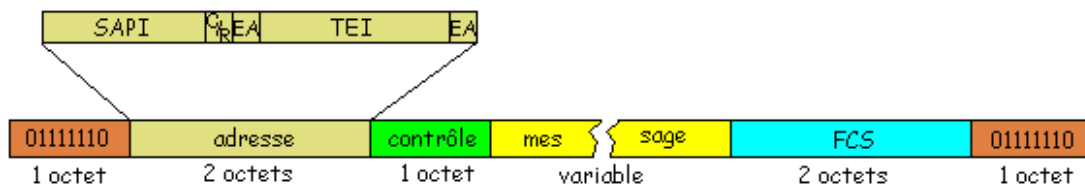
## Signalisation

Le RNIS se situe aux couches 1, 2, 3 du modèle OSI. Pour les données (canaux B), c'est uniquement la couche 1 qui est concernée. Pour la signalisation (canal D), les trois plus basses couches sont concernées. Le canal D sert à établir la communication et à la libérer.



Sur la **couche 1**, les données sont envoyées dans les trames indiquées plus haut.

Sur la **couche 2**, qui ne concerne que le canal D, on utilise les trames de la famille HDLC (trames HDLC-LAP-D).



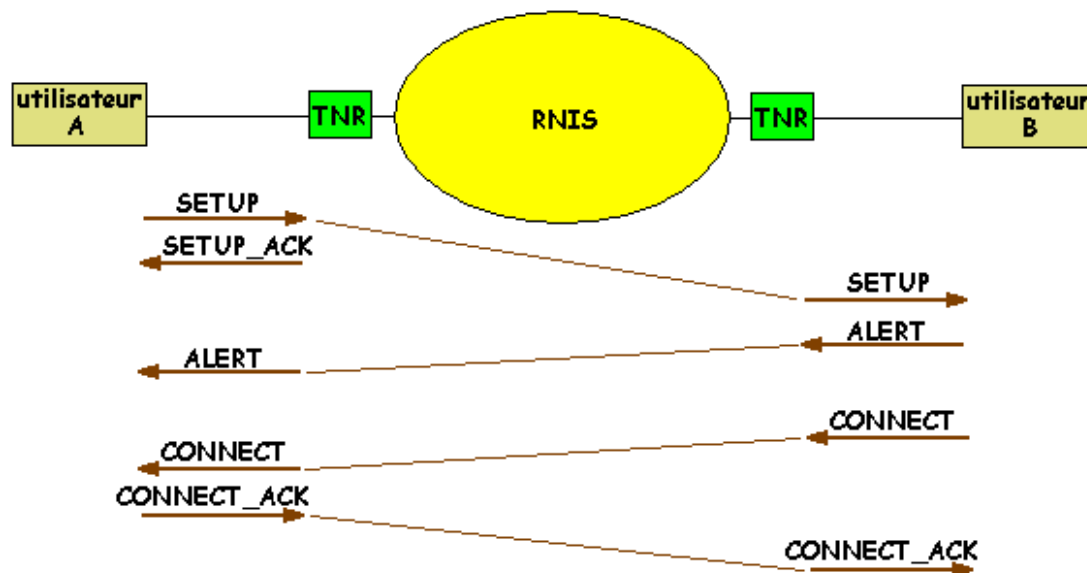
Le champ adresse du protocole HDLC est renseigné par deux paramètres principaux

- **SAPI** (Service Access Point Identifier) qui indique le protocole employé au niveau 3 : 0 : appel ; 16 demande de commutation de paquets, 63 : gestion des TEI
- **TEI** : Terminal Endpoint Identifier est une adresse de terminaux sur le bus usager. Cette adresse va de 0 à 127 (cette dernière étant réservée).

**C/R** identifie s'il s'agit d'une commande (0) ou d'une réponse (1) ; **EA** (Extension Address Bit) est un bit d'extension (il indique si le champ adresse est de 1 octet ou de 2 octets).

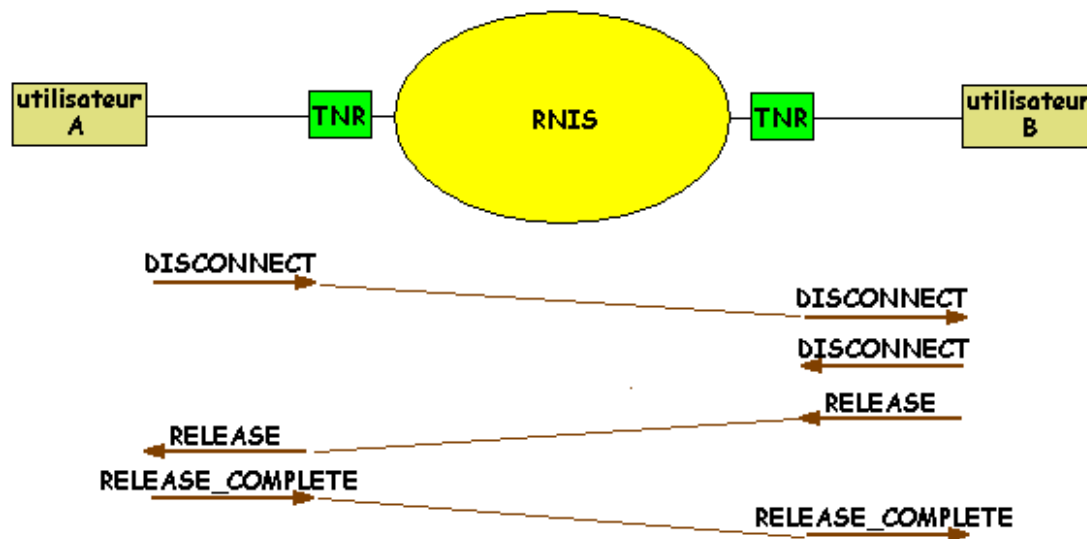
Sur la **couche 3**, qui ne concerne que le canal D, l'établissement et la libération de la voie et d'une manière générale l'interaction entre l'utilisateur et le réseau, s'effectue avec des messages.

Un appel normalement se conclut par l'établissement d'un canal B point à point entre l'utilisateur et son correspondant.



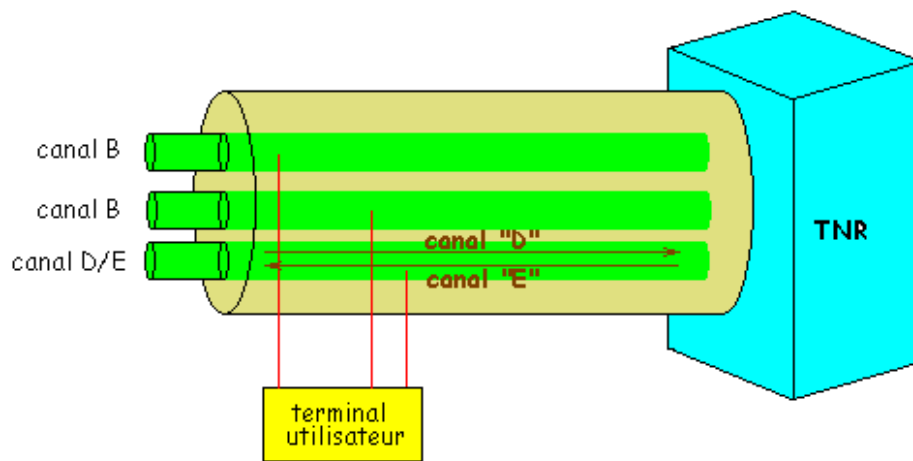
Sur un message **SETUP**, la TNR répond en proposant un canal B particulier (**SETUP\_ACK**) ce qui ne confirme pour l'instant que la demande de connexion. Le réseau établit une connexion et prévient le correspondant. Celui-ci répond avec un message **ALERT** signifiant qu'il est appelé. Le réseau transmet cette information à l'appelant. Puis l'utilisateur B envoie un message signifiant sa connexion. Ce message est répliqué par le réseau vers l'utilisateur A qui répond par un acquittement également relayé par le réseau.

La déconnexion utilise le protocole suivant :

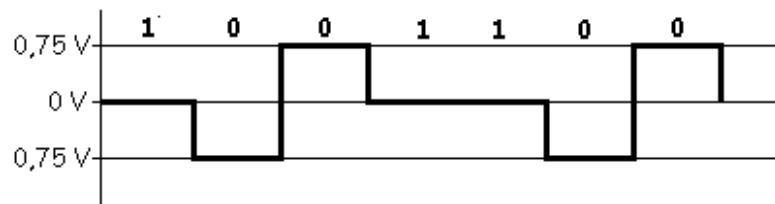


La demande de déconnexion **DISCONNECT** effectuée par le correspondant A est répliquée par le réseau vers le correspondant B, qui confirme au réseau sa réception. Puis il envoie un message de libération du canal (**RELEASE**) qui est répliqué par le réseau jusqu'à A. Celui-ci confirme à son tour la libération (**RELEASE\_COMPLETE**), message répliqué également par le réseau vers B.

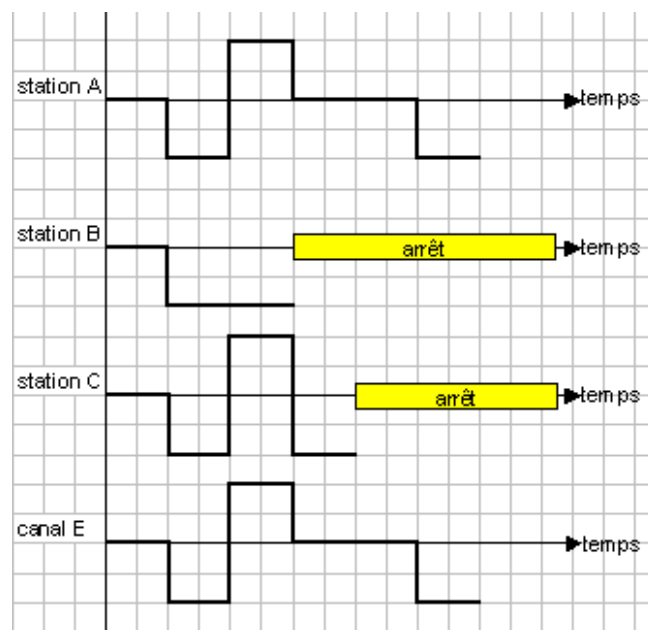
Un problème se pose cependant si l'on se rappelle que la liaison entre la TNR et l'utilisateur est en fait un bus semblable au bus Ethernet et que plusieurs terminaux y sont connectés. Un terminal qui désire envoyer des informations doit d'abord "écouter" si le bus est occupé ou non. Si le bus n'est pas occupé, alors le terminal envoie sa trame HDLC. Mais il se peut que plusieurs terminaux, constatant que le bus est inoccupé, envoient des trames simultanément. C'est pour cette raison que la TNR renvoie sur le canal "E" (canal d'écho), trame identique à celle du canal D mais répliquant les données du canal D, ce que les stations émettent (addition logique de signaux). Si les données du canal D et celles du canal E ne concordent pas, la station doit s'arrêter d'émettre.



On emploie un codage particulier à cet effet, le code AMI (Alternate Mark Inversion). Un "1" est toujours représenté par une tension nulle ; un "0" par une tension de +0,75v ou -0,75v de manière alternée :



Bien entendu, sur le canal E, un "0" masque un "1". Imaginons alors le scénario suivant où 3 stations RNIS sur le même bus souhaitent émettre en même temps.



Durant les deux premiers temps bit, les trois stations envoient les mêmes signaux. Elles continuent d'émettre. Au temps 3, la station B émet un 0 et le canal E lui retourne un 1 : elle s'arrête. Au temps 4, la station C émet un 0 et le canal E retourne un 1 : elle s'arrête. Finalement il ne reste que la station A qui continue d'émettre.

# Bibliographie

<b>D. BATTU</b>	<b>Télécommunications, Principes, Infrastructures et services</b>	<b>Dunod Informatiques</b>
<b>C. SERVIN</b>	<b>Telecoms 1, de la transmission à l'architecture de réseaux</b>	<b>Dunod Informatiques</b>
<b>C. SERVIN</b>	<b>Telecoms 2, de l'ingénierie aux services</b>	<b>Dunod Informatiques</b>
<b>W. STALLINGS</b>	<b>Data and Computer Communications</b>	<b>Prentice Hall</b>
<b>M. MAIMAN</b>	<b>Télécoms et Réseaux</b>	<b>Masson</b>
<b>P. ROLLIN, G. MARTINEAU, L. TOUTAIN, A. LEROY</b>	<b>Les Réseaux, principes fondamentaux</b>	<b>Hermes</b>
<b>A. TANENBAUM</b>	<b>Réseaux</b>	<b>InterEditions</b>
<b>G. PUJOLLE</b>	<b>Les réseaux</b>	<b>Eyrolles</b>
<b>K. JAMSA, K. COPE</b>	<b>Internet Programming</b>	<b>Jamsa Press</b>
<b>J-L. MELIN</b>	<b>Pratique des réseaux ATM</b>	<b>Eyrolles</b>

# Exercices et Tests

sommaire :

[Enoncés](#)

[Solutions](#)

[QCM](#)

## Exercice 1

Une voie à 32 Mbits/s est utilisée pour la transmission de messages multimédias. On suppose que le message à transmettre sur cette voie est un document composé d'un texte de 20 Ko, de 30 images fixes en format GIF de 10 Ko chacune, d'une minute de son numérisée à 22 KHz et codée sur 8 bits.

- a) Donner en octets, le volume du son à transporter en admettant qu'il n'y a aucune compression
- b) Si le message est transmis intégralement d'un seul bloc, quel est le temps nécessaire à son acheminement en supposant que le temps de propagation du signal est négligeable.
- c) La voie concernée fait partie d'un réseau à commutation de paquets. Chaque paquet a une longueur en octets de 1024 comprenant une partie de service (adresses, détection d'erreur, champs de service) de 256 octets. Combien de paquets correspondent au message précédent
- d) Même question pour le cas d'un réseau ATM ; combien de cellules ATM sont-elles nécessaires pour véhiculer le message.

---

## Exercice 2

On utilise la technologie du Relais de Trame (Frame Relay) pour effectuer du transfert de données. Le contrat souscrit porte sur un débit de 64 Kbits/s (Committed Information Rate) et la tolérance de dépassement est de 16 Kbits/s. Sachant que au bout d'une durée de 60 secondes, l'utilisateur totalise une quantité de 6 millions de bits que se passe-t-il pour la dernière trame ? (cochez la bonne réponse).

<input type="checkbox"/>	Elle est transmise
<input type="checkbox"/>	Elle est transmise mais sera détruite en cas de difficulté
<input type="checkbox"/>	Elle est détruite



---

### Exercice 3

Une image de 640x480 pixels codés sur 24 bits est envoyée non compressée sur un réseau de type ATM. A combien de cellules ATM, ce message correspondra-t-il ?

---

### Exercice 4

Un canal téléphonique utilise la plage de fréquence 300Hz – 4000Hz.

a) Le son est échantillonné à la fréquence minimale permise par le théorème de l'échantillonnage ( $2f_{\max}$ ), soit 8000 Hz et est codé sur 8 bits. On désignera dans la suite par M le message constitué d'une minute de son non compressé, numérisé de cette façon.

1) Quel est le volume du message M ?	
2) Quel est le débit nécessaire de la voie transmettant le son en temps réel ?	
3) Quel réseau peut-on utiliser ?	

b) Supposons que le débit de la voie soit 64 Kbits/s . On utilise des trames (MIC) de multiplexage temporel constituées de 32 IT (intervalles de temps) de longueur identique. On suppose que l'on utilise un IT par trame MIC pour transmettre le message M. Combien de temps faut-il pour transmettre de cette manière le message M ?

c) On utilise un réseau ATM pour le transport du message M

- Combien de cellules ATM faut-il pour transporter le message M ?
- En admettant que toute la voie puisse être utilisée (débit : 155 Mbits/s), pour transmettre deux messages du même type que M (les deux messages sont multiplexés) , combien de temps, au minimum, est nécessaire pour transmettre un message M ?

---

### Exercice 5

Un message de 1 Mo est transmis par un réseau ATM de débit 155,52 Mbits/s. Le contrat de service stipule que l'on peut envoyer des cellules en rafales à la cadence de 100000 cellules/s. On désigne par T la durée séparant l'émission de deux cellules successives. Toutefois, la tolérance h est de 20 microsecondes comme avance maximum permise sur T.

- a) Quel est le nombre de cellules ATM nécessaires ?
  - b) Quel est le débit souscrit ?
  - c) En fonctionnement "normal" quel est le temps T, en microsecondes.
  - d) Supposons que le "client" ATM envoie ses cellules à la cadence de 200 000 cellules/s. Combien de cellules pourra-t-il envoyer avant rejet ?
- 

## Solution de l'exercice 1

- a) Le volume en bits du son à transporter est  $V = 60 * 22000 * 8 = 10\,560\,000$  bits = 10,56 Mbits
- b) Calculons le volume total :

texte :  $20 * 1024 * 8 = 163\,840$  bits  
images :  $30 * 10 * 1024 * 8 = 2\,457\,600$  bits  
son : 10 560 000 bits d'après la question a)  
total : 13 181 440 bits

Le temps de propagation est approximativement  $T = 13\,181\,440 / 32\,000\,000 = 0,41$  s.

- c) La longueur utile d'un paquet est  $1024 - 256 = 768$  octets = 6 144 bits. Le nombre de paquets est donc  $13\,181\,440 / 6\,144 = 2\,146$  paquets
- d) Une cellule ATM contient 53 octets dont 48 utiles. Le nombre de cellules nécessaires est donc  $13\,181\,440 / (48 * 8) = 34\,327$  cellules

---

## Solution de l'exercice 2

Elle est détruite : On a droit à 4 926 720 bits au maximum avec la tolérance ; la dernière trame sera donc détruite

---

## Solution de l'exercice 3

Chaque cellule ATM contient 48 octets utiles, donc le nombre de cellules est 19200

---

## Solution de l'exercice 4

a)

1) Quel est le volume du message M ?	480 Ko
2) Quel est le débit nécessaire de la voie transmettant le son en temps réel ?	64 Kbits/s
3) Quel réseau peut-on utiliser ?	Numéris (RNIS)

b) Tout se passe comme si le débit était de  $64 \text{ Kbits/s} / 32$ , soit  $2 \text{ Kbits/s}$ . On en déduit le résultat : 1920 secondes. (Noter que, en fait, le débit offert par France Telecom pour les liaisons MIC est de  $2 \text{ Mbits/s}$ , ce qui revient à un débit de  $64 \text{ Kbits/s}$  par canal ; on obtiendrait alors un temps de transmission de 60 s.)

c)

Nombre de cellules :	10 000 cellules car la longueur cellulaire ATM est de 48 octets.
Temps de transmission :	0,05 seconde

---

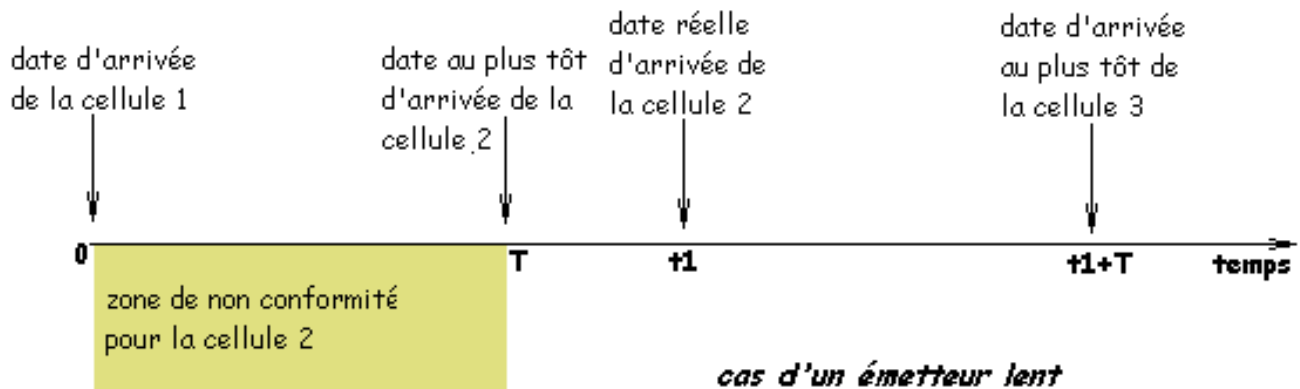
## Solution de l'exercice 5

a) 20834 cellules

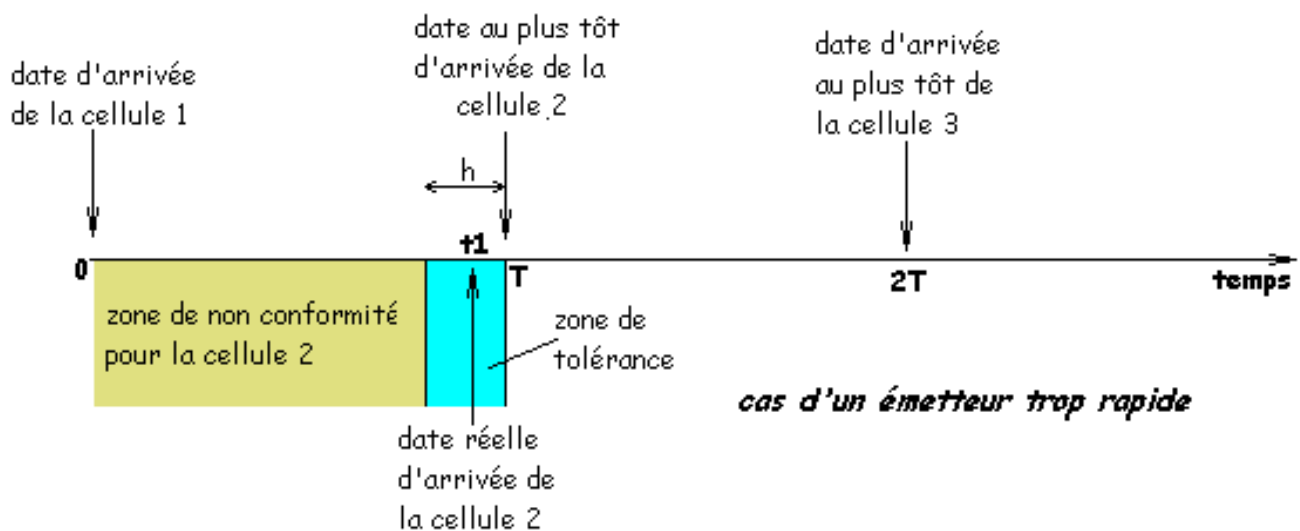
b) 42,4 Mbits/s

c) 10 microsecondes

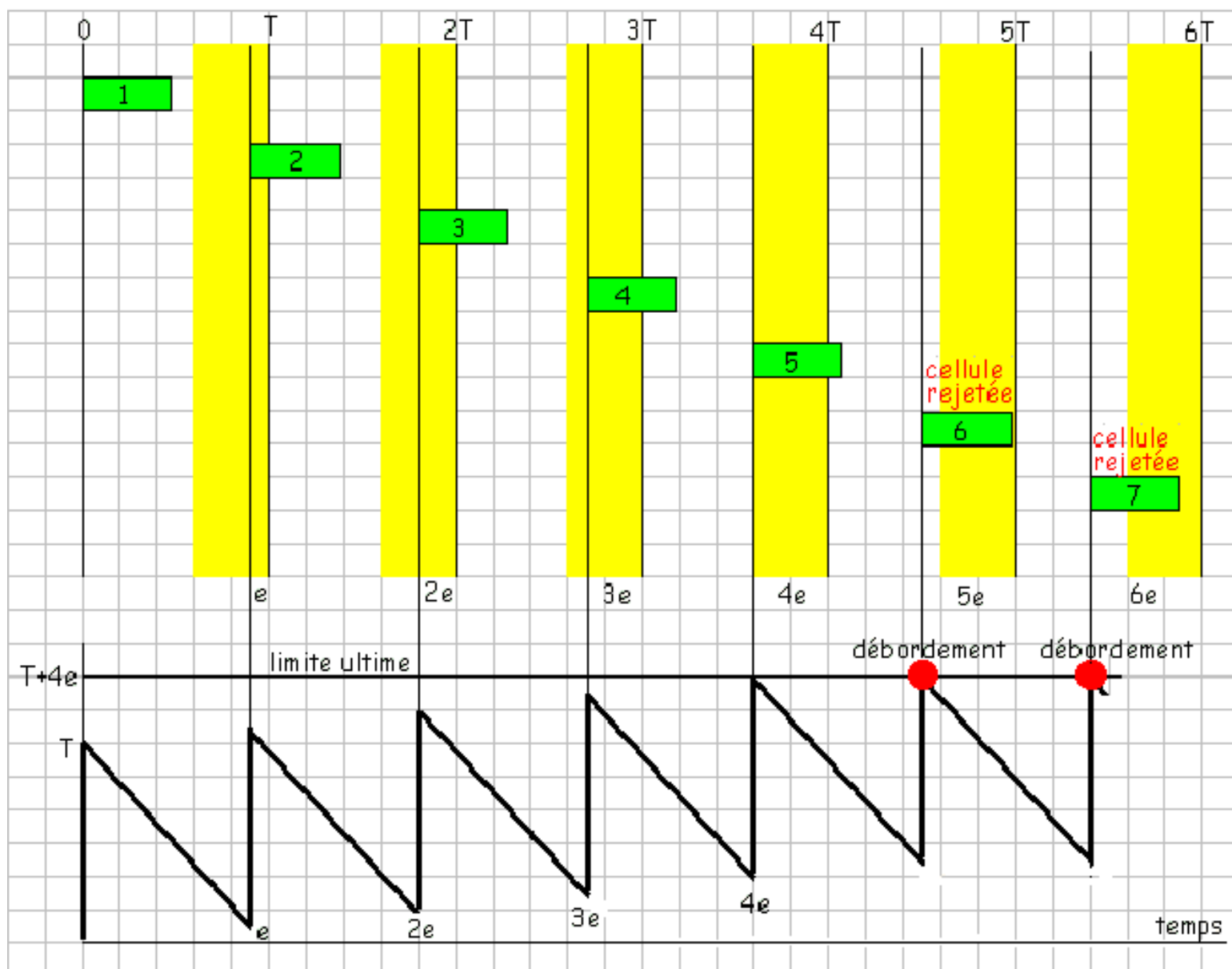
d) Imaginons que le contrat porte sur un débit  $D_0 = \text{PCR} = 100\,000$  cellules/s. Le temps écoulé entre deux envois successifs de cellules est donc  $T = 1/D_0$ . Une cellule ne doit donc pas arriver en un temps plus court que  $T$  après la réception de la cellule précédente. Elle peut par contre arriver dans un intervalle de temps plus grand que  $T$ . Une cellule qui se conforme à cette règle est "conforme". Si la cellule 1 arrive au temps 0 et si la cellule 2 arrive au temps  $t_1 > T$ , la cellule 3 devra arriver au plus tôt au temps  $t_1 + T$ .



Bien entendu, il se pose un problème lorsque l'émetteur "triche" en accentuant sa cadence : les cellules deviennent alors non conformes au contrat. On peut toutefois accepter une tolérance de  $h$  microsecondes (qui correspond au paramètre CVDT). On pourra considérer que la cellule est encore conforme si elle arrive au temps  $T-h$  après la cellule précédente. Mais il faudra que la cellule suivante arrive au plus tôt au temps  $2T$ .



Imaginons maintenant que l'émetteur trop rapide conserve sa cadence d'envoi. Les cellules suivantes vont s'enfoncer dans la zone de tolérance et au bout d'un moment avoir une date d'arrivée dans la zone stricte de non conformité. La cellule 2 va arriver trop tôt, la cellule 3 va arriver 2e trop tôt, la cellule 4 va arriver 3e trop tôt, etc... Le schéma ci-dessous montre une telle situation avec  $h = 4e$ . Il est clair que la cellule 6 est en totale "effraction". Elle pourra être détruite par les composants réseau.



Le bas du schéma précédent explique pourquoi l'algorithme est qualifié de "seau percé". La cellule est considérée comme un récipient apportant une quantité  $T$  litres de liquide. A l'arrivée, ce liquide est intégralement versé dans un seau de capacité  $T+h$  litres qui possède un trou et fuit à la cadence de  $T$  litres/seconde. Si les cellules respectent le contrat, elles trouveront toujours le seau vide. Si au contraire, elles arrivent en avance, elles trouveront un seau non vide ; le niveau de liquide du seau s'élèvera donc à chaque arrivée d'une cellule jusqu'au moment où le seau débordera.

On voit que temporairement, on peut dépasser le débit maximum, mais que seulement  $N$  cellules seront acceptées. Le calcul de  $N$  est assez simple si l'on se base sur le tableau ci-dessous :

cellule	date d'arrivée au plus tôt	date réelle d'arrivée
1	0	0
2	$T$	$T-e$
3	$2T$	$2(T-e)$
4	$3T$	$3(T-e)$
---	---	---
$N$	$(N-1)T$	$(N-1)(T-e)$

Pour la dernière cellule N, on aura atteint la limite de tolérance, ce qui correspond à la date  $(N-1)T-h$ , donc  $(N-1)(T-e) = (N-1)T - h$ , d'où la valeur de N :

$$N = 1 + \frac{h}{e}$$

e mesure l'intervalle de temps d'avance à chaque arrivée de cellules. Si le débit souscrit est  $D_0$  et si le débit réel est  $D > D_0$  (les débits étant mesurés en cellules par seconde), le temps d'arrivée au plus tôt de la cellule 2 est  $T = 1/D_0$  et son temps d'arrivée réel est  $T' = 1/D$ , d'où

$$e = T - T' = \frac{D - D_0}{D D_0}$$

Reprenons maintenant l'exercice : le débit souscrit est  $D_0 = 100\,000$  cellules/s d'où  $T = 8\,\mu\text{s}$ . Le récepteur envoie des cellules au débit  $D = 200\,000$  cellules/s.

La valeur de e est  $T - T' = 10 - 5 = 5\,\mu\text{s}$  ; on en déduit  $N = 5$  cellules.

## QCM

1) La commutation de circuit nécessite

- 
- 
- 
- 
- 

2) La commutation de paquets utilise des blocs d'information

- - 
  - 
  - 
  -
- 

### 3) Le routage par inondation consiste

- - 
  - 
  - 
  -
- 

### 4) Le protocole X25 comporte un protocole de couche 1 : X21, un protocole de couche 3 : X25 paquet et un protocole de couche 2 qui est

- - 
  - 
  - 
  -
- 

### 5) Dans le protocole X25, un paquet doit ouvrir la connexion. Il s'agit

- - 
  - 
  - 
  -
- 

### 6) Le relais de trame est

- 
- 
-

- -
- 

7) Le relais de trame utilise des trames portant un numéro DLCI qui représente

- - 
  - 
  - 
  -
- 

8) Dans la technologie "relais de trame", si pendant un temps  $T$ , le nombre  $N$  de bits émis dépasse la somme du Committed Burst Size  $B_c$  et du Excess Burst Size  $B_e$ , la trame courante est

- - 
  - 
  - 
  -
- 

9) Dans les cellules ATM, l'information utile a une longueur fixe de

- - 
  - 
  - 
  -
- 

10) Dans ATM, le contrôle de la conformité au débit souscrit peut être effectué par l'algorithme

- 
- 
- 
- 
-



## Bibliographie

D. BATTU	Télécommunications, Principes, Infrastructures et services	Dunod Informatiques
P. LECOY	Technologie des Télécoms	Hermes
C. SERVIN	Telecoms 1, de la transmission à l'architecture de réseaux	Dunod Informatiques
W. STALLINGS	Data and Computer Communications	Prentice Hall
G. BOUYER	Transmissions et réseaux de données	Dunod
M. MAIMAN	Télécoms et Réseaux	Masson
P. ROLLIN, G. MARTINEAU, L. TOUTAIN, A. LEROY	Les Réseaux, principes fondamentaux	Hermes
A. TANENBAUM	Réseaux	InterEditions
P-G. FONTOLLIET	Systèmes de télécommunications, bases de transmission	Dunod

Simon Nora et Alain Minc, "La télématique. Rapport...", éditions..., 197x

Guy Pujolle, "Les Réseaux", Eyrolles

Andrew Tanenbaum, "Computer Networks", Prentice Hall

Z 70-001, Norme expérimentale, Systèmes de traitement de l'information, Modèle le référence  
de base pour l'interconnexion de systèmes ouverts, AFNOR, 1982

## Les livres :

- **TCP/IP, Karanjit S.Siyan, CampusPress**  
Les protocoles IP et TCP sont très bien détaillés
- **TCP/IP, Joe Casad, Campus Press**  
Moins complet et technique que le précédent mais les protocoles de la couche Application sont présentés de façon assez complète

## Sur le Net :

- Les RFC de TCP/IP