Université Nice Sophia Antipolis

Systèmes et Réseaux (S5) / L3 Miage

Cours Réseaux / 2018-19

UNIVERSITÉ CÔTE D'AZUR

CM 2:
Couche Physique

D'après le cours de Bruno Martin et les slides du livre "Computer Networking: A Top Down Approach, 6th edition, Jim Kurose, Keith Ross, Addison-Wesley, March 2012"

Ramon APARICIO-PARDO

Ramon.Aparicio-Pardo@unice.fr

10/12/2018

PLAN CM 2

1. NOTIONS DE BASE
2. SUPPORTS PHYSIQUES
3. TRANSMISSION
4. NUMÉRISATION
5. ERREURS

La couche physique

- Elle détermine comment les éléments binaires sont transportés sur un support physique (câbles, ondes, lumière):
  1. D'abord, les données sont ***codés*** en une suite binaire de 0 et de 1.
  2. Ensuite, ces données sont ***transmis*** sur le *support* (le moyen de transmission: l'air, un câble, …) en modifiant une *variable* (p. ex. la fréquence ou l'amplitude de la tension électrique ou d'une onde électromagnétique) du monde réel selon une technique appropriée au support

- Parmi toutes les couches, celle-ci est le seule où la transmission physique de l'information a lieu. On peut considérer que les autres couches ne font que *traiter* les données, p. ex., encapsulant/décapsulant les paquets, écrivant/lisent des en-têtes, ….

Le codage et la transmission

- ***Première étape: Codage***
  - Toute information à transmettre peut être vue comme une suite des *symboles.*
    - P. ex. pensez à la voix humaine comme une suite des *phonèmes* ou à un texte comme une suite des *caractères*.
  - Le codage « code » chaque *symbole* comme une suite de *bits* (éléments binaires: 0 et 1).
  - Le nombre des *symboles* qu'on peut coder en binaire dépende du nombre de bits utilisés:
    - en ASCII simple, chaque symbole (un *caractère*) utilise 7 bits. Donc, on peut représenter 128 (2^7) caractères distincts.
    - en EBCDIC, chaque symbole (un *caractère*) utilise 8 bits. Donc, on peut représenter 256(2^8) caractères distincts.
    - en UNICODE (UTF-16), chaque symbole (un *caractère*) utilise 16bits. Donc, on peut représenter 65536 (2^{16}) caractères distincts.

Le codage et la transmission

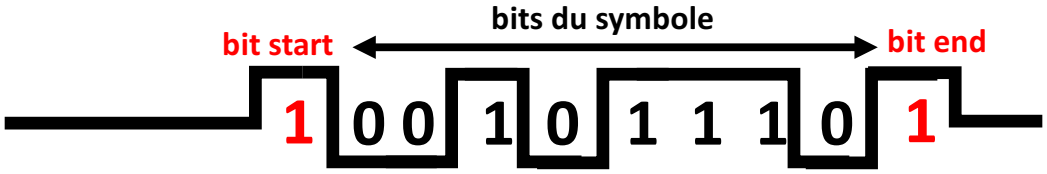
- ***Deuxième étape: Transmission***
  - C'est tout simplement l'envoi des suites binaires des symboles sur le support physique, …
  - … mais cela n'est pas si simple: ***elle suppose « adapter » les suites binaires au support de transmission*** en dépendant
    1. des caractéristiques physique du support physique
    2. de la solution technologique employée pour exploiter le support
  - Très souvent, il faut répondre au plusieurs questions:
    - Transmission en *série* ou en *parallèle ?*
    - Signaux *analogiques* ou *numériques ?*
    - *Quel support de transmission ?*
    - *Quel multiplexage ?*
    - *Quelle modulation ?*
    - *Quel codage ?*

Mode de transmission

- ***Mode de transmission en série***
  - Les bits sont envoyés les uns derrière les autres. La succession de symboles peut se faire soit en mode *synchrone, soit* en mode *asynchrone.*
- ***Mode de transmission en parallèle***
  - Les bits d'un même symbole sont envoyés simultanément sur un nombre de fils parallèles (distincts) *égal a un diviseur* du nombre de bits utilisés pour coder le symbole.
    - P. ex. si on utilise 8 bits, on peut envoyer sur 2, 4 ou 8 fils parallèles.
- ***Gestion de la synchronisation***
  - *Mode en parallèle:* Ce mode est très sensible aux arrivés tardives de bits en dépendant de fils de transmission. En effet, tous les bits d'un même caractère doivent impérativement arriver en même temps. On le privilège pour la courte distance (peu de perturbations)

Mode de transmission

- ***Gestion de la synchronisation***
  - *Mode en série asynchrone:* Aucune relation (coordination) est préétablie entre l'émetteur et le récepteur. Les bits d'un même symbole doivent donc être encadrés de signaux (délimiteurs). Le début d'une transmission est donc indifférente au temps.



  - *Mode en série synchrone:* L'émetteur *et* le récepteur doivent se mettre d'accord sur un intervalle constant de temps (ou *slot*), qui se répète sans arrêt dans le temps. Les bits d'un caractère sont envoyés les uns derrière les autres et sont synchronisés avec le début des intervalles (*slots*). Il n'y a donc aucune séparation (délimiteur). Les bits sont émis en séquence. Ce mode est traditionnellement utilisé pour les très forts débits

Le signal

- Lors de l'adaptation au support de transmission, l'émetteur transforme les informations à transmettre en variations temporelles d'une propriété *physique* (voltage, pression de l'air, champ électromagnétique) que le récepteur peut détecter de son côté. ***Ces variations temporelles sont des signaux.***
- Types des signaux:
  - **Analogiques** : Le signal prend des valeurs continues dans un intervalle.
    - P. ex. *la voix humaine:* des changements de *la pression atmosphérique* en prenant n'importe quelle valeur dans le plage audible des intensités sonores.
  - **Numériques** : Le signal ne prend que des valeurs discrètes définies dans un ensemble fini.
    - P. ex. *une suite binaire dans un fils:* des changements *du voltage électrique* dans les bornes d'un fil en prenant que *deux* valeurs possibles de voltage (-5V, +5V)

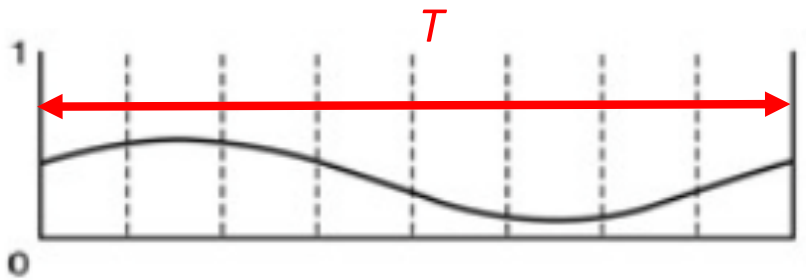
Le signal

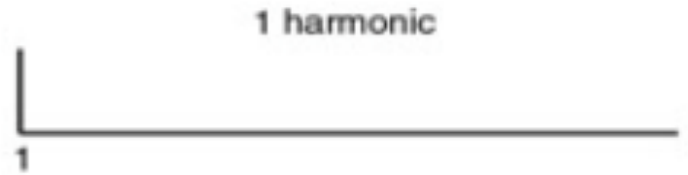
- **Numérisation:** *A grosso modo*, transformer un signal analogique en numérique.
- En pratique, les signaux qu'on retrouve dans les ordinateurs et dans les réseaux des ordinateurs sont des signaux numériques.
- Donc, les signaux numériques c'est le format de l'information dans les ordinateurs, puisque ils manipulent l'information sous cette forme (0 ou 1) .
- Mais à la sortie d'un équipement, on peut être amenée à changer la forme de l'information en fonction du support physique et de la technologie → *Tout simplement, on ne peut pas toujours envoyer la signal numérique sur le support sans avoir de dégradations*.
- Donc, on doit considérer les caractéristiques physiques du support:
  - bande passante
  - débit binaire
  - perturbations

Le signal

- Typiquement, un signal est transmis physiquement sur la forme d'une *onde* (électromagnétique, acoustique, …), c.-à-d.., comme la propagation d'une perturbation, souvent périodique, qui produit sur son passage une variation réversible des propriétés physiques.
  - Exemple: Une sinusoïde $y(t) = A \sin(2\pi f t)$ a un seul harmonique centré dans la fréquence *f*
  - *A* est amplitude, *f* la fréquence, *t* le temps
  - Période T (s) est le temps entre entre deux points consécutifs de la même amplitude
  - Fréquence*: F = 1/T (Hz)* est le nombre de cycles (*périodes* par sec.) →**X** cycles/sec = **X** Hz
  - Longueur d'onde λ (m) est la distance entre deux points consécutifs de la même amplitude. Donc, si l'onde se propage à une vitesse *v (m/s):*

$$\lambda = v\,T = v / f$$





Le signal

- Composition d'un signal :
  - Un onde périodique peut être décomposée en une suite infinie de fonctions périodiques sinusoïdales : les *composantes harmoniques* .
  - Chaque composante a une fréquence propre.
  - Les harmoniques de fréquence supérieure à la bande passante (BP) ne sont pas transmises → perte de qualité (déformation) du signal.

Le signal

❖ Décomposition de Fourier d'un signal selon ses harmonique

1 harmonique:

$y(t) = A_1 \sin(2\pi \mathbf{f_0} t) + B_1 \cos(2\pi \mathbf{f_0} t)$

2 harmoniques:

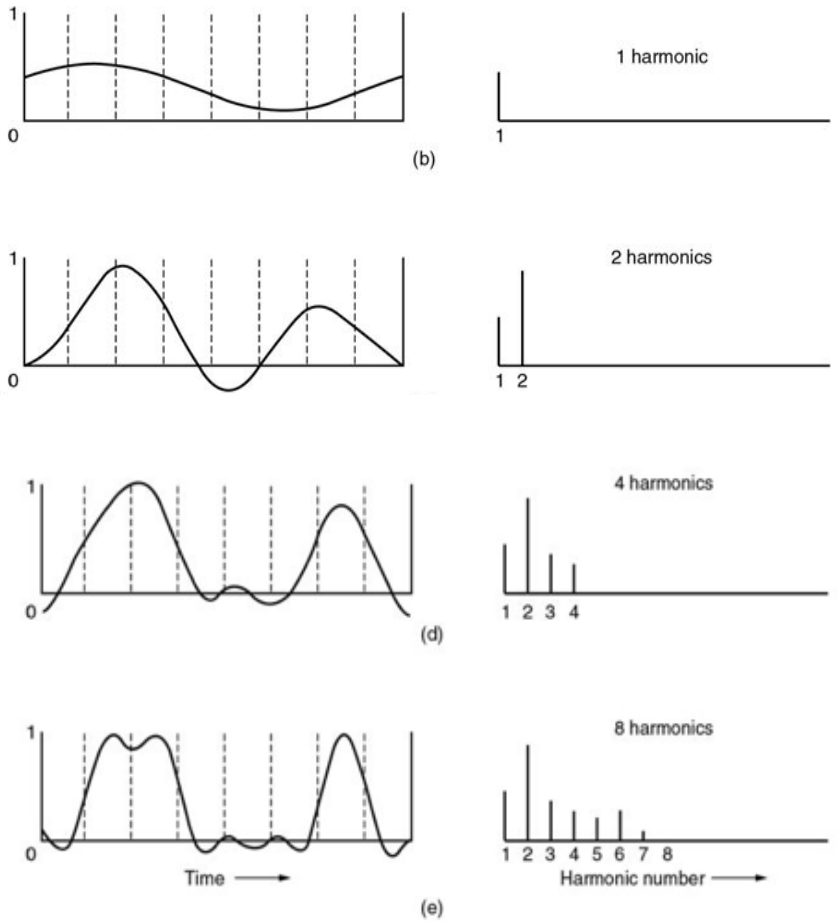
$y(t) = A_1 \sin(2\pi \mathbf{f_0} t) + B_1 \cos(2\pi \mathbf{f_0} t) +$
$+ A_2 \sin(2\pi \mathbf{2 f_0} t) + A_2 \cos(2\pi \mathbf{2 f_0} t)$

4 harmonique:

$y(t) = A_1 \sin(2\pi \mathbf{f_0} t) + B_1 \cos(2\pi \mathbf{f_0} t) +$
$+ A_2 \sin(2\pi \mathbf{2 f_0} t) + A_2 \cos(2\pi \mathbf{2 f_0} t)$
$+ A_3 \sin(2\pi \mathbf{3 f_0} t) + B_3 \cos(2\pi \mathbf{3 f_0} t) +$
$+ A_4 \sin(2\pi \mathbf{4 f_0} t) + A_4 \cos(2\pi \mathbf{4 f_0} t)$

8 harmoniques:

$y(t) = A_1 \sin(2\pi \mathbf{f_0} t) + B_1 \cos(2\pi \mathbf{f_0} t) +$
$+ \ldots +$
$+ A_8 \sin(2\pi \mathbf{8 f_0} t) + A_8 \cos(2\pi \mathbf{8 f_0} t)$



Le signal

- **Signal numérique (0 et 1) → Idéal !!!**
  - ***Flancs parfaits impossibles !!***
  - *Mais on peut l'approx. avec des harmoniques !*

1 harmonique:

$y(t) = A_1 \sin(2\pi \mathbf{f_0} t) + B_1 \cos(2\pi \mathbf{f_0} t)$

2 harmoniques:

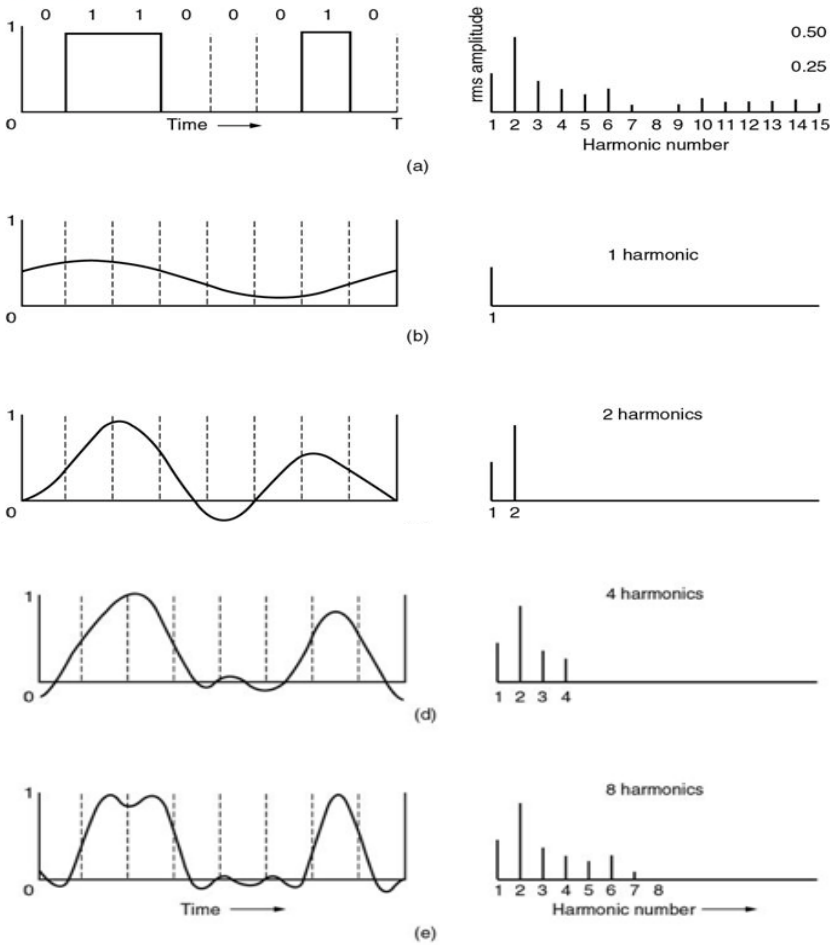
$y(t) = A_1 \sin(2\pi \mathbf{f_0} t) + B_1 \cos(2\pi \mathbf{f_0} t) +$
$+ A_2 \sin(2\pi \mathbf{2 f_0} t) + A_2 \cos(2\pi \mathbf{2 f_0} t)$

4 harmonique:

$y(t) = A_1 \sin(2\pi \mathbf{f_0} t) + B_1 \cos(2\pi \mathbf{f_0} t) +$
$+ A_2 \sin(2\pi \mathbf{2 f_0} t) + A_2 \cos(2\pi \mathbf{2 f_0} t)$
$+ A_3 \sin(2\pi \mathbf{3 f_0} t) + B_3 \cos(2\pi \mathbf{3 f_0} t) +$
$+ A_4 \sin(2\pi \mathbf{4 f_0} t) + A_4 \cos(2\pi \mathbf{4 f_0} t)$

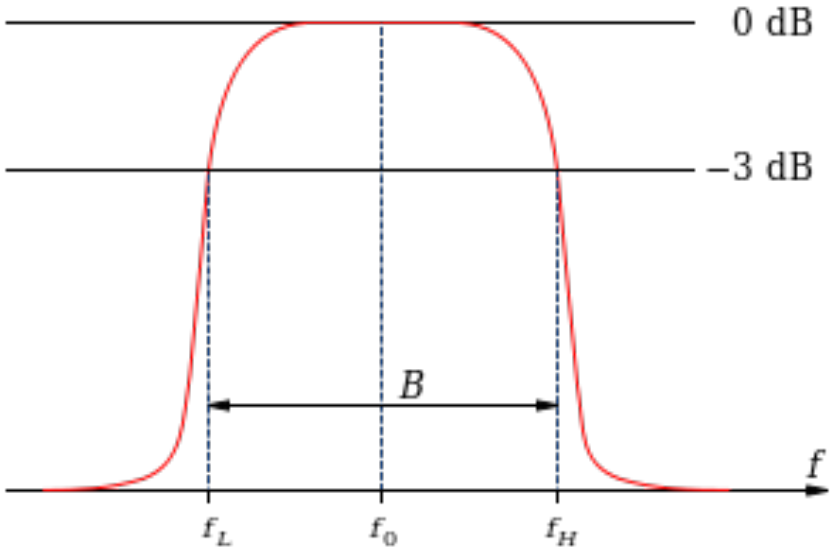
8 harmoniques: → Bonne approximation

$y(t) = A_1 \sin(2\pi \mathbf{f_0} t) + B_1 \cos(2\pi \mathbf{f_0} t) +$
$+ \ldots +$
$+ A_8 \sin(2\pi \mathbf{8 f_0} t) + A_8 \cos(2\pi \mathbf{8 f_0} t)$



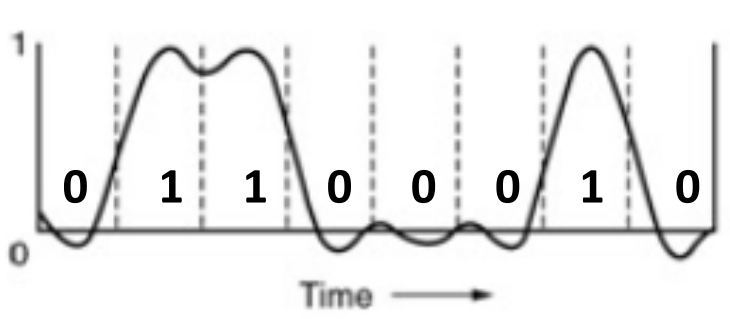
Bande passante

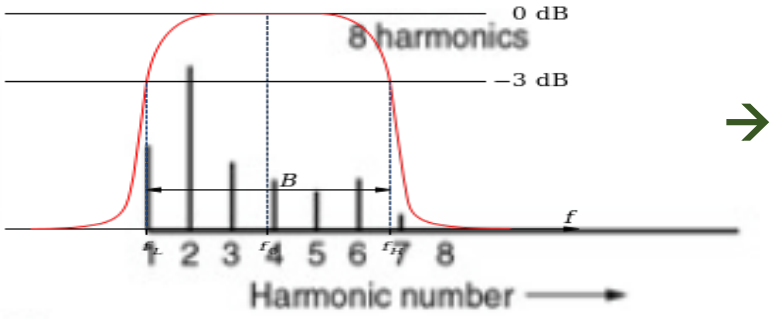
- Qu'est la *bande passante* pour un support physique ?
  - bande de fréquence dans laquelle les signaux sont correctement reçus (non ou très peu déformés) : fréquence de coupure max. (f_H) et min. (f_L)
  - caractérise tout support de transmission
  - exprimée en Hertz (Hz)
  - influence directement le *débit binaire*



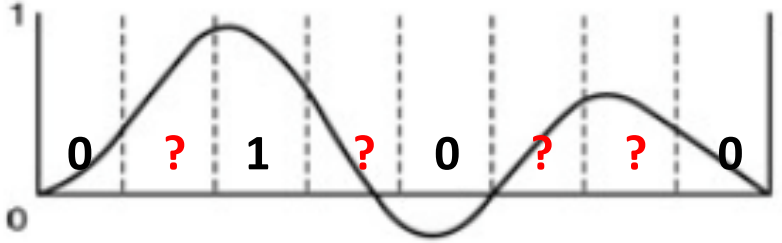
Bande passante et harmoniques

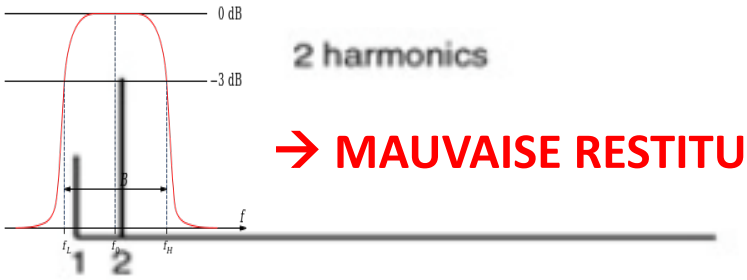
- Plus la bande passante est élevée, plus le nombre de composantes harmoniques transmises est grand et donc mieux le signal est transmis
- La qualité de restitution (identifier des 0 et des 1) du signal sera donc meilleure





→ OK !!!





→ MAUVAISE RESTITUTION !!!

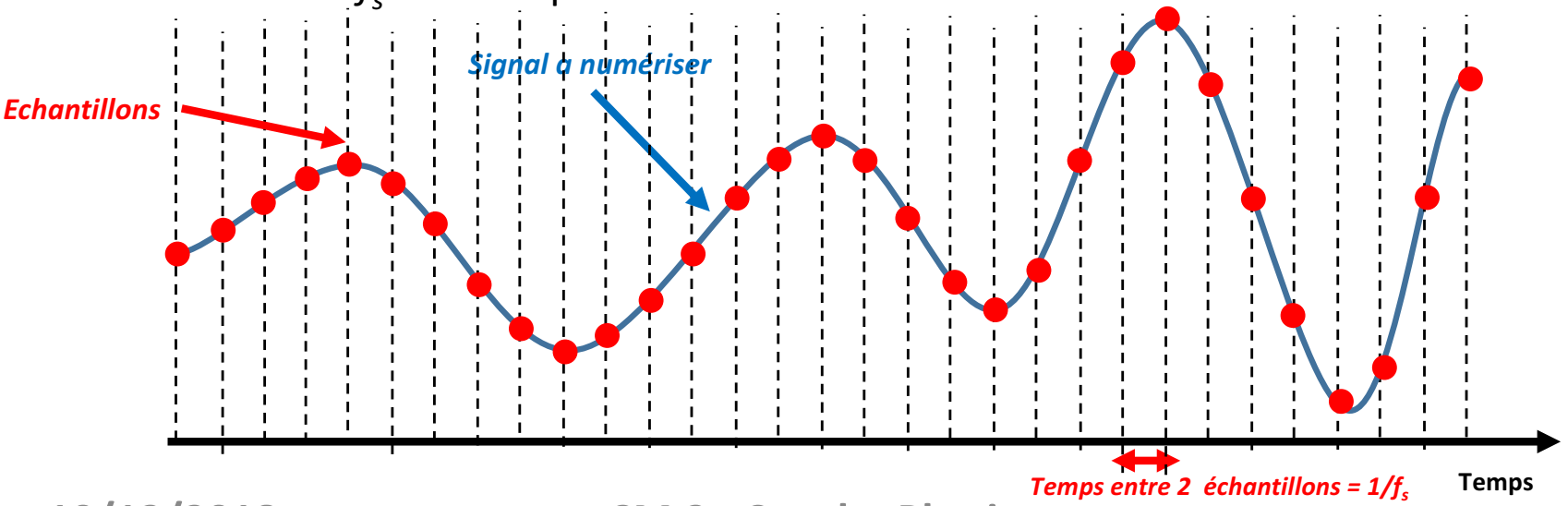
Numérisation du signal

- Bien que moins sensible aux perturbations, les signaux analogiques ont presque disparus au profit du tout numérique :
  - Signaux numériques → format de base à l'intérieur d'un ordinateur
  - Les ordinateurs sont partout maintenant
- Le processus de transformation analogique → numérique s'appelle la ***numérisation***
- Les 3 étapes de la numérisation :
  - l'échantillonnage
  - la quantification
  - le codage
- Ces 3 opérations sont sérialisées dans le temps

Numérisation du signal

❖ L'échantillonnage

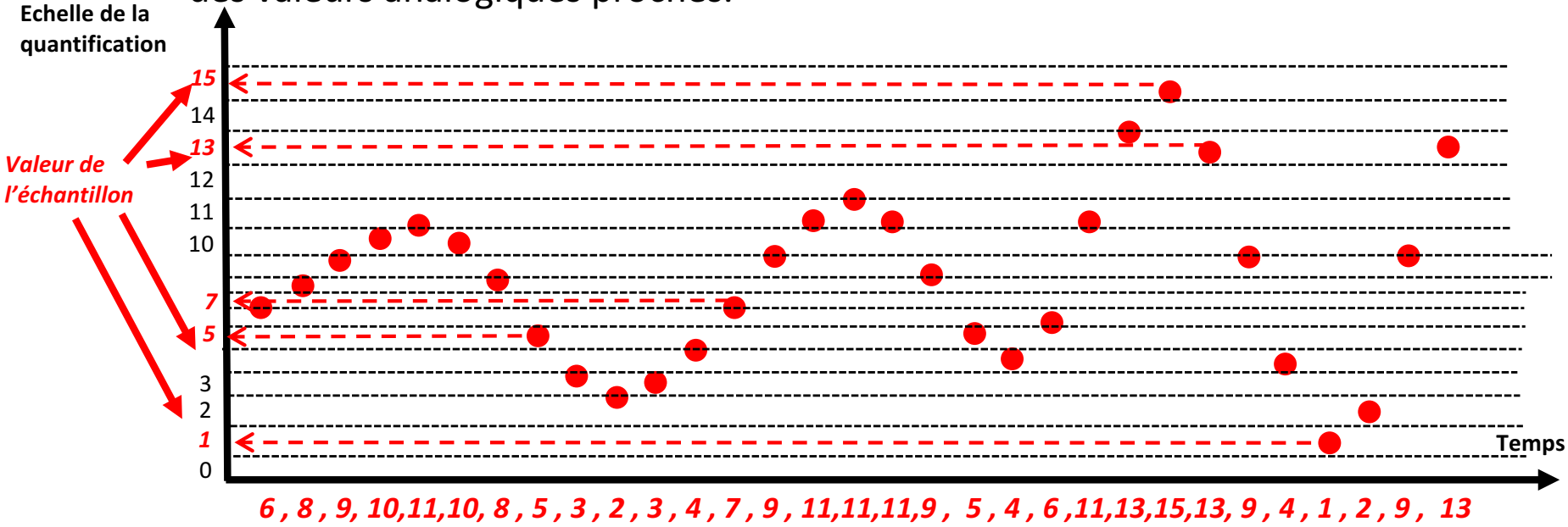
- consiste à prendre des points du signal analogique pendant son déroulement
- dépend directement de la BP. Plus elle est grande, plus il faut prendre d'échantillons par seconde
- combien d'échantillons ? : cfr. le ***théorème d'échantillonnage*** :
  - *soit un signal f(t) échantillonné à intervalle régulier. Si ce processus s'effectue à un taux supérieur au **double** de la fréquence significative la plus haute, alors les échantillons contiennent toutes les informations du signal original. En particulier, f(t) peut être reconstitué.*
- Exemple : soit un signal ayant une BP de 10.000 Hz, alors il faut l'échantillonner au moins à f_s =20.000 par seconde



Numérisation du signal

La quantification

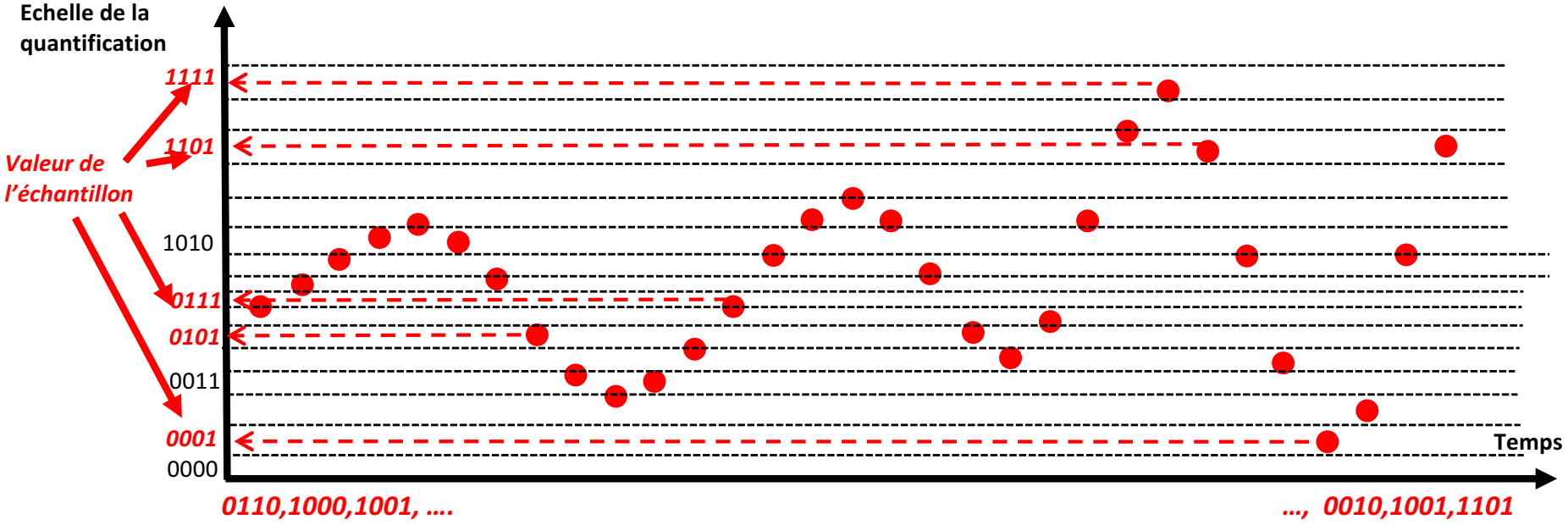
- Cette phase a pour objectif de représenter un échantillon par une valeur faisant partie d'un ensemble fini des valeurs de quantification.
- *Une loi de correspondance* défini des paliers séparant les valeurs analogiques: tous les valeurs analogiques entre deux paliers adjacents son attribuées à la même valeur de quantification.
- Cette loi ne doit pas être linéaire : la distance entre deux paliers adjacents peut être plus petite pour permettre une bonne discrimination des échantillons avec des valeurs analogiques proches.



Numérisation du signal

La codage

- Cette phase consiste à affecter une valeur numérique (*binaire*) sur base de la valeur déterminée via la loi de correspondance et selon le nombre de bits que l'on a prédéterminé
- Exemple: on utilise 4 bits pour coder les 16 niveaux précédents de quantification



Numérisation du signal

- Exemple: La voix analogique sur le support téléphonique
  - BP minimal de 3.200 Hz
  - Le théorème d'échantillonnage nous fournit une valeur de 6.400 échantillons par seconde. Cette valeur a été normalisée à 8.000
  - L'Europe a opté pour une quantification en 128 valeurs positives et autant de négatives, soit 256 valeurs. Ceci réclame 8 bits (7 aux USA)
  - Le débit binaire en Europe doit donc être égal à 8 * 8.000 = 64.000 b/s
    - NB : 8.000 éch./s = 1/8.000 = 1 éch./125 µs

- D'autres exemples des débits de signaux numérises (*sans compression*)
  - Image animées N/B : 16 Mbit/s
  - Image animées couleurs : 100 Mbit/s
  - Image télévision couleurs : 204 Mbit/s
  - Image vidéo conférence: 500Mbit/s
    - NB: en rajoutant de la compression (formats *mp3*, *mpeg*, …), on peut réduire de 10 à 50 fois ces valeurs

Débit binaire

- Pour un signal synchronisé :
  - la vitesse d'horloge donne le débit de la ligne en *bauds*, c'est-à-dire le nombre de *slots* par seconde, en sachant que on ne peut qu'émettre un symbole par slot
    - *exemple* : une ligne à 50 bauds signifie que, chaque seconde, 50 intervalles de temps (slots) se présentent.
  - Si on émet un seul bit par intervalle élémentaire (1 ou 0), débit de la ligne en *bauds* est égal au débit binaire.
  - Toutefois, rien n'empêche cependant d'utiliser 4 symboles (4 types de *signaux distincts*, qui signifieraient 0, 1, 2 ou 3). Dans ce cas, chaque symbole pourrait être codé en utilisant 2 bits (2^2=4), ce qui signifie qu'un symbole qui « pèse » 2 bits est transmis pendant le *slot*
    - *Exemple*: pour la même ligne à 50 bauds, cela revient à un débit binaire égal à 2 fois le débit de la ligne en *bauds* = 100 bps.
  - En d'autres termes, un signal a des symboles codés sur *n* bits si le nombre de niveaux transportés dans un intervalle de temps élémentaire est de 2^n
  - La *capacité de transmission* de la ligne vaut alors *n* multiplié par la vitesse exprimée en bauds

Débit binaire : perturbations

- Dans la réalité toute liaison subit des perturbations :
  - *atténuation* : consiste en l'affaiblissement homogène des différentes harmoniques du signal (pompage de l'énergie par le support dû à l'échauffement par le mouvement des électrons)
  - *bruit* : perturbations extérieures modifiant la forme initiale du signal (electromagnétisme, courant induit, ...)
  - *diaphonie* : dégradation du signal suite à la vitesse de propagation non homogène des différentes harmoniques (fréquences)

Débit binaire : perturbations

- *Formule de Nyquist* : situation idéale
  - lien entre la bande passante d'un support et son *débit binaire maximum*
    - *rappeler numérisation, fréquence d'échantillonage = 2H*
  - D_{max} (bps) = 2*H (Hz)* $\log_2$(# niveaux significatifs),
    - *H* est *la bande passante en HZ*
    - $\log_2$(# niveaux significatifs) est le nombre de bits par symbole
  - ***Si on double la BP → on double le débit binaire !!!***
  - ***Si on double le nombre de bits par symbole → on double le débit binaire !!!***
    - *Mais cela reste théorique parce que augmenter le nombre de niveaux significatifs (le nombre de bits) rend plus fragile les symboles aux perturbations réelles.*
  - *exemple* :
    - le débit binaire maximal d'une ligne avec une BP de 3.000Hz est, pour un signal ayant 4 niveaux (symboles différents), de : 2*3.000*log2(4) = 12.000bit/s
    - si le nombre de bits par symbole est de 6 : le débit max vaut alors : 36.000 bit/s

Débit binaire : perturbations

- *Shannon* : situation réelle
  - permet de déterminer la capacité maximale d'un support en tenant compte des perturbations
  - C_{max} (*bps*) = H $\log_2(1+S/N)$
    - *H* est *la bande passante en HZ*
    - S/N : rapport signal/bruit exprimé en rapport linéaire, pas en dB, p. ex, 30 dB $10^{30/10}=10^3=1000$
  - ***Si on double la BP → on double le débit binaire !!!***
  - ***Si on double le rapport S/N → on double le débit binaire !!!***
  - *Exemple:* :
    - le débit binaire maximal d'une ligne avec une BP de 3.000Hz et un rapport signal/bruit de 30 dbest de :
      - 3.000 * log2 (1001) ≅ 30.000 bits/s
      - *ceci est vrai* quel que soit le # de niveaux significatifs utilises et la fréquence d'échantillonnage
- Si on met en rapport les deux formules, on peut calculer le S/N minimal nécessaire pour supporter un certain nombre des niveaux significatifs.

Numérisation du signal

- Il découle du théorème de Shannon que la valeur du débit binaire obtenue par numérisation (Nyquist) du signal *requiert* un support physique dont la BP est « parfois » supérieure à celle nécessaire pour le transport du signal analogique
- Exemple : la voix téléphonique = 3.200 Hz.
  - On a vu avant que la numériser requiert un débit de 64.000 b/s selon Nyquist
  - Selon la formule de Shannon, sur un support ayant une BP de 3.200 Hz, le débit binaire maximal vaut : $3.200 * \log_2(1+S/B)$
  - en prenant un S/B de 10 (déjà élevé), cela ne suffit pas !!:

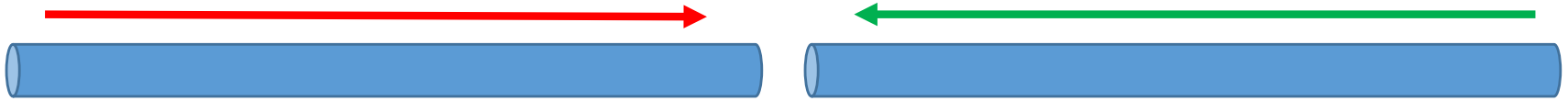
    $3.200 * \log_2(11) = \pm 11.000$ b/s < 64.000 b/s !!!
  - en prenant un S/B de 1 million, on arrive !!:

    $3.200 * \log_2(10^6) = \pm 64.000$ b/s ≈ 64.000 b/s !!!

Sens du Transfert

- Communication *simplex*
  - Un seule sens du transfert sur le même ligne
  - On a besoin de une deuxième ligne pour le sens opposé

- Communication *semi-duplex*
  - On peut transmettre dans les deux sens sur le même ligne ***mais pas au même temps***
  - Exemple: *walkie-talkie*



- Communication *full-duplex*
  - On peut transmettre dans les deux sens sur le même ligne ***au même temps***
  - Souvent, cela implique qu'on a « canal » *simplex* dédié à chaque sens de la communication

Les supports de transmission

- Supports avec guide physique
  - Câbles électriques
    - Notamment, le ***paire torsadée***
    - Historiquement, aussi le câble coaxial → *de moins en moins utilisé*
  - Fibres optiques

- Supports sans guide physique
  - Ondes radios
  - Ondes lumineuses

Les supports de transmission

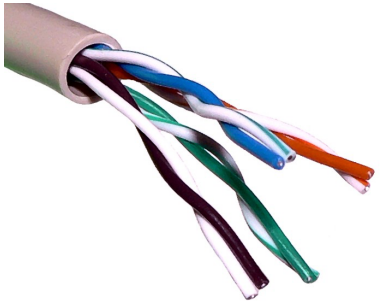
❖ Paire torsadée

- câble électrique à 4 paires
- chaque paire est formée de deux fils conducteurs enroulés en hélice l'un autour de l'autre
  - Ça a pour but principal de limiter la sensibilité aux interférences et la diaphonie

- Les quatre paires pas toujours toutes nécessaires:
  - on peut que utiliser que deux ou un
- connecteur : RJ45
- supporte jusqu'aux plusieurs Gbps:
  - Catégorie 5: 100 Mbps, 1 Gpbs sur Ethernet
  - Catégorie 6-7: 10Gbps sur Ethernet
  - ***Catégorie 8: 40Gbps sur Ethernet → en déploiement***
- exemple : câble du PC

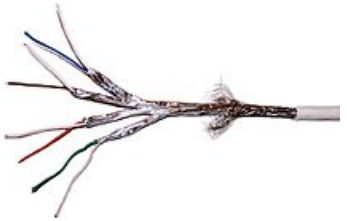
Les supports de transmission

- Il existe plusieurs types de paires torsadées :
  - **Paire torsadée non blindée** : *Unshielded twisted pair* (UTP



  - **Paire torsadée écrantée** : *Foiled twisted pair* (FTP):
    - L'ensemble des paires a un blindage global assuré par une feuille d'aluminium

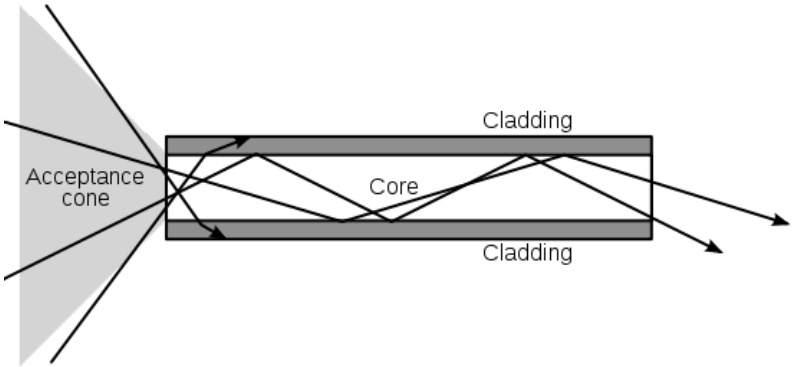
  - **Paire torsadée blindée** : *Shielded twisted pair* (STP).
    - Chaque paire est entourée d'un feuillard en aluminium

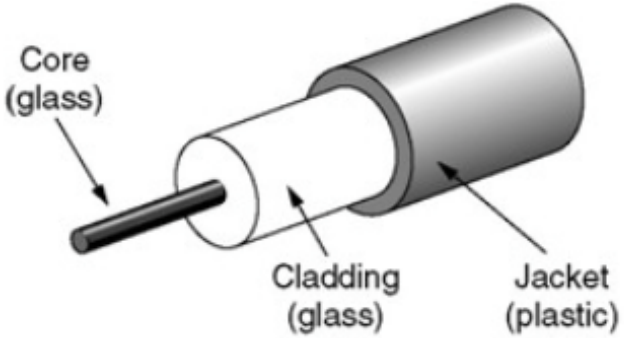


Les supports de transmission

❖ La fibre optique :

- Fibre de verre portant des impulsions lumineuses, chaque impulsion un bit.



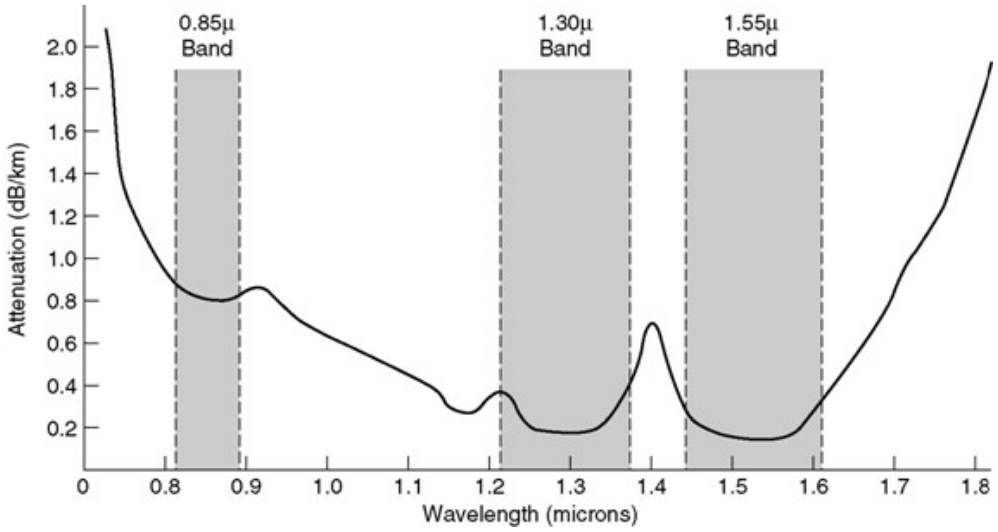


- Extrêmement bien adapté au multiplexage fréquentielle
  - Une seule fréquence (canal) peut avoir un débit binaire ente 1-100 Gbps
- 2 types de FOL
  - monomode : 1 seul rayon de lumière
  - multi mode : plusieurs rayons de lumière

Les supports de transmission

❖ La fibre optique :

- Intérêt Nº1 : Largeur de la bande passante → plus de débit binaire
  - Record 2018 Hong-Kong –LA : 100000 Gbps (100 canaux de 100 Gbps)
  - Des centaines ou des milliers de gigahertz de large!!
- Intérêt Nº2 : Moins d'atténuation avec la distance → moins des répéteurs
  - *Petit rappel atténuation: nombre des dBs de puissance qu'on perd par mètre*
- Intérêt Nº3 : C'est de la lumière → immunité contre le bruit électromagnétique !!!



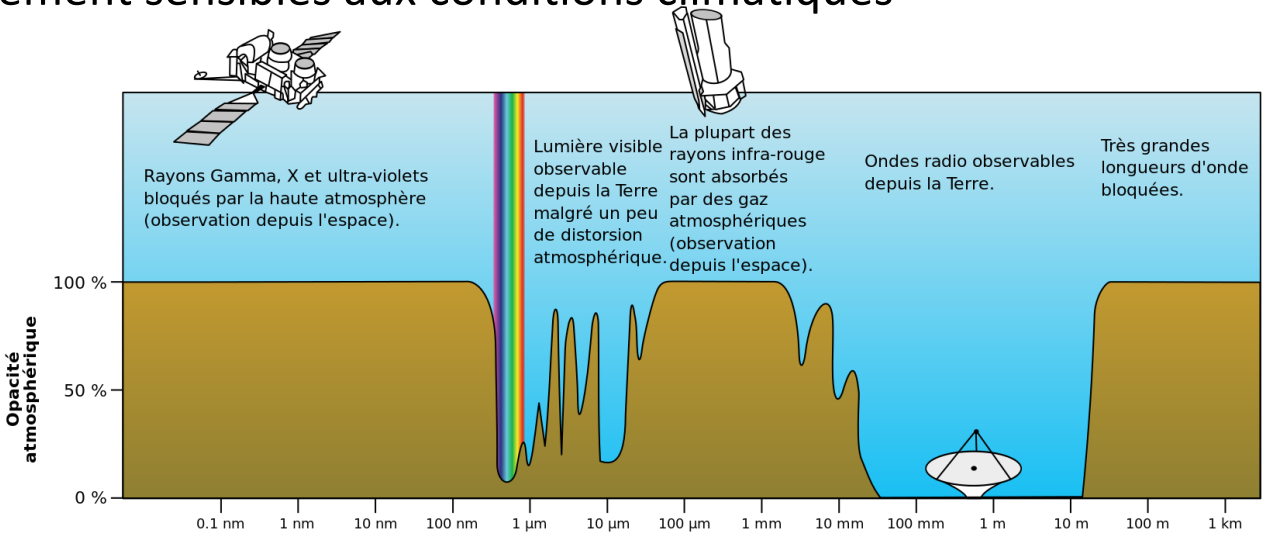
Pour la lumière, la vitesse *v = c*
Donc
λ = c T → λ = c / f → f = c / λ
ou
Longueur d'onde: λ m
Période: T s
Fréquence: F = 1/T (periodes /s = Hz)

$$\underline{f = c / \lambda}$$

Donc, des microns, deviennent des centaines ou des milliers de gigahertz !!!

Les supports de transmission

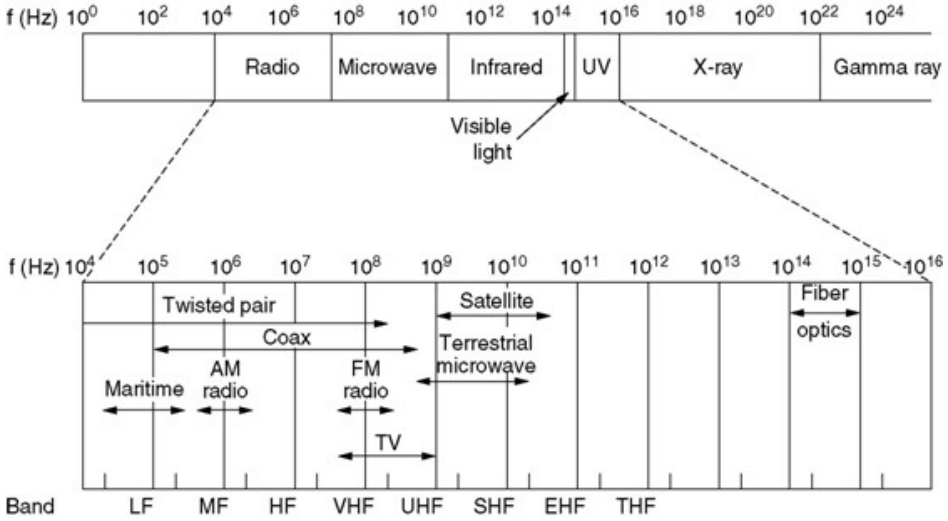
- **Des onde radioélectriques**
  - signal transporté dans le spectre électromagnétique
  - pas de "fil" physique
  - Effets due à l'environnement de propagation:
    - réflexion
    - obstruction par des objets
    - Interférences
  - Egalement sensibles aux conditions climatiques



Les supports de transmission

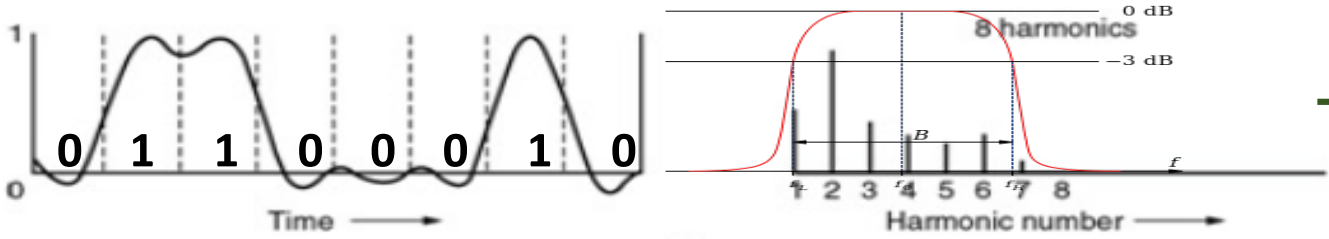
❖ Des ondes radioélectriques

- types de liaison radio:
  - micro-onde terrestre: par exemple. canaux jusqu'à 45 Mbps
  - LAN (WiFi) 11 Mbps (IEEE 802.11b), 54 Mbps (IEEE 802.11a), 1.3 Gbps (IEEE 802.11ac)
  - Téléphonie cellulaire 3G: ~ peu de Mbps, 4G centaines de Mbps
  - Satellite Canal
    - Dès quelques Kbps à 45 Mbps (ou plusieurs canaux plus petits)
    - Retard de 270 ms

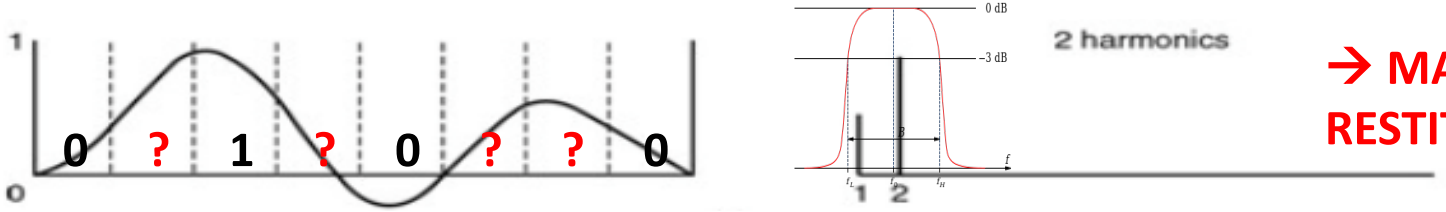


Les supports de transmission

- Le **spectre du signal** à transmettre doit être compris ***dans la bande passante*** du support physique
- Transmission d'un signal à spectre étroit sur un support à large bande passante → ***mauvaise utilisation du support***



→ OK !!!



→ MAUVAISE RESTITUTION !!!

- ***Solution:*** recours aux deux techniques pour pallier ce problème:
  - **Modulation**
  - **Multiplexage**

Multiplexage

- **Multiplexage :** *possibilité de faire cohabiter sur le même support physique plusieurs communications indépendantes*
  - une seule voie est décomposé en des plus petites voies (canaux de transmission)
  - chaque petite voie (canal) est attribuée à un seul utilisateur
  - souvent moins onéreux de faire transiter en même temps des données de plusieurs personnes et d'éviter ainsi qu'elles ne possèdent chacune toute l'infrastructure
  - optimise l'usage des canaux de transmission
    - La bande passante du canal est plus proches des besoins d'un utilisateur individuel
  - ***Exemple: LTE (4G)***
    - BP totale pour tous les utilisateurs: ~ 9 MHz
    - On fait de petits canaux de 15 kHz → On pet mettre jusqu'à 600 clients !!!

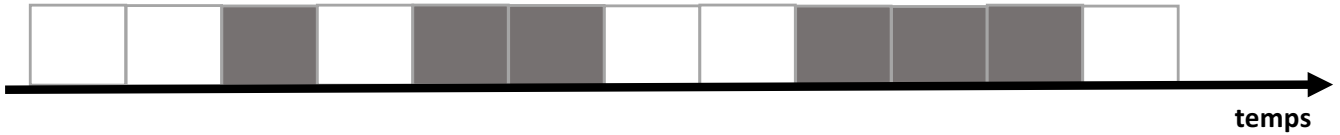
Multiplexage

- **Le multiplexeur :**
  - reçoit donc un ensemble de voies (lignes) individuelles, dites à *basse vitesse*
  - les transmet toutes ensemble sur une liaison unique, la voie dite à *haute vitesse*
- **A l'autre extrémité** :
  - un ***démultiplexeur*** effectue l'opération inverse, la séparation des voies basse vitesse
- L'équipement faisant le *multiplexage/démultiplexage* s'appelle un MUX
- Plusieurs types de multiplexage :
  - multiplexage temporel (TDM)
  - multiplexage fréquentiel (FDM)
  - multiplexage en longueur d'onde (WDM)

Multiplexage

- **Multiplexage temporel (*TDM: Time Division Multiplexing*)**
- Consiste à affecter à chaque utilisateur un quantum de temps (*slot*) pendant lequel il disposera de l'*intégralité* du débit binaire
- Différents types existent :
  - asynchrone :
    - laps de temps variable, pas synchronisé sur les débuts des slots
    - quantum de temps est *alloué sur demande*
    - utilise mieux la BP car pas (moins) de temps mort

  - synchrone :
    - laps de temps en permanence identique
    - quantum de temps est *alloué à chaque utilisateur*

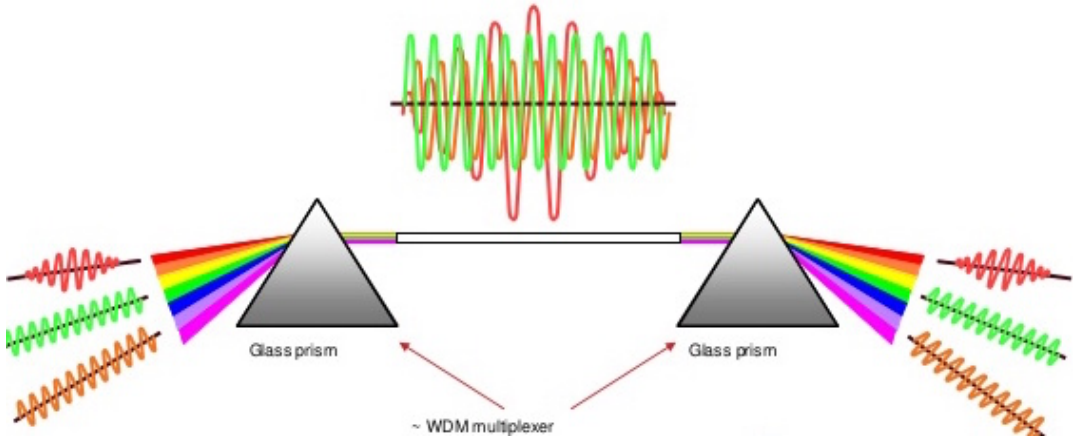


Multiplexage

- **Multiplexage fréquentiel (*FDM: Frequency Division Multiplexing*)**
  - Consiste à partager une bande passante de *X* Hz en *n* canaux de *X/n* Hz chacun
  - Les canaux sont *non jointifs* (bande de garde)
  - Tout le monde possède en *permanence* une *partie* de la bande passante
  - Notion de transmission en bande large :
    - plusieurs transmissions indépendantes et simultanées
  - Exemple : la radio

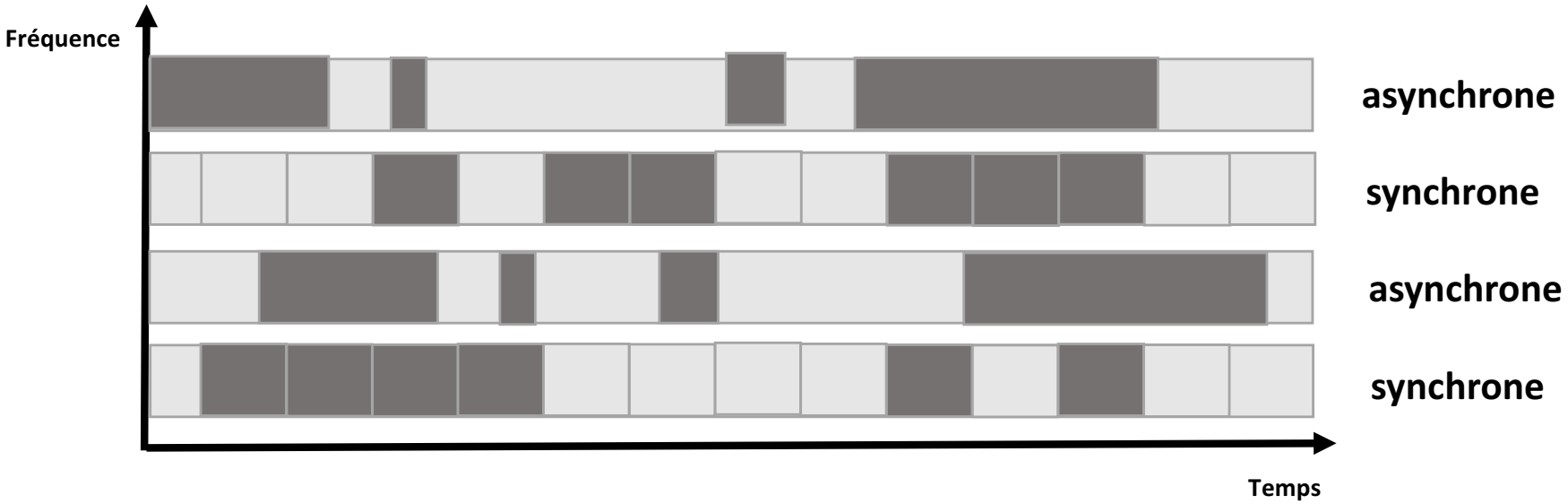
Multiplexage

- **Multiplex. en longueur d'onde (*WDM: Wavelength Division Multipl.*)**
  - Technique utilisée avec la fibre optique :
    - décomposition du spectre lumineux (ensemble de longueurs d'onde)
    - décomposition avec du matériel passif (prisme) qui n'engendre donc pas d'interference (différence avec le matériel actif de la FDM)
    - recomposition selon principe identique
    - utilisation de prismes pour multiplexer les flux lumineux en un seul, puis utilisation d'un second prisme pour décomposer ce même flux lumineux



Multiplexage

- **Combinaison des multiplexages**
  - Multiplexage fréquentiel avec multiplexage temporel intègré
  - Transmissions asynchrones, synchrones possibles en parallèle

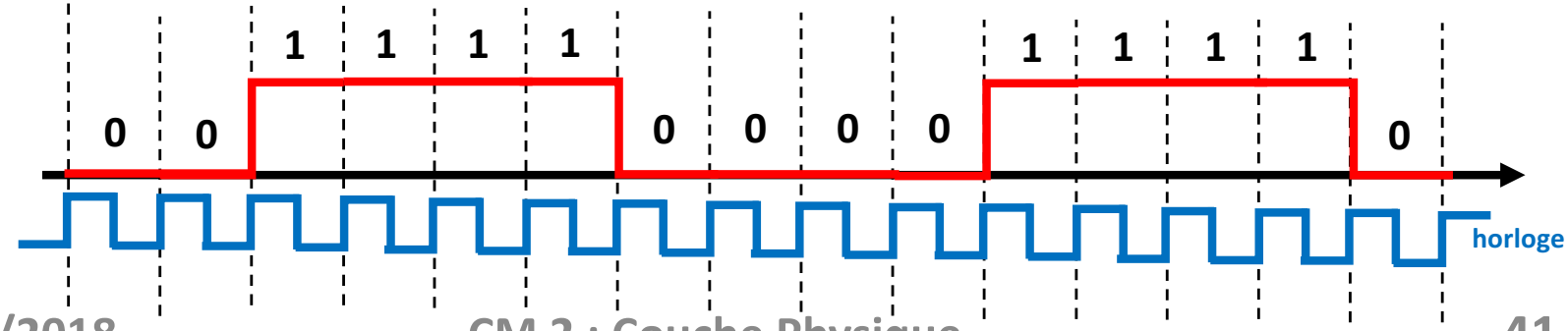


Codage et transmission en bande de base

- **Technique la plus simple** :
  - **Codage en ligne**: info transmise par des changements discrets dans les signaux représentant l'information binaire
    - Normalement, des variations d'un courant électrique (sous la forme de changements dans la tension électrique) qui reflètent les caractères sous forme de créneaux. dans la tension
  - **Transmission en bande de base:** on envoie le signal électrique ainsi codé directement sur la ligne (le support de transmission)
  - une seule communication à la fois sur le support
  - pas de multiplexage
  - ex. : LAN Ethernet
- **Défi** :
  - comment un émetteur peut-il envoyer un signal que le récepteur pourra interpréter CORRECTEMENT comme étant un 1 ou un 0 ?

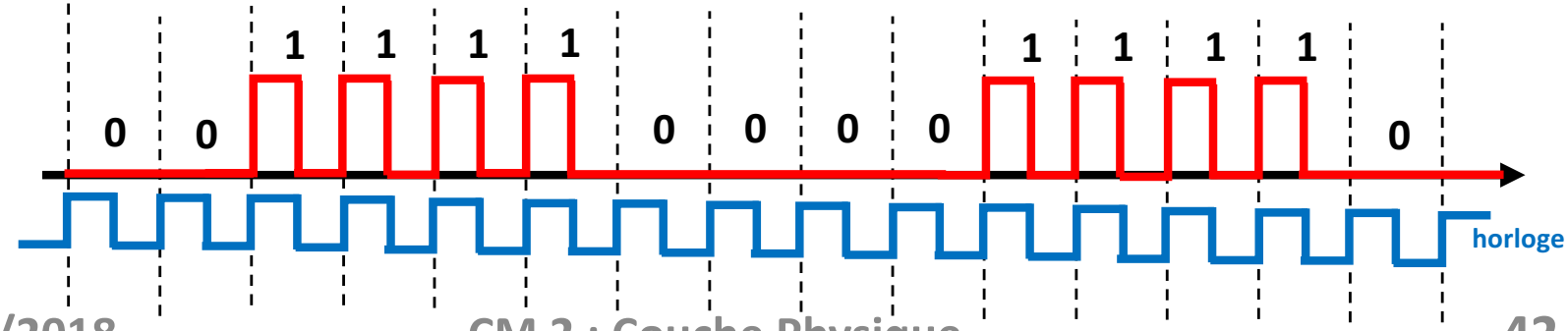
Transmission en bande de base

- ***Code unipolaire « tout ou rien »***
  - Courant positif pour un bit à 1
  - Courant nul (absence de tension) pour un bit à 0
  - Deux versions :
    - *NRZ (non-return-to-zero) :* Pas de retour à tension nulle après chaque pulse (*voir ci-dessous*)
    - *RZ (return-to-zero):* Retour à tension nulle après chaque pulse (prochaine slide)
- Difficultés :
  - Présence de courant en continu, CC (*DC, direct current*): la moyenne du signal n'est pas zéro
    - Des lignes « ont » des circuits électroniques qui filtrent las basses fréquences (CC)
    - Donc, (i) dégradation de la forme du signal (flancs), et (ii) pertes de puissances importantes.
  - Synchronisation difficile
    - Longue suite des 1 → Où commence mon signal ?
    - Longe suite des 0 → Mon signal est mort ou c'est des zéros ?
    - Besoin de synchronisation avec un signal d'horloge



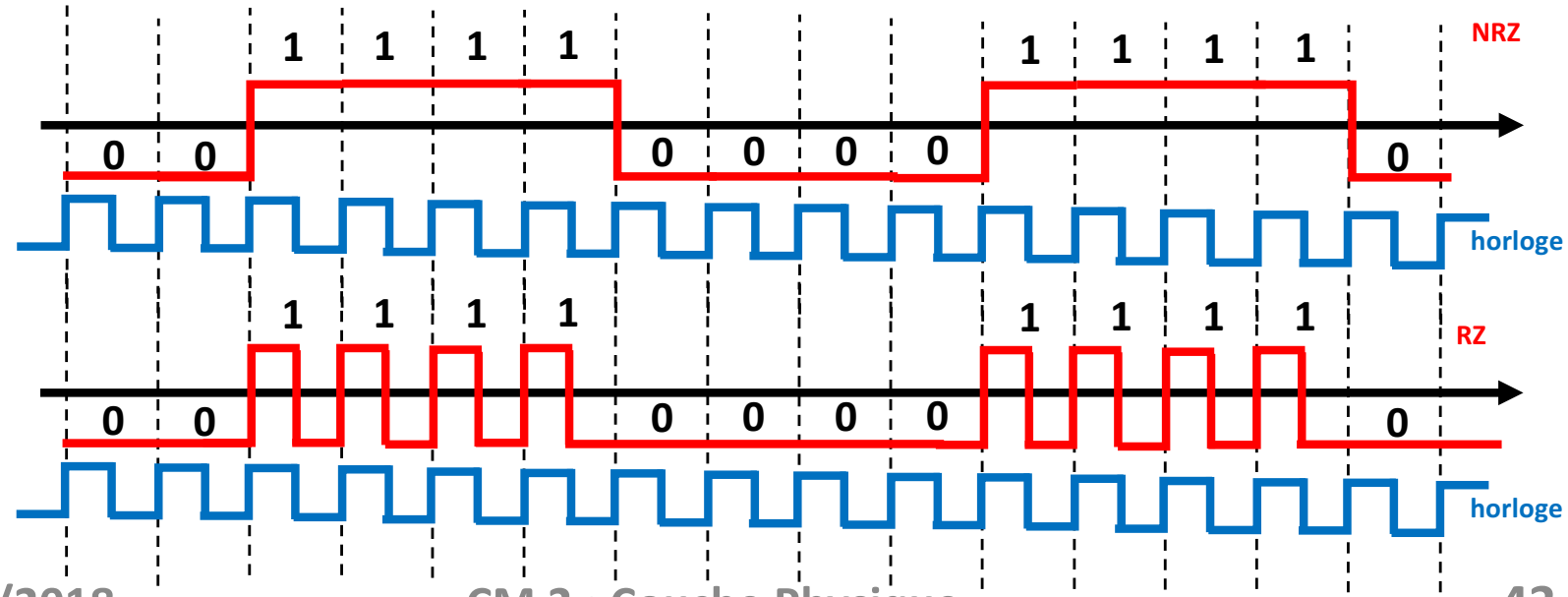
Transmission en bande de base

- ***Code unipolaire « tout ou rien »***
  - **Version « Retour-à-Zéro (*RZ, Return-to-Zero)* »**
    - Le signal retourne à la valeur zéro (tension nulle) après chaque pulse, typiquement au milieu de la période de l'horloge.
    - **Synchronisation plus facile** → Une longue séquence de bits « 1 » permet de récupérer le signal de horloge.
    - **Courant en continu** : Bien qu'on a réduit à moitie le temps en haut (bit à 1), il y a toujours une composante continue (DC) lorsqu'il y est une longue séquence de "1"
    - Elle consomme le **double de la bande passante** qui permet d'atteindre le même débit comparé au format *Non Return to Zero (NRZ)* puisque on a DEUX transitions par période de l'horloge → *Souvenez vous de Nyquist*



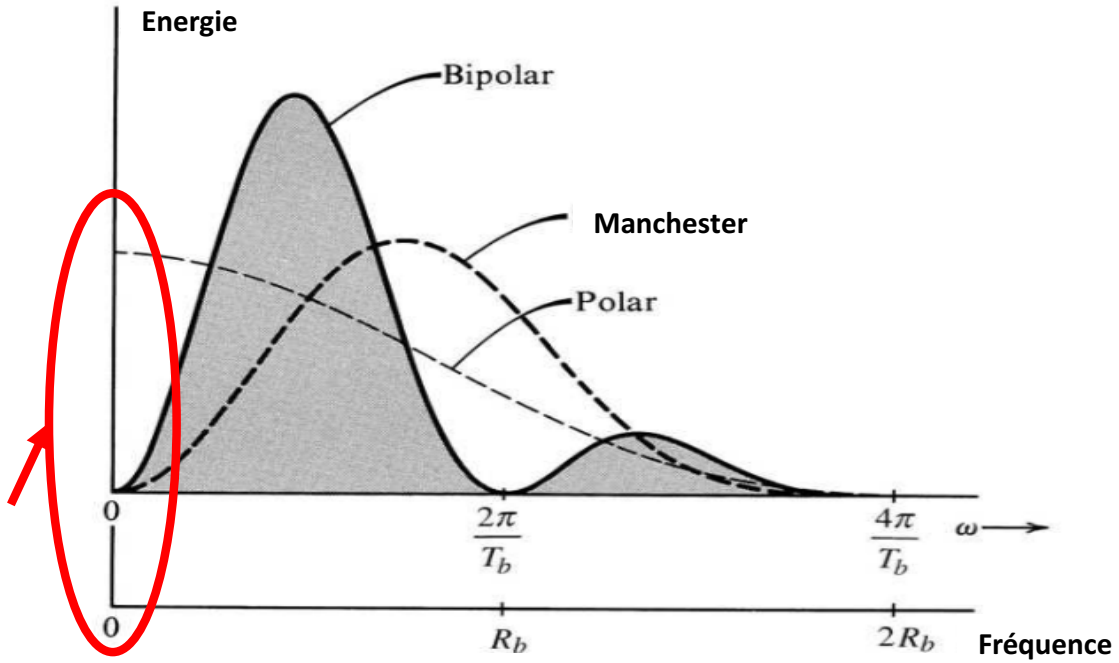
Transmission en bande de base

- ***Code polaire***
  - Une tension positive (p. ex. +5V) = bit à 1
  - une tension négative (p. ex. -5V) = bit à 0
  - Absence de tension (courant nul) = absence de communication
  - Deux versions : *NRZ (non-return-to-zero) et RZ (return-to-zero)*
- Difficultés :
  - **Courant en continu** : Moyenne du signal plus proche des zéro, donc composante DC affaiblie, mais encore présent pour des longue séquence de bits "0" ou de "1"
  - **NRZ vs RZ** : Synchronisation de RZ (horloge) plus facile mais double de la bande passante que NRZ



Transmission en bande de base

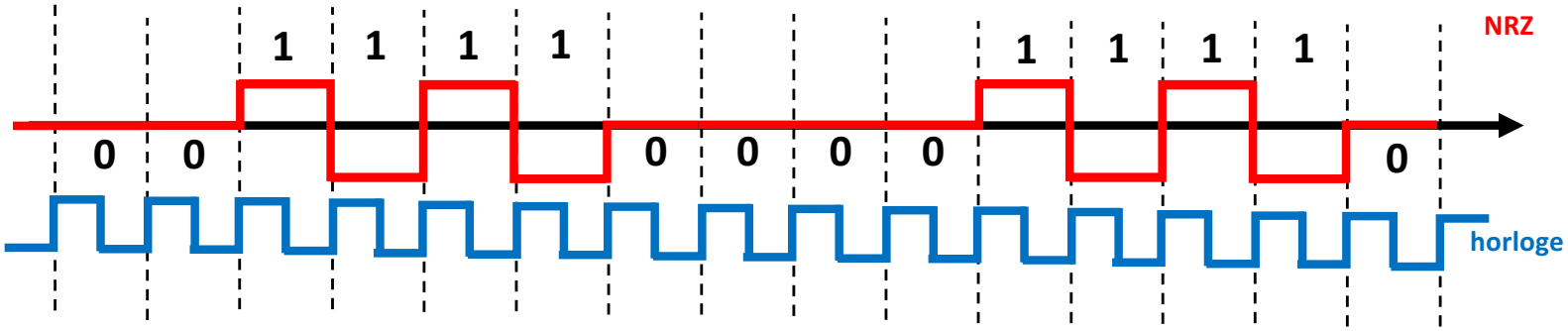
- Les codes de ligne avec une valeur DC non nulle diminuent les performances, car un composant DC sans information réchauffe les fils (perte d'eneergie).
- Il y a deux alternatives:
  - Utilisez des séquences d'impulsions dont les valeurs moyennes vont jusqu'à zéro (**code bipolaire**)
  - Utilisez des impulsions ayant une valeur moyenne nulle (**code biphase ou Manchester**)



Composante DC = Energie dans la fréq. 0

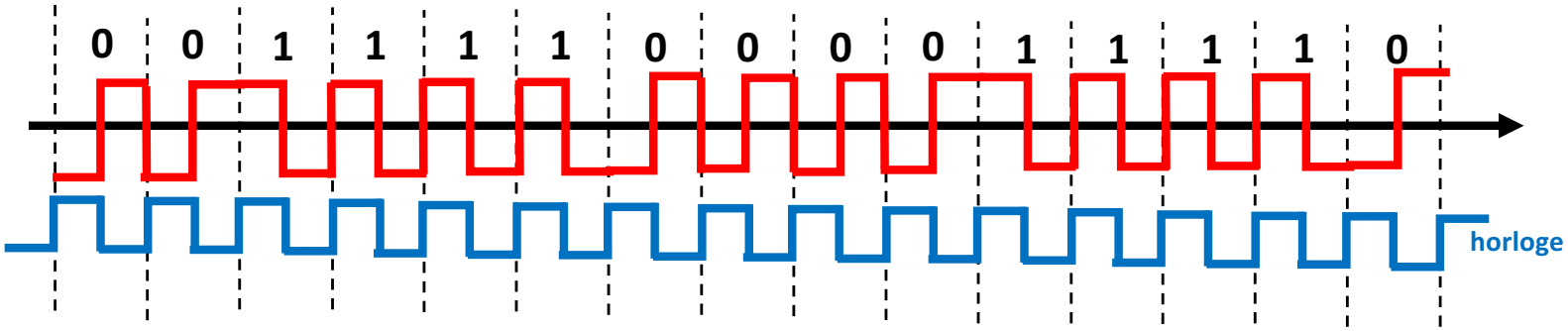
Transmission en bande de base

- ***Code bipolaire***
  - 3 niveaux des tension: **+, -** and **0**. Tel signal est un signal ***duo-binaire:***
  - Un exemple est ***Alternate Mark Inversion (AMI)*** :
    - Un bit à 1 est représenté alternativement par une tension positive (p. ex. +5V) ou négative (p. ex.-5 V).
    - Une tension nulle = bit à 0
  - Deux versions : *NRZ (non-return-to-zero, voire ci-dessous) et RZ (return-to-zero)*
- Avantages:
  - **Pas de courant en continu** : Moyenne du signal presque zéro grâce à l'alternance +/- pour le bit 1
  - **Bande passante réduite** par rapport au code polaire et unipolaire : un cycle complet de transitions requiert au moins 2 périodes d'horloge, a la place d'un transition par période d'horloge
  - Fournit de la **détection d'erreur** : chaque erreur binaire provoque une violation de l'alternance
  - Récupération de horloge simple avec des longues suites de bits « 1 », pas si simple avec suites de bits « 0 »



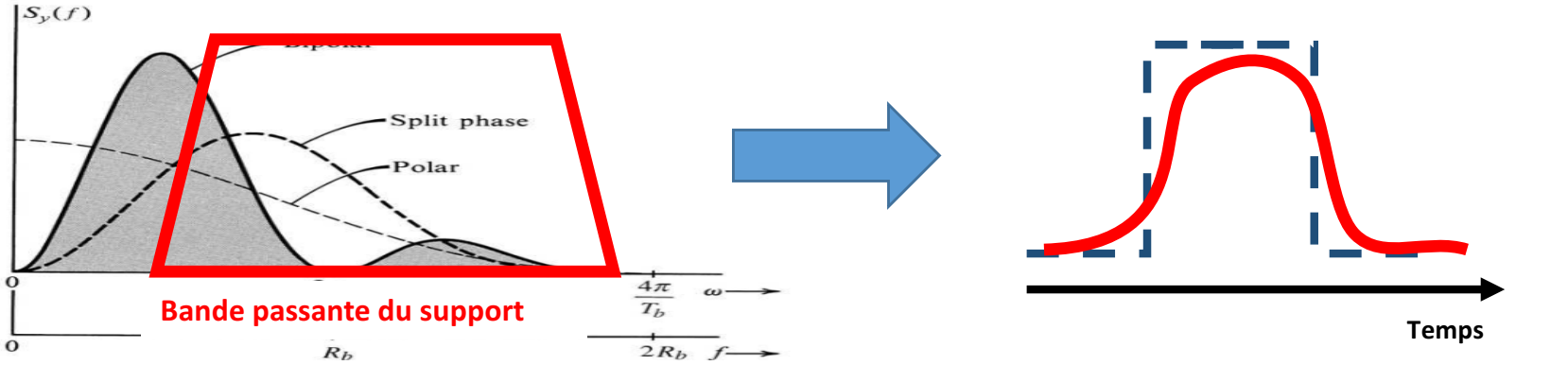
Transmission en bande de base

- ***Code Manchester***
  - un bit à 1 est représenté par une transition positif → négatif au milieu de l'intervalle
  - un bit à 0 est représenté par une transition négatif → positif au milieu de l'intervalle
- Avantages:
  - **Pas de courant en continu** : Moyenne du signal zéro car chaque bit a une partie + et –
  - **Synchronisation effective :** Dans tous les intervalles il y a une transition → Horloge
- Inconvénients :
  - Plus de bande passante que bipolaire
- Utilisé dans les premiers versions de *Ethernet*



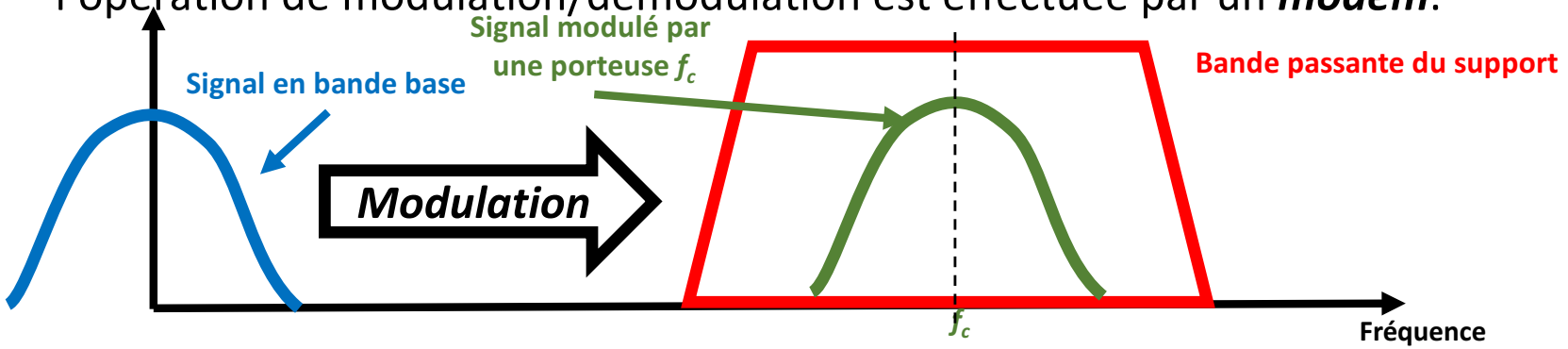
Transmission en bande de base

- **Dégradation très rapide du signal**
  - Principal problème de la bande de base
  - Les supports physiques ne laissent passer les basses fréquences: ils absorbent leur énergie
  - Malgré les codages les plus avancés, il y a de l'énergie dans des fréq. basses
  - On perd de de l'énergie en fonction de la distance (max 5 Km.) :
    - plus on avance, plus on perd
    - nécessité de le régénérer de temps en temps (répéteurs de signal)
  - Détérioration des fronts montants et descendants: car on a perdu une partie du signal dans les fréquences filtrées
- **Solution à la dégradation en bande base**
  - ***Transmission par modulation*** d'une onde porteuse



Transmission par modulation

- ***Transmission par modulation*** d'une onde porteuse
  - pour des distances plus longues, on utilise ***un signal sinusoïdal (l'onde porteuse),*** moins sensible: même affaibli, il peut être plus facilement décodé
  - *moduler* c'est « changer » une propriété de la porteuse en suivant les changements du *signal original à transmettre*
    - *Donc,* seule la *modulation* a une *signification*
  - le spectre des signaux modulés est *centré* sur la fréquence de la *porteuse*
  - la porteuse transporte les signaux dans la bande passante du support
    - normalement, cela veut dire sur des fréquences plus hautes où le support ne filtre pas de fréq.
  - l'opération de modulation/démodulation est effectuée par un ***modem***.

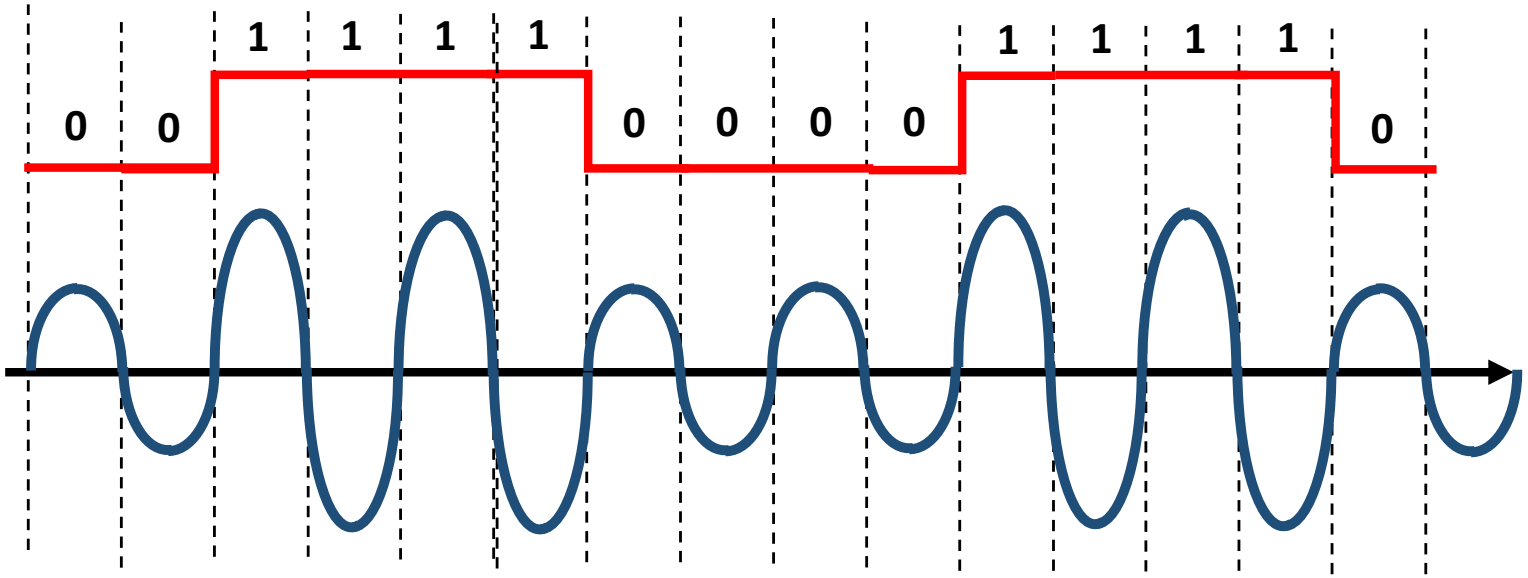


Transmission par modulation

- Trois grandes caractéristiques à « moduler »:
  - amplitude, fréquence, phase
- C'est selon ces caractéristiques que nous pouvons adapter (*moduler*) le signal afin de lui faire porter *l'information numérique* (modulations numériques) :
  - modulation d'amplitude : *Amplitude-shift keying (ASK)*
  - modulation de phase : *Phase-shift keying (PSK)*
  - modulation de fréquence : *Frequency-shift keying (FSK)*
- Afin d'effectuer la modulation :
  - un nouveau type de dispositif : modem = MOdulateur/DEModulateur
  - reçoit le signal en bande de base
  - le module (donne une forme sinusoïdale) en utilisant la porteuse
  - cela fait monter la signal en fréquence pour s'adapter à la BP.

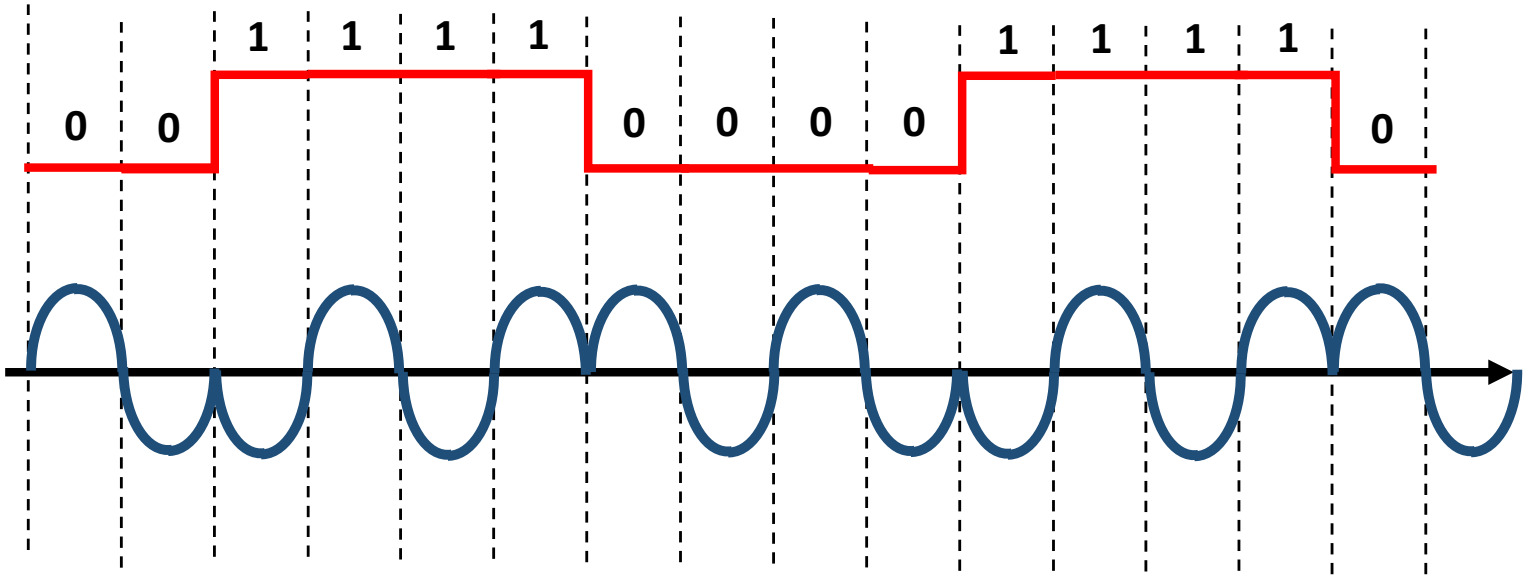
Transmission par modulation

- **Modulation numérique d'amplitude (*Amplitude-shift keying, ASK*)**
  - la différenciation des bits à 0 ou à 1 se fait par une modification de l'amplitude de la sinusoïde
- ***Pulse-amplitude modulation (PAM)***
  - Généralisation de ASK à plusieurs nivéaux discrets (paliers) au delà de deux
  - Exemple PAM-16: on a 16 paliers (chaque palier est codé sur 4 bits)
  - Largement utilisé par les versions récentes d'***Ethernet***



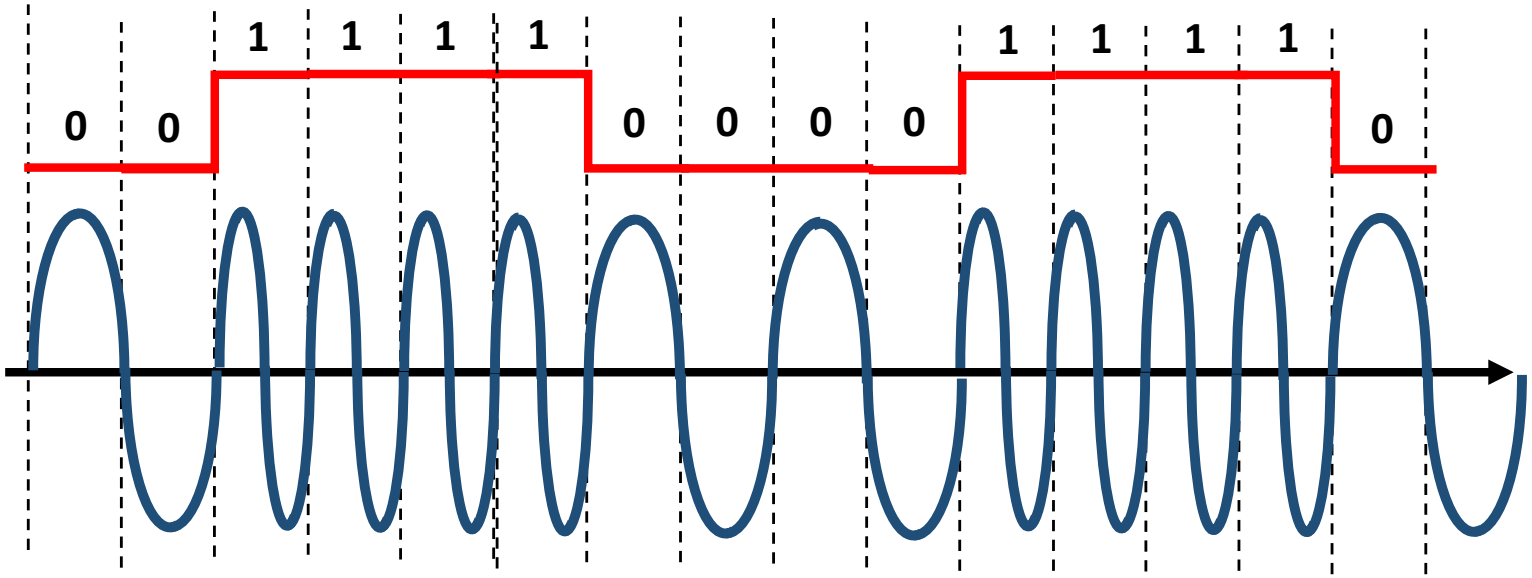
Transmission par modulation

- **Modulation numérique de phase (*Phase-shift keying, PSK*)**
  - la différenciation des bits à 0 ou à 1 se fait par un signal qui commence à des emplacements différents de la sinusoïde



Transmission par modulation

- **Modulation numérique de fréquence (*Frequency-shift keying, FSK*)**
  - la différenciation des bits à 0 ou à 1 se fait par par un signal qui possède une fréquence tantôt basse, tantôt élevée



Détection et correction d'erreurs

- Longtemps la détection et la correction d'erreurs relevaient de la couche de lien (niveau 2)
- En effet, la qualité des lignes physiques était très insuffisante pour obtenir des taux d'erreurs acceptables dans les trames de niveau 2
- La situation actuelle a très sensiblement changé pour deux raisons essentielles :
  - La *fiabilité* des lignes s'est beaucoup *accrue*. Le taux d'erreur est $< 10^{-9}$ et souvent moins encore. Ceci a été rendu possible par :
    - des techniques de codages plus efficaces
    - de nouveaux supports physiques filaires : fibre optique, ...
  - La *nature des applications* (*multimédia*, ...). Certaines ne tolèrent pas les pertes de temps associées aux reprises sur erreurs. La *perte* de quelques bits *ne change rien* quant à la qualité que l'oeil ou l'oreille peuvent percevoir. Des délais seraient, eux, beaucoup plus détectables

Détection et correction d'erreurs

- La détection et la correction restent toutefois indispensables :
  - sur des supports de mauvaise qualité (des ondes radioélectriques, …)
  - pour des applications ne tolérant pas la moindre erreur (transfert de fichier, ...)
- Pour la détection/correction, deux grandes possibilités existent :
  1. Toujours envoyer *avec* des bits redondantes pour *détecter et corriger*
  2. Toujours envoyer *avec* des bits redondantes pour *détecter,* après retransmettre si erreur détectée
- Quel système choisir ?
  - la *détection/correction* exige un *accroissement* d'environ *50%* de l'information transportée. Ainsi, pour envoyer 1.000 bits utiles en toute sécurité, il faut envoyer 1.500 bits
  - la *détection seule* se contente d'une augmentation de *16 à 32 bits*. Ce n'est qu'en cas d'erreurs avérées, que la retransmission intégrale de l'information doit être faite

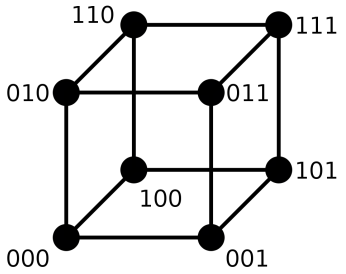
Détection et correction d'erreurs

- Le *point neutre* pour des trains de bits compris entre 1.000 et 10.000 bits est donc d'environ un taux d'erreurs de 10^{-4} :
  - si le taux d'erreurs est plus faible, une détection est utilisée
  - si le taux d'erreurs est plus élevé, la correction est plus intéressante
- Or, les supports ont quasiment tous un taux d'erreurs < 10^{-4} ➔ ***La détection avec retransmission est donc majoritairement utilisée***
- La détection / correction d'erreurs suppose :
  - l'utilisation d'algorithmes complexes (temps de traitement non négligeable)
  - une augmentation importante de l'information à transporter (redondance)
- Tout d'abord, le système trivial :
  - bien que n'offrant que peu de garantie, une méthode consiste à envoyer 3 fois la même information et à choisir la plus probable :
    - *envoi 3 bits à 1, réception 2 bits à 1 et 1 bit à 0*
    - *P(bonne valeur =1) = 2/3 , P(bonne valeur =0) = 1/3*
    - ***On choisit 1 comme bonne valeur correcte***

Détection et correction d'erreurs

❖ Notion de code de *Hamming* :

- Pour pouvoir corriger les erreurs, il faut pouvoir distinguer les différents caractères émis, même s'ils sont erronés
- supposons un mini alphabet de 4 caractères : *00, 01, 10, 11*
- si une erreur se produit, le caractère est 'transformé' en un autre caractère (valide) de ce même alphabet et l'erreur passe inaperçue :
  - envoi de 00 et réception de 10. Le caractère reçu bien qu'erroné fait partie de l'alphabet
- Il nous faut donc *ajouter* de l'information :
  - 00 → **00***000*
  - 01 → **01***111*
  - 10 → **10***110*
  - 11 → **11***001*



- Si ***une erreur*** se produit, on compare la donnée transmise avec les caractères valides de l'alphabet. On en déduit le bon en faisant le rapprochement avec celui qui ressemble le plus :
  - Si 10000 reçu, le plus proche est **00000** (distance +1) car les autre possibilités **10110** et **11001** (distance +2) sont plus éloignées.

Détection et correction d'erreurs

❖ Notion de code de *Hamming* :

- Si ***deux* erreurs** se produisent sur le même caractère, il devient impossible, dans le contexte ci-dessus, de récupérer la valeur exacte.
- Exemple : Emission de *10110* et réception de **10101**. L'estimation du caractère correct n'est plus possible. Il y a deux options (l'une est la bonne) à la distance la plus petite(2):
  - ***10110*** → la bonne à distance 2
  - **11001** → mais celle-ci est encore un caractère valide à distance 2

- Soit *d(x,y)* la distance entre deux caractères *x* et *y* de même longueur definie comme le nombre de bits (positions) différents, on définit la distance de *Hamming* comme étant :

$$d_H = inf\ d(x,y)$$

où la borne inferieure s'applique à l'ensemble des caractères valides de l'alphabet

- c.-à-d., la distance la plus petite entre deux *caractères valides de l'alphabet*

Détection et correction d'erreurs

- **Notion de code de *Hamming* :**
  - dans l'exemple ci-dessus, le calcul de d_H donne 3
  - pour pouvoir corriger *une seule erreur*, il faut que les différents caractères valides du même alphabet satisfasse à $d_H = 3$, de sorte que, en cas d'erreur, la distance entre le caractère correct et le caractère erroné soit de 1
    - *Encore n'importe quel autre caractère valide est à distance 2 par définition*
  - Pour corriger *2 erreurs à la fois*, la d_H doit valoir 5 :
    - le nouvel alphabet vaut alors :
      - 00 → *00000000*
      - 01 → *01111011*
      - 10 → *10110101*
      - 11 → *11001110*
  - En recevant le caractère *10001010*, on en déduit que le caractère correct est *11001110* puisque c'est le seul caractère de notre alphabet à avoir une distance de *Hamming* de 2 avec le caractère erroné reçu :
    - d(10001010,11001110) = 2
    - d(10001010,x) > 2 si x ≠ 11001110

Détection et correction d'erreurs

- De nombreuses méthodes existent :
  - parmi elles, les bits de parité, que l'on peut déterminer à partir d'un caractère (souvent un octet)
  - le bit de parité est un bit supplémentaire ajouté au caractère « protégé »
  - il est calculé en sorte que la *somme des éléments binaires (nombre de bits à 1) modulo 2* soit égale à 0 ou à 1
    - Dans le cas **paire**, le bit de **parité** est 1 si le nombre de bits à 1 est **impaire**.
    - Dans le cas **impaire**, le bit de **parité** est 1 si le nombre de bits à 1 est **paire**.
- Exemple : soit le caractère 10011001, on pose un *parité paire*
  - il faut donc ajouter un bit valant 0 : 10011001 **0**
  - si une erreur se produit lors de la transmission :
    - 10111001 **0**
    - alors, le nombre de bits à 1 n'étant pas pair, on détecte qu'une erreur s'est produite
  - problème :
    - Il faut ajouter un bit tous les 8 bits (12,5 % de charge)
    - deux erreurs sur le même octet ne sont pas détectables :
    - 01011001 0 passera inaperçu !!!!

Détection et correction d'erreurs

- Une autre méthode, plus efficace, repose sur une division de polynômes :
  - les deux parties (émetteur, récepteur) se mettent d'accord sur un polynôme (ex. de degré 16 : $X^{16}+X^8+X^7+1$) : le *générateur*
  - à partir des éléments binaires de la trame notés a_i, i = 0 → M-1, M étant le nombre de bits de la trame, on constitue le polynôme de degré M – 1 :
    - $P(x) = a_0 + a_1x + ... + a_{M-1}x^{M-1}$
- Fonctionnement :
  - l'émetteur divise le polynôme issu de sa trame, par le générateur
  - le reste de cette division est un polynôme dont le degré vaut au max. 15 :
    - $R(x) = r_0+r_1x+...+r_{15}x^{15}$
  - les valeurs binaires r_0, r_1, ..., r_{15} sont placées par l'émetteur dans la partie de contrôle de la trame
  - à l'arrivée, le récepteur effectue le même travail
  - il compare le reste qu'il a calculé avec celui se trouvant dans la partie de détection
  - si les deux restes sont identiques, alors la transmission s'est bien passée

Détection et correction d'erreurs

- Cette méthode détecte quasiment toutes les erreurs, mais :
  - si une erreur se glisse dans la partie de détection, on conclut à une erreur sur la partie donnée même si elle est correcte
- Exemple de division polynomiale :
  - soit à transmettre 1101011011 → ceci donne $P(x) = X^9+X^8+X^6+X^4+X^3+X^1+X^0$
  - soit le générateur 10011 → ceci donne $G(x) = X^4+X^1+X^0$
  - le *résultat* de la division vaut → ceci donne $D(x) = X^9+X^8+X^3+X^1$: 1100001010
  - le *reste* vaut alors : → ceci donne $R(x) = X^3+X^2+X^1$, 1110
- La zone de détection d'erreurs est communément appelée :
  - *Cyclic Redundancy Checksum*
- Un autre nom courant est :
  - *Frame Check Sequence*