Réseaux Informatiques et Internet

Support du Cours

Driss El Ouadghiri

Email : dmelouad@gmail.com

Université My Ismail
Faculté des Sciences
Département de Mathématiques et d'Informatique
Meknès

Plan général

- Concepts de bases
- Historique
- Classification
- Normalisation
- modèle OSI
- Couche Liaison de données
- Couche Réseau
- modèle TCP/IP
- réseaux locaux : Ethernet, interconnexion des réseaux

Définition générale :

Réseau :

- **Ensemble d'objets ou de personnes maintenus en liaison entre eux,**
- **Par extension, l'ensemble des liaisons établies,**
- **Vient du mot latin *rete* qui signifie filet,**
- **Les objets reliés sont appelés *"nœuds du réseau".***

Exemples:

Réseau social, réseau de transport, réseau téléphonique, réseaux de neurones, **réseau informatique, etc ...**

Le développement des systèmes informatiques et leur Intégration dans tous les domaines de la vie ont fait naître le besoin aux réseaux.

<u>Définition (Réseaux Informatiques)</u> :

Un réseau est un ensemble de machines reliés les unes aux autres qui servent à échanger des flux d'informations selon des règles définies.

L'intérêt des réseaux informatiques :

- **Échange et partage de données informatiques**
- **Partage d'une connexion Internet**
- **Média de communication (email,vidéoconférences, newsgroup, voix sur IP,…)**
- **Transfert de fichiers**

- Lancement de procédures distantes (client/ serveur)
- Accès à des bases de données centralisées ou réparties
- Partage de logiciels
- Partage de ressources (fichiers, applications, imprimantes...)
- Archivage : utilisation d'espace disque pour l'archivage ou la sauvegarde
- Etc.

- **Intranet : réseau interne d’une entité organisationnelle**

- **Internet : réseau de réseaux à l’échelle mondiale, liaisons effectuées grâce aux lignes téléphoniques et aux lignes dédiées. C’est une interconnexion des réseaux informatiques**

Historique des réseaux

- **Années 50, premiers réseaux informatiques : un ensemble de terminaux reliés à une entité centrale conçus pour un usage spécifique.**

- **1955 : premier réseau informatique à but commercial, SABRE (American Airlines)**

- **Années 60, projet ARPA : concevoir un système de communication susceptible de résister à une attaque militaire.**

- **1969 : premier lien ARPANET entre l'Université de Californie et Stanford.**

- **Dans la même année : ajout de deux autres universités (Utah et Santa Barbara), ce qui fait 4 nœuds.**

- **1970 : finalisation de NCP (Network Control Protocol), protocole pour le réseau ARPANET.**

- **1971 : premier échange de courrier électronique**

- **1972 : ARPANET rassemble une quarantaine de machines militaires et universitaires.**

- **1974 : première démonstration de l'ARPANET au public.**

- **1974, développement de TCP/IP : interconnecter des réseaux hétérogènes**
- **Mai 1982 : 235 machines connectées**
- **Fin de 1982 : passage de NCP vers TCP/IP**
- **1984 : Internet désigne un réseau mondial utilisant le protocole TCP/IP**
- **Fin de 1984 : mise en place du DNS (Domain Name Server)**

- **1991 : 535 000 machines reliées**
- **1992 : mise en place du Web**
- **1994 : ouverture au grand public du courrier électronique**
- **Mai 2001 : 379 millions d’internautes**

Classifications

Les réseaux peuvent être classés selon :

- leur taille
- leur mode de transmission
- leur mode de communication
- leur structure (topologie)
- leur mode de fonctionnement
- leur type de commutation

Classification selon la taille (étendu géographique)

- **(W)PAN = (*Wireless) Personnal Area Network* ou réseau local personnel (sans-fil).**
  - **Etendue : quelques mètres**
  - **Nombre d'abonnés : une dizaine**
  - **Débit courant : 1 Mbit/s**
  - **Mode de connexion : multipoint**
  - **Utilisation : connexion à un ordinateur de son imprimante, de son PDA …**
  - **Normes : 802.15, bluetooth**

- **(W)LAN = (*Wireless) Local Area Network* ou réseau local (sans-fil).**
  - **– Etendue : un mètre à quelques kilomètres**
  - **– Nombre d'abonnés : de deux à une centaine d’abonnés**
  - **– Débit courant : de 1 à 100Mbit/s**
  - **– Mode de connexion : multipoint**
  - **– Utilisation : Interconnexion de matériels dans une entreprise. On parle de réseau d’entreprise**
  - **– Normes : 802.11 (Sans-fil), 802.3 (CSMA/CD) …**

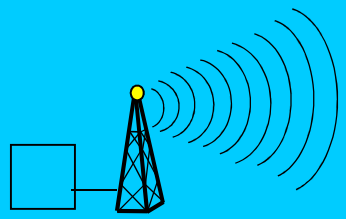
- **(W)MAN = (Wireless) *Metropolitan Area Network* ou réseau métropolitain (sans-fil).**
  - **Etendue : deux kilomètres à cent kilomètres**
  - **Nombre d'abonnés : jusqu'à mille abonnés**
  - **Débit courant : de 10 au Giga bit/s**
  - **Mode de connexion : multipoint**
  - **Utilisation : Interconnexion de réseaux locaux**
  - **Normes : 802.16 (WMAN), 802.6 DQDB (Distributed Queue Dual Bus), FDDI (FiberDistibuted Data Interface iso9314)**

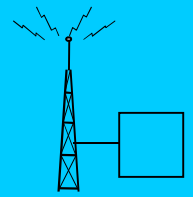
- WAN = *Wide Area Network* ou réseau grande distance.
  - **Etendue : plus de cent voir des milliers de kilomètres**
  - **Nombre d'abonnés : illimité**
  - **Débit courant : de 10 Mbit/s au Giga bit/s**
  - **Mode de connexion : point à point**

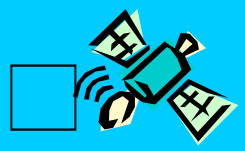
Exemple : réseaux à l'échelle d'un pays ou même d'un continent (Marwan qui est le réseau reliant les établissements universitaire du Maroc, Renater (REseau NAtional de Télécommunications pour la technologie, l'Enseignement et la Recherche) qui permet de relier plus de 600 établissements et d'accéder à l'internet).

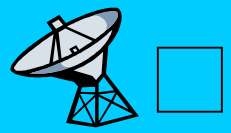
Moyens de transmission

- liaison physique
- ondes radio
- satellite









Modes de communication

- **unidirectionnelle (*one way*)**
  (clavier, souris, …)

- **bidirectionnelle à l'alternat (*half duplex*)**
  (talkies-walkies, …)

- **bidirectionnelle (*full duplex*)**
  (téléphone, …)

Diapositive 16

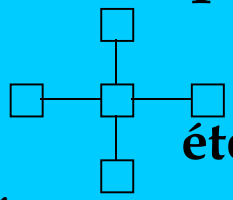
u1 user; 08/10/2007

Topologie

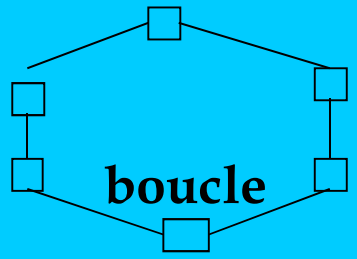
- **mode liens point à point**

point à point

(deux machines reliées par un câble entre elles est un réseau)



étoile

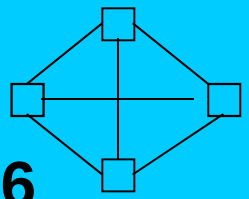


boucle

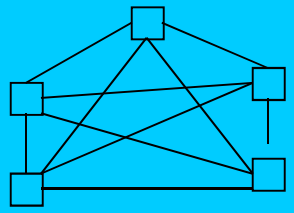
– **Inconvénients : nombre de liens (maille) !**

n stations : n(n-1)/2 liens

1



6

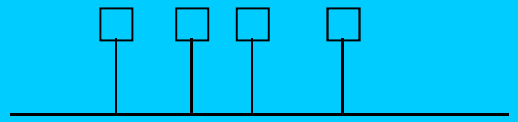


10

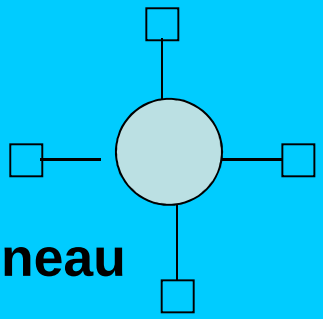
Constitués d'un grand nombre de connexions entre les machines (Pour aller d'une machine à une autre, un message passe par des machines intermédiaires, plusieurs routes possibles (grands réseaux).

Topologie (suite)

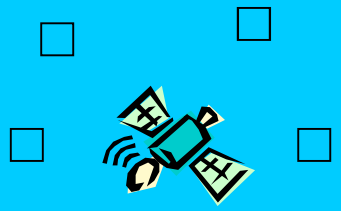
- **Mode de diffusion (ou multipoints)**



bus



anneau



satellite

Mode de fonctionnement

- **avec connexion**

  1) Demande d'établissement de connexion

  2) Si le récepteur refuse : pas de connexion

  3) Si le récepteur accepte : établissement d'un circuit

  4) Transfert des données

  5) Fermeture (Libération) de la connexion

- **sans connexion**
  - **le client poste une lettre**
  - **chaque lettre porte un nom et une adresse**
  - **un client a une adresse**
  - **les contenus des informations sont inconnus du prestataire**
  - **les supports du transport sont inconnus du client**

Type de commutation

- **commutation de circuits : téléphone**
- **commutation de messages: coûteuse (mémoire à chaque nœud)**
- **commutation de paquets : avec ou sans connexion (Internet)**
- **commutation de cellules : ATM**

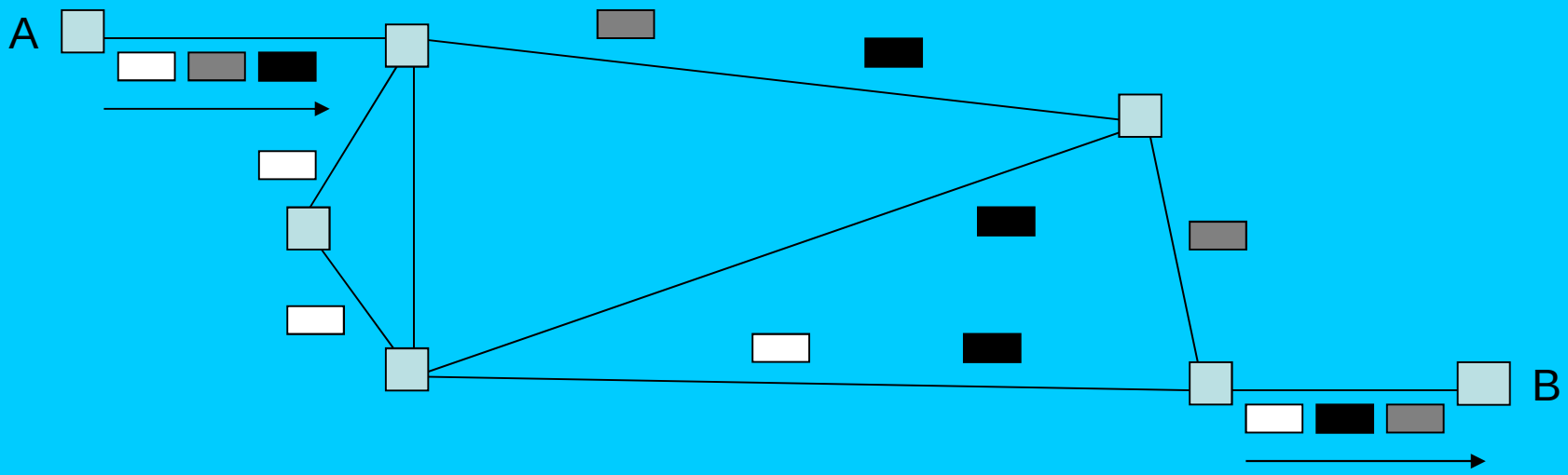
Commutation de circuits

- chaque commutateur gère un certain nombre de circuits
- temps d'établissement du circuit
- circuit entre deux interlocuteurs = suite de circuits entre commutateurs
- problème : mobilisation du circuit même si aucune données à transmettre
- une fois connecté, temps de propagation faibles et pas de congestion

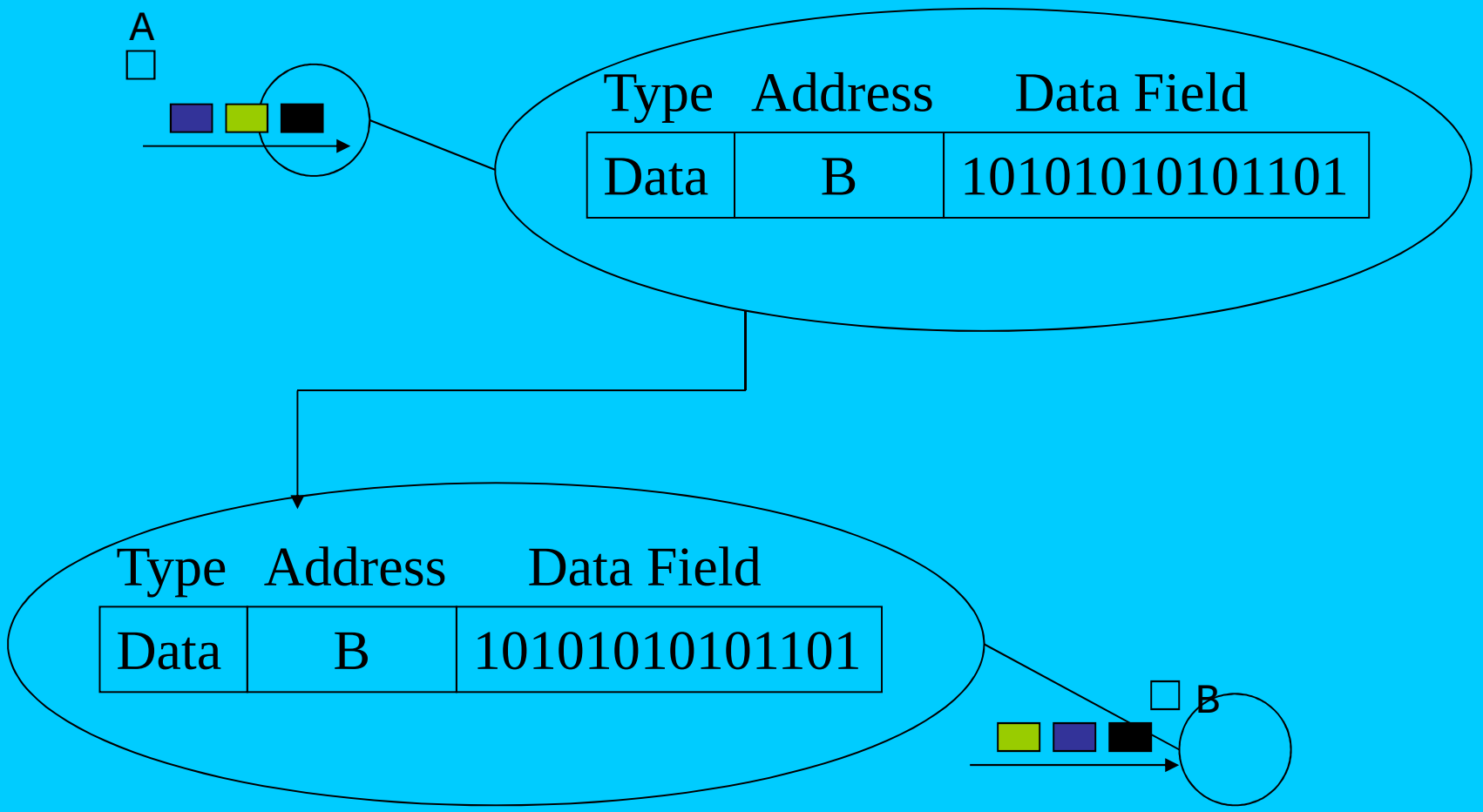
Commutation de messages

- Transmission du message de commutateur en commutateur jusqu'au destinataire
- Stockage et acheminement
- problème de la mémoire nécessaire
- si problème : retransmission de tout le message

Commutation de paquets (mode datagramme)



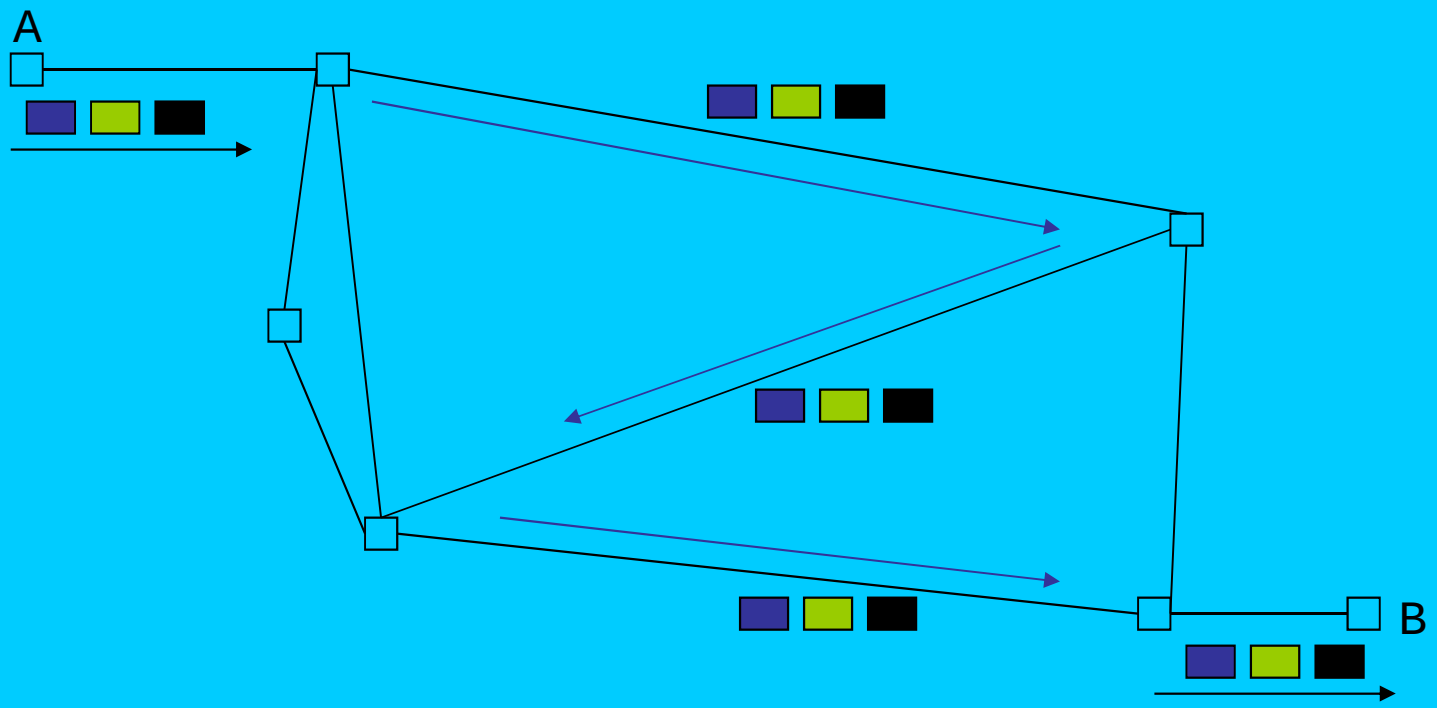
Commutation de paquets (mode datagramme, suite)



Commutation de paquets (mode datagramme, suite)

- mécanismes de contrôle de flux
- mécanismes de contrôle d'erreur
- ordre de remise non garanti
- non transparence du réseau

Commutation de paquets (circuit virtuel)



Commutation de paquets (circuit virtuel, suite)

Phases de communication :

1°) Établissement du circuit virtuel

2°) Utilisation du circuit virtuel

3°) Libération du circuit virtuel

Avantages :

séquencement des paquets

rétablissement en cas de coupure

Réémission

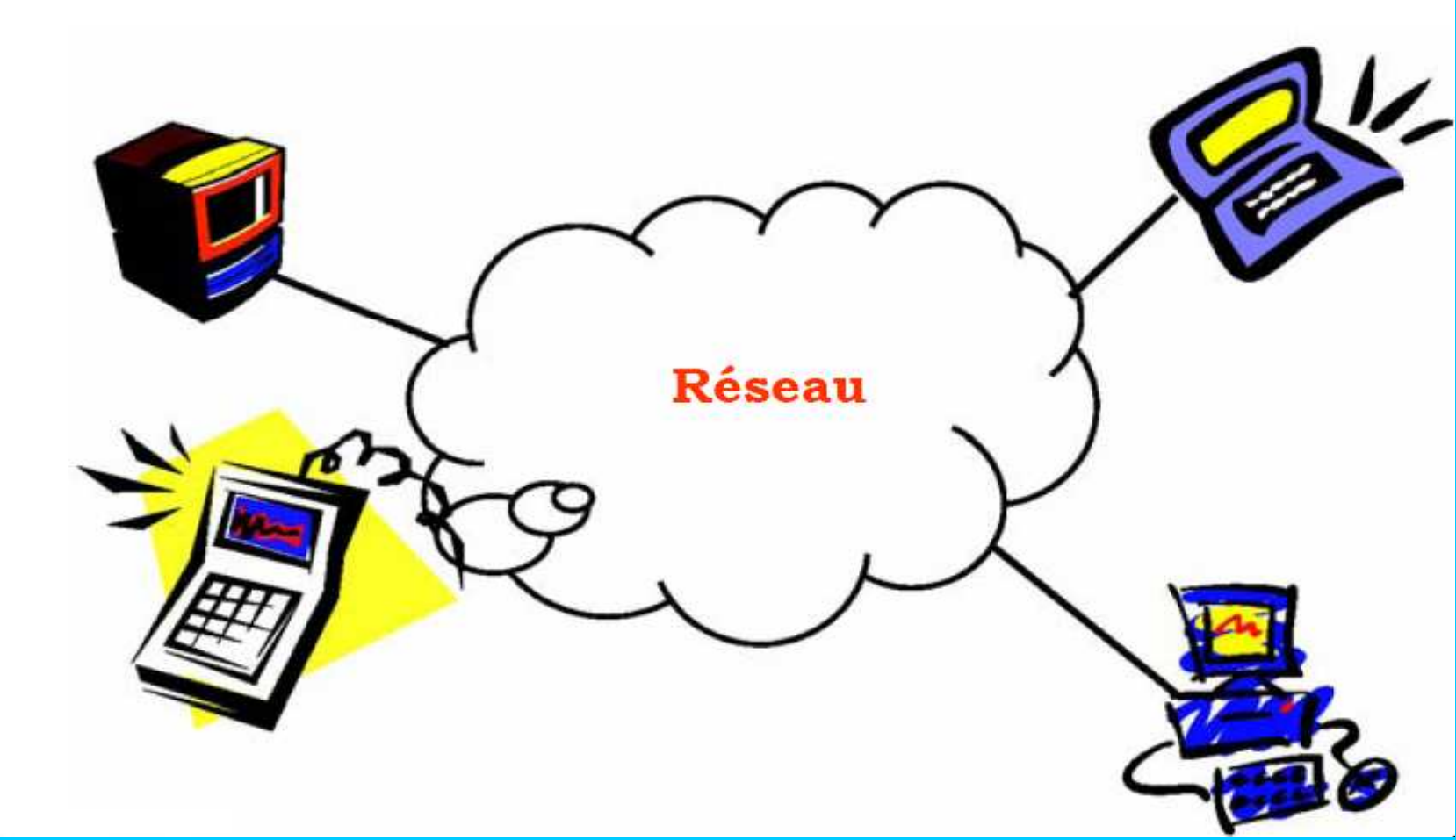
Commutation de cellule : ATM

- Cas particulier de la commutation de paquets pour les réseaux ATM (Asynchronous Transfert Mode).

- Tous les paquets, nommés ici cellules, ont une longueur fixe de 53 octets.
  - 48 octets pour les données
  - 5 octets pour l'en-tête

Normalisation

Motivation



Objectifs

- Assurer l'interopérabilité des systèmes hétérogènes
- Offrir une qualité minimum : Optimisation d'utilisation des ressources
- Faciliter la conception, la mise en oeuvre et la maintenance des systèmes
- Assurer la pérennité des choix de conception

Organismes

- **Deux organismes de normalisation pour les réseaux informatiques**

  – **ISO (I*nternational* S*tandardization* O*rganization)* : un organisme dépendant de l'ONU;**
  les représentants nationaux sont des organismes nationaux de normalisation :
    - **ANSI : American National Standard Institute pour les USA**
    - **AFNOR: Association Française de normalisation pour la France**
    - **…**

  – **UIT-T (U*nion* I*nternationale des* T*élécommunications-section* T*élécommunication)* : qui a remplacé le CCITT (*Comité Consultatif International pour le Télégraphe et le Téléphone*) ; L'UIT-T comprend des opérateurs et des industriels des télécoms**

- **Autres organismes :**
  – **IEEE : Institute of Electrical and Electronic Engineers**
  – **IETF/IRTF : Internet Engineering/Internet Research Task Force**
  – **EIA : Electronics Industries Association**
  – **ECMA : European Computer Manufacturer**
  – **…**

Identification des normes

- Les normes ISO sont préfixées par IS (ou ISO)
  - Exemple: IS 8802.3 (Réseau local: Ethernet).
  - Voir le site: www.iso.org
- Les normes UIT sont désignées par une lettre suivie d'un point et d'un numéro
  - Exemple: V.34 (Modem 33600 bauds)
  - Voir le site: http://www.itu.int/ITU-T/
- Les normes IETF sont désignées par RFC suivi d'un numéro
  - Exemple: RFC 791 (protocole IP)
  - Voir le site: www.ietf.org/rfc.Html

Le modèle OSI

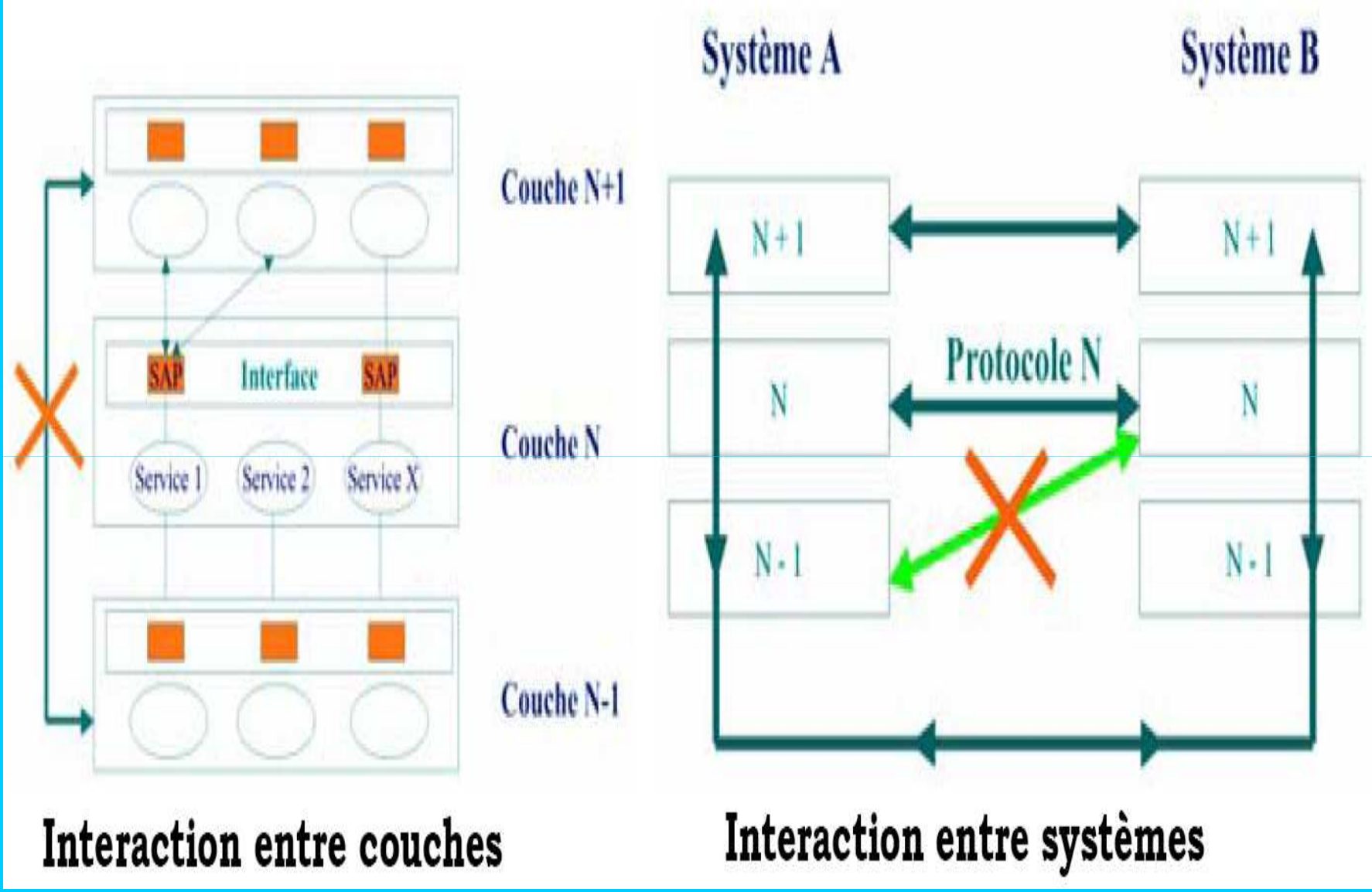
- **La norme : IS 7498**
  - **IS 7498-1 : le modèle de référence OSI de base**
  - **IS 7498-2 : l'architecture de sécurité**
  - **IS 7498-4 : le cadre général pour la gestion**
- **Modèle de référence fondé sur le principe « Diviser pour régner »**
- **Le principe de base est la représentation des réseaux sous la forme de couche de fonctions superposées les unes aux autres.**
- **L'étude du système de communication revient alors à l'étude de ses éléments et offre une plus grande :**
  - **Facilité d'étude**
  - **Indépendance des couches**
  - **Souplesse d'évolution**

Pourquoi le modèle OSI ?

- Structurer les techniques de conception des logiciels afin de diminuer leur complexité.
- Offrir un moyen standardisé pour échanger des informations.
- Augmenter la compatibilité des systèmes.
- Modèle de référence décrivant les différents éléments du problème de la communication.
- Modèle en 7 couches décrivant les fonctionnalités nécessaires à la communication et leur organisation.

Les couches OSI

- Une couche correspond à un niveau d'abstraction de la communication. Exemples :
  - communication entre applications,
  - communication entre routeurs
  - communication physique, etc.
- Chaque couche offre un ensemble de fonctions particulières
- Une couche offre des services à la couche supérieure et utilise les services offerts par la couche inférieure


Système A
Système B
Couche N+1
Couche N
Couche N-1
SAP
Interface
SAP
Service 1
Service 2
Service X
N+1
N+1
N
Protocole N
N
N-1
N-1
Interaction entre couches
Interaction entre systèmes

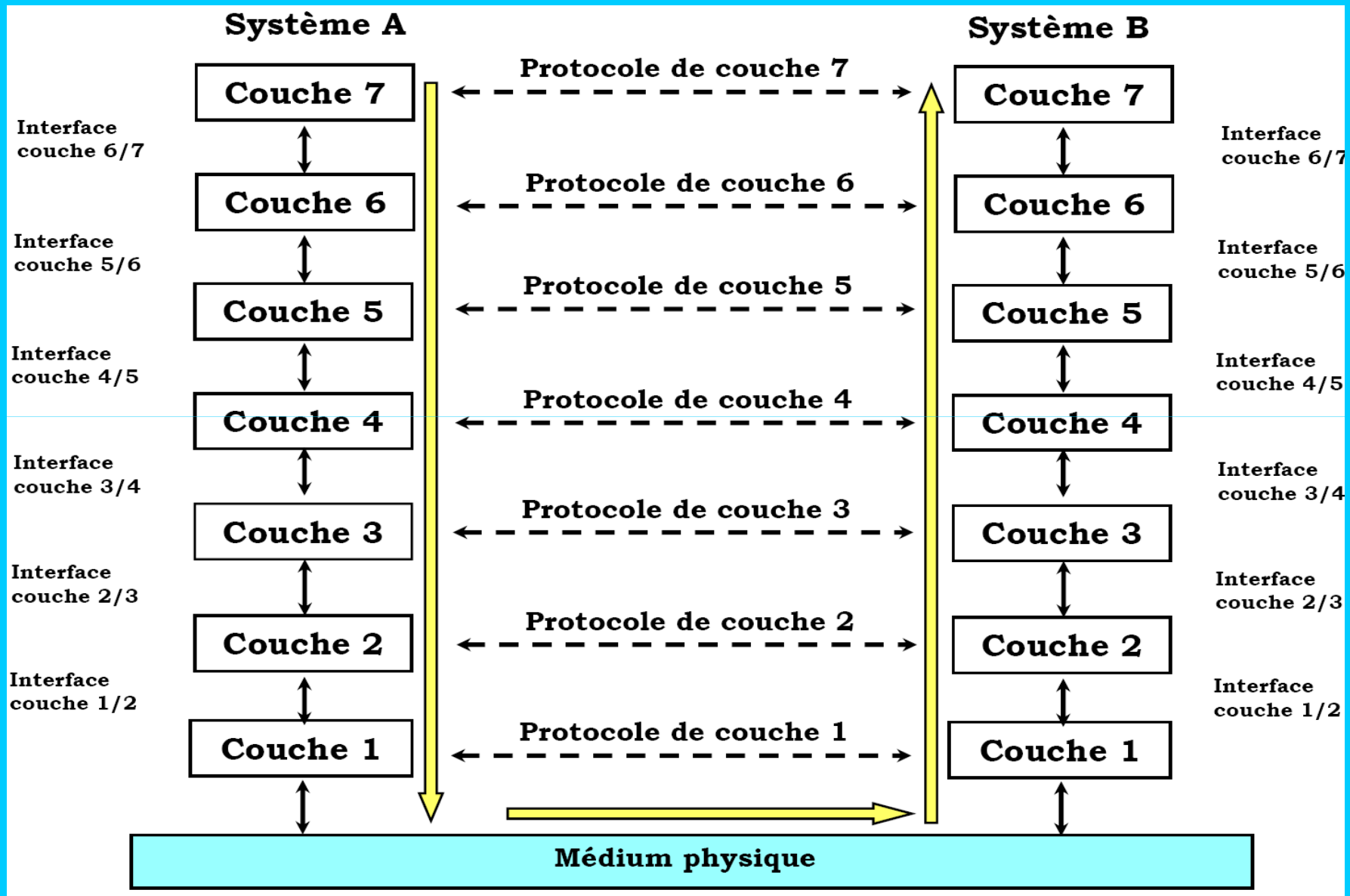
Service et protocole

<u>Remarque</u> : le service et le protocole sont deux concepts distincts même s'ils sont fréquemment confondus.

- **Un service est un ensemble de primitives (opérations) qu'une couche fournit à la couche immédiatement supérieure. Un service se rapporte à une interface entre deux couches, la couche inférieure étant le fournisseur du service, la supérieure l'utilisateur du service**

- **Un protocole est un ensemble de règles et conventions utilisées pour communiquer entre deux couches de même niveau d'abstraction de deux systèmes différents (format et signification des trames, paquets ou messages). Les entités utilisent les protocoles pour implanter leurs spécifications de service.**

Les couches OSI

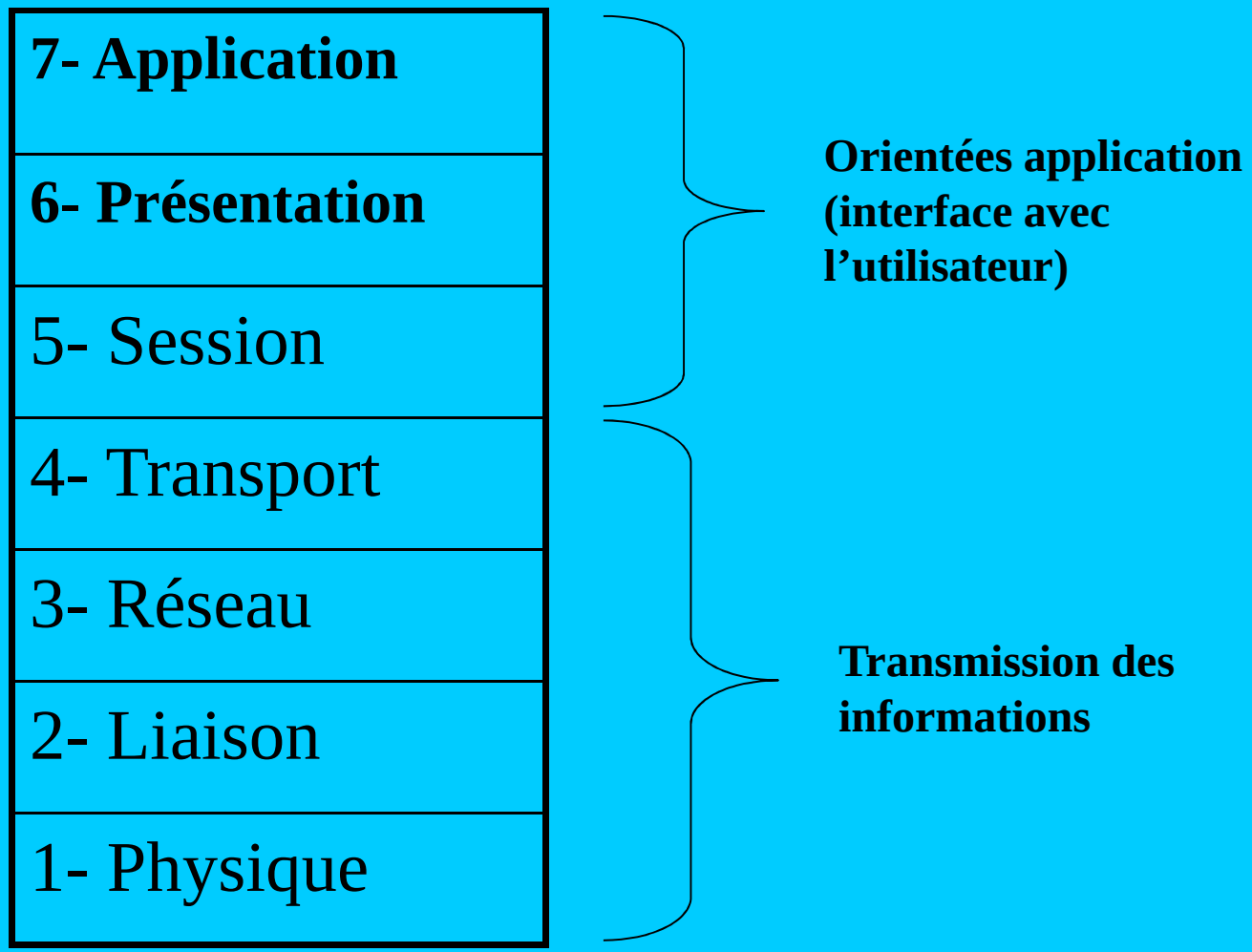


dmelouad@gmail.com

Le modèle OSI

- Application : quelles sont les données à envoyer ?
- Présentation : sous quelle forme ?
- Session : qui est le destinataire ?
- Transport : où est le destinataire ?
- Réseau : quelle route faut-il prendre?
- Liaison : quelles sont les caractéristiques du réseau ?
- Physique : quel est le support physique ?

Le modèle OSI



7- Application
6- Présentation
5- Session
4- Transport
3- Réseau
2- Liaison
1- Physique

1- Couche physique

- **Chargée de la transmission du signal (bits) sur le canal de communication.**
- **Service : émission ou réception d'un bit ou d'une suite de bits continue.**
- **Questions : durée transmission d'un bit, types de connectiques (câble coaxial, USB, Bluetooth, ADSL,…)**

2- Couche de liaison

7- Application
6- Présentation
5- Session
4- Transport
3- Réseau
2- Liaison
1- Physique

- **Assure la transmission des données sur le support physique.**

- **Service : découpage en trames, transmission des trames en séquence de bits, gestion des trames d'acquittement, régulation du flux.**

- **Questions : format des trames, mode de régulation du trafic, accès au canal (token ring, Ethernet,…)**

3- Couche de réseau

- Détermine la façon dont les paquets sont acheminés de la source au destinataire.
- Service : routage et relayage des paquets
- Questions : déterminer le chemin à suivre (IPv4, IPv6)

7- Application
6- Présentation
5- Session
4- Transport
3- Réseau
2- Liaison
1- Physique

4- Couche de transport

- **Assure un transport des informations de bout en bout**
- **Service : découpage des données, multiplexage des connections, contrôle et correction des Erreurs**
- **Questions : gérer l'établissement et le relâchement des connexion, contrôle le flux (TCP, UDP)**

7- Application
6- Présentation
5- Session
4- Transport
3- Réseau
2- Liaison
1- Physique

5- Couche de session

- **Permet d'établir des sessions de communication entre différentes machines.**
- **Service : gestion du dialogue, Synchronisation**
- **Questions : manière de se connecter, reprise après interruption,…**

7- Application
6- Présentation
5- Session
4- Transport
3- Réseau
2- Liaison
1- Physique

6- Couche de présentation

- **Chargée du codage des données Applicatives**

- **Service : encodage des données dans une norme reconnue**

- **Questions : norme d'encodage (MIME, XML,…), représentation de structures abstraites**

7- Application
6- Présentation
5- Session
4- Transport
3- Réseau
2- Liaison
1- Physique

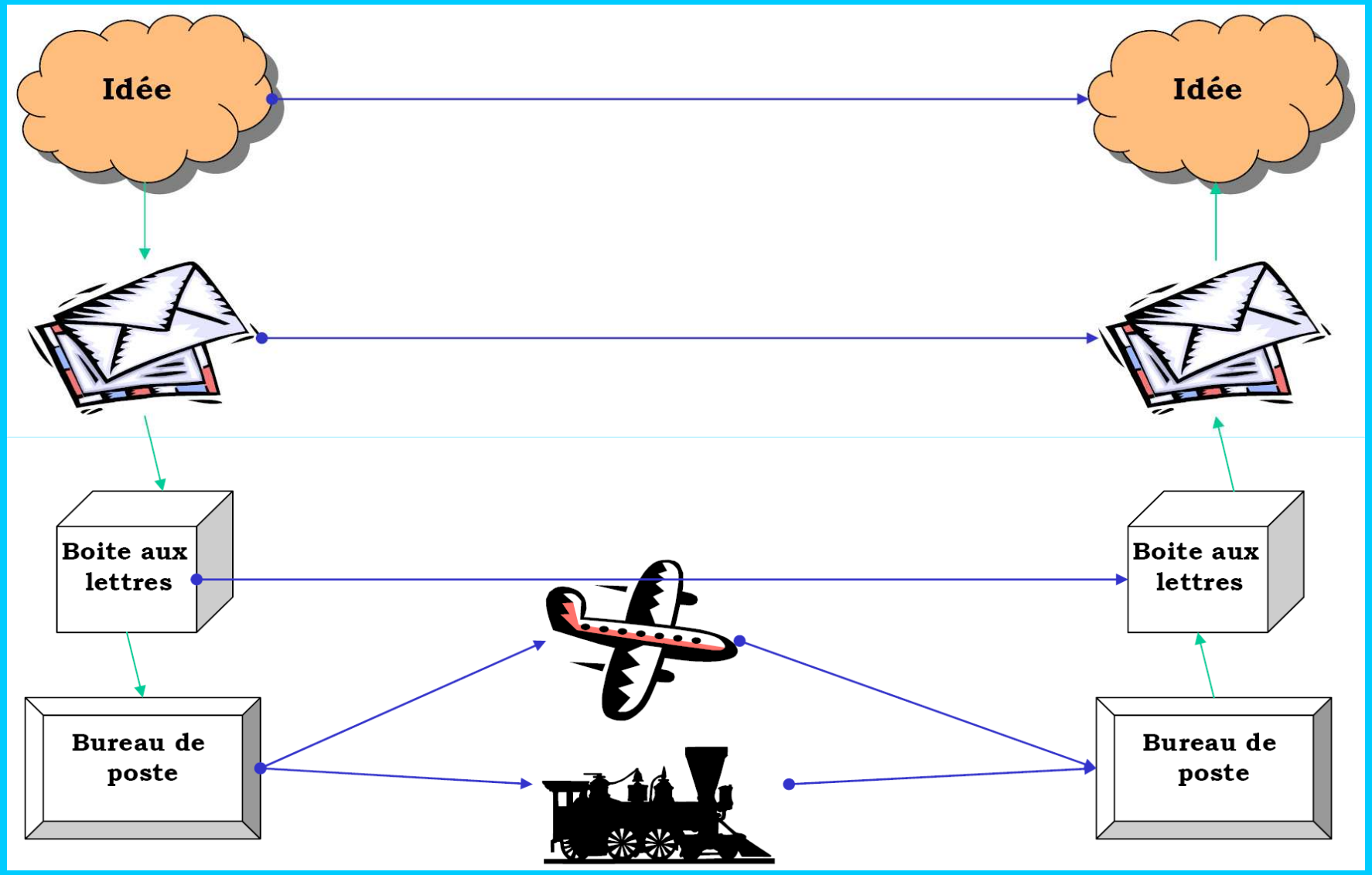
7- Couche application

- **Assure l'interface avec l'utilisateur**
- **Services : afficher les informations reçues, envoyer les informations fournies par l'utilisateur, transfert de fichier**
- **Questions : définir les commandes à disposition des utilisateurs, présentation des réponses, gestion des incompatibilités entre systèmes de fichiers**

Critères de choix des couche

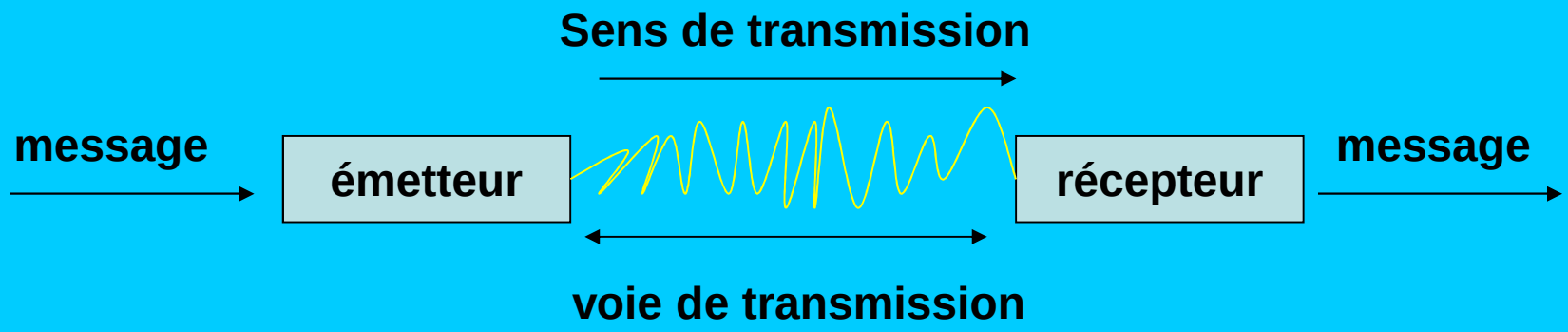
- **Une couche peut être créer quand un niveau d'abstraction est déterminé.**
- **Une couche doit avoir une fonction bien définie.**
- **La fonction d'une couche doit tenir compte les standards internationaux définis pour les protocoles.**
- **Les frontières d'une couche (aspects interfaces) doivent être choisies de manière à minimiser le flux d'informations à travers les interfaces.**
- **Le nombres des couches doit être assez grand pour distinguer les fonctions de chaque couche et peut être aussi petit que possible afin que l'architecture ne devient pas très compliquée.**

La simplicité du modèle OSI



Transmission des données

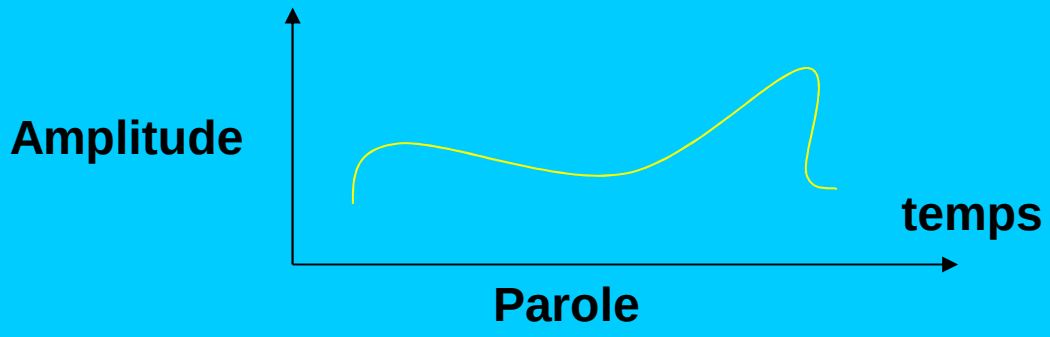
La transmission d'un message nécessite son encodage en signaux de type électrique ou électromagnétique.



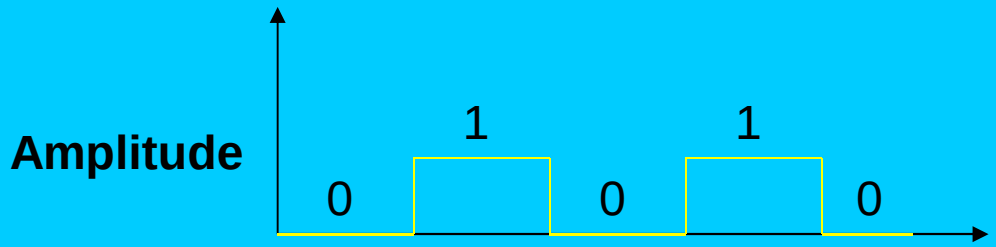
Un signal est une fonction du temps (peut être une fonction de fréquence).

- Un signal S(t) est continu si lim S(t) = S(a) ∀ a

Exemple : t➔a



Parole



Binaire 0 et 1

- Un signal est périodique si et seulement si :

$S(t+T) = S(t)$ $-\infty < t < +\infty$, T est la période du signal

<u>Caractéristique d'un signal périodique</u>

- **Amplitude : valeur instantané d'un signal (dans le cas d'une onde électromagnétique ou électrique, l'amplitude est exprimé en volts)**
- **Fréquence (1/T) : nombre de répétitions de la période dans une seconde.**
- **Phase (exprimée en radian)**

Ainsi, une sinusoïde peut être exprimée par :

$$S(t) = A * \sin(2*\pi*f*t + \theta)$$

amplitude **fréquence** **phase**

Analyse de fourrier

- **Un signal répétitif S(t) avec une période T peut être décomposé en une somme (peut être infini) de signaux sinusoïdaux avec des fréquences multiples de 1 / T.**

$$S(t) = 1/2\ C + \sum_{n=1}^{\infty} \left(a_n \sin(2 \pi n f t) + b_n \cos(2 \pi n f t) \right), \quad f = 1 / T$$

avec :

$$a_n = 2/T \int_0^T S(t) \sin(2 \pi n f t)\, dt$$

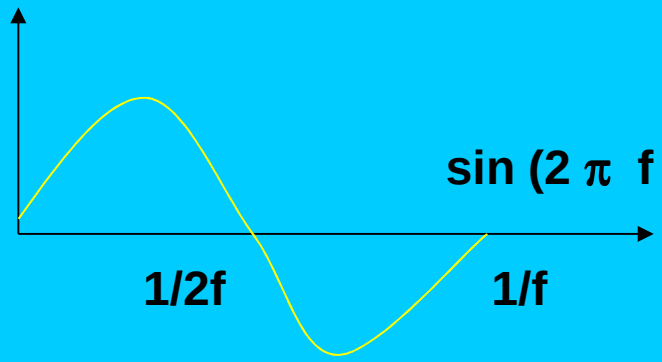
$$a_n = 2/T \int_0^T S(t) \cos(2 \pi n f t)\, dt$$

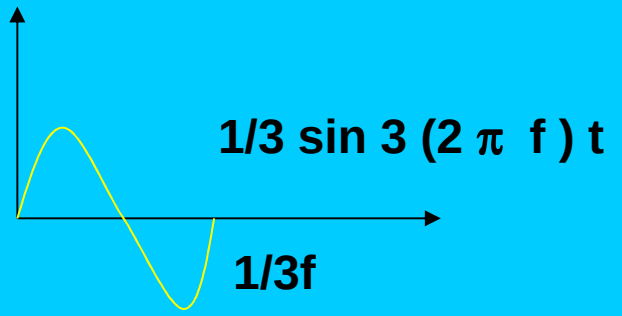
$$C = 2/T \int_0^T S(t)\, dt, \text{ C correspond au cas ou } n = 0$$

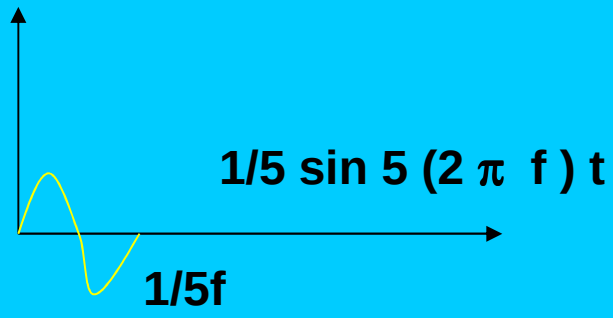
Exemple :

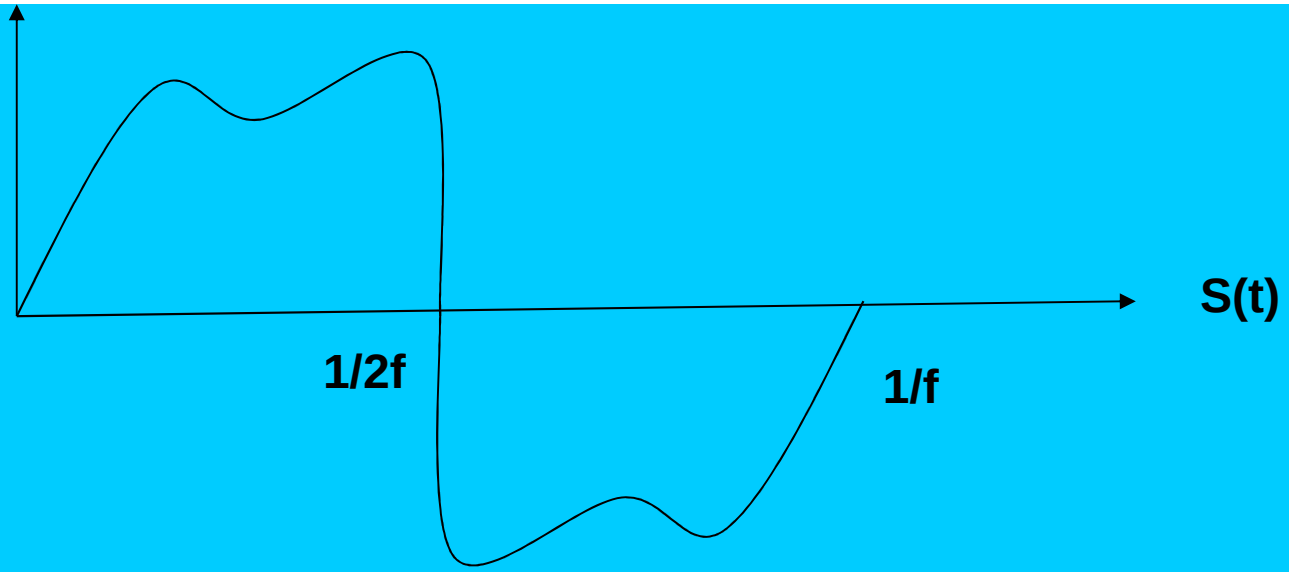
S(t) = sin (2 π f) t + 1/3 sin 3 (2 π f) t + 1/5 sin 5 (2 π f) t

Les composantes de S sont trois ondes sinusoïdales de fréquences respectives f, 3f et 5f.









- **Le spectre d'un signal est la porté de fréquence qu'il contient**

- **La bande passante d'un signal est la largeur de son spectre**

<u>Exemple</u> : Si le spectre s'étend de f à 5 f , la bande passante sera 4 f.

- **Le débit binaire est le nombre de bits transmis par unité du temps.**
  Exemple : 64 Kbits / s ou 200 Mbits / s

<u>Exercice</u> : Si la durée de transmission d'un bit est 50 ms, quel est le débit binaire ?

Autres exemple :

Supposons que :

- une pulsation positive représente le ‘1’ binaire et une pulsation négative représente 0 binaire.

- La durée de chaque pulsation est 1/2f.

Alors le débit d’information en bps (bits par seconde) est 2f.

Ainsi, pour f = 100 Hz, le débit est : 200 bps

Transmission Analogique et Digitale

Les termes 'Digital' et 'Analogue' sont fréquemment utilisés dans le domaine de communication dans aux moins trois niveaux :

- **Données**
- **Signaux**
- **Transmission**

- **Données analogiques et données digitales (numériques)**

- Une donnée analogique est une donnée qui prend des valeurs continues dans un intervalle du temps.

Exemple : Voix, image, …

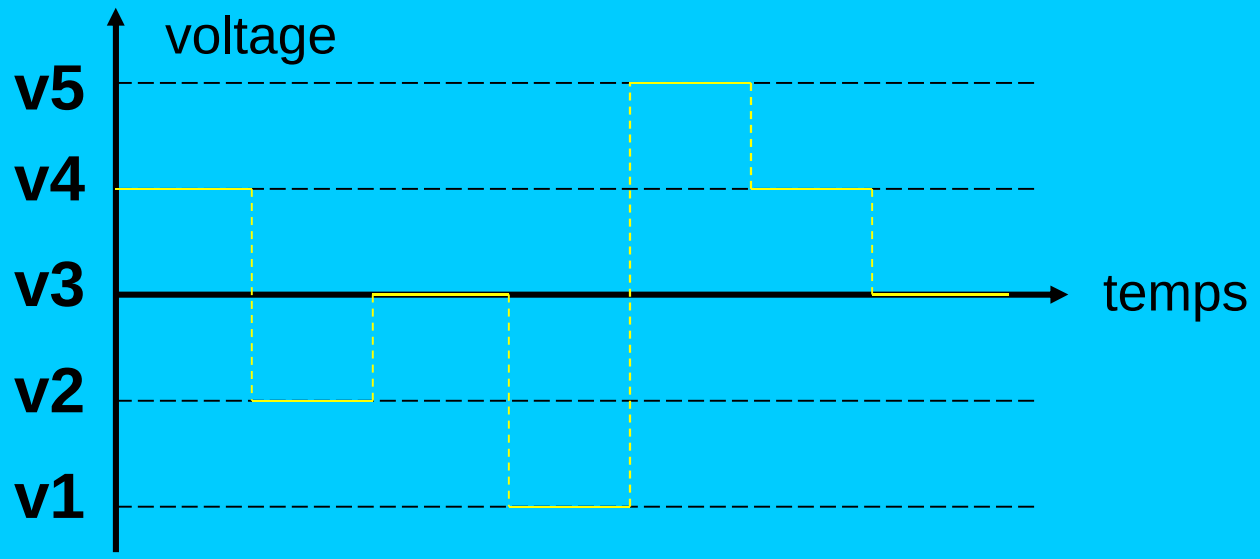
- Une donnée digitale est une donnée ayant des valeurs discrètes ou discontinues

Exemple : entiers, textes, …

- **Signaux**

- un signal alogique est une onde électromagnétique qui varie d'une façon continue et qui peut se propager à travers un medium (paire torsadé, câble coaxial, fibre optique …)

- un signal digital est une séquence de pulsation de voltage qui peut traverser un medium



- **Transmission**

- Transmission analogique : est un moyen pour transmettre des signaux analogiques sans se préoccuper de leurs contenus (le signal peut représenter des données analogiques (voix) ou des données digitales (données binaires).

A cause de l'atténuation du signal après une longue distance, on utilise des amplificateurs qui intensifie le signal.

Inconvénient : en cas de bruit, l'amplificateur intensifie le bruit aussi.

- Transmission digitale (numérique) : donne plus d'importance au contenu du signal.

On utilise des répéteurs pour retransmettre le signal digital de nouveau (le bruit n'est pas cumulatif).

Capacité théorique d'un canal

Théorème de Nyquiest (1924) :

Le débit maximum pour un canal parfait (sans bruit) pour un système ayant une bande passante W et un signal à M niveaux (nombre d'états possibles) est donné par :

$$C = 2 W \log_2 M$$

Exemple:

Pour un signal digital à deux états 0 ou 1 (M = 2), la capacité du canal (débit maximum) est : $C = 2 W$

Si on a un canal sans bruit ayant W = 3 Khz. Ce canal ne peut pas transmettre un signal binaire à une vitesse qui dépasse 6000 bps

Canal avec bruit

Le Décibel est une mesure de bruit. C'est une mesure de rapport entre deux niveaux de puissance.

Nombre de décibels ⟵ $Ndb = 10 \ Log10 \ P1/P2$

Si un signal de puissance P2 = 10W est inséré dans une ligne de transmission et après une certaine distance, la puissance mesuré du même signal est P1 = 5 W, alors :

$$Perte = 10 \ Log \ (5/10)$$

Rapport Signal/Bruit

S/N est dit rapport Signal/Bruit. C'est un paramètre important pour déterminer la performance d'un système de transmission.

S : puissance du signal, N : puissance du bruit

La quantité S/N est souvent convertie en décibel par :

$$10 \log_{10} (S/N) = (S/N)_{db}$$

S/N = 10 → 10 dbs

= 100 → 20 dbs

= 1000 → 30 dbs

<u>Théorème de Shannon (1948) :</u>

La capacité théorique d'un canal ayant une bande passante W (Hz) et un rapport Signal/Bruit (S/N) est donnée par :

$$C = W \log_2(1+S/N)$$

Nombre maximal de bits par seconde (bps)

Exemple :

Sur une ligne téléphonique dont la bande passante 3200 Hz, pour un rapport Signal/Bruit de 30 dbs, on peut atteindre théoriquement une capacité de 30 Kbits/s (30 894 bps).

En pratique un débit de 9 600 bps est considéré comme excellent.

Medium de Transmission : Lien physique entre émetteur et Récepteur.

Deux types :

- **Hardware : Câbles**
- **Software : air, vide**

La qualité d'une transmission dépend aussi bien de la nature du signal que celle du medium utilisé.

Medium	Débit	Bande Passante
Câble torsadé	1 M bps	250 Khz
Câble coaxial	500 M bps	350 Khz
Fibre Optique	1 G bps	1 Ghz

Câble torsadé :

- **Paire de fil à base de cuivre**
- **très utilisé par le système téléphonique**

Câble coaxial :

- **Opère sur une grande fréquence**
- **Diamètre de 0,4 à 1 pouce**
- **Utilisé pour les longues distances**
- **Utilisé dans les réseaux locaux**

Fibre Optique :

- **fibre mince qui permet de conduire un rayon optique.**
- **Peut être à base de verre ou du plastique**
- **Au début, utilisée pour les applications militaires**

Satellite :

- Utilisé pour relier deux ou plusieurs stations terrestres
- Utilisé pour transmettre voix, images, données …
- bande passante allant de 1 à 10 Ghz

Modulation

Un signal numérique est un signal qui a des formes rectangulaires ou carrées.

La transmission des signaux numérique (qui consiste à émettre sur la ligne des courants de bits à transmettre) est souvent qualifiée de transmission bande de base.

<u>Problème</u> : Le signal transmis en bande de base subit des dégradations très rapides avec la distance parcourue. Si le signal n'est pas généré très souvent, il prend une forme quelconque que le récepteur sera incapable de comprendre (transmission sur de très courtes distances).

Un signal de forme sinusoïdale, même affaibli, pourra bien être décodé par le récepteur.

Définition :

la modulation est la transmission des données digitales sur des signaux analogiques.

Trois grandes catégories de modulation sont dénombrées :

1- Modulation d'amplitude (ASK : Amplitude-Shift Keying)

2- Modulation de fréquence (FSK : Frequency-Shift Keying)

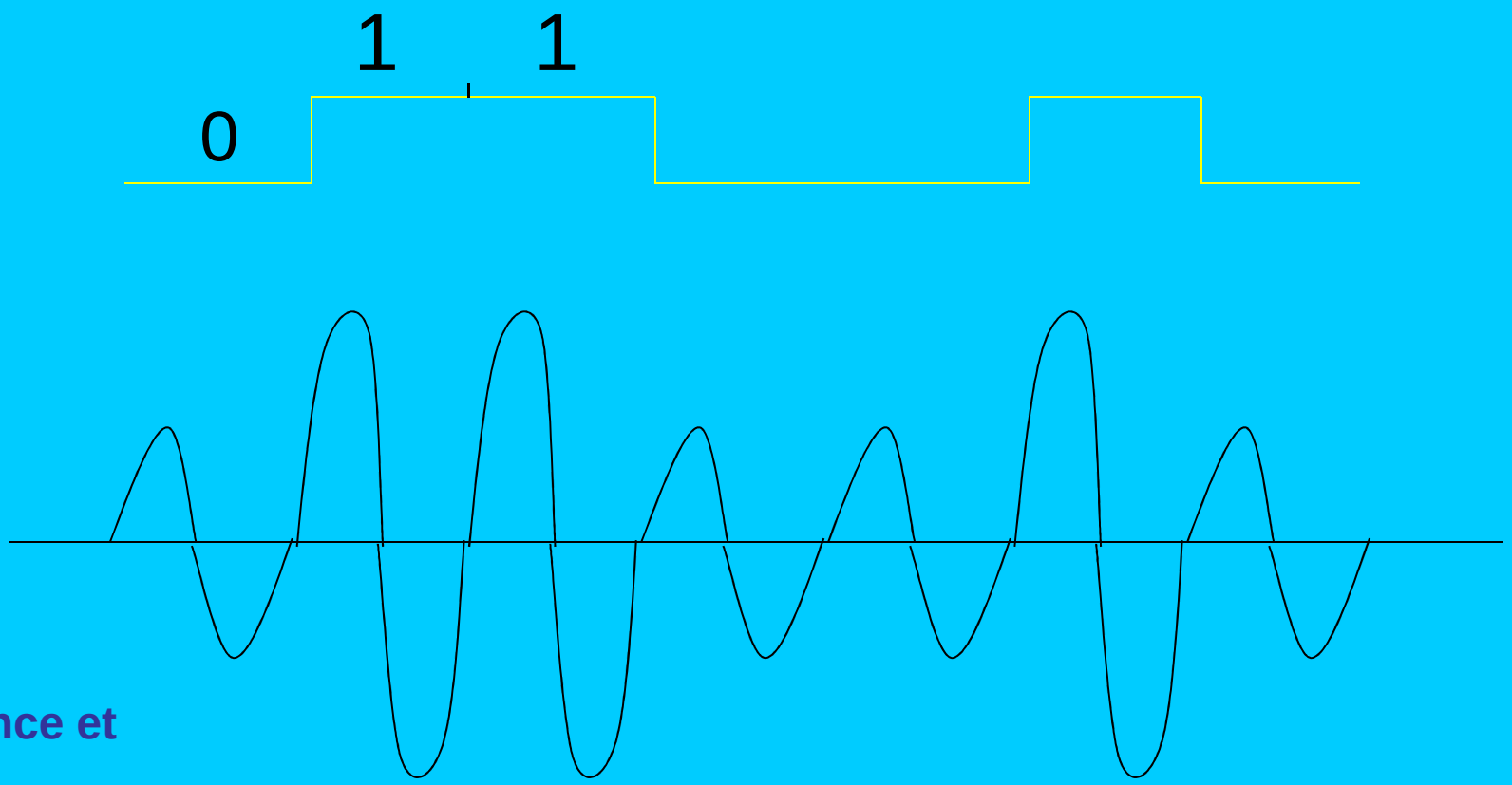
3- Modulation de phase (PSK : Phase-Shift Keying)

Avant de les décrire, précisons qu'il faut un matériel intermédiaire pour moduler un signal sous la forme sinusoïdale.

Le modem (Modulateur, DEModulateur) prend un signal en bande de base et va le moduler (le mettre sous forme analogique particulière).

1- Modulation d'amplitude

Consiste à modifier l'amplitude du signal porteur. Une valeur d'amplitude est attribuée à l'information binaire '0' et une autre à l'information binaire '1'.

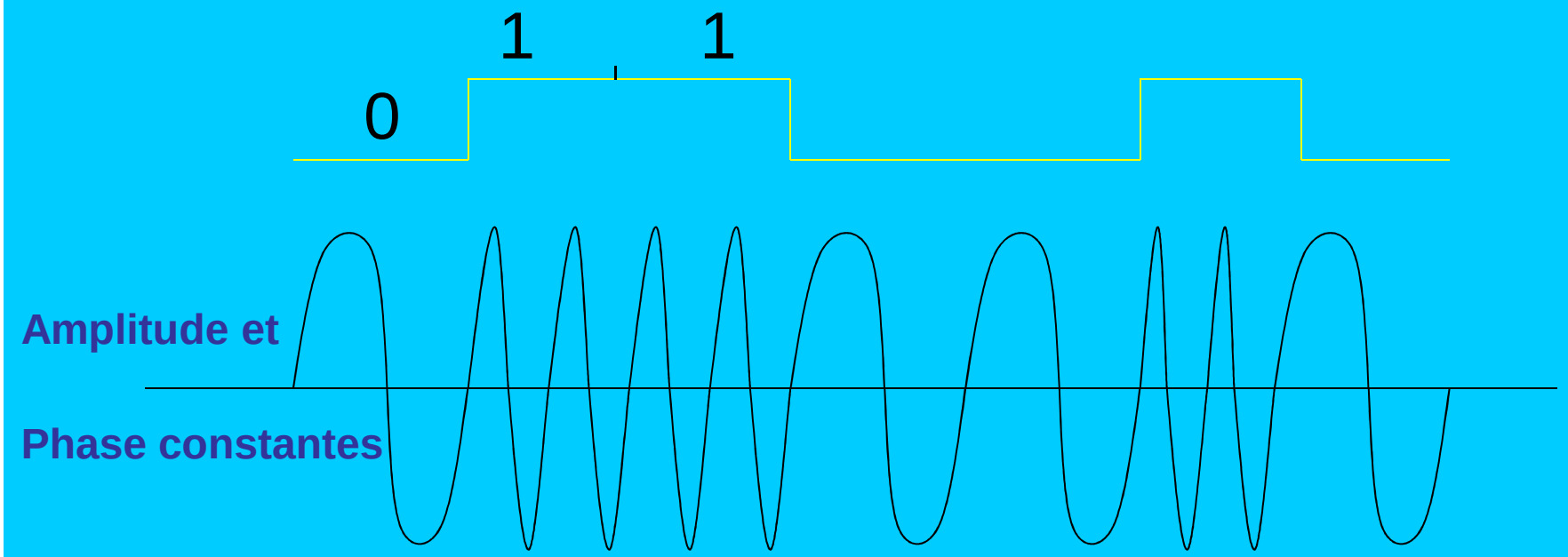


Fréquence et

Phase constantes

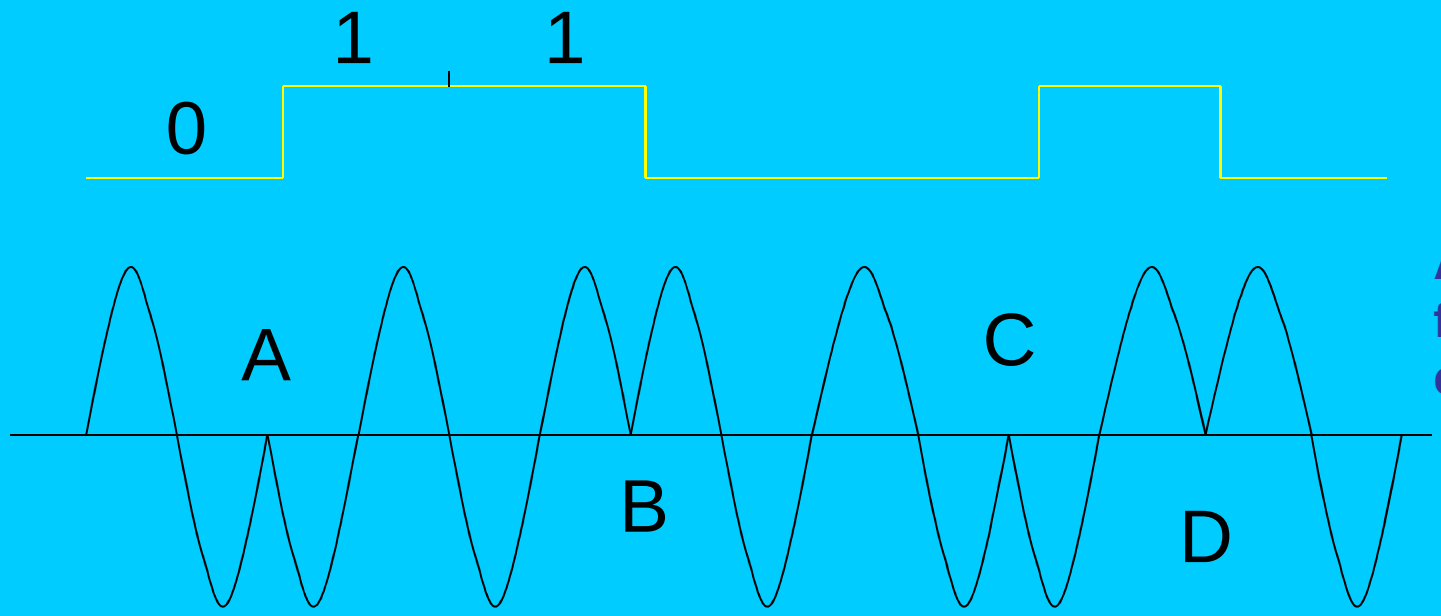
2- Modulation de fréquence

Cette modulation est caractérisée par la possibilité pour l'émetteur de changer la fréquence d'envoie des signaux suivant que l 'élément binaire à émettre est '0' ou '1'.



3- Modulation de phase

La distinction entre l'information '0' et '1' est éffectuée par un signal qui commence à des emplacements différents de la sinusoïde (phase). Sur la figure les valeur '0' et '1' sont représentées par des pahses respectives de 0 à 180°.



Amplitude et fréquence constantes

Techniques de communication de données

Coopération entre Émetteur/Récepteur

On distingue 2 techniques :

1- Transmission Asynchrone

• La transmission se fait caractère par caractère. Le début d'un caractère est signalé par un bit début (Start-Bit) ayant la valeur 0.

• La ligne est en état de repos si aucun caractère n'est pas transmis (par convention l'élément binaire 1 est maintenu).

• Après le Start-Bit on envoie un caractère (dans certains cas, un bit de parité est ajouté à la fin).

Définition :

Le bit de parité est émis de façon à ce que le nombre total de ‘1’ dans le caractère soit pair ou impaire.

Le dernier bit de caractère est suivi par un bit fin (Stop-Bit) qui a pour valeur ‘1’.

Si une chaîne de caractère est envoyée, l’intervalle entre deux caractères est uniforme et est égale au Stop-Bit.

Remarque :

- **Chaque caractère contient deux bits d'entête (Start-Bit et Stop-Bit), un bit de parité et 7 autres bits (code de caractère).**

- **Un caractère est donc transmis sur 10 bits**

- **On a donc un OverHead de 3/10 = 0.20 = 20 %**

- **1/5 de la capacité de canal est utilisé par la transmission asynchrone : ce qui est très énorme.**

2- Transmission Synchrone

♦ Cette technique consiste à transmettre des blocs de caractères ou de bits au lieu de caractère par caractère.

♦ Pour déterminer le début et la fin d'un bloc on utilise les informations de contrôle appelés préambule et post-ambule

♦ Le bloc de données + les informations de contrôles constituent ce qu'on appelle une trame.

Exemple : HDLC (High level Data Link Control)

48 bits de contrôle pour un message de 1000 bits.

On a un OverHead de 48/1000 = 4.6%

Détection et traitement des erreurs

Effets thermique
Electromagnétique } ⇒ **Erreurs**

Plusieurs techniques de détection des erreurs :

Bit de parité : consiste à ajouter un bit à la fin de chaque caractère de la trame.

- **Parité paire : ajouté un bit de façon à ce que le nombre de '1' soit paire (y compris le bit de parité).**
- **Parité impaire : même définition (paire ➔ impaire)**

Le Receveur examine si ∑ '1' est impaire.

Inconvénient : Changement de plusieurs bits ➔ problème de détection.

<u>VRC & LRC</u> (Vertical & Langitudinal Redundancy Checks) :

La trame est considérée comme un tableau de caractères arrangés en deux dimensions.

	Bit 1	Bit 2	…	Bit n	Bit de parité
Caractère 1	b11	b12	…	b1n	R1
Caractère 2	b21	b22	…	b2n	R2
	⋮	⋮	⋮	⋮	⋮
Caractère m	bm1	bm2	…	bmn	Rm
	C1	C2	…	Cn	cn+1

- **Rj = ȸj1⊕ bj2 ⊕ … ⊕ bjn (bit de parité de jème caractère).**

Le bits de parité de chaque caractère est dit VCR

- **bij : jème bit de ième caractère.**
- **n : nombre de bits par caractère**
- **m : nombre de caractères par trame**
- **Cİ = b1i⊕ b2i ⊕ … ⊕ bni**

Le caractère de parité de chaque bit est dit LCR

Exemple :

							VRC
0 1	0	1	1 0	1	0	1	0
1	1	1	1	0	0	0	0
1	0	0	0	1	0	1	1
0 1	1	0	0 1	0	0	0	1

LRC

- **Le 1er et le 4ème bits du 1er caractère sont en erreur**
- **Le 1er et le 4ème bits du caractère LRC sont en erreur**

Mais aucune détection d'erreur.

Donc certains types d'erreurs ne sont pas détectés par cette technique.

CRC (Cyclic Redundancy Checks) :

C’est une technique basée sur le fait qu’on considère une suite de bits comme étant une représentation polynomiale avec les coefficients 0 et 1. Cette technique s’appelle code polynomial.

Autrement, un message de k bits est considéré comme une liste de coefficients d’un polynôme de k termes x^0 jusqu’à x^{k-1} (degré K-1).

<u>Exemple</u> :

110001 est représenté par le polynôme $x^5+x^4+x^0$ à 6 termes et qui a pour coefficients 1, 1, 0, 0, 0 et 1.

Idée de cette technique :

1- Choisir un polynôme générateur G(x)

2- s'arranger pour que le message soit divisible par G(x)

3- A la réception, vérifier si le message reçu est divisible par G(x).

Si on veut adopter la technique du code polynomial, il faut que l'émetteur et le récepteur se mette d'accord sur un polynôme générateur.

Du coté de l'émetteur on ajoute un Check Sum (FCS : Frame Check Sequence) à la fin de message de manière à ce que le polynôme soit divisible par G(x).

De son coté le récepteur vérifie la validité du message reçu en effectuant la division du message par G(x) : si le reste de la division est non nul alors le message reçu n'est pas valide (erreur).

Soit M(x) le polynôme associé à un message de m bits.

<u>Algorithme pour calculer FCS</u>

1- Soit G(x) de degré r. jouter r bits à zéro, du coté droit, pour le message. Le polynôme obtenu est x^r M(x).

2- S'arranger pour que le message envoyé soit divisible par G(x).

3- On effectue la division de x^r M(x) par G(x) et on ajoute le reste de la division R(x) à la chaîne x^r M(x). Le message à transmettre est $T(x) = x^r M(x) \oplus R(x)$

4- Le récepteur vérifie si le message reçu est divisible par G(x).

Exercice :

M(x) = 101110

$G(x) = x^3 + 1$

$M(x) = x^5 + x^3 + x^2 + x$

$x^3 M(x) = x^8 + x^6 + x^5 + x^4$

Trouver le message T(x) à transmettre?

$$
\begin{array}{l|l}
x^8 + x^6 + x^5 + x^4 & x^3 + 1 \\
\hline
-\ x^8 + \quad x^5 & \\
\hline
\quad x^6 + x^4 & \\
-\ x^6 + x^3 & \\
\hline
\quad x^4 - x^3 & \\
-\ x^4 + x & \\
\hline
\quad - x^3 - x & \\
-\ - x^3 - 1 & \\
\hline
\quad - x + 1 = R(x) &
\end{array}
$$

$$x^5 + x^3 + x - 1$$

Donc $T(x) = x^8 + x^6 + x^5 + x^4 - x + 1 = 101110011$

Autre méthode :

$G(x) = 1001$

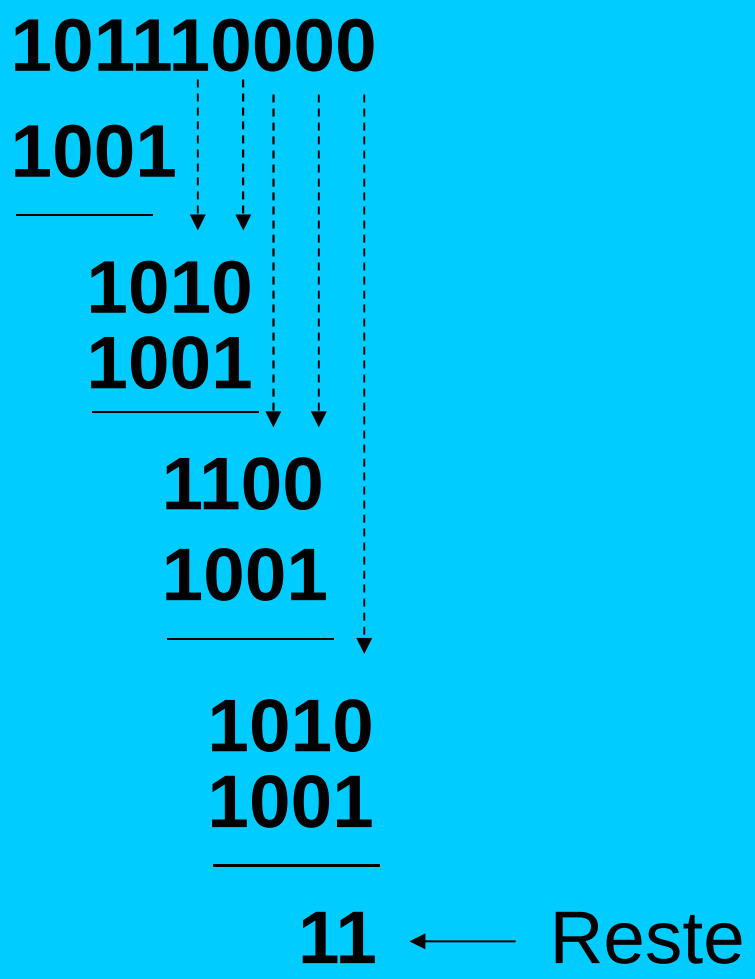
$x^r M(x) = 101110000$

$R(x) = 11 = 000000011$

Donc

$T(x) = 101110000 + 000000011$

$= 101110011$



Exercices :

1) - Vérifier est ce que T(x) est bien reçu.

- Dans le cas ou on change le $5^{ème}$ et le $6^{ème}$ bit dans T(x), est ce que T(x) est bien reçu.

2) Considérons le message 101011000110 reçu.

$G(x) = x^6 + x^4 + x + 1$

Ce message est-t-il correct?

Les versions les plus utilisées de G(x) :

$$CRC - 12 = x^{12} + x^{11} + x^3 + x^2 + x + 1$$

$$CRC - 16 = x^{16} + x^{15} + x^2 + 1$$

$$CRC - CCITT = x^{16} + x^{12} + x^5 + 1$$

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

CRC – 32 est utilisée par ETHERNET.

Niveau lien

0bjectifs :

Avoir une communication efficace entre deux stations directement liées :

- Synchronisation des trames : le début et la fin d'une trame doivent être clairement identifiés.

- Contrôle de flux.

- Contrôle d'erreur

- adressage : les identités des stations en communication doivent être bien connues.

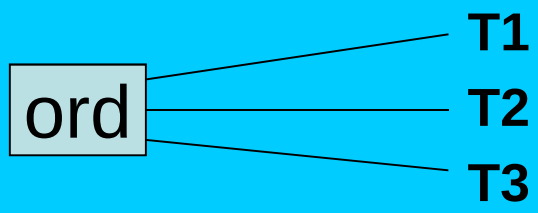
Topologie

Deux stations ➔ lien point à point

Plus que deux stations ➔ lien multipoint

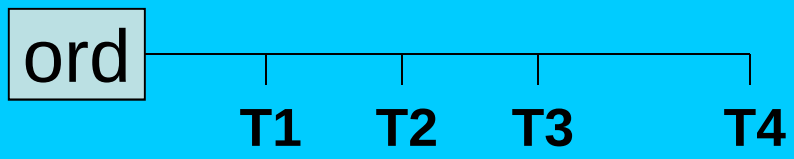
Exemple :

- **Point à point :**



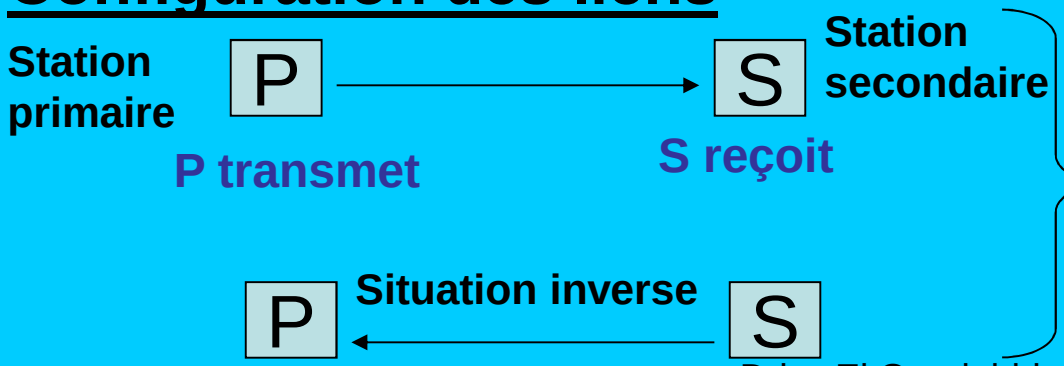
Nécessite :
-un port par I/O par terminal
- Une ligne/lien

- **Multipoint : (cas de réseaux locaux)**



Un seul port I/O partagé

Configuration des liens



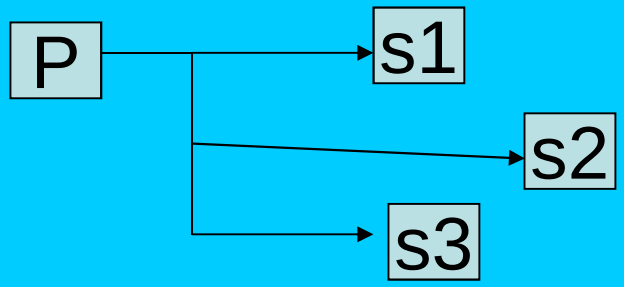
Ce genre de connexion S'appelle :
Half-Duplex point à point

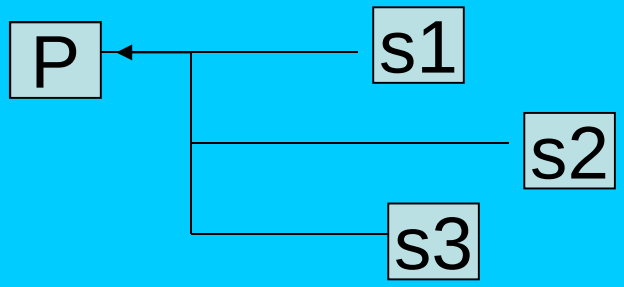
- **On peut avoir aussi :**

Connexion Full-Duplex

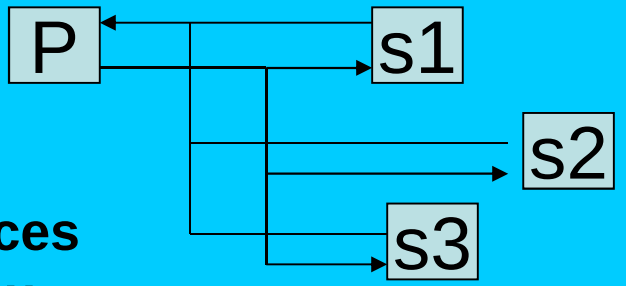
Transmission/Réception au même temps

- **Half–Duplex multipoint**





- **Full-Duplex multipoint**



On peut avoir un mixage de tous ces configuration dans un même réseau.

Les liaisons de transmission ne sont pas parfaites :

- **le débit binaire est limité**
- **le délai de propagation est non nul**
- **des erreurs de transmission peuvent survenir**
- **temps de traitement des machines (nœuds)**

Tous ces facteurs influent fortement sur le transfert de données et sont pris en compte dans l'élaboration des protocoles.

La couche liaison

- détermine la façon dont les bits venant de la couche physique sont regroupés en trames,

- se charge de traiter les erreurs de transmission

- et effectue un contrôle de flux pour régulariser le volume d'informations échangées entre sources et destination.

Remarque :

Le niveau lien s'applique juste entre deux nœuds adjacents (type point à point). Ça veut dire que les deux machines sont connectées par un canal de transmission qui délivre les bits dans l'ordre dans lequel ils ont été émis.

- Quand la couche liaison de données accepte un message (paquet), venant de la couche réseau, elle l'encapsule en lui ajoutant des informations de contrôle.

- Une trame contient donc un paquet encapsulé avec des informations de contrôle.

- La trame est envoyée, par la suite, à la couche liaison de données de la machine distante.

- La couche liaison de données de la machine distante va transmettre à la couche réseau de la même machine la partie paquet de la trame (les protocole de la couche liaison de données et de la couche réseau sont indépendants).

- Les protocoles de communications de chaque couche peuvent évoluer ainsi indépendamment les uns des autres.

Par la suite, on va étudier des protocoles élémentaires de la couche liaison de données par ordre de complexité croissante.

Protocoles élémentaires du niveau lien

Trame = message + infos de contrôle

Structures de données :

```
const Nbrebits : … ; {taille des messages}
NumSequence = 0..MaxSeq ; { Numéroter les trames}
message = ensemble de bits  ;
TypeTrame = ( Data, ACK, NAK) ;

trame = record
            kind : TypeTrame ;
            seq, ACK : NumSequence ;
            info : message ;
        end ;
```

Procedure send(s : trame) ;
{ envoyer d'une trame à la couche physique }

Procedure get(var r : trame) ;
{ recevoir une trame de la couche physique}

Procedure wait(var e : event) ;
{ event = (trame_ok, trame_err, time_out, …) }

Procedure FromHost(var m : message) ;
{ la couche réseau remet un message à la couche liaison}

Procedure ToHost(var m : message) ;
{ la couche liaison remet un message à la couche réseau}

Procedure StartTimer(k : NumSequence);
{ déclenchement de temporisateur, activation de timer}

Procedure StopTimer(k : NumSequence)
{ arrêt de temporisateur, désactivation de timer}

Protocole 1 : **Supposons que**

- la communication est unidirectionnelle (circulation des données dans un seul sens).

- Il n'y a pas d'erreurs (canal parfait).

- l'émetteur et le récepteur sont toujours prêt à émettre et à recevoir.

- on a suffisamment d'espace de stockage (infini).

- les temps de traitement sont ignorés.

```
Procedure sender1;
        var s : trame ;
              buffer : message ;
Begin
        repeat
            fromHost(buffer) ; { prendre ce qu'il y a à
                                 envoyer }
           s.info := buffer ; { copie dans s pour
                                transmission }
            send(s) ; { envoie vers la couche physique }
        until     FinDeTemps { on boucle }
end;  { émetteur 1 }
```

```
Procedure    receiver1;
   Var  r  : trame;
          e : event ;
Begin
   repeat
        wait(e) ; { seul événement possible : trame_ok }
        get(r); { acquisition de la trame }
        tohost(r.info); { on passe les données à la couche
                                  réseau }
   until     FinDeTemps { on boucle }
End ; { récepteur 1 }
```

Le protocole 1 est dit utopique (Conception idéale, Parfaitement irréaliste).

Ce protocole se décompose en deux procédures : sender1 et receier1.

- **La procédure sender1 est implantée dans la couche liaison de données de la machine source et receiver1 dans la couche liaison de données de la machine destination.**

- **L'émetteur réalise une boucle infinie, qui consiste à émettre des données aussi vite sur la ligne.**

- **Le corps de la boucle est composé de trois actions :**

  - demander un paquet à la couche réseau

  - construire une trame en utilisant la variable s

  - envoyer la trame construite

- **Dans ce protocole, on utilise que le champ info de la trame (Le canal étant parfait et le contrôle de flux est inexistant).**

La procédure receivr1 est basée aussi sur une conception simple :

- **Le récepteur attend l'arrivé d'une trame.**

- **l'appel à la procédure *get* enlève la trame venant d'arriver d'une mémoire physique et l'arrange dans la variable r.**

- **Par la suite, la partie de la trame correspondant au champ info est passer seulement à la couche réseau.**

- **La couche liaison de données attend de nouveau l'arrivée de la prochaine trame.**

Protocole 2 : on suppose que :

- la Circulation des données est unidirectionnelle (émetteur vers récepteur)
- le Canal parfait (pas d'erreurs)
- Le protocole empêche l'émetteur d'inonder le récepteur (ralentir la transmission quand le récepteur n'est plus en mesure de traiter les données : c'est le contrôle de flux).

Ainsi, la suppression de l'hypothèse la plus irréaliste faite dans le protocole 1 est nécessaire : l'aptitude de la couche réseau de la machine distante à traiter les données arrivant de façon instantanée (ou la présence dans la couche liaison de données d'une capacité infinie de mémoire pour stocker les trames en attendant qu'elles soient traiter par la couche réseau.

L'émetteur doit attendre jusqu'à ce que le récepteur soit prêt à recevoir. C'est un protocole de type « envoyer et attendre ».

```
Procedure    sender2 ;
    var s : trame ;
          buffer : message ;
          e : event ; { seul événement possible : trame_ok }
Begin
    repeat
          FromHost(buffer) ; { acquisition du message
                               (paquet) venant de la couche
                               réseau }
          s.info := buffer ; { copie dans s pour transmission }
          send(s) ; { envoie de la trame }
          wait(e) ;  { attendre un acquittement }
    until         FinDeTemps
End ; { émetteur 2 }
```

```
Procedure    receiver2 ;
    var  r,s : trame ;
             e : event ;
Begin
    repeat
         wait(e) ; { seul événement possible : trame_ok }
         get(r) ; { acqusition de la trame }
         ToHost(r.info) ; { transmission du paquet à la
                                couche réseau }
         send(s) ; { envoie d'une trame acquittement }
    until        FinDeTemps ;
End ; { récepteur 2}
```

Dans ce protocole, le récepteur informe l'émetteur de son état :

- **après transmission d'un paquet à la couche réseau, le récepteur envoi une petite trame (ne contenant aucune information) qui donne la permission à l'émetteur d'envoyer la trame suivante.**

- **Après l'envoie d'une trame, l'émetteur doit attendre jusqu'à l'arrivée de cette petite trame (l'acquittement).**

La différence entre sender1 et sender2 est que après avoir transmis un paquet à la couche réseau, sender2 envoie une trame d'acquittement avant de boucler et d'attendre l'arrivée de la prochaine trame.

Bien que le sens de la circulation des données est unidirectionnel, ce protocole permet l'alternance de flots de données : l'émetteur envoie d'abord une trame de données puis le récepteur envoie une trame d'acquittement, et ainsi de suite.

Il s'agit d'un canal de transmission à l'alternat.

Protocole 3 : Un protocole simple pour un canal bruité

Dans ce cas le canal n'est pas parfait. Des trames peuvent être erronées ou perdues.

Le récepteur n'émet une trame d'acquittement que si la trame est correctement reçue. Si une trame erronée arrive, elle ne sera pas pris en compte.

Après l'expiration de temporisateur, l'émetteur retransmet la même trame.

On répète cette séquence jusqu'à ce que la trame soit correctement parvenue au récepteur.

L’algorithme ci-dessus présente des défaillance.

Rappelons que la communication doit être transparente et sans erreurs entre les couches réseaux de deux machines :

- La couche réseau de la machine A passe une série de paquets à la couche Liaison de données de la même machine.

- La couche liaison de données de la machine A doit s’assurer que cette série de paquets sera délivrée à la couche réseau de la machine B par sa couche liaison de données.

Or la couche réseau de B n’a aucun moyen de savoir si un paquet à été perdu ou dupliqué. C’est la couche liaison de données qui doit garantir qu’aucune erreur ne peut être remise à la couche réseau (un même paquets deux fois remis).

Considérons le scénario suivant :

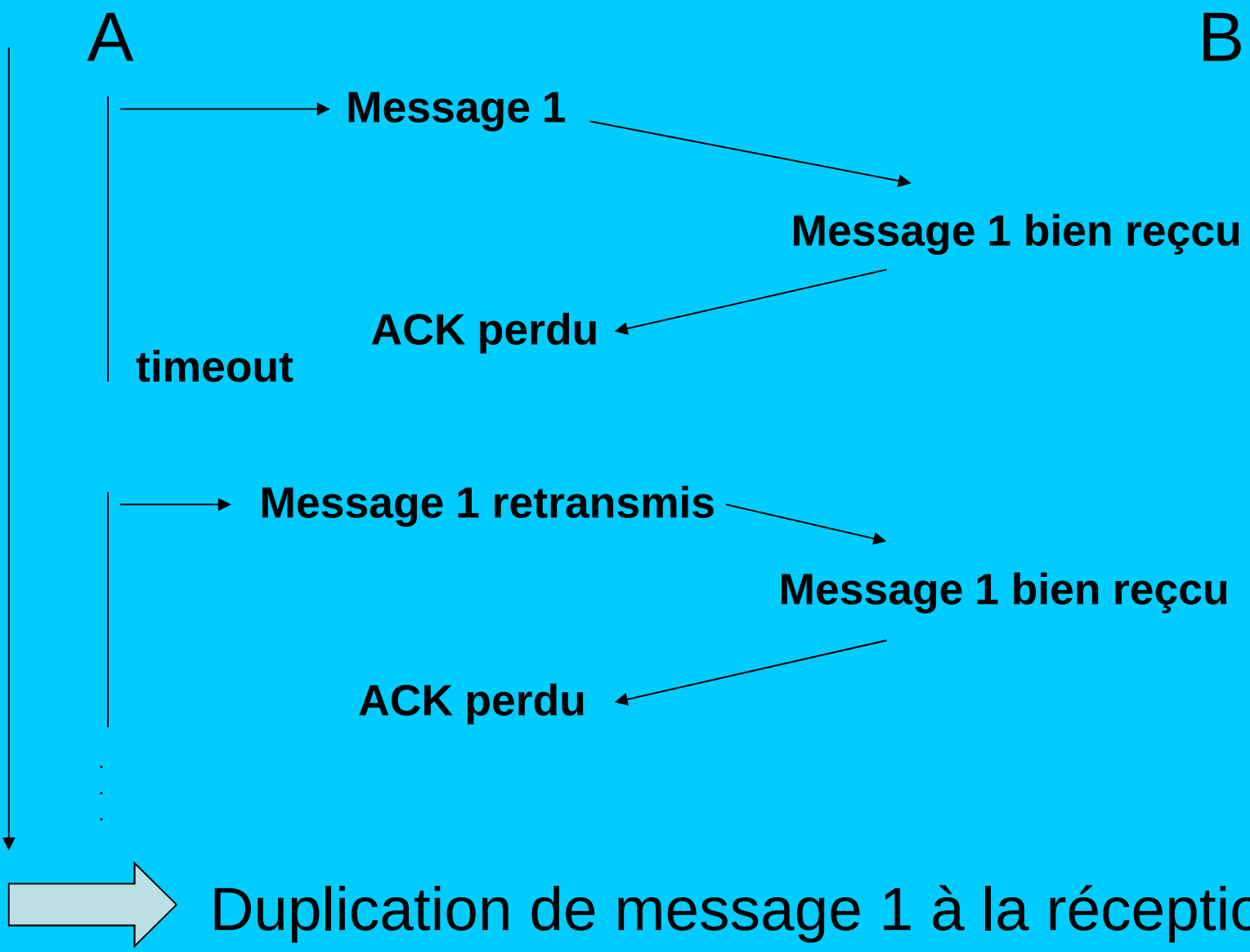
a- la C.R de A passe le paquet 1 à sa C.L.D. Le paquet est bien reçcu par B et transmis à la C.R de B. B envoie un AKT à A.

b- ACK perdu et n'arrive pas à A.

c- Le temporisateur, déclenché par la C.L.D de A lors de l'envoie de l'ACK,expire. LC.L.D suppose que le paquet à été endommagé ou perdu et elle envoie de nouveau une autre trame contenant le paquet 1.

d- La trame dupliquée arrive correctement à la C.L.D de B et est transmisse à la C.R.

Si A envoie un fichier à B, une partie de fichier sera dupliquée (la copie du fichier reçu par B n'est pas correcte et l'erreur n'a pas été Détectée). Le protocole, ainsi défini, ne convient pas.



Duplication de message 1 à la réception

Le récepteur doit pouvoir détecter une trame dupliquée.

L'émetteur place un numéro de séquence dans l'entête de chaque trame.

Le récepteur n'a qu'a vérifié le numéro de séquence de chaque trame arrivée et savoir s'il s'agit d'une nouvelle ou d'une trame dupliquée.

Garder le Header de la trame court, donc un bit suffit pour coder le numéro de séquence. Lorsqu'une trame arrive avec un bon numéro de séquence, le récepteur l'accepte et l'a transmis à la C.R. Le numéro de séquence est incrémenté de 1 modulo 2 (0 devient 1 et 1 devient 0).

Comme dans les protocoles déjà vus, la transmission des données se fait dans un seul sens.

Ce protocole prévoit la perte éventuelle des trames, grâce à l'utilisation de temporisateurs. La durée d'un temporisateur doit être suffisamment longue pour permettre l'arrivée des acquittements.

Les protocole dans lesquels l'émetteur attend un acquittement avant de transmettre les trames suivantes sont désignés par l'acronyme PAR *(Positive Acknowledgement with Retransmission).*

```
const   MaxSeq = 1;
type event = (trame_ok, trame_err, timeout) ;

Procedure sender3;
    var Nextout : NumSequence ; { numéro de la prochaine trame}
        s : trame ;
        buffer : message ;
        e : event ;
Begin
    Nextout := 0 ; { initialisation }
    FromHost(buffer) ; { chargement du premier paquet }
    repeat
        s.info := buffer ; { construction d'une trame }
        s.seq := Nextout ; { insertion du numéro dans la trame }
        send(s) ; { envoie de la trame }
```

```
        StartTimer(s.seq) ; { temporisateur déclenché }
        wait(e) ; { tous les événements sont possibles }
        if      (e = trame_ok)      then
            begin { un acquittement arrive }
                FromHost(buffer) ;
                Nextout := Nextout ⊕ 1 ; { incrémentation du
                                           numéro de séquence}
            end ;
        until   FinDeTemps ;
End ;  { sender3 }
```

```
Procedure receiver3 ;
    var Nextin : NumSequence ; { Num Seq Attendu }
        r, s : trame ; { variables temporaires }
        e : event ;
Begin
    Nextint := 0 ;
    repeat
        wait(e) ; { possiblités : trame_ok, trame_err }
        if  (e = trame_ok)   then
        begin  { arrivée d'une trame correcte }
            get(r);  { acquisition de la trame }
            if ( r.seq = Nextin ) then
                begin { c'est bien la trame attendue }
```

```
            ToHost(r.info) ; { transmission du paquet à la C.R}
             Nextin := Nextin ⊕ 1 ; { attendre le numero
                                                suivant }
        end ;
        send(s) ; { envoie d'un acquittement }

    end;
    until FinDeTemps ;
End ; { receiver3 }
```

Après transmission d'une trame, le temporisateur est déclenché. Sa durée doit être suffisante pour que la trame puisse arriver à destination et être traitée dans le pire des cas (récepteur très chargé) et pour permettre ensuite à l'émetteur de recevoir l'ACK.

Si le temporisateur expire, c'est que la trame ou l'ACK ont été perdus. Dans ce cas la trame doit être retransmise.

Après déclenchement de temporisateur, l'émetteur attend soit :

- **l'arrivée d'une trame d'acquittement correct**
- **l'arrivée d'une trame incorrecte d'acquittement**
- **l'expiration de temporisateur**

Si un ACK correct arrive, l'émetteur cherche le paquet suivant dans la couche Réseau et le met dans une mémoire en écrasant le paquet précédent. Il incrémente le numéro de séquence.

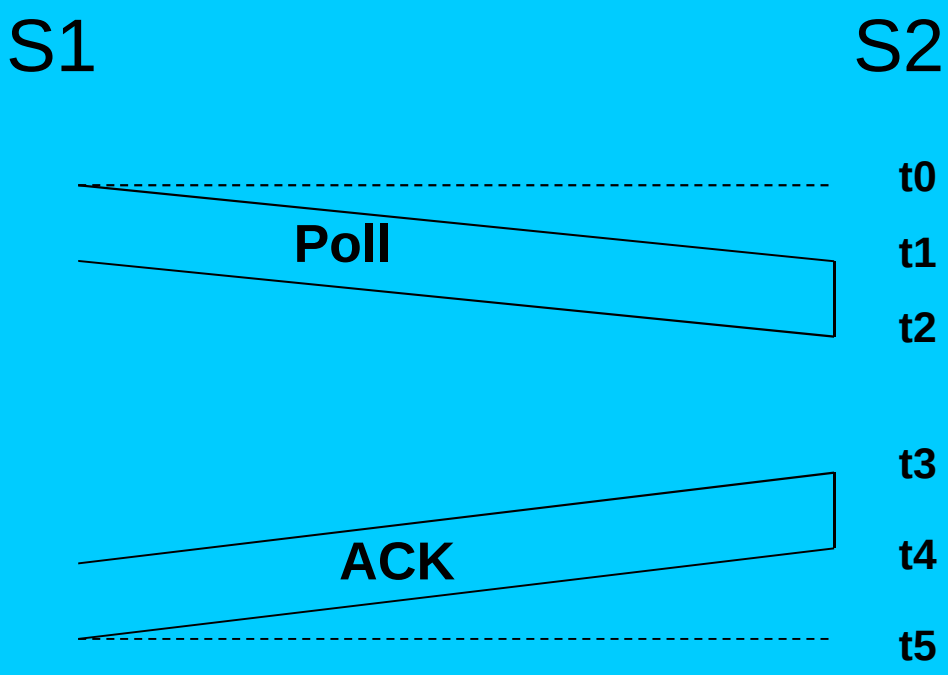
Si une trame erronée arrive ou si aucune trame n'arrive, on ne change ni numéro de séquence ni le contenu de la mémoire. Donc retransmission de la même trame.

À la réception d'une trame du coté du récepteur, ce dernier vérifie son numéro de séquence pour savoir s'il s'agit ou non d'une trame dupliquée. Si ce n'est pas le cas il la transmis à la couche réseau.

Les trames dupliquées ne sont pas transmis à la C.R.

Utilisation d’un Canal

Poll : message spécial envoyé par une station primaire pour demander à une station secondaire des données.



t1 – t0 = t5 - t4 : temps de propagation = t_{prop}
t2 – t1 = temps de transmission de Poll = t_{pol}
t3 – t2 = temps de traitement d'un Poll = t_{proc}
t4 – t3 = temps de transmission d'un ACK = t_{ack}

l'émetteur envoie une trame et attend un ACK (Stop-Wait).

Soit un message composé de n trames : m = (f1, …, fn).

L'envoie du message m se fait comme suit :

S1 envoie un Poll à S2
S2 répond avec f1
S1 envoie un ACK
S2 envoie f2
…
S2 envoie fn
S1 envoie un ACK

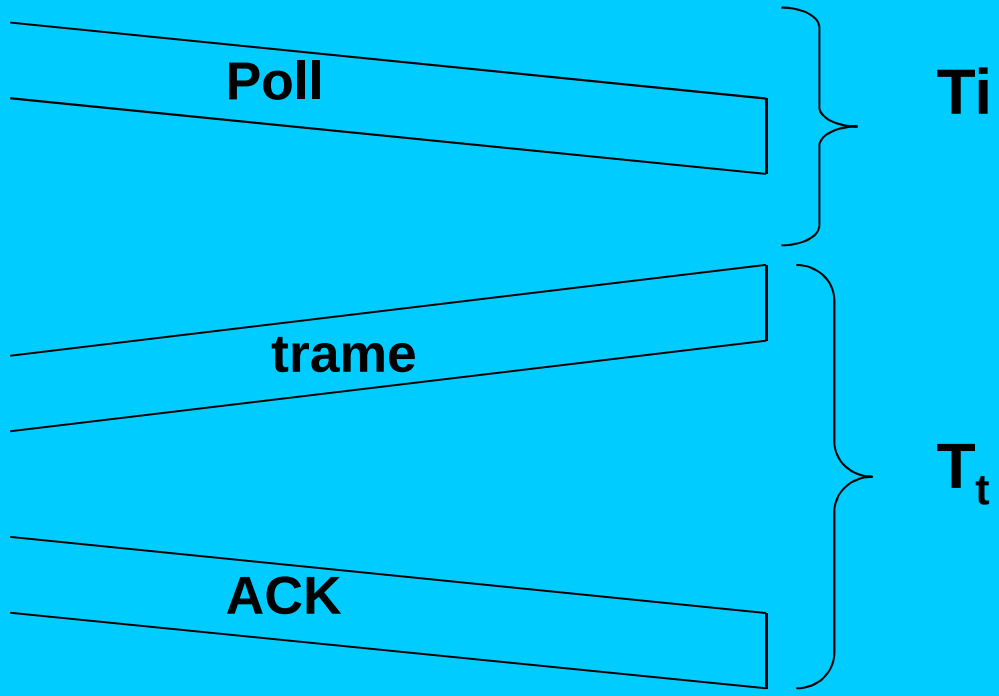
Le temps nécessaire pour envoyer le message m est :

$T = Ti + nT_t$ avec

Ti = temps pour effectuer un Poll = $t_{prop} + t_{pol} + t_{proc}$

T_t = **temps pour envoyer une trame**

= $t_{trame} + t_{prop} + t_{proc} + t_{prop} + t_{ack} + t_{proc}$



Ti est relativement petit quand on a une large séquence de trames. Donc Ti est négligeable.

Supposons que le temps de traitement est négligeable et que la taille de la trame acquittement est très petite par rapport à celle de données. Donc :

$$T = n(2\, t_{prop} + t_{trame})$$

Du temps T, seulement $n * t_{trame}$ est réellement utilisé pour transmettre le message m. Le reste c'est de l'OverHead.

Ainsi, le taux d'utilisation du Canal est donné par :

$$U = \frac{(n * t_{trame})}{n(2\, t_{prop} + t_{trame})}$$

si on prend $a = t_{prop} / t_{trame}$, alors $U = 1 / (1+2a)$

Si u ➔1 (c'est-à-dire a ➔ 0 ($t_{prop} << t_{trame}$))
alors le canal est presque utilisé à 100%

Exemple :

$$t_{prop} = \frac{\text{d (longueur du canal)}}{\text{V (vitesse de propagation)}} \quad , \quad t_{trame} = \frac{\text{L (Longueur d'une trame)}}{\text{B débit du canal}}$$

donc $a = B*d/V*L$

B exprimé en bits/s
d exprimé en mètres
V exprimé en m/s
L exprimé en bits

l'expression (B*d/V) mesure la longueur de médium en Bits. a est donc représenté par la longueur du médium comparée à celle de la trame en bits.

Exemples : $V = 3 * 10^8$ m/s pour l'air

$V = 2 * 10^8$ m/s pour le câble

1- Cas d'un satellite avec un Aller/Retour = 270 ms, avec un débit pratique de 56 Kbps.

Considérons des trames de 4000 bits. U?

t_{trame} = 4000 / 56000 = 71 ms donc a = 270 / 71 = 3.8

et par la suite : U = 1 / (1+ 2*3.8) = 0.12

Seulement une utilisation de 12% du canal.

2- cas d'un réseau local

La longueur des canaux varie entre 0.1 et 10 Km, avec des débits de 0.1 à 10 Mbps et une vitesse de $2 * 10^8$ m/s. Les trames de longueur 500 bits.

Les valeurs de a sont entre 0.01 et 0.1

Donc les valeur de U sont entre 0.83 et 0.98

<u>Problème des protocoles de type « Stop-Wait »</u>

Une seule trame peut être en circulation sur le canal.

Dans le cas ou a > 1 (long du canal en bits supérieur à celui de la trame), l'utilisation du canal sera très faible.

Solution : avoir plusieurs trames qui circulent sur le canal à la fois.

Communication entre A (source) et B (destination) :

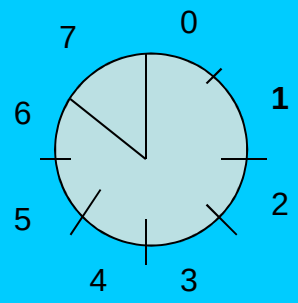
- B va réserver 7 tampons (buffers), au lieu de 1, pour la réception.
- A peut envoyer 7 trames sans attendre un ACK.
- On utilise les numéros de séquences de 0 à 7 (mod 8).
- B acquitte une trame en envoyant le numéro de la séquence attendue.

C'est un protocole à fenêtre d'anticipation de langueur 7 ou à fenêtre coulissante (Sliding Windows).

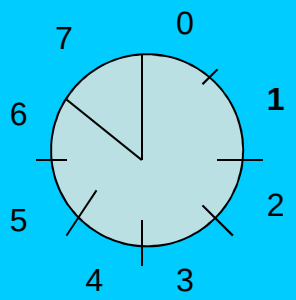
Un protocole à fenêtre d'anticipation de langueur 1 est de type « envoyer et attendre » : l'émetteur envoie une trame et attend son acquittement avant d'envoyer la suivante.

Exemple :

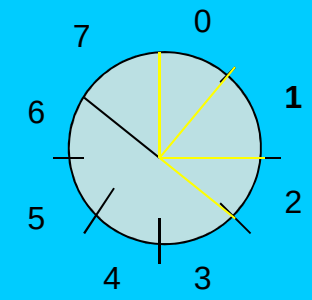
Fenêtre d'anticipation de taille 7 avec un numéro de séquence codé sur 3 bits.

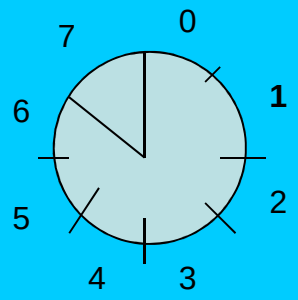


Le source A peut envoyer 7 trames

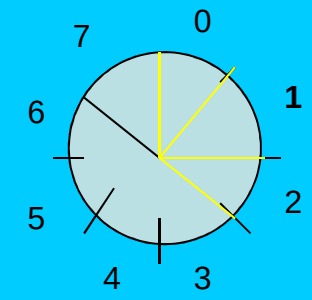


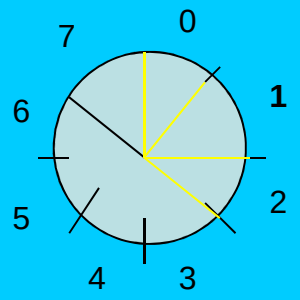
Le récepteur B est prêt à recevoir 7 trames



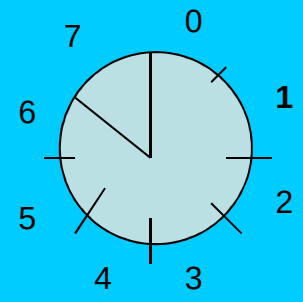


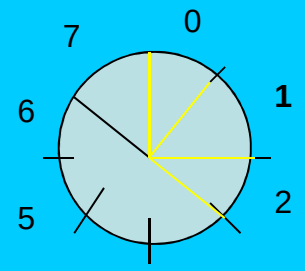
A envoie 3 trames 0, 1 et 2





B reçois les trois trames



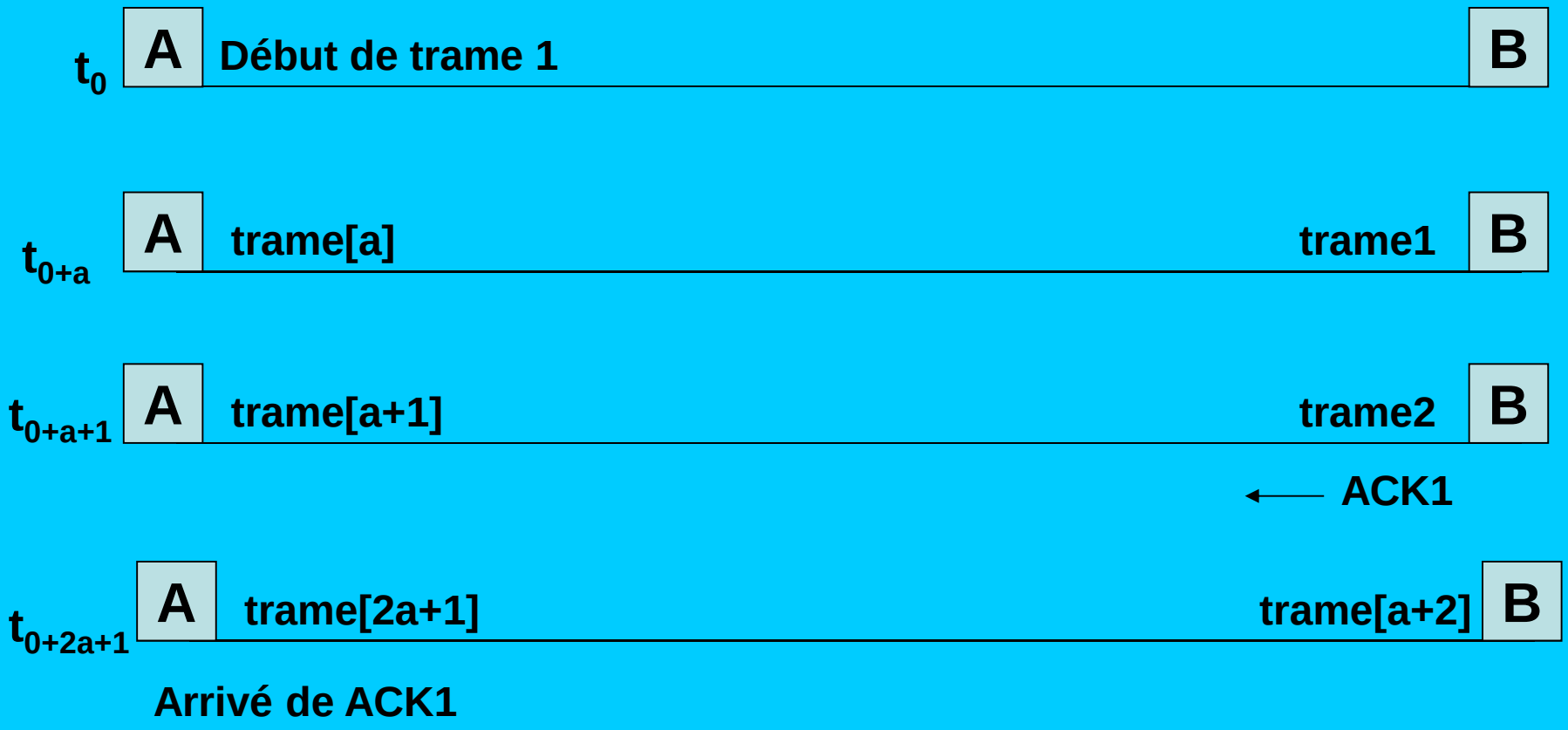


Après réception des trois acquittements, A est de nouveau prêt à envoyer jusqu'à 7 trames.

Remarque :

B peut faire un contrôle de flux en limitant la taille de sa fenêtre.

Utilisation de la ligne dans le cas de la Fenêtre coulissante :



[a] plus petit ou égal à a (longueur en bits du canal)

$a = t_{prop} / t_{trame}$,

Considérons que : t_{trame} = unité = 1 , donc $a = t_{prop}$
Cela signifie : si A envoie une trame à t_0 alors le début de la première trame arrive à t_{0+a}
À t_{0+a+1} la première trame est entièrement reçue.

On a deux cas : (N taille de la fenêtre)
a- N > 2[a]+1 , scénario de la page précédente
b- N < 2[a]+1 , même scénario sauf au niveau suivant :

Arrivé de ACK1

Lorsque A envoie plusieurs trames , l'une des deux situations se présente :

- **Dans le cas N > 2[a]+1, l'ACK1 arrive à A avant qu'il expédie toute la fenêtre.**

- **Dans le cas ou N < 2[a]+1, A a expédié toute la fenêtre à t_{0+N} et attend jusqu'à $t_{0+2[a]+1}$ pour envoyer d'autres trames.**

L'utilisation u canal est donc :

$$U = \begin{cases} 1 & \text{si } N > (2a+1) \quad \text{utilisation à 100\%} \\ N / (2a+1) & \text{si } N < (2a+1) \end{cases}$$

Si le numéro de séquence est codé sur n bits, alors on peut avoir une fenêtre de taille maximale égale à $2^n - 1$ et non pas 2^n.

Notons qu'il peut y avoir à chaque instant $2^n - 1$ (MaxSeq) trames non acquittées et non pas 2^n trames alors qu'on dispose de 2^n (MaxSeq+1) nombres pour numéroter les trames : 0, 1, 2, …, $2^n - 1$ (MaxSeq).

Afin de comprendre la raison de cette restriction, considérant le scénario suivant, avec MaxSeq = 7 :

1- L’émetteur envoie les trames de 0 à 7.

2- L’émetteur reçoit un acquittement pour la trame 7.

3- L’émetteur envoie une nouvelle série de huit trames portant les numéro de 0 à 7.

4- De nouveau, l’émetteur reçoit un acquittement de la trame 7.

Ambiguïté : Est-ce que toutes les trames appartenant au second envoi sont-elles arrivées ou bien ont-elles étés perdues?

Dans les deux cas, l’émetteur acquittera la trame 7. Pour Cette raison le nombre de trames non acquittées est limité à $2^n - 1$ (MaxSeq).

Le Protocole HDLC (High level Data Link Control)

C’est l’un des protocole les plus utilisé pour le niveau lien.

Ce protocole est conçu pour répondre aux différents besoins de différents types de lien :

- Les liens points à points & multipoints

- Opération Half-Duplex et Full-Duplex

- Interaction Terminal-Ordinateur ou Ord-Ord

- liens à longues distances (Sat) & courtes distances

Objectifs du protocole HDLC

- Indépendance du code utilisé (ASCII, EBCDIC, …).
- Adaptable à tous types de lien.
- Efficacité : OverHead Minimum avec un bon contrôle de de flux et detection des erreurs.

HDLC est de la famille des protocole Orienté Bit

- ADCCP (Advanced Data Communication Control), ANSI
- LAP-B (Link Acces Procedure-Balanced), CCITT
- SDLC (Synchronous Data Link Control), IBN, SNA

Caractéristique de HDLC

a) HDLC définie trois types de stations :

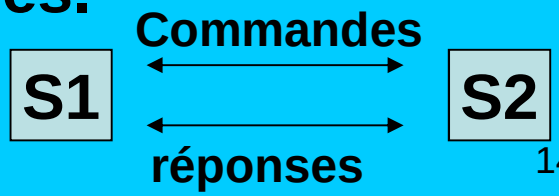
- Station primaire : contrôle les opérations de lien.

- Station secondaire : fonctionne sous le contrôle de la station primaire.

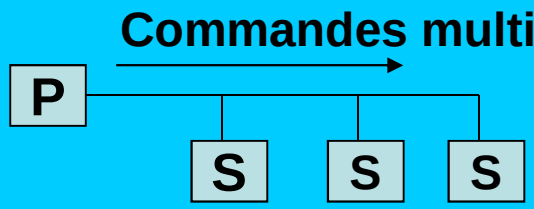
- Station combinée : peut émettre des commandes et des réponses (joue les deux rôles).

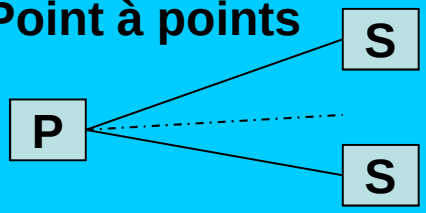
b) HDLC supporte deux types de configurations :

- Configuration balancée : utilisée en point à point et consiste en deux stations combinées.



- Configuration non balancée : utilisée en point à point et en multipoint entre une station primaire et plusieurs stations secondaires.



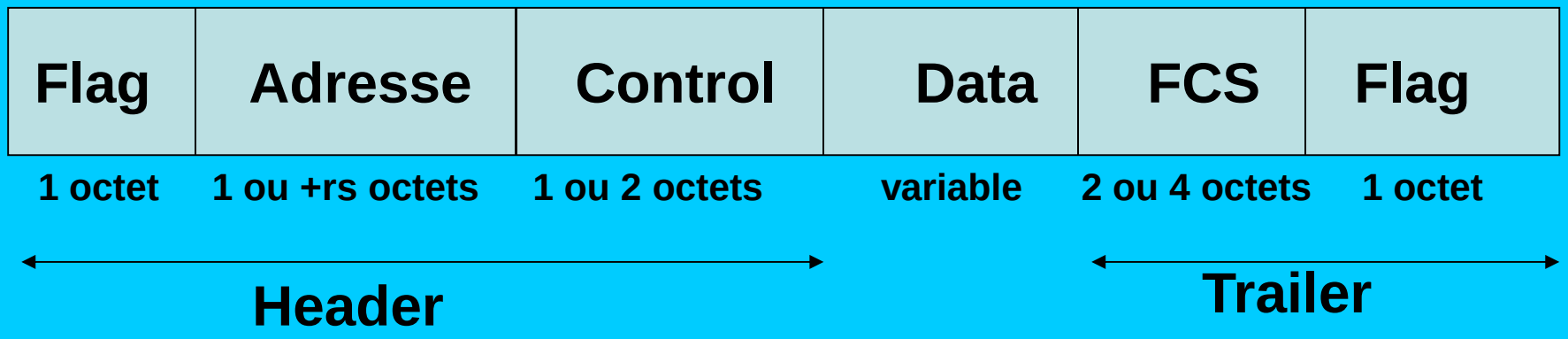


b- Mode de transfert des opérations

- NRM (Normal Respons Mode) : dans une configuration balancée, la station secondaire peut seulement transmettre des données en réponse à un Poll de la station primaire.

- ABM (Asynchronous Balanced Mode) : dans une configuration balancée, n'importe quelle station peut initier la transmission sans permission de l'autre.

Structure d'une trame dans le protocole HDLC



Dans le niveau lien on s'intéresse au prochain nœud.
Adresse = Adresse du prochain nœud

Description des champs

1- Flag : délimiteur de la trame qui est formé de 8 bits (= 01111110). Un seul Flag peut être utilisé pour terminer une trame et commencer une autre

<u>Problème</u> : Le Flag appartient ou non à la trame?

Solution :

- L'émetteur insère un bit à zéro dans la trame à l'exception du Flag.

- le récepteur examine la trame est enlève les extras bits.

Si une trame = 01111101111011110110, le récepteur enlève les deux extraits bits et on a la trame = 01111111011111110

2- Adresse souvent on utilise 8 bits pour identifier la destination, l'adresse 11111111 désigne l'adresse de tout le monde (utiliser pour la diffusion).

3 - Champ de contrôle : Il existe 3 types de trames HDLC

- I-Frame : Trame de données

- S-Frame : Trame de supervision utilisée pour le contrôle de flux et d'erreur.

- U-Frame : (Unumbred Frame) sert à d'autres fonctions de contrôle (exp : Poll).

Ce champ de contrôle contient :

- 1 ou 2 bits pour indiquer le type de la trame.

- 3 bits pour les numéros de séquences.

4- Le champ FCS : CRC-16 ou CRC-32

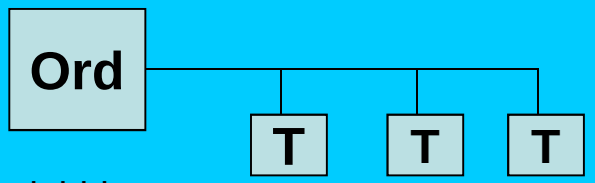
HDLC est une proposition de l'ISO suite à des modifications apportées sur ADCCP adopté par AINSI, qui est aussi issu de SDLC de SNA et d'IBM.

Multiplexage

Définition :

Le multiplexage est le partage de la capacité d'un canal par deux ou plusieurs stations.

Exemple : L'ordinateur et les terminaux partagent la même ligne.



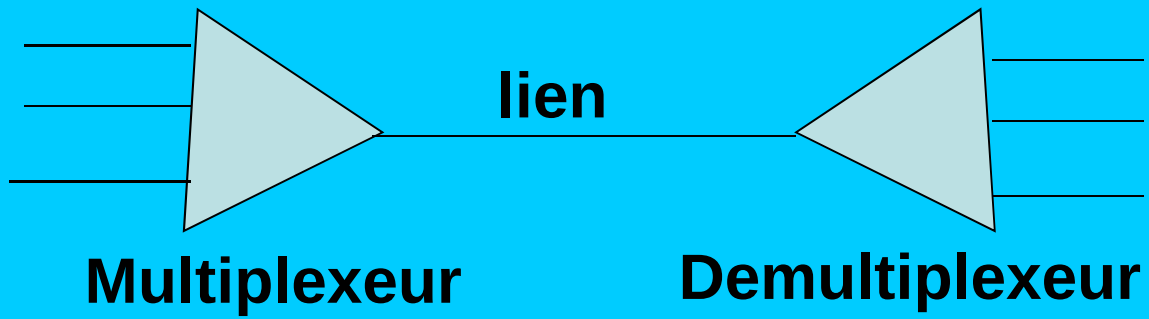
Il existe trois types de multiplexage :

1- FDM (Frenquency Division Multiplexing)

2- TDM (Synchronous Time Dvision)

3- TDM statique (Asynchronous TDM)

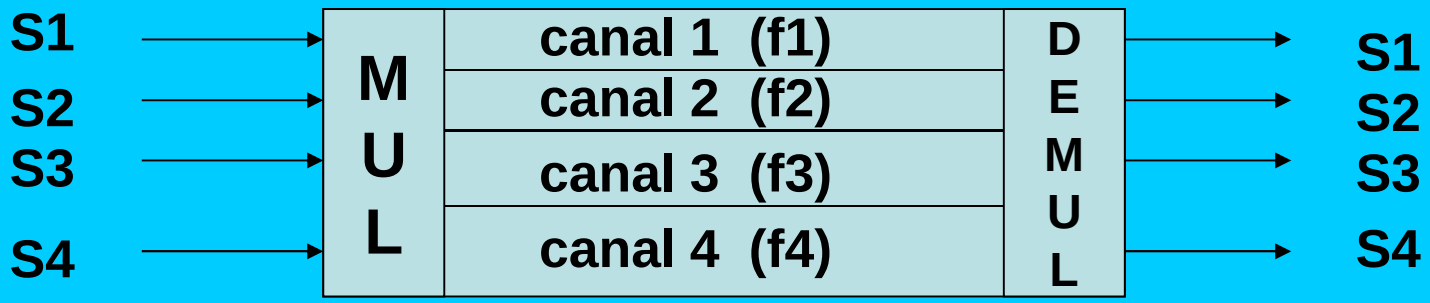
Principe :



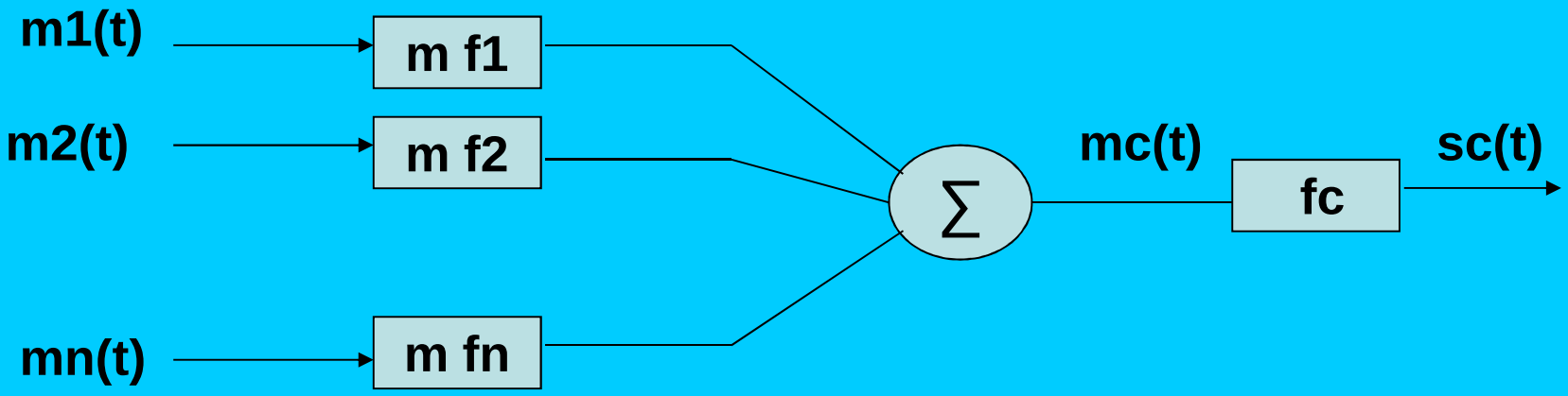
1- FDM

La bande passante de médium doit être plus grande que celles des signaux à transmettre.

Exemple :



Une certaine bande sépare les canaux pour éviter les interférences.

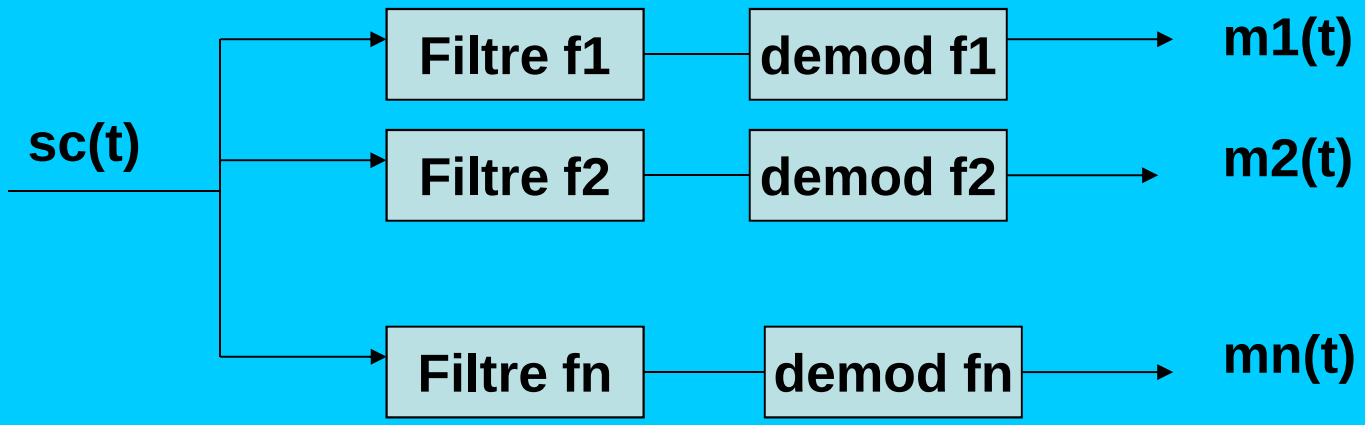


- mi(t) sont des signaux à multiplexer sur une même ligne.

- Chaque signal mi(t) est modulé sur une onde porteuse de fréquence fi.

- Le signal analogue mc(t) est la somme des si(t).

- mc(t) est ensuite modulé sur une autre fréquence fc pour produire sc(t).

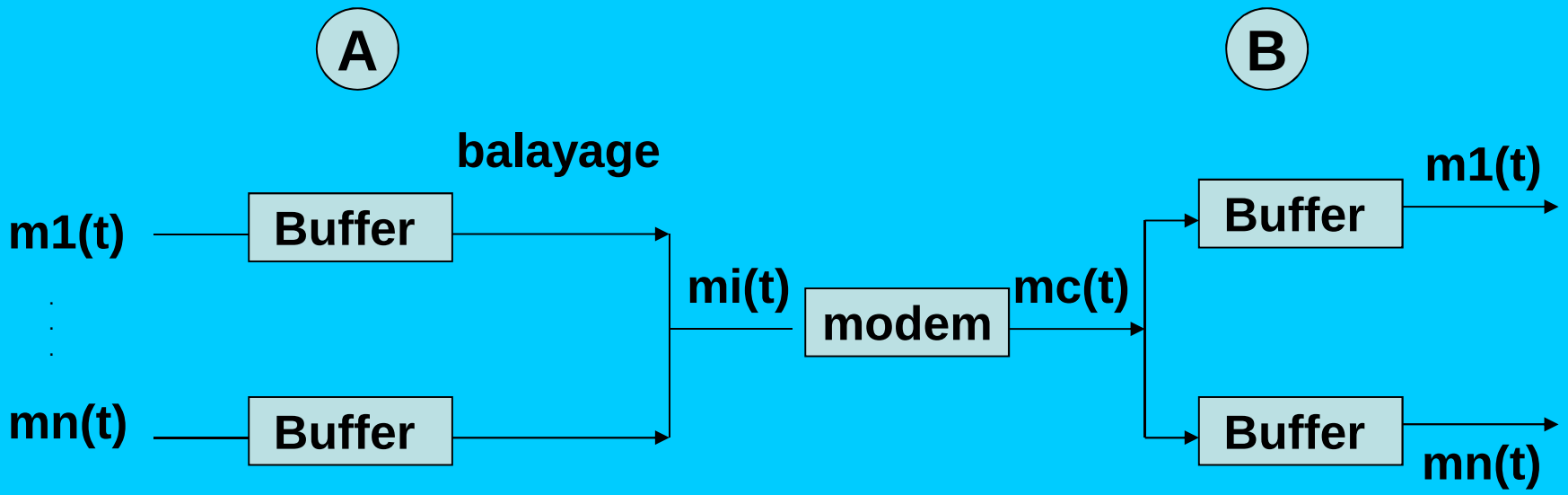


2- TDM Synchrone

- Les données provenant de chaque source sont brièvement stockées un tampon (de longueur 1 ou 8 bits).
- Ces tampons sont balayés séquentiellement pour former une chaîne m(t).
- m(t) est introduit ensuite dans un modem.

Les données transmises auront la forme

Exemple :



Remarque :

Le slot du temps associé à chaque source est transmis même si la source n'a pas de données à transmettre (gaspillage de la bande passant).

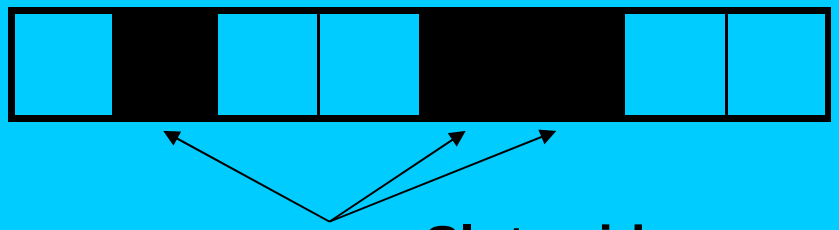
3- TDM statique

<u>Principe</u> : allocation dynamique des slot du temps sur demande.

- On a k slots disponibles pour n lignes

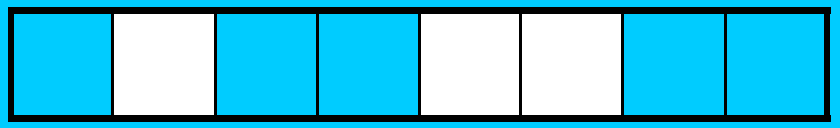
- on balaye les buffers pour collectionner les données et on construit la trame à transmettre.

- Le Demultiplexeur reçoit la trame et distribue les données aux buffers de sortie.



Slots vides

TDM Synchrone



TDM statique

Remarque :

Si les n station émettent les données aux même temps et si n > k, alors la capacité est insuffisante, donc pertes des données.

Le multiplexeur doit disposer d'une mémoire supplémentaire.

Couche réseau

En général les réseaux sont classés en deux catégories :

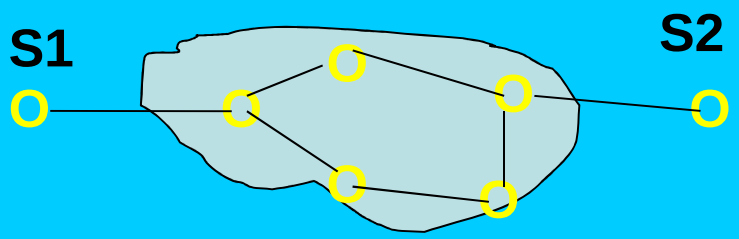
1) Réseaux à commutation :

- **Commutation de circuit**
- **Commutation de message**
- **Commutation de paquet**

2) Réseaux à diffusion :

- **Radio**
- **Satellite**
- **Réseaux locaux**

Dans les réseaux à commutation les nœuds en communication ne sont pas nécessairement adjacents, on peut avoir des nœuds internes.



Dans les réseaux à diffusion chaque station est attachée à un Émetteur/Récepteur qui communique avec d'autre stations via un médium partagé.

Commutation de circuit

connexion physique entre deux stations.

1- Établissement de circuit (réservation de la voie)
2- Transfert de données
3- Déconnexion (libération de la voie)

Exemple : téléphone.

Inconvénient : gaspillage de la capacité du canal s'il n'y a pas de transfert continu.

Avantage : Délai dans chaque nœud très négligeable.

Commutation de message

L’unité de données est un message (exemple : fichier).

- Pas de chemin préétabli.

- Dans chaque message on ajoute l’adresse de la destination (la seule information connue à l’avance).

- Chaque message passe en entier d’un nœud à l’autre.

Remarque :

Chaque nœud doit disposer d’une mémoire de stockage pour pouvoir recevoir un message en entier.

Avantages :

- **Meilleur utilisation de la capacité de la ligne.**

- L'émetteur et le récepteur n'ont pas besoin d'être disponibles aux même temps pour échanger des données.

- Un message peut être envoyé vers plusieurs destinations à la fois.

Inconvénients :

Le délai de stockage et de traitement de chaque message dans un nœud est relativement grand.

Donc, ce genre de commutation n'est pas approprié aux applications interactives (les applications temps réel qui sont sensibles à la notion du temps).

Commutation de paquet:

Principe : même que celui de la commutation de message sauf que la longueur de l'unité de données est limitée. Il s'agit des messages très courts (paquets).

Comment une suite de paquets traverse un tel Réseau ?

- **Mode Datagramme (mode non connecté) :**

- chaque paquet est envoyé et traité indépendamment des autres.

- Décision de routage pise pour chaque paquet.

- chaque paquet suit un chemin indépendamment des autres.

- Les paquets arrivent à destination en désordre.

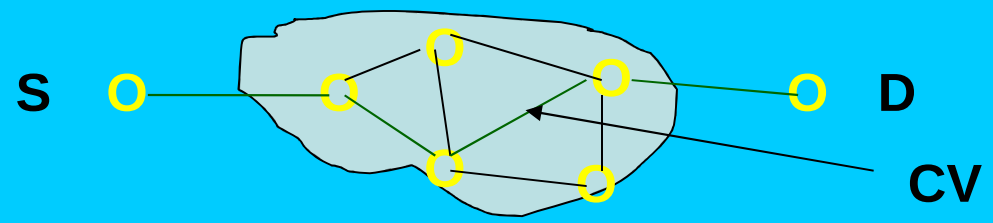
- Ce sont les fonctions de la couche Réseau et celles de la couche Transport qui réordonnent les paquets pour former le bon message à la destination.

- **Circuit Virtuel (mode connecté) :**

- un chemin est établie à l'avance (connexion logique). Aucune décision de routage n'est prise.

- Même principe que le système téléphonique.

- La technique d'inondation permet de diffuser un « Set Up Call » pour établir le chemin le moins court.



Exemple :

- **Pour les applications interactives (base de données), le Mode Circuit Virtuel est le mieux approprié.**
- **Pour le transfert de fichiers (FTP, Email), on peut utiliser le Mode Datagramme.**

Dans le Mode Datagramme, on a pas besoin de diffuser un « Set Up Call ». On envoie des paquets sans informer la destination. Ce mode se base sur l'état du réseau. En cas de congestion ou de panne dans un nœud interne, on choisit un chemin ne passant pas par ce nœud.

Pour les commutation à base de Circuit Virtuel, lorsqu'un nœud tombe en panne toutes les connexions logiques passant par ce nœud seront perdues (tombées).

Les performances de chacune de ces techniques dépendent de :

- Nombre de stations (Sources et destinations) en communication.

- Nombre de nœud (topologie).

- Charge du système (nombre total des paquets se trouvent dans le réseau à un instant données.

- Vitesse de traitements des nœuds.

- Taille des paquets.

Les éléments principaux d'un réseau à commutation de paquet sont :

- **Le routage : le réseau doit assurer l'acheminement des paquets d'un nœud à l'autre jusqu'à destination.**

- **Le contrôle de trafic : le réseau doit être maintenu dans un état stable.**

- **Le contrôle d'erreur : les paquets peuvent être perdus dans le réseau.**

Les techniques de routage

Algorithme de DijKstra (1959)

Problème :

Étant donné une source (nœud externe) . Trouver le chemin au coût minimum de ce nœud à chacun des autres nœud du réseau.

N : l'ensemble de tous les nœuds.

S : le nœud source

M : ensemble qui contient à tous moment l'ensemble des nœuds dont la distance minimale de la source est connue.

L(i,j) : coût du lien entres les nœuds i et j (L(i,j) = ∞ si l'arc (i, j) n'existe pas, i et j ne sont pas directement liés).

C(n) : coût du chemin à coût minimal de S à n.

Algorithme :

1) M = { S }

2) Pour chaque nœud n ∈ N – M faire :

C(n) = L(S, n)

3) trouver w ∈ N – M tel que C(w) est minimum

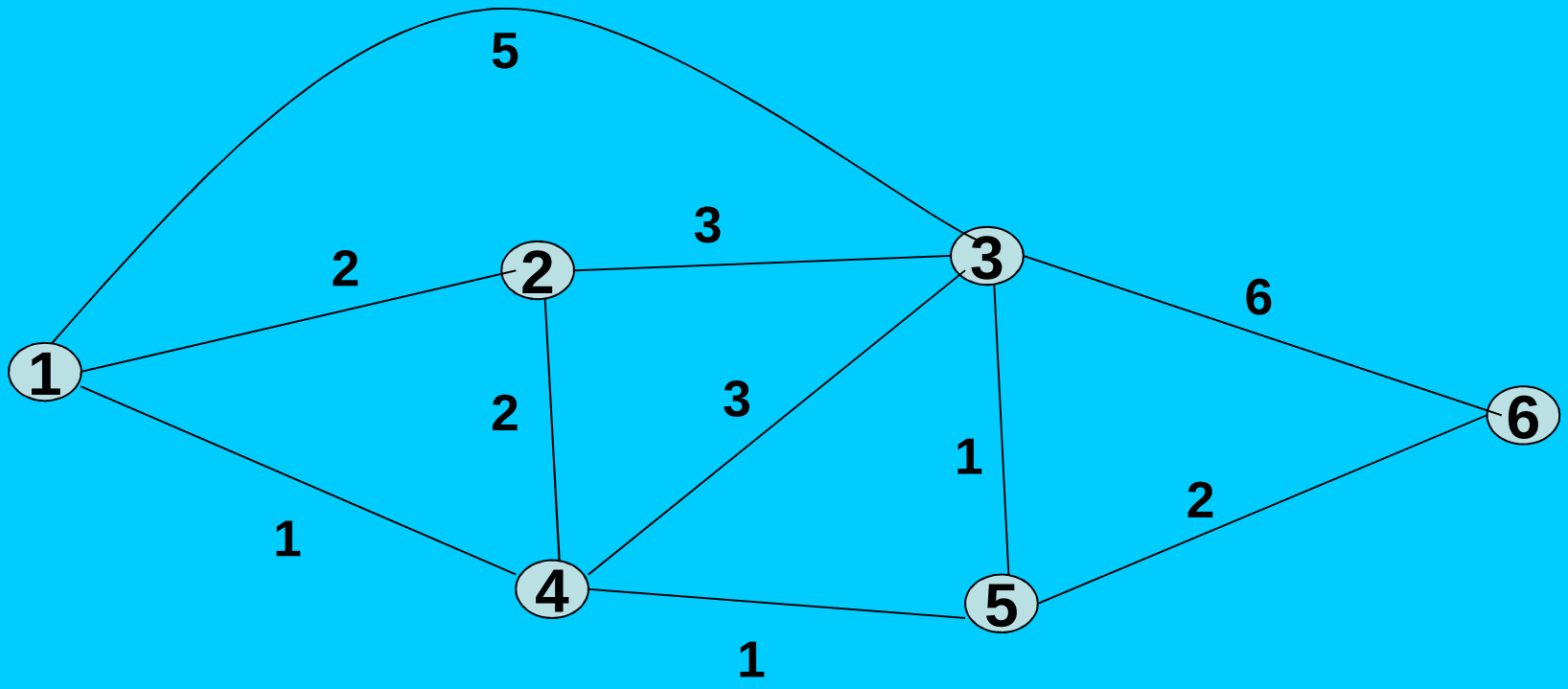
l'ajouter à M , M ← M + { w }

4) Pour chaque nœud n ∈ N – M faire

C(n) = min[C(n), C(w)+L(W,n)]

5) répéter 3 et 4 Jusqu'à ce que M = N.

Exercice :



Appliquer l'algorithme Jijkstra pour la cas ou S = 1

M	N – M	C(1)	C(2)	C(3)	C(4)	C(5)	C(6)
1	2, 3, 4, 5, 6		$2_{(1,2)}$	$5_{(1,3)}$	$1_{(1,4)}$	∞	∞
1, 4	2, 3, 5, 6		$2_{(1,2)}$	$4_{(1,4,3)}$	$1_{(1,4)}$	$2_{(1,4,5)}$	∞
1, 4, 2	3, 5, 6		$2_{(1,2)}$	$4_{(1,4,3)}$	$1_{(1,4)}$	$2_{(1,4,5)}$	∞
1, 4, 2, 5	3, 6		$2_{(1,2)}$	$3_{(1,4,5,3)}$	$1_{(1,4)}$	$2_{(1,4,5)}$	$4_{(1,4,5,6)}$
1, 4, 2, 5, 3	6		$2_{(1,2)}$	$3_{(1,4,5,3)}$	$1_{(1,4)}$	$2_{(1,4,5)}$	$4_{(1,4,5,6)}$
1, 4, 2, 5, 3, 6	Φ						

Algorithme de ford :

Consiste à trouver le chemin à coût minimum pour se rendre à un nœud à partir de tous les autres nœuds.

Étant donné une destination D. Trouver le chemin à coût minimum de n'importe qu'il autre nœud pour se rendre à D.

N ensemble des nœuds du réseau.

D le nœud destination.

L(i,j) : coût du lien entres les nœuds i et j (L(i,j) = ∞

si l'arc (i, j) n'existe pas, i et j ne sont pas directement connétables)

C(n) : coût du chemin à coût minimal de D à n.

Algorithme :

1) C(D) = 0, pour chaque nœud n $\in$ N – { D }

faire : C(n) = ∞

2) Pour chaque nœud n $\in$ N – { D } faire :

$$C(n) = \min_{W \in N} [\, C(n), C(w) + L(n,W) \,]$$

(si le dernier terme est le minimum, alors le chemin de n à D sera le lien de n à w concaténé au lien w à D)

3) répéter « 2) » jusqu'à ce que aucun coût ne puisse changer.

Exercice :

Appliquer l'algorithme de Ford à la configuration précédente dans le cas ou D = 1.

Technique de Flooding (Inondation) :

- Un paquets est envoyé d'un nœud à tous ces voisins.

- A chaque nœud, le paquet reçu est retransmis dans toutes les lignes sauf celles par ou le paquet est arrivé.

Propriétés de Flooding :

- Toutes les routes sont essayées.

- Au moins une copie aura parcouru le chemin minimum

- On peut l'utiliser pour établir le chemin minimum (Set Up Call).

- Augmente la charge du réseau (Inc).

- Pour arrêter la retransmission des paquets, on peut inclure un compteur de « saut » dans chaque paquet. Il sera initialisé à une valeur max (exp : diamètre du réseau).

- On le décrémente à chaque passage par un nœud.

- Une fois le compteur est à zéro, on écarte le paquet..

Remarque :

Dans les techniques précédentes, la décision de routage ne tient pas compte des mesures ou des estimations sur le trafic dans le réseau.

Ces technique sont des algorithmes non adaptatifs.

Les algorithmes Adaptatifs :

1) Routage centralisé

- La construction des tables de routage est similaire au routage fixe (chaque nœud dispose d'une table).

- La différence est dans la construction des tables :

 # En routage fixe le chargement est manuel

 # Le routage centralisé fait appel à un RCC (Centre de Contrôle de Routage)

Chaque nœud envoie périodiquement des informations sur son état au RCC ; à savoir :

- **Liste des noeuds voisins opérationnels**
- **Langueur des files d'attente.**
- **Variation du trafic traité par les lignes depuis le dernier rapport.**

Le RCC recalcule les chemins optimaux et transmet de nouvelles tables aux différents noeuds.

<u>Avantages</u> :

* **décharge les nouds du calcul des tables.**
* **Le RCC dispose des informations complètes sur l'état du réseau.**

Inconvénients :

- RCC en panne

donc tous le réseau est affecté

- La collecte des information entraîne une surcharge du réseau.

- Congestion des lignes ramenant au RCC.

Algorithme « Back Word Learning »

Chaque paquet contient l'identité du nœud source et un compteur qui sera incrémenté à chaque saut (passage par un nœud).

Si un nœud reçoit un paquet sur la ligne i provenant du nœud B (compt = s), alors ce noeud sait que B ne peut pas être à plus de s sauts via a ligne i.

Si la distance qui existe dans la table est supérieure à s, alors s sera la nouvelle valeur.

Références

- Andrew Tanenbaum, ***Réseaux :*** *architectures, protocoles et applications*, InterEditions, 1990.

- Guy Pujolle, ***Les réseaux***, Eyrolles, 1997.